cadence™

Virtuoso Visualization and Analysis XL User Guide

Product Version IC23.1
November 2023

© 2023 Cadence Design Systems, Inc. All rights reserved.
Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Trademarks: Trademarks and service marks of Cadence Design Systems, Inc. contained in this document are attributed to Cadence with the appropriate symbol. For queries regarding Cadence's trademarks, contact the corporate legal department at the address shown above or call 800.862.4522.

All other trademarks are the property of their respective holders.

Restricted Permission: This publication is protected by copyright law and international treaties and contains trade secrets and proprietary information owned by Cadence. Unauthorized reproduction or distribution of this publication, or any portion of it, may result in civil and criminal penalties. Except as specified in this permission statement, this publication may not be copied, reproduced, modified, published, uploaded, posted, transmitted, or distributed in any way, without prior written permission from Cadence. Unless otherwise agreed to by Cadence in writing, this statement grants Cadence customers permission to print one (1) hard copy of this publication subject to the following conditions:

1. The publication may be used only in accordance with a written agreement between Cadence and its customer.
2. The publication may not be modified in any way.
3. Any authorized copy of the publication or portion thereof must include all original copyright, trademark, and other proprietary notices and this permission statement.
4. The information contained in this document cannot be used in the development of like products or software, whether for internal or external use, and shall not be used for the benefit of any other party, whether or not for consideration.

Disclaimer: Information in this publication is subject to change without notice and does not represent a commitment on the part of Cadence. Except as may be explicitly set forth in such agreement, Cadence does not make, and expressly disclaims, any representations or warranties as to the completeness, accuracy or usefulness of the information contained in this document. Cadence does not warrant that use of such information will not infringe any third party rights, nor does Cadence assume any liability for damages or costs of any kind that may result from use of such information.

Cadence is committed to using respectful language in our code and communications. We are also active in the removal and replacement of inappropriate language from existing content. This product documentation may however contain material that is no longer considered appropriate but still reflects long-standing industry terminology. Such content will be addressed at a time when the related software can be updated without end-user impact.

Restricted Rights: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in FAR52.227-14 and DFAR252.227-7013 et seq. or its successor

Contents

1
Overview of Virtuoso Visualization and Analysis XL 23
Licensing Requirements 25
Supported Data Formats 26
Opening the Virtuoso Visualization and Analysis XL 28
Opening the Tool in Stand-Alone Mode 28
Opening the Tool Using CIW 30
Opening the Tool Using Virtuoso Analog Design Environment 31
Closing the Virtuoso Visualization and Analysis XL 31
Saving a Session 31

2
Results Browser 33
Opening the Results Browser 35
Opening the Results Browser in Standalone Mode 35
Opening the Results Browser from Virtuoso 35
Opening the Results Browser from Virtuoso Analog Design Environment 35
Results Browser Graphical User Interface 37
Toolbar of the Results Browser 38
Top Panel of the Results Browser 40
Bottom Panel of the Results Browser 41
Opening a Results Directory 43
Setting the In-Context Results Directory 47
Selecting a Signal in the Results Browser 49
Plotting a Signal from the Results Browser 50
Plotting a Transient Signal Over a Time Range 54
Plotting an AC Signal over a Frequency Range 56
Plotting Parametric Sweep Data 58
Plotting the Difference of Two Signals 63
Plotting YvsY for Two Signals 65
Comparing Signals from Different Results Directories 68

© 2023 Cadence Design Systems, Inc.

Exporting Waveforms from a Single Results Directory . . . 71
Contents of the Exported VCSV File . . . 75
Exporting Waveforms from Multiple Results Directories . . . 77
Sending a Signal to Calculator from Results Browser . . . 79
Sending a Signal to Table from Results Browser . . . 81
Setting Colors to Signal Types . . . 83
Automatic Calculation of Two-Port Impedance Equivalent Parameters . . . 86
Searching for Signals in the Results Browser . . . 88
Examples . . . 89
Filtering Signals in the Results Browser . . . 93
Plotting Modes . . . 97

3
Virtuoso Visualization and Analysis XL Table . . . 99

Menu Bar of the Table . . . 101
Toolbar of the Table . . . 103
Opening the Virtuoso Visualization and Analysis XL Table . . . 104
Opening the Table from the Results Browser . . . 104
Opening the Table from the Graph Window . . . 106
Opening Table from the Calculator . . . 107
Saving Table Contents to a CSV File . . . 112
Exporting Contents of the Table . . . 113
Hiding Columns in the Table . . . 116
Revealing Columns in the Table . . . 116
Formatting Attributes of the Table . . . 118
Sorting Columns in the Table . . . 119
Transposing Rows and Columns of the Table . . . 120
Merging X-Axis Data in a Single Column . . . 120
Changing Column Colors in the Table . . . 121
Renaming Column and Row Headers of the Table . . . 121
Limiting the Number of Actions to Undo or Redo . . . 122
Filtering Table Data . . . 123
Changing the Format of Complex Data in the Table . . . 124
Resizing Columns and Rows of the Table . . . 126
Printing Contents of the Table . . . 127

© 2023 Cadence Design Systems, Inc.

Setting Printer Properties . . . 129

4
Working with Graphs . . . 133

About the Graph Window . . . 135
Opening the Graph Window . . . 136
Graphical User Interface . . . 138
Menu Bar . . . 139
Toolbars . . . 157
Status Bar . . . 163
Assistants . . . 163
Creating a Graph . . . 163
Working with Subwindows . . . 165
Customizing a Graph . . . 189
Determining the Active Window . . . 190
Setting the Graph Colors . . . 190
Renaming and Closing Window . . . 191
Handling Graph Objects . . . 191
Panning and Zooming Graphs . . . 192
Editing Graph Properties . . . 196
Setting Properties for Multiple Graphs . . . 199
Working with Graph Axis . . . 202
Editing Graph Axis Attributes . . . 202
Changing Digital Dependent Axis Properties . . . 206
Changing Axes Scale to Logarithmic . . . 206
Displaying X-Axis Labels in String Format . . . 207
Adding Multiple Y-Axes . . . 211
Changing Dependent Axis (Y-Axis) . . . 213
Plotting Multiple Signals on a Common Axis . . . 220
Merging Two Y-Axes . . . 222
Locking Graphs . . . 224
Evaluating Graph Expressions . . . 225
Working With Assistants . . . 225
Spectrum Assistant . . . 226
Browser Assistant . . . 240

© 2023 Cadence Design Systems, Inc.

Marker Toolbox Assistant . . . 241
Eye Diagram Assistant . . . 242
Working with the Eye Measurements Tab . . . 265
Working with the Eye Mask Tab . . . 288
Horiz Marker Table Assistant . . . 307
Trace Info Assistant . . . 307
Vert Marker Table Assistant . . . 308
Transient Measurement Assistant . . . 308
Direct Measurements Assistant . . . 321
Editing Properties of Direct Measurements . . . 358
Deleting Direct Measurements . . . 359
Customize Trace Groups Assistant . . . 359
Working with Workspaces . . . 365
Workspace Types . . . 365
Saving a Workspace . . . 366
Loading a Workspace . . . 367
Deleting a Workspace . . . 367
Setting the Default Workspace . . . 367
Showing and Hiding Assistants . . . 367
Working with Traces . . . 368
Filtering Trace Legends . . . 370
Some Useful Features . . . 382
Selecting Traces . . . 387
Using Trace Menu Commands . . . 388
Dragging Traces . . . 391
Hiding and Showing Traces . . . 391
Deleting Traces . . . 391
Setting Trace Properties . . . 392
Displaying Symbols on a Trace . . . 395
Exporting a Trace . . . 396
Working with Strips . . . 398
Splitting Strips . . . 398
Combining Graph Strips . . . 405
Moving Traces . . . 406
Copying Traces . . . 407
Setting Strip Properties . . . 407

© 2023 Cadence Design Systems, Inc.

Locking and Unlocking Strips . . . 410
Working with Sweeps . . . 411
Working with Graph Labels . . . 422
Deleting Graph Labels . . . 423
Creating Multiple Graph labels . . . 424
Plotting WREAL Signals . . . 424
Plotting YvsY Graph . . . 428
Plotting Normal Quantile Graphs . . . 430
Saving and Loading Graphs . . . 430
Saving a Graph . . . 431
Loading a Graph . . . 433
Saving a Graph as an Image . . . 434
Reloading Graphs . . . 440
Reloading Graph When Opened in Stand-Alone Mode . . . 441
Reloading Graph When Opened From Within ADE Explorer, or ADE Assembler . . . 447
Disabling Trace Reload . . . 447
Graph Settings Retained During Reloading . . . 448
Printing Graphs . . . 449
Supporting Mixed Signals . . . 456
Loading Vector Files in Virtuoso Visualization and Analysis XL . . . 458
Converting an Analog Signal into a Digital Signal . . . 465
Converting a Digital Signal to an Analog Signal . . . 467
Generating Derived Plots . . . 469
Plotting Histogram . . . 471
Working with Buses . . . 477
Creating a Bus . . . 477
Plotting Parametric Buses . . . 480
Creating a Mnemonic Map . . . 482
Applying a Mnemonic Map to a Bus . . . 486
Importing Mnemonic Maps from SimVision . . . 487
Setting Bus Properties . . . 490
Expanding a Bus . . . 491
Collapsing a Bus . . . 492
Exporting a Bus . . . 492
Sending a Bus Signal to Calculator . . . 493
Sending a Bus Signal to Table . . . 493

© 2023 Cadence Design Systems, Inc.

Working with Markers . . . 493
Adding Markers . . . 495
Marker Labels . . . 497
Editing the Marker Label of a Point Marker . . . 499
Adding a Point Marker . . . 501
Adding a Fixed-Y Marker . . . 502
Adding a Horizontal Marker . . . 504
Adding a Vertical Marker . . . 505
Adding an Edge Marker . . . 506
Adding a Period Marker . . . 508
Adding a Dx/Dy Marker . . . 508
Adding a Delta Marker . . . 508
Repositioning Marker Labels . . . 509
Setting Marker Properties . . . 509
Snapping Markers . . . 514
Customizing Markers . . . 515
Working with Delta Markers . . . 521
Working With Edge Markers . . . 531
Working with Circular Graphs . . . 539
Creating a Circular Graph . . . 539
Adding Markers on Circular Graphs . . . 548
Snapping Markers on Circular Graphs . . . 554
Plotting Circular Graphs with Different Axis Units . . . 557
Zooming Circular Graphs . . . 557
Editing Circular Graph Properties . . . 557
Filtering Data Using the Frequency Slider Bar . . . 559
Setting Smith Grid Properties . . . 562
Setting Bindkeys . . . 565
Working with Tabular Graph . . . 569
Creating a Tabular Graph . . . 569
Editing Tabular Graph Properties . . . 572
Using Tabular Trace Properties Commands for a Column . . . 573
Using Tabular Trace Properties Commands for a Row . . . 574
Customizing Display of S-Parameter Data . . . 575
Graph Summary Label . . . 577
Hiding a Graph Summary Label . . . 581

© 2023 Cadence Design Systems, Inc.

Viewing Graph Summary Information . . . 582
Editing the Properties of a Graph Summary Label . . . 584
Annotating Scalar Outputs for Single-Point Simulation . . . 585
Hiding the Summary Label for Scalar Outputs . . . 590
Viewing Summary Information of Scalar Outputs . . . 591
Editing Properties of the Summary Label . . . 593
Graph Properties Form (Circular Graphs) . . . 594

5
Working with the Calculator . . . 601

About the Calculator . . . 602
Opening the Calculator Window . . . 604
Using the Calculator Graphical User Interface (GUI) . . . 607
Menu Bar . . . 609
Toolbars . . . 612
Buffer . . . 621
Assistants . . . 625
Setting Calculator GUI Using .cdsenv Variables . . . 634
Working with Expressions . . . 635
Selecting Signals and Waveforms to Build Expressions . . . 636
Building Expressions in the Buffer . . . 638
Editing an Expression in Buffer . . . 646
Building Dependent Expressions . . . 647
Saving Expressions . . . 649
Dragging Expressions across Buffer and Assistants . . . 651
Loading Expressions from an Expression File . . . 651
Evaluating and Plotting Expressions . . . 652
Displaying Results in a Table . . . 653
Creating Custom Calculator Functions . . . 655
Working with the Calculator in ADE . . . 665
Opening the Calculator using ADE . . . 665
Additional Features in ADE . . . 671
Creating Expressions For DC Operating Point Functions . . . 674
Working with Expressions in ADE . . . 678
. . . 679

© 2023 Cadence Design Systems, Inc.

6
Distributed Plot 681

Advantages of Distributed Plot 681
Enabling or Disabling Distributed Plot 683
Specifying Linger Time 685
Opening the Distributed Plot Assistant 686
Opening the Distributed Plot Remotely 688

7
Virtuoso Visualization and Analysis XL Forms 691

Apply Range Form 693
Create Bus Form 694
Create Graph Marker Form 695
Point Tab 695
FixedY Tab 696
Horizontal Tab 697
Vertical Tab 698
Edge Tab 698
Digital Bus Trace Properties Form 699
The Edit Marker Label form 700
The Export Waveforms Form 702
The Format Attributes Form 704
The Graph Properties Form (Circular Graphs) 706
The General Tab 706
The Grid Tab 708
The Annotations Tab 710
The Markers Tab 711
The Horizontal Marker Properties Form 713
The Point Marker Properties Form 716
The Point Marker Properties for Circular Graph Form 720
The Polar Reference Point Marker Properties Form 723
The Preferences Form 725
The Print Form 726
The Printer Properties Form 727

© 2023 Cadence Design Systems, Inc.

The Page Tab . . . 727
The Job Options Tab . . . 728
The Reference Point Marker Properties Form . . . 730
The Reveal Columns Form . . . 733
The Sample Values Form . . . 734
The Save Image Form . . . 735
The Save As CSV Form . . . 739
The Save Image Form . . . 740
The Select Color Form . . . 744
The Select Waveform Database Form . . . 745
The Set Sweep Ranges Form . . . 746
The Smith Reference Point Marker Properties Form . . . 747
The Swap Sweep Var Form . . . 749
The Value At Form . . . 750
The Vertical Marker Properties Form . . . 751
The Waveforms to be Exported Form . . . 755
The Window Properties Form . . . 756
The Y vs Y Form . . . 757
The Zoom To Form . . . 758

A
Virtuoso Visualization and Analysis XL Tool Environment Variables . . . 759

Graph Variables . . . 761
Font String . . . 762
Graph Frame Variables . . . 765
linkDepAxes . . . 772
linkIndepAxes . . . 773
useLinkAxesUnitFilter . . . 774
linkAxesUnitFilterSet . . . 775
graphMinWidth . . . 776
graphMinHeight . . . 777
Graph Environment Variables . . . 778
Rectangular Graphs Environment Variables . . . 783
Strip Environment Variables . . . 784

© 2023 Cadence Design Systems, Inc.

Digital Strip Environment Variables . . . 785
Circular Graph Environment Variables . . . 786
legendPosition . . . 787
plotToSmithView . . . 788
showFreqFilterBar . . . 789
traceMarkerDisplay . . . 790
Axis Environment Variables . . . 792
Dependent Axis Environment Variables . . . 793
Independent Axis Environment Variables . . . 797
String Independent Axis Environment Variables . . . 802
Trace Environment Variables . . . 803
Trace Legend Environment Variables . . . 804
Digital Trace Environment Variables . . . 807
Digital Bus Trace Environment Variables . . . 808
font . . . 809
Histogram Environment Variables . . . 812
Horizontal Marker Environment Variables . . . 813
Reference Line Marker Environment Variables . . . 819
Vertical Marker Environment Variables . . . 823
Point Marker Environment Variables . . . 829
Reference Point Marker Environment Variables . . . 834
Specification Marker Environment Variables . . . 838
Intercept Marker Environment Variables . . . 845
Circle Marker Environment Variables . . . 847
Delta Marker Environment Variables . . . 850
Transient Edge Markers Environment Variables . . . 856
defaultLabel . . . 857
significantDigits . . . 858
font . . . 859
notation . . . 860
threshold . . . 861
foreground . . . 862
lineStyle . . . 863
showMeasSummary . . . 864
snapPoint . . . 865
Graph Label Environment Variables . . . 866

© 2023 Cadence Design Systems, Inc.

Probe Environment Variables . . . 867
Polar Grid Environment Variables . . . 868
Smith Grid Environment Variables . . . 869
font . . . 870
autoSizeLabels . . . 873
showDetailed . . . 874
showLabels . . . 875
showCompressed . . . 876
showCoefficientRing . . . 877
showVSWRCircles . . . 878
VSWRCircleValues . . . 879
VSWRCirclePenColor . . . 880
VSWRCirclePenWidth . . . 881
VSWRCirclePenStyle . . . 882
showQContours . . . 883
QContourValues . . . 884
QContourPenColor . . . 885
QContourPenWidth . . . 886
QContourPenStyle . . . 887
impedanceStandardGridPenColor . . . 888
impedanceStandardGridPenWidth . . . 889
impedanceStandardGridPenStyle . . . 890
impedanceDetailedGridPenColor . . . 891
impedanceDetailedGridPenWidth . . . 892
impedanceDetailedGridPenStyle . . . 893
impedanceCompressedGridPenColor . . . 894
impedanceCompressedGridPenWidth . . . 895
impedanceCompressedGridPenStyle . . . 896
admittanceStandardGridPenColor . . . 897
admittanceStandardGridPenWidth . . . 898
admittanceStandardGridPenStyle . . . 899
admittanceDetailedGridPenColor . . . 900
admittanceDetailedGridPenWidth . . . 901
admittanceDetailedGridPenStyle . . . 902
admittanceCompressedGridPenColor . . . 903
admittanceCompressedGridPenWidth . . . 904

© 2023 Cadence Design Systems, Inc.

admittanceCompressedGridPenStyle . . . 905
immittanceZStandardGridPenColor . . . 906
immittanceZStandardGridPenWidth . . . 907
immittanceZStandardGridPenStyle . . . 908
immittanceZDetailedGridPenColor . . . 909
immittanceZDetailedGridPenWidth . . . 910
immittanceZDetailedGridPenStyle . . . 911
immittanceZCompressedGridPenColor . . . 912
immittanceZCompressedGridPenWidth . . . 913
immittanceZCompressedGridPenStyle . . . 914
immittanceYStandardGridPenColor . . . 915
immittanceYStandardGridPenWidth . . . 916
immittanceYStandardGridPenStyle . . . 917
immittanceYDetailedGridPenColor . . . 918
immittanceYDetailedGridPenWidth . . . 919
immittanceYDetailedGridPenStyle . . . 920
immittanceYCompressedGridPenColor . . . 921
immittanceYCompressedGridPenWidth . . . 922
immittanceYCompressedGridPenStyle . . . 923
Application Environment Variables . . . 924
confirmCloseWindow . . . 927
confirmDeleteAll . . . 928
confirmDeleteAllMarkers . . . 929
confirmDeleteAllTraces . . . 930
XRange Bookmark Variables . . . 931
defaultLabel . . . 932
font . . . 933
significantDigits . . . 934
sigDigitsMode . . . 935
foreground . . . 936
background . . . 937
fillColor . . . 938
notation . . . 939
lineStyle . . . 940
showLabel . . . 941
zoomScaleFactor . . . 942

© 2023 Cadence Design Systems, Inc.

YRange Bookmark Variables . . . 943
defaultLabel . . . 944
font . . . 945
significantDigits . . . 946
sigDigitsMode . . . 947
foreground . . . 948
background . . . 949
fillColor . . . 950
notation . . . 951
lineStyle . . . 952
showLabel . . . 953
zoomScaleFactor . . . 954
Region Bookmark Variables . . . 955
defaultLabel . . . 956
font . . . 957
significantDigits . . . 958
sigDigitsMode . . . 959
foreground . . . 960
background . . . 961
fillColor . . . 962
notation . . . 963
lineStyle . . . 964
showLabel . . . 965
zoomScaleFactor . . . 966
Results Browser Variables . . . 967
Calculator Variables . . . 971
restoreCustomCalcFunctions . . . 995
Table Environment Variables . . . 998

B
Calculator Functions . . . 999
Basic Steps For Running Calculator Functions . . . 999
Performing Floating-Point Comparison . . . 1001
Using Unary Minus Operator . . . 1001
Using Division Operator in the Expression for a Calculator Function . . . 1002

© 2023 Cadence Design Systems, Inc.

Special Functions . . . 1003
aaSP . . . 1007
analog2Digital . . . 1009
Example 4 . . . 1013
abs_jitter . . . 1015
average . . . 1019
bandwidth . . . 1025
busTransition . . . 1029
calcVal . . . 1032
clip . . . 1035
compare . . . 1043
compression . . . 1046
compressionVRI . . . 1049
convolve . . . 1052
cross . . . 1055
d2a . . . 1059
dBm . . . 1062
delay . . . 1064
delayMeasure . . . 1070
deriv . . . 1074
dft . . . 1077
dftbb . . . 1083
dnl . . . 1088
dutyCycle . . . 1093
evmQAM . . . 1096
evmQpsk . . . 1099
eyeDiagram . . . 1103
eyeHeightAtXY . . . 1107
eyePeakToPeakJitter . . . 1116
eyeWidthAtXY . . . 1119
eyeAperture . . . 1127
fallTime . . . 1137
firstVal . . . 1146
flip . . . 1147
fourEval . . . 1149
Example 2 . . . 1151

© 2023 Cadence Design Systems, Inc.

freq . . . 1155
freq_jitter . . . 1160
frequency . . . 1163
gainBwProd . . . 1165
gainMargin . . . 1167
getAsciiWave . . . 1169
getData . . . 1177
groupDelay . . . 1180
harmonic . . . 1182
harmonicFreq . . . 1185
histogram2D . . . 1187
i . . . 1192
iinteg . . . 1195
inl . . . 1197
integ . . . 1201
intersect . . . 1204
ipn . . . 1206
ipnVRI . . . 1209
lastVal . . . 1212
loadpull . . . 1213
lshift . . . 1216
normalQQ . . . 1218
numConv . . . 1222
overshoot . . . 1224
pavg . . . 1228
peak . . . 1232
peakToPeak . . . 1235
period_jitter . . . 1237
phaseMargin . . . 1240
phaseNoise . . . 1242
PN . . . 1244
pow . . . 1249
prms . . . 1252
psd . . . 1255
psdbb . . . 1260
pstddev . . . 1265

© 2023 Cadence Design Systems, Inc.

pzbode . . . 1269
pzfilter . . . 1273
riseTime . . . 1278
rms . . . 1283
rms_jitter . . . 1285
rmsNoise . . . 1289
root . . . 1292
rshift . . . 1295
sample . . . 1298
settlingTime . . . 1301
kurtosis . . . 1310
normalQQPValue . . . 1312
skewness . . . 1313
slewrate . . . 1314
spectralPower . . . 1317
spectrumMeasurement . . . 1320
stddev . . . 1325
swapSweep . . . 1329
tangent . . . 1332
thd . . . 1335
triggeredDelay . . . 1339
unityGainFreq . . . 1343
v . . . 1345
value . . . 1350
waveVsWave . . . 1356
xmax . . . 1365
xmin . . . 1368
xval . . . 1371
ymax . . . 1374
ymin . . . 1376
Modifier Functions . . . 1378
conjugate . . . 1379
dB10 . . . 1382
dB20 . . . 1385
imag . . . 1388
real . . . 1391

© 2023 Cadence Design Systems, Inc.

mag . . . 1394
phase . . . 1397
phaseRad . . . 1401
phaseDeg . . . 1404
phaseDegUnwrapped . . . 1405
phaseRadUnwrapped . . . 1406
Trigonometric Functions . . . 1407
cos . . . 1408
acos . . . 1411
cosh . . . 1414
acosh . . . 1417
sin . . . 1420
asin . . . 1423
sinh . . . 1426
asinh . . . 1429
tan . . . 1432
atan . . . 1435
tanh . . . 1438
atanh . . . 1441
Math Functions . . . 1444
1/x . . . 1445
10**x . . . 1448
abs . . . 1451
exp . . . 1454
int . . . 1457
ln . . . 1460
log10 . . . 1463
sqrt . . . 1466
x**2 . . . 1469
y**x . . . 1472
RF Functions . . . 1476
How to Run Examples? . . . 1477
Rn . . . 1478
B1f . . . 1480
GA . . . 1482
gac_freq . . . 1484

© 2023 Cadence Design Systems, Inc.

gac_gain . . . 1486
Gmax . . . 1488
Gmin . . . 1490
Gmsg . . . 1492
GP . . . 1494
gpc_freq . . . 1496
gpc_gain . . . 1498
GT . . . 1500
Gumx . . . 1502
Kf . . . 1504
loadStability . . . 1506
nc_freq . . . 1509
nc_gain . . . 1511
NF . . . 1513
NFmin . . . 1515
rn . . . 1516
s11 . . . 1518
s12 . . . 1520
s21 . . . 1522
s22 . . . 1524
sourceStability . . . 1526
Spectre RF Functions . . . 1529
ifreq . . . 1530
ih . . . 1533
itime . . . 1536
pir . . . 1539
pmNoise . . . 1542
pn . . . 1544
pvi . . . 1546
pvr . . . 1551
rfGetMinDampFactor . . . 1554
spm . . . 1555
totalNoise . . . 1558
vfreq . . . 1561
vh . . . 1564
vtime . . . 1567

© 2023 Cadence Design Systems, Inc.

ypm . . . 1570
zpm . . . 1572

C
More About Eye Diagrams . . . 1575
NRZ Eye Measurements . . . 1576
Additional Measurements . . . 1597
PAM4 Eye Measurements . . . 1602
Eye Linearity . . . 1604
NRZ vs PAM3 vs PAM4 Eye Measurements . . . 1606
PAM3 Eye Diagram . . . 1608
Eye Measurements for Different Levels . . . 1610
Eye Parameters and Eye Measurements . . . 1616

D
Constants . . . 1619

E
Defining New SKILL Functions . . . 1621
Defining a Form . . . 1621
Defining a Callback Procedure . . . 1622
Using Stack Registers in the Procedure . . . 1622
Registering the Function . . . 1622
Defining a Custom Function . . . 1623

F
Working With Function Templates . . . 1625
Function Templates . . . 1626
Function Template Search Paths . . . 1626
Template Catalog Summary File . . . 1626
Creating a template file . . . 1628
Working with Template File . . . 1629
Header . . . 1629

© 2023 Cadence Design Systems, Inc.

The Analysis Section 1629
Examples 1631
Example 1: Sample template with single argument: average.ocn 1631
Example 2: Sample template with multiple arguments: delay.ocn 1632
Example 3: Signature described by a format statement: compression.ocn 1635
Example 4: Creating your own template 1636
Advanced Features to provide GUI Hints 1639

Index 1643

© 2023 Cadence Design Systems, Inc.

1

Overview of Virtuoso Visualization and Analysis XL

The Virtuoso® Visualization and Analysis XL is an analog and mixed-signal waveform display tool. This user guide describes this tool in detail and explains how you can use the various features of this tool.

The tool helps you analyze the data generated by your simulator.

The tool consists of the following components:

- Results Browser displays simulation data in the hierarchical arrangement of your design.
- Graph offers features that simplify the processing of your signal data.
- Calculator provides an extensive expression building capability that addresses the needs of a wide variety of analysis types.

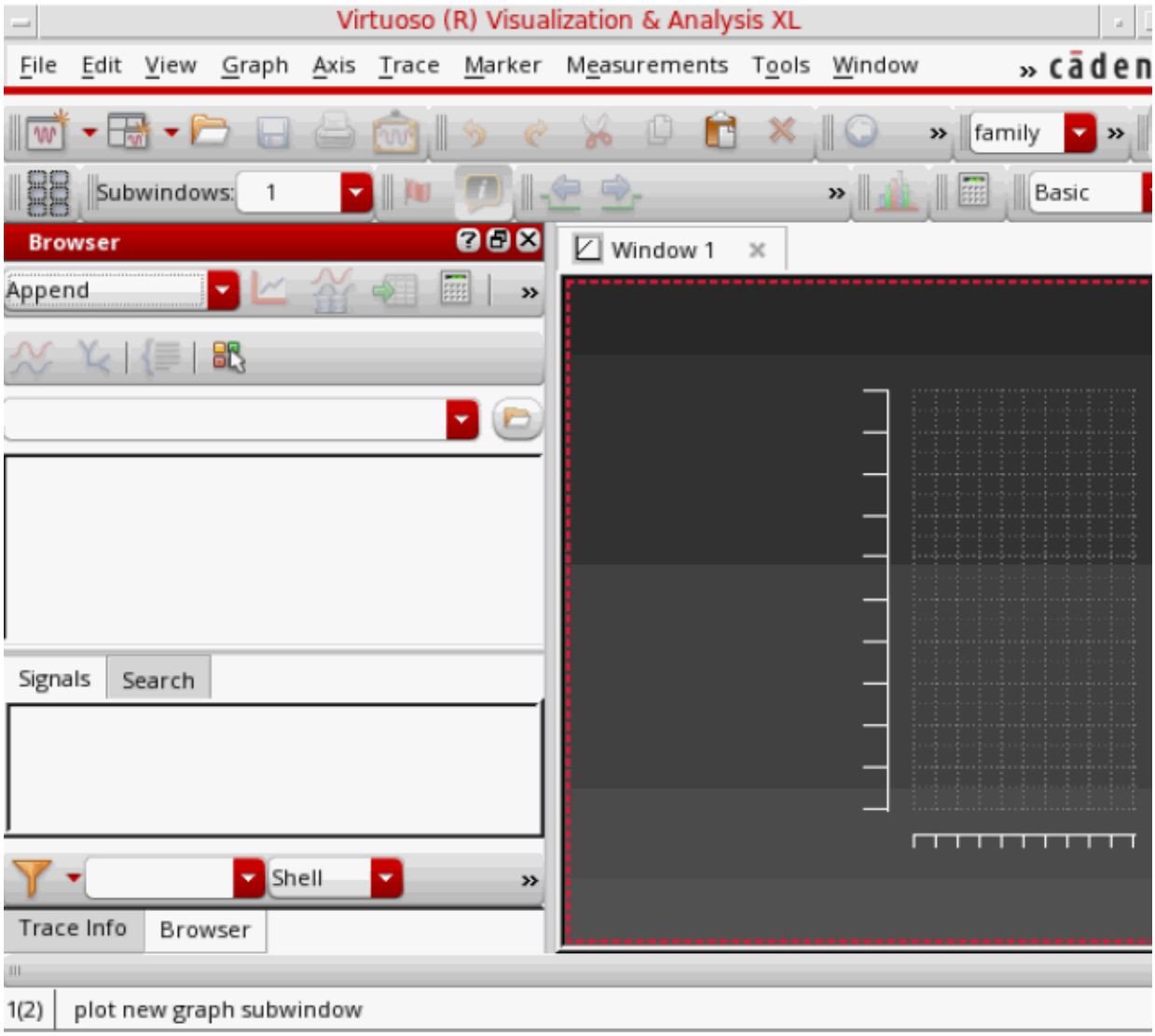
This user guide is aimed at developers and designers of integrated circuits and assumes that you are familiar with:

- The Virtuoso Studio design environment and application infrastructure mechanisms designed to support consistent operations between all Cadence® tools.
- The applications used to design and develop integrated circuits in the Virtuoso Studio design environment, notably, the Virtuoso Layout Suite, and Virtuoso Schematic Editor.
- The Virtuoso Studio design environment technology file.

© 2023 Cadence Design Systems, Inc.

- Component description format (CDF), which lets you create and describe your own components for use with Layout XL

You can use the Virtuoso Visualization and Analysis XL to browse, evaluate, analyze, and plot the simulation results.



Virtuoso Visualization and Analysis XL tool has the following three components:

- Results Browser
- Graph window

© 2023 Cadence Design Systems, Inc.

- Calculator

Results Browser is used to open the saved simulation results (signals). The signals are displayed in a hierarchical arrangement that corresponds to the hierarchy of your design, making it convenient for you to locate and manage simulation data. For more information about the Results Browser, see Results Browser

Graphs are displayed in the various graph windows and subwindows that can be opened in the graph display area in the Virtuoso Visualization and Analysis XL window. For detailed information about graphs, see Chapter 4, “Working with Graphs.”

Calculator opens in a separate window and is used to create, store, and evaluate expressions for the signals displayed in the Results Browser. For detailed information about Calculator, see Chapter 5, “Working with the Calculator.”

This chapter includes the following sections:

- Opening the Virtuoso Visualization and Analysis XL on page 28
- Closing the Virtuoso Visualization and Analysis XL on page 31
- Saving a Session on page 31

Tip

You can read the Knowledge Booster Training Bytes blog to learn more about the training videos and online courses available for Virtuoso Visualization and Analysis XL.

Licensing Requirements

Following is the licensing scheme for the Virtuoso Visualization and Analysis XL tool:

Virtuoso Visualization and Analysis XL when opened from ADE Explorer, Assembler, VVO

- When you close the ADE window, the Virtuoso Visualization and Analysis XL continues to hold the ADE license tokens, which are in effect until all Virtuoso Visualization and Analysis XL windows are closed.

© 2023 Cadence Design Systems, Inc.

Virtuoso Visualization and Analysis XL when opened in stand-alone mode or from Virtuoso:

Checks out either the Virtuoso Visualization and Analysis XL license, ADE license tier, or ADE Verifier license depending on the preferences you have set by using the `VIVALicenseCheckoutOrder` .cdsenv variable. By default, this variable is set to `ViVA`, `ADE`, `Verifier`.

For more information, see `VIVALicenseCheckoutOrder` in the `CheckoutOrder Variables` section of *Virtuoso Software Licensing and Configuration User Guide*.

Supported Data Formats

The Virtuoso Visualization and Analysis XL can interpret the following data formats.

Format	Description
PSF	Format created by Virtuoso® Spectre Circuit Simulator (including the Virtuoso® Spectre RF Simulation Option) and other simulators integrated into the Virtuoso® Analog Design Environment. The visualization and analysis tool can read files larger than 2G.
SST2	Format created by Virtuoso® AMS and Spectre Verilog Simulators. Spectre and UltraSim can also create this format. This format can also be created by the digital simulators, such as NCsim.

© 2023 Cadence Design Systems, Inc.

Format	Description
PSF XL	Format created by Virtuoso® Spectre Circuit Simulator (including the Virtuoso® Spectre RF Simulation Option) and other simulators integrated into the Virtuoso® Analog Design Environment. This format provides a high rate of lossless data compression. Using this format, Virtuoso Visualization and Analysis XL can render extremely large datasets at reasonable time rate. Alias support is added to PSF XL. This alias support is required for Spectre XPS and for unified waveform DB feature driven by AMS Designer. PSF XL also supports the reading of in-memory simulator data, which provides external viewers an access to the PSF XL waveform data that is still buffered in the simulator memory. This feature is useful if you use AMS, because in this case you often need to pause the simulation when certain conditions are met, and then inspect the analog and digital waveform data up to the last simulation time point. It is currently supported by SST2, which is the default output format for Spectre, APS, and Ultrasim. **Note:** When you use the PSF XL format, the MMSIM and stand-alone Spectre license is checked out.
Touchstone	Format created by all simulator. This is an industry-standard format. The touchstone file is an ASCII file, also known as the *SnP* file, which includes a large signals S-parameter results. The touchstone files are of `.sxp` or `.snp` extensions, where `x` is the integer that varies depending on the number of network ports of the device. For example, if the touchstone file contains the network parameters for a two-port device, the value of *x* is `2` and the file has `.s2p` extension. For an N-port network, the file has `.snp` extension, where the number of network ports are not specified in the file extension. Virtuoso Visualization and Analysis XL also supports reading files with `.snp` extensions. The s-parameter reader supports the Touchstone 2.0 file format that can read multiple port impedance information. **Note:** The `ypm` and `zpm` Calculator functions support the touchstone format.

© 2023 Cadence Design Systems, Inc.

Points to Note:

- Virtuoso Visualization and Analysis XL does not support WSF format.
- Virtuoso Visualization and Analysis XL supports the reading of fast signal database (FSDB) file format. To know the latest FSDB version supported by Virtuoso, run the following command:

```
srrversion -fsdb
```

Opening the Virtuoso Visualization and Analysis XL

You can start the tool either from within ADE or in the standalone mode. When you open the tool from within ADE, you work on the simulation results for the latest run. However, in the standalone mode, you work on the saved simulation results that can be accessed through the Results Browser.

This section includes the following topics:

- Opening the Tool in Stand-Alone Mode on page 28
- Opening the Tool Using CIW on page 30
- Opening the Tool Using Virtuoso Analog Design Environment on page 31

Opening the Tool in Stand-Alone Mode

To open the Virtuoso Visualization and Analysis XL window from CIW, do the following:

- Type the following command in a terminal window to open the tool in SKILL mode:

```
viva
     [-expr skill
     |-V
     |-W
     |-datadir
     |-mode XL]
     | -log logFileName
     | -nocdsinit
     | -noblink
     | -45
     | -noxshm
     | -nograph
     | -nographE
```


© 2023 Cadence Design Systems, Inc.

```
| -replay inputLog
| -restore fileName
| -load_graph fileName ]
```

Parameter	Description
`-h` or `-help`	Displays information about how to run the tool.
`-expr skill`	Starts the tool in SKILL mode. By default, Virtuoso Visualization and Analysis XL opens in SKILL mode.
`-v`	Displays the version number for the tool.
`-w`	Displays the subversion number for the tool.
`-datadir`	Specifies the data directory to be opened on startup.
`-mode XL`	Specifies the product mode to be used. This determines the license to be checked out. The default mode is XL.
`-log` *`logFileName`*	Logs session information to the file *`logFileName`*.
`-nocdsinit`	Skips reading the `.cdsinit` file.
`-noblink`	Turns off blinking.
`-45`	Enhances drawing of 45-degree diagonal lines.
`-noxshm`	Prevents the use of X Shared Memory Access.
`-nograph`	Starts the tool in non-graphical mode.
`-nographE`	Emulates non-graphical mode. Uses the default display or the display specified with the `-display` command-line option. Note that windows and forms are drawn on the specified display. Nograph emulation can also be invoked by defining the environment variable `CDS_NOGRAPH_DISPLAY` to specify the display along with the `-nograph` command-line option or this option.
`-replay` *`inputLog`*	Specifies the name of the log file to be replayed.

© 2023 Cadence Design Systems, Inc.

Parameter	Description
`-restore` *`fileName`*	Specifies the name of the session file to be restored.
`-load_graph` *`fileName`*	Specifies the name of the graph state file (.grf) to be loaded.

Note: The command line options for the Virtuoso Visualization and Analysis XL tool are not case-sensitive.

The Virtuoso Visualization and Analysis XL window appears.

When the Virtuoso Visualization and Analysis XL window is opened in the stand-alone mode, the CIW window includes the following options on the menubar:

- *File*
  - *Exit*
- *Tools*
  - *Calculator*
  - *Results Browser*
  - *Waveform*
  - *SKILL IDE*
- *Options*
  - *Bindkey*
  - *User Preferences*
  - *Log Filter*
- *Help*

Opening the Tool Using CIW

To open the tool using CIW, first you need to open Virtuoso and then do the following to open different tools in Virtuoso Visualization and Analysis XL:

- To open the Virtuoso Visualization and Analysis XL tool:
  - Choose *Tools – ViVA XL – Waveform.*

© 2023 Cadence Design Systems, Inc.

- To open the tool with the Results Browser displayed:
  - Choose *Tools – ViVA XL – Results Browser.*
- To open the Virtuoso Visualization and Analysis XL Calculator:
  - Choose *Tools – ViVA XL – Calculator*.

Opening the Tool Using Virtuoso Analog Design Environment

You can open the tool from within ADE Assembler and ADE Explorer.

To know about how to open the graph window from within ADE, see Opening the Graph Window from ADE.

To know about how to open Results Browser from within ADE, see Opening a Results Directory.

To know about how to open Calculator from within ADE, see Opening the Calculator using ADE.

Closing the Virtuoso Visualization and Analysis XL

- To close the graph window:
  - In the graph window, choose *File – Close All Windows.*
  - Choose *File – Close Window*.

    Closes the active window. If there is only one open window, the *Close* command exits the tool.
- To close the Calculator window:
  - In the Calculator window, choose *File – Close*.

Saving a Session

When you save a session, the current state of the application is saved. When you reload a saved session, the same windows and settings as in the saved session come up. For more information about the graph attributes that are saved, see Saving and Loading Graphs on page 430.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

2

Results Browser

Results browser is an assistant in Virtuoso Visualization and Analysis XL that you can use to access the simulation results that you saved to work on later in a different session. Results Browser displays simulation results (signals) in a hierarchical arrangement that corresponds to the hierarchy of your design, making it convenient for you to locate and manage simulation data.

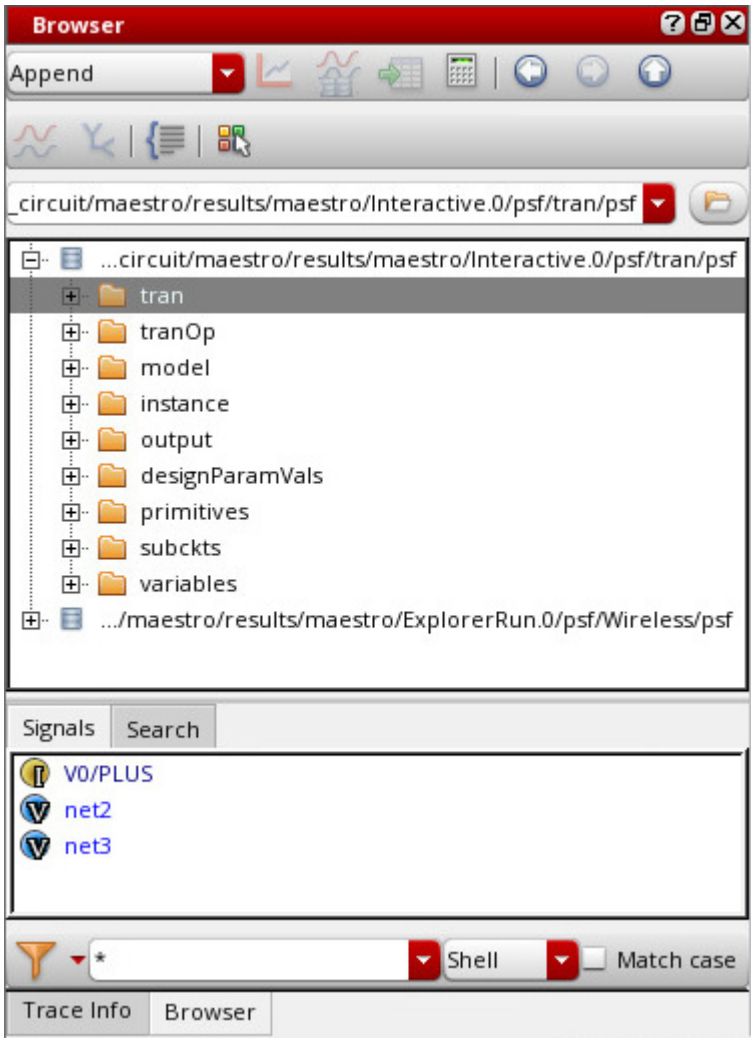
The results for each simulation are stored in a separate results directory. In addition, the results for different analysis types are stored in separate folders. For example, all simulation results for all transient analyses are contained in the `tran` directory. The various signals in a results directory are displayed in the Results Browser in a hierarchy that is determined by the design.

You can load multiple results directories in the Results Browser. The first directory that you load in the Results Browser becomes the in-context results directory. This means that the expressions in the Calculator are evaluated and the signals displayed in the graph are plotted by using the data available in the in-context results directory.

To change the in-context results directory, see Changing In-Context Results Directory.

© 2023 Cadence Design Systems, Inc.

The following figure shows the Results Browser assistant.



Related Topics

Opening the Results Browser

Results Browser Graphical User Interface

Opening a Results Directory

Selecting a Signal in the Results Browser

Plotting a Signal from the Results Browser

© 2023 Cadence Design Systems, Inc.

Opening the Results Browser

You can open the Results Browser in any of the following ways:

- Opening the Results Browser in Standalone Mode
- Opening the Results Browser from Virtuoso
- Opening the Results Browser from Virtuoso Analog Design Environment

Opening the Results Browser in Standalone Mode

To open the Results Browser in standalone mode:

1. Run the following command in a terminal window:

   ```
   viva -expr SKILL
   ```

   The Virtuoso (R) Visualization & Analysis XL window appears with the Results Browser displayed on the left.

2. To open a result directory in the Results Browser, specify the `viva -dataDir` command-line option as follows:

   ```
   viva -dataDir "./simulation/WirelessLib/wireless_LTE/maestro/results/maestro/
   ExplorerRun.0/psf/Wireless/psf"
   ```

   The specified results directories are opened in the Results Browser with the last-specified directory as in-context results directory.

   By default, Virtuoso Visualization and Analysis XL opens in SKILL mode.

Opening the Results Browser from Virtuoso

To open the Results Browser from Virtuoso:

➡ From the CIW, choose *Tools – ViVA XL – Results Browser*.

The Virtuoso (R) Visualization & Analysis XL window appears with the Results Browser displayed on the left.

Opening the Results Browser from Virtuoso Analog Design Environment

To open the Results Browser from Virtuoso ADE Assembler or Virtuoso ADE Explorer:

➡ From the menu bar, choose *Tools – Results Browser*.

© 2023 Cadence Design Systems, Inc.

The Virtuoso (R) Visualization & Analysis XL window appears with the Results Browser displayed on the left.

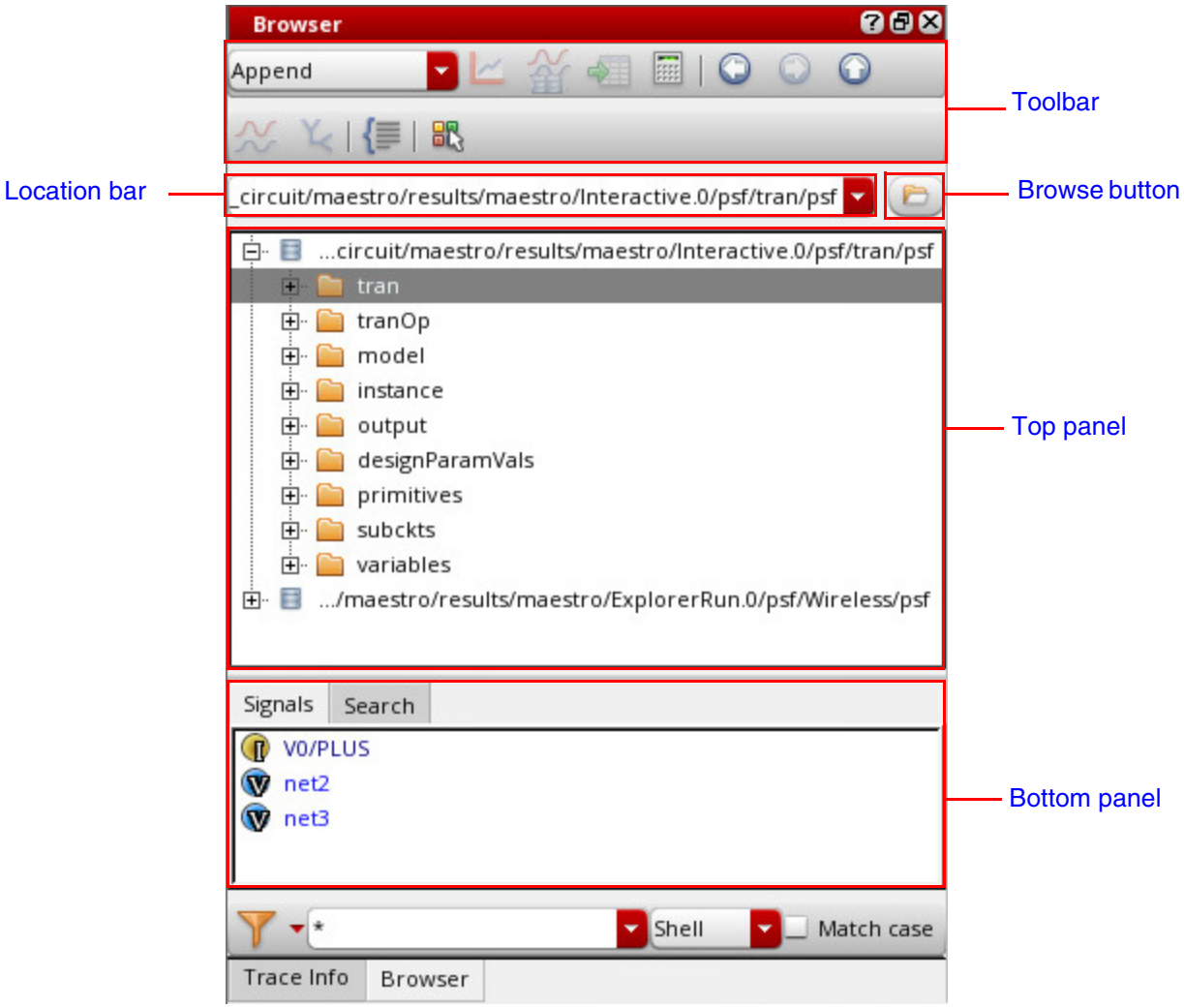
Related Topics

Results Browser Graphical User Interface

© 2023 Cadence Design Systems, Inc.

Results Browser Graphical User Interface

The following figure shows the graphical user interface (GUI) of the Results Browser:



The Results Browser GUI consists of the following elements:

- Toolbar of the Results Browser
- Top Panel of the Results Browser
- Bottom Panel of the Results Browser

© 2023 Cadence Design Systems, Inc.

Toolbar of the Results Browser

The following table describes the toolbar icons of the Results Browser.

Icon	Name	Description
	Plotting Mode	Specifies mode of plotting signals: ■ *Append*: Adds the signal to the selected graph. ■ *Replace*: Replaces the signal in the selected graph with the new signal. ■ *New Subwindow*: Plots the signal in a new subwindow within the current window. ■ *New Window*: Plots the signal in a new window.
	Plot Signals	Plots the selected signals in the current subwindow.
	Plot Signal from all Open DBs	Plots the selected signal from all open results directories in append mode. This helps compare signals that are common to all open results directories.
	Send to Table	Displays the signal data in the Virtuoso Visualization and Analysis XL Table.
	Send to Calculator	Opens the Calculator and displays the expression for the selected signal in the Calculator Buffer.
	Previous	Moves the control to the previous results directory selected in the Results Browser. When you open a new session, this button is not available.
	Next	Moves to control to the next results directory opened in the Results Browser.
	Up to Parent Directory	Moves the control to one level up in the results directory hierarchy.

© 2023 Cadence Design Systems, Inc.

Icon	Name	Description
	Graph Type	Lets you select the graph type: ■ *Default*: Plots the signal to the default graph type. ■ *Rectangular*: Plots the signal to a rectangular graph. ■ *Tabular*: Plots the signal to a tabular graph. ■ *Polar*: Plots the signal to a polar graph. ■ *Impedance*: Plots the signal to an impedance graph. ■ *Admittance*: Plots the signal to an admittance graph. ■ *Immittance*: Plots the signal to an immittance graph. ■ *RealvsImag*: Plots real versus imaginary. This option is available only for AC data.
	Graph Modifier	Lets you select the graph modifier: ■ *Magnitude*: Plots magnitude versus frequency. ■ *Phase*: Plots phase versus frequency. ■ *WPhase*: Plots wrapped phase versus frequency. ■ *Real*: Plots real versus frequency. ■ *Imaginary*: Plots imaginary versus frequency. ■ *dB10*: Plots dB10 versus frequency. ■ *dB20*: Plots dB20 versus frequency. ■ *dBm*: Plots dBm versus frequency. This option is available only for AC data.

© 2023 Cadence Design Systems, Inc.

Icon	Name	Description
	Diff of 2 Signals	Plots the difference of two signals in the graph window.
	YvsY of 2 Signals	Plots the Y Vs Y of two signals.
	Select Sweep Data	Lets you select the range for the sweep data.
	Change Color of Signals	Lets you change the default color of different signal types.
	Location bar	Lists the paths for the last 20 results directories opened in the Results Browser. If the results directory path name is long, it is displayed in the right-justified format in the location bar. This helps differentiate between the name of the results files when multiple results directories are opened in the Results Browser. To open a results directory, select it in the drop-down list. To open a new results directory that was loaded earlier in the Results Browser, enter the path in the box, or click the Browse button on the right.
	Browse	Lets you select the path to the results directory.

Top Panel of the Results Browser

The top panel of the Results Browser displays all loaded results directories in a tree view. The icon displayed next to a results directory indicates that it is a PSF directory.

The following table describes the context-menu commands that appear when you right-click a results directory in the top panel.

Icon	Command Name	Description
	Close Results	Closes the selected results directory.
	Open Terminal	Opens a terminal for the selected results directory.

© 2023 Cadence Design Systems, Inc.

Icon	Command Name	Description
	Set Context	Sets the selected directory as the in-context results directory.

Bottom Panel of the Results Browser

The bottom panel displays the contents of the selected database in a list view.

The following table describes the context-menu commands that appear when you right-click a signal in the bottom panel.

Icon	Name	Description
	Up One Level	Moves the control to one level up in the results directory hierarchy.
	Plot Signal	Plots the selected signals in the current subwindow. By default, signals are plotted in append mode.
	Plot from all DBs	Plots the selected signal from all open results directories in append mode. This helps compare signals that are common to all open results directories.
	Export	Exports the waveforms of the selected signals.
	Send to Export List	Sends the waveforms of the selected signals to an export list.
	Send to Calculator	Opens the Calculator and displays the expression for the selected signal in the Calculator Buffer.
	Send to Table	Displays the signal data in the Virtuoso Visualization and Analysis XL Table.
	Diff of 2 Signals	Plots the difference of two signals in the graph window.
	YvsY of 2 Signals	Plots the Y Vs Y of two signals.
–	*Append*	Adds the signal to the selected graph.

© 2023 Cadence Design Systems, Inc.

Icon	Name	Description
–	*Replace*	Replaces the signal in the selected graph with the new signal.
–	*New Subwindow*	Plots the signal in a new subwindow within the current window.
–	*New Window*	Plots the signal in a new window.

Related Topics

Opening the Results Browser

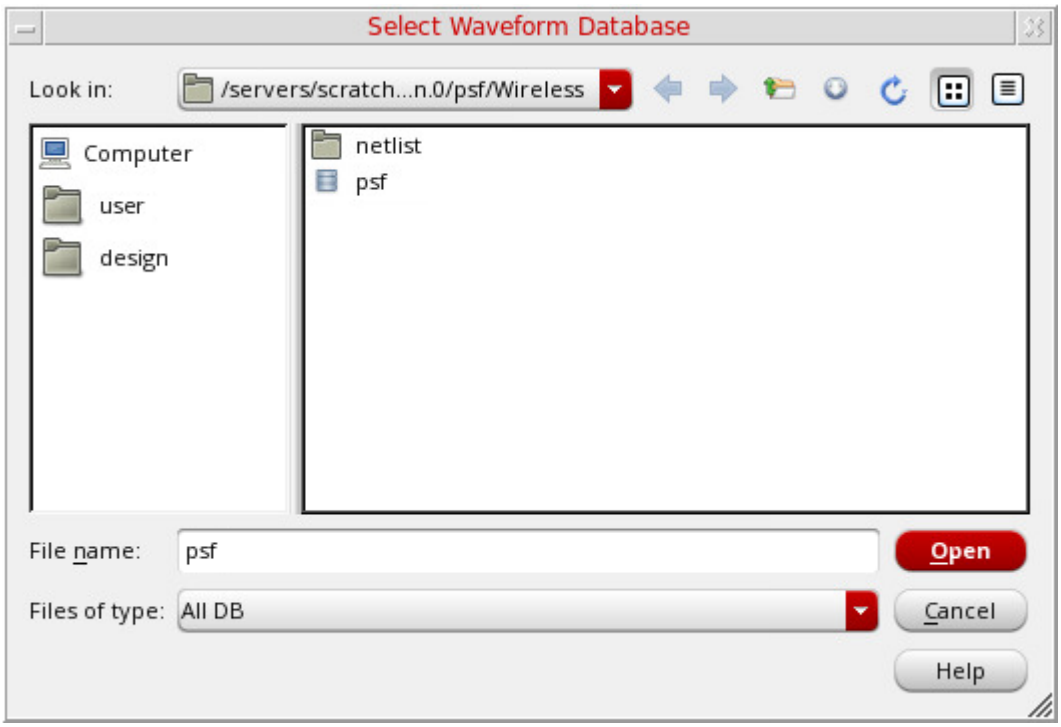
© 2023 Cadence Design Systems, Inc.

Opening a Results Directory

To open a results directory in the Results Browser:

1. From the menu bar of the Virtuoso Visualization and Analysis XL, choose either *File – Open Results* or *Browser – Results – Open Results*.

   The Select Waveform Database form opens.

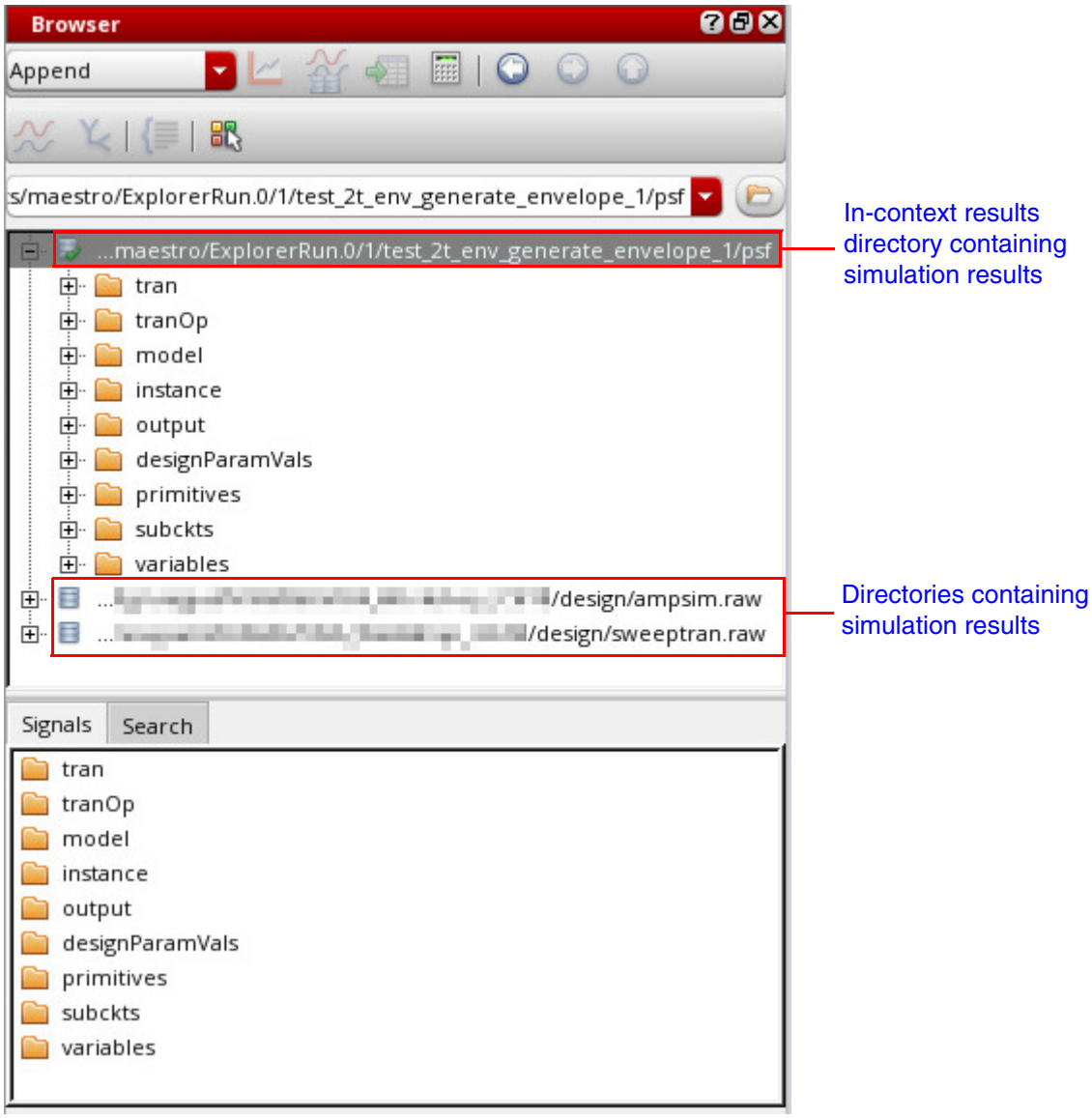


2. In the *Look in* field, browse to locate the results directory that you want to open. If required, you can open multiple results directories. By default, this field displays the path of the directory from where the you opened the results recently in the Results Browser.
3. Select the results directory that you want to open. You can select multiple directories at a time by using the `Ctrl` key.
4. Click *Open*.

   The selected results directory is opened in the Results Browser. If this is the first directory that is opened in Results Browser, it is displayed in green and the icon is displayed next to the directory name to indicate that it is the in-context results directory.

© 2023 Cadence Design Systems, Inc.

The icon is displayed next to the directories that contain simulation data.

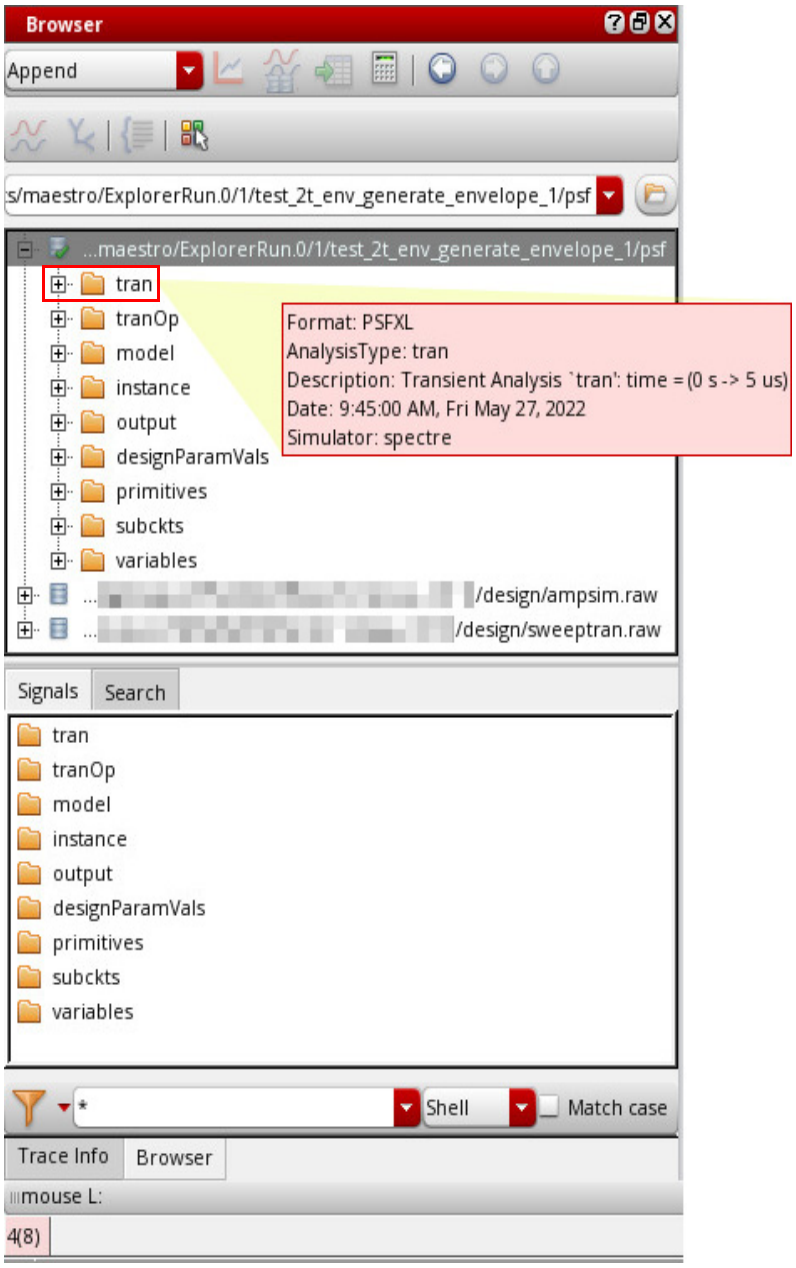


When you point to a folder of a database in the Results Browser, the following information is displayed as a tooltip.

- **Format**: Format in which simulation results are saved in the results directory.
- **Analysis Type**: Type of analysis performed.
- **Description**: Description of the analysis performed.
- **Date**: Date and time of the simulation results.

© 2023 Cadence Design Systems, Inc.

- **Simulator**: Name of the simulator.



Related Topics

The Select Waveform Database Form

Selecting a Signal in the Results Browser

© 2023 Cadence Design Systems, Inc.

Setting the In-Context Results Directory

Plotting a Signal from the Results Browser

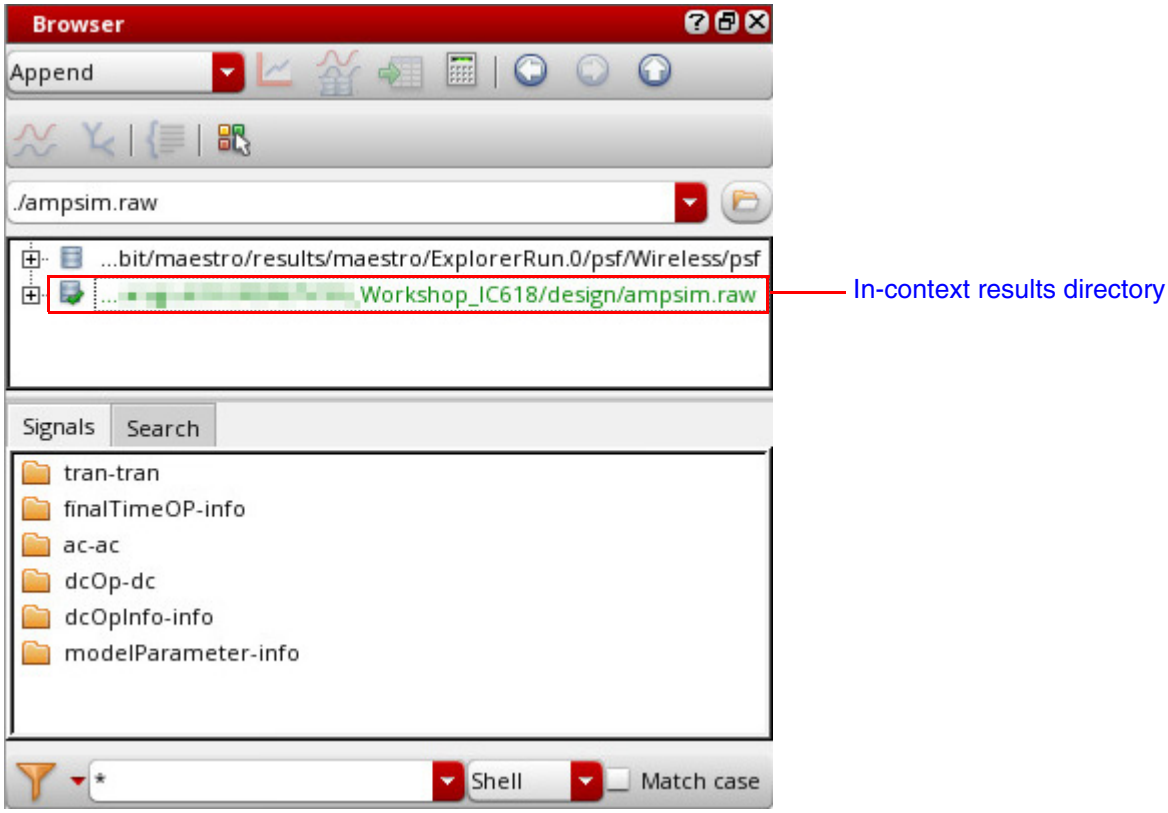
© 2023 Cadence Design Systems, Inc.

Setting the In-Context Results Directory

To change the database context directory from the Results Browser:

1. Open the results directory that you want to set as the context directory in the Results Browser.
2. Right-click the results directory and choose *Set Context*.

   The database context is set to the selected results directory.



   The results directory that is selected as the context directory is displayed in green and the icon is displayed next to it.

Setting the in-context results directory plays an important role in performing the following tasks:

Important

When you set the context to a new results directory after running a simulation and plotting results, using *Reload* will plot traces from all the corners of the new context results directory in the graph window. For example, you are currently in the context

© 2023 Cadence Design Systems, Inc.

`Interactive.3` and you plot the signal `/OUT` with three traces `C0_0`, `C0_1` and `C0_2` from corner `C0`. Now, if you change the context to `Interactive.4`, which has the five traces `C0_0`, `C0_1`, `C1_0`, `C1_1`, `C1_2` from corner `C0` and `C1`. Using *Reload* plots all these five traces from corner and `C1`.

Related Topics

Opening the Results Browser

Opening a Results Directory

Selecting a Signal in the Results Browser

Plotting a Signal from the Results Browser

© 2023 Cadence Design Systems, Inc.

Selecting a Signal in the Results Browser

In the Results Browser, the top panel displays the results directories and the bottom panel displays the datasets in the selected results directory.

To select a signal in the Results Browser:

1. Double-click a results directory in the top panel.

   The results directory is expanded and the folders containing data from different analyses are displayed in the bottom panel.

   The icon, displayed next to a directory in a list, indicates that the directory contains subdirectories. You can double-click the directory to display or to hide the subdirectories.

2. Double-click a dataset directory.

   The signals are displayed in the bottom panel. Each signal has an icon displayed next to it, which indicates the signal type.

   - indicates a voltage signal.
   - indicates a current signal.
   - indicates a power signal.
   - indicates a digital signal.
   - indicates a digital bus signal.
   - indicates an S-Parameter.
   - indicates an expression.
   - indicates a WREAL signal.

3. Right-click the signal and choose the command to either plot it or to send it to the Calculator or Table.

Related Topics

Plotting a Signal from the Results Browser

Sending a Signal to Calculator from Results Browser

Sending a Signal to Table from Results Browser

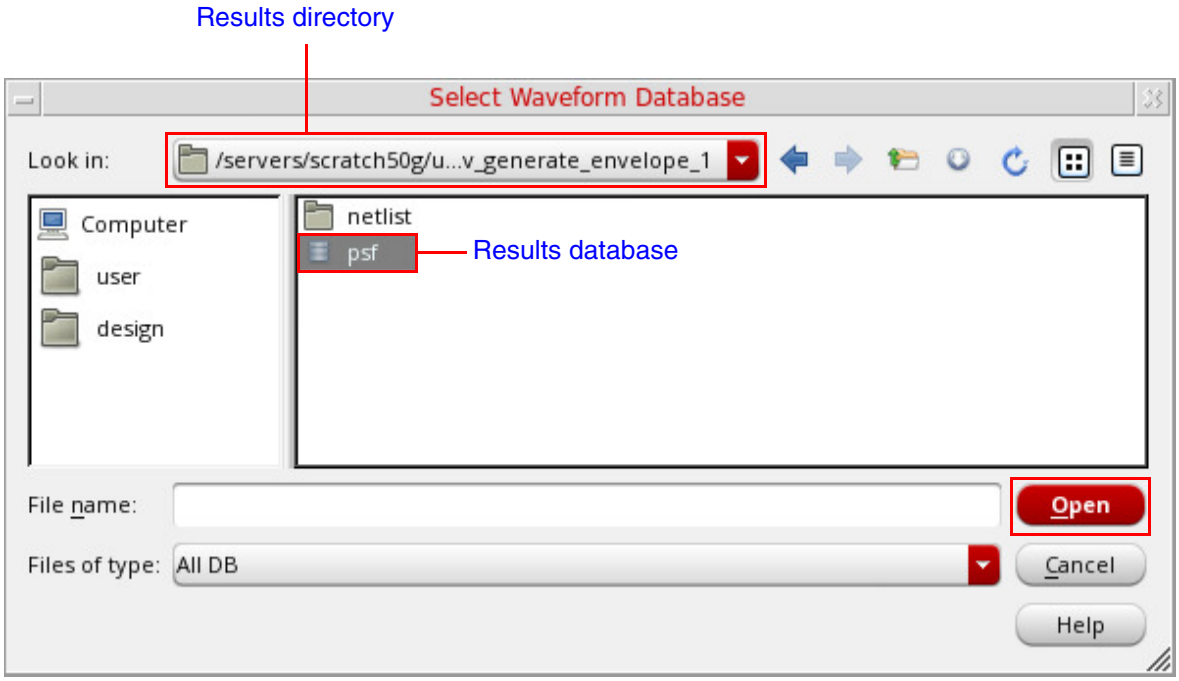
© 2023 Cadence Design Systems, Inc.

Plotting a Signal from the Results Browser

To plot a signal from the Results Browser:

1. In the Results Browser, click the Open Results icon.

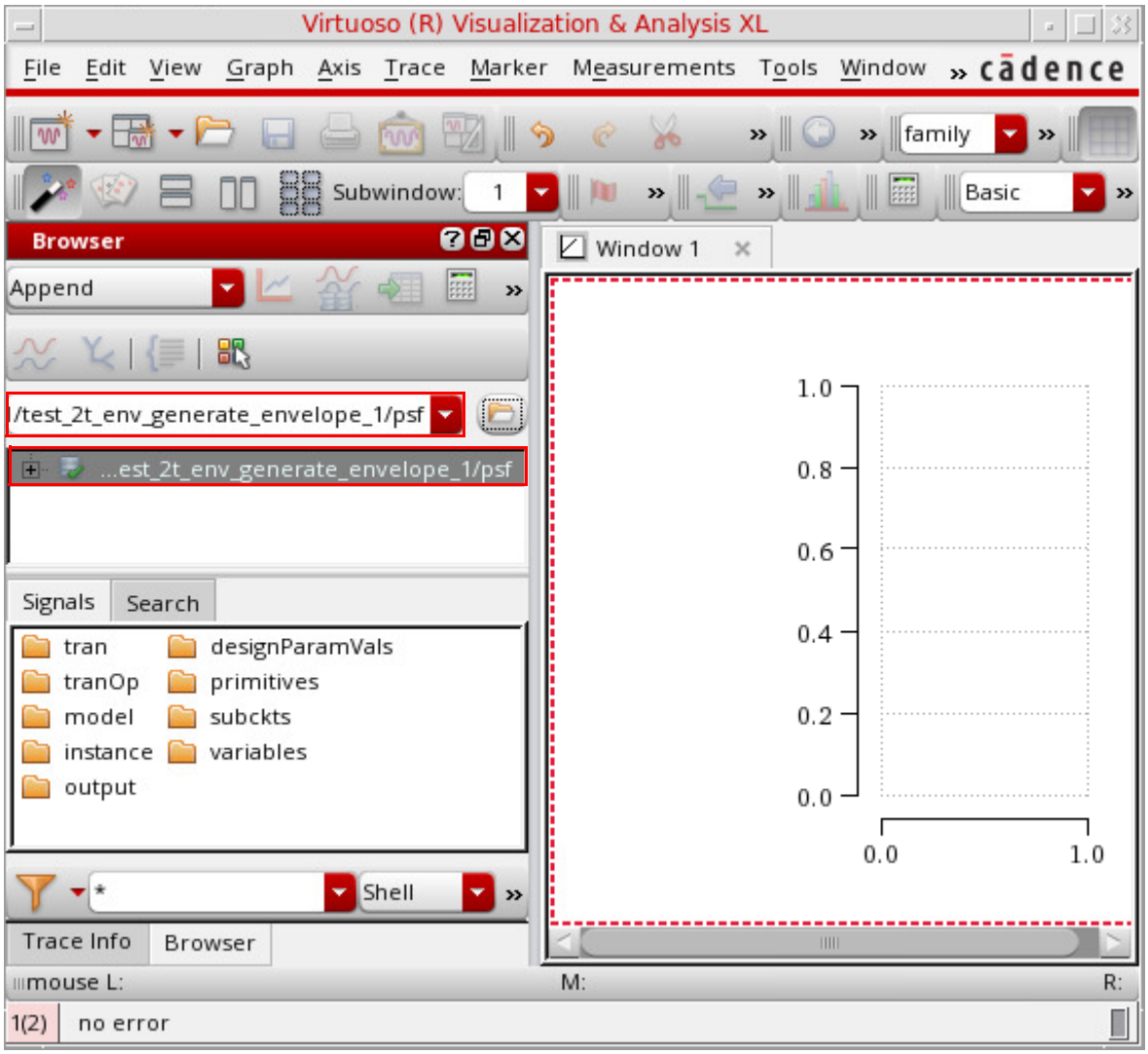
   The Select Waveform Database form opens.



2. Navigate to a results directory.
3. Select a results database.
4. Click *Open*.

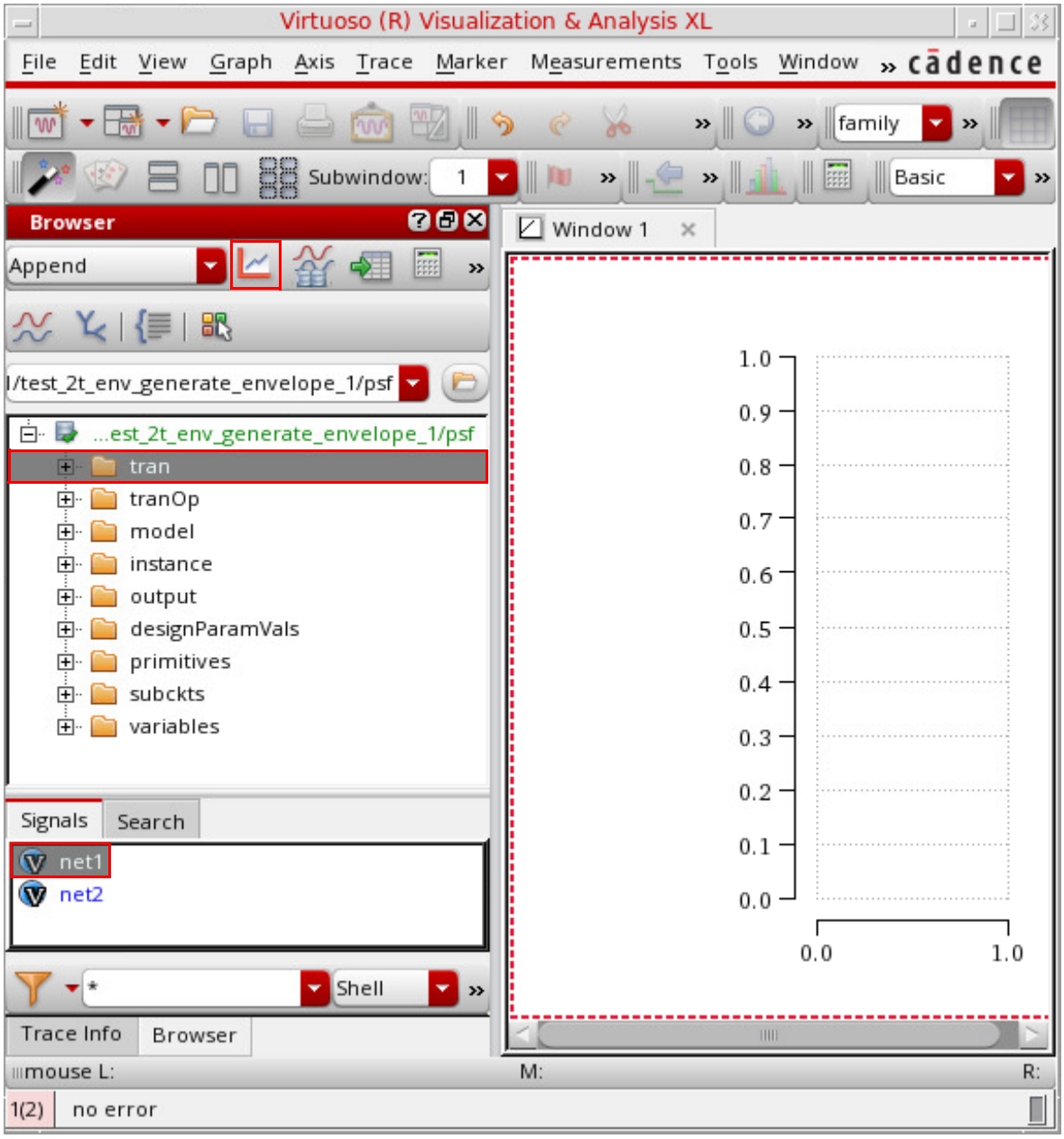
© 2023 Cadence Design Systems, Inc.

The selected results directory opens in the Results Browser.



© 2023 Cadence Design Systems, Inc.

5. In the results directory, navigate to the results folder that contains the signal you want to plot.

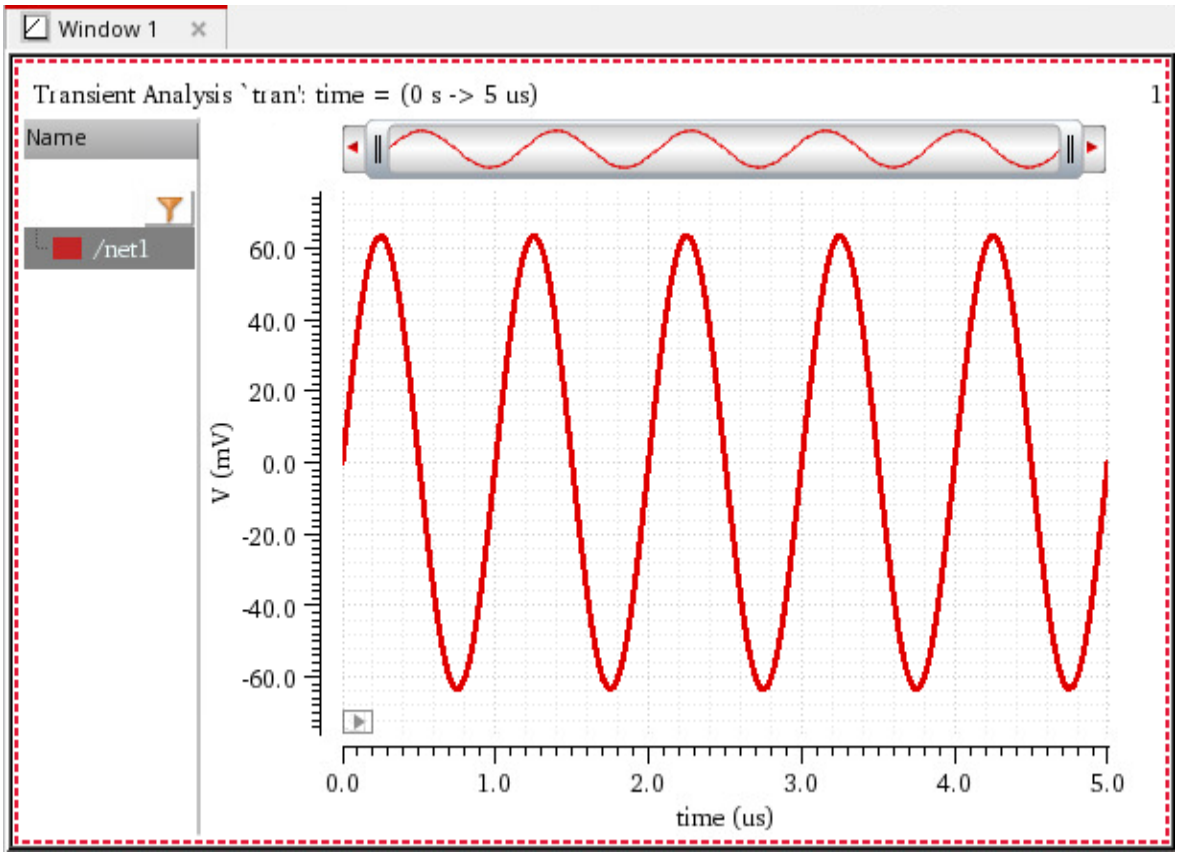


6. Select the signal you want to plot and click the *Plot Signals* icon in the toolbar of the Results Browser. Hold down the `Ctrl` key to select multiple signals.

   Alternatively, right-click the signal and choose *Plot Signal*.

© 2023 Cadence Design Systems, Inc.

The selected signal is plotted in the current graph window or subwindow.



Related Topics

The Select Waveform Database Form

Opening a Results Directory

Plotting a Transient Signal Over a Time Range

Plotting an AC Signal over a Frequency Range

Plotting Parametric Sweep Data

Plotting the Difference of Two Signals

Plotting YvsY for Two Signals

Comparing Signals from Different Results Directories

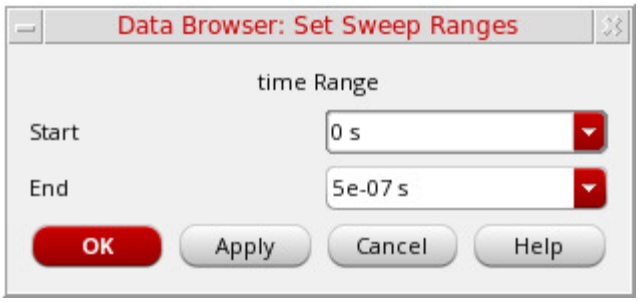
© 2023 Cadence Design Systems, Inc.

Plotting a Transient Signal Over a Time Range

You can plot transient data over a time range. To specify the time range for a transient dataset:

1. In the Results Browser, select a transient signal.
2. From the toolbar of the Results Browser, click *Select Sweep Data* .

   The Data Browser: Set Sweep Ranges form opens.



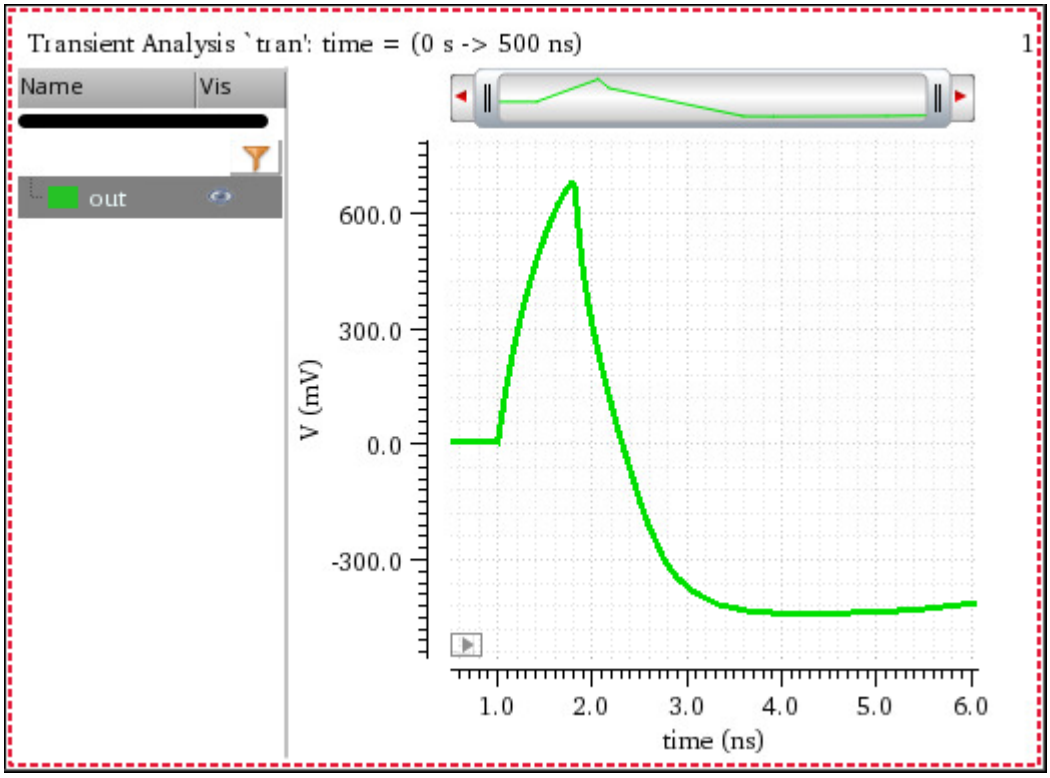
3. In the *Start* and *End* fields, specify the time range over which you want to plot the transient signal.

   If you want the graph to be plotted between the first and last data points of the signal, select *Default* from the *Start* and *End* drop-down lists, respectively.

   The default values of *Start* and *End* fields depend on the selected transient signal.
4. Click *OK*.
5. Click *Plot Signals* from the toolbar of the Results Browser to plot the transient data over the specified time range.

© 2023 Cadence Design Systems, Inc.

The following example shows the transient signal `out` plotted over the time range between `1ns` and `6ns`.



Related Topics

Plotting an AC Signal over a Frequency Range

Plotting Parametric Sweep Data

Plotting the Difference of Two Signals

Plotting YvsY for Two Signals

Comparing Signals from Different Results Directories

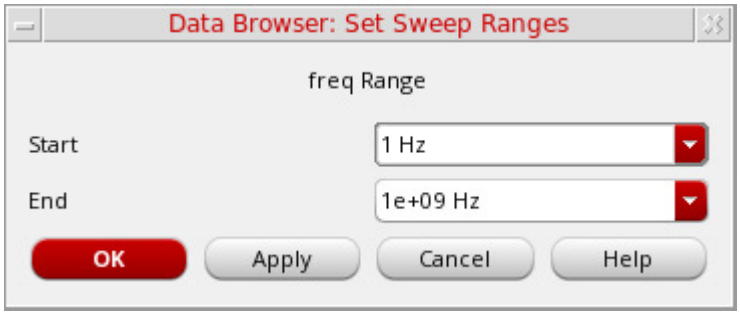
© 2023 Cadence Design Systems, Inc.

Plotting an AC Signal over a Frequency Range

You can plot an ac signal over a frequency range. To specify the frequency range for an ac signal:

1. In the Results Browser, select an ac signal.
2. From the toolbar of the Results Browser, click *Select Sweep Data* .

   The Data Browser: Set Sweep Ranges form opens.



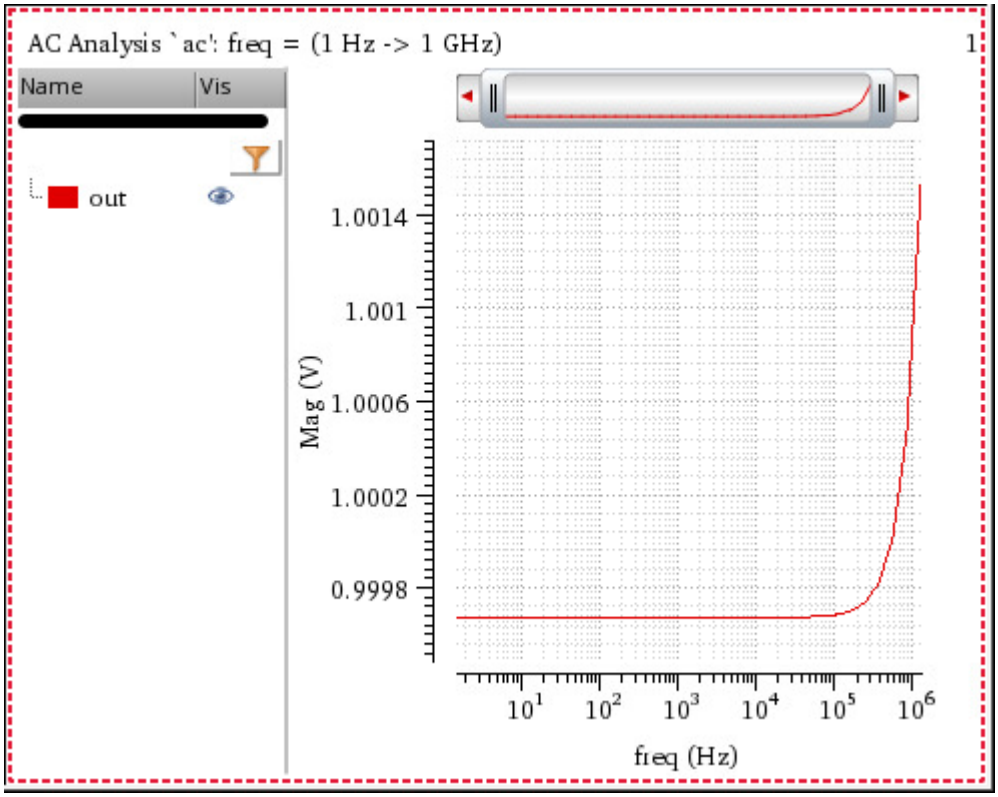
3. In the *Start* and *End* fields, specify the frequency range over which you want to plot the ac signal.

   If you want the graph to be plotted between the first and last data points of the signal, select *Default* from the *Start* and *End* drop-down lists, respectively.

   The default values of *Start* and *End* fields depend on the selected signal.
4. Click *OK*.
5. Click *Plot Signals* from the toolbar of the Results Browser to plot the ac signal over the specified frequency range.

© 2023 Cadence Design Systems, Inc.

The following example shows an ac signal `out` plotted over the frequency range between `2Hz` and `1e+06Hz`.



Related Topics

Plotting a Signal from the Results Browser

Plotting a Transient Signal Over a Time Range

Plotting Parametric Sweep Data

Plotting the Difference of Two Signals

Plotting YvsY for Two Signals

Comparing Signals from Different Results Directories

© 2023 Cadence Design Systems, Inc.

Plotting Parametric Sweep Data

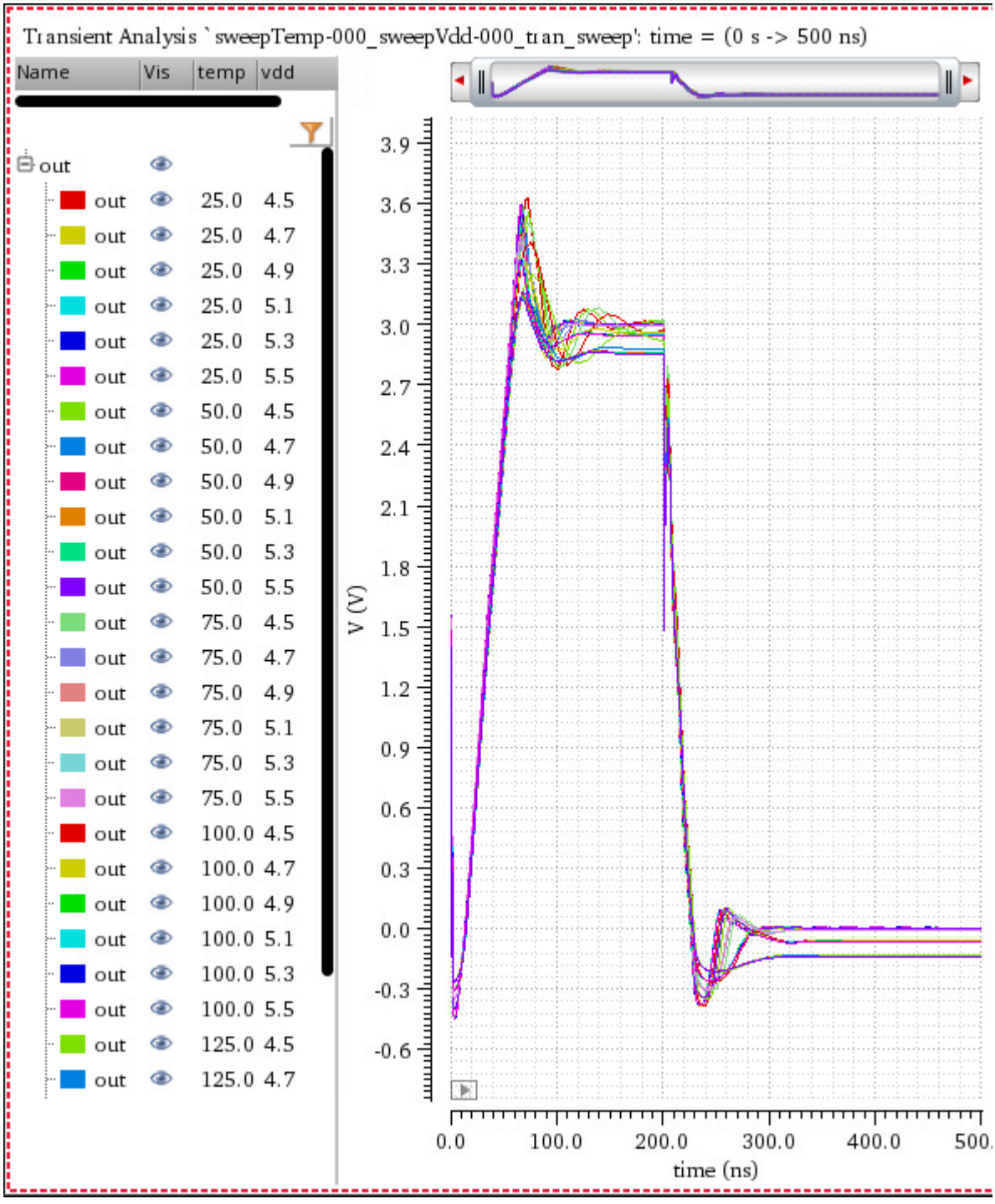
A parametric analysis sweeps a parameter or a group of parameters and runs one or more analyses for each combination of parameters. The Virtuoso Visualization and Analysis XL tool helps you analyze the resulting data efficiently.

To plot a part of parametric swept data:

1. In the Results Browser, select a parametric signal.

© 2023 Cadence Design Systems, Inc.

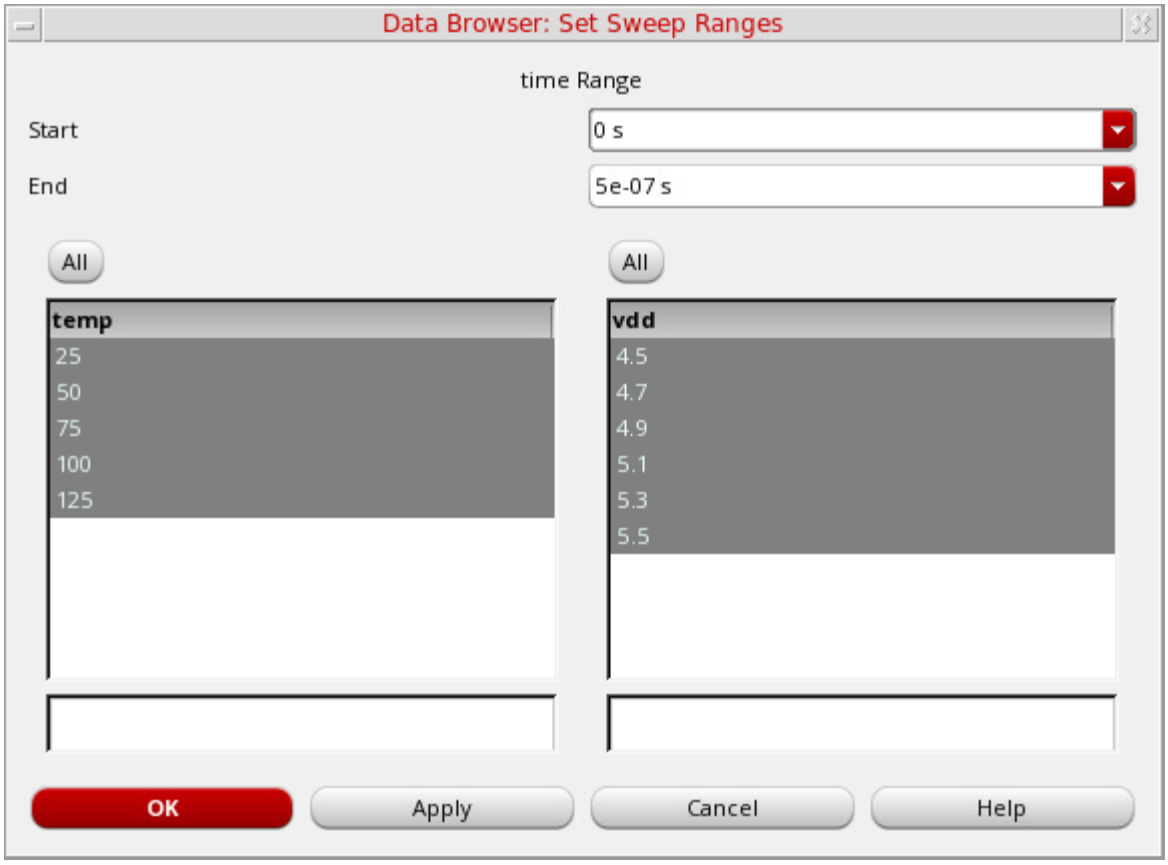
The following example shows a parametric signal `out`, which has two sweeping variables `temp` and `vdd`.



2. From the toolbar of the Results Browser, click *Select Sweep Data* .

© 2023 Cadence Design Systems, Inc.

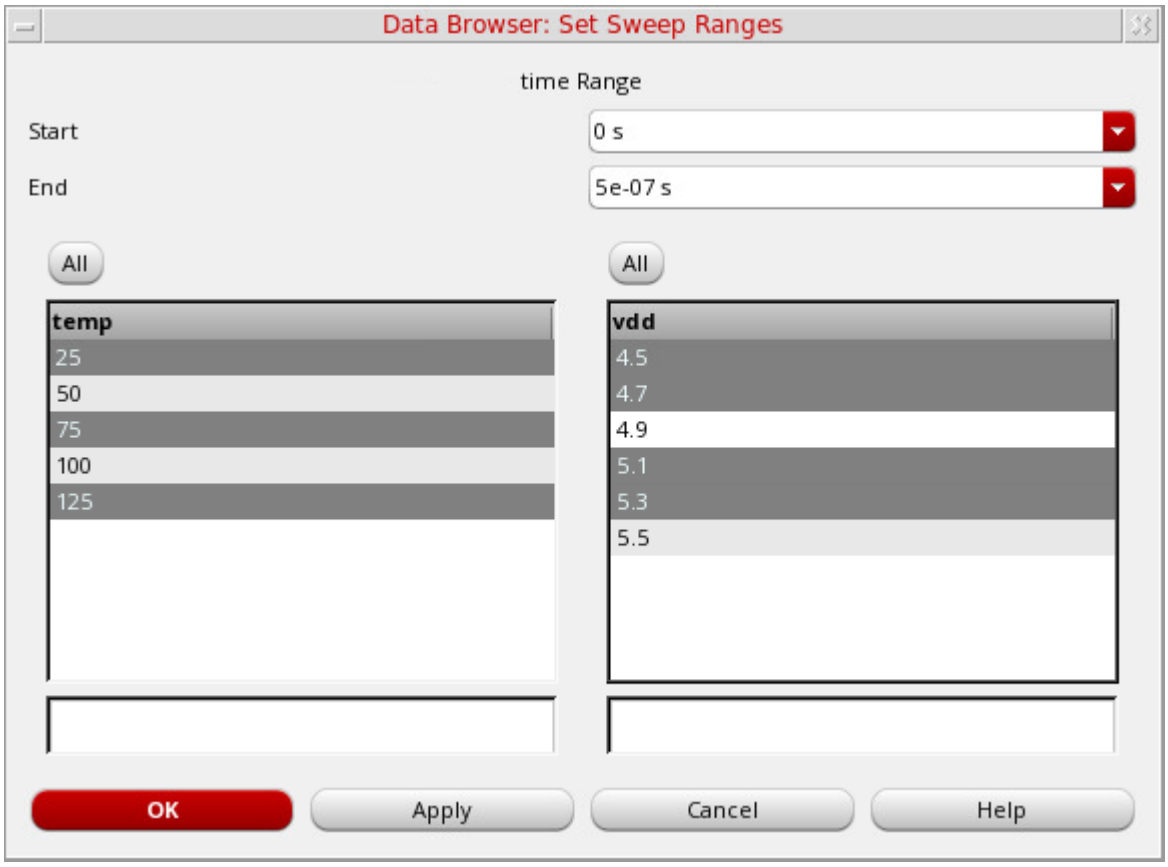
The Data Browser: Set Sweep Ranges form opens. The *Start* and *End* fields display the range of the signal. All the `temp` (temperature) and `vdd` (inner sweep variable) values are selected by default.



3. From the *temp* list, click `25.0`. Hold down the `Ctrl` key to select multiple values, and click `75.0`, and `125.0`.

© 2023 Cadence Design Systems, Inc.

4. From the `vdd` list, click `4.5`. Hold down the `Ctrl` key, and click `4.7`, `5.1`, and `5.3`.

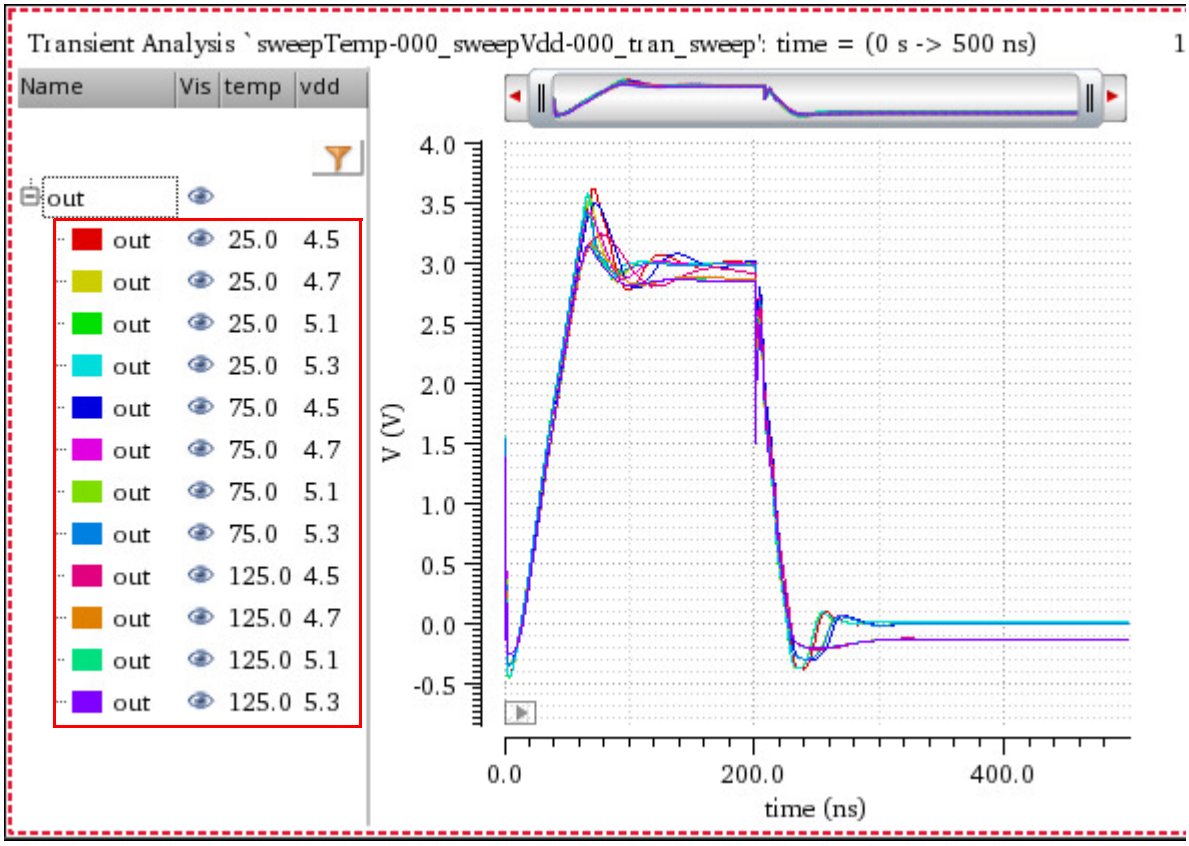


You can also use this form to filter and sort sweep variables by typing their values in the boxes at bottom.

5. Click *OK*.
6. In the Results Browser, right-click the `out` signal and choose *New Window*.

© 2023 Cadence Design Systems, Inc.

The graph window appears with the traces plotted for the parametric family. Each trace in the family is annotated by a sweep path that describes the parameter-value pair.



Related Topics

Plotting a Signal from the Results Browser

Plotting a Transient Signal Over a Time Range

Plotting an AC Signal over a Frequency Range

Plotting the Difference of Two Signals

Plotting YvsY for Two Signals

Comparing Signals from Different Results Directories

© 2023 Cadence Design Systems, Inc.

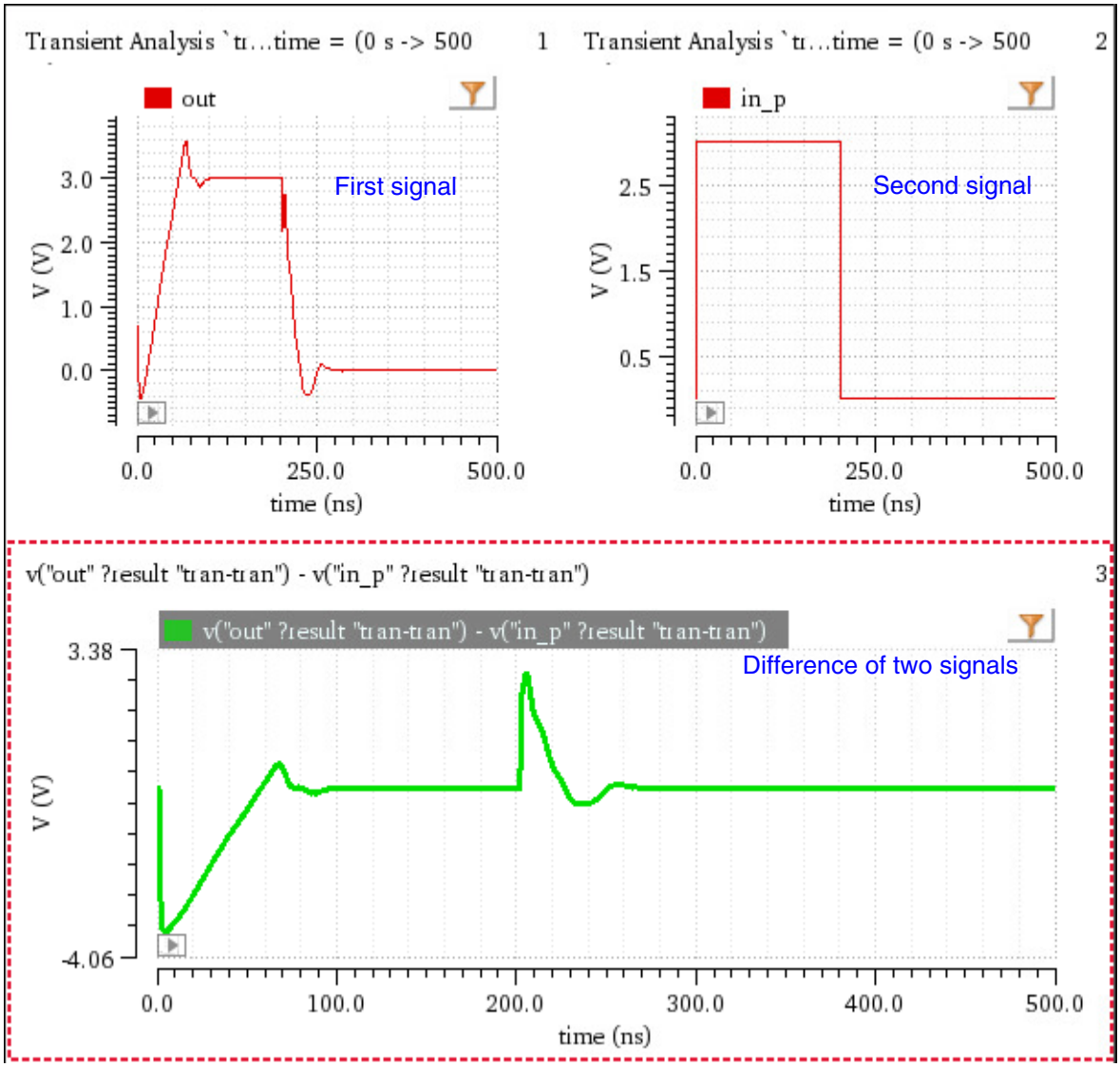
Plotting the Difference of Two Signals

To plot the difference of two signals:

1. In the Results Browser, select the first signal. For example, `out`.
2. In the Results Browser, select the second signal (for example, `in_p`), and then click *Diff of 2 Signals* .

   Alternatively, you can first click *Diff of 2 Signals* , and then select the second signal.

   The difference of the two signals is plotted in the current subwindow.



© 2023 Cadence Design Systems, Inc.

Related Topics

Plotting a Signal from the Results Browser

Plotting a Transient Signal Over a Time Range

Plotting an AC Signal over a Frequency Range

Plotting Parametric Sweep Data

Plotting YvsY for Two Signals

Comparing Signals from Different Results Directories

© 2023 Cadence Design Systems, Inc.

Plotting YvsY for Two Signals

You can plot y-axis values of one signal versus y-axis values of another signal. You can plot YvsY to measure input offset voltage, which displays the offset between the input and output of the circuit.

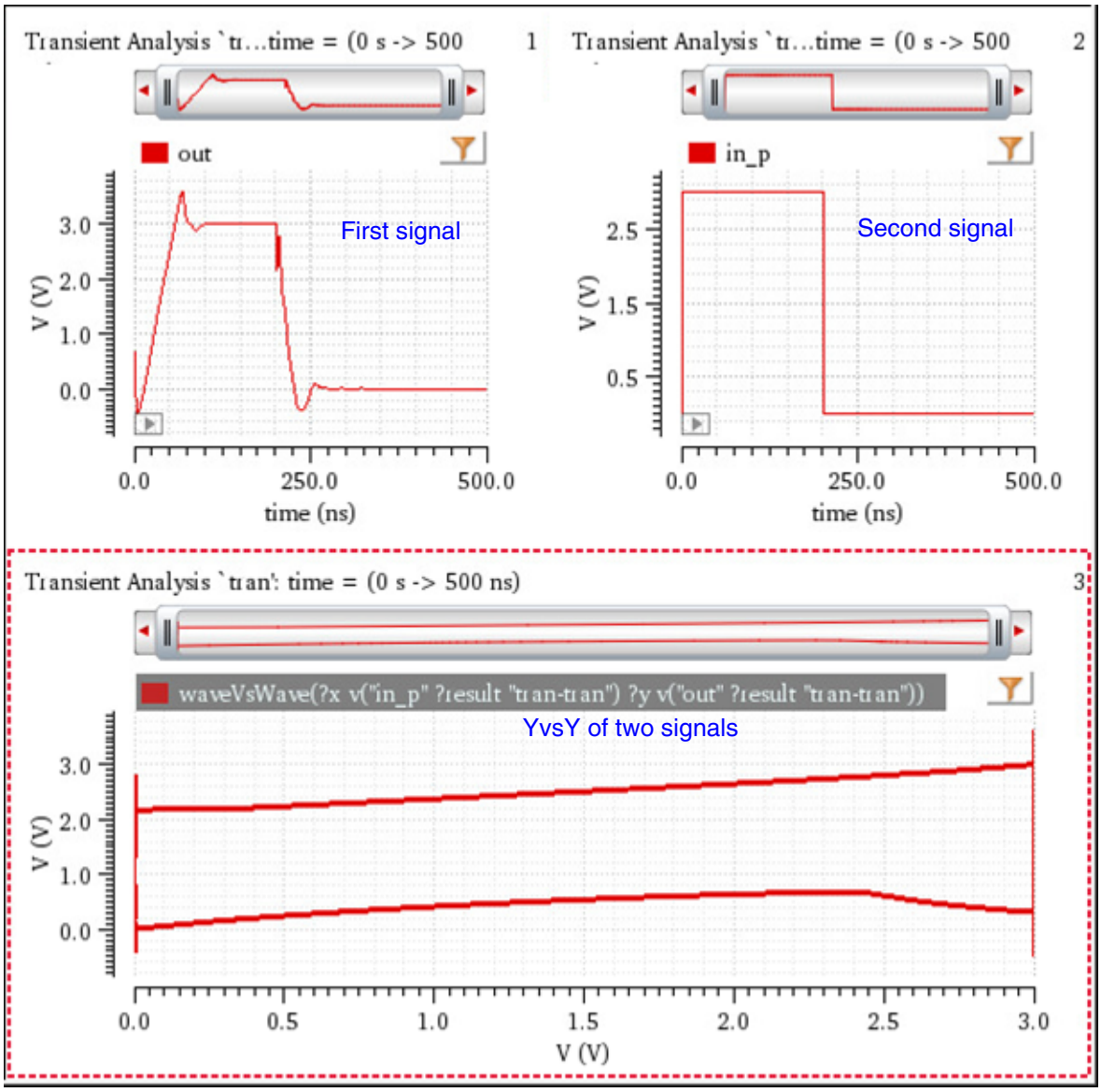
To plot YvsY for two signals:

1. In the Results Browser, select the first signal. For example, `out`.
2. In the Results Browser, select the second signal (for example, `in_p`), and then click *YvsY of 2 Signals* .

   Alternatively, you can first click *YvsY of 2 Signals* , and then select the second signal.

© 2023 Cadence Design Systems, Inc.

The YvsY of the two signals is plotted in the current subwindow.



Related Topics

Plotting a Signal from the Results Browser

Plotting a Transient Signal Over a Time Range

Plotting an AC Signal over a Frequency Range

Plotting Parametric Sweep Data

© 2023 Cadence Design Systems, Inc.

Plotting the Difference of Two Signals

Comparing Signals from Different Results Directories

© 2023 Cadence Design Systems, Inc.

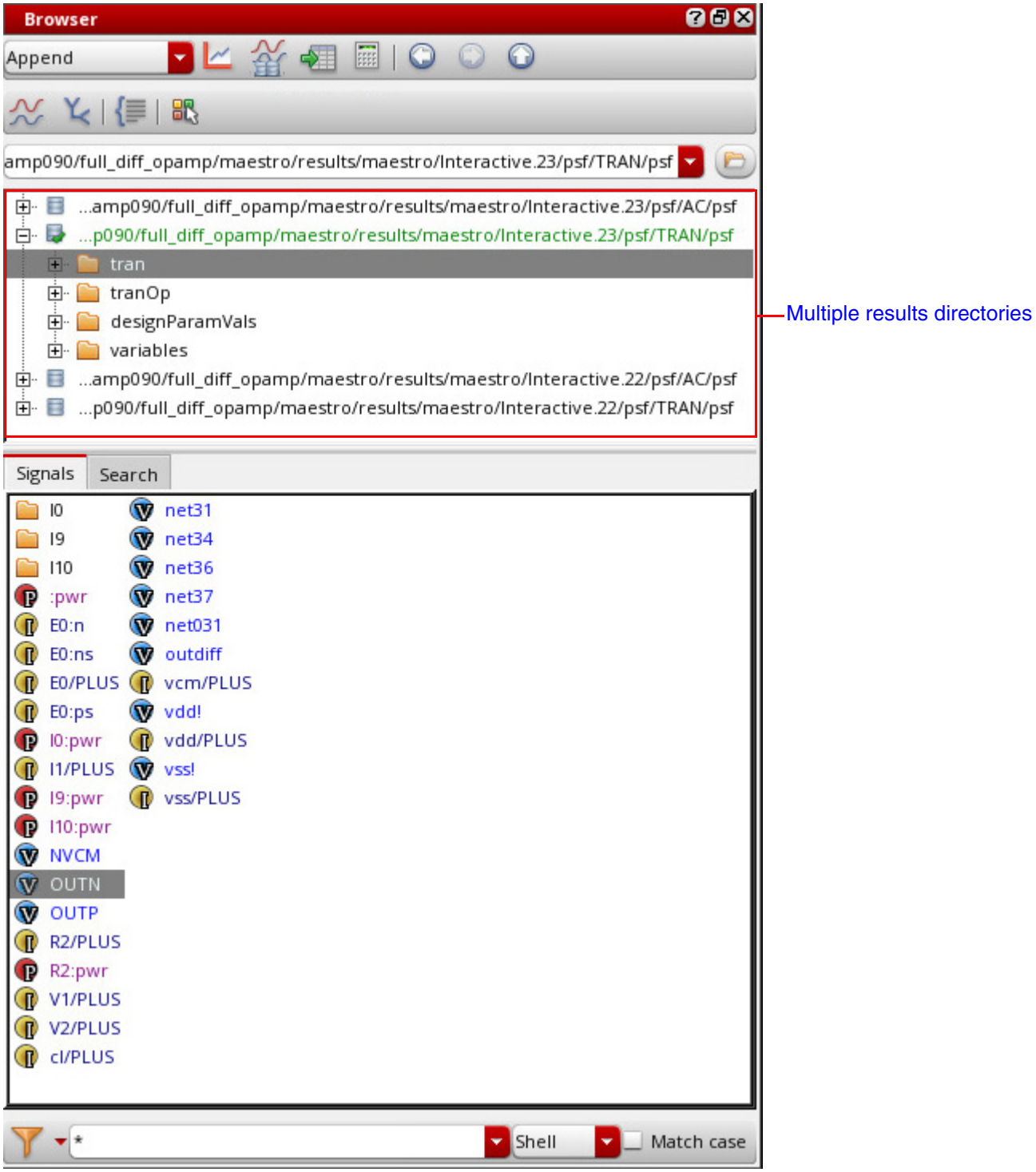
Comparing Signals from Different Results Directories

If you run multiple simulations on the same data, multiple results directories containing the same signals with different values are generated.

You can open these results directories in the Results Browser to compare the signals contained in the results directories. The signals from all the results directories are plotted in a graph window in the append mode to facilitate comparison.

© 2023 Cadence Design Systems, Inc.

The following figure shows multiple results directories that are created after running simulations on same data.



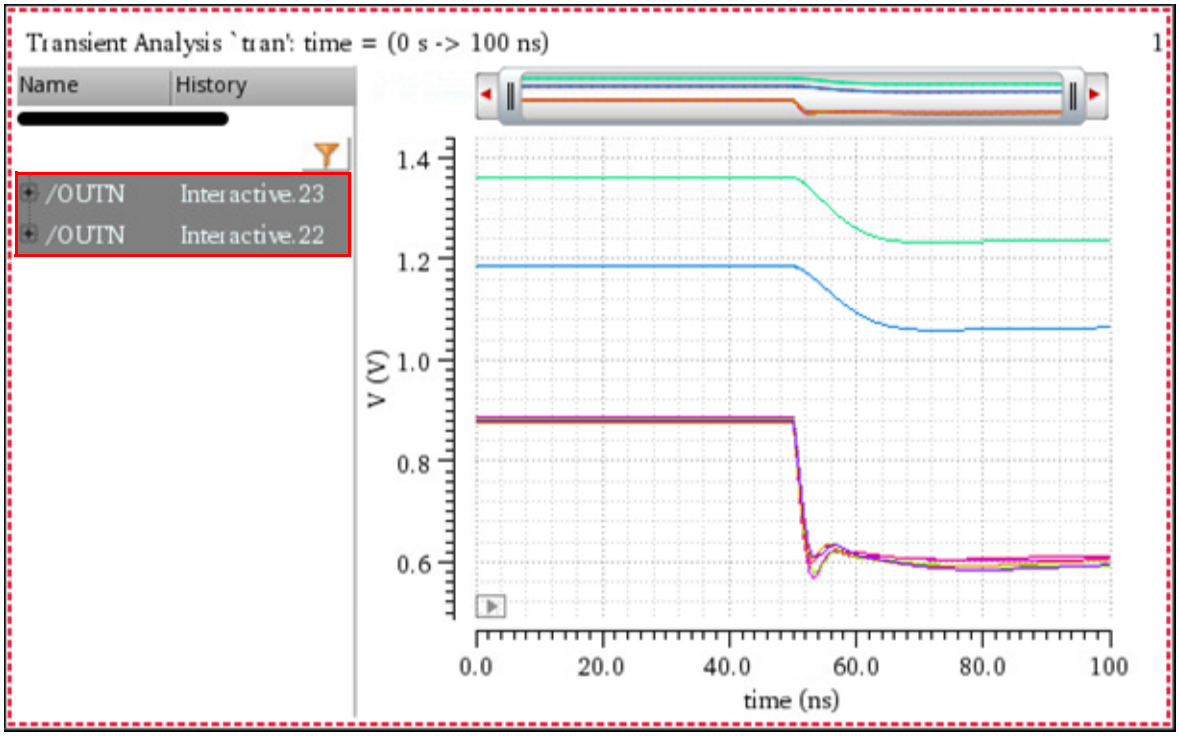
© 2023 Cadence Design Systems, Inc.

To compare signals from different results directories:

➡ Select a signal in the Results Browser, and click *Plot signal from all the open DBs* in the toolbar of the results browser.

Alternatively, right-click a signal in the Results Browser, and click *Plot from all DBs* .

The following figure shows the signal `OUTN` plotted from two different results directories. The two signals are overlapped because they have same data values.



Related Topics

Selecting a Signal in the Results Browser

Plotting a Signal from the Results Browser

© 2023 Cadence Design Systems, Inc.

Exporting Waveforms from a Single Results Directory

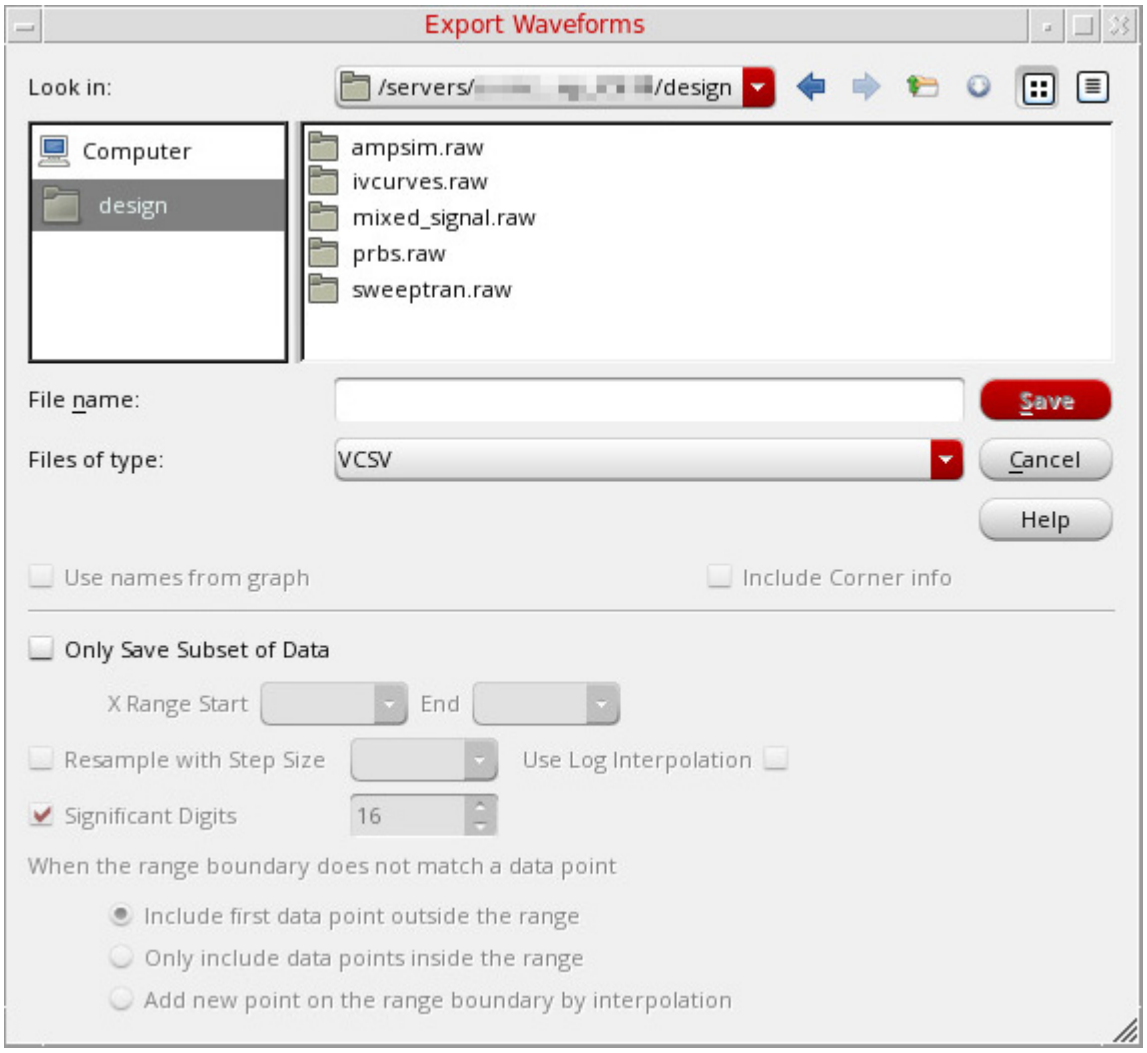
You can export a signal from a results directory in a variety of formats and later load these signals in the required application. You can also save a part of the dataset by specifying the start and end values or interpolate data before saving it.

To export a waveform from a single results directory:

1. Right-click a signal in the Results Browser and choose *Export*.

   Alternatively, select a signal in the Results Browser and choose *Browser – Results – Export* from the menu bar of Virtuoso Visualization and Analysis XL.

   The Export Waveforms form opens.



© 2023 Cadence Design Systems, Inc.

Note: You can select multiple signals by holding down the `Ctrl` key while you click the required signals in the Results Browser.

2. In the *File name* field, type the name of the file to which you want to export the selected waveforms.
3. From the *Files of type* drop-down list, select one of the following formats: *VCSV*, *Matlab*, *PSF*, *SPECTRE*, *SST2*, *CSV*, or *PSFXL*.
4. Select the *Use names from grap*h check box to use the trace names that are displayed on the graph.

   This check box is disabled when you export a signal from the Results Browser.
5. If you are exporting signals in CSV format, select the *Include Corner info* check box to print the corner information in the header of the CSV file.

   The *Include Corner info* check box appears dimmed and becomes available only if *CSV* is selected from the *Files of type* drop-down list.
6. Select the *Only Save Subset of Data* check box to export only the required part of the waveform or signal.
7. In the *X Range Start* and *End* fields, specify the start and the end points of the x range for which you want to export the waveform.
8. Select the *Resample with Step Size* check box to specify the step size to resample the exported data based on linear interpolation. The greater the step size, the fewer data points in the exported data.
9. Select the *Use Log Interpolation* check box to use the logarithmic scale for interpolation.
10. In the *Significant Digits* field, specify the number of significant digits in the exported values from waveform data.
11. From the *When the range boundary does not match a data point* options, select one of the following:

    - *Include first data point outside the range*: Includes the first data point that immediately falls before the specified *X Range Start* value and the last data point that falls immediately after the specified *X Range End* value when there are no actual data points corresponding to the specified *X Range Start* and *End* values
    - *Only include data points inside the range*: Includes only actual data points inside the specified x range.
    - *Add new point on the range boundary by interpolation*: Includes the interpolated values for the specified range boundary (*X Range Start* and *End*

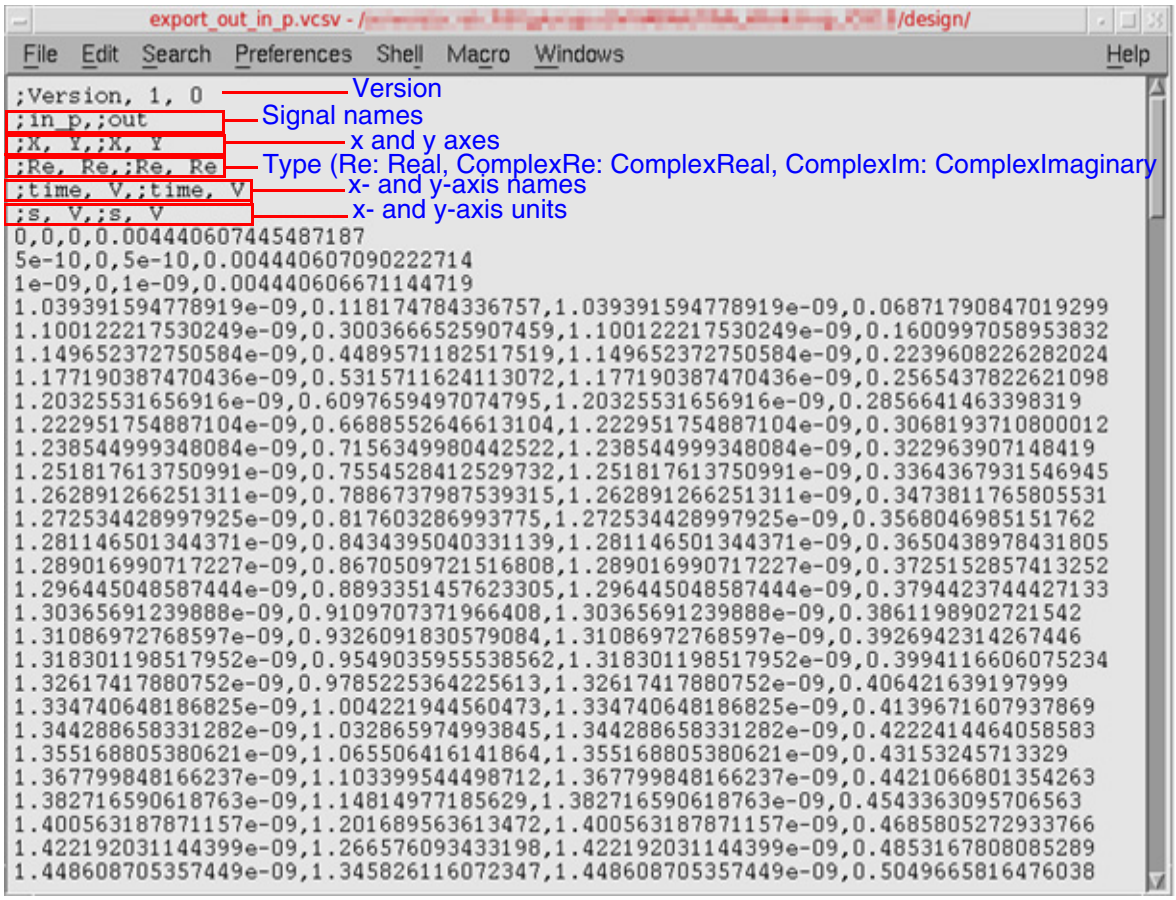
© 2023 Cadence Design Systems, Inc.

values) when there are no actual data points corresponding to the specified *X Range Start* and *End* values.

12. Click *Save*.

Waveform data is exported to a file in the specified format.

The following example shows the contents of the VCSV file when you export signals `out` and `in_p` in VCSV format.



When you export a signal in the MATLAB format, a file with the `.matlab` extension is created, for example, `test.matlab`. To open this file in MATLAB, type the following command in the MATLAB terminal window:

```
T=readtable('test.matlab','FileType','text')
```

This command creates a text-based table from the file and lists the file contents in tabular format.

© 2023 Cadence Design Systems, Inc.

Related Topics

The Export Waveforms Form

Exporting Waveforms from Multiple Results Directories

The Waveforms to be Exported Form

© 2023 Cadence Design Systems, Inc.

Contents of the Exported VCSV File

The header of the VCSV file contains the version, signal names, x and y axes names, type of data, and x and y axes units.

The description of the VCSV file format is as follows.

```
<ASCII> ::= <HEADER> <WAVEFORMS> <br>
<HEADER>    ::= <VERSION> <TITLES> <AXES> <TYPES> <AXISNAMES> <UNITS> <br>
<VERSION>   ::= <COM> Version, <MAJOR>, <MINOR> <br>
<TITLES>    ::= <COM> <TITLE> [,<TITLE>]... <br>
<AXES>      ::= <COM> <X_AXIS>, <Y_AXIS> [,<X_AXIS>, <Y_AXIS>]...<br>
<TYPES>     ::= <COM> <X_TYPE>, <Y_TYPE> [,<X_TYPE>, <Y_TYPE>]...<br>
<UNITS>     ::= <COM> <X_UNIT>, <Y_UNIT> [,<X_UNIT>, <Y_UNIT>]...<br>
<WAVEFORMS> ::= <XVAL>|<BLANK>, <YVAL>|<BLANK> [, <XVAL>|<BLANK>, <YVAL>|<BLANK>]
<TITLE>     ::=string representing signal name<br>
<MAJOR>     ::=     non-negative integer<br>
<MINOR>     ::=     non-negative integer<br>
<X_AXIS>    ::=     Re | Logic | LogicBus | String | Int
<Y_AXIS>    ::=    Re | Logic | LogicBus | String | Int | WRe | ComplexRe,ComplexIm
<X_TYPES>   ::=     string with the name of the axes
<Y_TYPES>   ::=     string with the name of the axes
<X_UNIT>    ::=     string with the name of the unit
<Y_UNIT>    ::=     string with the name of the unit
<XVAL>      ::=     string representing the value of corresponding <X_TYPE>
<YVAL>      ::=     string representing the value of corresponding <Y_TYPE>
<COM>       ::=     ;
```

Note the following points:

- The number of titles, axes, type and unit pairs match the number of `VAL|BLANK` pairs.
- The VCSV format can hold any number of waveforms.
- The waveform data is displayed in a tabular format. Each column of the table holds an `XVAL` and a `YVAL` for each point in the waveform.
- The column may be padded with blank entries for each waveform that is shorter than the longest waveform.
- Each `XVAL` or `YVAL` can either be a single real number or a pair of real numbers. If it is a pair, then the value is a complex number and the pair represents the real and imaginary portions. All elements after the semicolon are separated using commas (,).

© 2023 Cadence Design Systems, Inc.

Related Topics

The Export Waveforms Form

Exporting Waveforms from a Single Results Directory

Exporting Waveforms from Multiple Results Directories

The Waveforms to be Exported Form

© 2023 Cadence Design Systems, Inc.

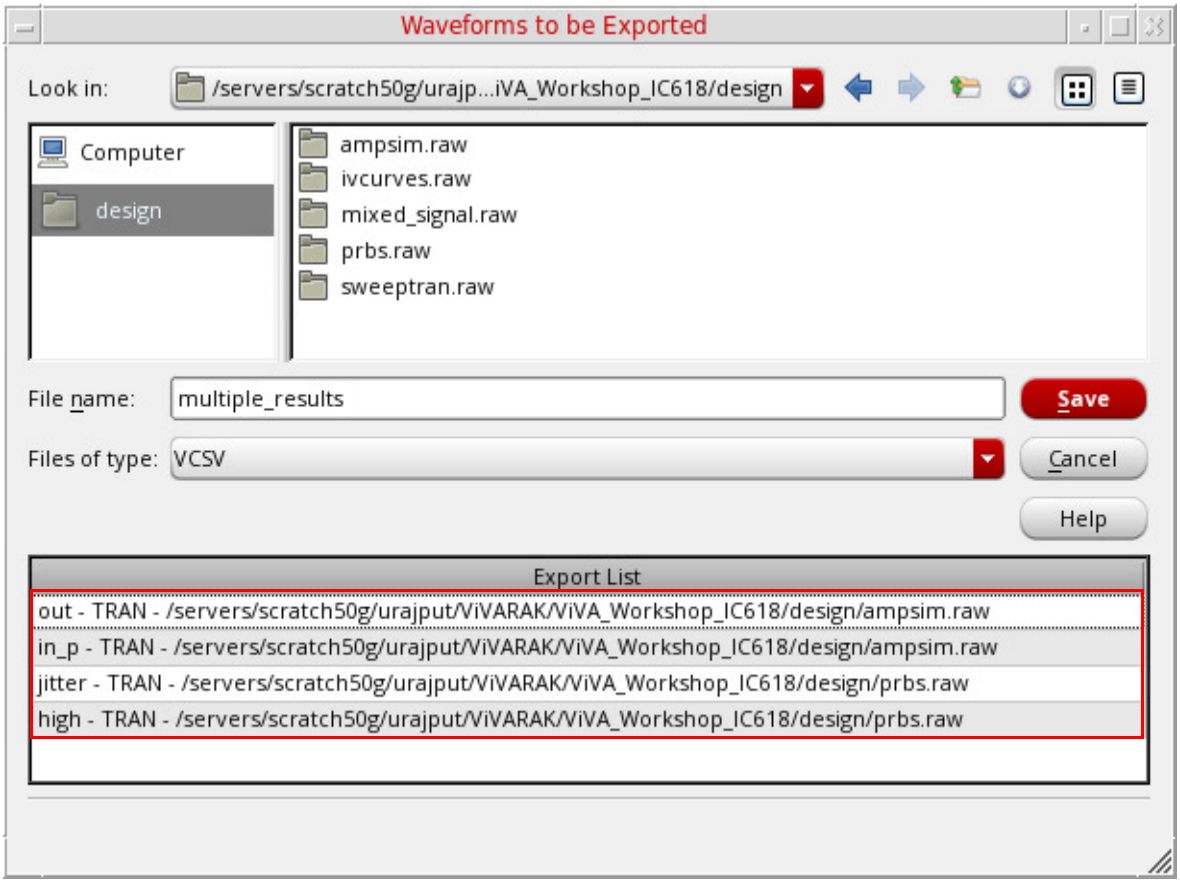
Exporting Waveforms from Multiple Results Directories

You can select signals either from the same or form different results directories and export them to a common file.

To export signals from multiple results directories:

1. Open a results directory in the Results Browser.
2. Right-click a signal and choose *Send to Export List*. Hold down the `Ctrl` key to select multiple signals from the same results directory.

   The Waveforms to be Exported form opens with the selected signals displayed in the *Export List*.



   To delete a signal from the *Export List*, right-click the signal and choose *Delete*. To delete all signals from the *Export List*, right-click any signal and choose *Delete All*.

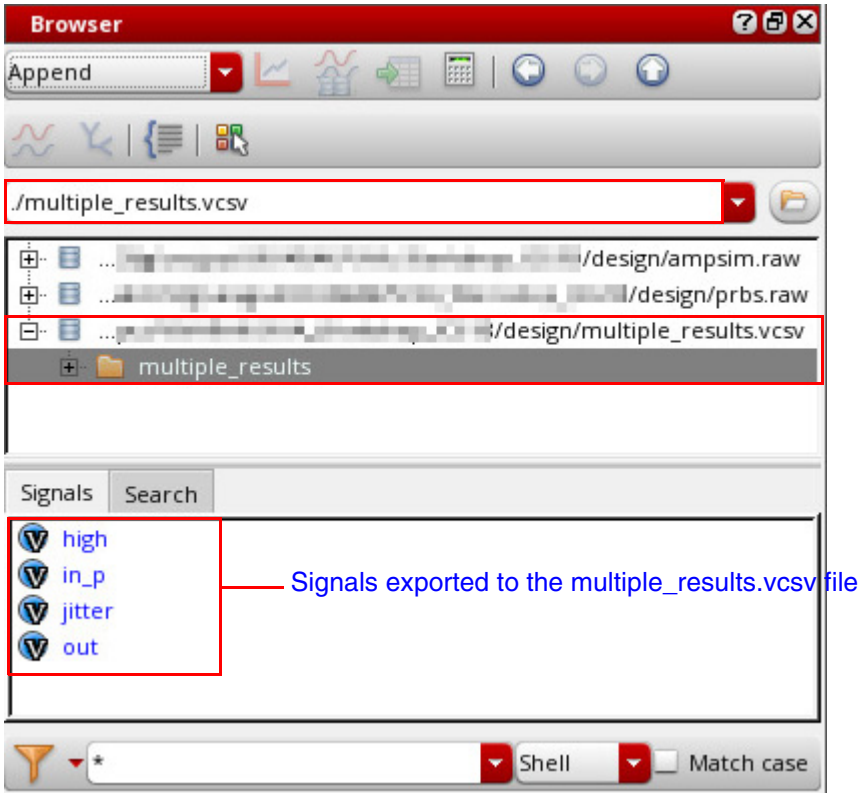
3. Open another results directory in the Results Browser, and repeat step 2.

© 2023 Cadence Design Systems, Inc.

4. In the *File name* field, enter the name of file to which you want to export the signals.
5. From the *Files of type* drop-down list, select one of the following formats: *VCSV*, *Matlab*, *PSF*, *SPECTRE*, *SST2*, *CSV*, or *PSFXL*
6. Click *Save*.

   The selected signals are exported to file of specified format. For example, `mutiple_results.vcsv` file contains signals `out` and `in_p` exported from `ampsim.raw` results directory and signals `jitter` and `high` exported from `prbs.raw` results directory.

   You can reload these signals into the Results Browser using the exported file, as shown in the following figure.



Related Topics

The Waveforms to be Exported Form

The Export Waveforms Form

Exporting Waveforms from a Single Results Directory

© 2023 Cadence Design Systems, Inc.

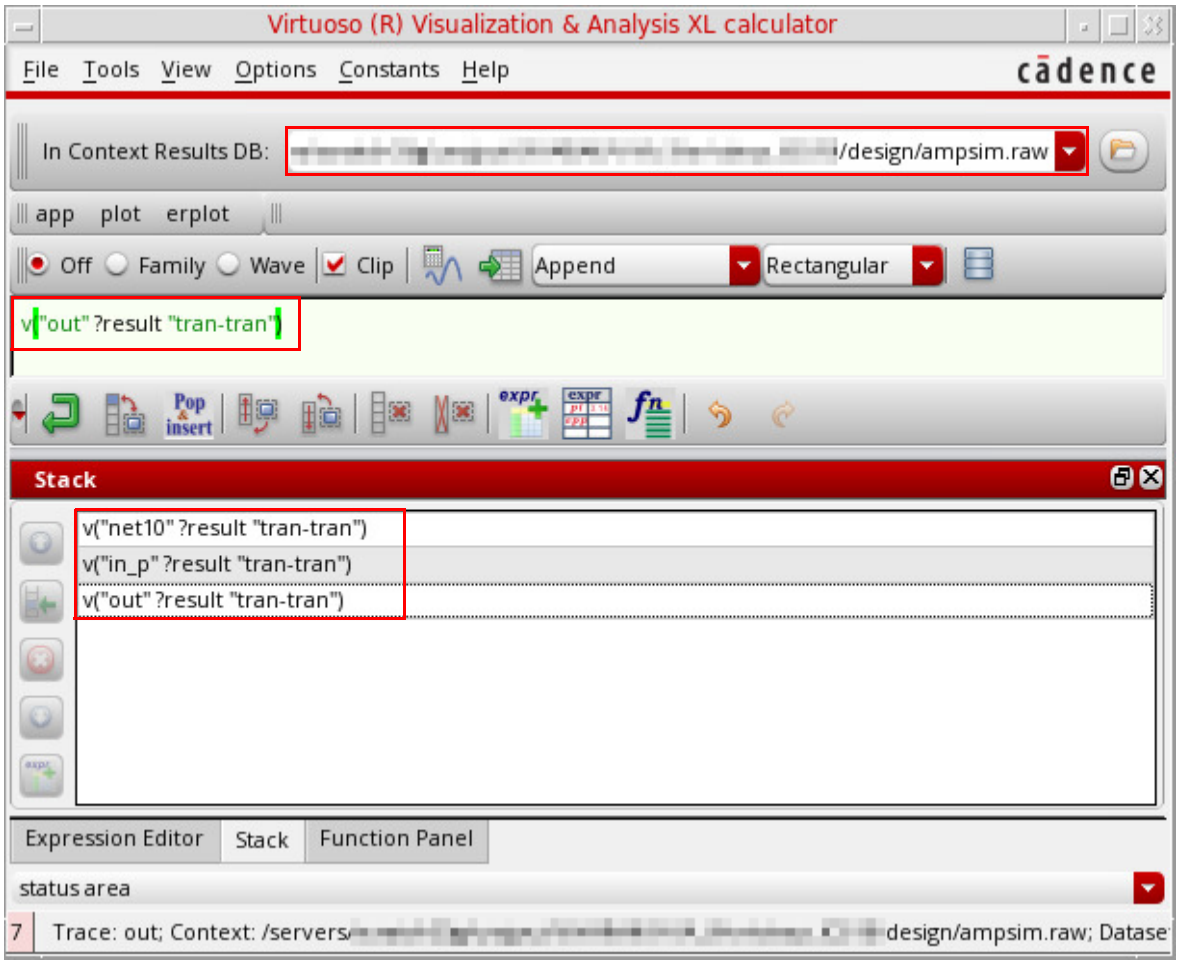
Sending a Signal to Calculator from Results Browser

To send a signal to Calculator from the Results Browser:

1. In the Results Browser, select a signal. Hold down the `Ctrl` key to select multiple signals.
2. From the toolbar of the Results Browser, click the *Send to Calculator* icon.

   Alternatively, right-click the selected signals in the Results Browser and then choose Calculator.

   The Calculator window opens, displaying the selected signals. The results directory of these signals is also set as the in-context results database in the Calculator.



© 2023 Cadence Design Systems, Inc.

Related Topics

Sending a Signal to Table from Results Browser

© 2023 Cadence Design Systems, Inc.

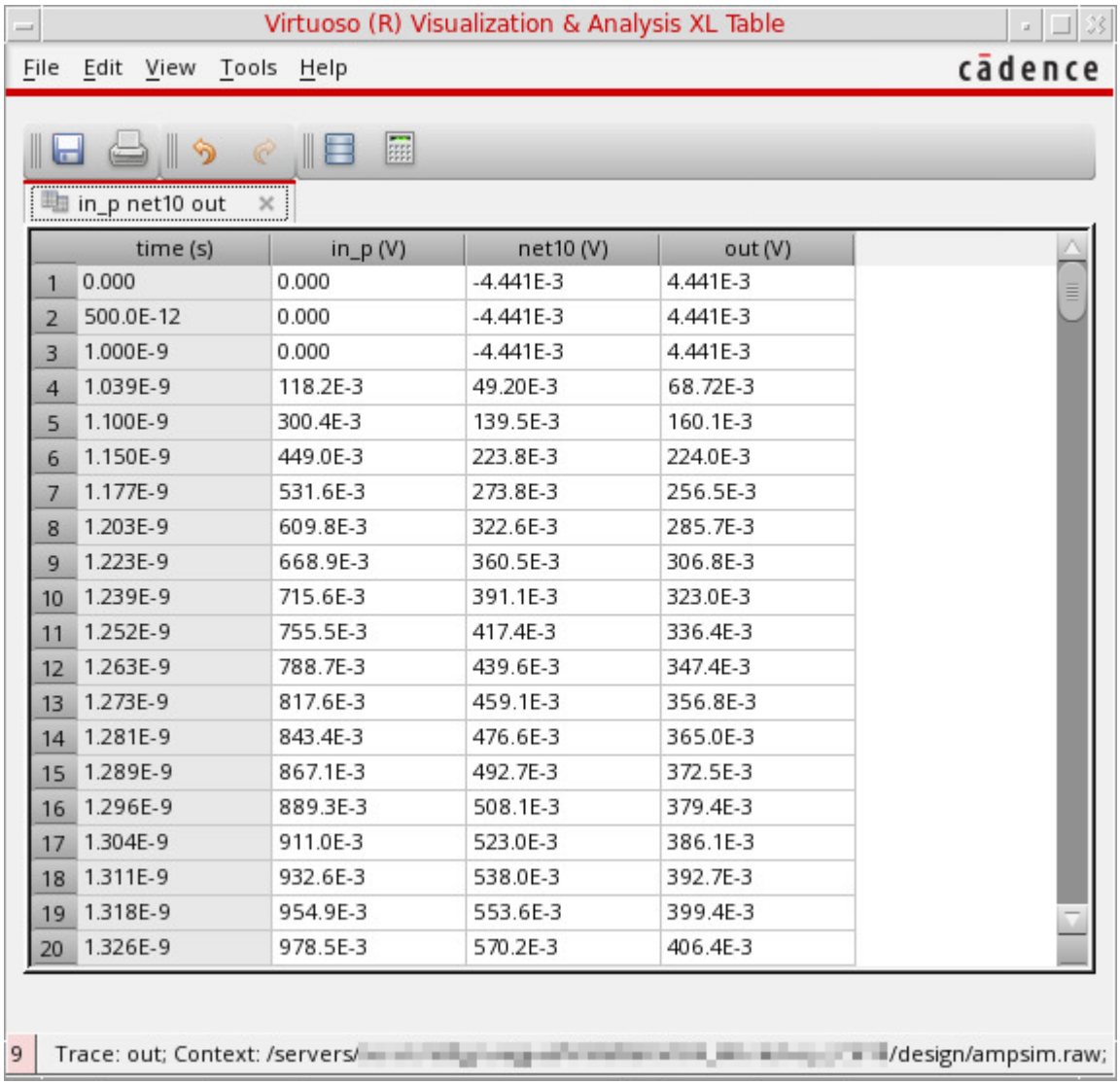
Sending a Signal to Table from Results Browser

To send a signal from Results Browser to Virtuoso Visualization and Analysis XL Table:

1. In the Results Browser, select a signal. Hold down the `Ctrl` key to select multiple signals.
2. From the toolbar of the Results Browser, click the *Send to Table* icon.

   Alternatively, right-click the selected signals in the Results Browser and then choose *Table*.

   The Virtuoso Visualization and Analysis XL Table opens, displaying data points of the selected signals.



	time (s)	in_p (V)	net10 (V)	out (V)
1	0.000	0.000	-4.441E-3	4.441E-3
2	500.0E-12	0.000	-4.441E-3	4.441E-3
3	1.000E-9	0.000	-4.441E-3	4.441E-3
4	1.039E-9	118.2E-3	49.20E-3	68.72E-3
5	1.100E-9	300.4E-3	139.5E-3	160.1E-3
6	1.150E-9	449.0E-3	223.8E-3	224.0E-3
7	1.177E-9	531.6E-3	273.8E-3	256.5E-3
8	1.203E-9	609.8E-3	322.6E-3	285.7E-3
9	1.223E-9	668.9E-3	360.5E-3	306.8E-3
10	1.239E-9	715.6E-3	391.1E-3	323.0E-3
11	1.252E-9	755.5E-3	417.4E-3	336.4E-3
12	1.263E-9	788.7E-3	439.6E-3	347.4E-3
13	1.273E-9	817.6E-3	459.1E-3	356.8E-3
14	1.281E-9	843.4E-3	476.6E-3	365.0E-3
15	1.289E-9	867.1E-3	492.7E-3	372.5E-3
16	1.296E-9	889.3E-3	508.1E-3	379.4E-3
17	1.304E-9	911.0E-3	523.0E-3	386.1E-3
18	1.311E-9	932.6E-3	538.0E-3	392.7E-3
19	1.318E-9	954.9E-3	553.6E-3	399.4E-3
20	1.326E-9	978.5E-3	570.2E-3	406.4E-3

© 2023 Cadence Design Systems, Inc.

Related Topics

Sending a Signal to Calculator from Results Browser

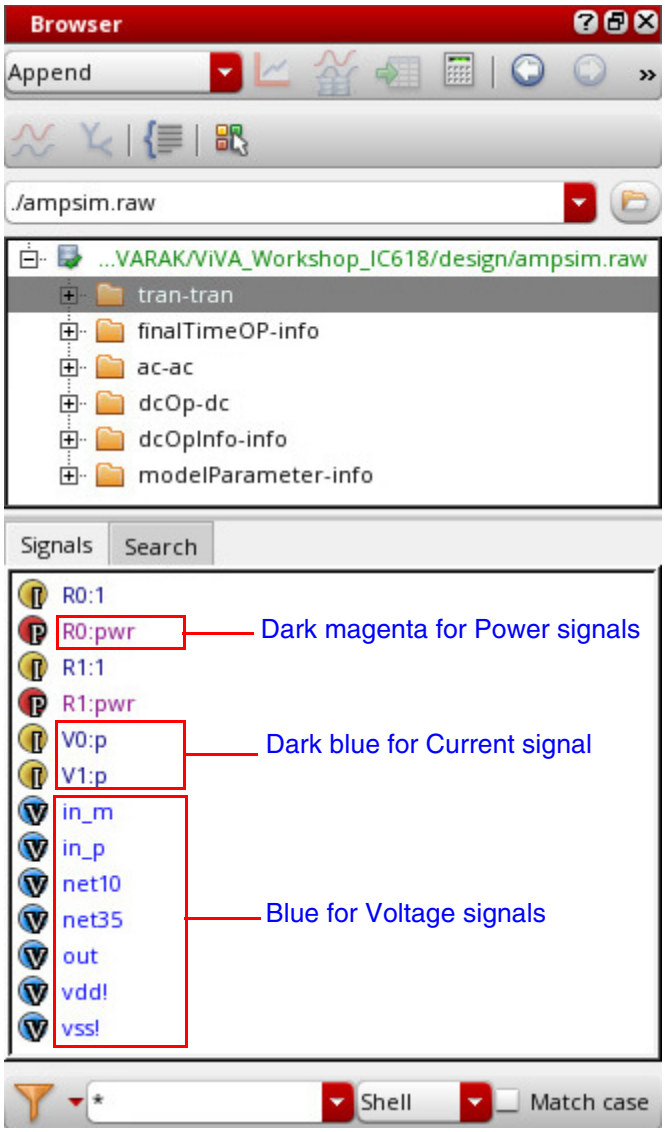
Opening the Table from the Results Browser

© 2023 Cadence Design Systems, Inc.

Setting Colors to Signal Types

You can set different colors to different signal types, such as voltage, current, analog, and digital signals. This makes it easy to differentiate between various signal types.

The following figure shows the default colors set for various signal types, such as voltage, power, and



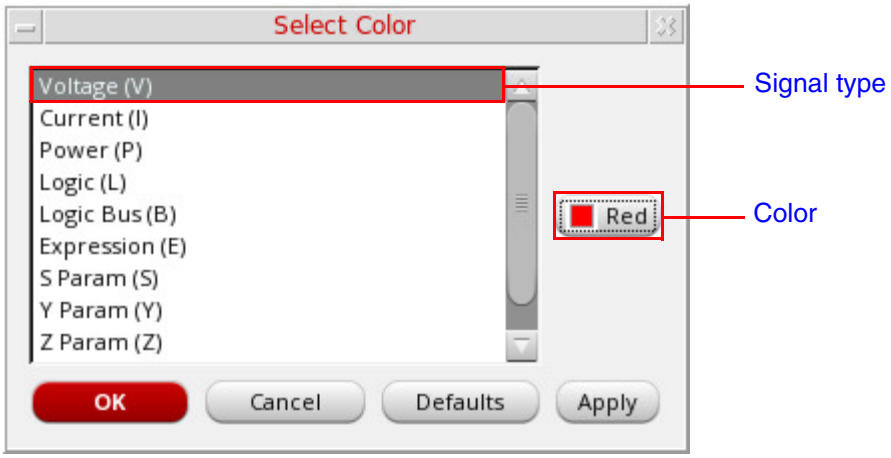
To set a different color for a signal type:

1. From the toolbar of the Results Browser, click the Change color of signals icon.

   The Select Color form opens.

© 2023 Cadence Design Systems, Inc.

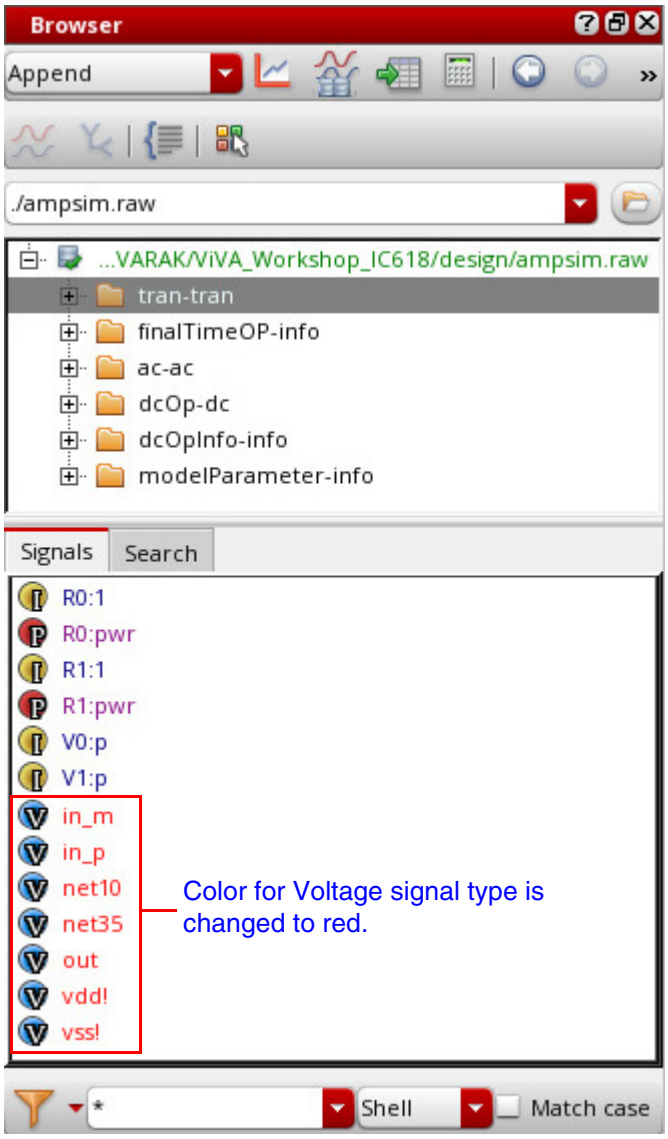
2. From the list of signal types, select the signal type for which you want to change the color.
3. Click the corresponding color and choose a different color you want to set for the selected signal type.



4. Click *Apply* and then click *OK*.

© 2023 Cadence Design Systems, Inc.

The color for the selected signal type is changed, as shown in the following figure.



Related Topics

Selecting a Signal in the Results Browser

Plotting a Signal from the Results Browser

© 2023 Cadence Design Systems, Inc.

Automatic Calculation of Two-Port Impedance Equivalent Parameters

Virtuoso Visualization and Analysis XL supports the industry-standard Touchstone format that can read data files created by any simulator. This format also helps plot n-port S-Parameter data if you use Spectre and SpectreRF as simulators.

The touchstone file is an ASCII file, which is also known as the `SnP` file. This file includes a large signals S-Parameter results. The touchstone files are of `.snp` extension, where `n` is the number of network ports of the device. For example, if the touchstone file contains the network parameters for a two-port device, it has a `.s2p` extension.

For a two-port network, the two-port impedance equivalent parameters, `C`, `L`, and `R` and the quality factor `Q` are calculated using the following formula:

Parameter	Formula
Effective Resistance	$R_{eff} = \text{real}(Z)$
Effective Inductance	$L_{eff} = \frac{\text{imag}(Z)}{\omega}$
Effective Capacitance	$C_{eff} = \frac{1}{(\text{imag}(Z) \times \omega)}$
Effective Quality Factor (Q-Factor)	$Q_{eff} = \frac{\text{imag}(Z)}{\text{real}(Z)}$

Where:

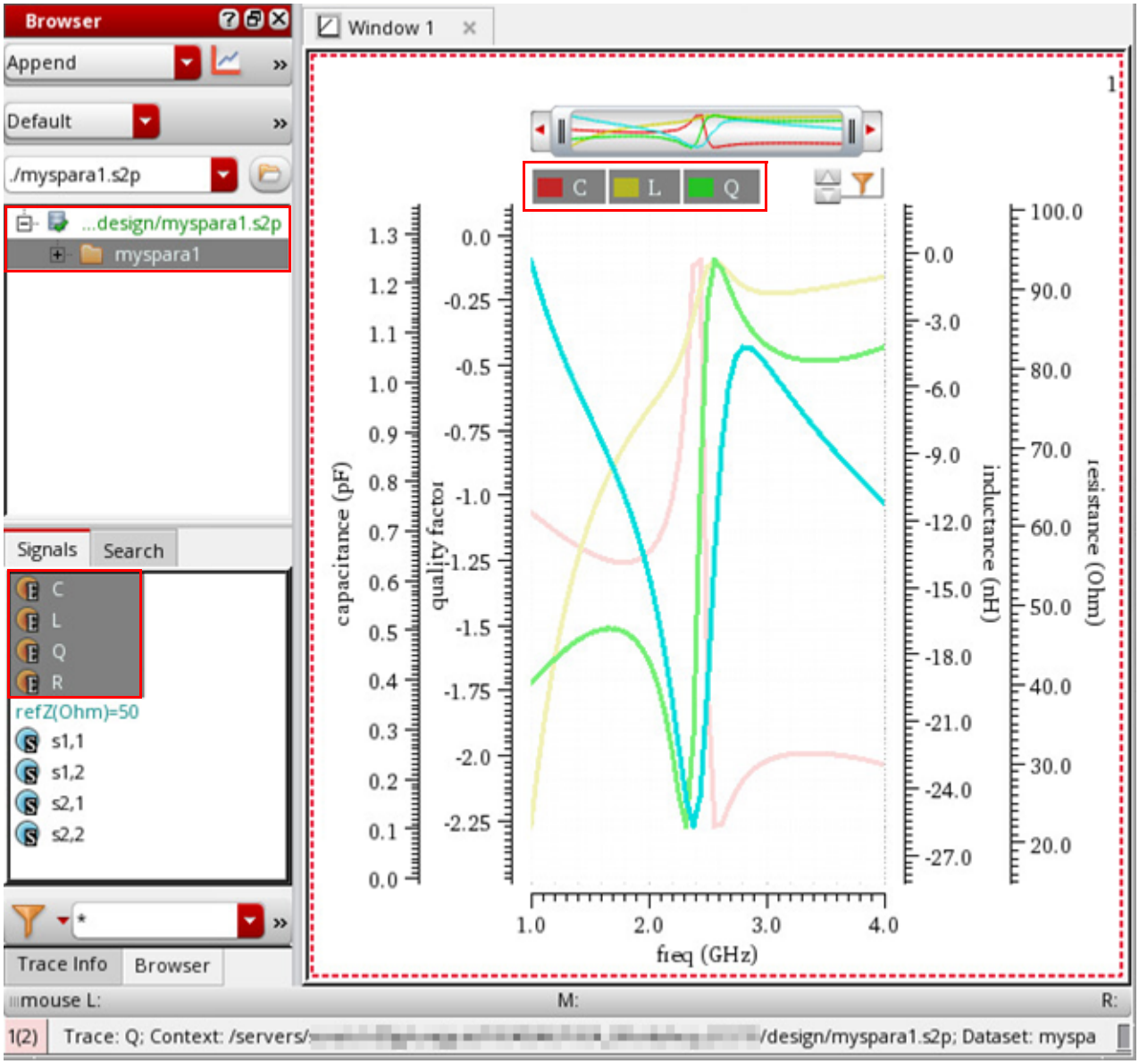
$\omega = 2 \times \Pi \times frequency$

$Z = \frac{1}{Y_{1,1}}$

These calculations are done automatically when you load the `.s2p` file in the Results Browser.

© 2023 Cadence Design Systems, Inc.

Using the S-Parameter touchstone file for a two-port network, you can also plot these values directly from Results Browser.



© 2023 Cadence Design Systems, Inc.

Searching for Signals in the Results Browser

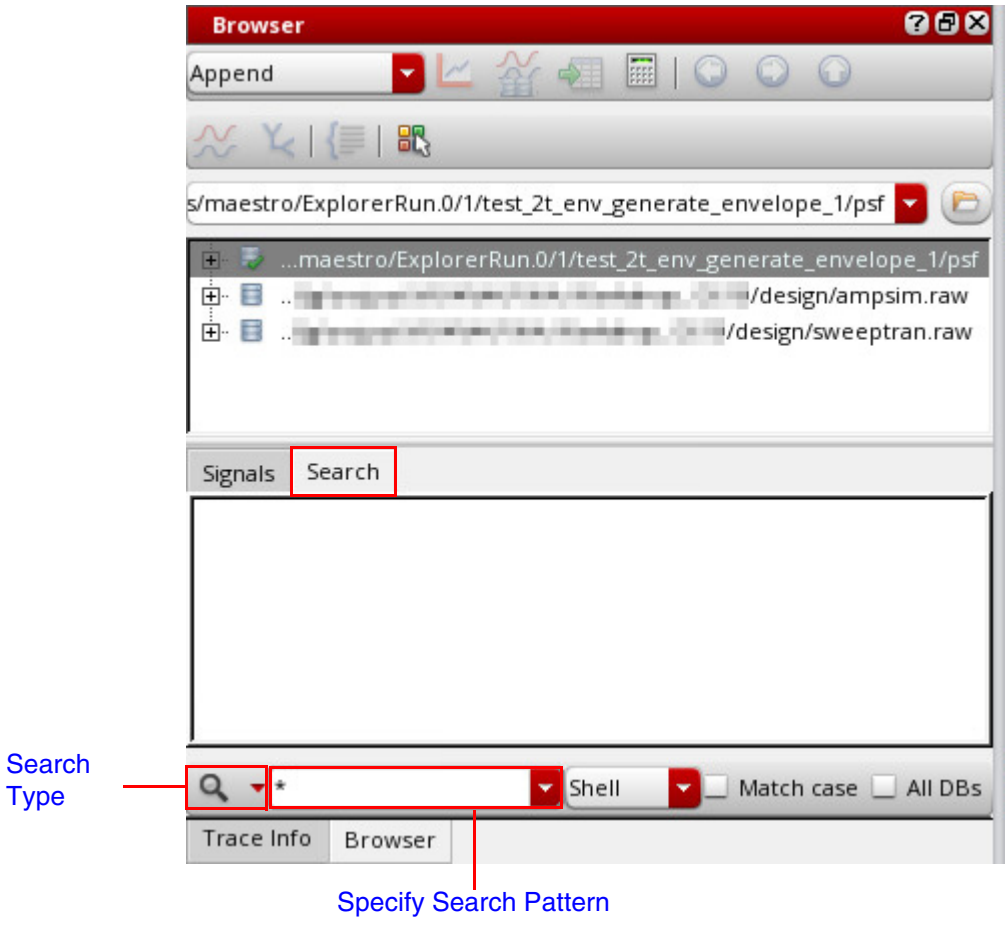
You can use the *Search* tab in the bottom panel of the Results Browser to search for the signals in the currently selected database or in databases that are currently open.

By default, signal search is performed at the directory level, which means only the currently selected directory and its subdirectories are searched to find the signals that match the specified search pattern.

However, you can also search for a signal in all databases that are currently open in the Results Browser by selecting the *All DBs* check box.

To search for a signal:

1. Click the *Search* tab in the bottom panel of the Results Browser.



2. Click the down arrow next to the *Search Type* icon to specify the type of signal you want to search.

© 2023 Cadence Design Systems, Inc.

3. In the *Specify Search Pattern* field, type the search pattern for the signals you want to search. For example, type:
   - `v[dd|ss]*` to search signals that contain `vdd` and `vss` in their names.
   - `net[1-9]*` to search signals that contain `net` and any number from `0-9` in their names.
4. From the drop-down list, select one of the following options:
   - *Shell*: To search signals using a shell expression. This is the default selection.
   - *RegExp*: To search signals using a regular expression.
5. Select the *Match Case* check box to perform a case-sensitive search. By default, search is not case sensitive.
6. Select the *All DBs* check box to search signals in all open results databases in the Results Browser. If this check box is not selected, the signals are searched only in the currently selected results database.
7. Either click the *Search* icon or press the *Enter* key on the keyboard.

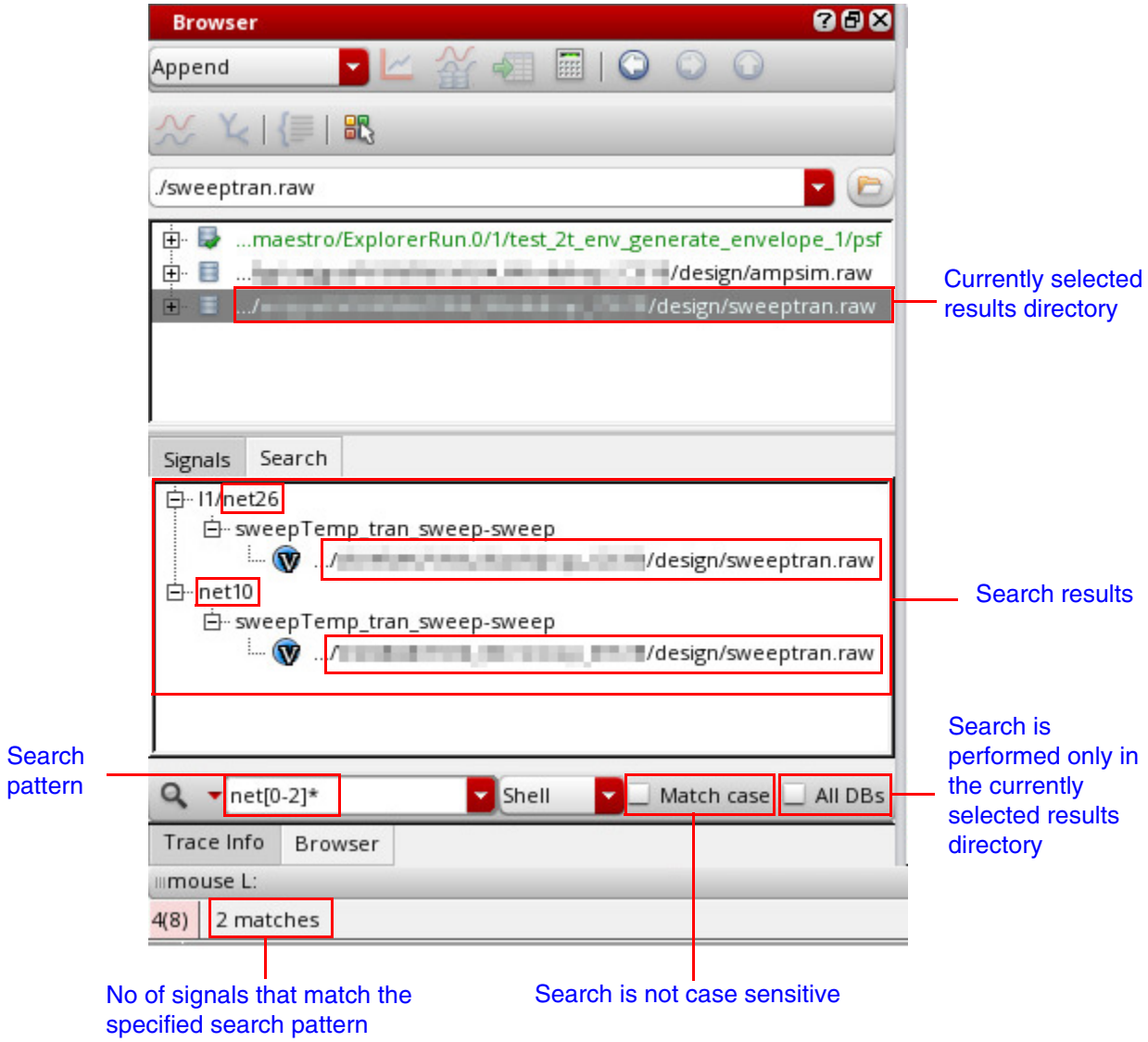
   The signals that match the specified search pattern are displayed.

Examples

The following example searches the signals that have `net` and a number between `0-2` in their names. Signals are searched only in the currently selected results directory. The search is not case sensitive.

© 2023 Cadence Design Systems, Inc.

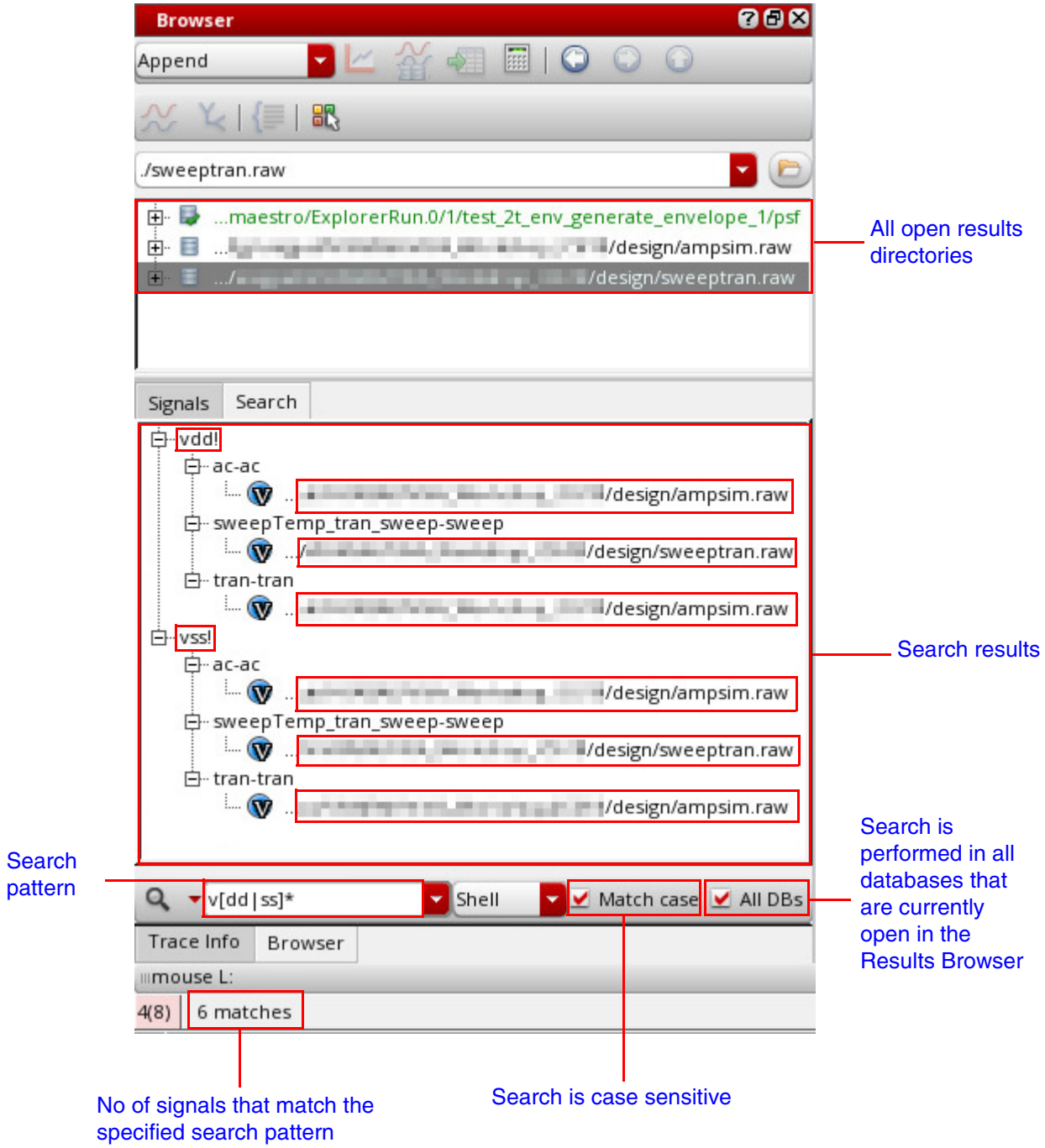
When you click the *Search* icon or press the *Enter* key on the keyboard, following search results are displayed. Signals that are returned in the search results are `net26` and `net10`.



The following example searches the signals that have either `vdd` or `vss` in their names. Signals are searched in all results databases that are currently open in the Results Browser. The search is case sensitive.

© 2023 Cadence Design Systems, Inc.

When you click the *Search* icon or press the *Enter* key on the keyboard, following search results are displayed. Signals that are returned in the search results are `vdd` and `vss`.



© 2023 Cadence Design Systems, Inc.

Related Topics

Opening the Results Browser

Results Browser Graphical User Interface

Opening a Results Directory

Filtering Signals in the Results Browser

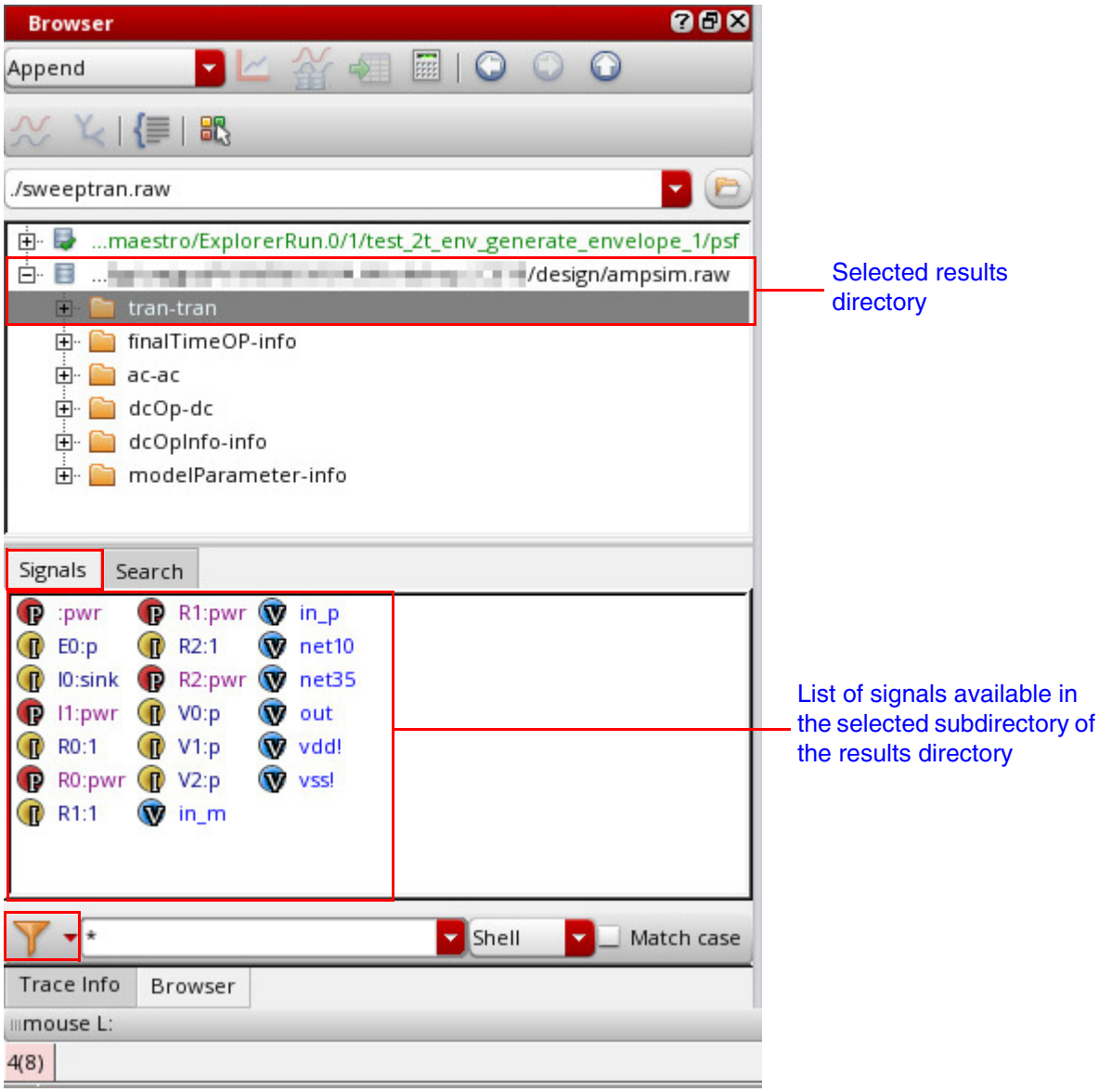
© 2023 Cadence Design Systems, Inc.

Filtering Signals in the Results Browser

You can use the *Signals* tab in the bottom panel of the Results Browser to filter the signals of your interest in the results directory that is currently selected in the results browser.

To filter signals in the Results Browser:

1. Open a results directory in the Results Browser and navigate to the subdirectory in which you want to filter the signals.



2. Click the *Signals* tab in the bottom panel of the Results Browser.

© 2023 Cadence Design Systems, Inc.

3. Click the down arrow next to the *Filter* icon to specify the type of signal you want to search. Depending on the type of signals the selected results subdirectory contain, options in this drop-down may vary.

   You can also select the *Show hierarchical node* check box to view folders that represent the hierarchical sub-blocks in the Results Browser.

4. In the *Specify Search Pattern* field, type the search pattern for the signals you want to filter. For example, type:

   - `:*` to selectively view only those signals whose names start with a `:`.
   - `*:*` to selectively view only those signals that contain a `:` in their names.
   - `*[0-2]*` to selectively view only those signals that contain any number from `0-2` in their names.

5. From the drop-down list, select one of the following options:

   - *Shell*: To filter signals using a shell expression. This is the default selection.
   - *RegExp*: To filter signals using a regular expression.

6. Select the *Match Case* check box to apply a case-sensitive filter.

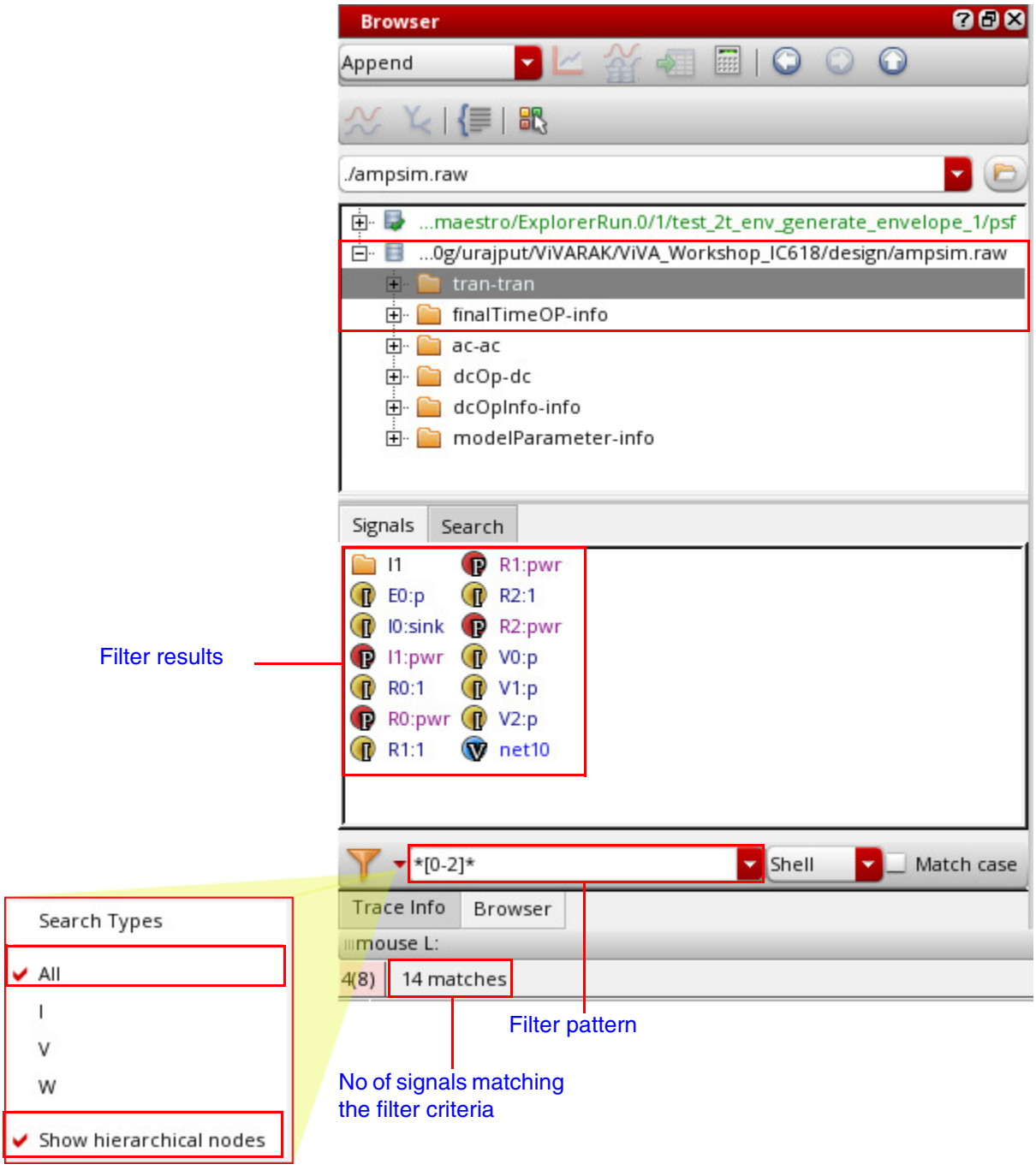
7. Either click the *Filter* icon or press the *Enter* key on the keyboard.

   The signals that match the specified filter pattern are displayed.

The following example filters the signals that have a number from `0-2` in their names. Signals are filtered in the currently selected subdirectory of the results directory. The filter applied is not case sensitive.

© 2023 Cadence Design Systems, Inc.

When you click the *Filter* icon or press the *Enter* key on the keyboard, following signals are displayed in the results.



© 2023 Cadence Design Systems, Inc.

Related Topics

Opening the Results Browser

Results Browser Graphical User Interface

Opening a Results Directory

Searching for Signals in the Results Browser

© 2023 Cadence Design Systems, Inc.

Plotting Modes

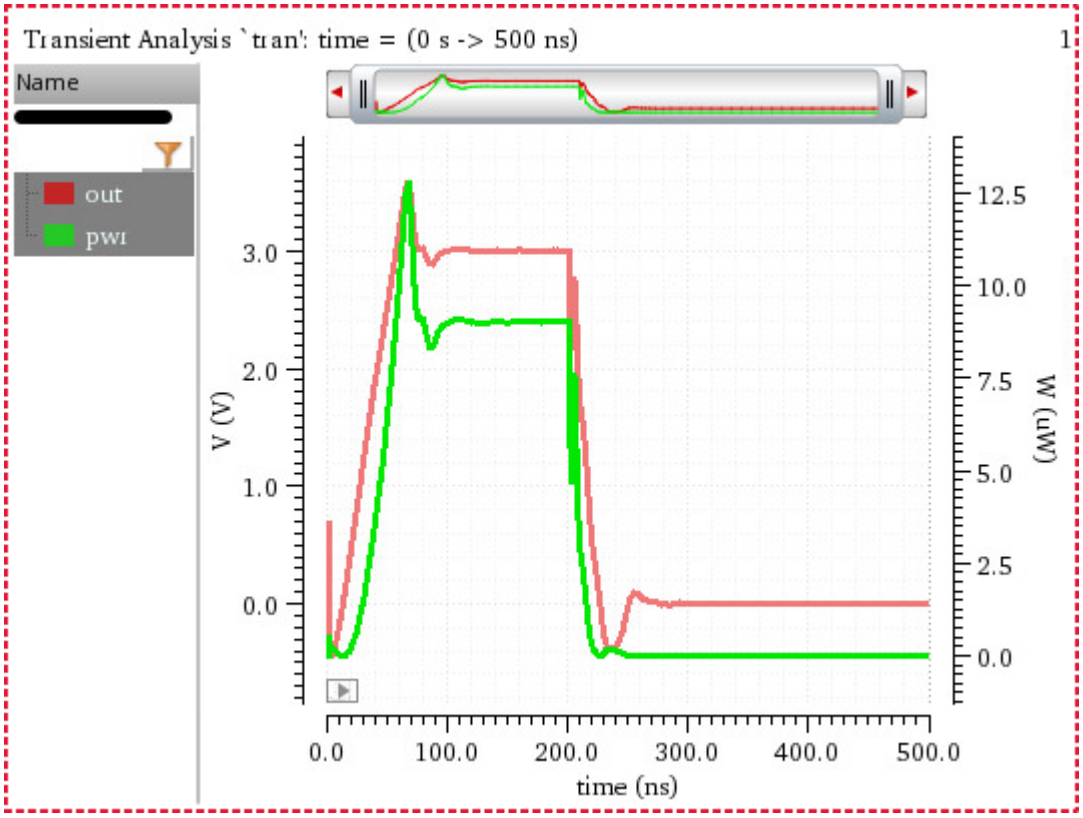
The first graph that you plot in a session is displayed in a new graph window. If you already have a graph window open, you need to specify the destination to plot the new signals.

You can plot a new signal in any of the following modes:

- **Append**

  Appends the trace for the selected signal to the graph that is already plotted in the graph window or subwindow. It the new trace does not share the same y-axis unit as that of the existing trace, the new trace is assigned a new y-axis. If the graph window already has four y axes, or if the units do not match, Virtuoso Visualization and Analysis XL plots the new signal in a new subwindow in that graph window.

  The following example shows the signal `pwr` appended to the signal `out` that is already plotted in the subwindow `1` of a graph window. The signal `out` has `volt` as its y axis, whereas the signal `pwr` has `watt` as its y axis.



- **Replace**

© 2023 Cadence Design Systems, Inc.

Replaces a signal or group of signals in the selected graph window or subwindow with the new signals.

- **New Subwindow**

  Plots the selected signals in a new subwindow of the current graph window.

- **New Window**

  Plots the selected signals in a new graph window.

Related Topics

Selecting a Signal in the Results Browser

Plotting a Signal from the Results Browser

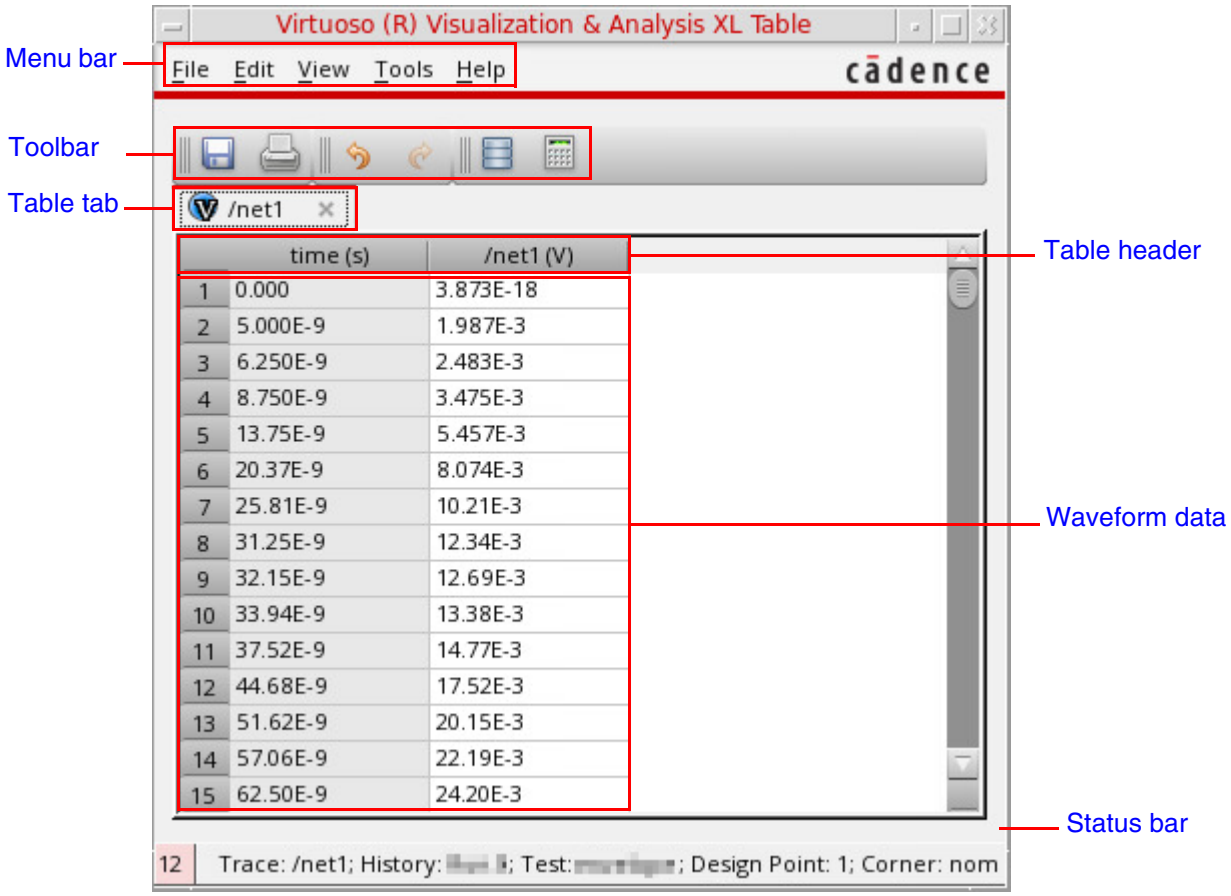
© 2023 Cadence Design Systems, Inc.

3

Virtuoso Visualization and Analysis XL Table

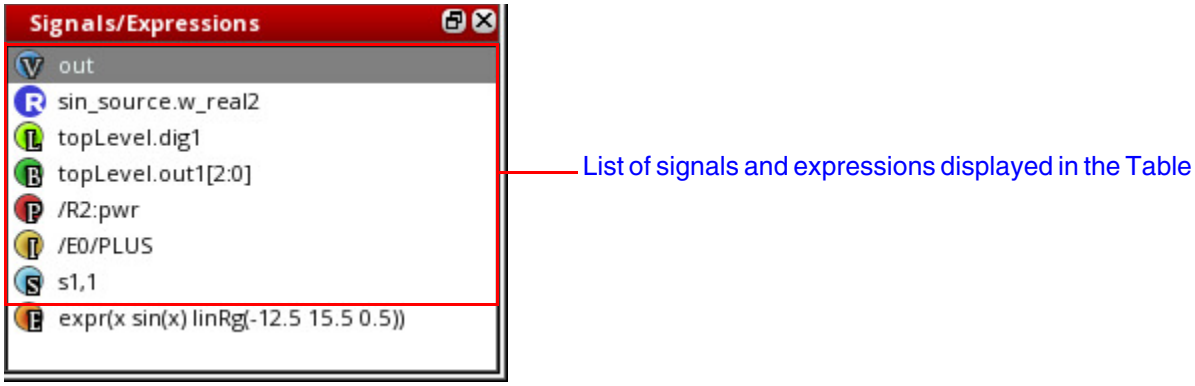
Virtuoso Visualization and Analysis XL Table (hereinafter referred to as Table) displays data for the selected traces or signals in a table for quick analysis.

The graphical user interface (GUI) of the Table consists of a menu bar, a toolbar, and a status bar. The following figure shows GUI of the Table.



© 2023 Cadence Design Systems, Inc.

To view signal names or expression for traces, right-click the Table toolbar or the status bar and choose *Signals/Expressions*. The Signal/Expressions window appears on the left, displaying the signal names or expressions for which data is displayed in the Table.



Related Topics

Menu Bar of the Table

Toolbar of the Table

Opening the Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

Menu Bar of the Table

The following table described the menu options available in the menu bar of the Virtuoso Visualization and Analysis XL Table.

Menu	Option	Icon	Description
File	*Export*		Exports waveform data to the specified format.
	Print		Prints the contents of the Table.
	Close	–	Closes the Table.
Edit	*Undo*		Undoes the previously performed action.
	Redo		Redoes the previously performed action. This command is available only when you have undone a previous action.
	Preferences	–	Opens the Preferences form where you can specify the number of operations that can be undone and redone.
View	*Hide*	–	Hides the selected column of the Table. This command is available after you select a column in the Table.
	Reveal	–	Opens the Reveal Columns form where you can select the hidden columns that you want to reveal again. This command is not available if there are no hidden columns in the Table.
	Format	–	Opens the Format Attributes form where you can specify the format of the cell and header contents of the Table.
	Sort	–	Sorts the selected column in the ascending or descending order.
	Transpose	–	Transposes the rows and columns of the Table.
	Default Column Order	–	Sets the order of the columns in the Table to the default setting. The default column order displays the column containing x-axis values to the left of the column containing y-axis data.

© 2023 Cadence Design Systems, Inc.

Menu	Option	Icon	Description
Tools	*Browser*		Opens the Results Browser.
	Calculator		Opens the Virtuoso Visualization and Analysis XL Calculator.
Help		–	Hosts the links to various help contents.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Toolbar of the Table

Exporting Contents of the Table

Printing Contents of the Table

Hiding Columns in the Table

Revealing Columns in the Table

Sorting Columns in the Table

Transposing Rows and Columns of the Table

Formatting Attributes of the Table

© 2023 Cadence Design Systems, Inc.

Toolbar of the Table

The following table describes the commands that are available in the toolbar of the Virtuoso Visualization and Analysis XL Table.

Menu	Option	Icon	Description
File	*Save as CSV*		Exports waveform data to a CSV file.
	Print		Prints the contents of the Table.
Edit	*Undo*		Undoes the previously performed action.
	Redo		Redoes the previously performed action. This command is available only when you have undone a previous action.
Tools	*Browser*		Opens the Results Browser.
	Calculator		Opens the Virtuoso Visualization and Analysis XL Calculator.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

Saving Table Contents to a CSV File

Printing Contents of the Table

Limiting the Number of Actions to Undo or Redo

© 2023 Cadence Design Systems, Inc.

Opening the Virtuoso Visualization and Analysis XL Table

You can open the Virtuoso Visualization and Analysis XL Table in any of the following ways:

- Opening the Table from the Results Browser
- Opening the Table from the Graph Window
- Opening Table from the Calculator

Opening the Table from the Results Browser

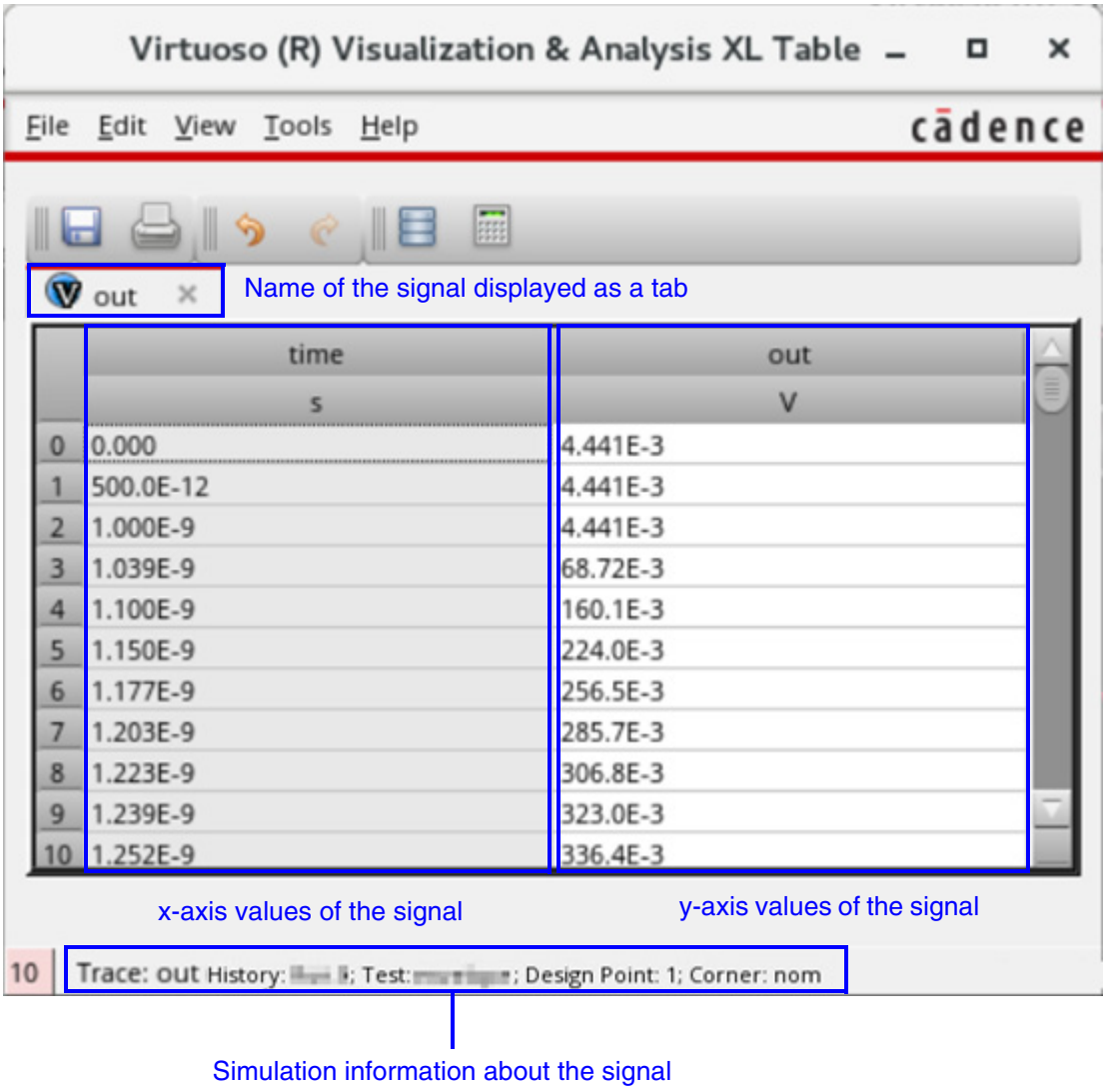
To open the Table from the Results Browser:

1. In the Results Browser, select the signal whose waveform data you want to display in the Table. Hold down the `Ctrl` key to select multiple signals.
2. From the toolbar of the Results Browser, click *Send to Table* .

   Alternatively, right-click the selected signals in the Results Browser and then choose *Table*.

© 2023 Cadence Design Systems, Inc.

The Virtuoso Visualization and Analysis XL Table opens, displaying data points of the selected signals.



The default order of the columns displayed in the Table is as follows:

- The column to the left displays x-axis values of the signal.
- The column to the right displays y-axis values of the signal.

© 2023 Cadence Design Systems, Inc.

Opening the Table from the Graph Window

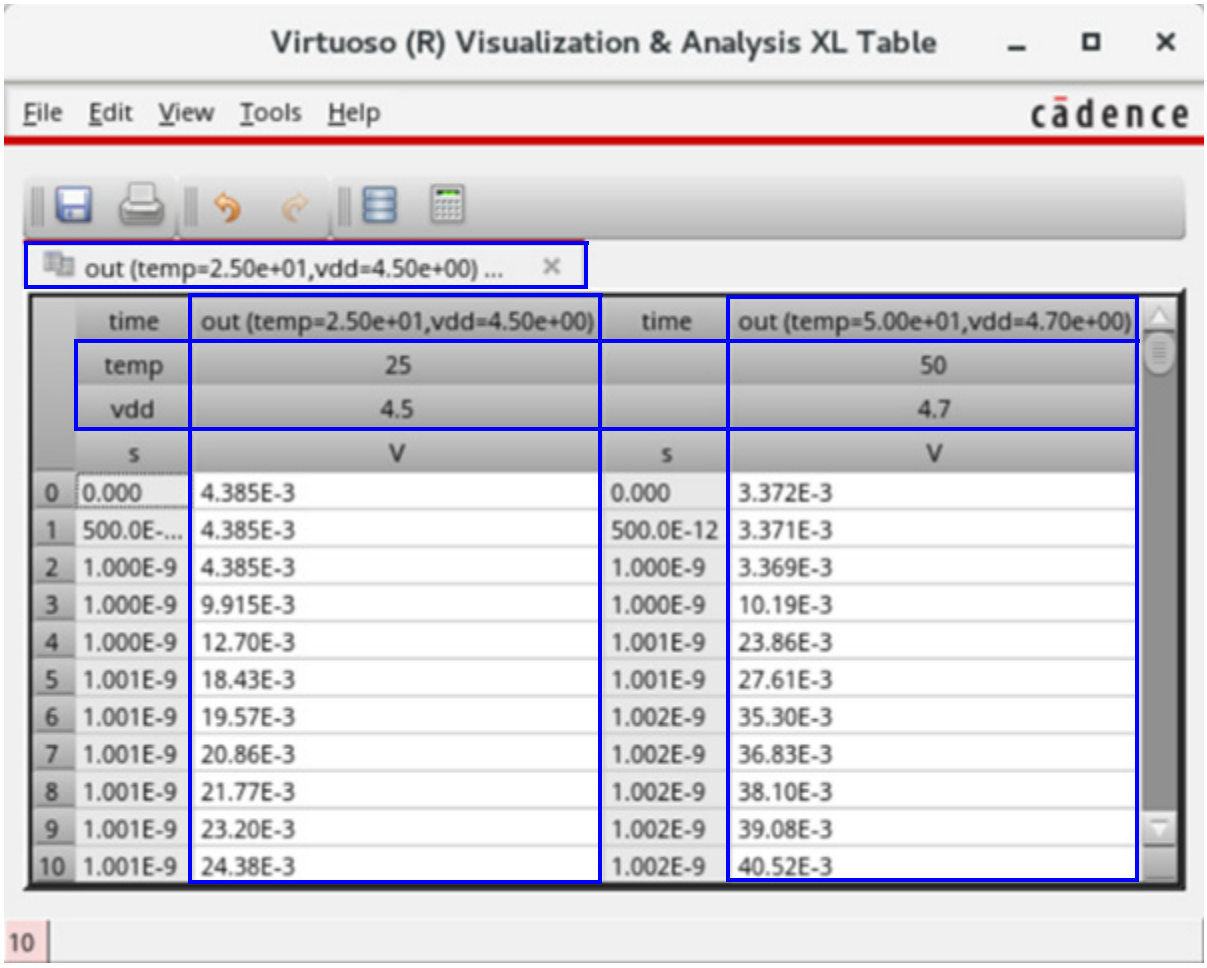
To open the Table from the graph window:

1. Select a trace in the trace legend area of the graph window.
2. Right-click a trace and point to *Send To – Table* and then choose one of the following options:
   - *New Window*: Displays waveform data of the selected trace in a new Tab of the Table.
   - *Append*: Appends waveform data of the selected trace in the active tab of the Table.
   - *Replace*: Replaces existing waveform data with that of the selected trace in the active tab of the table.

The trace containing parametric sweep data displays waveform data for each sweep variable value in separate columns of the Table. The sweep variables are displayed as headers.

© 2023 Cadence Design Systems, Inc.

The following figure shows waveform data for the trace `out` that contains parametric sweep data for sweep variables `temp` and `vdd`. These sweep variables are also displayed as headers.



Opening Table from the Calculator

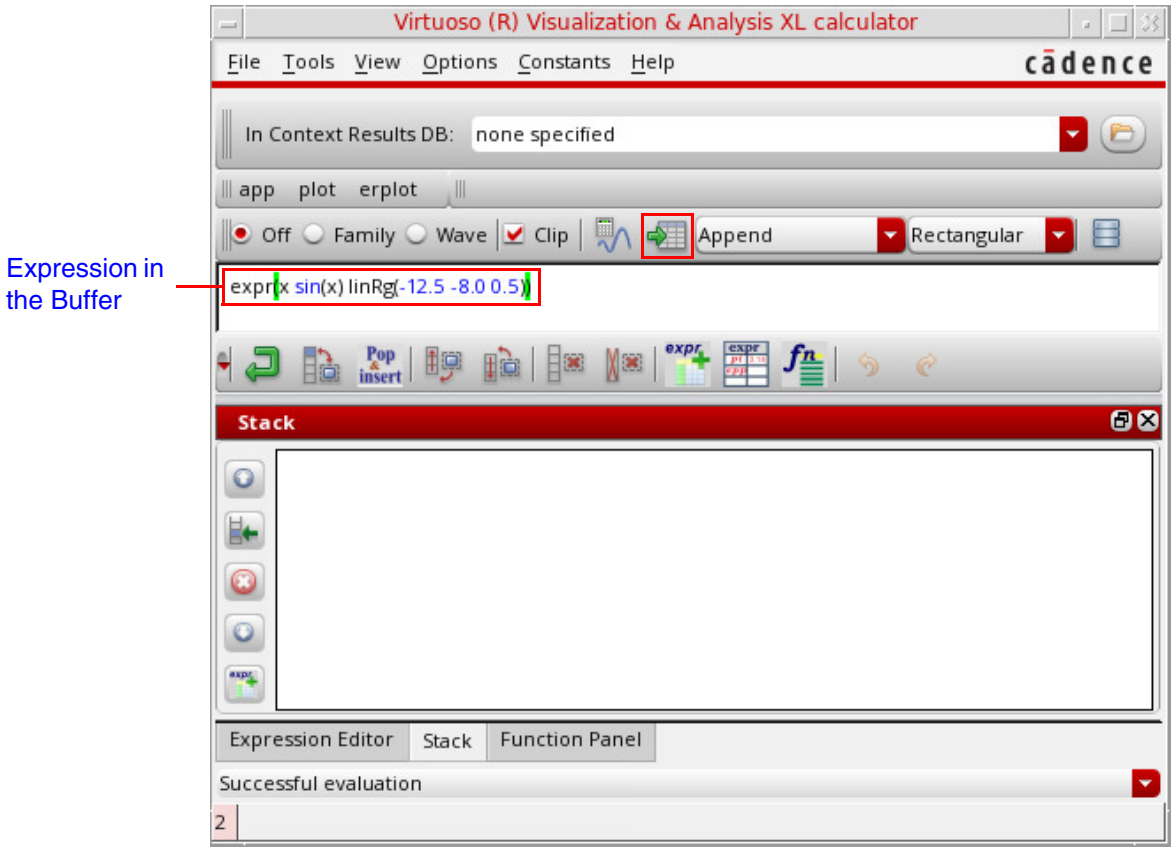
You can send the following from Calculator to the Table:

- An expression from the Calculator Buffer.
- An output signal obtained from simulation results.

To send an expression to the Table from the Calculator:

© 2023 Cadence Design Systems, Inc.

1. Type an expression in the Calculator Buffer.

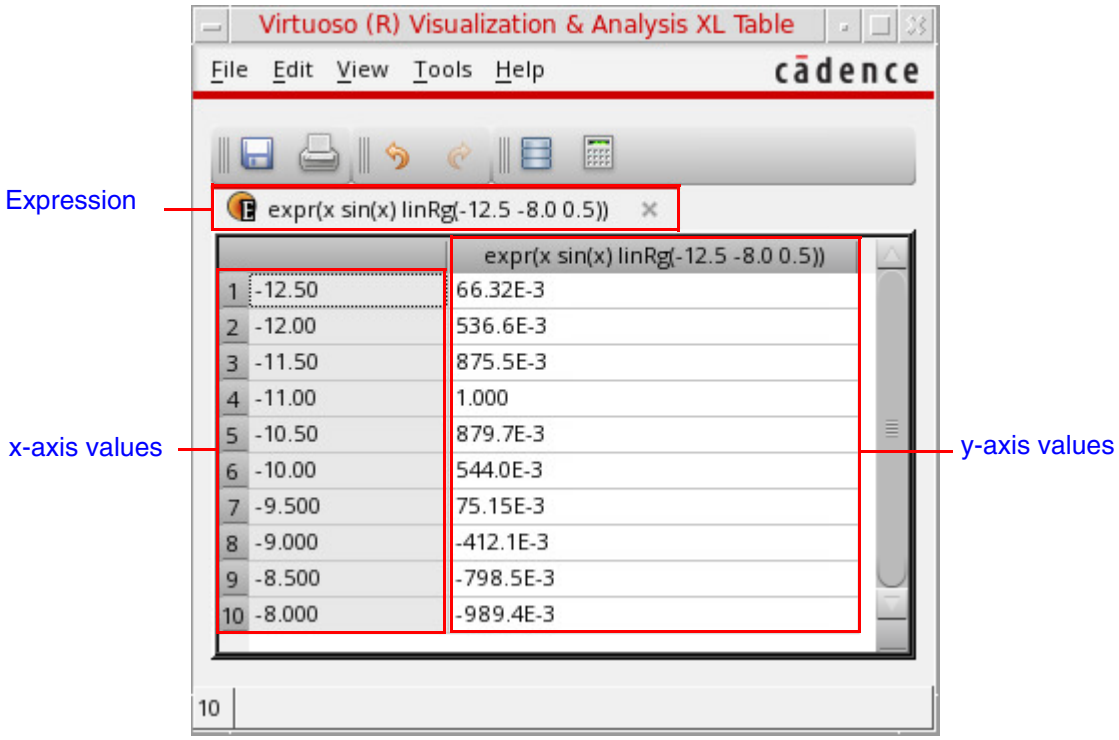


2. From the menu bar of the Calculator, choose *Tools – Table*.

   Alternatively, click the icon in the *Selection* toolbar of the Calculator.

© 2023 Cadence Design Systems, Inc.

The Virtuoso Visualization and Analysis Calculator opens, displaying expression data.

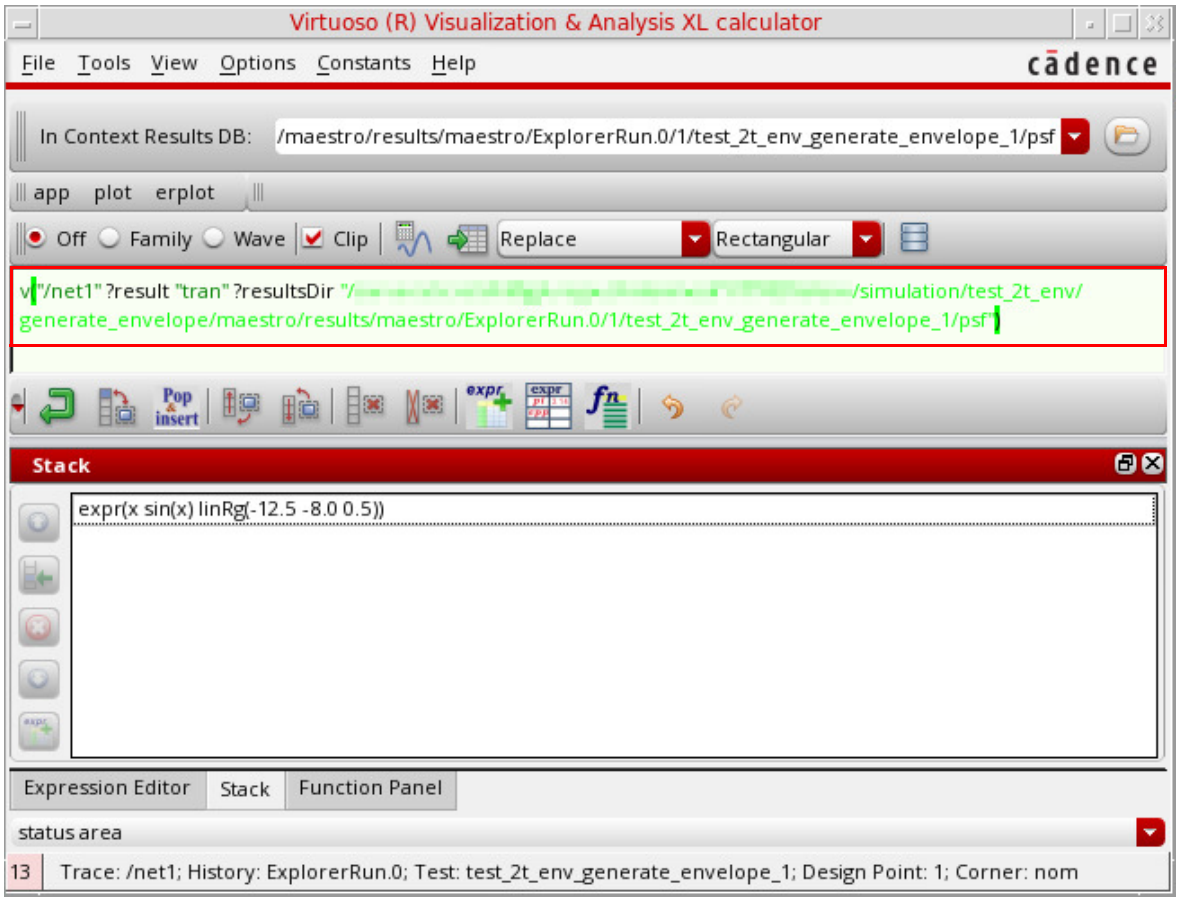


To send a signal from Calculator to the Table:

1. Send the output signal to the Calculator from the Results Browser or from the graph window.

© 2023 Cadence Design Systems, Inc.

The signal is displayed in the Buffer of the Calculator.

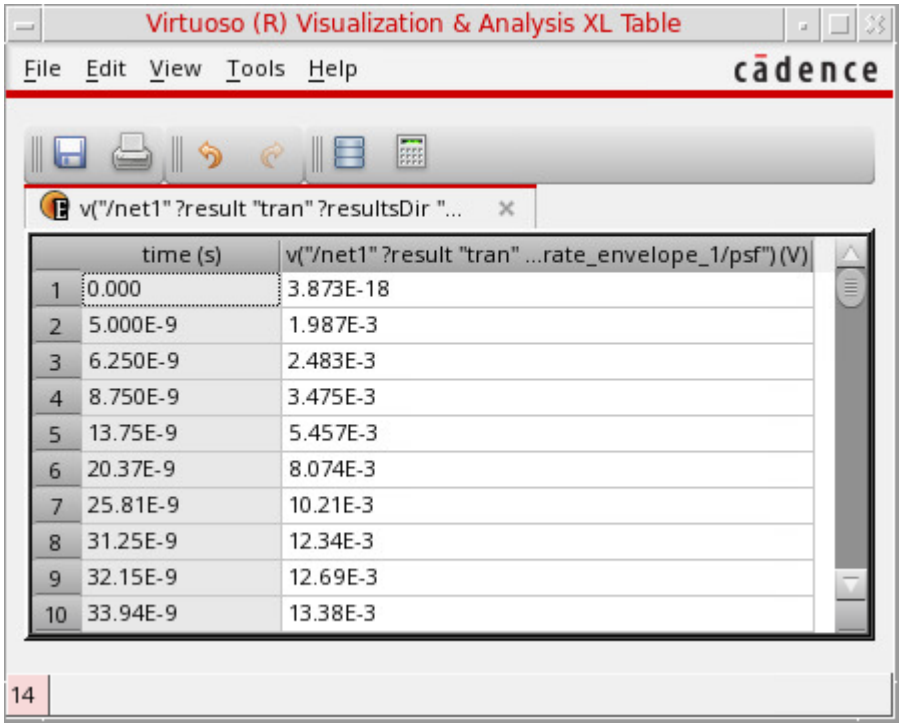


2. From the menu bar of the Calculator, choose *Tools – Table*.

   Alternatively, click the icon in the *Selection* toolbar of the Calculator.

© 2023 Cadence Design Systems, Inc.

The Virtuoso Visualization and Analysis Calculator opens, displaying waveform data of the signal.



Related Topics

Virtuoso Visualization and Analysis XL Table

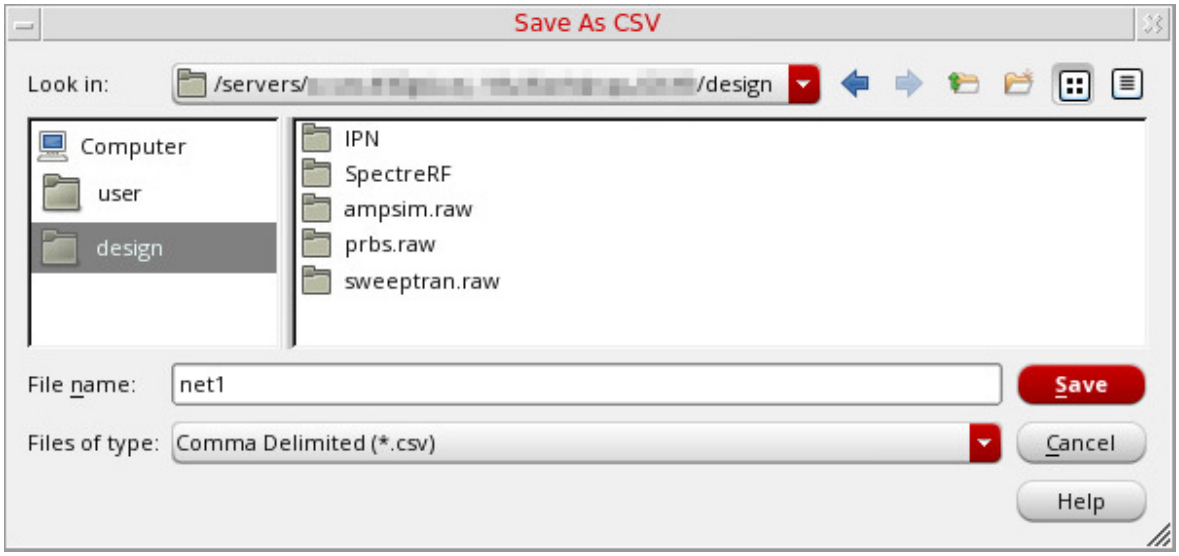
Results Browser

© 2023 Cadence Design Systems, Inc.

Saving Table Contents to a CSV File

To save the contents of the Table to a CSV file:

1. From the *File* toolbar, click *Save as CSV* .
2. The Save as CSV form opens.



3. In the *Look in* field, select the directory where you want to save the CSV file.
4. In the *File name* field, enter the name of the CSV file to which you want to save the contents of the Table.
5. Click *Save*.

   The contents of the Table are saved in the specified CSV file.

Related Topics

The Save As CSV Form

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

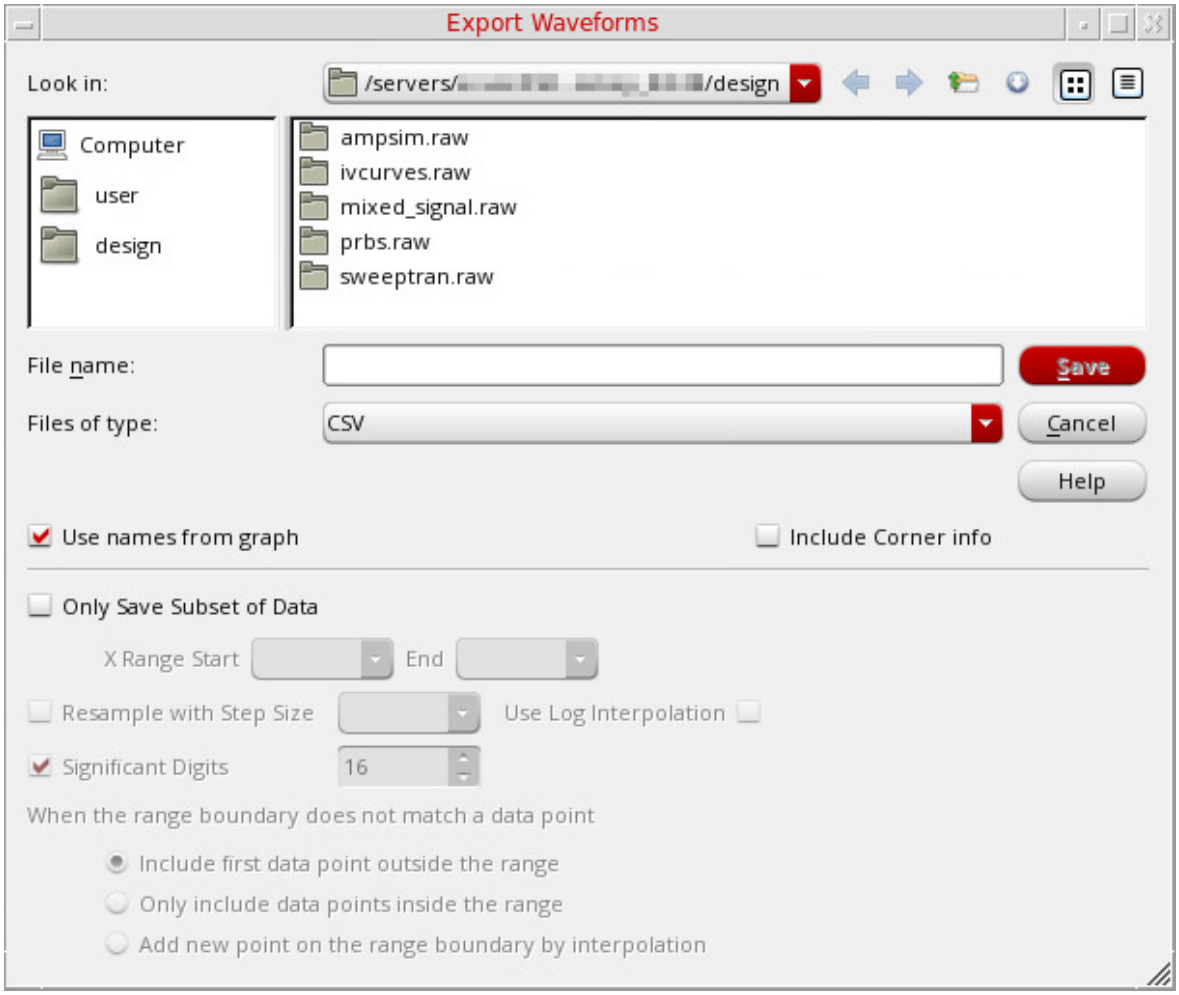
Exporting Contents of the Table

You can export the contents of the Virtuoso Visualization and Analysis XL Table in various formats that you can use later.

To export the contents of the Table:

1. From the menu bar of the Table, choose *File – Export*.

   The Export Waveforms form opens.



2. In the *File name* field, type the name of the file to which you want to export the selected waveforms.
3. From the *Files of type* drop-down list, select one of the following formats: *VCSV*, *Matlab*, *PSF*, *SPECTRE*, *SST2*, *CSV*, or *PSFXL*.

© 2023 Cadence Design Systems, Inc.

4. Select the *Use names from grap*h check box to use the trace names that are displayed on the graph.
5. If you are exporting signals in the CSV format, select the *Include Corner info* check box to print the corner information in the header of the CSV file.

   The *Include Corner info* check box appears dimmed and becomes available only if *CSV* is selected from the *Files of type* drop-down list.
6. Select the *Only Save Subset of Data* check box to export only the required part of the waveform or signal.
7. In the *X Range Start* and *End* fields, specify the start and the end points of the x range for which you want to export the waveform.
8. Select the *Resample with Step Size* check box to specify the step size to resample the exported data based on linear interpolation. The greater the step size, the fewer data points in the exported data.
9. Select the *Use Log Interpolation* check box to use the logarithmic scale for interpolation.
10. In the *Significant Digits* field, specify the number of significant digits in the exported values from waveform data.
11. From the *When the range boundary does not match a data point* options, select one of the following:

    - *Include first data point outside the range*: Includes the first data point that immediately falls before the specified *X Range Start* value and the last data point that falls immediately after the specified *X Range End* value when there are no actual data points corresponding to the specified *X Range Start* and *End* values
    - *Only include data points inside the range*: Includes only actual data points inside the specified x range.
    - *Add new point on the range boundary by interpolation*: Includes the interpolated values for the specified range boundary (*X Range Start* and *End* values) when there are no actual data points corresponding to the specified *X Range Start* and *End* values.
12. Click *Save*.

Waveform data is exported to a file in the specified format.

© 2023 Cadence Design Systems, Inc.

Related Topics

The Export Waveforms Form

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

© 2023 Cadence Design Systems, Inc.

Hiding Columns in the Table

To hide a column in the Virtuoso Visualization and Analysis XL Table:

1. Click a column to select it. Hold down the `Ctrl` key to select multiple columns.
2. From the menu bar of the Table, choose *View – Hide*.

   Alternatively, right-click the selected columns and choose *Hide Column*.

   The selected columns are hidden from the Table.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

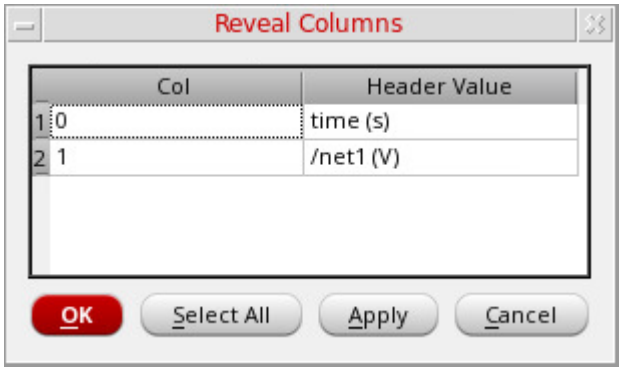
Revealing Columns in the Table

To reveal a hidden column in the Virtuoso Visualization and Analysis XL Table:

1. From the menu bar of the Table, choose *View – Reveal*.

   Alternatively, right-click anywhere inside the Table and choose *Reveal Column*.

   The Reveal Columns form opens, displaying numbers and names of the columns that are currently hidden in the Table.



   The *Reveal* and *Reveal Column* commands are not available if there are no hidden columns in the Table.

© 2023 Cadence Design Systems, Inc.

2. In the Reveal Columns form, click the row for the column that you want to reveal. Hold down the `Ctrl` key to select multiple rows.

   If you want to reveal all hidden columns, click *Select All*.

3. Click *Apply* or click *OK*.

   The selected hidden columns are revealed again in the Table.

Related Topics

The Reveal Columns Form

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

© 2023 Cadence Design Systems, Inc.

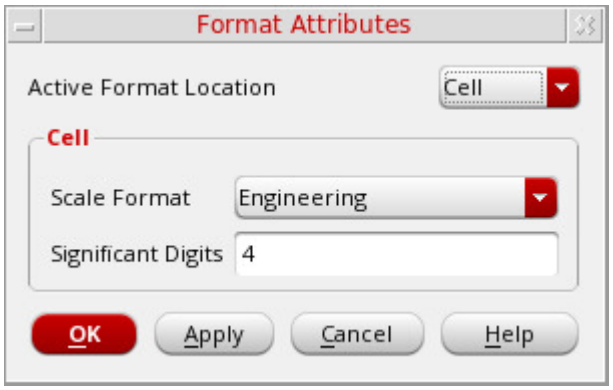
Formatting Attributes of the Table

To format attributes of cells and headers of the Virtuoso Visualization and Analysis XL Table:

1. Click a column to select it. Hold down the `Ctrl` key to select multiple columns.
2. From the menu bar of the Table, choose *View – Format*.

   Alternatively, right-click the selected columns and choose *Format*.

   The Format Attributes form opens.



3. From the *Active Format Location* drop-down list, select one of the following options:
   - *Cell*: To change the format attributes only for the cells of the Table.
   - *Header*: To change the format attributes for both cells and headers of the Table.
4. From the *Scale Format* drop-down list, select one of the following options:
   - *Scientific*: To change the notation of values to scientific. For example: `1.234E-8`.

     This notation is not available if *Header* is selected in the *Active Format Location* drop-down list.
   - *Engineering*: To change the notation of values to engineering. For example: `12.34E-9`.
   - *Suffix*: To change the notation of values to suffix. For example: `12.34n`.
5. From the *Scale Factor* drop-down list, select the units to which you want to convert values in the Table. The units displayed in the *Scale Factor* drop-down list depends on the selected scale format.

   The *Scale Factor* field is available only when Header is selected in the *Active Format Location* drop-down list.

© 2023 Cadence Design Systems, Inc.

6. In the *Significant Digits* field, specify the number of significant digits to be displayed for data.
7. Click *Apply* or click *OK*.

Related Topics

The Format Attributes Form

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

Sorting Columns in the Table

When you sort a column the first time, values in the column are sorted in an ascending order. If you sort the same column again, values in the column are sorted in the descending order.

To sort a column in the Table:

1. Click to select the column in which you want to sort the values.
2. From the menu bar of the Table, choose *View – Sort*.

   Alternatively, right-click the column you want to sort and choose *Sort*.

   The values in the columns are sorted either in the ascending or descending order.

When a column is sorted, an arrow key appears at the column header. The direction of the arrow key indicates whether the values are currently sorted in the ascending or descending order. The key indicates that values in the column are sorted in the ascending order. The key indicates that values in the column are sorted in the descending order.

You can click this arrow key to change the sorting order of the column.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

© 2023 Cadence Design Systems, Inc.

Transposing Rows and Columns of the Table

To transpose rows into columns and column into rows:

➡ From the menu bar of the Table, choose *View – Transpose*.

Alternatively, right-click anywhere inside the Table and choose *Transpose*.

The rows and columns of the Table are interchanged.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

Merging X-Axis Data in a Single Column

You can merge two or more columns that contain similar x-axis data in a single column.

The merged column includes x-axis values from all columns being merged. The corresponding y-axis values are either interpolated or extrapolated.

To merge columns that contain similar x-axis data:

1. Hold down the `Ctrl` key and click the columns that you want to merge.
2. Right-click the Selected columns and choose one of the following options:
   - *Merge X*: To merge only the selected columns that contain similar x-axis data.
   - *Merge All X*: To merge all columns that contain similar x-axis data.

   The columns that contain similar x-axis data are merged into a single column.

Note the following points:

- If you try to merge columns that do not contain similar x-axis data, for example one column contains time values and another column contains frequency values, a dialog box appears indicating the error.
- The *Merge X* and *Merge All X* options are available only when there are more than one column that contain similar x-axis data.

© 2023 Cadence Design Systems, Inc.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Changing Column Colors in the Table

To change the color of a column in the table:

1. Select the column whose color you want to change. Hold down the `Ctrl` key to select multiple columns in the Table.
2. Right-click the selected columns and point to *Change Column Color* and then choose the color you want to set to the columns.

   The selected color is applied to the columns.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Renaming Column and Row Headers of the Table

To rename the header of a column or row of the Table:

1. Click a column or row whose header name you want to change.
2. Right-click the selected column or row and choose *Rename Header*.

   The Rename Header form opens.
3. Enter the new name of the header of the column or row.
4. Click *Apply* and click *OK*.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

Limiting the Number of Actions to Undo or Redo

You can specify the number of actions that can be undone or redone in the Virtuoso Visualization and Analysis XL Table:

To specify the number of actions that can be undone or redone:

1. From the menu bar of the Table, choose *Edit – Preferences*.

   The Preferences for opens.

2. In the *Limit Undo Actions* field, specify the number of actions that can be undone and redone. The valid values are from `1–10`.

   For example, if you set `3` in this field, you can undo up to last `3` actions performed in the Table.

3. Click *OK*.

Related Topics

The Preferences Form

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

© 2023 Cadence Design Systems, Inc.

Filtering Table Data

You can apply filters to columns containing x-axis data to selectively view data that match the filter pattern.

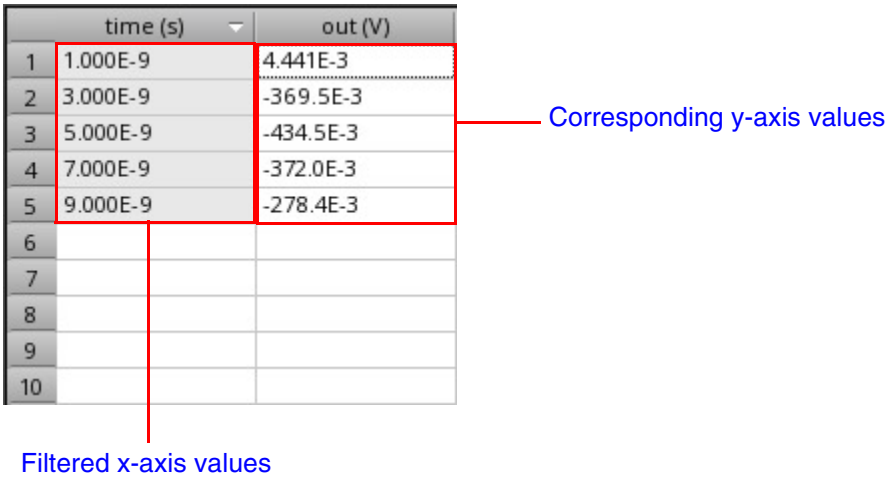
To filter data in the Table:

1. Select a column that contains x-axis data.
2. Right-click the selected column and choose one of the following options:
   - *Apply Range*: In the Apply Range form, specify the start and end values of the range within which you want to view x-axis and corresponding y-axis data.
   - *Value At*: In the Value At form, specify the x-axis value corresponding to which you want to view the y-axis value. If the specified x-axis value does not match a data point, the corresponding y-axis value is obtained using either interpolation or extrapolation.
   - *Sample Values*: Filters data that match the sample values specified in the Sample Values form. To specify the sample values:
     - In the *Start* field, specify the starting value of the sample. For example: `1e-9`.
     - In the *End* field, specify the end value of the sample. For example, `9e-9`.
     - In the *Step* field, specify the step value. For example, `2e-9`.
     - Select the *Log* check box to include logarithmic values.

       Do not specify the step size in the *Step* field when the *Log* check box is selected.
     - Click *OK*.

© 2023 Cadence Design Systems, Inc.

The x-axis data and the corresponding y-axis data is filtered according to the specified sample values:



Related Topics

Apply Range Form

The Value At Form

The Sample Values Form

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Changing the Format of Complex Data in the Table

To change the format of complex data displayed in the Table:

1. Select the column that contain complex data. Hold down the `Ctrl` key to select multiple columns.
2. Right-click the selected columns and point to *Display Complex As*, and choose one of the following options:
   - *Imaginary*: Displays the imaginary part of the complex numbers.
   - *Real*: Displays the real part of the complex numbers.

© 2023 Cadence Design Systems, Inc.

- *Real and Imaginary*: Displays both the real and the imaginary parts of the complex numbers.
- *Magnitude*: Displays the magnitude of the complex numbers.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

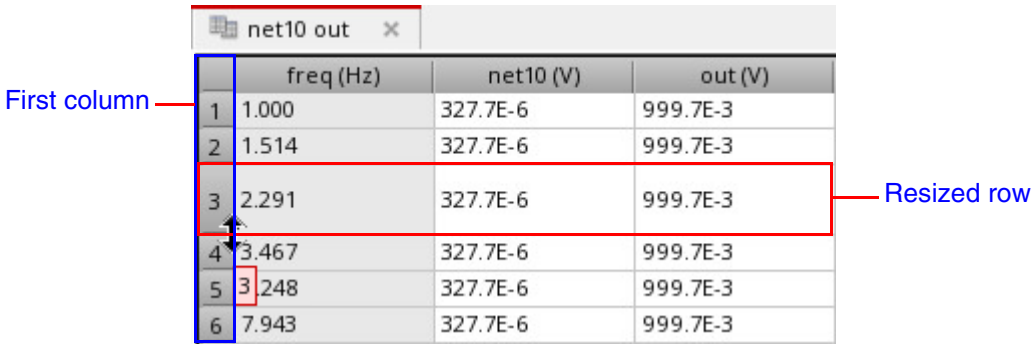
Resizing Columns and Rows of the Table

To resize a column in the Virtuoso Visualization and Analysis XL Table:

- Drag the right edge of a column header to the left or to the right to make the column narrower or wider, respectively.

To resize a row in the Virtuoso Visualization and Analysis XL Table:

- Drag the bottom edge of a row in the first column to downward or upward to make the row narrower or wider, respectively.



To resize column widths to fit the contents:

- Right-click anywhere inside the Table and choose *Resize Columns to Contents*.

To reset the column width and row height to their default values:

- Right-click anywhere inside the table and choose *Reset To Default Cell Size*.

Related Topics

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

Printing Contents of the Table

To print the contents of the Virtuoso Visualization and Analysis XL Table:

1. From the menu bar of the Table, choose *File – Print*.

   Alternatively, from the *File* toolbar, click *Print* .

   The Print form opens, displaying the name, location, and type of the default printer.

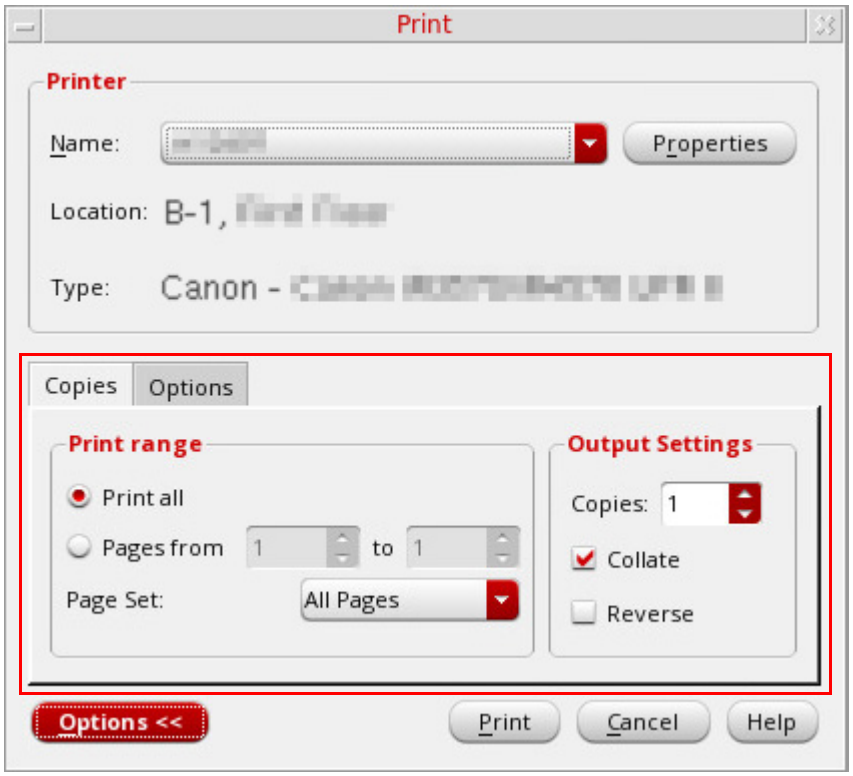
2. In the *Printer* group box, do the following:

   a. From the *Name* drop-down list, select the printer you want to use.

   b. Click *Properties* to configure page properties and job options.

© 2023 Cadence Design Systems, Inc.

3. Click *Options* to display additional options in the form.



4. In the *Copies* tab, specify the following options:
   - In the *Print range* group box, specify the following options:
     - Click *Print* all to print all the table.
     - Click *Pages* from to specify the range of pages you want to print.
     - From the *Page Set* drop-down list, select *All Pages*, *Odd Pages*, or *Even Pages*.
   - In the *Output Settings* group box, specify the following options:
     - In the *Copies* field, specify the number of copies you want to print.
     - When printing multiple copies, select the *Collate* check box to print all the pages of a single copy together.
     - Click the *Reverse* check box to start printing from the last page.
5. Click the *Options* tab and specify the following options:
   - From the *Duplex Printing* options, select one of the following options:

© 2023 Cadence Design Systems, Inc.

- *None*: To print data only on one side of one sheet of paper.
- *Long side*: To print data on both sides of one sheet of paper in the portrait mode.
- *Short side*: To print data on both sides of one sheet of paper in the landscape mode.

- From the *Color Mode* options, select either *Color* or *Grayscale*.

6. Click *Print* to print the contents of the Table.

Related Topics

Setting Printer Properties

The Print Form

The Printer Properties Form

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

Toolbar of the Table

Setting Printer Properties

To set the properties of the printer:

1. From the menu bar of the Table, choose *File – Print*.

   Alternatively, from the *File* toolbar, click *Print* .

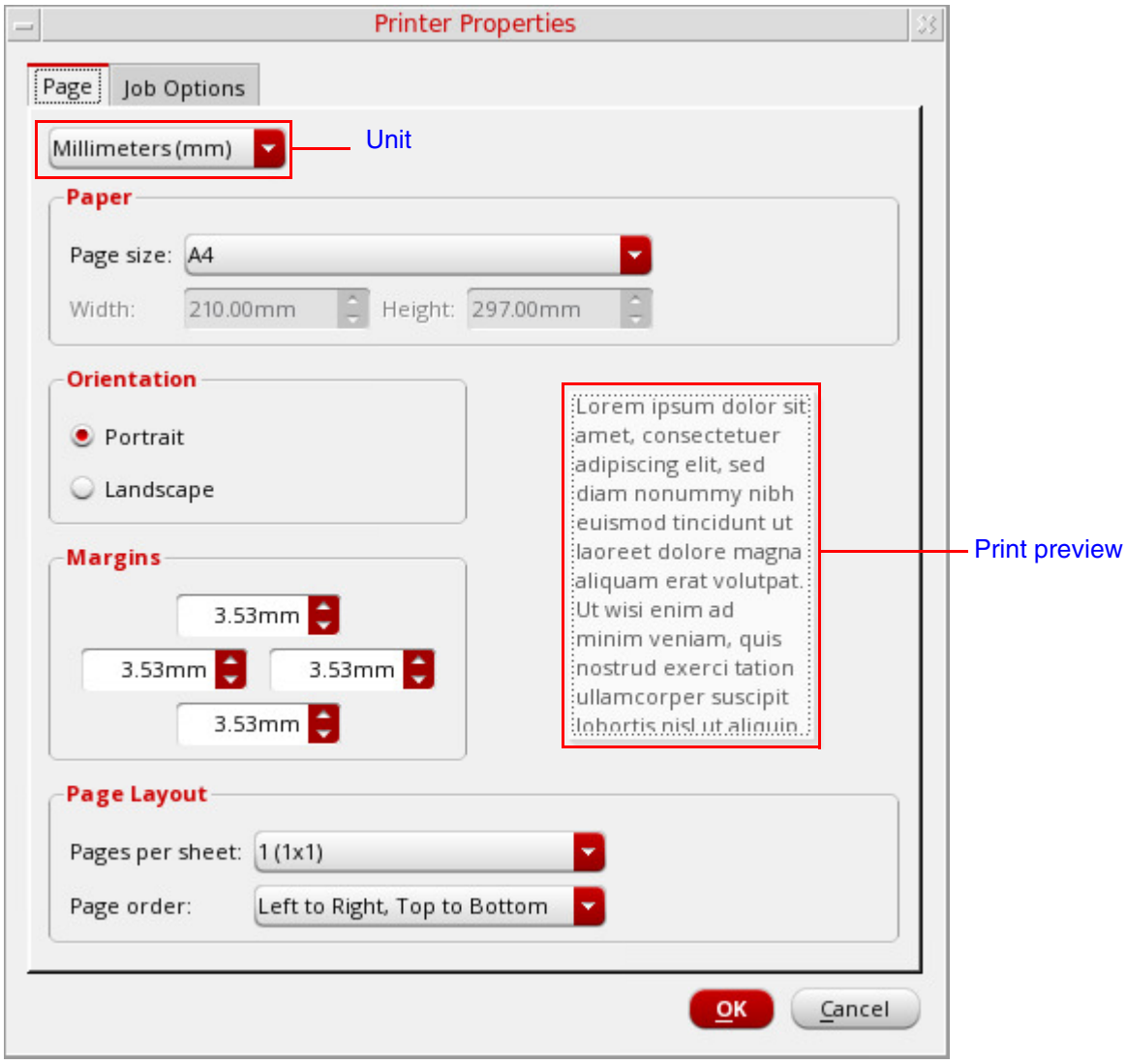
© 2023 Cadence Design Systems, Inc.

The Print form opens, displaying the name, location, and type of the default printer.

2. In the *Printer* group box, click *Properties* to configure page properties and job options.

© 2023 Cadence Design Systems, Inc.

The Printer Properties form opens.



3. On the *Page* tab, specify the following options:
   a. From the *Unit* drop-down list, select the unit of measurement for page settings.
   b. In the *Paper* group box, select the size of the paper from the *Page size* drop-down list.
   c. From the *Orientation* options, select either *Portrait* or *Landscape*.
   d. In the *Margins* group box, select the top, left, right, and bottom margins.
   e. In the *Page Layout* group box, select how many pages you want to print per sheet from the *Pages per sheet* drop-down list. You can also control the order in which

© 2023 Cadence Design Systems, Inc.

pages are printed from the *Page order* drop-down list. By default, pages are printed from left to right and then from top to bottom.

4. Click the *Job Options* tab and specify the following options:

   f. In the *Job Control* group box, from the *Scheduled printing* drop-down list, choose when do you want to schedule the printing job. You can also specify the billing information and the job priority in the respective fields.

   g. In the *Banner Pages* group box, from the *Start* and *End* drop-down lists, select the level of the information sensitivity that you want to print at the start and the end of the banner page, respectively.

5. Click *OK* to save the printer properties.

Related Topics

Printing Contents of the Table

The Print Form

The Printer Properties Form

Virtuoso Visualization and Analysis XL Table

Opening the Virtuoso Visualization and Analysis XL Table

Menu Bar of the Table

Toolbar of the Table

© 2023 Cadence Design Systems, Inc.

4

Working with Graphs

The Virtuoso Visualization and Analysis XL graph window is a tool that you use to present the simulation data in a graphical format.

This chapter includes the following topics:

- About the Graph Window on page 135
- Graphical User Interface on page 138
- Creating a Graph on page 163
- Working with Subwindows on page 165
- Customizing a Graph on page 189
- Working with Graph Axis on page 202
- Working With Assistants on page 225
- Working with Workspaces on page 365
- Working with Traces on page 368
- Working with Strips on page 398
- Working with Sweeps on page 411
- Working with Graph Labels on page 422
- Plotting WREAL Signals on page 424
- Plotting YvsY Graph on page 428
- Saving and Loading Graphs on page 430
- Reloading Graphs on page 440
- Printing Graphs on page 449
- Supporting Mixed Signals on page 456

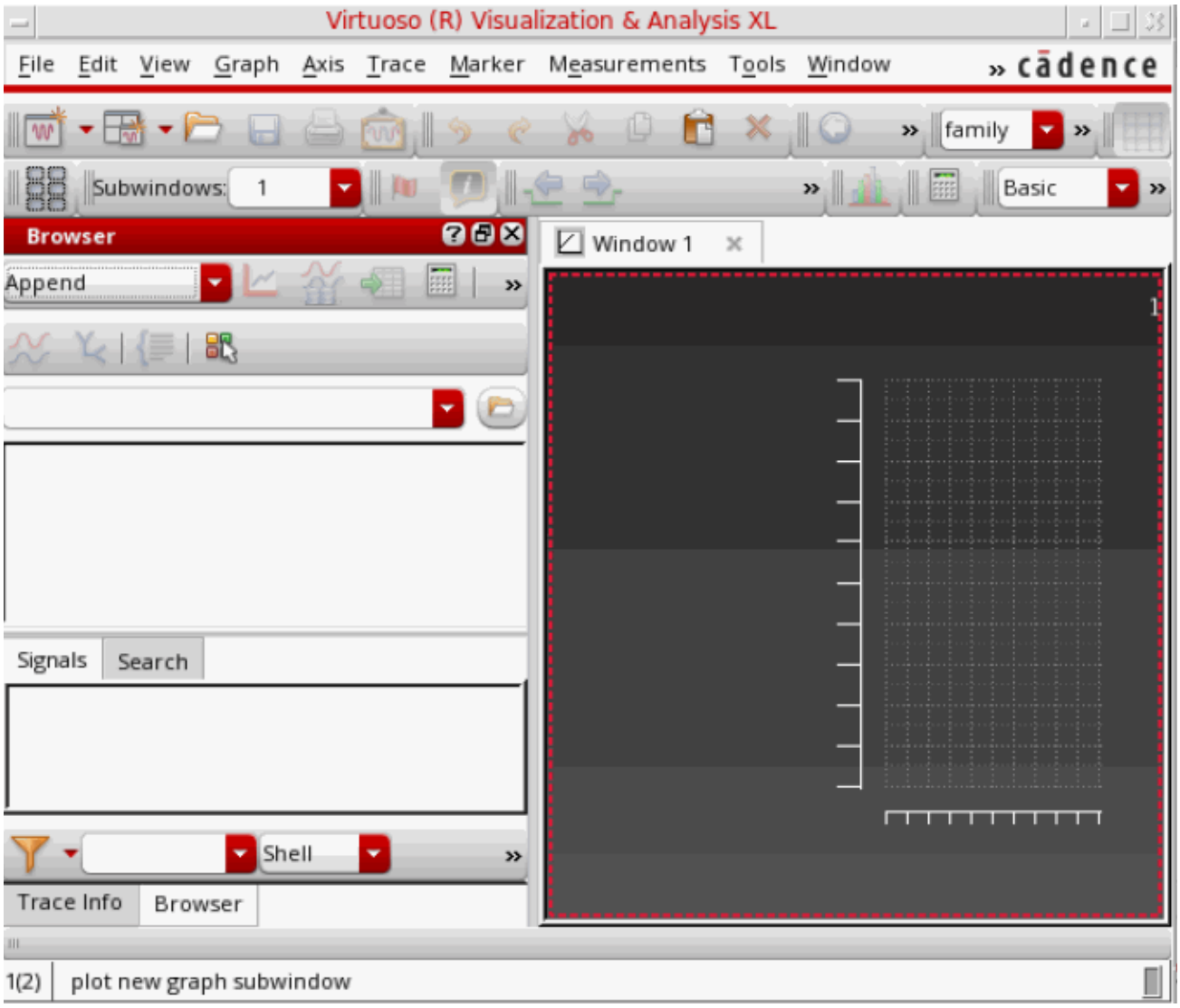
© 2023 Cadence Design Systems, Inc.

- Working with Buses on page 477
- Working with Markers on page 493
- Working with Circular Graphs on page 539
- Setting Bindkeys on page 565
- Working with Tabular Graph on page 569
- Graph Summary Label on page 577
- Annotating Scalar Outputs for Single-Point Simulation on page 585

© 2023 Cadence Design Systems, Inc.

About the Graph Window

Virtuoso Visualization and Analysis XL graph window is a tool that you use to present simulation data in a graphical format. This helps you analyze simulation results. The ability to plot multiple graphs at a time enables you to compare simulation results. You can also customize your graphs by changing the background color and layout, and add markers and labels to annotate the graphs.



When you run the tool for the first time in a new session, a default graph is opened in a tab named *Window 1*. You can rename the window tab names by double-clicking the tab name or by setting the `viva.graphFrame` .cdsenv variable. You can also close the tabs that are not required. In a window, you can open multiple subwindows.

© 2023 Cadence Design Systems, Inc.

The graph window terminology is explained below:

- Window—Window is the plotting area where you plot signals and open multiple subwindows. A window consists of a trace legend area, X-axis zoom and pan bar, dependent and independent axes, and graph objects. The window names appear on different window tabs, which means when you open a new window, it is opened in a new tab. You can also close and rename the windows if required.
- Subwindow or Graphs—A window can be divided into multiple subwindows, which include all properties of a window, such as trace legend area, pan bar, dependent and independent axes, and graph objects. The subwindow names are displayed in the Subwindows drop-down list on the Graph toolbar. By default, a new graph window always has one subwindow defined. For more information about subwindows, see Working with Subwindows on page 165.
- Strip—A subwindow can be further divided into different strips that include one or more signals. All the strips in a subwindow share the same independent axis, and X-axis zoom and pan bar.
- Traces, markers, labels, and axes are the graph objects that can be inserted in a window, subwindow, or a strip.

In the graph window tool, you can also choose the assistants you want to display, which is defined by the selected workspace. The default workspace is *Classic*. For more information about Workspaces, see Working with Workspaces on page 365.

You can open the graph window from Virtuoso Visualization and Analysis XL or from the Analog Design Environment (ADE). If you open the graph Window from the Virtuoso Visualization and Analysis XL, you can work with previously saved simulation data. However, if you open the graph window from ADE, you work with the simulation data for the latest run. In both cases, when you select a signal, the graph window appears with the selected signal plotted.

Opening the Graph Window

You can use the following methods to open the graph window:

- Opening the Graph Window from Virtuoso Visualization and Analysis XL on page 137
- Opening the Graph Window from Virtuoso in Stand-Alone Mode on page 137
- Opening the Graph Window from ADE on page 137

© 2023 Cadence Design Systems, Inc.

Opening the Graph Window from Virtuoso Visualization and Analysis XL

You can open the graph window in either SKILL mode from the Virtuoso Visualization and Analysis XL tool. By default, the graph window is opened in the SKILL mode.

Perform the following steps to open the graph window from Virtuoso Visualization and Analysis XL tool:

➡ Start the Virtuoso Visualization and Analysis XL tool by typing the following command in a terminal window:

```
viva -expr skill &
```

The *Virtuoso (R) Visualization and Analysis XL* appears.

Note: If you type only `viva &` in the terminal window, the default mode is SKILL.

For information about how to create a graph in the graph window, see Creating a Graph on page 163.

Opening the Graph Window from Virtuoso in Stand-Alone Mode

To open the graph window from Virtuoso in the stand-alone mode, perform the following step:

■ From the CIW, choose *Tools – Analog Environment – Waveform.*

The *Virtuoso (R) Visualization and Analysis XL* appears.

Opening the Graph Window from ADE

You can run simulations in ADE and plot the simulation results in the Virtuoso Visualization and Analysis XL graph window. The Virtuoso Visualization and Analysis XL graph window supports simulation analysis types such as transient, AC, DC, and RF measurement.

To open the graph from ADE, do the following:

➡ In the ADE window, choose *Tools – Waveform*.

The *Virtuoso Visualization and Analysis XL* appears.

The graph window also appears after a simulation is run in ADE Explorer, displaying the output signals from the selected analysis types.

In ADE Assembler, after you run the simulation, you can specify whether you want to save the simulation results to a results database or plot the simulation results in a window. Each item that appears on the *Outputs Setup* tab has a *Plot* check box and a *Save* check box. Select

© 2023 Cadence Design Systems, Inc.

the *Plot* check box to display the selected outputs in the window after the simulation run is complete. Select the *Save* check box to save the selected output results to a results database.

Notice the following when you open the window by using ADE:

- The graph window tab names in Virtuoso Visualization and Analysis XL correspond to the test names in ADE.
- The subwindow titles display the measurement or analysis names.
- The subwindow titles also display the simulation time or measurement evaluation time.

Graphical User Interface

The Virtuoso Visualization and Analysis XL window user interface consists of a menu bar, toolbars, dockable assistants, and subwindows that are displayed according to the selected workspace. You can hide and show these GUI components based on the workspace you select. By default, the assistant panes appear on the left and the graphs appear in the display area on the right. When you plot a signal in the new window, a new window tab is created. You can click the tab to view the required graph and can rename window tabs by double-clicking the tab name. You can also close the window tabs that are not required.

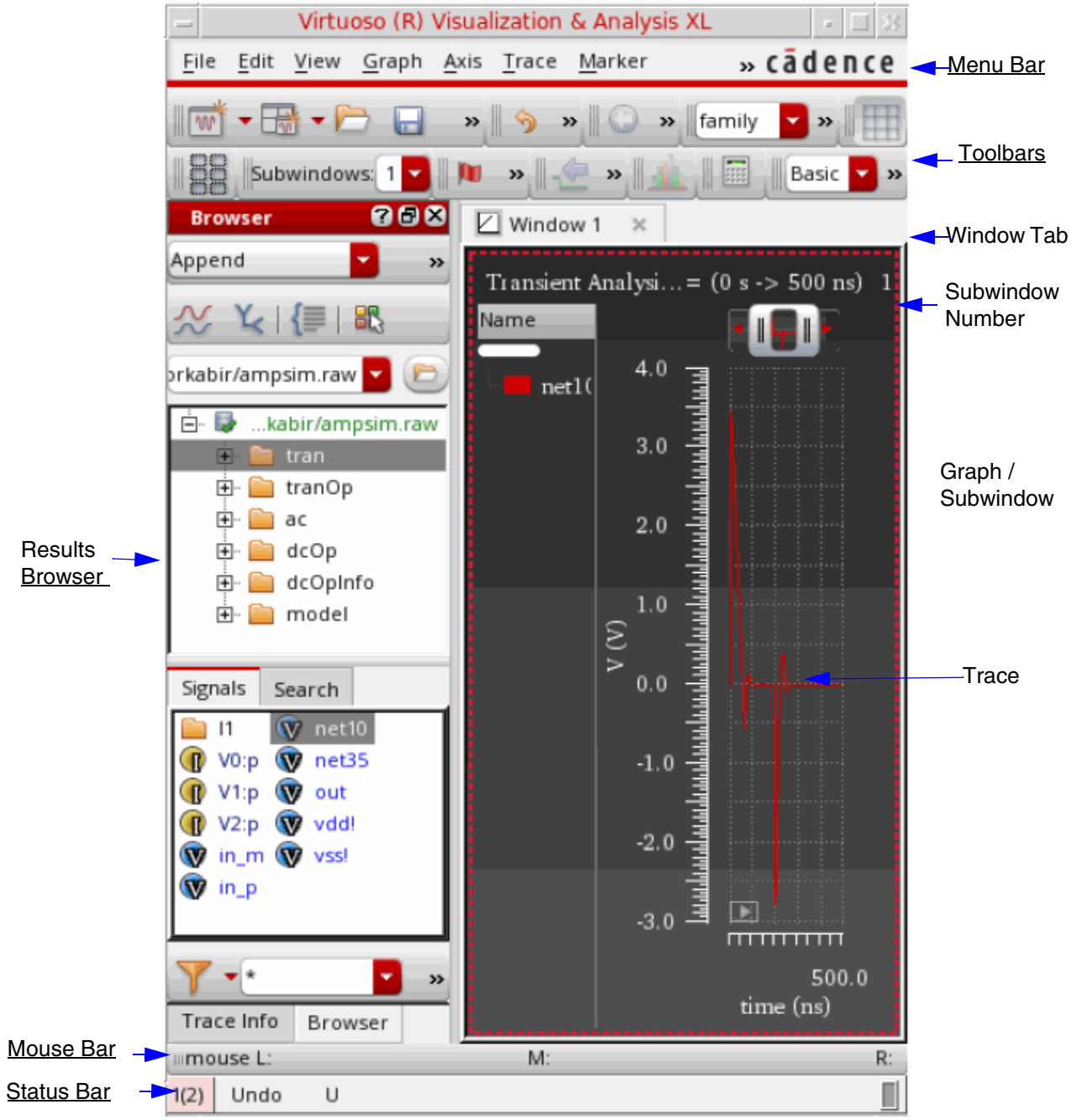
In a window, you can open multiple subwindows. The subwindows includes all properties of a graph window and can be further divided into subwindows.

Note: By default, the window background is black. If required, you can set the background to white. For more information, see Setting the Graph Colors on page 190.

The Virtuoso Visualization and Analysis XL window includes the following elements:

- Menu Bar on page 139
- Toolbars on page 157
- Status Bar on page 163
- Assistants on page 163

© 2023 Cadence Design Systems, Inc.



Menu Bar

The menu bar has the following menus:

- File on page 140

© 2023 Cadence Design Systems, Inc.

- Edit on page 142
- View on page 143
- Graph on page 144
- Axis on page 145
- Trace on page 146
- Marker on page 148
- Measurement on page 149
- Tools on page 151
- Window on page 151
- Browser on page 152
- Help on page 154

File

The table below lists the *File* menu commands.

Command	Description
Open Results	Opens the *Select Waveform Database* form that you can use to select a results database. This form displays the current results directory.
Close Results	Closes the selected results directory in the Results Browser.
New Window	Opens a new graph window in the display area. This creates a new window tab.
New Subwindow	Opens a new subwindow in the active window. For more information, see Working with Subwindows on page 165.
Load Window	Loads a file containing a graph in the active window. For more information, see Loading a Graph on page 433.
Save Window	Saves the graph in the active window to a file. For more information, see Saving a Graph on page 431.

© 2023 Cadence Design Systems, Inc.

Command	Description
Save Window As	Saves a copy of the graph in the active window to a file. For more information, see Saving a Graph on page 431.
Window Properties	Changes the background color of all the subwindows in the active window.
Reload	Updates data for all the traces in the active subwindow or all the subwindows. For more information, see Reloading Graphs on page 440.
Create Maestro Plotting Template	Saves a graph window as a plotting template after the simulation is run and results are plotted. The saved plotting template can be used to plot waveform outputs in the specified format for next simulation runs. All the formatting, markers, layout changes, and other interface changes will be retained as specified in the plotting template. For more information, see Working with Plotting Templates in *Virtuoso ADE Explorer User Guide*. Video The video ViVA XL - Plotting Templates demonstrates how you can create plotting templates.
Print	Sends the graph to a printer or saves the graph in a PDF format. For more information, see Printing Graphs on page 449.
Save Image	Saves the graph displayed in the active window as an image file. For more information, see Printing Graphs on page 449.
Close Window	Closes the active window. If there is only one open window, the *Close* command exits the tool.
Close All Windows	Closes all windows and exits the tool.

© 2023 Cadence Design Systems, Inc.

Edit

The table below lists the *Edit* menu commands.

Command	Description
Undo	Undoes the most recent action in the active window.
Redo	Redoes the most recent action in the active window.
Cut	Moves the selected graph objects to the clipboard.
Copy	Copies the selected graph objects to the clipboard.
Paste	Pastes the contents of the clipboard to the selected location.
Delete	Deletes selected objects, such as labels, markers, traces, or graphs. **Note:** If no object is selected, a message appears confirming the deletion of the active subwindow. If only one window is open, the *Delete* command is not available.
Delete All	Deletes all the objects and graphs in the active window. If only one window is open, the *Delete All* command is not available.
Properties	Enables you to modify the properties of the recently selected graph object. By default, the graph properties form appears if you do not select any object.
Multi-Graph Properties	Enables you to modify the properties of all the windows in one step.

You can cut, copy, and paste between multiple Virtuoso Visualization and Analysis XL sessions within the same Virtuoso process. You can move entire graphs in this manner, or individual traces along with the associated markers. The clipboard contents are retained even when a Virtuoso Visualization and Analysis XL session is closed and another new session is invoked from same Virtuoso process.

© 2023 Cadence Design Systems, Inc.

View

The table below lists the *View* menu commands.

Command	Description
ZoomIn by 2	Zooms in the graph by a factor of two.
ZoomOut by 2	Zooms out the graph by a factor of two.
Zoom to End	Zooms the trace to the end of the graph.
Fit	Returns the graph to the original size to fit data in the window. This command works for both rectangular and circular graphs.
Previous	Enables you to view the graph at the magnification specified before the last zoom in or zoom out command was run. You can use this option when you zoom in or out a graph multiple times.
Next	Undoes the *Previous* command. You can use this option when you zoom in or out a graph multiple times. **Note:** The *Next* and *Previous* commands help you navigate through the zoom and the pan stack.
Fit Trace	Returns the selected trace to its original size to fit in the window. When you select this option, the X-axis zoom of all the strips is displayed in the original size, where as the Y-axis zoom of only selected axis is changed. This command works for both rectangular and circular graphs.
Fit Visible Traces	Fits all the traces visible on the graph to their original size.
Fit Y to Visible X	Fits the visible part of the trace to Y-axis. This command finds the minimum and maximum Y-axis values that are visible in a strip and then performs a Y-axis zoom of those Y values. This command works only for the zoomed-in graphs.

© 2023 Cadence Design Systems, Inc.

Command	Description
Fit Y to Visible X all Strips	Fits the visible part of the traces to Y-axis for all the strips displayed in the active graph. This command finds the minimum and maximum Y-axis values that are visible in each strip and then performs a Y-axis zoom of those Y values. This command works only for the zoomed-in traces.
Fit Smith	Returns the selected Smith chart to its original size so that it fits into the window. This command is available only for the circular graphs.

Graph

The table below lists the *Graph* menu commands.

Command	Description
Layout	Specifies how subwindows are displayed in the active window. You can select the layout as *Auto*, *Vertical*, *Horizontal*, and *Card*. For more information about graph layouts, see Setting the Graph Colors on page 190.
Link	Links rectangular graphs in the subwindows of a graph window. For more information, see Linking Graphs in Subwindows.
Add Label	Adds a label to the graph. For more information about graph labels, see Working with Graph Labels on page 422.
Lock	Locks the graph from any data updates. For more information, see Locking Graphs on page 224.
Split Current Strip	Splits the graph into as many strips as there are traces and displays each trace in the graph in a separate strip. You can also select this command from the Strip toolbar. For more information, see Working with Strips on page 398.

© 2023 Cadence Design Systems, Inc.

Command	Description
Split All Strips	Splits the traces in all the strips in the graph into individual strips. This is useful if the graph contains more than one strip.
Plot to New Strip	Plots the selected trace in a new strip.
Combine All Analog Traces	Combines all the individual analog traces into a single graph. For more information, see Combining Graph Strips on page 405.
Filter By Sweep Var	Displays the traces for the selected sweep variable range.
Redraw	Refreshes the graph and plots the updated graph in the same window. This command also refreshes the trace legend area.
Toggle Major and Minor Grids	Displays or hides the major and minor grids in the selected axis. Alternatively, you can use bindkey `G` to toggle between the major and minor grids. To use this bindkey, ensure that an axis is selected.
Properties	Sets the graph properties. You can set the general graph properties as well as the strip properties in the *Graph Properties* form that appears when you select *Properties*. For more information, see Editing Graph Properties on page 196.

Axis

The table below lists the *Axis* menu commands.

Note: These commands are available only if you select an axis in the graph.

Command	Description
Major Grids	Displays the major grid lines for the selected X or Y axis.
Minor Grids	Displays the minor grid lines for the selected X or Y axis.
Log	Displays the logarithmic scale for the selected X or Y axis.

© 2023 Cadence Design Systems, Inc.

Command	Description
Select Attached Traces	Selects all the traces that are attached to the axis you select.
Y vs Y	Displays the YvsY plot of the selected axis in the window. This command is available only for the sweep data. For more information, see Plotting YvsY Graph on page 428.
Swap Sweep Var	Enables you to swap sweep variables. This command is available only if you select the sweep data. For more information, see Swapping Sweep Variables on page 412.
Properties	Sets the attributes for the selected X or Y axis. For more information, see Editing Graph Axis Attributes on page 202.

Trace

The table below lists the *Trace* menu commands.

Command	Description
Symbols On	Displays symbols on individual data points for the selected trace. **Note:** This command is available only if you select one or multiple trace in the graph.
Select by Family	Selects all the traces with the parametric sweep data that belong to a family. When you enable this command, and select a trace in the family, all traces that belong to the same family are selected.
Strip by Family	Displays traces that belong to the same family in a single strip when you split traces into strips. If more than one family of traces are present, each family is displayed in a separate strip.

© 2023 Cadence Design Systems, Inc.

Command	Description
Fit Trace	Returns the selected trace to its original size to fit in the window. When you select this option, the X-axis zoom of all the strips is displayed in the original size, where as the Y-axis zoom of only selected axis is changed.
Fit Y to Visible X	Fits the visible part of the trace to Y-axis. This command finds the minimum and maximum Y-axis values that are visible in a strip and then performs a Y-axis zoom of those Y values. This command works only for the zoomed-in graphs.
Disable Reload	Disables the reloading of a trace by locking the database context and the trace is not reloaded with new data when the in-context results directory is changed. For more information, see Disabling Trace Reload on page 447.
Select All	Selects all traces in a graph.
Delete All	Deletes all the traces displayed in the active graph. The independent axis, window title, and pan bar are not deleted.
Move to	Moves the selected trace to the following locations: ■ *New Window*—Moves the selected trace to a new window. ■ *New Subwindow*—Moves the selected trace to a new subwindow. ■ *New Strip*—Moves the selected traces to a new strip.
Copy to	Copies the selected trace to the following locations: ■ *New Window*—Copies the selected trace to a new window. ■ *New Subwindow*—Copies the selected trace to a new graph subwindow. ■ *New Strip*—Copies the selected trace to a new graph strip.

© 2023 Cadence Design Systems, Inc.

Command	Description
Bus	This command has the following options: *Create*—Creates a bus from the selected digital traces. For more information, see Creating a Bus on page 477. *Expand*—Expands a bus to its component signals. For more information, see Expanding a Bus on page 491. **Note:** This option is available only if you select a digital bus. *Collapse*—Collapses the bus components to display the complete bus. **Note:** This option is available only if you expand the bus.
Export	Exports the selected trace in the active window in a variety of formats and later loads it in the required application.
Properties	Enables you to specify properties of the selected trace. For more information, see Setting Trace Properties on page 392. **Note:** This option is available only if you select a trace in the graph.

Marker

The table below lists the *Marker* menu commands.

Command	Description
Tracking Cursor	Enables or disables the tracking cursor for the graph. When you move the mouse pointer on a trace or on a graph object, the tracking cursor displays the trace name and the graph object information.

© 2023 Cadence Design Systems, Inc.

Command	Description
Snap Tracking Cursor	Snaps the tracking cursor to the simulation points. When you move the mouse pointer on the simulation points on a trace, the tracking cursor displays the trace name and the graph object information
Create Marker	Creates a new marker for the trace in the graph. For more information, see Adding Markers on page 495.
Create Delta Marker	Creates a new delta marker. To create a delta marker, you need to place a point marker on the trace or select an existing point marker. For more information, see Adding AB Marker on page 524.
Show Delta Child Labels	Shows or hides marker labels for the delta markers.
Delete all	Deletes all the markers displayed in the active subwindow or a graph.
Export Table	Exports the selected marker information in a given format.
Properties	Specifies the properties for a marker. For more information, see Setting Marker Properties on page 509. **Note:** This command is available only if you select a marker on the graph.

Measurement

The table below lists the *Measurement* menu commands.

Command	Description
Eye Diagram	Plots an eye diagram for the selected graph. The eye diagram divides the waveforms into fixed time periods, which are then superimposed on each other. When you select this command, the Eye Diagram assistant appears. For more information, see Eye Diagram Assistant on page 242.

© 2023 Cadence Design Systems, Inc.

Command	Description
Spectrum	Plots a spectrum for the selected graph. When you select this command, the Spectrum assistant appears. For more information, see Spectrum Assistant on page 226.
Analog to Digital	Converts an analog signal into a corresponding digital signal. For more information, see Converting a Digital Signal to an Analog Signal on page 467. **Note:** This command is available only in the SKILL mode.
Digital to Analog	Converts a digital signal into a corresponding analog signal. For more information, see Converting an Analog Signal into a Digital Signal on page 465. **Note:** This command is available only in the SKILL mode.
Derived Plots	Generates the derived plots that are the risetime or falltime waveforms derived from the entire set of edges and plotted against time. For more information, see Generating Derived Plots on page 469.
Histogram	Generates the histogram plot directly on a graph. For more information, see Plotting Histogram on page 471.
Transient Measurement	Opens the Transient Measurement assistant that displays the calculated measurements for the transient markers on specific edges. For more information, see Transient Measurement Assistant on page 308.

© 2023 Cadence Design Systems, Inc.

Tools

The table below lists the *Tools* menu commands.

Command	Description
Calculator	Opens the Virtuoso Visualization and Analysis XL Calculator window. For detailed information about working with the Calculator, see Chapter 5, "Working with the Calculator."

Window

The table below lists the *Window* menu commands.

Command	Description
Assistants	Displays or hides the selected assistant panes. The available assistants are—Spectrum, Browser, Marker Toolbox, Eye Diagram, Direct Measurements, Horiz Marker Table, Trace Info, Vert Marker Table, Customize Trace Groups, and Subwindows. For more information, see Assistants on page 163. For more information about assistant panes, see the *Virtuoso Studio Design Environment User Guide*.
Workspaces	Displays, saves, loads, and configures the selected workspace. The available workspaces are—Basic, Browser, Classic, and MarkerTable. For more information, see Working with Workspaces on page 365. For more information about workspaces, see "Getting Started with Workspaces" in *Virtuoso Studio Design Environment User Guide.*

© 2023 Cadence Design Systems, Inc.

Command	Description
Toolbars	Displays or hides the selected toolbars. The available toolbars are—Edit, View, Graph, Calculator, Snap, Marker, Strip, Measurement, Axis, and Workspaces. For more information about toolbars, see Toolbars on page 157.

Browser

The table below lists the *Browser* menu commands.

Command	Description
Results	Includes the following Results Browser commands: ■ *Open Results*—Opens the results directory in the Results Browser. When you select this command, the *Select Waveform Database* form appears that you can use to select the results database. ■ *Export*—Exports a selected signal from the Results Browser. ■ *Close Results*—Closes the results directory in the Results Browser. This is available only if you select a results directory in the Results Browser. ■ *Reload*—Reloads the results directory that was last open into the Results Browser. ■ *Set Context*—Enables you to set the database in-context results directory, which you use to plot signals in the Results Browser. The first results directory that you load in the Results Browser is set as the in-context results directory.

© 2023 Cadence Design Systems, Inc.

Command	Description
Options	Includes the following commands: ■ *Graph Modifier*—Includes commands that you can use to specify how the graph is plotted: ❑ *Magnitude*—Plots magnitude versus frequency. ❑ *Phase*—Plots phase versus frequency. ❑ *WPhase*—Plots wrapped phase versus frequency. ❑ *Real*—Plots real value of the dependent data versus frequency. ❑ *Imaginary*—Plots imaginary value of the dependent data versus frequency. ❑ *dB10*—Plots *dB10* value of the dependent data versus frequency. ❑ *dB20*—Plots *dB20* value of the dependent data versus frequency. ❑ *dBm*—Plots mili dB value of the dependent data versus frequency. ■ *Plot Style*—Enables you to select the mode in which a graph is to be plotted. The signal in the graph can be plotted in the following modes: ❍ *Append*—Adds the signal to the selected graph. ❍ *Replace*—Replaces the signal in the selected graph with the new signal. ❍ *New Subwindow*—Plots the signal in a new subwindow within the active window. ❍ *New Window*—Plots the signal in a new window. ■ *Select Data*—Sets the sweep range for the data. When you select this command, the *Set Sweep Ranges* form appears. This command is available only if the dataset selected in the Results Browser supports ranging, such as the PSF transient dataset, or contains parametric sweep data.

© 2023 Cadence Design Systems, Inc.

Command	Description
	■ *Enable Fast Waveforms*—Enables the fast waveform format in which the Virtuoso Visualization and Analysis XL tool can render extremely large datasets within seconds.

Help

The table below lists the *Help* menu commands.

Command	Description
Search	A text field that lets you enter a search string. Press `Enter` to view the search results. **Note:** Do not enclose the search string in double quotes.
User Guide	Opens Virtuoso Visualization and Analysis XL User Guide (at the section that provides information about using CIW) in Cadence Help.
What's New	Opens the Virtuoso What's New document in Cadence Help.
Known Problems and Solutions	*Opens the Virtuoso Known Problems and Solutions document* in Cadence Help.
Virtuoso Documentation Library	Opens the Cadence Help home page, which provides quick access links to the following local and online resources: ■ What's New ■ Video Demos and Tutorials ■ Featured Content ■ Known Problems and Solutions ■ Other web resources

© 2023 Cadence Design Systems, Inc.

Command	Description
Virtuoso Video Library	Opens the Video Library page available on Cadence Online Support (COS). This page lists the videos available for various Virtuoso products. **Note:** You must have a COS account to access the content available on COS. **Note:** Contact your IT support to ensure that the Internet ports required for video playback are enabled.
Virtuoso Rapid Adoption Kits	Opens the Rapid Adoption Kits page on COS. This page lists Rapid Adoption Kits (RAKs) available for various Virtuoso products.
Virtuoso Learning Map	Lists domain-specific training available on Cadence Training Services. Cadence Training Services learning maps provide a comprehensive visual overview of the learning opportunities for Cadence customers. They provide recommended course flows as well as tool experience and knowledge levels to guide customers through a complete learning plan.
Virtuoso Custom IC Community	Opens the Virtuoso Custom IC Community web page. This page provides access to the latest blogs and discussion threads on various Virtuoso products and design topics, information about software downloads and support and training, and other related information. You too can contribute to the community forum by creating a Cadence account. This gives you additional benefits such as alerts about topics of interest and access to online webinars.

© 2023 Cadence Design Systems, Inc.

Command	Description
Cadence Online Support	Opens COS, which you can use to access information about Cadence products, documentation, videos, RAKs, application notes, troubleshooting information, alerts, and so on. Improvements are regularly made to COS to make it convenient for you to look up the information you want. We recommend that you bookmark this web site and use it as your first point of reference for any Virtuoso-related information. *Tip* You can also access COS by clicking the Cadence logo available in the upper-right banner in each Virtuoso window.
Cadence Training	Opens the Cadence training web page. You can find on this page information about the training courses available in different regions. Information is available about both classroom and online courses.
Cadence Community	Opens the Cadence Community web page. This page provides access to the latest blogs and discussion threads on various Cadence products and solutions, and EDA Industry Insights. You too can contribute to the community forum by creating a Cadence account. This gives you additional benefits such as alerts about topics of interest and access to online webinars.
Cadence OS Platform Support	Provides information about the current Cadence software releases and the supported platforms.
Contact Us	Opens the Cadence Customer Support web page, which provides customer support contact information for different regions.
Cadence Home	Opens the Cadence corporate web site.
About Virtuoso	Displays Virtuoso Studio Design Environment version information.

© 2023 Cadence Design Systems, Inc.

Toolbars

Do one of the following to show or hide toolbars in Virtuoso Visualization and Analysis XL:

- Choose *Windows – Toolbars – Toggle Visibility*.
- Right-click anywhere in the menu bar and select the toolbars that you want to show.
- Press the bindkey `Ctrl+F11` to toggle the visibility of toolbars and press the bindkey `Shift+F11` to toggle the visibility of toolbars and assistants.

The Virtuoso Visualization and Analysis XL has the following toolbars:

- Mouse Bar on page 157
- Edit Toolbar on page 158
- View Toolbar on page 158
- Layout Toolbar on page 159
- Graph Toolbar on page 159
- Calculator Toolbar on page 159
- Snap Toolbar on page 159
- Marker Toolbar on page 160
- Measurement Toolbar on page 160
- File Toolbar on page 160
- Strip Toolbar on page 162
- Axis Toolbar on page 162
- Workspace Toolbar on page 162

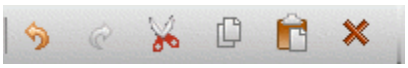
Mouse Bar

Displays at the bottom of the Virtuoso Visualization and Analysis XL window to indicate the left, middle, and right mouse movements.

© 2023 Cadence Design Systems, Inc.

Edit Toolbar

The Edit toolbar contains the following buttons:



- Undo
- Redo
- Cut
- Copy
- Paste
- Delete

For information about these toolbar buttons, refer to the Edit menu commands.

View Toolbar

The View toolbar contains the following buttons:

- Previous
- Next
- Fit
- ZoomIn by 2
- ZoomOut by 2
- Fit Trace
- Fit Y Visible
- Fit Smith

For information about these toolbar buttons, refer to the View menu commands.

© 2023 Cadence Design Systems, Inc.

Layout Toolbar

Displays the Layout button to specify the layout of the subwindows in an active window. For more information, see Specifying the Subwindows Layout on page 168.

Graph Toolbar

The Graph toolbar contains the following icons:

- *Layout Icons*—Shows the graph layouts that you can use to change the layout of the active window:
  - Auto
  - Card
  - Vertical
  - Horizontal
  - Layout

  For more information about graph layouts, see Setting the Graph Colors on page 190.
- *Subwindow*—Lists all the subwindows that are open in an active window. When you select a subwindow in this list, the selected subwindow is highlighted in the window. For more information about subwindows, see Working with Workspaces on page 365.

Calculator Toolbar

Displays the Calculator button to send the selected trace to the Calculator Buffer.

Snap Toolbar

The Snap toolbar contains the following buttons:

- *Previous POI*—Moves the selected marker to previous point of interest based on the snapping criteria selected.

© 2023 Cadence Design Systems, Inc.

- *Next POI*—Moves the selected marker to the next point of interest based on the snapping criteria selected.
- *Snapping Criterion*—Displays the criterion based on which the selected marker is snapped.
- *Value*—Displays the value of the snapping criterion.

This toolbar is available for both analog and digital signals and the toolbar options work if you select a marker. For more information, see Snapping Markers on page 514 and Snapping Markers on Circular Graphs on page 554

Marker Toolbar

The Marker toolbar contains the following buttons:

- *Create Marker*—Creates a marker on the selected trace. For more information about how to create a marker, see Adding Markers on page 495.
- *Tracking Cursor*—Turns the tracking cursor on and off for the selected window. For more information, see Tracking Cursor on page 391.

Measurement Toolbar

The Measumerent toolbar includes the following button:

- *Histogram*—Opens the Histogram form that you can use to plot a histogram for the selected signal. For more information, see Plotting Histogram on page 471.

File Toolbar

The File toolbar has the following buttons:

- *Create New Window*—Creates a new window. You can choose the type of the window to be created from the drop-down list that includes the following options:

© 2023 Cadence Design Systems, Inc.

- *Rectangular*
- *Tabular*
- *Polar*
- *Impedance*
- *Admittance*
- *Immittance*

- *Create New Subwindow*—Creates a new subwindow. You can choose the type of the subwindow to be created from the drop-down list that includes the following options:
  - *Rectangular*
  - *Tabular*
  - *Polar*
  - *Impedance*
  - *Admittance*
  - *Immittance*
- *Load Window*—Loads a graph window. For more information, see Loading a Graph on page 433
- *Save Window*—Saves the selected graph window. For more information, see Saving a Graph on page 431.
- *Print*—Prints the graph window. for more information, see Printing Graphs on page 449.
- *Save Image*—Saves the graph window in an image format. For more information, see Saving a Graph as an Image on page 434.
- *Create Maestro Plotting Template*—Saves a graph window as a plotting template after the simulation is run and results are plotted. The saved plotting template can be used to plot waveform outputs in the specified format for next simulation runs.

  Note: This icon appears dim if the Virtuoso Visualization and Analysis XL window is opened in the stand-alone mode.

  For more information about plotting templates, see Working with Plotting Templates in *Virtuoso ADE Explorer User Guide*.

© 2023 Cadence Design Systems, Inc.

Strip Toolbar

The Strip toolbar contains the following buttons:

- *Strip By*—Specifies how you want to strip the traces in a graph. The available options in the drop-down list are *family*, *leaf*, and *trace*. If you work on the traces from sweep data, the sweep variables are also included in this drop-down as strip options.
- *Combine All Analog Traces*—Displays all analog traces from individual strips to a single g. For more information, see Combining Graph Strips on page 405.
- *Split Current Strip*—Displays the traces in a window in individual strips. For more information, see Working with Strips on page 398.
- *Copy to a New Strip*—Copies the selected trace to a new strip in the same window.
- *Move to a New Strip*—Moves the selected traces to a new strip.

Axis Toolbar

You can use the Axis toolbar to turn on or turn off the grid from a graph.

Alternatively, you can do the following:

- Right click the graph and select *Toggle Major and Minor Grids* to turn on or off the grids.
- Select an axis and press the bindkey G.

Workspace Toolbar

You can use the Workspace toolbar to work with the available workspaces.

For more information about workspaces, see Working with Workspaces on page 365.

© 2023 Cadence Design Systems, Inc.

Status Bar

The status bar displayed at the bottom of the window displays the following information:

- Warnings and error messages.
- Static information, such as the name of the toolbar button selected in the window.
- Dynamic information, such as the toolbar names are displayed when you perform mouse-hover on toolbars.

Assistants

The Virtuoso Visualization and Analysis XL includes the following assistants:

- Spectrum Assistant
- Browser Assistant
- Marker Toolbox Assistant
- Eye Diagram Assistant
- Horiz Marker Table Assistant
- Trace Info Assistant
- Vert Marker Table Assistant
- Customize Trace Groups Assistant
- Direct Measurements Assistant

For detailed information about assistants, see Working With Assistants on page 225.

Creating a Graph

You can create a graph by plotting a signal selected in the Results Browser in the window. To group similar graphs or to compare two graphs, you can open multiple subwindows in a window.

To create a graph, perform the following steps:

1. In the Results Browser, open a results directory and select the signal you want to plot.
2. To select the window where you want to plot the signal, do one of the following:

© 2023 Cadence Design Systems, Inc.

- Choose *Browser – Options – Plot Style.*

The *Plot Style* can be of the following types:

- *Append*—Adds the waveform expressions or signals to the active window or subwindow. If the data in the selected subwindow is incompatible with the waveform expression or signal being appended, this waveform expression or signal is plotted in the next compatible window. If none of the existing subwindows are compatible, the waveform expression or signal is plotted in a new subwindow. The incompatible data refers to the waveforms generated from different analyses, such as AC, transient, and so on. In addition, if the X-axis of active subwindow is different from the X-axis of the waveform or signal being appended, the waveform is plotted in a new subwindow.
- *Replace*—Replaces the graph in the active window with a new graph.
- *New Window*—Plots the signals in a new window. When you create a graph for the first time, it is always displayed in a new window.
- *New Subwindow*—Plots the graph in a new subwindow within the active window.

- Right-click the required signal in the Results Browser and choose the destination graph. The plot options that appear on the shortcut menu are the same as those explained above.

3. After you specify the destination graph, do one of the following to plot the signal:
   - Double-click the signal.
   - Right-click the signal and select the *Plot Signal* option.
   - Click the button in the Results Browser.

   The graph appears in the selected destination window.

Dragging Graphs Across Multiple Virtuoso Visualization and Analysis XL Sessions

You can drag traces and graphs across different Virtuoso Visualization and Analysis XL opened within the same Virtuoso session. To copy a trace or a group of traces from one Virtuoso Visualization and Analysis XL session to another, select the traces by using the `Ctrl` key and then drag and drop the selected traces in the destination window of the another session.

Note: You cannot drag traces across two Virtuoso Visualization and Analysis XL sessions that are opened from within different Virtuoso processes. In this case, you can export or

© 2023 Cadence Design Systems, Inc.

import graph files to move traces from one session to another. For more information about how to export a trace, see Exporting a Trace on page 396.

While dragging, only the waveform data is copied to the subwindow of the another session. The drag operation does not save trace properties, such as color, linestyle, symbol, the markers and the Y-axis modifiers, such as dB10, dB20, phase.

Limitations

Dragging is not supported in the following cases:

- Traces or graphs generated after swapping the sweep variable
- YvsY plots
- When you drag traces from one session to another session, the database context of the copied waveform does not change, which means the copied waveform has the database context from the source session.

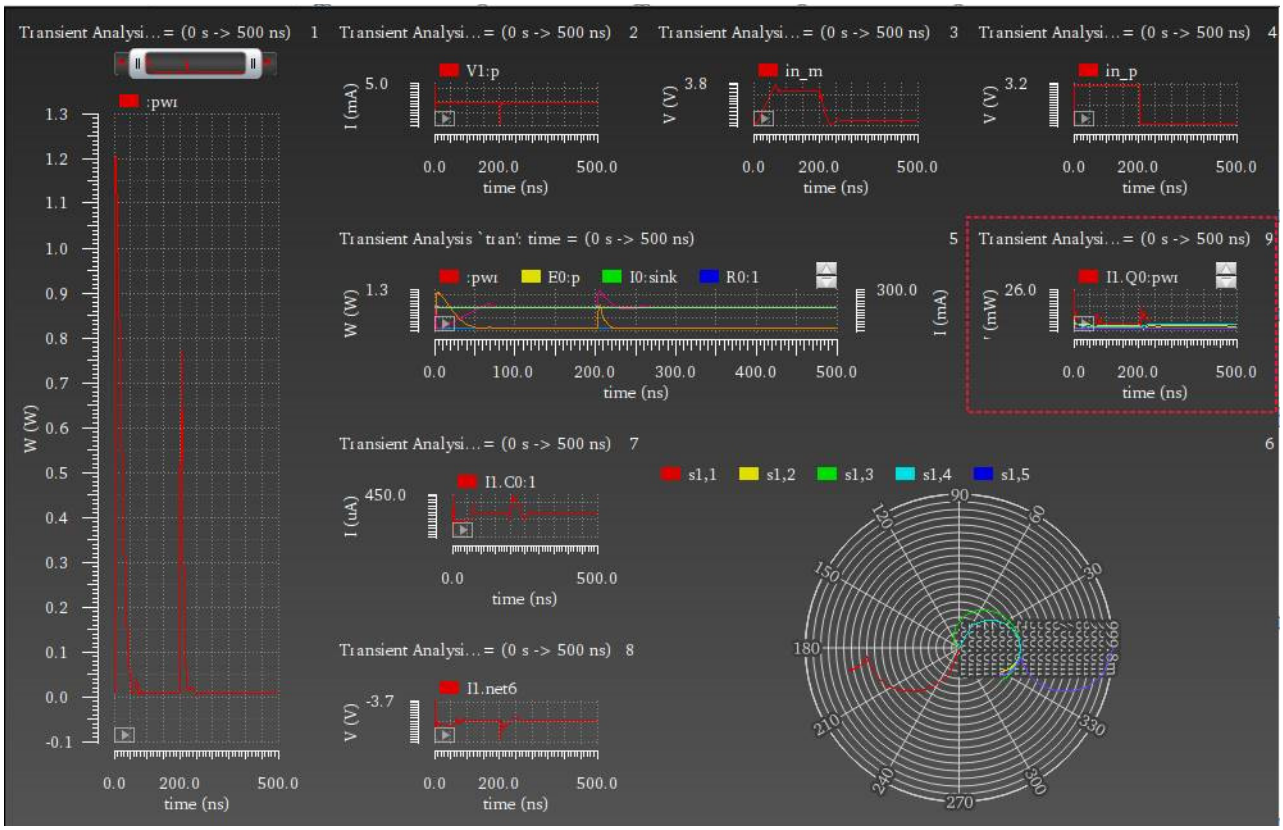
Working with Subwindows

A window can be divided into subwindows that displays waveforms in individual mini-graphs. Each subwindow can be set to show the rectangular, tabular, polar, impedance, admittance or immittance plots. You can open multiple empty subwindows in a graph window tab. When you open Virtuoso Visualization and Analysis XL for the first time with no results database open, by default, a graph window tab is opened with a blank subwindow showing the subwindow number as 1. You can open multiple empty subwindows and choose the graph type for them.

You can also customize these subwindows according to your requirements. The customizations can include – resizing of subwindows, combining subwindows with other existing subwindows, dragging subwindows to any location, merging subwindows, and swapping subwindows with other subwindows. You can also specify the layout in which subwindows appear in a graph window.

© 2023 Cadence Design Systems, Inc.

The figure below shows an example of the customized subwindows in a window.



Note: By default, subwindows are created based on a non-overlapping grid structure. You can also expand subwindows into any empty space to cover multiple grids together.

This section includes the following topics:

- Creating a New Subwindow on page 167
- Specifying the Subwindows Layout on page 168
- Customizing a Subwindow on page 172
- Selecting the Graph Type for Blank Subwindows on page 175
- Deleting a Subwindow on page 178
- Copying a Subwindow on page 179
- Resizing a Subwindow on page 180
- Moving a Subwindow on page 180
- Changing the Subwindow Properties on page 180

© 2023 Cadence Design Systems, Inc.

- Changing the Background Color of Subwindows on page 181

Note: You can also read the Virtuosity: New Flexible Subwindows blog for more information about the subwindows.

Creating a New Subwindow

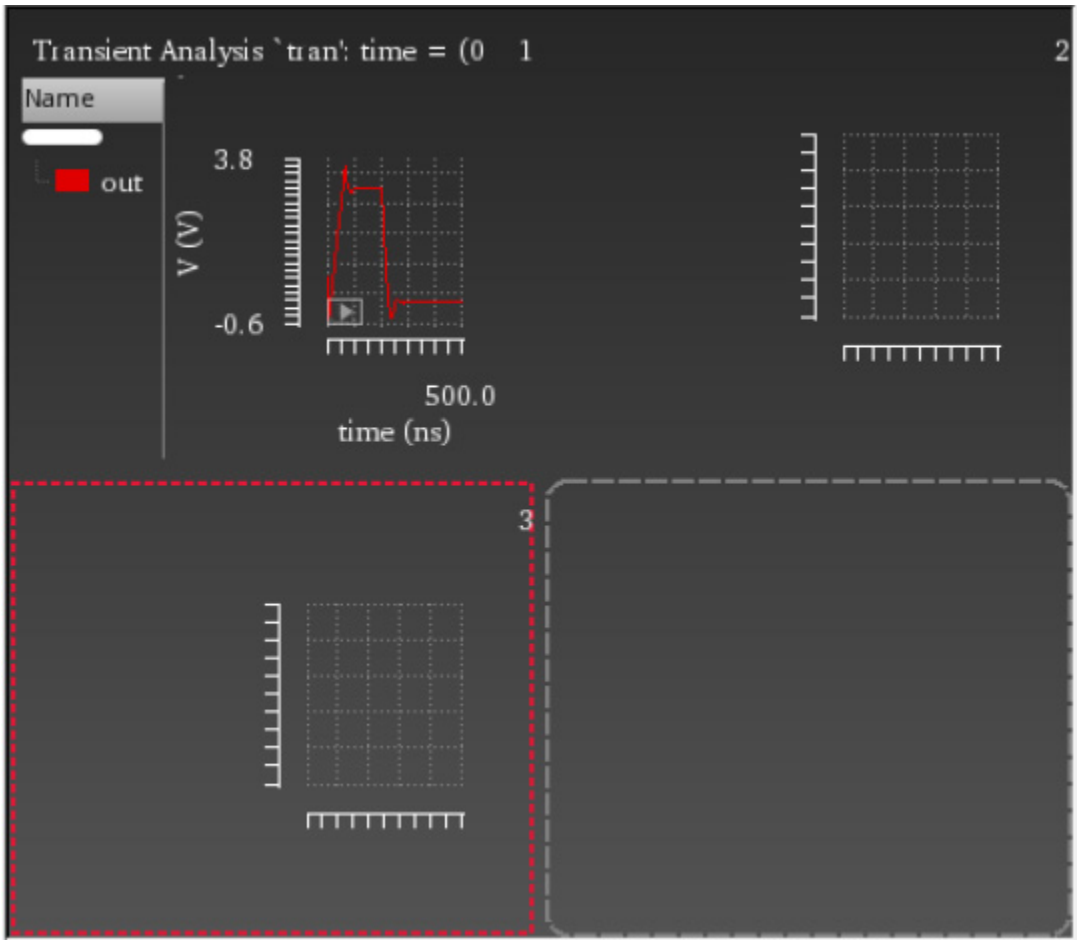
To create a new empty subwindow, do one of the following:

- Choose *File – New Subwindow – Rectangular/Tabular/Polar/Impedance/Admittance/Immittance*.
- On the Graph toolbar, click the *Create New Subwindow* ( ) drop-down list and choose *Rectangular/Polar/Impedance/Admittance/Immittance*.

  Selecting the *Rectangular* option creates a rectangular graph and selecting one of the *Polar*, *Impedance*, *Admittance* or *Immittance* options creates circular graphs. For

© 2023 Cadence Design Systems, Inc.

example, the following figure shows three rectangular subwindows opened in a graph window in the default subwindow layout.



Notice that the title of a subwindow includes the analysis name when the graphs are plotted. If a graph includes an expression, the expression name is also displayed in the subwindow title. The subwindow number is displayed in the *Subwindows* drop-down list, which includes an index number to indicate the subwindow number. This index number is also displayed at the right-most corner of the subwindow.

Specifying the Subwindows Layout

A window can have several subwindows that are displayed in the specified layout. When the subwindows contain plots, do one of the following to specify a layout for the subwindows:

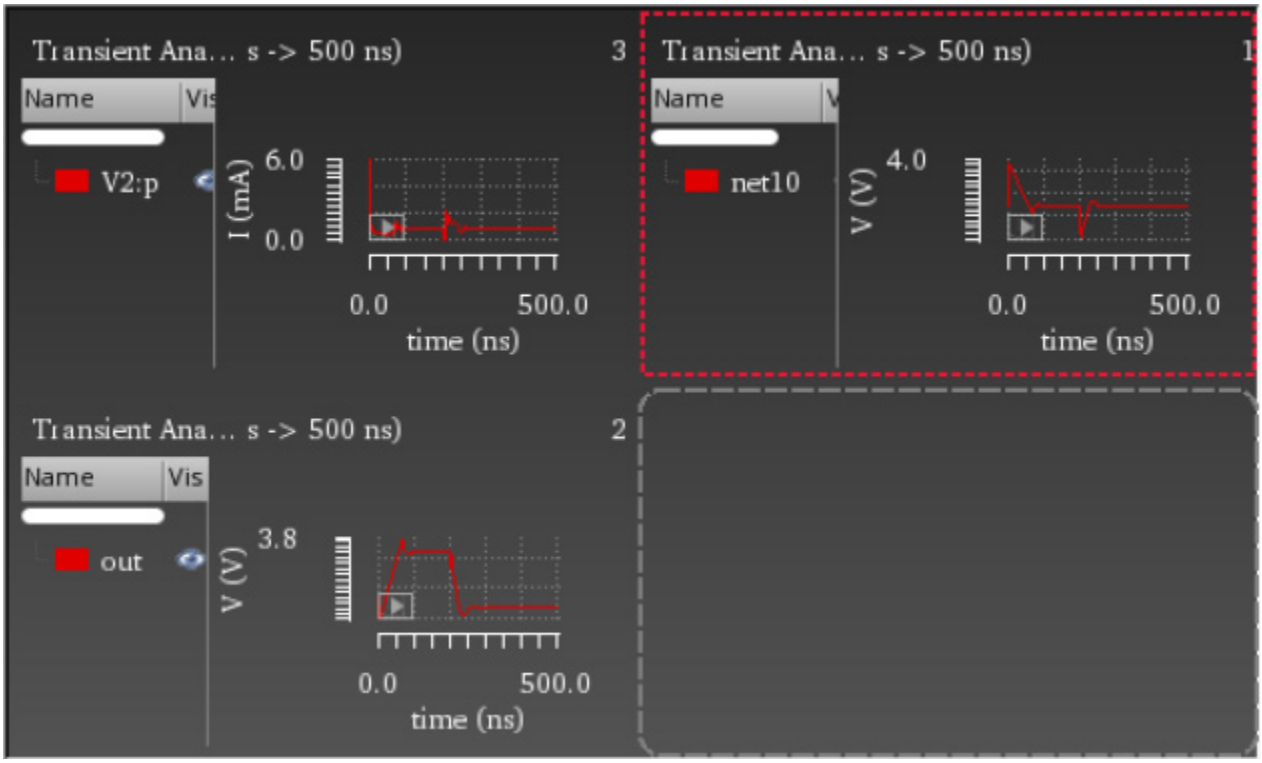
- Choose *Graph – Layout*.
- On the Layout toolbar, click the Layout icon.

© 2023 Cadence Design Systems, Inc.

- Right-click anywhere in the rectangular graph subwindow and choose the required layout from the *Layout* menu.

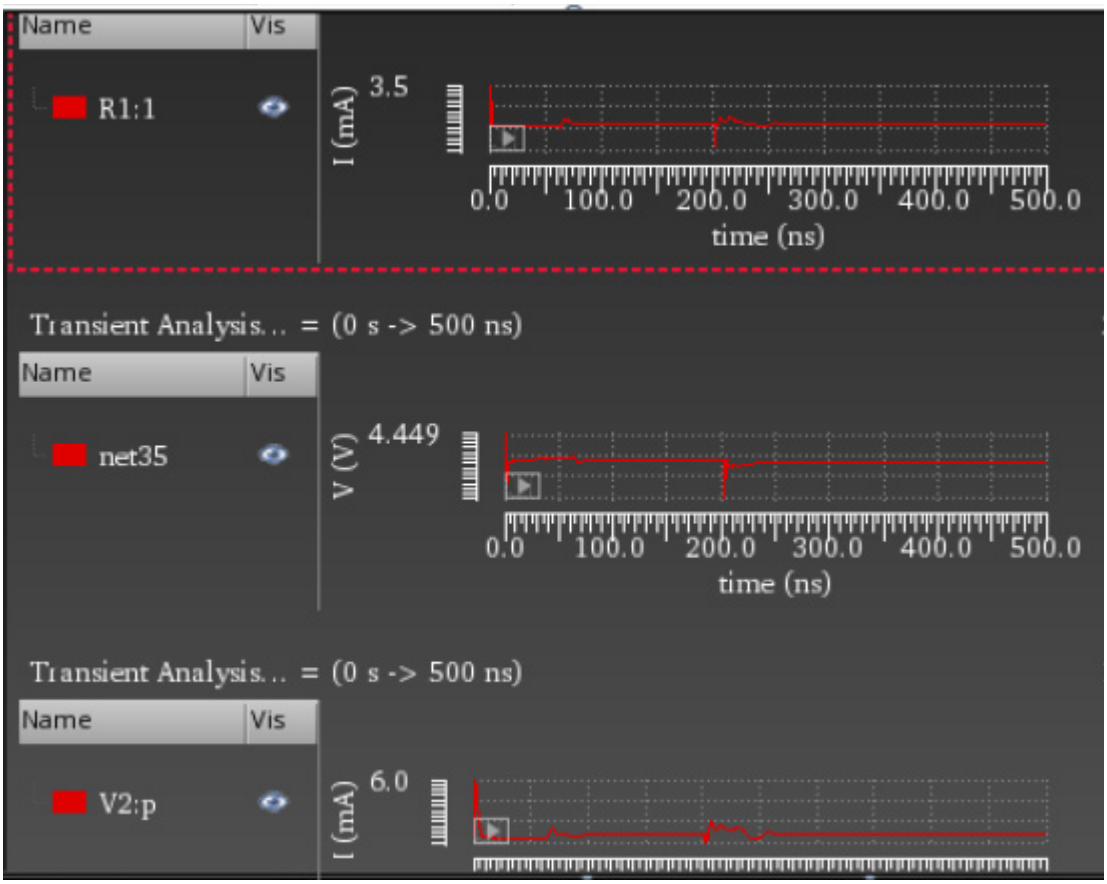
The following buttons for subwindows layout are displayed:

- *Auto*—This is the default layout. In this layout, subwindows are displayed by dividing the active window vertically and horizontally. The aspect ratio determines how the active window is divided. The following figure displays subwindows arranged in the Auto graph layout.



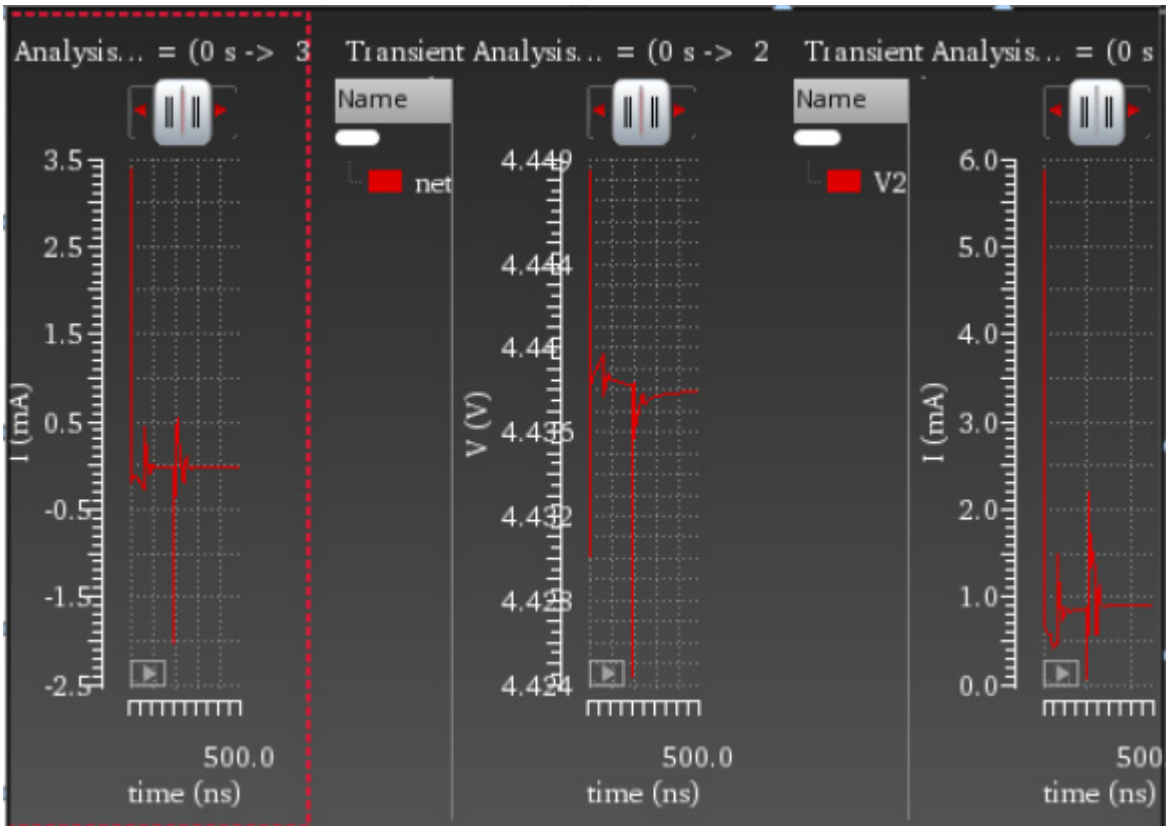
© 2023 Cadence Design Systems, Inc.

- *Vertical*—In this layout, subwindows are displayed one below the other in the active window. The following figure displays subwindows arranged vertically.



© 2023 Cadence Design Systems, Inc.

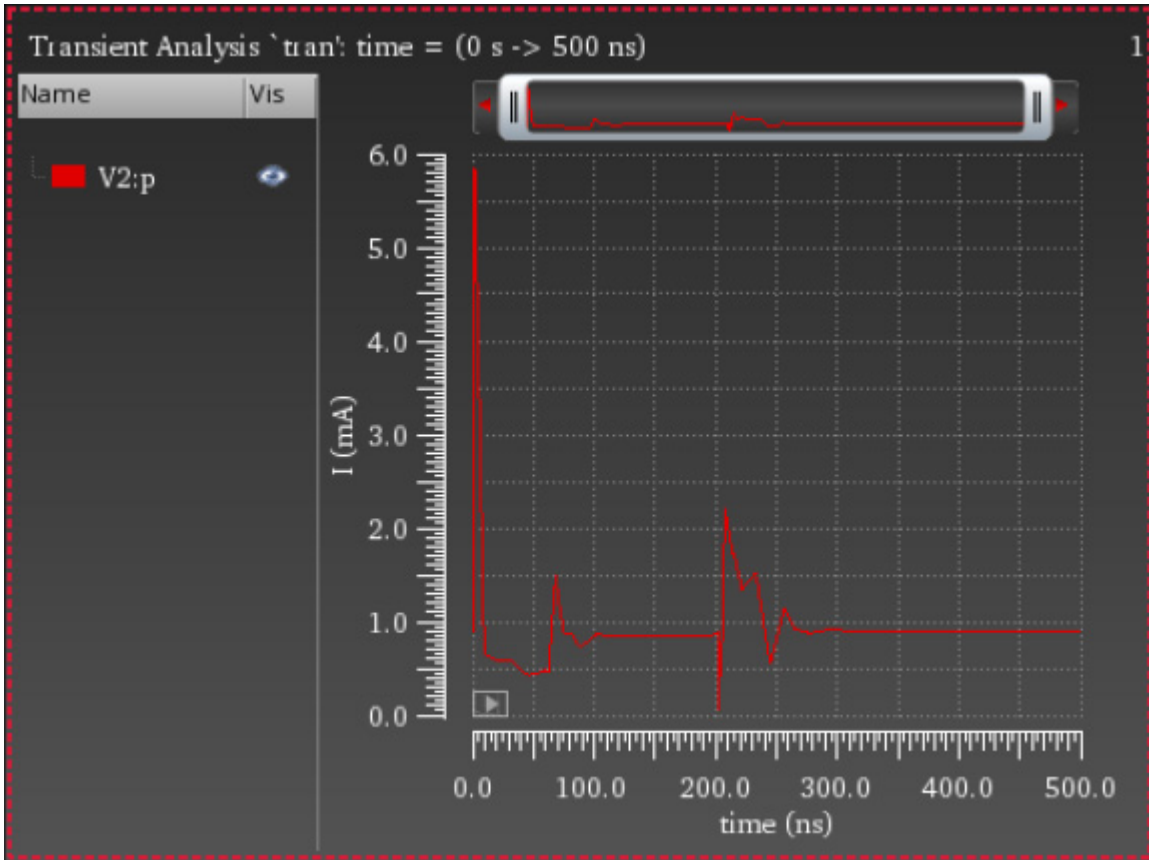
- *Horizontal*—In this layout, subwindows are displayed side by side in the active window. The following figure displays subwindows arranged horizontally.



- *Card*—In this layout, subwindows are stacked, one on top of the other, like a deck of cards. By default, the active window displays only the selected graph. If you want to view another subwindow, you can select the required subwindow in the Subwindows

© 2023 Cadence Design Systems, Inc.

drop-down list on the subwindows toolbar. The following figure displays the subwindows arranged in the card layout.



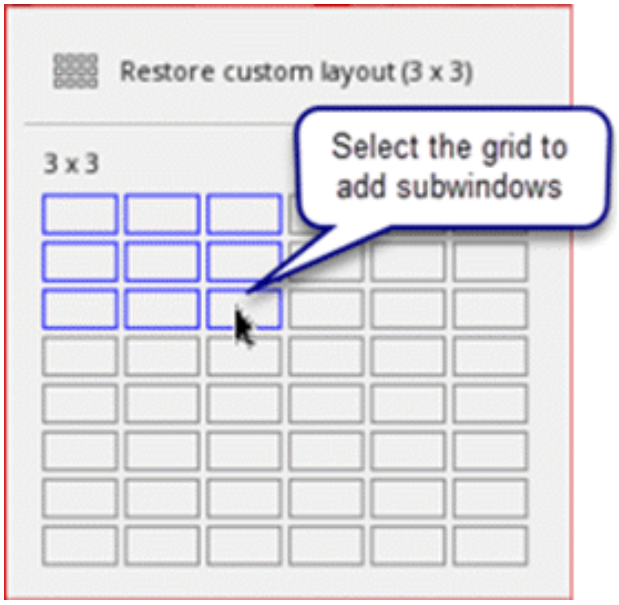
- *Custom*—You can use this layout to specify a new customized subwindows display.
- *Restore custom layout*—You can use this layout to reinstate the specified custom subwindows layout.

Customizing a Subwindow

By default, the subwindows are displayed in the Auto mode. To specify your own layout to customize the subwindows in a graph window:

© 2023 Cadence Design Systems, Inc.

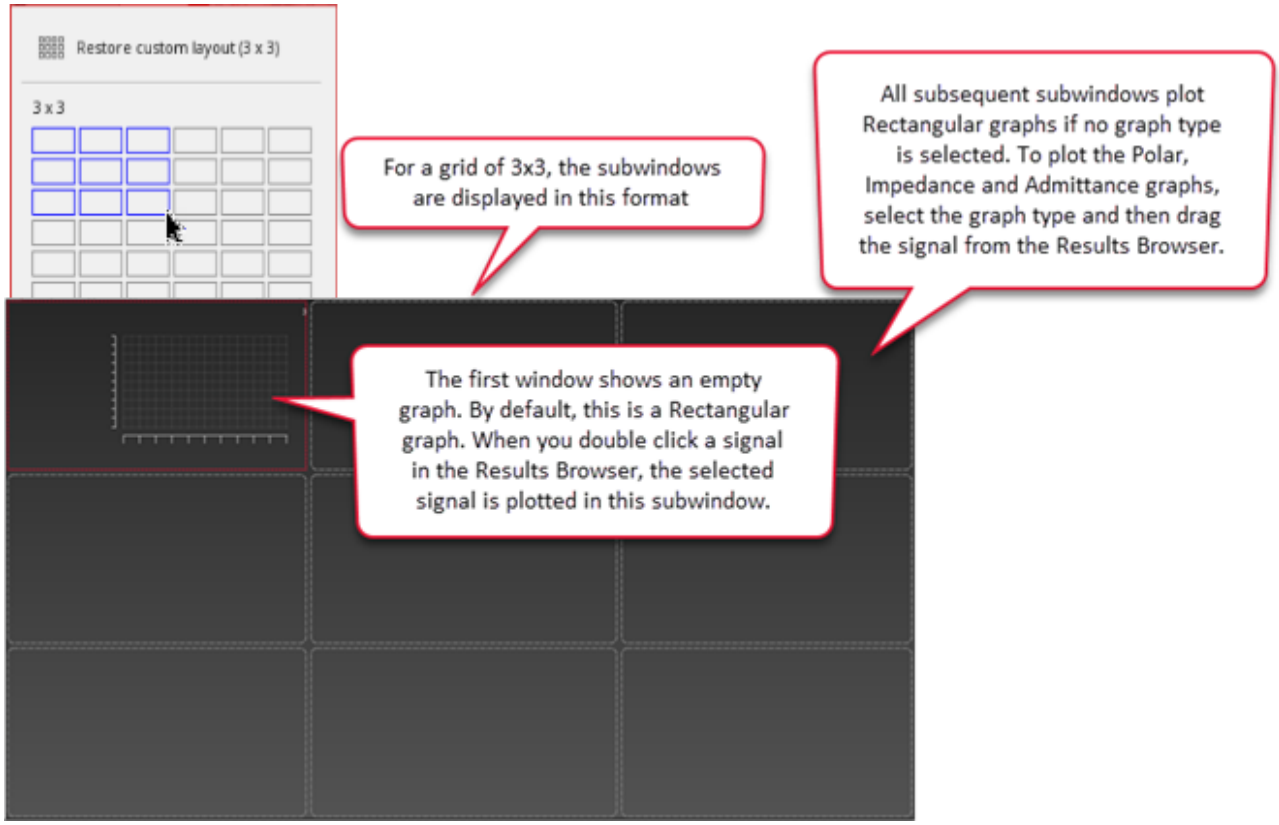
➡ In the Layout selection grid, select the grids, as shown in the figure below.



For the Auto, Horizontal, Vertical, and Card layouts, you first need to plot the signals in subwindows and then select the graph layout. The subwindows are displayed in the selected layout. However, for the custom layout, you first select a grid available on the layout options, which will open the empty subwindows to the right of the graph window. For example, if you choose a grid of `3x3`, a total of 9 subwindows are displayed with 3 rows, one below the other, and 3 columns, one after the other, as shown in the figure

© 2023 Cadence Design Systems, Inc.

below.The maximum available grid is `6x8`, which means maximum of `48` subwindows can be opened in a graph window.



Important

If you have created a customized layout, you can select another layout such as Auto to switch to a predefined layout. You can return to your custom layout simply by selecting the *Restore custom layout* option in the *Custom* layout selection grid.

The first subwindow to appear at the top left is the active subwindow. When you select a signal for plotting in the *Results Browser*, it is plotted in the active subwindow. By default, this subwindow plots only rectangular graphs. If you choose to plot a circular graph, its rectangular equivalent is displayed in this subwindow.

To plot the graphs in other subwindows:

- Ensure that the plot mode is set as *New Subwindow*.
- Double-click the signal in Results Browser. To plot multiple signals, select the signals using the `Ctrl` key, right-click the selection and choose *Plot Signal*.

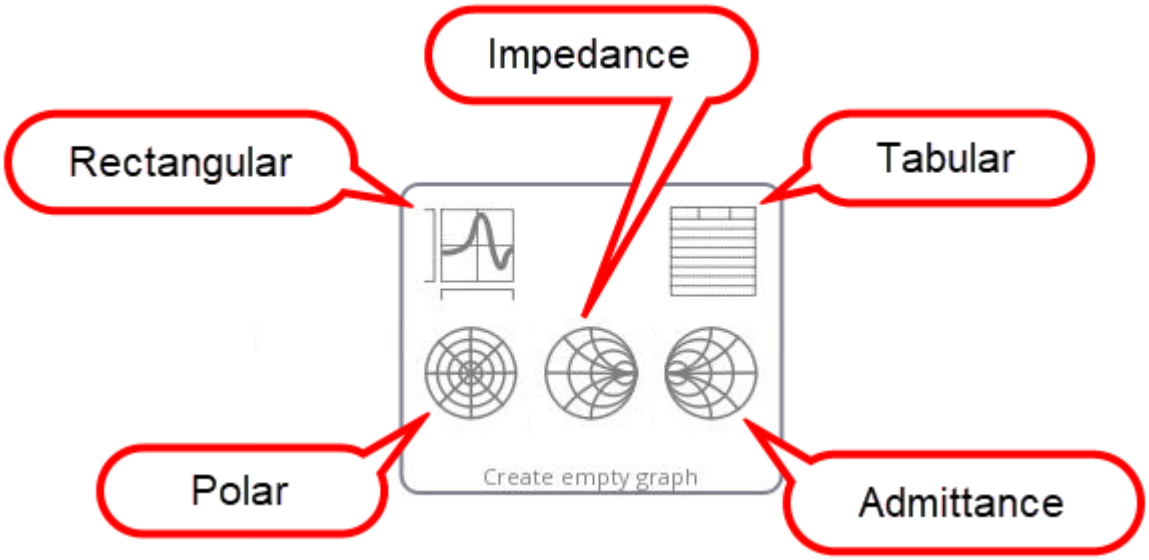
© 2023 Cadence Design Systems, Inc.

The selected signals are plotted sequentially in subwindows.

Selecting the Graph Type for Blank Subwindows

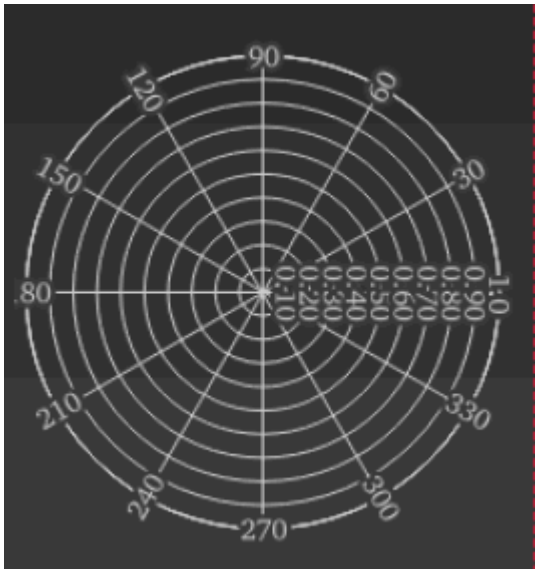
To select the graph type for blank subwindows:

➡ Place the pointer on a blank subwindow. The graph type options to plot rectangular, polar, impedance, admittance, and tabular graphs appear, as shown in the figure below:



© 2023 Cadence Design Systems, Inc.

- Select a graph type that you want to plot. For example, if you select the polar option, a blank circular graph of type polar is displayed in the subwindow, as shown in the figure below:

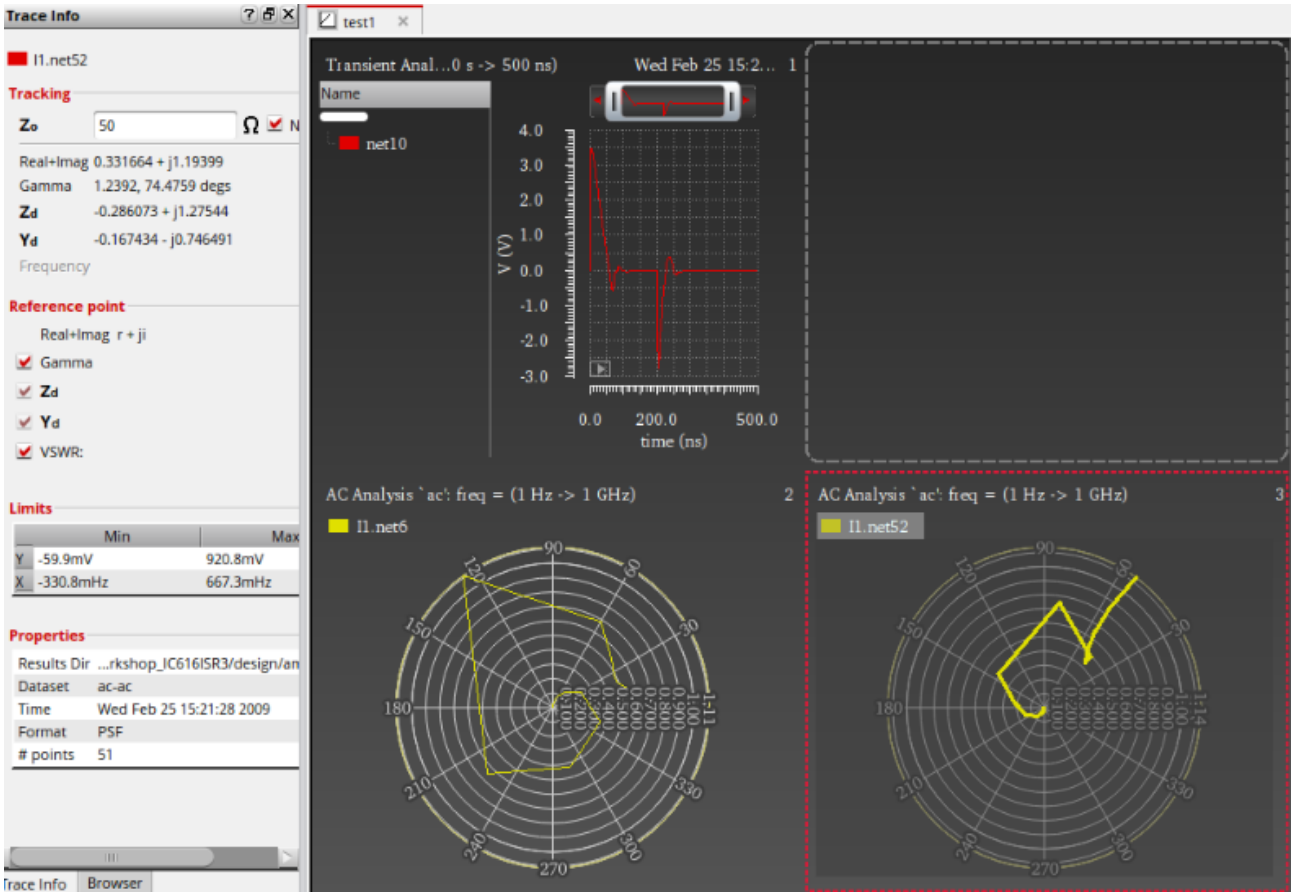


Important Points to Note

- When you drag a signal to a subwindow without selecting a rectangular or circular graph, the signal is plotted by default, as a rectangular graph. To select a circular graph, select any one of the Polar, Impedance or Admittance graph types in the blank subwindow, and drag the signal from the Results Browser to the subwindow.

© 2023 Cadence Design Systems, Inc.

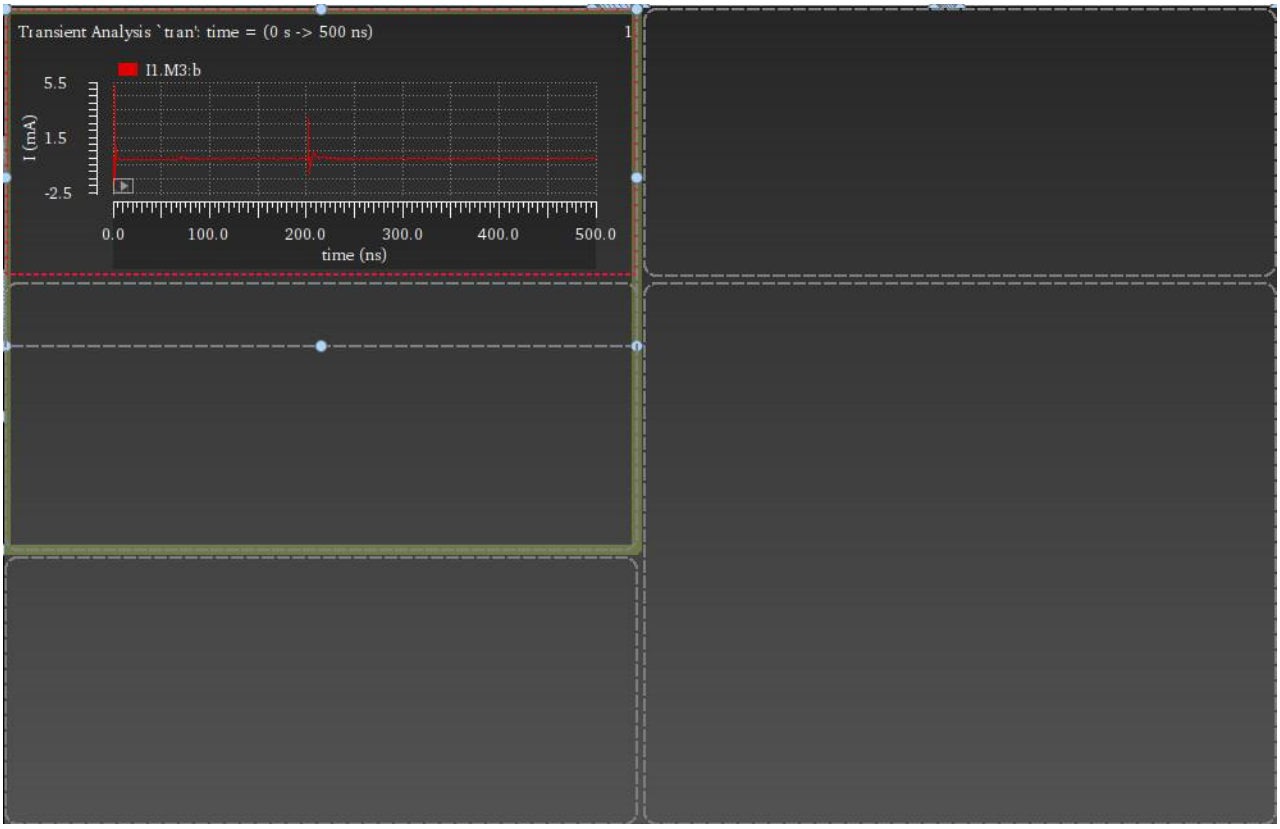
- When you plot a circular graph, the plotting information about the circular graph is displayed in the Trace Info assistant, as shown in the figure below:



- When you select a plot type for a blank subwindow, the adjacent subwindow cannot be expanded in the direction of this subwindow.
- When you click the blue dots (drag handle) on the subwindow border to expand the subwindow, a green border indicates that the subwindow can be expanded. For example, in the figure below, you can drag subwindow1 to the empty subwindow at the bottom.

© 2023 Cadence Design Systems, Inc.

However, you cannot drag the top right empty subwindow, adjacent to subwindow1, in the left direction because a graph is already plotted in subwindow1.



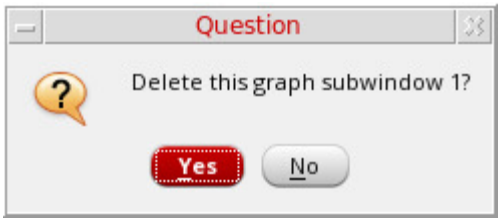
Deleting a Subwindow

To delete a subwindow, do one of the following:

- Right-click the graph in the subwindow and choose *Delete*.
- Select the subwindow you want to delete and choose *Edit – Delete*.
- Select the subwindow you want to delete and press the `Delete` key.

© 2023 Cadence Design Systems, Inc.

A confirmation dialog box appears asking you to confirm whether you want to delete the selected subwindow. Click *Yes* to confirm.



The selected subwindow is deleted.

Important points to note:

- You can choose not to display the confirmation dialog box by setting the confirmCloseWindow environment variable to `false`.
- In the non-custom layouts (Auto, Card, Vertical, and Horizontal), when you delete a subwindow, the remaining subwindows resize themselves to occupy the empty space created by the deleted subwindow provided that the number of remaining subwindows is an even number.

Copying a Subwindow

To copy a subwindow to a new window or a subwindow:

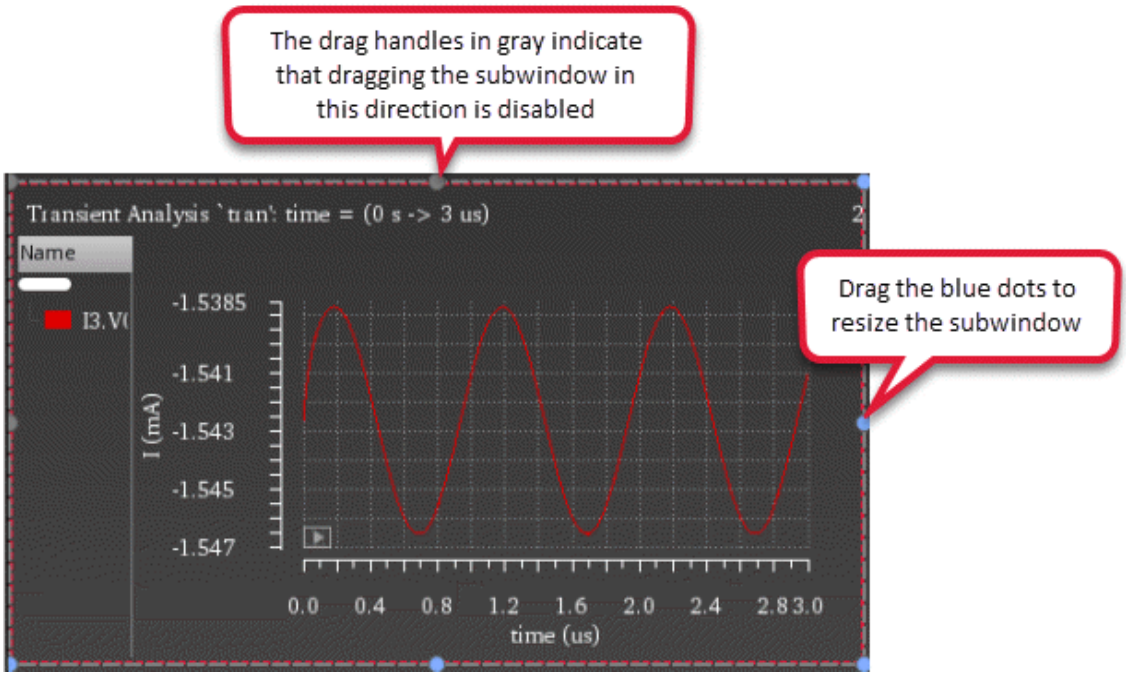
➡ Right-click a subwindow and choose *Copy To – New Window* or *New Subwindow* menu. The options are *Rectangular*, *Tabular*, *Polar*, *Impedance*, *Admittance* or *Immittance*.

The selected subwindow is copied to a new window or subwindow.

© 2023 Cadence Design Systems, Inc.

Resizing a Subwindow

To resize a subwindow, place the pointer on the subwindow. The resize box with drag handle appears. Drag these blue dots to resize the subwindow in the horizontal or vertical directions.



Note: The drag handle is displayed in gray to indicate that it is disabled and you cannot drag the subwindows in that direction.

Moving a Subwindow

To move the graph plotted in a subwindow to another subwindow, drag the subwindow using the pointer and place it in the desired subwindow or an empty subwindow.

Changing the Subwindow Properties

To change the properties of a particular subwindow:

➡ Select the subwindow and choose *Edit – Properties* or *Graph – Properties*.

The Graph Properties from appears in which you can change the graph attributes.

To change the attributes of all the windows and subwindows together:

➡ Choose *Edit – Multi-Graph Properties*.

© 2023 Cadence Design Systems, Inc.

The Multi-Graph Properties form appears in which you can change the graph attributes of all the windows and subwindows in one step. For more information, see Setting Properties for Multiple Graphs on page 199.

Changing the Background Color of Subwindows

By default, all the subwindows are displayed with a black background. To change the background color of the subwindows:

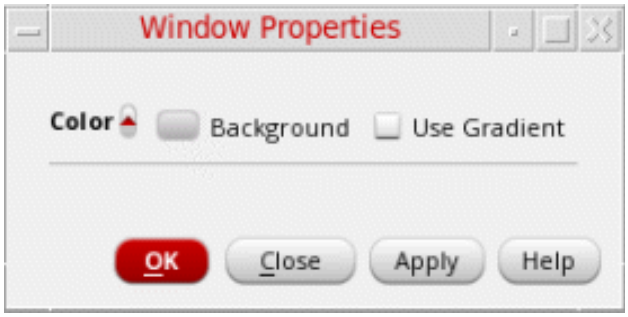
- Choose *Edit – Multi-Graph Properties*.

  The Multi-Graph Properties form appears in which you can change the graph background color of all the windows by selecting the *Background* button. For more information, see Setting Properties for Multiple Graphs on page 199.

- Choose *File – Window Properties*.

  The Window Properties form appears, as shown in the figure below. In this form:

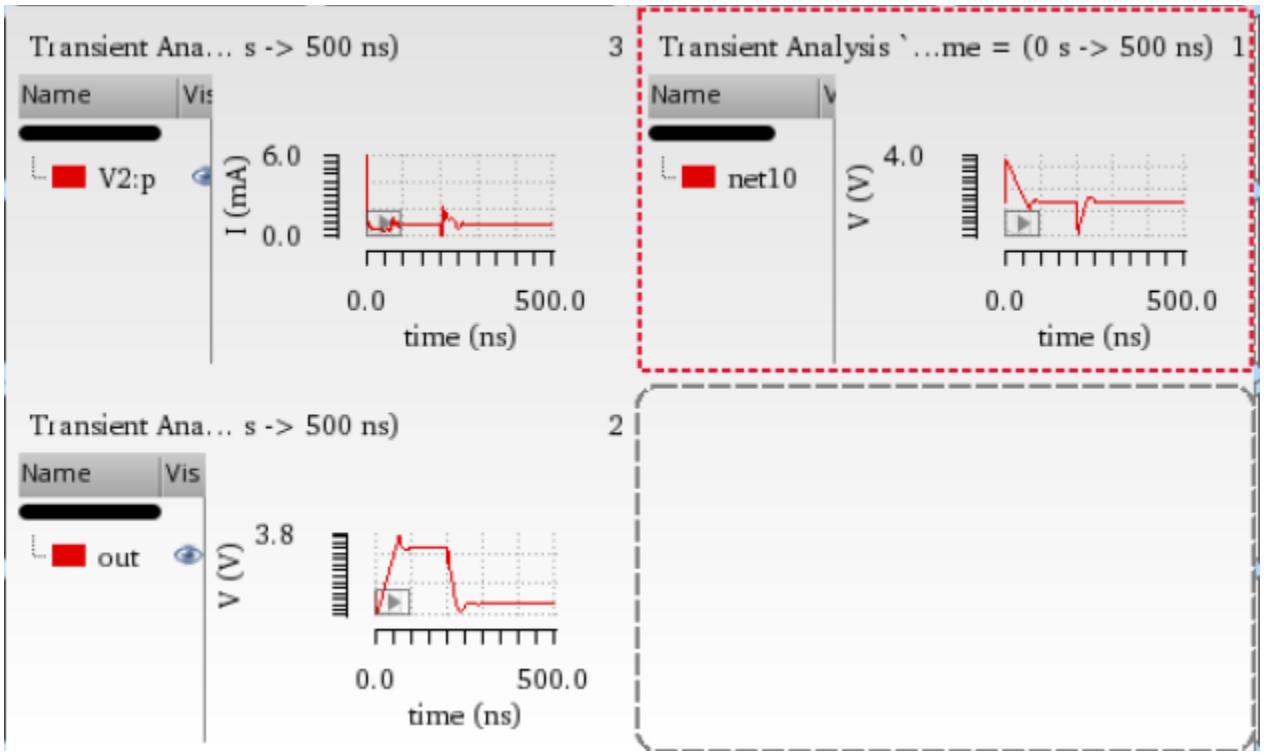
  - Click the *Background* button to set the background color of the selected window.
  - Select the *Use Gradient* option to add a gradient to the background, which means the background color fades at the bottom of the graph. By default, this checkbox is disabled.



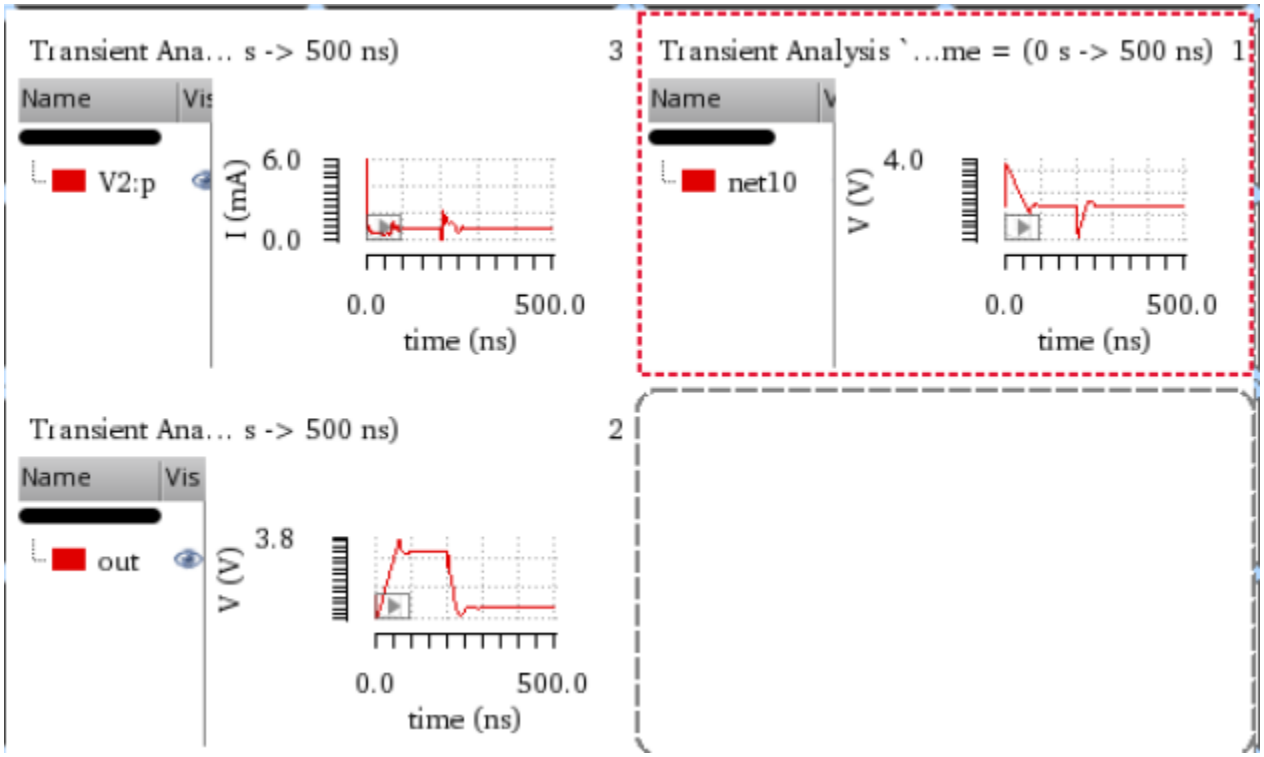
The following figures show the background color changed to white.

© 2023 Cadence Design Systems, Inc.

With Gradient:



Without Gradient:



© 2023 Cadence Design Systems, Inc.

Linking Graphs in Subwindows

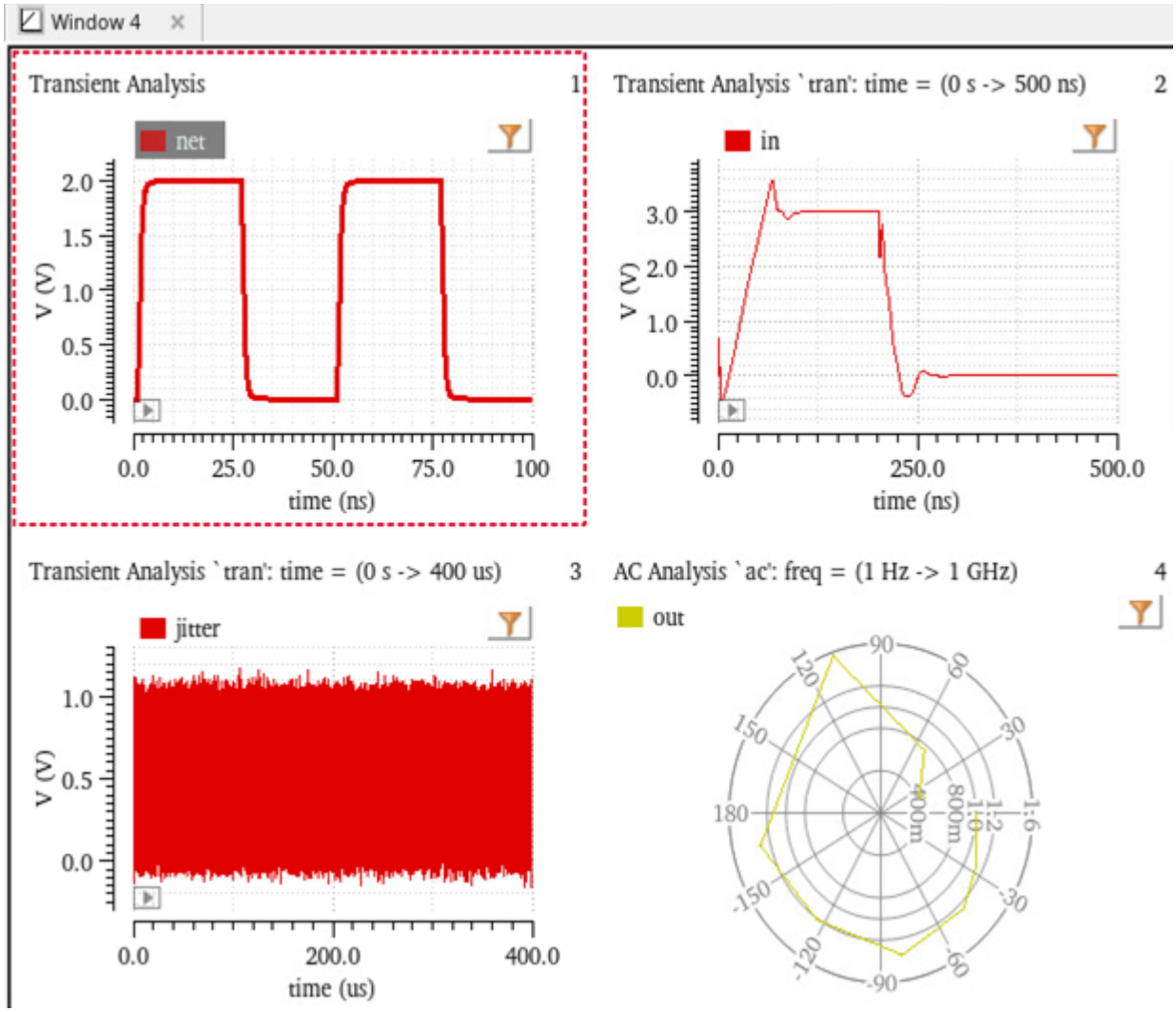
You can link graphs in subwindows to automatically set the same minimum and maximum values of x and y axes for the linked graphs. You can link only those subwindows of a graph window that share the same units for x and y axes.

Important

You can link only rectangular graphs.

To link graphs in the subwindows:

1. Select the subwindow that you want to link and choose *Graph – Link – Add*.

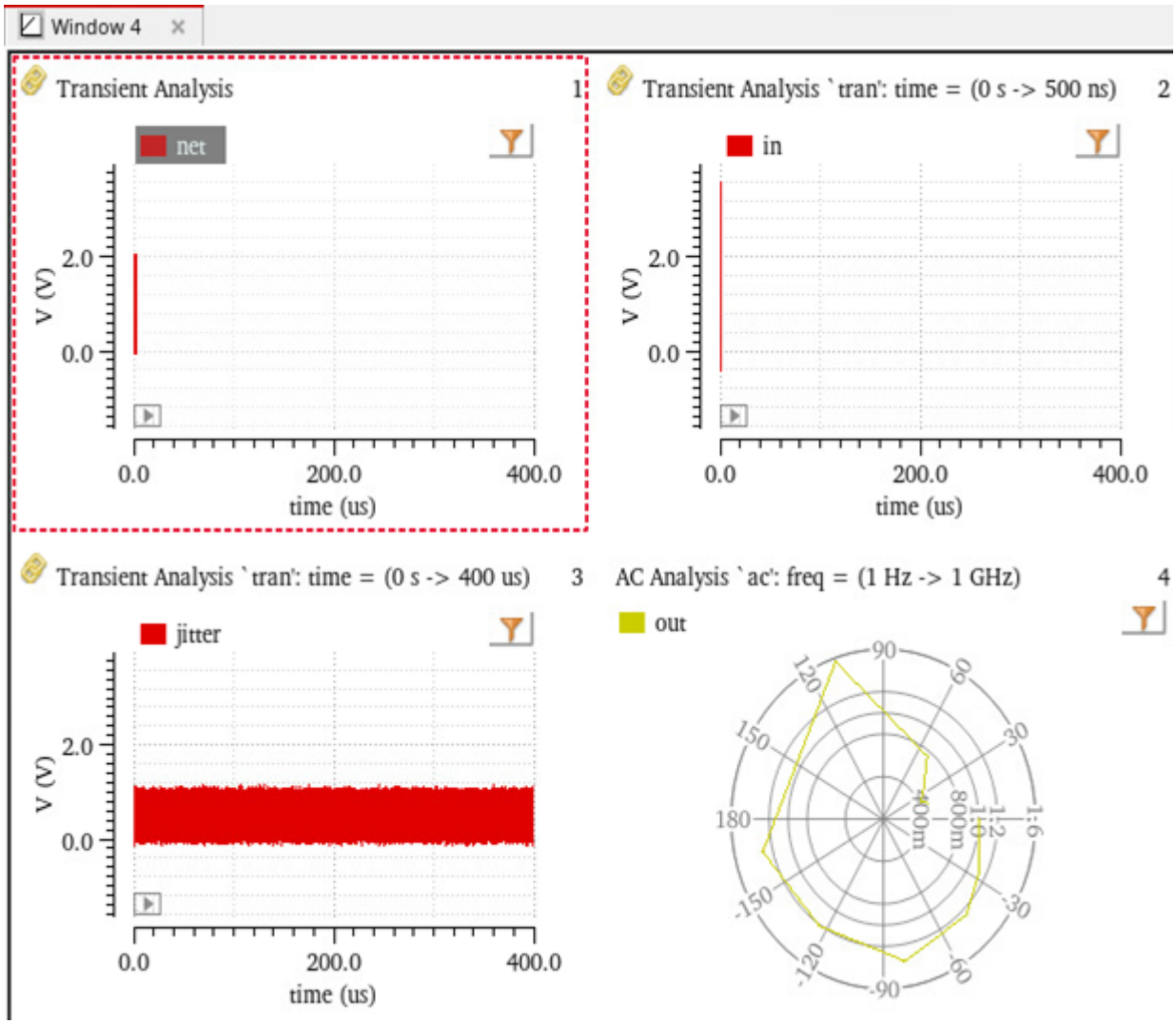


2. Repeat the first step for all the subwindows that you want to link.

© 2023 Cadence Design Systems, Inc.

To link all graphs in the subwindows of a graph window, first select any subwindow and choose *Graph – Link – All Graphs on Page*.

A link icon is displayed at the top-left corner of each linked subwindow. Note that the rectangular graphs in the subwindows `1`, `2`, and `3` have the same unit `us` for x axis and `voltage` for y axis. Also, the graph in the subwindow `4` is not linked because it is a polar graph.



Zoom operations on x and y axes performed on a subwindow are applied to all the linked subwindows.

© 2023 Cadence Design Systems, Inc.

Unlinking Graphs in Subwindows

To unlink graphs in the subwindows of a graph window:

➡ Select a subwindow that you want to unlink and choose *Graph – Link – Break*.

To unlink all graphs in the subwindows of a graph window, first select any subwindow and choose *Graph – Link – Break All Links*.

Linking Axes in Subwindows

By default, both x and y axes of graphs are linked when you link graphs in subwindows. However, you can choose which axes you want to selectively link.

To selectively link axes:

1. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Graph – Link – Link Options*.

   The Link Options form opens.

2. Select the *Link Dependent (Y) Axes* check box to link only y axes in the linked graphs.

   All y axes that have the same units are linked.

3. Select the *Link Independent (X) Axes* check box to link only x axes in the linked graphs.

   All x axes that have the same units are linked.

4. Select the *Only Link Axes with These Units* check box and specify the base units of the axes in the text field if you want to link only axes with the specified base units. You can specify multiple base units as a list of comma-separated values. For example, `V,A,Hz,s`.

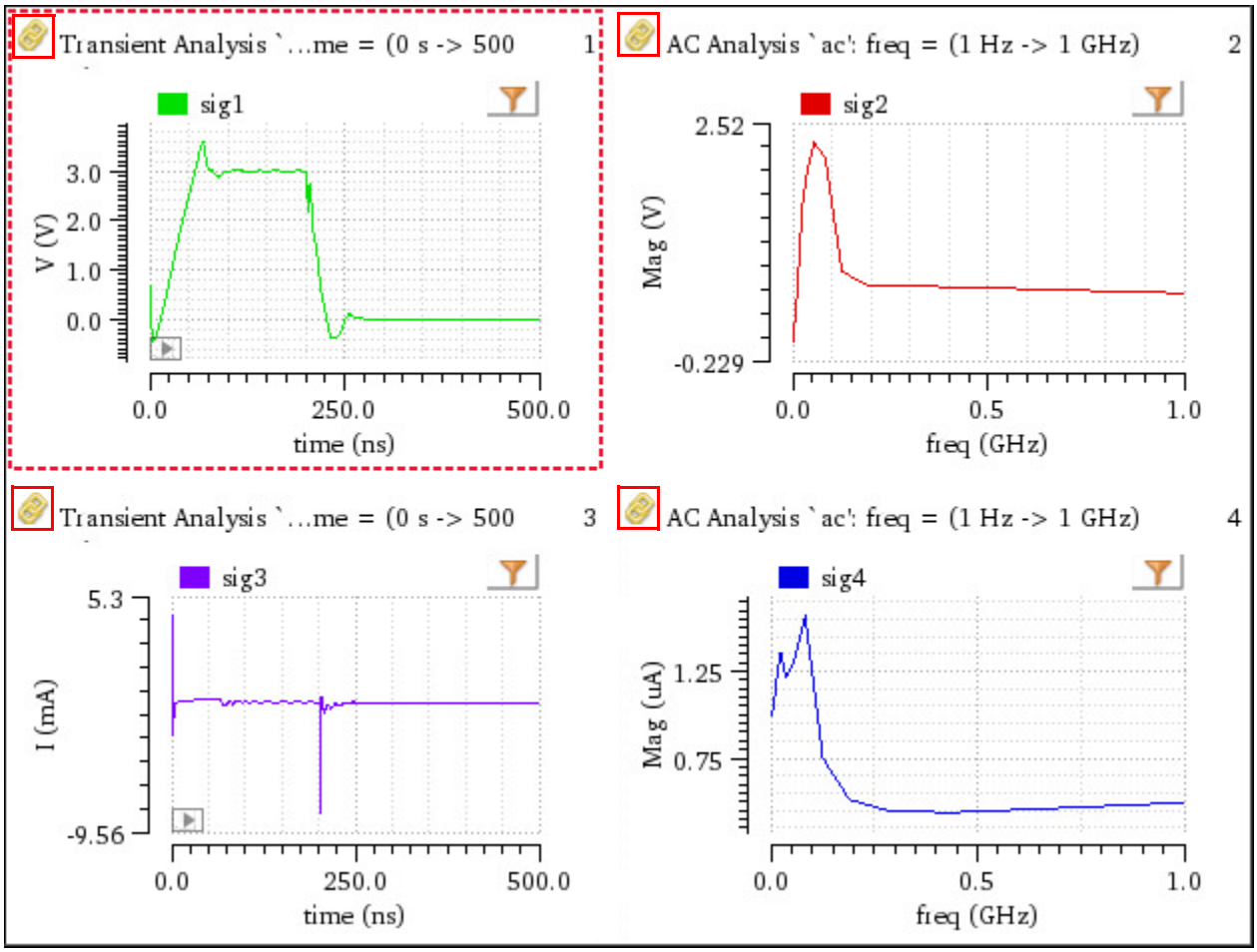
   This option is helpful when there are multiple units available for both dependent and independent axes and you want to link only the axes which have specific units.

Example

Consider the following graph window in which four waveforms `sig1`, `sig2`, `sig3`, and `sig4` are plotted in the subwindows `1`, `2`, `3`, and `4`, respectively.

© 2023 Cadence Design Systems, Inc.

Note that all subwindows of the graph window are linked together.



- For dependent (y) axes:
  - Graphs in subwindows `1` and `2` have the same base unit, `V`, for dependent (y) axes.
  - Graphs in subwindows `3` and `4` have the same base unit, `A`, for dependent (y) axes.
- For dependent (x) axes:
  - Graphs in subwindows `1` and `3` have the same base unit, `s`, for independent (x) axes.
  - Graphs in subwindows `2` and `4` have the same base unit, `Hz`, for independent (x) axes.

© 2023 Cadence Design Systems, Inc.

The following table shows how dependent and independent axes of subwindows `1`, `2`, `3`, and `4` are linked when different combinations of options are selected in the Link Options form.

Link Dependent (Y) Axes	Link Independent (X) Axes	Only Link Axes with These Units	Result
Selected	Selected	Not selected	■ y axes of subwindows `1` and `2` are linked together. ■ y axes of subwindows `3` and `4` are linked together. ■ x axes of subwindows `1` and `3` are linked together. ■ x axes of subwindows `2` and `4` are linked together.
Selected	Not Selected	Not Selected	■ y axes of subwindows `1` and `2` are linked together because they have the same base unit `V` for their y axes. ■ y axes of subwindows `3` and `4` are linked together because they have the same base unit `A` for their y axes.
Selected	Not selected	Selected and base unit `A` is specified in the text field.	Only y axes of subwindows `3` and `4` are linked together because only they have base unit `A` for their y axes.

© 2023 Cadence Design Systems, Inc.

Link Dependent (Y) Axes	Link Independent (X) Axes	Only Link Axes with These Units	Result
Selected	Not selected	Selected and base unit `V` is specified in the text field.	Only y axes of subwindows `1` and `2` are linked together because only they have base unit `V` for their y axes.
Selected	Not selected	Selected and base units `V`,`A` are specified in the text field.	■ y axes of subwindows `1` and `2` are linked together because they have the same base unit `V` for their y axes. ■ y axes of subwindows `3` and `4` are linked together because they have the same base unit `A` for their y axes.
Not selected	Selected	Not selected	■ x axes of subwindows `1` and `3` are linked together because they have the same base unit `s` for their x axes. ■ x axes of subwindows `2` and `4` are linked together because they have the same base unit `Hz` for their x axes.
Not selected	Selected	Selected and base unit `s` is specified in the text field.	Only x axes of subwindows `1` and `3` are linked together because only they have base unit `s` for their x axes.

© 2023 Cadence Design Systems, Inc.

Link Dependent (Y) Axes	Link Independent (X) Axes	Only Link Axes with These Units	Result
Not selected	Selected	Selected and base unit `Hz` is specified in the text field.	Only x axes of subwindows `2` and `4` are linked together because only they have base unit `Hz` for their x axes.
Not selected	Selected	Selected and base units `s,Hz` are specified in the text field.	■ x axes of subwindows `1` and `3` are linked together because they have the same base unit `s` for their x axes. ■ x axes of subwindows `2` and `4` are linked together because they have the same base unit `Hz` for their x axes.

Related Topics

linkDepAxes

linkIndepAxes

useLinkAxesUnitFilter

linkAxesUnitFilterSet

Customizing a Graph

After you have created a graph, you can customize it to analyze the graph data.

This section contains the following topics:

■ Determining the Active Window on page 190

■ Setting the Graph Colors on page 190

■ Renaming and Closing Window on page 191

© 2023 Cadence Design Systems, Inc.

- Handling Graph Objects on page 191
- Panning and Zooming Graphs on page 192
- Editing Graph Properties on page 196
- Setting Properties for Multiple Graphs on page 199

Determining the Active Window

The active window tab appears white and the inactive window tabs appear gray. When you click a window tab, that window becomes active and the tab color changes to white.

The video Using Subwindows in Qt Graph demonstrates how to change the graph layout and how to plot signals in different subwindows.

Setting the Graph Colors

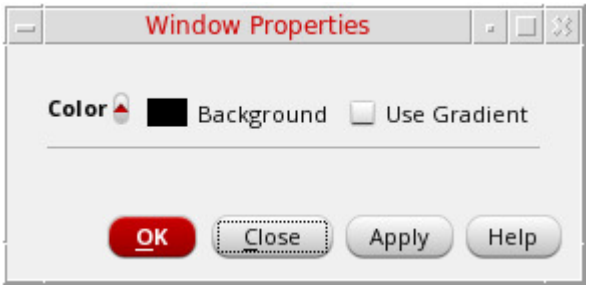
The default color scheme for graphs is determined by the `viva.graphFrame background` variables in the `.cdsenv` file. The default color scheme is as follows:

- Black when you open the graph by using Virtuoso Analog Design Environment (ADE) mode or in the stand-alone SKILL mode

To change the background color of a window, do the following:

1. In the window, choose *File – Window Properties.*

   The *Window Properties* form appears.



2. In this form:

   a. Click the *Background* button to set the background color of the selected window.

   b. Select the *Use Gradient* check box to add a gradient to the background, which means the background color fades at the bottom of the graph. By default, this check box is disabled.

© 2023 Cadence Design Systems, Inc.

You can also change the background color of a graph by setting the following environment variable:

```
envSetVal("viva.graphFrame" "background" 'string "<color>")
```

Note: When you change the background color of the graph, Virtuoso Visualization and Analysis XL automatically adjusts the color contrast of various graph objects, such as traces, markers, labels, and tracking cursor to make them clearly visible on the graph.

Renaming and Closing Window

To rename a window, double-click the window tab and type the new name.

To close the window, close the window tab by clicking the cross button.

Handling Graph Objects

This section describes how you can select and delete a graph or its components, such as traces, axes, markers, and labels.

Selecting Objects

Click the graph or its component, such as trace, marker, or label, to select it. You can select multiple objects by holding down the `Ctrl` key while you click the required graph objects.

Deleting Objects

You can delete the objects, such as graphs, labels, markers, legends, and traces. You can also delete a window or a subwindow.

To delete an object, do the following:

1. Select the object you want to delete.
2. Choose *Edit – Delete*, or press the `Delete` key.

   The object selected in the window is deleted.

To delete all objects, select an object or a subwindow and do one of the following:

- Choose *Edit – Delete All.*
- Press `E`.

© 2023 Cadence Design Systems, Inc.

To delete all markers in a window, choose *Marker – Delete All* or press `Ctrl+E`.

To delete all traces in a window, choose *Trace – Delete All* or press `Shift+E`.

Panning and Zooming Graphs

You can pan and zoom a graph by using the pan bar and scroll bar displayed in the graph window.

This section covers the following topics:

- Panning a Graph on page 192
- Zooming a Graph on page 193
- Zooming In a Trace Along X- and Y-Axis on page 194
- Panning And Zooming Graph With Mouse on page 195

Panning a Graph

You can use the pan bar located at the top of the window to pan a graph. When you plot a signal in the graph, the width of the pan bar is adjusted so that the entire graph is visible. Resize the pan bar by dragging either end of the pan bar inward to view the required portion of a graph. The selected portion of the graph is zoomed in to display greater detail. When the size of the pan bar is less than maximum, drag the pan bar to the left or right to view portions of the graph that are currently outside the display area. You can also click anywhere in the pan bar area to move the pan bar to that location.

Alternatively, to pan a graph, do one of the following:

- Right-click anywhere in a graph and choose *View – Pan Left / Pan Right / Pan Up / Pan Down*.
- Press the arrow keys to pan the graph in the required direction.
- Hold the `Ctrl` and `Alt` keys simultaneously. The mouse pointer changes to a hand symbol. Now, you can pan the graph by using the mouse left button.

Note: You can hide the pan bar by de-selecting the *Display zoom bar* check box in *Graph Options* tab of the Graph Properties form.

© 2023 Cadence Design Systems, Inc.

Zooming a Graph

The Virtuoso Visualization and Analysis XL tool supports multiple zooming operations.

To zoom in or out a graph, choose one of the following options from the *View* menu or click the relevant button on the zoom toolbar.

- *ZoomIn by 2*

  Zooms in the graph by a factor of two. A vertical scroll bar appears on the right side to help view graph areas outside the current window. To move horizontally to view areas outside the window use the pan bar displayed on the top of the window.

- *ZoomOut by 2*

  Zooms out of the graph by a factor of two.

- *Fit*

  Fits the graph in the window. Alternatively, right-click the trace and choose *Fit Trace* to fit the graph in the window. You can also use the bindkey `f` to perform the zoom fit.

- *Previous*

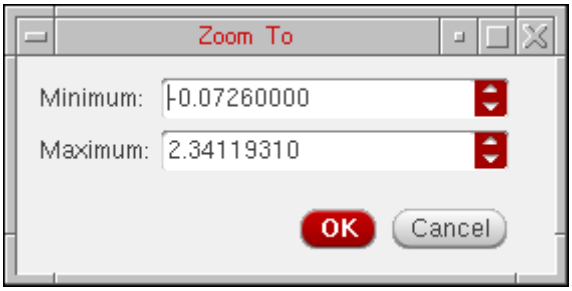
  Incrementally reverses a series of zoom and pan actions.

- Next

  Incrementally undoes the effect of the *Previous* command.

Alternatively, to pan and zoom in or out a graph, do the following:

- To pan a graph, right-click anywhere on the pan bar and choose one of the following options:
  - *Zoom to*—Zooms out the graph according to the specified size of the pan bar. When you select this option, the *Zoom To* form appears. In this form, select the maximum and minimum values for the pan bar.



  - *Zoom in X2*—Zooms in the graph in X direction by a factor of 2.

© 2023 Cadence Design Systems, Inc.

- *Zoom out X2*—Zooms out the graph in X direction by a factor of 2.
- *Fit X*—Fits the graph to the X-axis.

■ To zoom in or out a graph, right-click anywhere on the scroll bar that is displayed at the right and choose one of the following options:

Note: Scroll bar appears only if you zoom out the graph or strip.

- *Zoom to*—Zooms out the graph according to the specified size of the scroll bar. When you select this option, the *Zoom To* form appears. In this form, select the maximum and minimum sizes for the scroll bar.
- *Zoom in X2*—Zooms in the graph in Y direction by a factor of 2.
- *Zoom out X2*—Zooms out the graph in Y direction by a factor of 2.
- *Fit Y*—Fits the graph to the Y-axis.
- *Fit Y to Visible X*—Fits the visible part of the trace to Y-axis based on the X-axis . This option is specific to an individual strip. This option finds the minimum and maximum Y-axis values that are visible in a strip and then performs a Y-axis zoom of those Y values.

Zooming In a Trace Along X- and Y-Axis

To zoom in a trace along X- and Y-axis, hold down the mouse button and drag the pointer to select the area on the graph that you want to zoom in. When you release the mouse button, the area you selected is zoomed in.

To zoom in the trace along one axis, do the following:

1. Press bindkey `X` to zoom in the graph along the X-axis or press bindkey `Y` to zoom in the graph along the Y-axis.
2. Hold down the right mouse button and drag the pointer to select the graph area that you want to zoom in.

   After you release the mouse button, the zoom is complete and the right mouse button zoom is reset to XY zoom, which means you can now zoom in or out the graph along both the axes.

Note: You cannot zoom in the graph by using the left mouse button. Also, ensure that you zoom the area toward the right of the Y-axis; otherwise, a shortcut menu appears.

© 2023 Cadence Design Systems, Inc.

Tip

If you have multiple strips in a window that you want to zoom in, place the pointer on the left or right edges of the strip container to start the zoom.

Panning And Zooming Graph With Mouse

To pan a graph with the help of mouse, perform the following steps:

1. Hold down the `Ctrl` and `Alt` keys simultaneously. Notice that the mouse pointer is changed to a hand symbol, which indicates that you pan the graph now.
2. Drag the mouse pointer to pan the graph in left or right direction.

Note: If a graph includes multiple strips, the panning procedure is performed on all the strips at the same time.

To zoom in or out a graph or a strip in Y direction with the help of mouse, do the following:

➡ Hold down the `Ctrl` key and move the mouse wheel button upward or downward.

If you move the wheel button upward, the selected strip is zoomed in, and if you move the wheel button downward, the selected strip is zoomed out. After the graph is zoomed in, a vertical scroll bar appears on the right side of the strip that you can use to view the remaining portion of the trace in the graph. You can also view the complete trace by using the mouse wheel button.

To zoom in or out all the strips in a graph in X direction, do the following:

➡ Hold down the `Shift` key and move the mouse wheel button.

When you move the wheel button upward, the selected strip is zoomed in, and if you move the wheel button downward, the selected strip is zoomed out.

To zoom in in or out all the strips in a graph in both X and Y direction, do the following:

➡ Hold down the `Ctrl` and `Shift` keys and move the mouse wheel button.

Video

The video Panning and Zooming Qt Graph demonstrates how you can pan and zoom the Qt Graph.

Note: The zoom operations that you perform on the graph are added to a zoom stack and you can switch between the various zoom levels by selecting zoom commands in the View toolbar. The zoom stack can store maximum of 100 zoom operations.

© 2023 Cadence Design Systems, Inc.

Consider the scenario in which multiple traces are plotted in different strips in the graph window and the graph window size is small such that scrollbars appear on the graph. Now, if you zoom in the trace in one of these strips a few times and then use the mouse scroll wheel to zoom out the within that strip, the trace in the strip zooms out until it is fit and after that mouse scroll wheel starts scrolling through the outer scrollbar of the graph window.

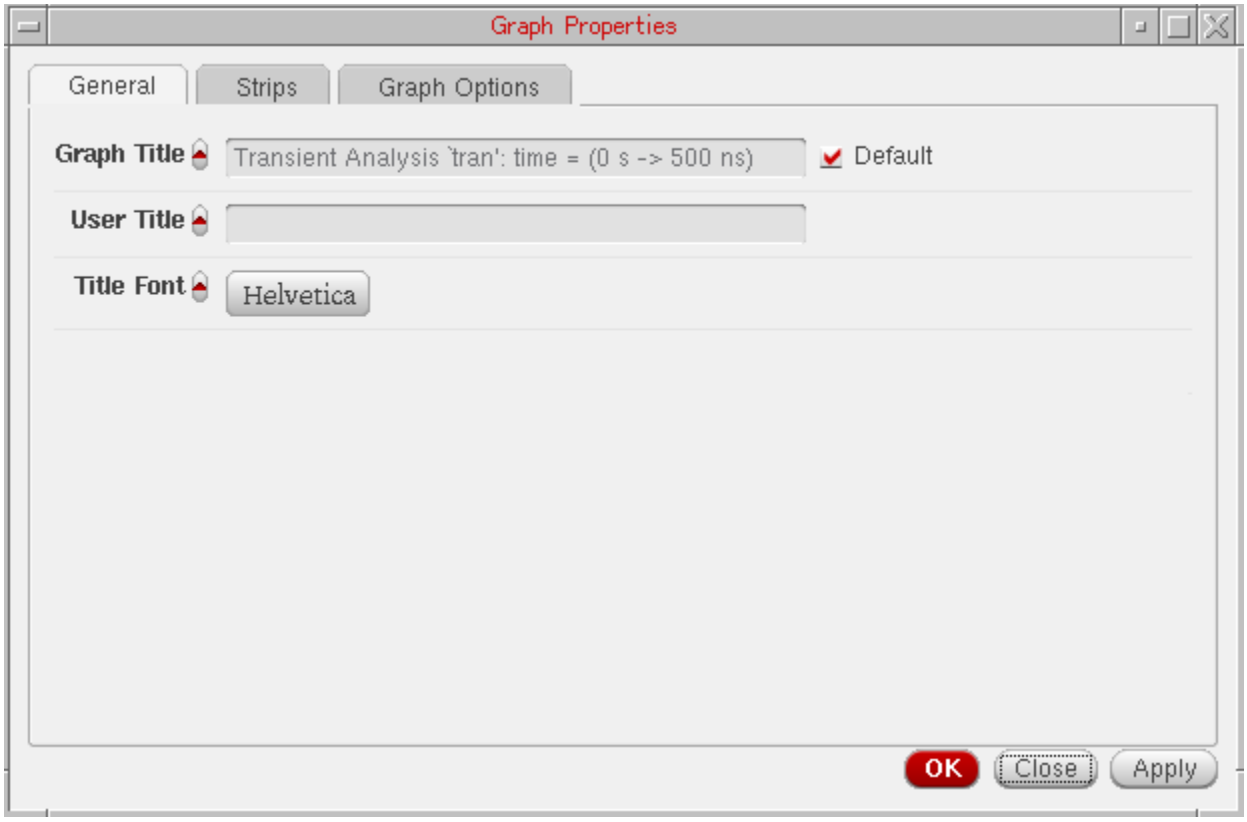
Editing Graph Properties

To set the properties of a graph, do one of the following:

- Choose *Graph – Properties.*
- Right-click anywhere in the window and choose *Graph Properties*.
- Double-click anywhere in the trace legend area.
- Press the bindkey `Shift+Q`.

  The *Graph Properties* from appears. This form includes three tabs—*General, Strips,* and *Graph Options.*

  Note: Double-click in the graph area does not open the graph properties form.



© 2023 Cadence Design Systems, Inc.

On the *General* tab, specify the following values:

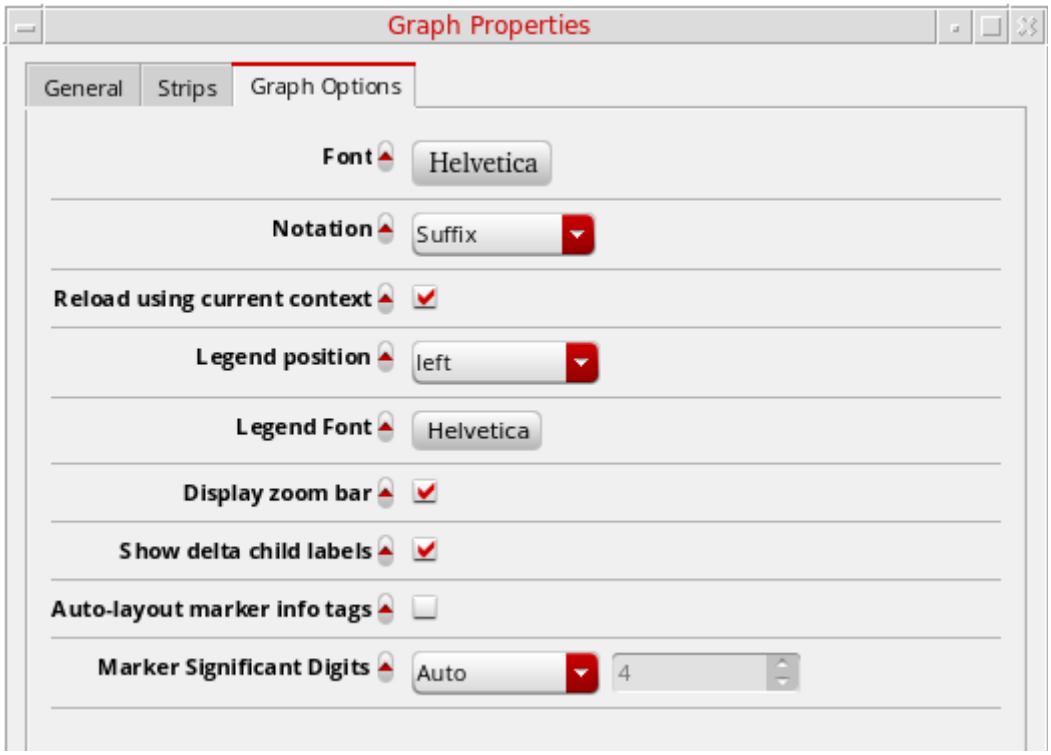
- *Graph Title*—The title of the graph that is displayed at the top of the graph. When you select the *Default* check box next to this field, you cannot edit the graph title or provide a new graph title. The default graph name includes the name of the analysis and the Y-axis name.

  Note: In addition, to edit the graph title directly in the graph window, double-click the title. The title becomes editable and you can specify a new title.

- *User Title*—A name for the window. You can edit this field if the default check box is not selected. Specify a name and click *OK*, this name is saved, and the *Graph Properties* form title is displayed as *Graph Properties for <graph-name>* when you open the form next time.

- *Title Font*—The font properties for the window title.

For information about the fields on the *Strips* tab, see Setting Strip Properties on page 407.



On the *Graph Options* tab, specify the following values:

- *Font*—The font properties for the graph and its components, such as labels, axes, and markers.

© 2023 Cadence Design Systems, Inc.

- *Notation*—The graph notation that can be `Scientific`, `Engineering`, and `Suffix`.
  Default value: `suffix`.
- *Reload using current context*—When this check box is selected, signals in the graph are reloaded according to the data from the current in-context results directory. If this check box is not selected, signals are reloaded based on their individual databases. This check box is selected by default.
- *Legend position*—Sets the trace legend position as `left`, `inside`, or `above`. Default value: `left`.
- *Legend Font*—Sets the font type of the signal names in the trace legend.
- *Display zoom bar*—Shows or hides the zoom bar displayed on the current window or subwindow.
- *Show delta child labels*—Select this check box to hide the labels for all point markers that are used to form a delta marker.
- *Marker Significant Digits*—Set the marker significant digits to `Auto` or `Manual`. If you select `Manual`, you need to provide the number of significant digits.

■ Click *OK*.

Note: A red button is displayed with each form field in the properties form for graph and graph objects, which acts as a toggle switch. By default all the form fields are opened in the edit mode. If you click this button, you cannot edit the form fields.

Editing the Graph Title

To edit a graph title, double-click anywhere in the graph title area. The mouse pointer changes into a cursor. You can now delete the existing grapg title and type the new title. While editing, you can also use the keyboard arrow, `Home`, or `End` keys.

© 2023 Cadence Design Systems, Inc.

Setting Properties for Multiple Graphs

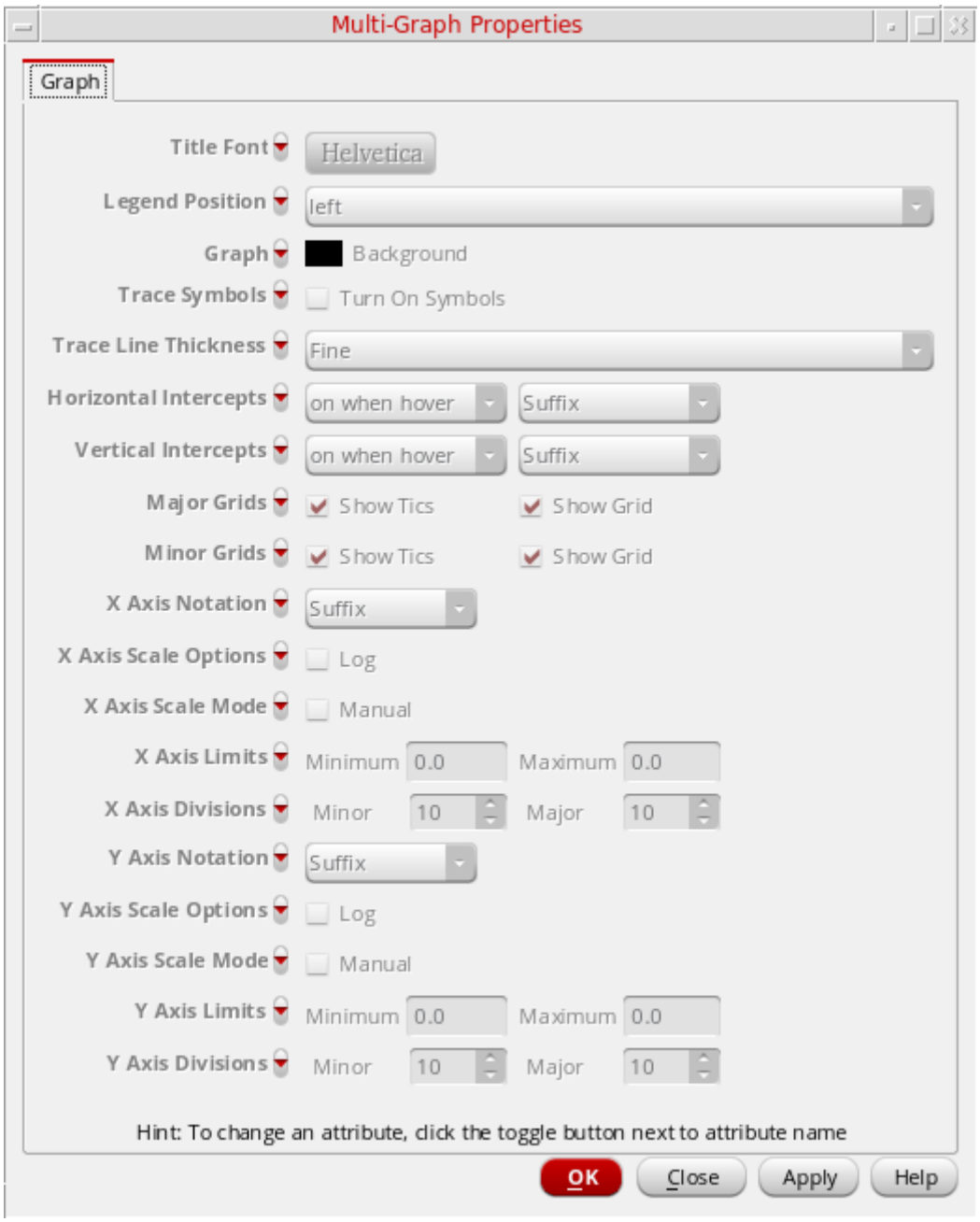
Virtuoso Visualization and Analysis XL helps you change the properties of graphs plotted in different windows in a single step. This is useful when you open multiple graphs in several windows and want to apply the same settings in all the opened graphs.

To set the properties of multiple graphs plotted in different windows,

➡ Choose *Session – Multi-Graph Properties*.

© 2023 Cadence Design Systems, Inc.

The Multi-Graph Properties form appears. All the fields of this form are disabled by default. To activate and modify the field value, click the toggle button displayed next to the field name.



This form includes the following fields that you can set for all the graphs:

- *Title Font*—Set the font style for the graph title.

© 2023 Cadence Design Systems, Inc.

- *Legend Position*—Specify where to display the trace legend on the graph. The available options are `left`, `inside`, and `above`.
- *Graph*—Set the graph backgound color. The default color is black.
- *Trace Symbols*—Specify whether to display symbols on the trace plotted in the graphs.
- *Trace Line Thickness*—Specify the line thickness.
- *Horizontal Intercepts*—Specify when to display the horizontal intercepts on the trace. The available drop-down values are `on`, `off`, `on when hover`. The default value is `on when hover`. In addition, specify the notation of intercept values (suffix, engineering, and scientific).
- *Vertical Intercepts*—Specify when to display the vertical intercepts on the trace. The available drop-down values are `on`, `off`, `on when hover`. The default value is `on when hover`. In addition, specify the notation of intercept values. (suffix, engineering, and scientific).
- *Major Grids*—Select the *Show Tics* check box to enable tics on the major grids. Select the *Show Grid* check box to display the major grid. By default, these check boxes are selected.
- *Minor Grids*—Select the *Show Tics* check box to enable tics on the minor grids. Select the *Show Grid* check box to display the minor grid. By default, these check boxes are selected.
- *X Axis Notation*—Set the notation of the independent axis. The available values are `Suffix`, `Engineering`, and `Scientific`. The default value is `Suffix`.
- *X Axis Scale Options*—Select the *Log* check box if you want to use the logarithmic scale for the X-axis.
- *X Axis Scale Mode*—Select the *Manual* check box if you want to manually specify the scale for X-axis.
- *X Axis Limits*—Set the X-axis minimum and maximum limits in the *Minimum* and *Maximum* fields respectively.
- *X Axis Divisions*—Specify the X-axis minor and major divisions in the *Minor* and *Major* fields respectively.
- *Y Axis Notation*—Set the notation of the dependent axis. The available values are `Suffix`, `Engineering`, and `Scientific`. The default value is `Suffix`.
- *Y Axis Scale Options*—Select the *Log* check box if you want to use the logarithmic scale for the Y-axis.

© 2023 Cadence Design Systems, Inc.

- *Y Axis Scale Mode*—Select the *Manual* check box if you want to manually specify the scale for Y-axis.
- *Y Axis Limits*—Set the Y-axis minimum and maximum limits in the *Minimum* and *Maximum* fields respectively.
- *Y Axis Divisions*—Specify the Y-axis minor and major divisions in the *Minor* and *Major* fields respectively.

  When you click *Apply* or click *OK*, the specified graph, trace, and axis settings are applied to all the graphs plotted in different windows and subwindows.

Working with Graph Axis

This section covers the following sections to describe how to use X- and Y-axis while plotting and analyzing signals in a graph.

- Editing Graph Axis Attributes on page 202
- Changing Axes Scale to Logarithmic on page 206
- Displaying X-Axis Labels in String Format on page 207
- Adding Multiple Y-Axes on page 211
- Changing Dependent Axis (Y-Axis) on page 213
- Plotting Multiple Signals on a Common Axis on page 220
- Merging Two Y-Axes on page 222

Editing Graph Axis Attributes

The X-axis attributes set the attributes for the X-axis. The X-axis attributes provide a mechanism to create YvsY plots.

Note: Eye diagrams can also be plotted from the Calculator, while a limited version of YvsY plot is available from the Results Browser.

The default graph attributes are controlled by the values assigned to variables in the `.cdsenv` file. For more information, see Appendix A, “Virtuoso Visualization and Analysis XL Tool Environment Variables.”

You can edit the attributes of the axes by doing one of the following:

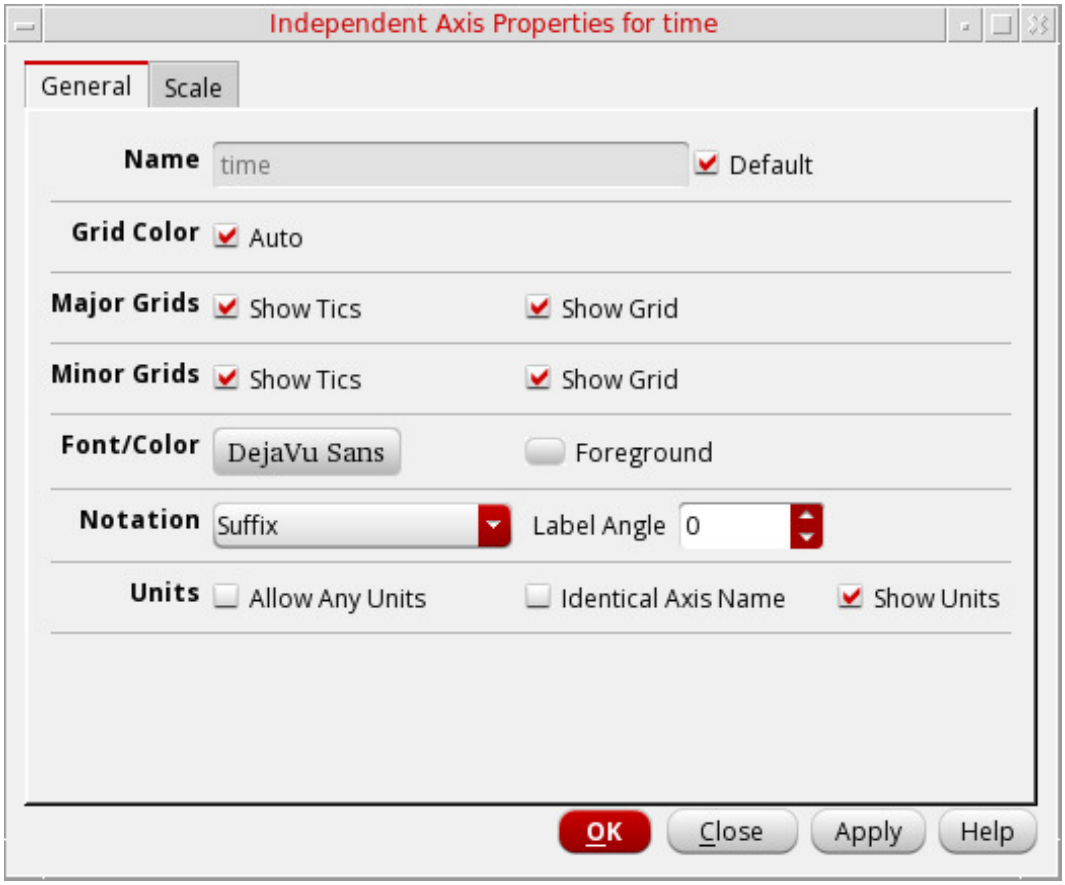
- Double-click an axis in the window.

© 2023 Cadence Design Systems, Inc.

- Select an axis and choose *Axis – Properties*. You can select more than one axis in the subwindows by using the `Shift` key.
- Right-click an axis and choose *Axis Properties*.

  The *Independent Axis Properties for <X-axis-name>* form appears for X-axis and *Dependent Axis Properties for <Y-axis>* appears for Y-axis.

  This form includes two tabs—*General* and *Scale*.



  On the *General* tab, specify the following values:

  - *Name*—The default name of the selected axis. You can change the name, if required. The changed axis name is displayed when you click *OK*. If you select the *Default* check box next to this field, you cannot change the axis name.
  - *Label with Axis Number*—Select this check box if you want to display the axis number in the selected axis name. This field is displayed only for the Y-axis (dependent axis) properties form.
    For more information about axis number, see Adding Multiple Y-Axes on page 211.

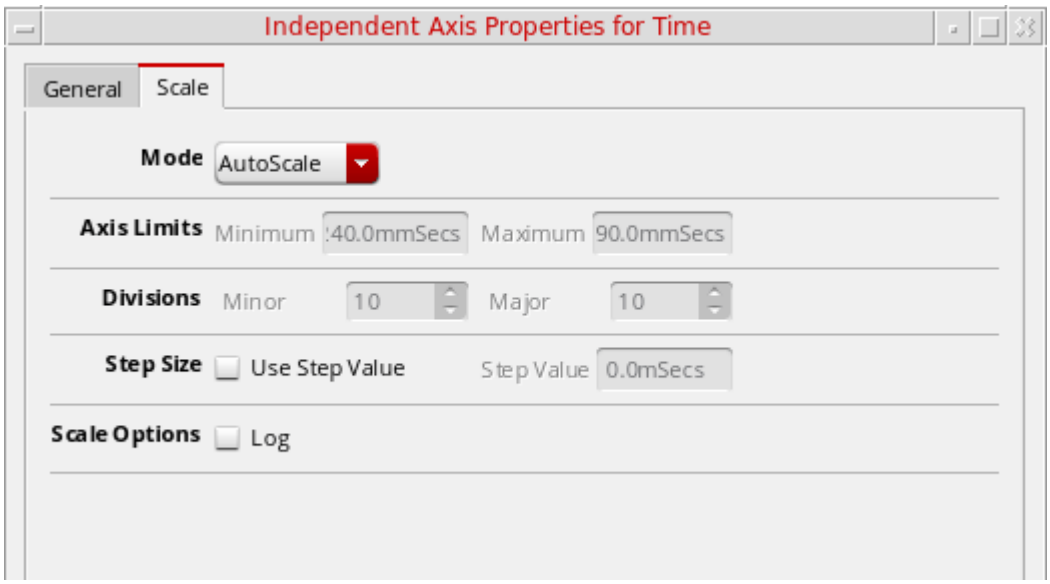
© 2023 Cadence Design Systems, Inc.

- *Grid Color*—Select to set the default grid color.
- *Major Grids*
  - *Show Tics*—Select to display the major axis divisions.
  - *Show Grid*—Select to display the major grids.
- *Minor Grids*
  - *Show Tics*—Select to display the minor grid divisions.
  - *Show Grid*—Select to display minor grids.
- *Font/Color*—The font of the axes labels and divisions.
- *Notation*—The notation displayed for the axis labels values. The available values are—`Engineering` and `Suffix`. Default value: `Suffix`.
- *Label Angle*—Angle through which axis labels are rotated. If the labels are lengthy, you can fit them better in the graph by rotating the label angle. The default label angle is `0`.

  Note: You can also specify the label angle of the independent axis by setting the labelAngle environment variable.
- *Units*—Controls the units displayed for the X-axis.
  - *Allow Any Units*—Plots traces irrespective of their axis units. If the axis unit of new trace does not match with the axis unit of the existing graph, the trace is by default plotted in a new window. However, if this check box is selected, the trace with any axis units can be plotted in the existing graph. By default this variable is not selected.
  - *Identical Axis Name*—Determines whether to plot a new trace in active window or a new window. By default, this check box is not selected, which means the signal is plotted in the active window irrespective of the X-axis name. When this check box is selected, the signal is plotted in the active window only if the X-axis units and names match.
  - *Show Units*—Select to display the axes units. Alternatively, you can display or hide the axes units by right-clicking the axes and choosing *Show Units*. By default this variable is not selected

© 2023 Cadence Design Systems, Inc.

On the *Scale* tab, specify the following values:



- *Mode*—The scaling mode as *AutoScale* or *Manual*.
- *Axis Limits*—The maximum and minimum range of the selected axis on which a signal can be plotted.

  Note: If you select the *AutoScale* mode, you cannot change the axis limits. To change the axis limit, select the *Manual* option. The minimum size supported by axes is `1e-24`. This covers the full range supported by Si suffixes.
- *Divisions*—Specify the minor and major axis divisions for the selected axis. If you selected the *AutoScale* mode, you cannot change the axis divisions. To change the axis divisions, select the *Manual* option.
- *Step Size*—Select the *Use Step Value* check box to specify a step value for major grids. This step value indicates the spacing between major grids on the graph.

  Note: If a negative value is specified in the *Step Value* field, it will be automatically converted to a positive value after you click the *Apply* button.
- *Scale Options*—Select the *Log* check box to display the axis in logarithmic scale.

■ Click *OK.*

© 2023 Cadence Design Systems, Inc.

Changing Digital Dependent Axis Properties

To change the properties of the digital dependent axis, right-click the trace and choose Digital Axis Properties. The Digital Dependent Axis Properties form appears. You can use this form to change the axis font and foreground color by setting the *Font/Color* field.

Changing Axes Scale to Logarithmic

To display the dependent or independent axes scale in logarithmic values, do one of the following:

- Right-click the axis and choose *Log Scale*.
- Select an axis and choose *Axis – Log Scale*.
- Select the *Log* check box under *Scale Options* in the Axis Properties form.

  The scale for the selected axis changes to Logarithmic. Now, if you drag another signal from the Results Browser and plots the signal in the same graph in append mode. The dragged signal is plotted on the logarithmic scale on X-axis.

© 2023 Cadence Design Systems, Inc.

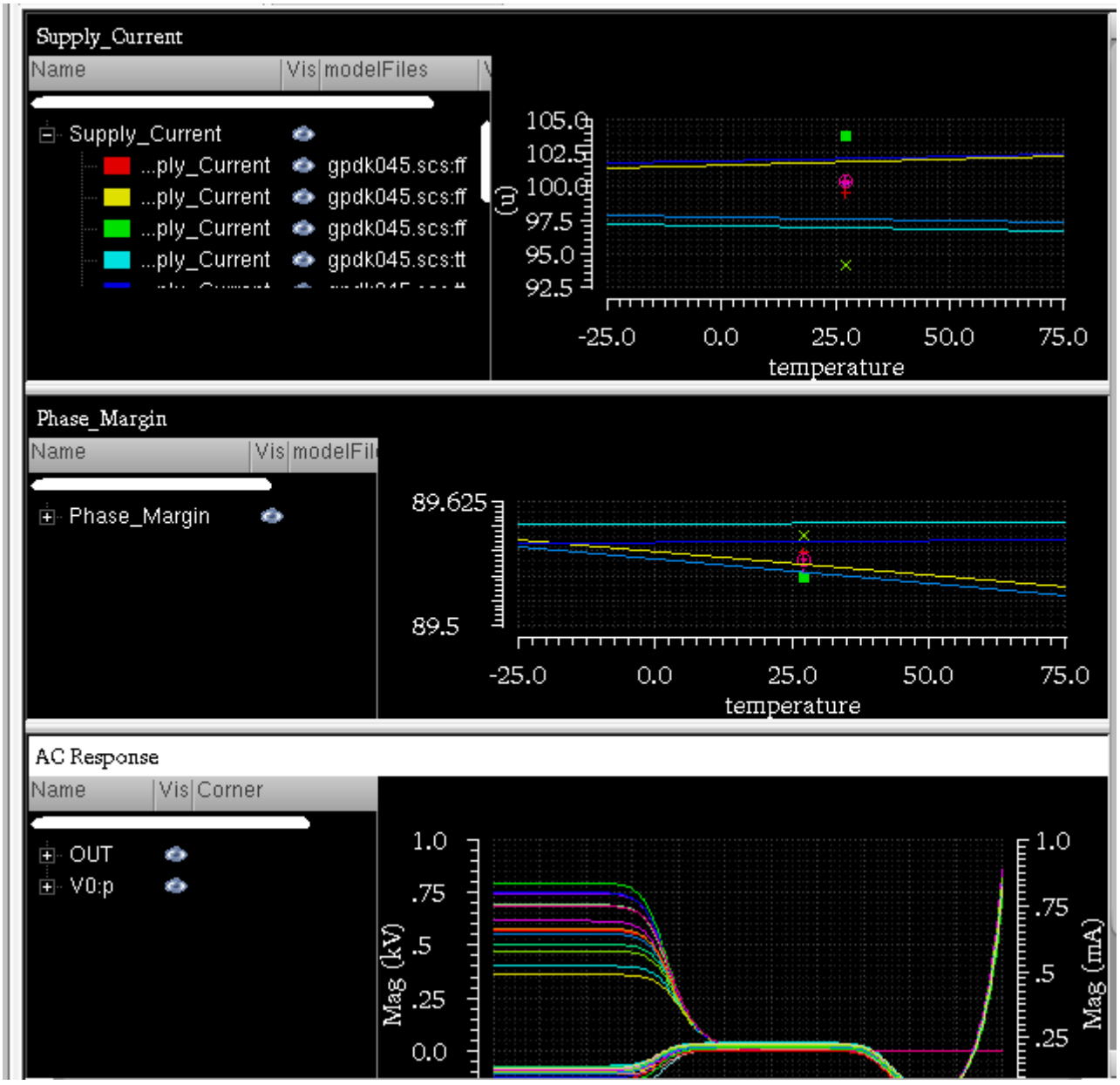
Displaying X-Axis Labels in String Format

If you select the X-axis variable as model file or Corner while plotting the results for a simulation run in ADE XL for sweep data, the labels on X-axis are displayed in string format.

In the figure below, you can see the simulation results plotted in the graph window. The sweep variables for this simulation are—*VDD*, *modelFiles*, and *temperature*. This simulation also contains corner values. After you run the simulation, the different outputs are listed in the *Output* section of ADE XL. When you plot all the outputs, the waveforms are plotted in individual subwindows. See the figure below. Notice that the plots shown in the figure below have temperature as the sweep variable on X-axis. You can change the X-axis variable to

© 2023 Cadence Design Systems, Inc.

modelFiles, *VDD*, or *Corner*. When you change the X-axis variable to *modelFiles* or *Corner*, the X-axis labels are displayed as string values.



Note: Labels with string values are not supported for transient data.

To change the sweep variable to *modelFiles* or *Corner*, do the following:

1. Right-click the X-axis and choose *Swap Sweep Var*.

   The *Swap Sweep Var* form appears.

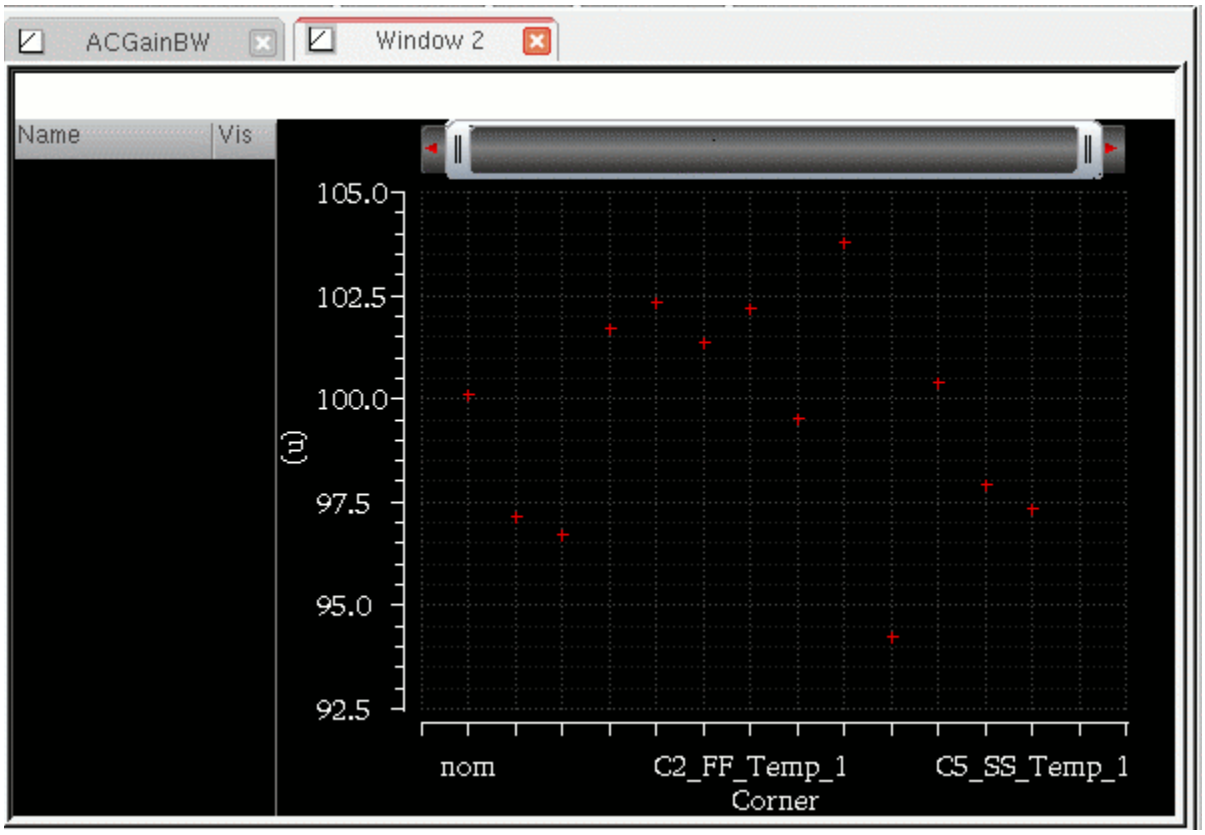
© 2023 Cadence Design Systems, Inc.

2. Select *modelFiles* or *Corner.*

   The graph is plotted with the selected variable displayed on X-axis.

   In the following figure, the sweep variable plotted on X-axis for `Supply_Current` plot, shown in the figure above, is changed to *Corner*. Notice that the X-axis labels for corners are displayed as string values and the trace is displayed as a sequence of points.

   Note: To display the trace as a continuous line, right-click the trace and choose *Type – Continuous line.*



Also, the string for all the intercept points may not be visible on the X-axis when the trace is displayed in its normal size. To view any specific string for a data point, you need to zoom in the graph. For information about how to zoom and pan a graph, see Panning and Zooming Graphs on page 192.

If the string is long, it is displayed as an elided string with the ... symbol, for example, `CO_VDD...Temp_1`. To view the complete string, place the pointer on a data point.

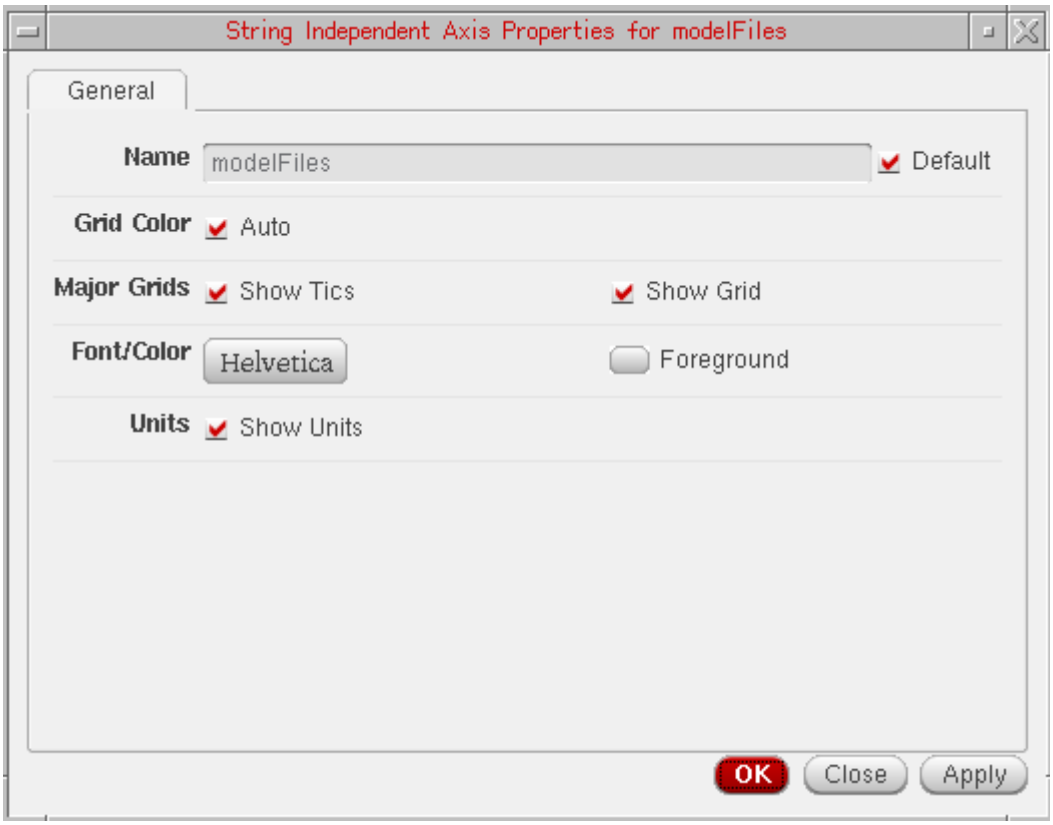
© 2023 Cadence Design Systems, Inc.

Changing Properties of the String Independent Axis

To change the properties of the independent axis (X-axis) that includes the string intercept values, do the following:

- Right-click the Independent axis and choose *Axis Properties*.
- Select the independent axis and choose *Axis – Properties*.

The *String Independent Axis Properties for <axis name>* form appears.



This form has only the *General* tab, which includes the following fields:

- *Name*—The default name of the selected axis. You can change the name, if required. The changed axis name is displayed when you click *OK*. If you select the *Default* check box next to this field, you cannot change the axis name.
- *Grid Color*—Select to set the default grid color.
- *Major Grids*
  - *Show Tics*—Select to display the major axis divisions.

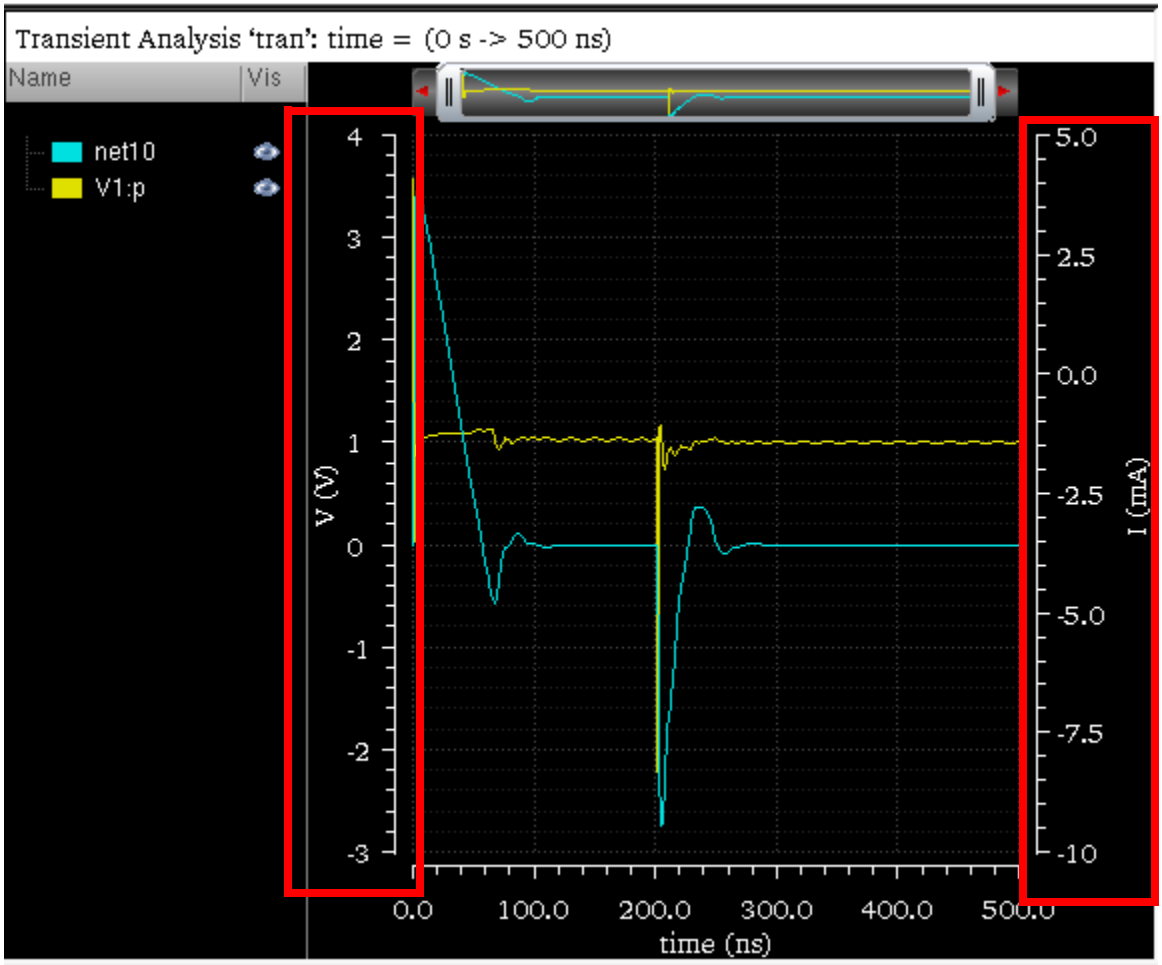
© 2023 Cadence Design Systems, Inc.

- *Show Grid*—Select to display the major grids.

- *Font/Color*—The font of the axes labels and divisions and foreground color.
- *Units*—Select to display the axes units. Alternatively, you can display or hide the axes units by right-clicking the axes and choosing *Show Units*.

Adding Multiple Y-Axes

From Results Browser or ADE, if you plot two or more signals that contain different Y-axis (dependent axis) data in the same window, the graph displays separate Y-axes for both the signals. For example, when the voltage (`net10`) and current (`V1:p`) signals are plotted in the same graph, the graph displays two Y-axes, displayed on the left and the right of the graph respectively, as shown in figure below:



© 2023 Cadence Design Systems, Inc.

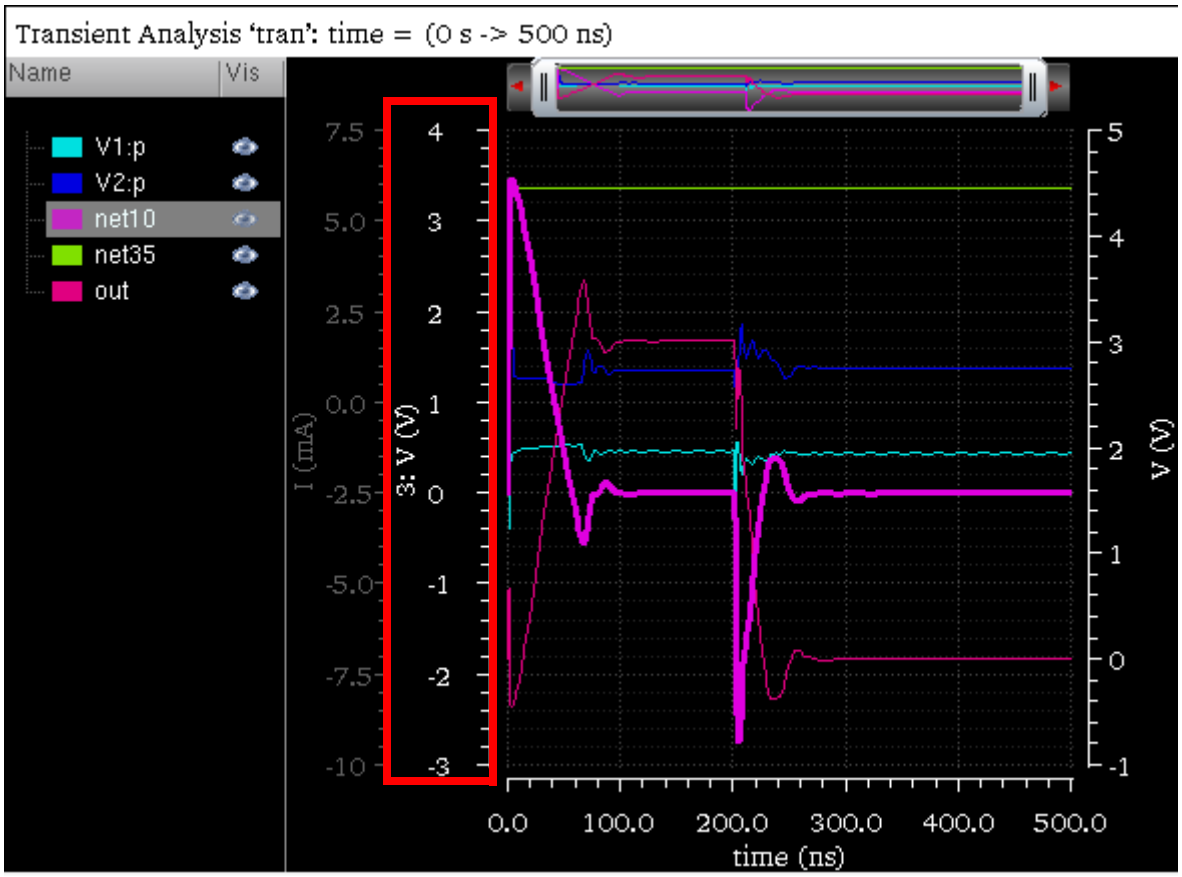
Note: A graph or a strip can have four Y-axes at the maximum. If the graph already contains four axes and you plot a trace that requires a new Y-axis to be added to the graph, then the new trace is plotted in a new strip.

When you have more than one trace plotted in a graph and you want to analyze a particular trace, you can move the selected trace to a new Y-axis. You can also change the Y-axis of a trace to another existing Y-axis if the graph contains two or more Y-axes. To know more about how to change the axis of a trace, see Changing Dependent Axis (Y-Axis) on page 213.

To assign a new Y-axis to the selected trace:

➡ Right-click the trace and choose *Change Y Axis – New*.

A new Y-axis is added in the graph and the selected trace is detached from the existing axis and is attached to the new axis. For example, the figure below contains five traces (two voltage and three current) plotted in a graph. When you assign a new Y-axis to the trace for the `net10` signal, the trace moves to a new Y-axis (`3: V(V)`) displayed on the left of the graph. The new axis is the third Y-axis in the graph; therefore, the axis number is 3.



© 2023 Cadence Design Systems, Inc.

Important

If an axis does not have any traces attached to it, the axis is removed from the graph.

When you add a new Y-axis for a trace, the new axis name is displayed in the following format:

```
axis_number: axis_title(axis_unit)
```

For example, the above figure displays the following axis name for the new axis that you have added manually to the `net10` trace.

```
3: V(V)
```

where,

- `axis_number` is `3` because this is the third Y-axis in the graph
- `axis_title` is `V`, which indicates this is a voltage signal
- `axis_unit` is `V`, Volts

By default, axis name displays the `axis_number`. To hide `axis_number` from the axis name, do one of the following:

- Right-click the axis and de-select the *Show Axis Number* check box.
- In the *Dependent Axis Properties* form, on the *General* tab, deselect the *Label with Axis Number* check box and click *OK*. To know how to open the *Dependent Axis Properties* form, see Editing Graph Axis Attributes on page 202.
- Choose *Axis – Axis Number*.

  The `axis_number` is removed from the selected axis name.

Changing Dependent Axis (Y-Axis)

If you have more than one Y-axis in a graph, you can change the Y-axis of the trace to another Y-axis. You can assign a common axis to the traces that have the same signal type. However, you cannot assign the same axis to signals of different data types. For example, a voltage signal cannot be assigned an axis of the signal representing current.

To change the Y-axis of a selected trace:

➡ Right-click the trace and choose *Change Y Axis – Move to axis_name*.

The shortcut menu displays the names for all the Y-axes that are currently visible in the graph. The name of the axis to which the trace is currently attached is disabled.

© 2023 Cadence Design Systems, Inc.

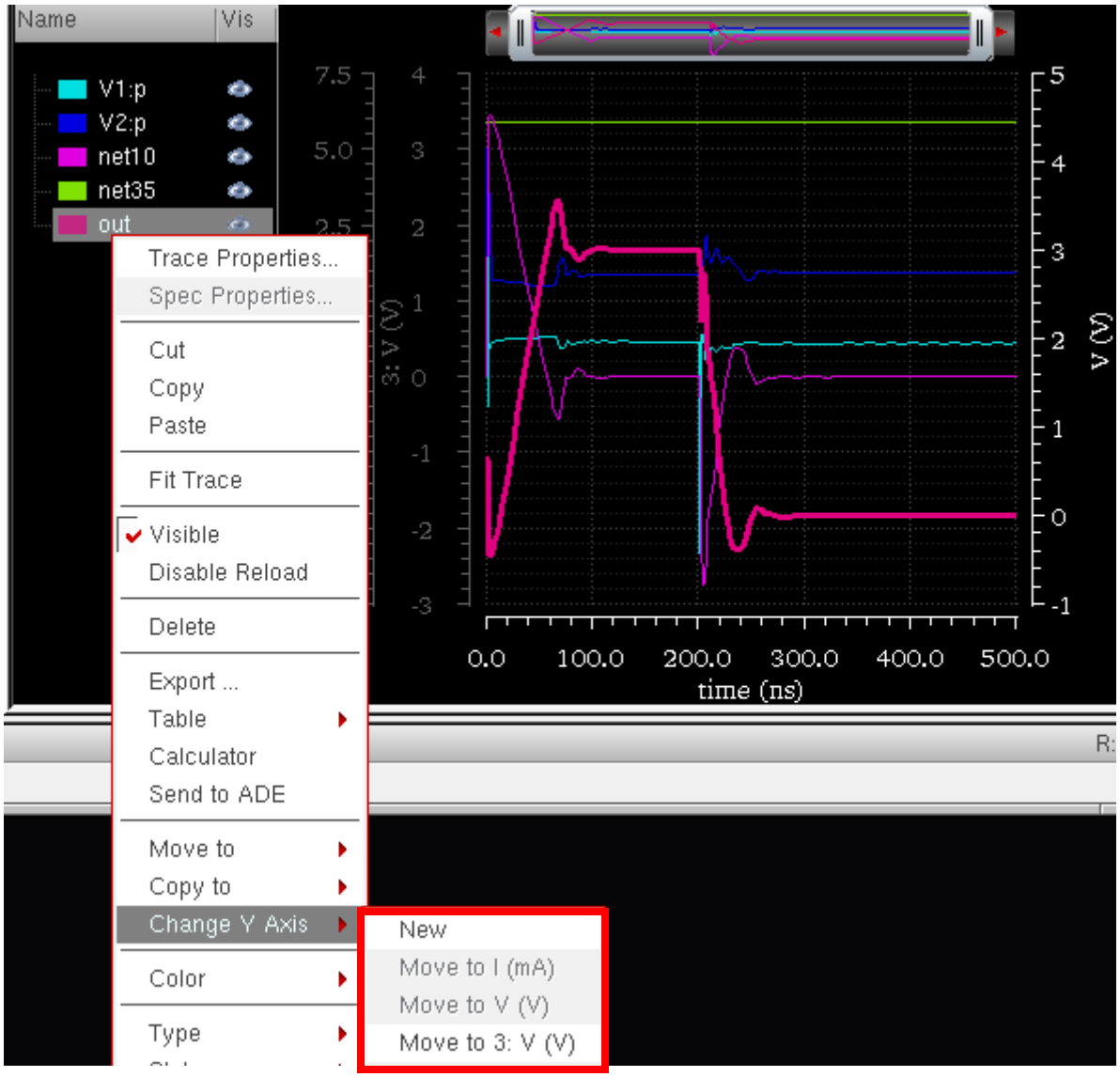
For example, in the following figure, the graph includes five traces:

- `V1:p`—Current signal plotted on `I(mA)` axis
- `V2:p`—Current signal plotted on `I(mA)` axis
- `net10`—Voltage signal plotted on `3:V(V)` axis
- `net35`—Voltage signal plotted on `V(V)` axis
- `out`—Voltage signal plotted on `V(V)` axis

The `net10`, `net35`, and `out` signals are the voltage signals. Therefore, you can move these signals to any axis that represents the voltage signal. In this example, you can move the `out` signal to only `3: V(V)` axis. The `I(mA)` axis is disabled in the shortcut menu because it is

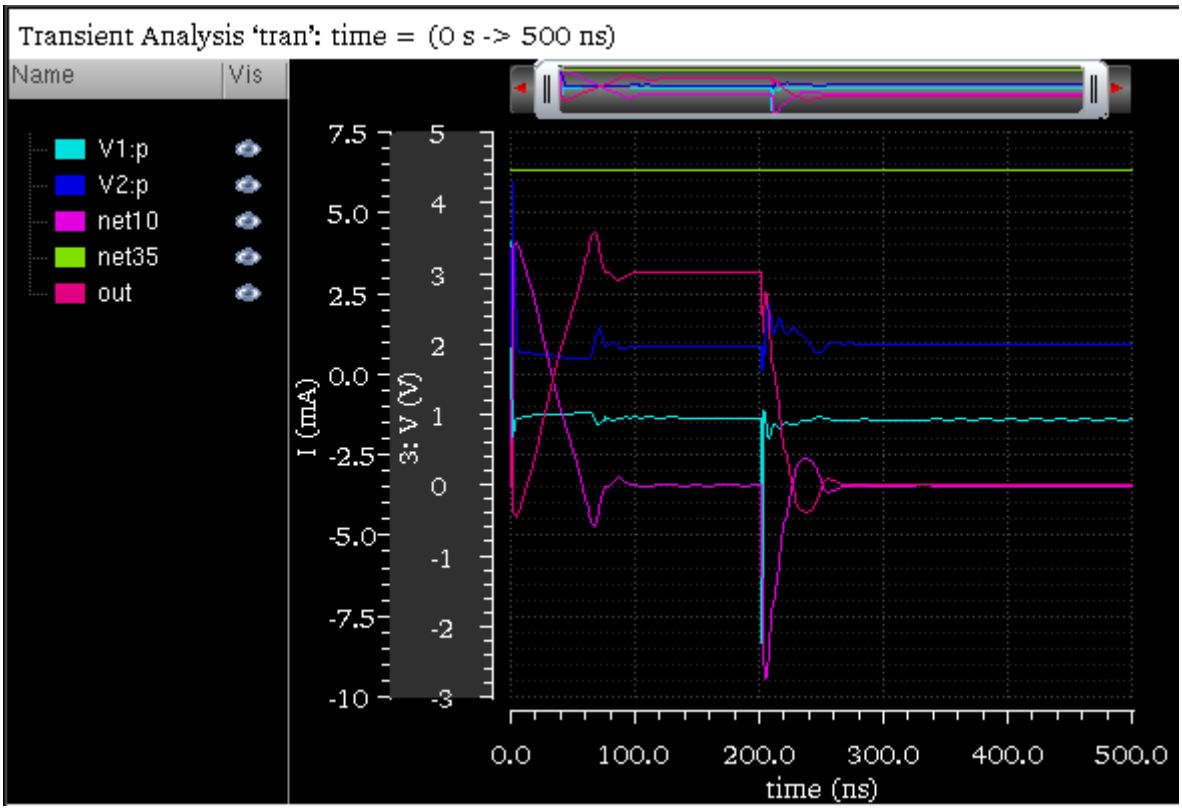
© 2023 Cadence Design Systems, Inc.

incompatible with the `out` signal. The `V(V)` axis is disabled because the `out` signal is already assigned to this axis.



© 2023 Cadence Design Systems, Inc.

When you move both the `out` and `net35` signals to the `3: V(V)` axis, the `V(V)` axis is removed from the graph because no signal is attached to this axis (see the figure below).



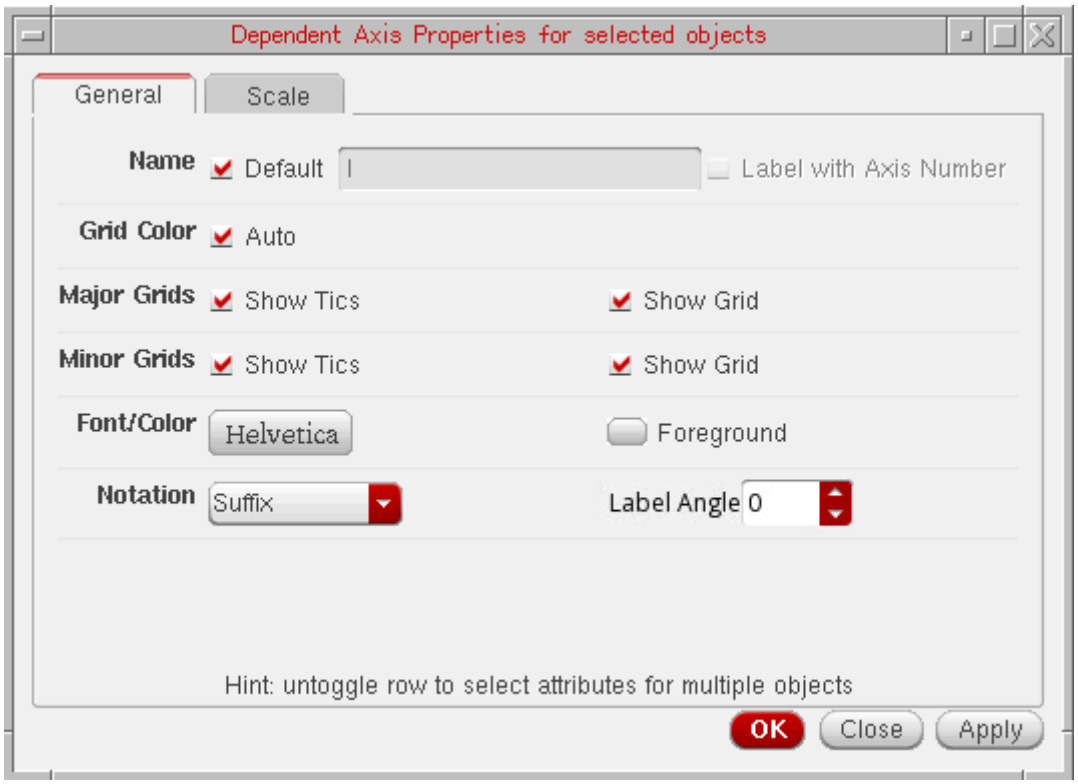
To find which traces are assigned to a particular axis, right-click the axis and choose *Select Attached Traces*. The traces attached to the selected axis are highlighted in the graph. Also, when you select a trace, only the axis for the selected trace is highlighted and all the other axes are dimmed.

To change the properties of all the dependent axis at the same time, do the following:

- Press the `Ctrl` key and click the axes for which you want to change the axis properties.
- Choose *Axis – Properties*.

© 2023 Cadence Design Systems, Inc.

The *Dependent Axis Properties for selected objects* form appears.



For more information about the properties form fields, see Editing Graph Axis Attributes on page 202.

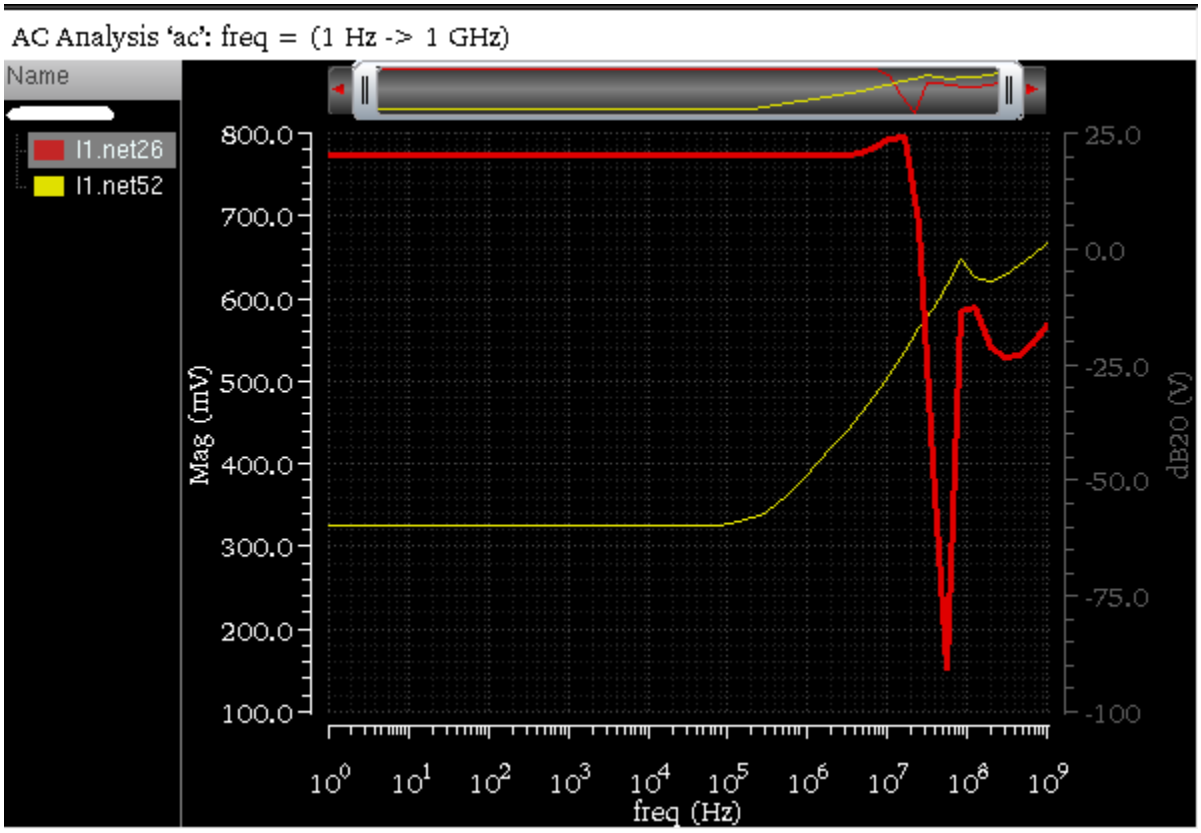
Note: You can specify the label angle for dependent axis by setting the labelAngle environment variable.

Plotting Traces Using Different Modifiers

You can plot two signals with different modifiers in the same graph along different Y-axes. In the example below the `l1.net26` signal is plotted with `Magnitude` as dependent modifier

© 2023 Cadence Design Systems, Inc.

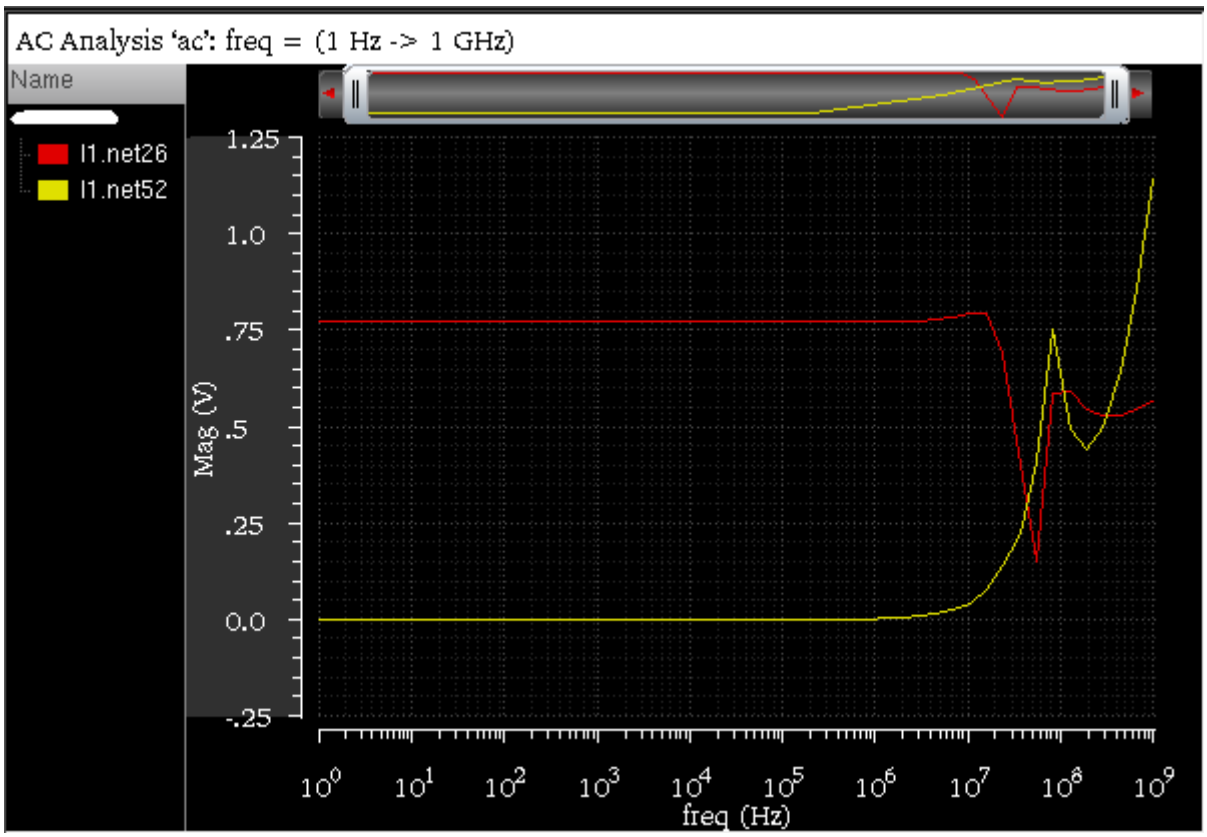
and `l1.net52` signal is plotted with `dB20` as the dependent modifier, the traces are plotted along different Y-axes in the same window.



By default, the Y-axis label displays the name of the modifier, such as `Mag (mV)`. You can change the Y-axis label by using the Axis Properties form.

© 2023 Cadence Design Systems, Inc.

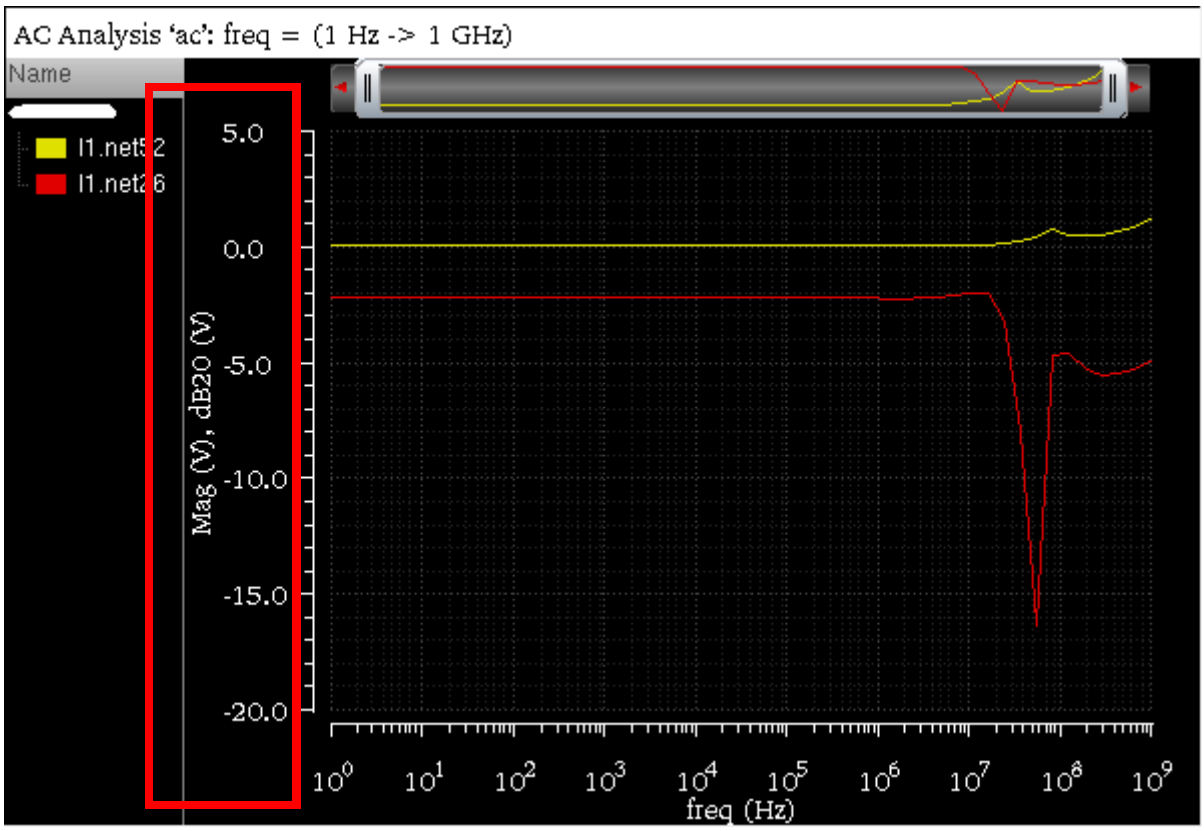
Now, if you change the dependent modifier for trace `l1.net52` to `Magnitude`, the trace is moved to the existing Y-axis `Mag (V)`.



When you plot these two signals in the same graph with `Magnitude` as dependent modifier and then change the dependent modifier value of signal `l1.net26` to `dB20`. In this case,

© 2023 Cadence Design Systems, Inc.

both the signals remain plotted on the same Y-axis with label, `Mag (V), dB20 (V)` (as shown in figure below).



Plotting Multiple Signals on a Common Axis

When you plot signals with different Y-axis values, the signals are plotted on different Y-axes in the same graph. Similarly, when you plot signals with different X-axis values, the signals are plotted in different windows. However, there are some situations where you may need to plot signals with different Y- or X-axis data on a common Y- or X-axis. Following are a few examples where you require plotting signals on the same Y-or X-axis:

- Plotting phase noise and AM component of noise on the same Y-axis
- Generating Stability plots—Plotting dB and phase of the same signal on a common Y-axis for easier measurement
- Plotting measurement expressions with different, but equivalent units on the same Y-axis
- Plotting results from two different simulators on the same X-axis
- Plotting simulated and measurement data on the same Y- and/or X-axis.

© 2023 Cadence Design Systems, Inc.

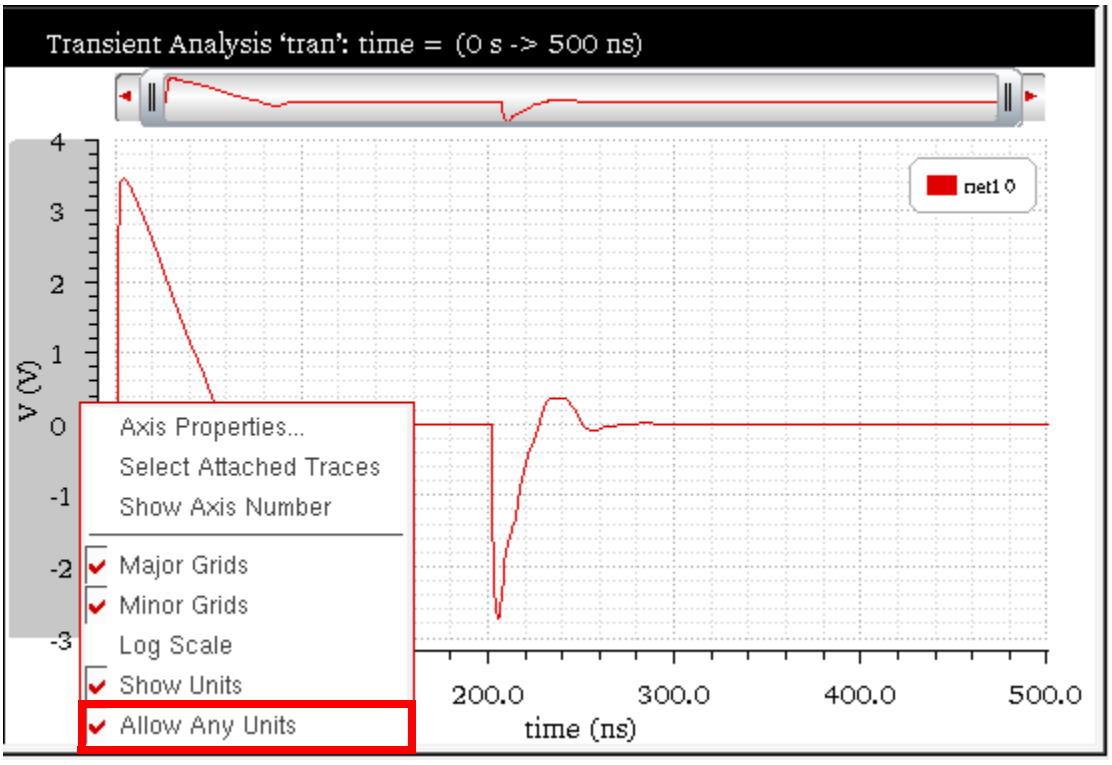
To plot multiple signals with different Y- or X-axis values on the same Y- or X-axis, you can do one of the following:

- Right-click Y- or X-axis and choose *Allow Any Units*.
- On the *Scale* tab of the Y- or X-axis Properties form, select *Units – Allow Any Units* check box.

  Now, when you plot signals with different Y- or X-axis values in the same graph, they are plotted on the same Y- or X-axis. After the signals are plotted, the Y-axis title changes to `Y(*)` and the X-axis title changes to `X(*)`.

Example

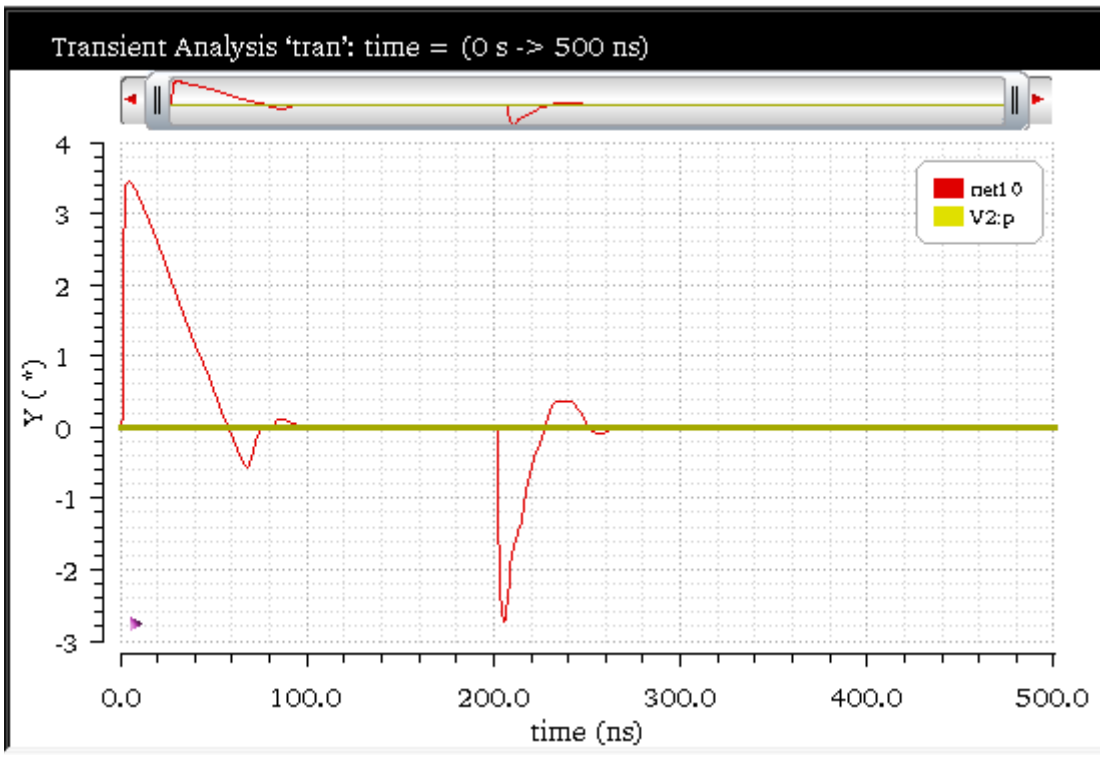
The following figure displays a voltage signal, `net10`, plotted on a graph. When you right-click the Y-axis, `V(V)`, on this graph and choose *Allow Any Units*, the Y-axis becomes flexible to plot any signal.



Now, in append mode, if you plot a current signal, `V2:p`, in the same window, the signal is plotted based on the existing Y-axis values, as shown in the figure below. In this figure, the Y-

© 2023 Cadence Design Systems, Inc.

axis label has changed to `Y(*)`, which shows that you can plot signals of different Y-axis values on this graph.



When you do not select the *Allow Any Units* option and drag traces on the graph that has different X- or Y-axis units, a context-sensitive message appears in the status bar, which suggests you to either change the axis units or select the *Allow Any Units* option. Similar context-sensitive messages are displayed when the pointer is moved during other drag-and-drop situations in which dragging of traces is not feasible. These messages vary according to the type of data, intended drop location, and settings of graph and axis onto which the data is dropped.

Merging Two Y-Axes

If you have two or more Y-axes present in a graph to display signals from different Y-axis data, you can merge these two Y-axes and plot the signals along one Y-axis. To do so, perform the following step:

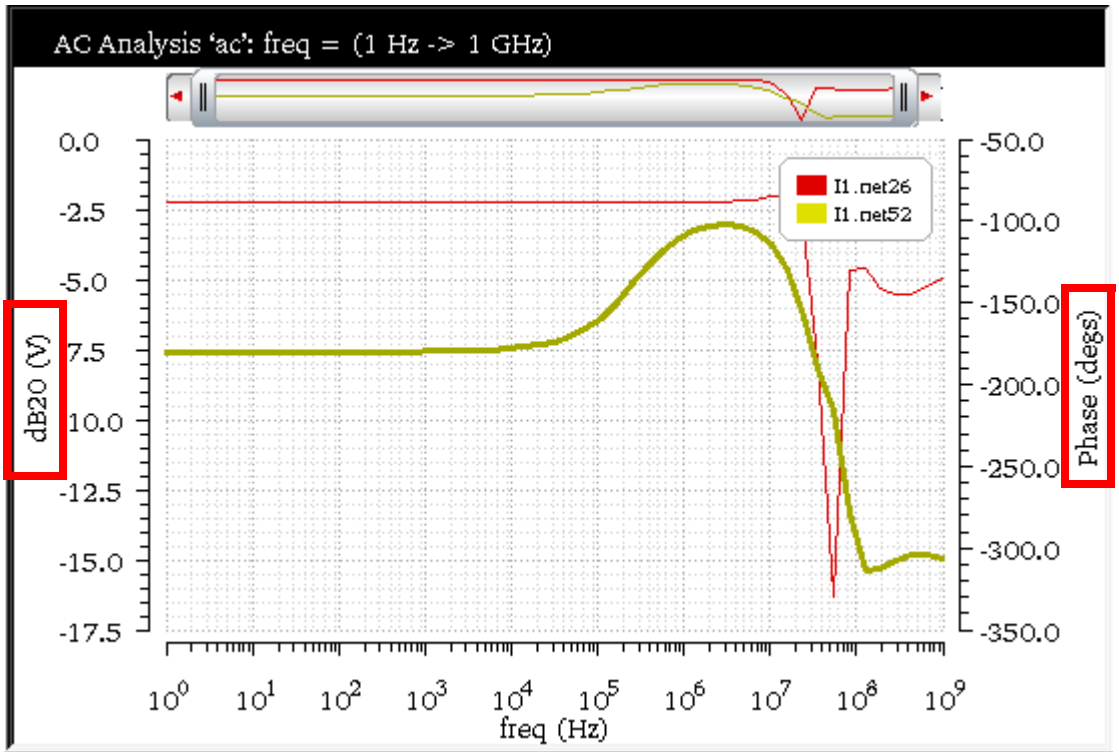
➡ Right-click the trace for which you want to merge the axis and choose the target axis-name with which you want to merge.

Now, when you plot a signal from Results Browser, Calculator, or ADE, the signals are plotted on the merged Y-axis.

© 2023 Cadence Design Systems, Inc.

Example

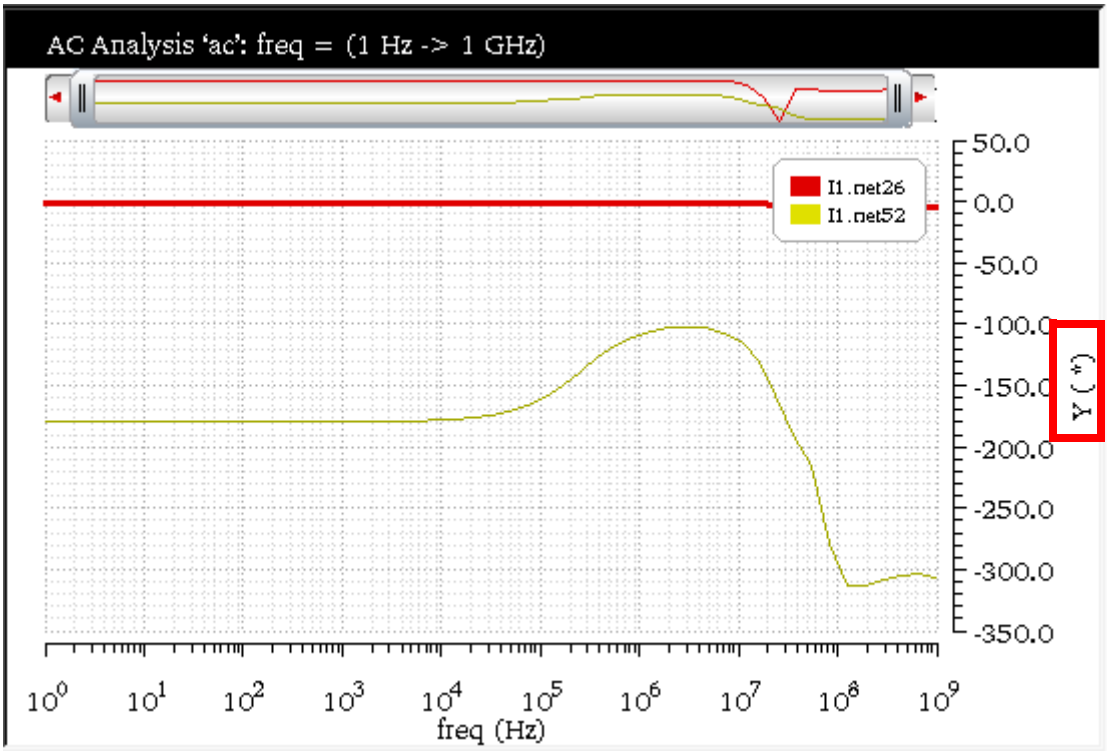
The following figure displays the `net26` and `net52` signals plotted on two Y-axes, `dB20` and `phase`, respectively.



Now, if you want to change the axis of the `net26` signal, right-click the trace for the `net26` signal and choose *Change Y-axis – Move to phase(deg)*. The `net26` signal is now plotted based on the units on the selected axis. The Y-axis title has been changed to `Y(*)`, which

© 2023 Cadence Design Systems, Inc.

indicates that you can plot any signal on this axis. Also, the *Allow Any Units* check box for this axis is always selected when the two axes are merged.



Locking Graphs

You can lock a graph to ensure that it does not change even when the simulation results in the data directory are reloaded. No signal can be added or removed from a locked graph. If you try to append a signal to a locked graph, it is plotted in a new subwindow. The graph operations like zooming and adding or moving markers are supported in a locked graph.

To lock a graph:

- In the window, choose *Graph – Lock*.

  A lock icon appears on the top-right corner of the window indicating that the graph is locked

To lock the size of a strip:

- Right-click anywhere in the strip and choose *Lock Strip Size*. When you view the menu again, you see a red check mark displayed next to the option, indicating that the strip size is locked.

© 2023 Cadence Design Systems, Inc.

A lock icon appears on the upper-left corner of the window tab indicating that the strip is locked.

Evaluating Graph Expressions

The signals plotted in the window can have expressions associated with them. How expressions are evaluated in the window depends on whether you are in the SKILL mode.

SKILL Calculator

When you open a saved graph that contains expressions, the expressions are evaluated by default within the context of the current results directory.

For example, if you plot an expression from a results directory and save the graph and later select a different results directory. When you open the saved graph again, the expression in the graph is evaluated in the context of the new results directory.

In the ADE mode, you can set the `ignoreTokenContext` variable to `false` so that expressions from the saved graphs are evaluated in the context of the results directory in which the graph was saved.

Working With Assistants

The Virtuoso Visualization and Analysis XL includes the following assistants:

- Spectrum Assistant on page 226
- Browser Assistant on page 240
- Marker Toolbox Assistant on page 241
- Eye Diagram Assistant on page 242
- Horiz Marker Table Assistant on page 307
- Trace Info Assistant on page 307
- Vert Marker Table Assistant on page 308
- Transient Measurement Assistant on page 308
- Direct Measurements Assistant on page 321
- Customize Trace Groups Assistant on page 359

© 2023 Cadence Design Systems, Inc.

- Working with Workspaces on page 365

Note: You can hide or show assistants by using the `F11` key. Alternatively, click *the Toggle Assistants Visibility* button on the Workspace toolbar.

Spectrum Assistant

The Spectrum assistant is used to plot and calculate the Fast Fourier Transform (FFT) of a periodic waveform and its different measurements—Signal-to-Noise-and-Distortion Ratio (SINAD), Spurious Free Dynamic Range (SFDR), Effective Number of Bits (ENOB), and Signal-to-Noise Ratio (SNR without distortion) ENOB, SINAD, SNR, SFDR, THD, sigpower, thddb, totalharmpower, peakharmpower, snb, snrh, dcpower—for a given input signal. The spectrum measure is used for characterizing A-to-D converters and is typically supported for transient simulation data.

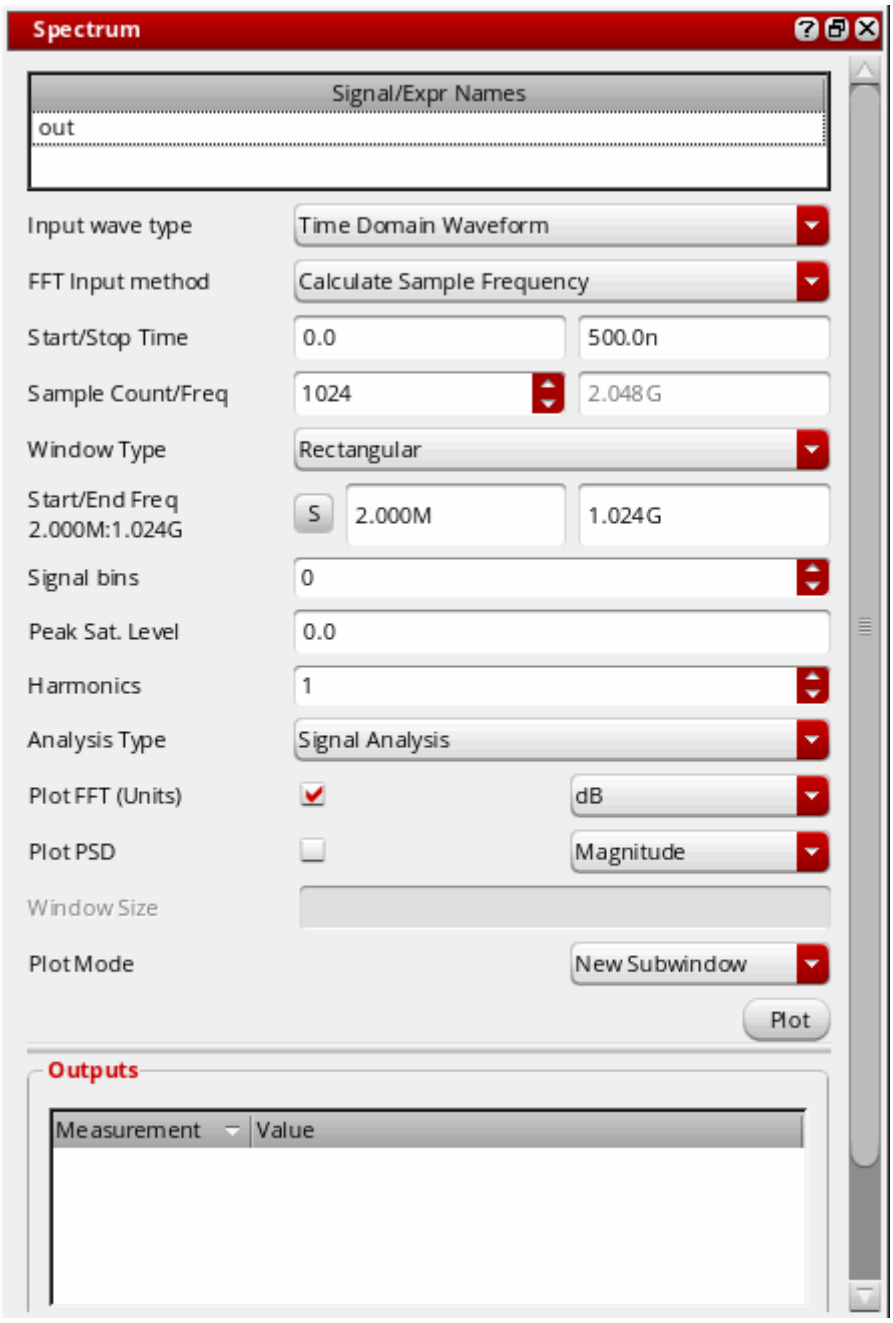
Note: In Virtuoso Visualization and Analysis XL, Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) are the same.

To open the Spectrum assistant, select a signal in the window and do one of the following:

- Choose *Window – Assistants – Spectrum*.
- Choose *Measurements – Spectrum.*

© 2023 Cadence Design Systems, Inc.

The Spectrum assistant appears on the right in the window by default.



The *Signal/Expr Names* field in the Spectrum assistant displays the name of the selected traces. The traces are displayed on the basis of their selection order; however, you can rearrange the trace order either by clicking the column header or by using the drag operation. The assistant has the following fields:

© 2023 Cadence Design Systems, Inc.

- *Input Wave Type*—Select the input wave type as `Time Domain Waveform` or `Frequency Domain Waveform`.

  You can calculate the FFT for the time domain waveform. However, the frequency domain waveform is already an FFT waveform and is used only to calculate measurements. You can use the frequency domain waveform if you are running simulation with the Spectre Fourier component. The Spectre Fourier component performs a Fourier integral and outputs the results in the frequency domain.

- *FFT Input Method*—Select the Fast Fourier Transform (FFT) input method from the drop-down list box.

  If you select *Calculate Sample Frequency*, which is the default option, you need to specify the start and stop time. The Sample Count and Frequency fields display the values calculated based on the specified start and stop time.

  `Start Time` = Start time of the waveform
  `End Time` = End time of the waveform
  `Sample Frequency = SampleCount/(StopTime - StartTime)`

  If you select *Calculate Start Time*, you need to specify the stop time, sample count, and frequency. The *Start Time* field displays the value calculated based on the specified stop time, sample count, and frequency.

  ```
  StartTime= StopTime - SampleCount/SampleFreq
  ```

  If you select *Calculate Stop Time*, you need to specify the start time, sample count, and frequency. The *Stop Time* field displays the value that is calculated based on start time, sample count, and frequency that you specify.

- *Start/Stop Time*—Specify the start and stop time of the input time domain periodic waveform.

- *Sample Count/Freq*—Specify the sample count that is used to determine the number of frequency bins in the FFT waveform. Also, in the *Freq* field, specify the sampling frequency that determines the size of the frequency bin of the FFT waveform.

  The *Sample Count* value must be greater than zero. By default, this field displays the number of data points in the selected signal.

  ```
  SampleFrequency=SampleCount/(StopTime-StartTime)

  StartTime=StopTime-(SampleCount/SampleFreq)

  StopTime=Startime+(SampleCount/SampleFreq)
  ```


© 2023 Cadence Design Systems, Inc.

The number of input periods in the period of the Fourier transform needs to be an integer and a prime number, where the period of the Fourier transform is 1/ the fundamental frequency of the Fourier transform.

- *Window Type*—Specify the window function that you can apply to the waveform, such as `Hanning` or `Kaiser`. Window function is a mathematical function with zero-values outside of a given interval. The function is called rectangular window function when it is constant inside the given interval and zero elsewhere, which also describes the shape of its graphical representation. When another function or waveform is multiplied by a window function, then the result is also zero-valued outside the given interval.

  By default, this field displays `Rectangular`. Some of these window types increase the amplitude of the signal. This helps in distinguishing the noise in the signal.

  You can use `Rectangular` window type when you have an integer number of cycles. However, if you are simulating a sigma-delta modulator, you can use `Hanning` instead of `Rectangular` even when an integer number of cycles occur between `StartTime` and `EndTime`.

  For Nyquist Rate converters, use `Rectangular` window. For Delta-Sigma converters, use `Cosine2 (Hanning)` window instead of Rectangular even when an integer number of cycles occur between `startTime` and `endTime`.

  In D/A converters, the fourier component is from `analogLib`. The output of a D/A converter is an analog signal, which is continuous. When Fourier Transform samples the input, some information can be lost; therefore, the Integral is used.

- *Plot FFT (units)*—Select this check box if you want to plot the FFT waveform. In the drop-down list to the right, select the values you want to plot in the FFT waveform. The values can be `dB`, `Imaginary`, `Magnitude`, and `Real`.

- *Plot PSD*—Select this check box if you want to plot the power spectral density waveform. By default, it is not selected. In the drop-down list to the right, select the values you want to plot in the PSD waveform. These values can be *Magnitude* and *dBV/Hz*.

  If you select *Magnitude*, the PSD waveform is plotted with `V^2/Hz` as the y-axis unit. If you select *dBV/Hz*, the PSD waveform is plotted with `dBV/Hz` as the y-axis unit.

- *Window Size*—Specify the number of frequency domain points to use in the Fourier analysis. A larger window size results in an expectation operation over fewer samples, which leads to larger variations in the power spectral density. A small window size can smear out sharp steps in the power spectral density that might be present.
  Default value= Sample count. The value of window size is automatically filled with

© 2023 Cadence Design Systems, Inc.

the sample count when Plot PSD is first checked.
The field is only enabled when the *Plot PSD* option is selected.

- *Start/End Freq*—Specify the lower limit and the upper limit of the frequency range for spectrum measures. By default, the lower limit field displays the first frequency point of the FFT and the upper limit field displays the last frequency point of the FFT.

  To synchronize the start and end frequency values with the specified start and stop time, click the `S` button. You can also change the start and end frequency values, if required.

- *Signal Bins*—Specify the number of signal bins. When you select a window type, this field displays the default number of bins for the selected window type. For example, if you select the *Window Type* as `Kaiser` that has two signal bins, this field displays `2`. You can increase the number of signal bins to up to half the value of the sample count. For example, if the sample count is `16` for the window type `Kaiser`, you can increase the signal bin count in the *Signal Bins* field up to `8`. You cannot decrease the displayed signal bin value.

  By default, this field displays zero to indicate the rectangular window type. This specifies the number of bins on each side of the signal bin or harmonic bin of the FFT waveform that are to be considered as part of the signal or harmonic.

  Signal bin for `Hanning`, `Hamming`, `Cosine2` windows is 1.

  Signal bins for `Blackman`, `ExtCosBell`, `Kaiser`, `Cosine4` signal bin are 2.

  Signal bins for `HalfCycleSine`, `Half3CycleSine`, `HalfCycleSine3`, `Half6CycleSine`, `HalfCycleSine6`, `Parzen` signal bin are 3.

  Bins are used to calculate the total signal power, `P_Noise`, and `P_distortion`. The `startBin` and `endBin` values are obtained from the start and end frequencies.

  If you change the number of the signal bins, the measurements in the Outputs section are changed. For example, if you change the value of signal bins, the signal power and noise power are also changed. This also changes the SINAD and ENOB values.

- *Peak Sat. Level*—Specify the peak saturation level of the FFT waveform. Magnitude of the FFT wave is divided by the Peak Sat Level before using it in calculations. Peak sat level is the full-scale span ignoring any DC offsets and used in ENOB calculation.
  Valid values: Any floating point number.
  Default value: If the peak sat level is not specified or `nil`, it is assumed to be 0 and is taken to be the peak-to-peak value of the fundamental.

© 2023 Cadence Design Systems, Inc.

For more information, see How the Peak Sat. Level value is Calculated? on page 236.

- *Harmonics*—Specify the number of harmonics for the waveform that you want to plot. For example, If this variable is n, where n should be greater than 1 and the fundamental frequency is harmonic 1, the n harmonics are considered for the harmonic power calculation. The signal bins are used for calculating the harmonic power.

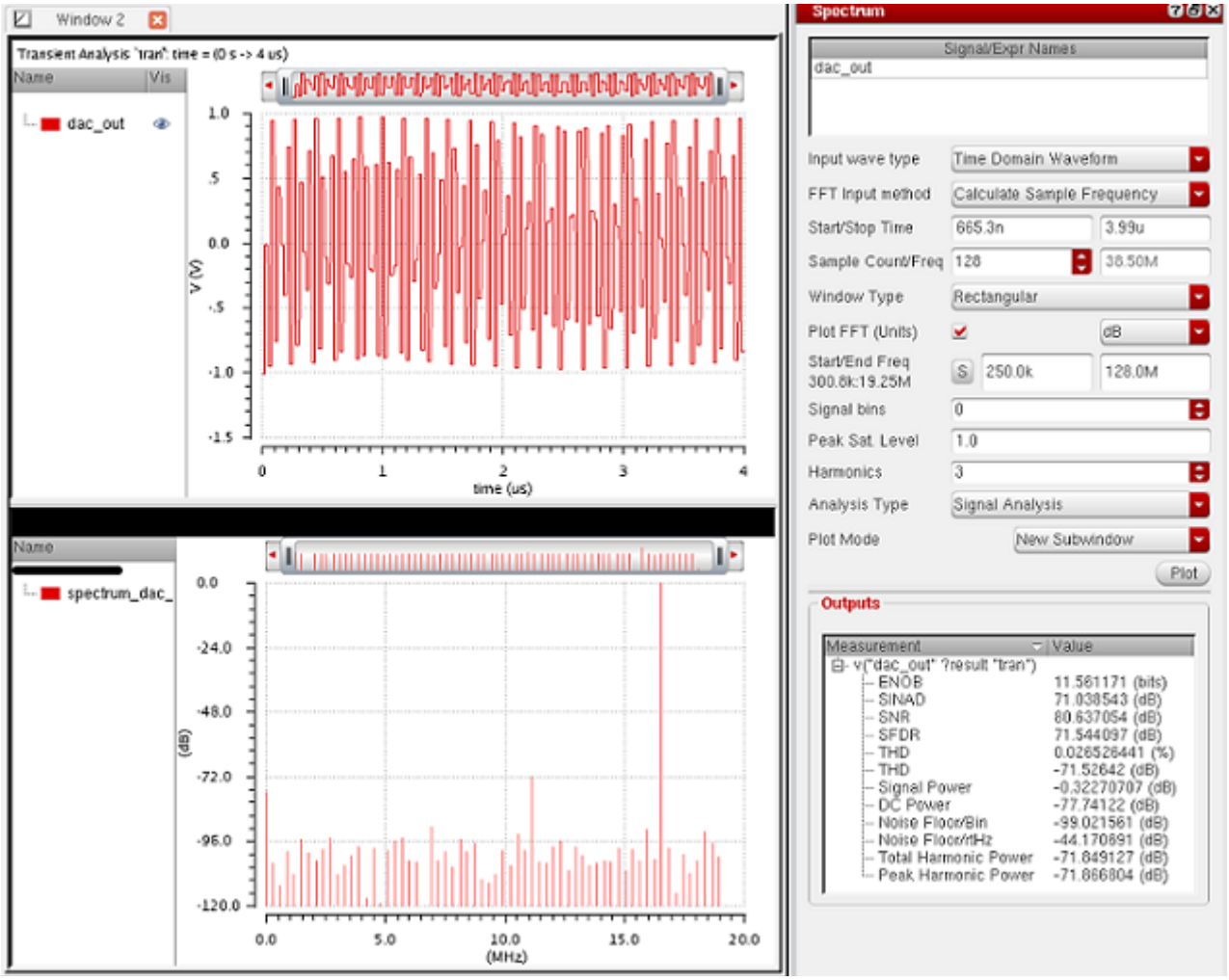
  For example, to calculate the total harmonic distortion (THD), if you set the *Harmonics* value to `n`, where `n` is greater than `1`, and the fundamental frequency is harmonic 1, the number of harmonics used to calculate THD is `2`,...,`n`. If `n=3`, the `2`nd and `3`rd harmonics are used to calculate THD.

- *Analysis Type*—Specify the analysis type as `Signal Analysis` or `Noise Analysis`. Based on the analysis type that you select, the measurement values are calculated and displayed in the *Outputs* section. If you want to use only the noise in the frequency domain waveform as input, select `Noise Analysis`. However, if you want to use both the signal and noise in the frequency domain waveform as input, select `Signal Analysis`.

- *Plot Mode*—Specify whether you want to append the FFT waveform to the existing graph, replace the existing graph, or plot the waveform in a new window or a subwindow.

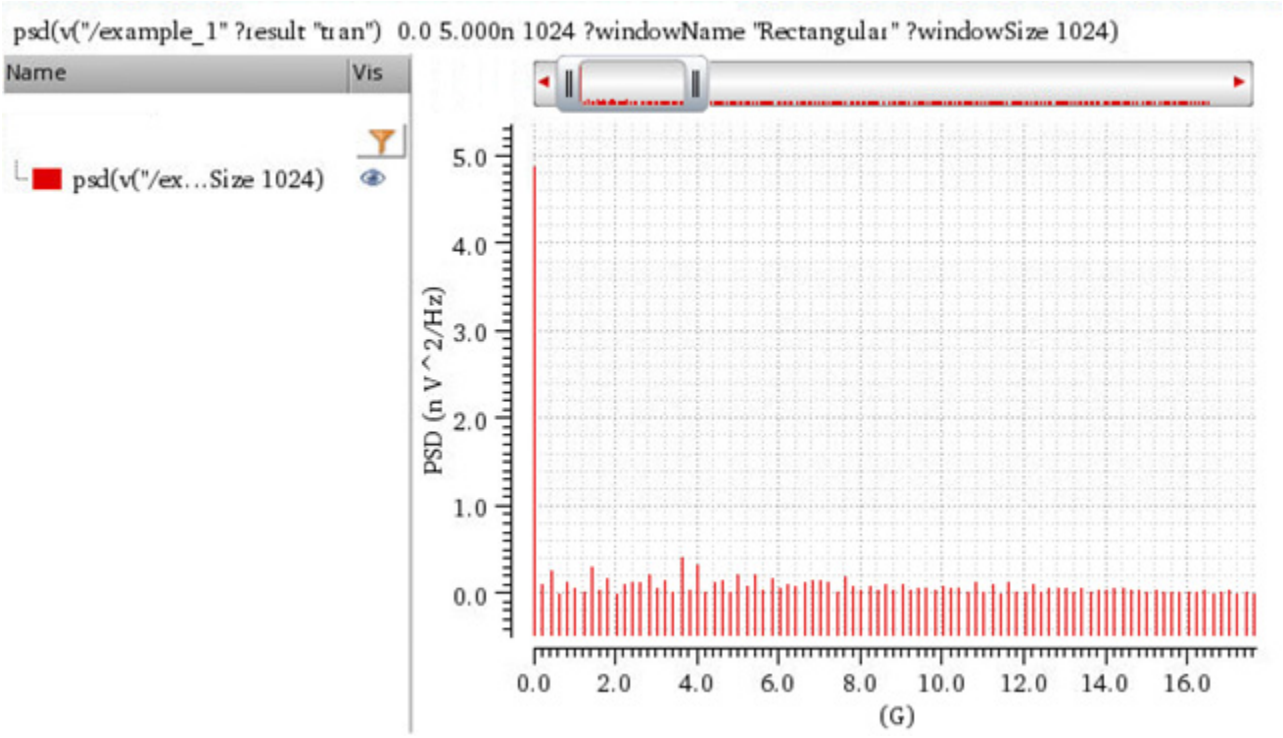
© 2023 Cadence Design Systems, Inc.

- *Plot*—Click the *Plot* button to plot and evaluate the FFT waveform and its measurements.



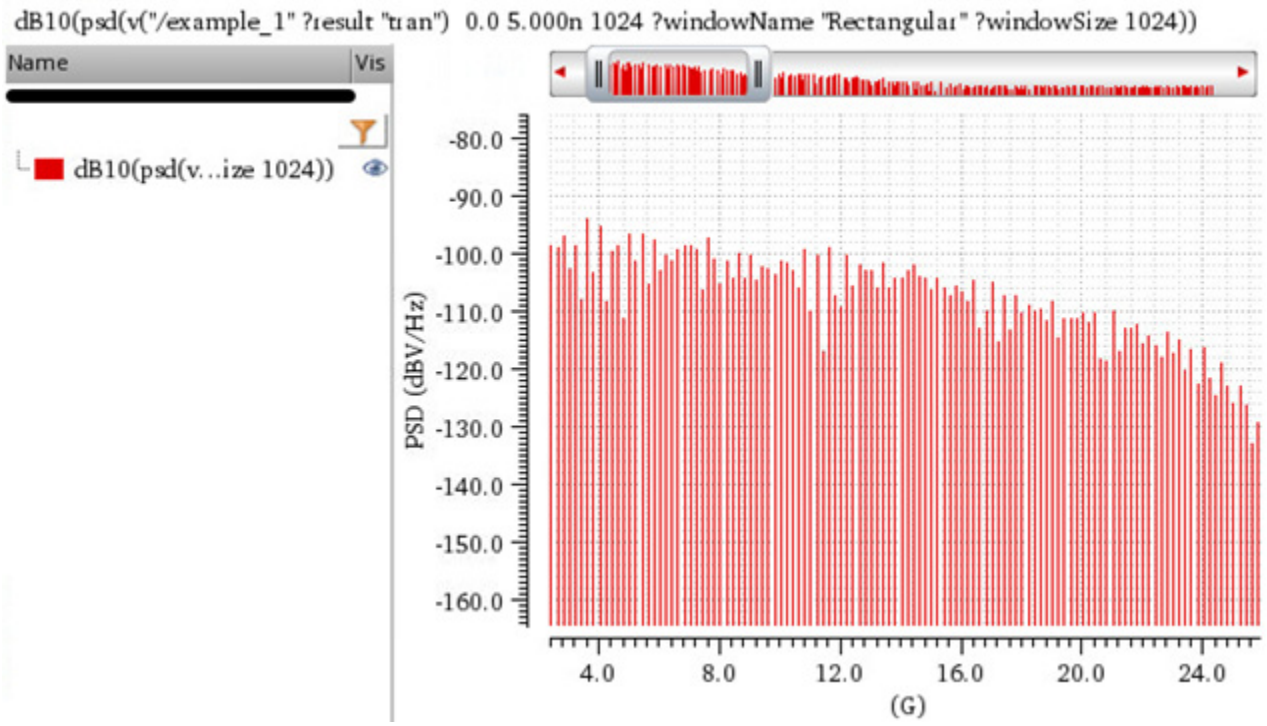
© 2023 Cadence Design Systems, Inc.

The following figure shows the PSD waveform plotted with *Magnitude*.



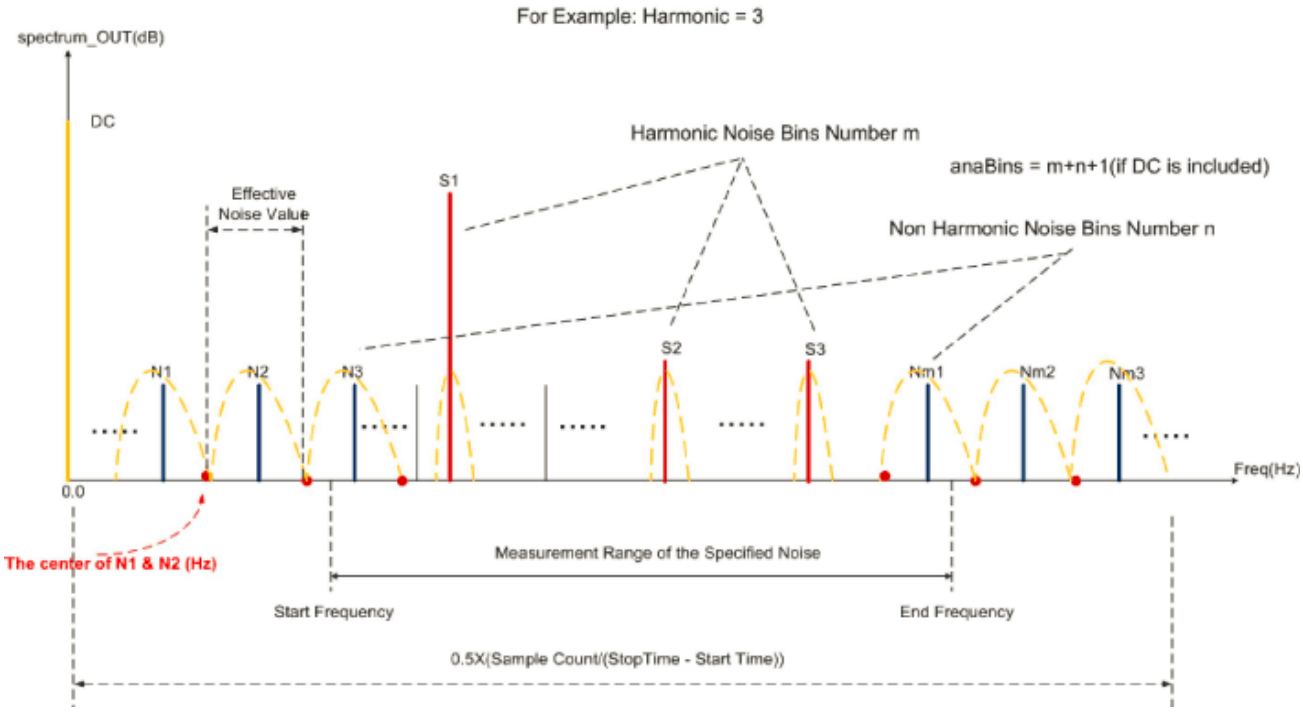
© 2023 Cadence Design Systems, Inc.

The following figure shows the PSD waveform plotted with *dBV/Hz*.



© 2023 Cadence Design Systems, Inc.

The figure below displays a Spectrum plot labeled with input fields, Start and End Frequency, Signal bins, Harmonics, in the Spectrum assistant and explains how these fields are mapped to the output plot.



Outputs

The *Outputs* section displays the following measurements and their values:

- *SINAD* (Signal to Noise and Distortion Ratio)
- *SNR* (Signal to Noise Ratio)
- *SFDR* (Spurious Free Dynamic Range)
- *ENOB* (Effective Number Of Bits)
- *Signal Power*
- *DC Power*
- *Noise Floor/Bin*
- *Noise Floor/rtHz*
- *Total Harmonic Power*

© 2023 Cadence Design Systems, Inc.

- *Peak Harmonic Power*

How the Peak Sat. Level value is Calculated?

The Peak Sat. Level or ADC Span is the maximum amplitude the ADC can convert. For example, if the input voltage range of the ADC is [-500mV, 500mV], then the Peak Sat. Level is 500mV.

- If the peak sat. level is not specified, default value is 0, peak sat. level is assumed to be 1.

  The noise floor/bin is the noise power calculated from the sum of all the noise bins, which are not dc, signal, or harmonic, divided by the number of non-DC bins, and then converted to decibels 10*log10(result).

  ```
  Noise floor/rtHz is the NoiseFloor/bin divided by the frequency
  width of a bin.
  ```

- If peak sat. level is specified, for example 500mV, and you need to calculate noise floor/bin, calculate the power sum of all the noise bins.

  Power of each noise bin is:

  ```
  [(magnitude of each noise bin)/(peak sat.lvl)]^2 = [normalized
  magnitude of each noise bin]^2
  ```

  In this case, peak or RMS value is not meaningless while calculating power because each noise bin uses the normalized power.

- If peak sat. level is not specified, default value is 0, the Peak Sat. Level is assumed to be 1. The power of each noise bin is:

  ```
  (magnitude of each noise bin)/(peak sat. lvl)^2 = (magnitude of
  each noise bin)^2
  ```

  In this case, the power is a peak value.

Calculations

All power values are calculated relative to the Peak Sat. Level. When it is not given or is 0, the peak sat. level is assumed to be 1. No correction factor will be used when calculating ENOB.

- ENOB (Effective Number Of Bits)

  If Peak Sat. Level is not given, which means it is 0, ENOB is calculated as below:

  ```
  ENOB=(SINAD-10*log10(1.5))/(20*log10(2))
  ```


© 2023 Cadence Design Systems, Inc.

If Peak Sat Level is given, a correction factor is considered as below:

```
ENOB=(SINAD-10*log10(1.5)-20*log10(correction factor))/
(20*log10(2))
```

Correction factor = Magnitude of fundamental/Peak Sat. Level

- SINAD (Signal to Noise and Distortion Ratio)—SINAD is the ratio of total signal power to the noise-plus-distortion power. It is calculated using the following algorithm:

  ```
  SINAD = (Signal Power)/(Noise and Harmonic Power + Noise
  Correction Factor)
  ```

  ```
  Noise Correction Factor =Noise Power * (Number of signal bins)/
  (Number of non-harmonic noise bins)
  ```

  The noise correction factor estimates the noise power in the signal bins.

- SNHR—Ratio of fundamental (meaningful information of signal) to the background noise floor, excluding the power at DC and harmonics.

  ```
  snrh= dB10(totalSigPwr/noiseTotalNonHarm)
  ```

- SNR (Signal to Noise Ratio)

  ```
  SNR = dB10(P_Signal/TotalNonHarmNoise)
  ```

- SFDR (Spurious Free Dynamic Range)

  SFDR is the strength ratio of the fundamental signal to the strongest spurious signal in the specified frequency band set by the Start and End frequency fields.

  ```
  SFDR = dB10(Max_Signal_Power/max(maxNoise, maxHarmNoise))
  ```

  `Max_Signal_Power` is the maximum value of y-vector in the total signal power calculation.

  ```
  maxNoise = Max of arr[i];
  ```

  where `i` is the index of array elements that were not considered while calculating total signal power, total harmonic power and power at DC. `MaxNoise` is the maximum noise component.

  ```
  maxHarmNoise = Max(arr[i]) ;
  ```

  where, `i` is the index of the harmonic.

- THD (Total Harmonic Distortion)

  ```
  THD = 100 * sqrt(total_Harmonic_Pwr/P_Signal)
  ```


© 2023 Cadence Design Systems, Inc.

where, `total_Harmonic_Pwr` is the sum of all harmonic powers

- Signal Power

  Signal power is the power at the fundamental frequency relative to the Peak Sat. Level. It is calculated as:

$$\text{Signal Power} = 10\log_{10}\frac{A^2_{\text{fund,rms}}}{A^2_{\text{peak sat. lvl, rms}}}$$

  Here, A indicates amplitude.

  In the case of smoothing (non-Rectangular) windows, frequencies near the fundamental are also included and a coherent power gain factor is applied. For example, in the case of the Hanning window, one bin before and after the fundamental frequency is included and the resulting sum is divided by a coherent power gain factor of 0.375.

- DC Power

  ```
  dc_power= arrI[0]*arrI[0]+arrR[0]*arrR[0]
  ```

  `arrI` is Imaginary part and `arrR` is Real part

- Noise Floor/Bin

  ```
  NoiseFloor/Bin = dB10(TotalNonHarmNoise/anaBins)
  ```

  ```
  noiseNonHarmPwr = Sum of arr[i] * arr[i];
  ```

  where, `i` is the index of array elements which were not considered in the calculation of total signal power (the fundamental), total harmonic power, and DC.

  ```
  Average [noiseNonHarmPwr] = (noiseNonHarmPwr / n)
  ```

  where, `n` is the total number of bins used

  ```
  TotalNonHarmNoise = (noiseNonHarmPwr / n) * anaBins
  ```

  `anaBins` is length of the array `arr[]`

- Noise Floor/rtHz

  ```
  NoiseFloor/rtHz = dB10((TotalNonHarmNoise/(anaBins-1))/binSize)
  ```

  `binSize` is difference of the two nearest X points `(frequency - previousFrequency)` in the FFT wave.

© 2023 Cadence Design Systems, Inc.

Important

When you send the computed measurement values from the Spectrum toolbox to ADE Outputs and create an expression for them using ADE, the `spectrumMeasurement` function is used in the expression. For more information about spectrumMeasurement function, see spectrumMeasurement in the *OCEAN Reference*.

Additional Information

Variations on SNR, THD, SINAD and THDN with the Harmonics value of Spectrum assistant:

Total power (T) = Sum of signal, distortion and noise

$$T = S + D + N$$

where,

S = Power in the fundamental signal peak

D = Power in harmonic peaks

N = Noise power between harmonics

Signal to Noise ratio, $SNR = S/N$

Total Harmonic Distortion, $THD = D/S$

Signal to Noise and Distortion Ratio, $SINAD = S/(N + D)$

Total Harmonic Distortion Plus Noise, $THD + N = (N + D)/S$

Number of Harmonics in Spectrum assistant defines how the power is divided between noise (N) and distortion (D).

When Harmonics = 1, all the power except for the fundamental signal peak will be attributed to the noise such that D = 0, THD = 0, and SINAD = SNR.

Increasing the number of harmonics will increase the value of D and reduce N, and thus, greater value of SNR.

For no harmonic peaks in the spectrum, it is assumed that there is no distortion and you can set harmonics equals to 1. Otherwise, always set the harmonics parameter equal to the number of harmonic peaks in the spectrum.

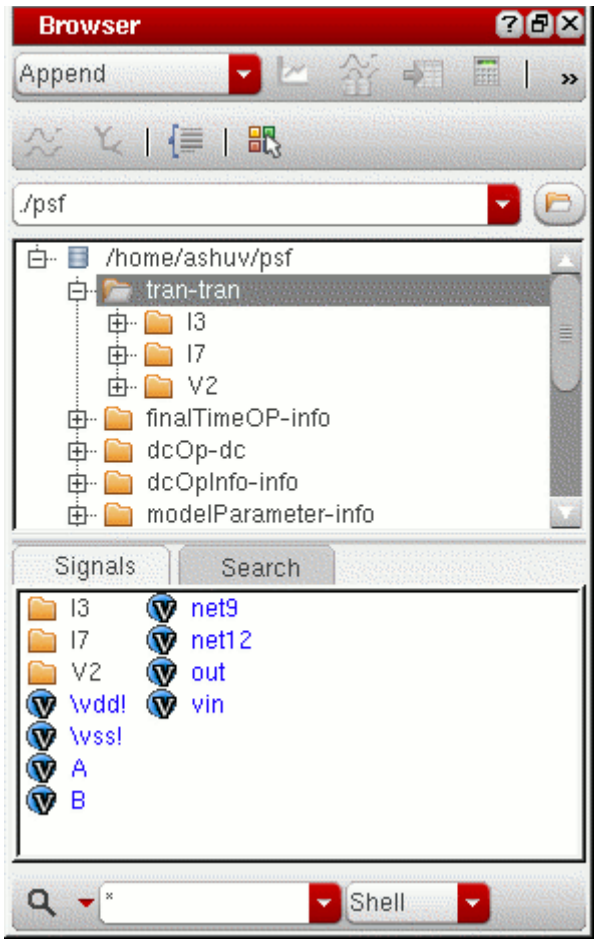
© 2023 Cadence Design Systems, Inc.

The overlapping between harmonic and DC bins can happen when the width of the signal peak is larger than the frequency resolution.

The *Signal bins* field controls the width of the signal and harmonic peaks. Frequency resolution is inverse of the time window.

Browser Assistant

The Browser assistant displays the Results Browser that you can use to open simulation results saved earlier. To open the Results Browser, choose *Window – Assistants – Browser.* For more information about the Results Browser, see Results Browser.



© 2023 Cadence Design Systems, Inc.

Marker Toolbox Assistant

Use this assistant to add point, vertical, horizontal, and reference point (ARefPoint or BRefPoint) markers to the trace. For more information about Marker Toolbox, see Adding Markers with Marker Toolbox on page 496.

© 2023 Cadence Design Systems, Inc.

Eye Diagram Assistant

You use the Eye Diagram assistant to create an eye. An eye diagram is a way of representing a data signal by repetitively sampling the signal and overlaying the repeated samples on the same X-axis.

The result is a plot that has many overlapping lines enclosing an empty space known as the eye. The quality of the receiver circuit is characterized by the dimension of the eye. An open eye means that the detector can distinguish between `1's` and `0's` in its input, while a closed eye means that a detector is likely to give errors for certain input bit sequences.

This section includes the following topics:

- Overview on page 242
- Opening the Eye Diagram Assistant on page 244
- Creating Triggered Eye Diagram on page 249
- Example: Creating an Eye Diagram on page 262
- Working with the Eye Measurements Tab on page 265
- Working with the Eye Mask Tab on page 288

Overview

Consider an example circuit generated by using a Pseudo-Random Binary Sequence (PRBS) source in the Spectre and Verilog-A model that can add random amplitude variation and random delays to the edges.

Spectre Netlist prbs.scs

```
// prbs.scs
v1 (high 0) vsource type=prbs period=40n val0=0 val1=1 rise=0.3n \
   fall=0.3n delay=3n
vclk (clk 0) vsource type=pulse period=80n val0=0 val1=1 rise=0.3n \
   fall=0.3n delay=3n
del (withdelay high) randdelay sd=1n rise=0.3n
rl (withdelay jitter) resistor r=100
cl (jitter 0) capacitor c=50p
r1 (high 0) resistor r=1k
ahdl_include "randdelay.va"
tran tran stop=400u
```

Verilog-A model randdelay.va

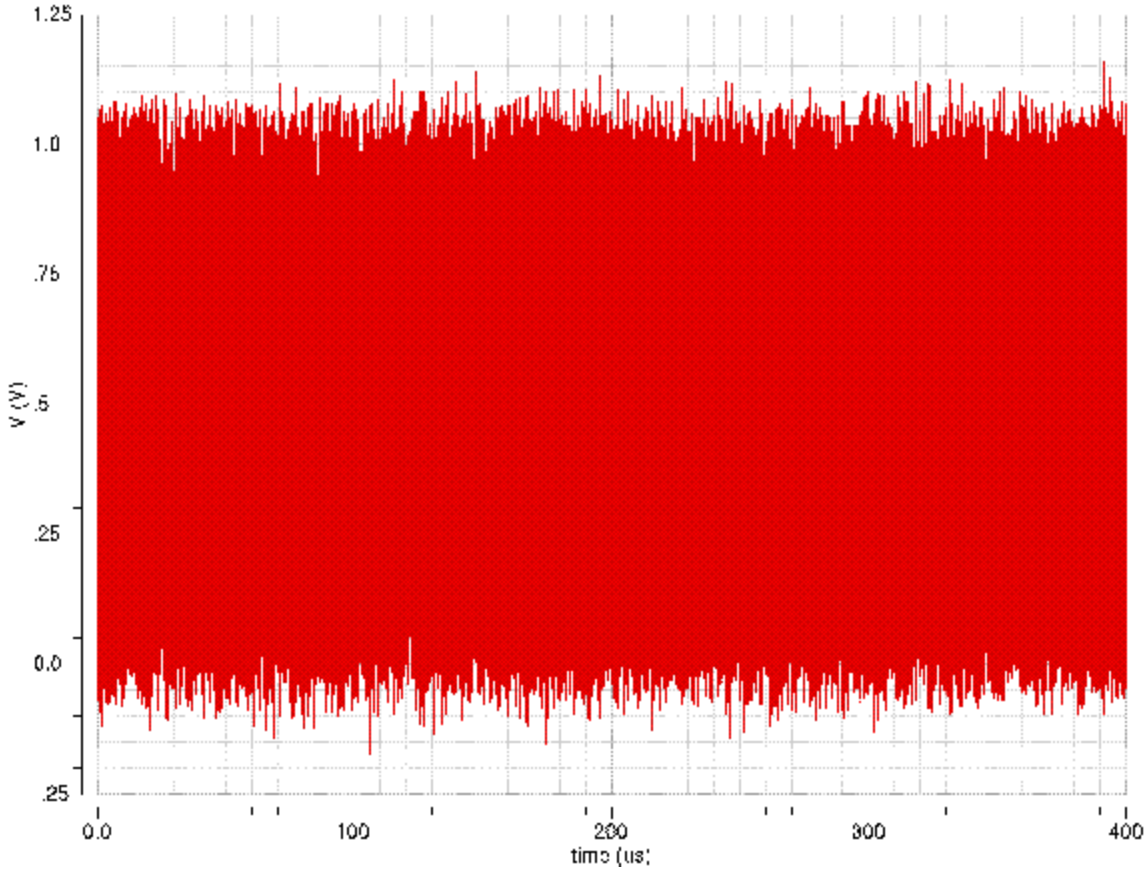
© 2023 Cadence Design Systems, Inc.

```
// VerilogA for randdelay
`include "constants.h"
`include "discipline.h"
module randdelay (op,ip);
output op;
input ip;
electrical op,ip;
parameter real sd=2.0n;
parameter real gainsd=0.1;
parameter real del=5.0n;
parameter real rise=0.1n;
parameter real thresh=0.5;
parameter integer seed = 23133;
integer vseed;
real randnum,gain,delayed;
analog begin
@(cross(V(ip)-thresh)) begin
    randnum=$rdist_normal(vseed,0,1);
    randnum=randnum*sd+del;
    gain=$rdist_normal(vseed,0,1)*gainsd+1;
    if(randnum<0) randnum=0.0;
  end
delayed= transition(V(ip),randnum,rise,rise);
V(op) <+ gain*(delayed-thresh)+thresh;
end
endmodule
```

Note: To run the simulation in the example shown above, type `spectre prbs.scs`.

© 2023 Cadence Design Systems, Inc.

When you plot the `jitter` signal as shown in the previous example, which is the output of the PRBS source that has passed through the Verilog-A model to add random delay and amplitude variation and then passed through a simple RC filter, you get the following plot:



This plot does not show how much the original signal is distorted. The zoomed in waveform also does not help visualize the timing and amplitude variation in the data. These issues can be resolved by using the Eye Diagram assistant.

Note: For more information about the Eye Diagram assistant, see Appendix C, “More About Eye Diagrams.”

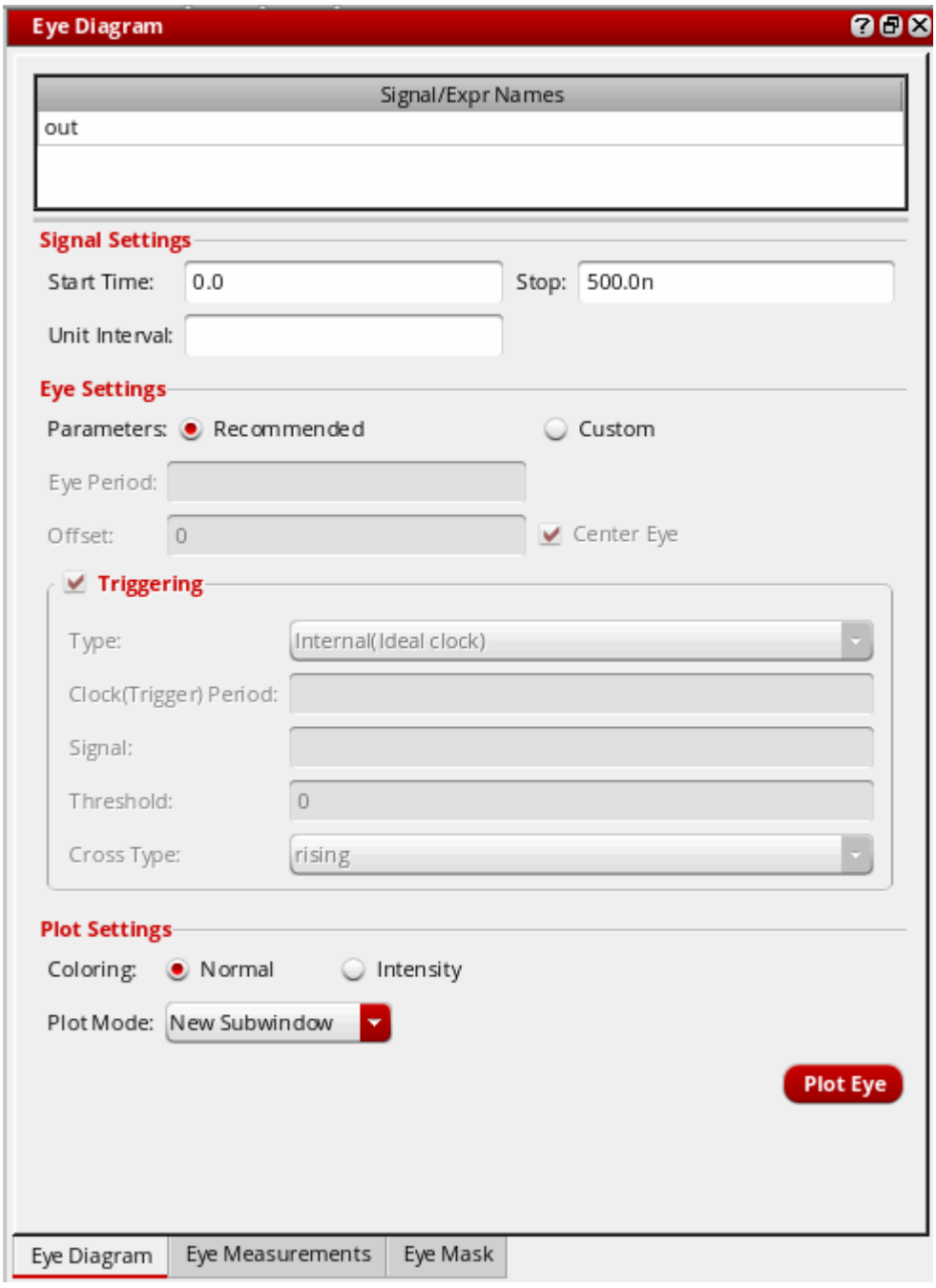
Opening the Eye Diagram Assistant

Select a signal in the window and do one of the following to open the Eye Diagram assistant:

- Choose *Measurements – Eye Diagram*.
- Choose *Window – Assistants – Eye Diagram.*

© 2023 Cadence Design Systems, Inc.

The *Eye Diagram* assistant appears on the right in the window.



The Eye Diagram form includes three tabs at the bottom—*Eye Diagram*, *Eye Measurement*, and *Eye Mask*. By default, the *Eye Diagram* tab is displayed when you open this assistant. This tab has the following fields:

© 2023 Cadence Design Systems, Inc.

- The *Signal/Expr Names* field displays the name of the traces that you select in an active window. The traces are displayed on the basis of their selection order; however, you can rearrange the trace order either by clicking the column header or by dragging the traces. To delete the traces that are not required, select the traces and press the `Delete` key.

Note: You can also drag signals from the Results Browser and directly send them to the Eye Diagram *Signal/Expr Names* field.

Note: The signals that do not result in generating an eye diagram cannot be added to the *Signal/Expr Names* field.

- In the *Signal Settings* section, specify the following fields:
  - In the *Start* field, specify the X-axis value from where the eye diagram plot needs to be plotted.
  - In the *Stop* field, specify the X-axis value where the eye-diagram plot must end.
  - In the *Unit Interval* field, specify the symbol period of the signal.
- In the *Eye Settings* section, select one of the following *Parameters* options:
  - *Recommended*—This is selected by default. With this option, a centered eye diagram is created with the eye period set to `2UI` and trigger period set to `1UI`.

  - *Custom* — Select this option if you want to specify your own settings. The *Eye Period*, *Offset*, and *Centre Eye* option fields become editable.

    - In the *Eye Period* field, specify the time period for the eye diagram. This is the period after X-axis starts repeating.

© 2023 Cadence Design Systems, Inc.

- In the *Offset* field, specify an offset value that is used to shift the eye. If you specify a positive value, the eye right is shifted to the right as per the specified value, and if you specify a negative value, the eye left is shifted to left. This field becomes inactive if you select the *Centre Eye* option.
- Select the *Center Eye* check box if you want the eye diagram to be placed in the center of the graph, which means the center of the eye opening is placed at the half of the eye period.

  Note: When the *Center Eye* check box is selected, the eye diagram is centered based on the crossing times across the average signal value between the start and the stop time.

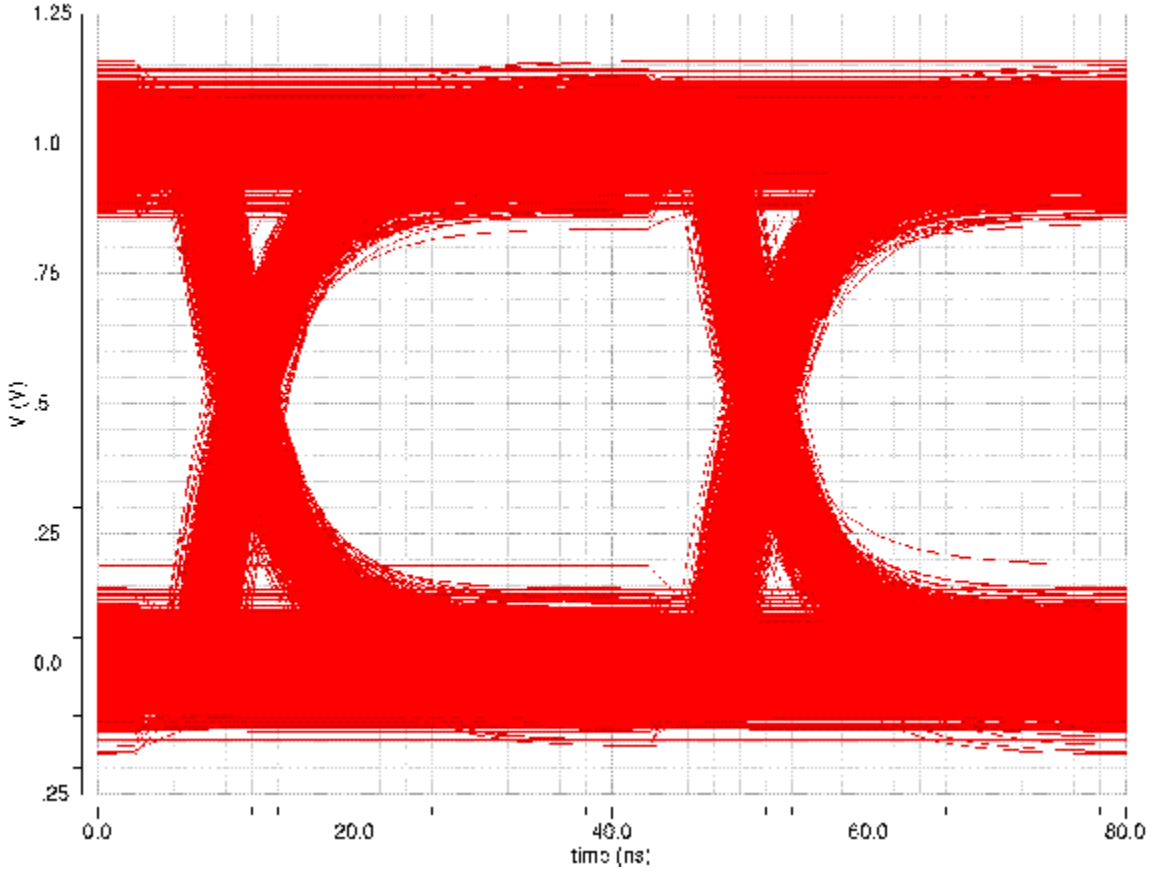
- Select the *Triggering* check box if you want to set up a trigger period (Internal Triggering) or use another signal to trigger the eye diagram instead of a fixed period (Edge Triggering). For more information about how to plot the edge triggered eye diagram, see Creating Triggered Eye Diagram on page 249.

  Note: If the trigger period is not specified, the eye period is used as the trigger period.

- In the *Plot Settings* section, specify the following fields:
  - Select the *Coloring* option as:
    - *Normal* to create the eye diagram with the default color scheme.
    - *Intensity* to highlight the intersection points of the eye diagram.
  - In the *Plot Mode* list, specify whether you want to *append* the eye diagram to an existing graph, *replace* an existing graph with the eye diagram, or add the eye diagram to a *new subwindow* or a *new window*.
- Click the *Plot Eye* button to plot the eye diagram for the selected signal. By default, the name of the eye diagram is *eye_<signal-name>.*

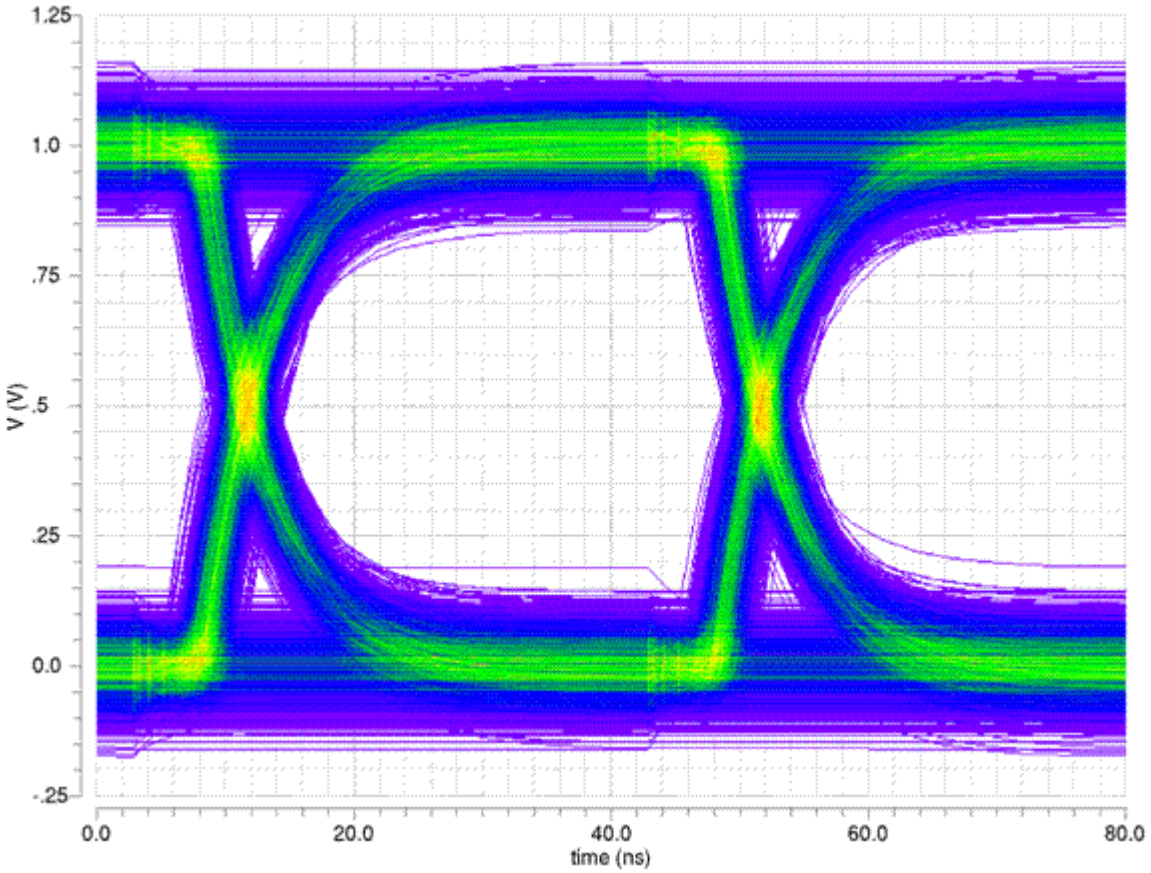
© 2023 Cadence Design Systems, Inc.

The following eye diagram appears.



© 2023 Cadence Design Systems, Inc.

When you select the *Intensity* check box, the graph is displayed as shown below:



This graph helps find the extent of the amplitude variation and timing variation in the transitions. Colors displayed in the graph are used to show regions of greater density, which helps visualize the distribution of amplitude and timing variation.

The opening in the middle of the graph is known as eye. The circuit or the system is considered bad if the opening is small.

Creating Triggered Eye Diagram

The eye diagram is obtained by overlapping several slices of an input transient signal. The width of each slice is called eye period. You can use the triggering options to specify the starting point from where each slice should begin. In a basic eye diagram plot, the slices begin at every eye period interval.

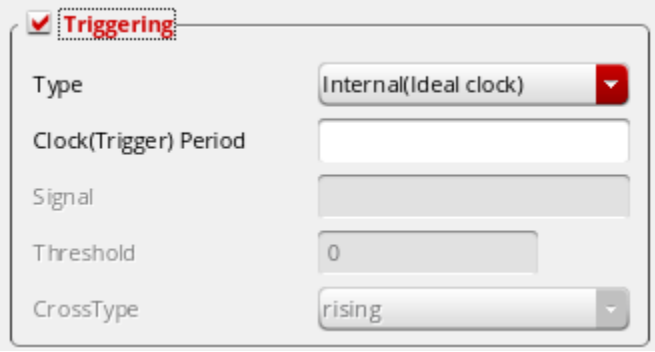
© 2023 Cadence Design Systems, Inc.

You can select the *Triggering* check box in the Eye Diagram assistant if you want to setup a trigger period (Internal Triggering) or another signal to trigger the eye diagram instead of a fixed period (Edge Triggering).

You can generate two types of the triggered eye diagram—Internal (Ideal Clock) and External (edge). In the *Type* field, select the type in the drop-down list.

Internal Triggering

When you select type as `Internal(Ideal Clock)`, the *Clock (Trigger) Period* field is enabled.



- In the *Clock(Trigger) Period* field, specify the time interval between triggering events.

  Note: When the *Clock (Trigger) Period* field is set to half of the period value, the resulting eye diagram is symmetrical across the vertical line that cuts through the eye center.

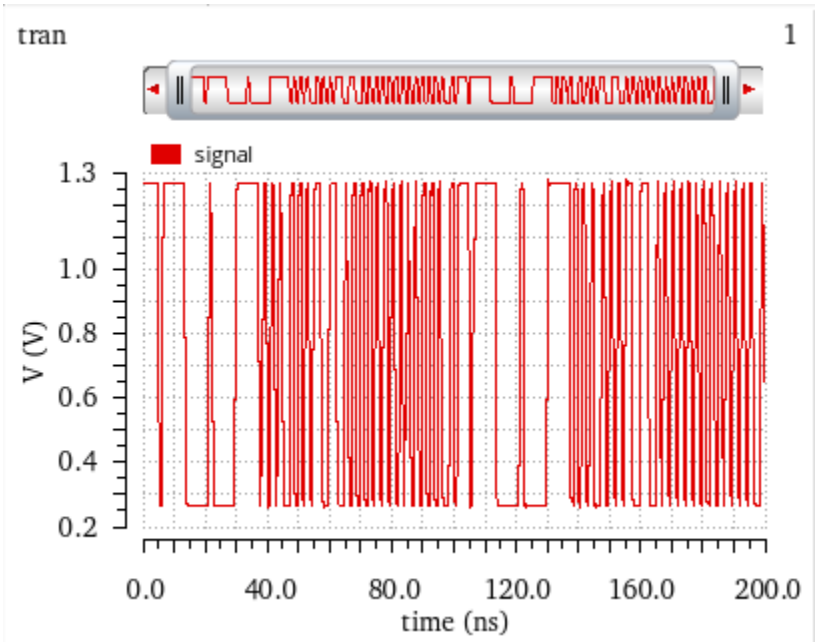
- Specify the plot mode type in the *Plot Mode* drop-down and select the *Intensity* check box if required.
- Click the *Plot Eye* button to plot the internal triggered eye diagram.

Example

Consider the below example in which transient analysis results are used to generate and analyze the triggered eye diagram of type internal.

© 2023 Cadence Design Systems, Inc.

1. Plot `Signal` from the transient results directory.

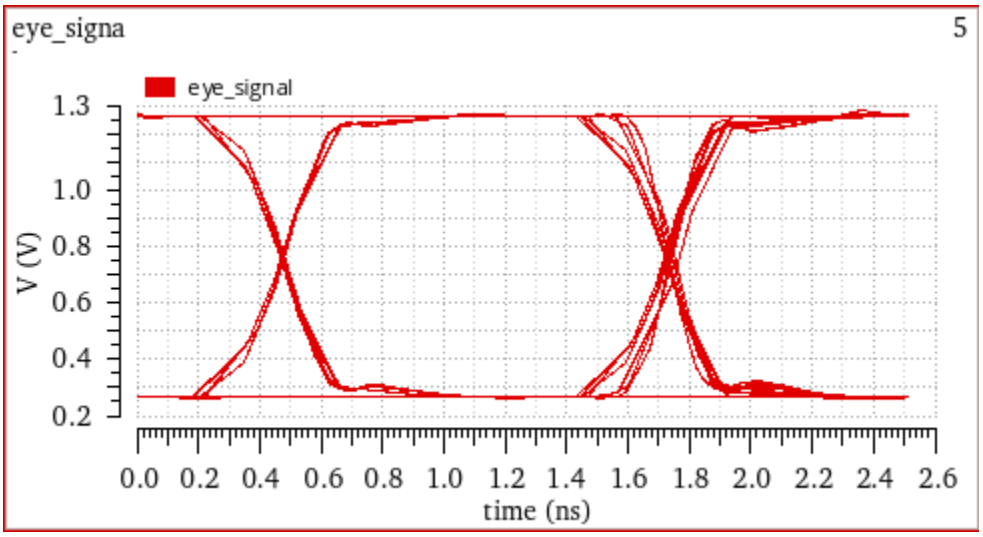


Perform the following steps to generate a triggered eye diagram of type internal:

1. Open the Eye Diagram assistant and plot an eye diagram for this signal with the following field values:
   - Select `Signal` in the *Signal/Expr Names* field.
   - *Start*—`0n`
   - *Stop*—`200n`
   - *Eye Period*—`2.5n`
   - *Offset*—`0`

© 2023 Cadence Design Systems, Inc.

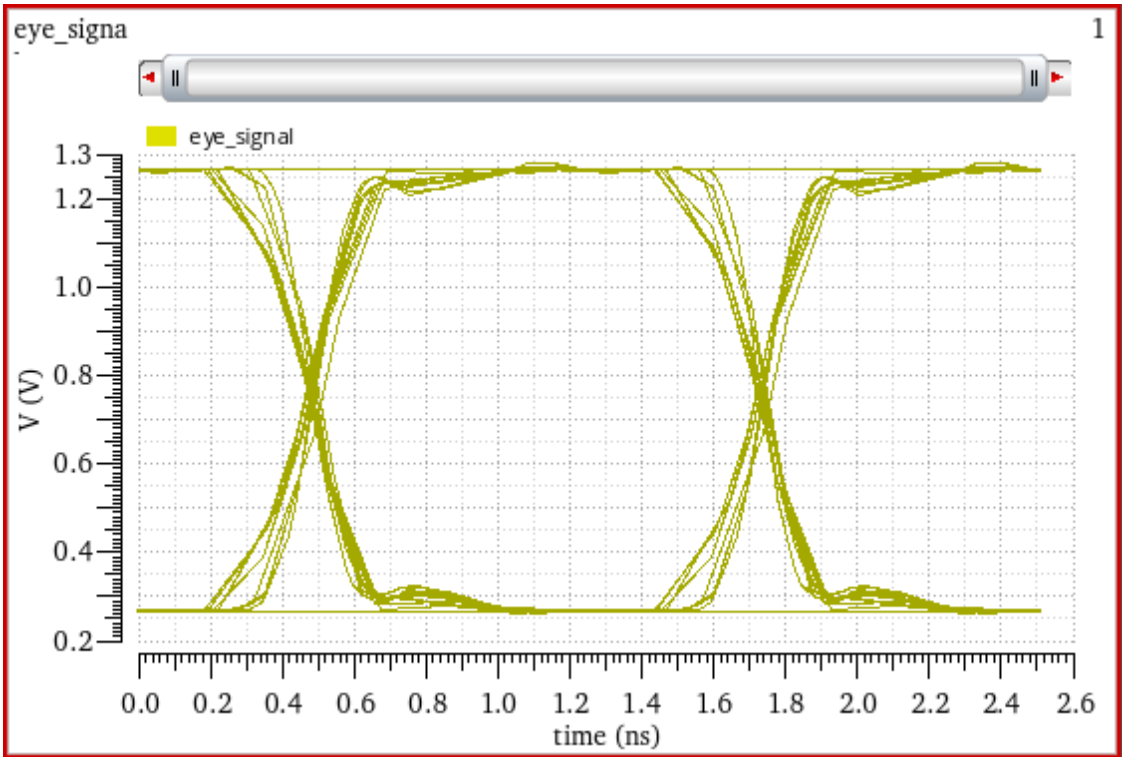
2. Click the *Plot Eye* button to plot the eye diagram. The following eye diagram appears:



3. Select the *Triggering* check box to enable the eye diagram triggering options.
4. Specify the following field values:
   - *Type*—`Internal(ideal clock)`
   - *Clock(Trigger) Period*—`1.25n`
   - *Plot Mode*—`New Window`
5. Click the *Plot Eye* button. The following eye diagram is generated. The new eye generated is placed on the previous eye diagram plot in such a way that the resultant eye

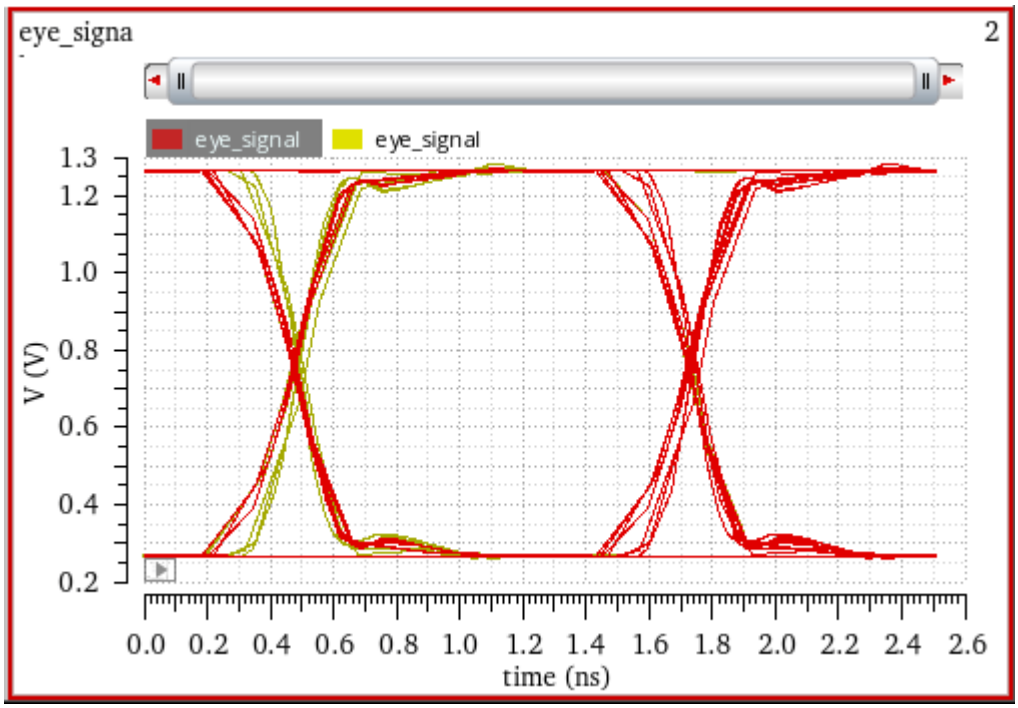
© 2023 Cadence Design Systems, Inc.

diagram is complete and does not skip any period value. Notice that the edges in the Internally triggered eye diagram are symmetrical placed on both the sides.



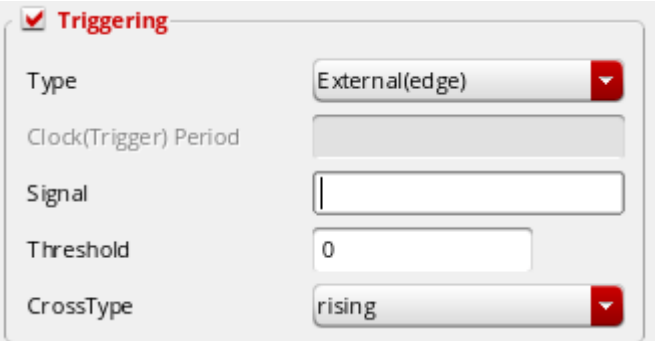
© 2023 Cadence Design Systems, Inc.

When you specify the plot mode as append, the internally triggered eye diagram is plotted as shown in the below figure. In this plot, notice that the resultant eye (in yellow) with trigger period = 1.25n is symmetrical on both the sides as compared to the red eye.



External Triggering

When you select type as `External(Edge)`, the *Signal*, *Threshold*, *Offset*, *Center Eye*, and *CrossType* fields are enabled.



- Click in the *Signal* field and select a signal that you want to use as a triggering signal. The selected signal name appears in this field.

© 2023 Cadence Design Systems, Inc.

- In the *Threshold* field, specify a threshold value. When a signal to be triggered crosses the specified threshold value in a given direction, the eye diagram starts a new period. You can specify direction in the *CrossType* field.
- Select the *CrossType* as `rising`, `falling`, or `either`.
- Specify the plot mode type in the *Plot Mode* drop-down and select the *Intensity* check box if required.
- Click the *Plot Eye* button to plot an edge triggered signal.

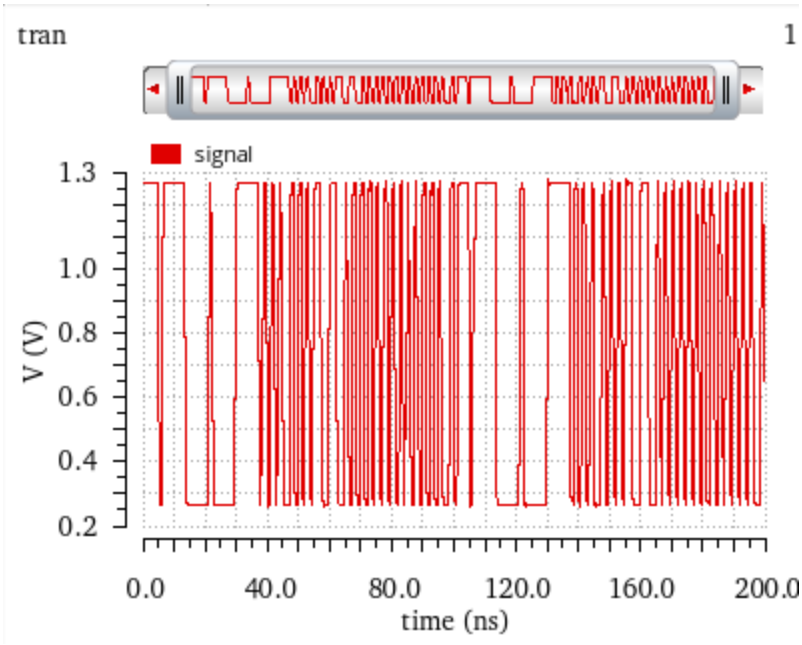
Example

Consider the below example in which transient analysis results are used to generate and analyze the triggered eye diagram of type external.

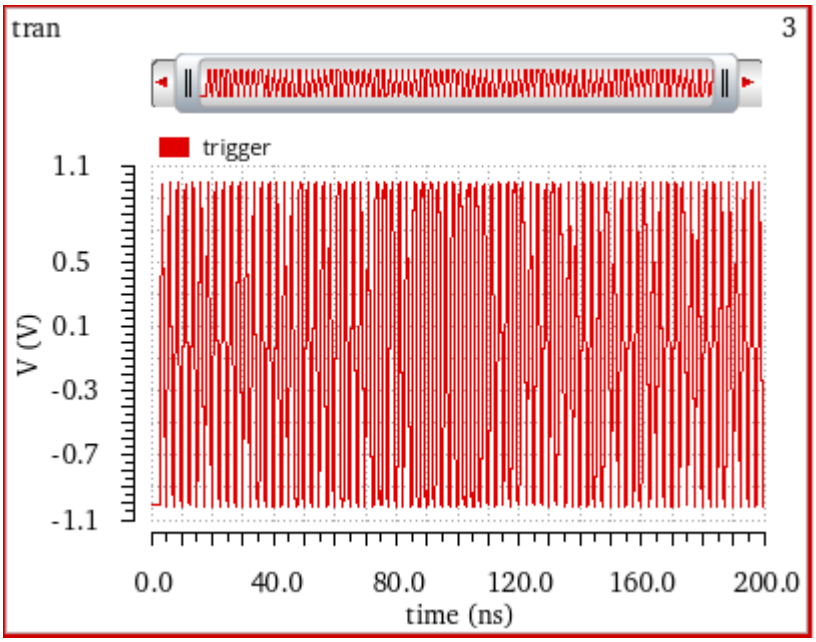
1. Plot `signal` and `trigger` (to be used as a trigger signal) from the transient results directory.

© 2023 Cadence Design Systems, Inc.

Signal:



Trigger:



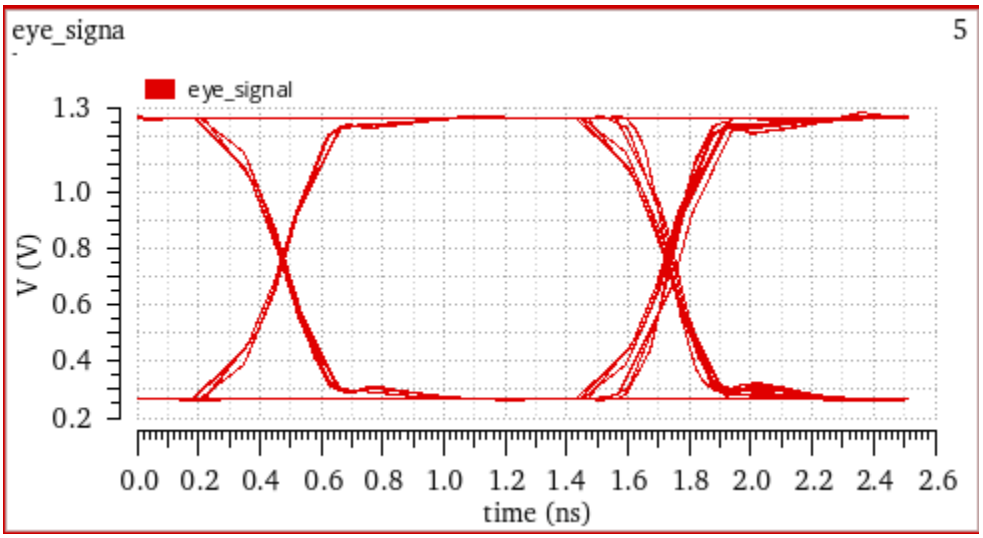
Perform the following steps to generate a triggered eye diagram of type external:

1. Open the Eye Diagram assistant and plot an eye diagram for this signal with the following field values:
   - ❑ Select `Signal` in the *Signal/Expr Names* field.

© 2023 Cadence Design Systems, Inc.

- *Start*—`0n`
- *Stop*—`200n`
- *Eye Period*—`2.5n`
- *Offset*—`0`

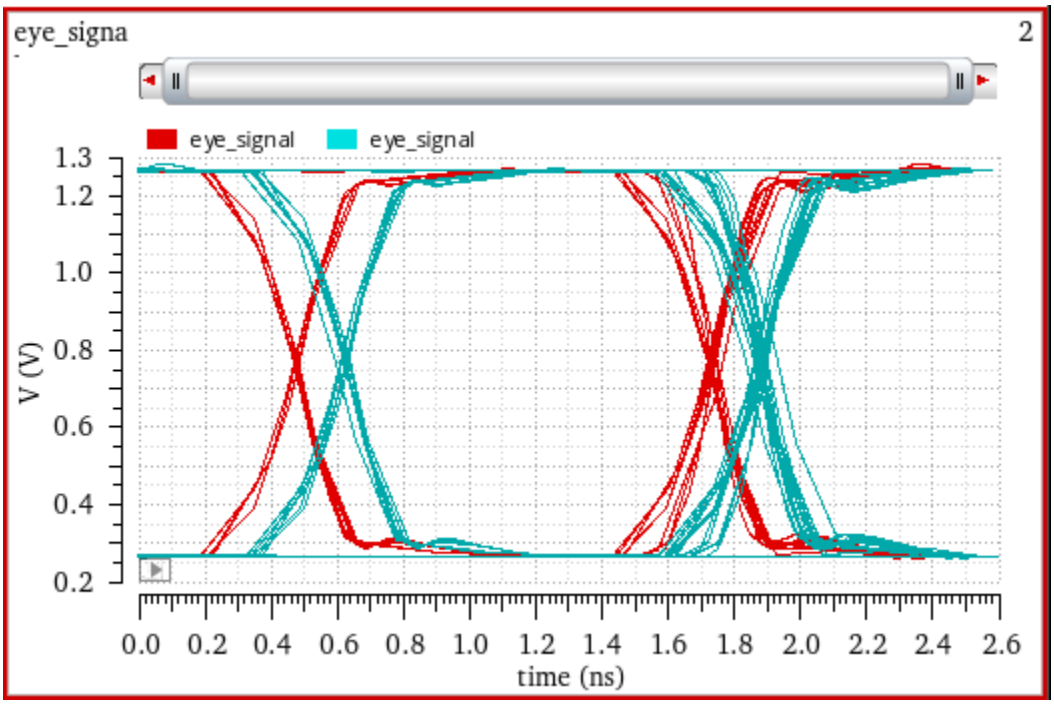
2. Click the *Plot Eye* button to plot the eye diagram. The following eye diagram appears:



3. Select the *Triggering* check box to enable the eye diagram triggering options.
4. Specify the following field values:
   - *Type*—`External(edge)`
   - *Signal*—`trigger`
   - Threshold—`0`
   - CrossType—`rising`
   - *Plot Mode*—`Append`

© 2023 Cadence Design Systems, Inc.

5. Click the *Plot Eye* button. The following eye diagram is generated.



Shifting Eye Diagram to Center

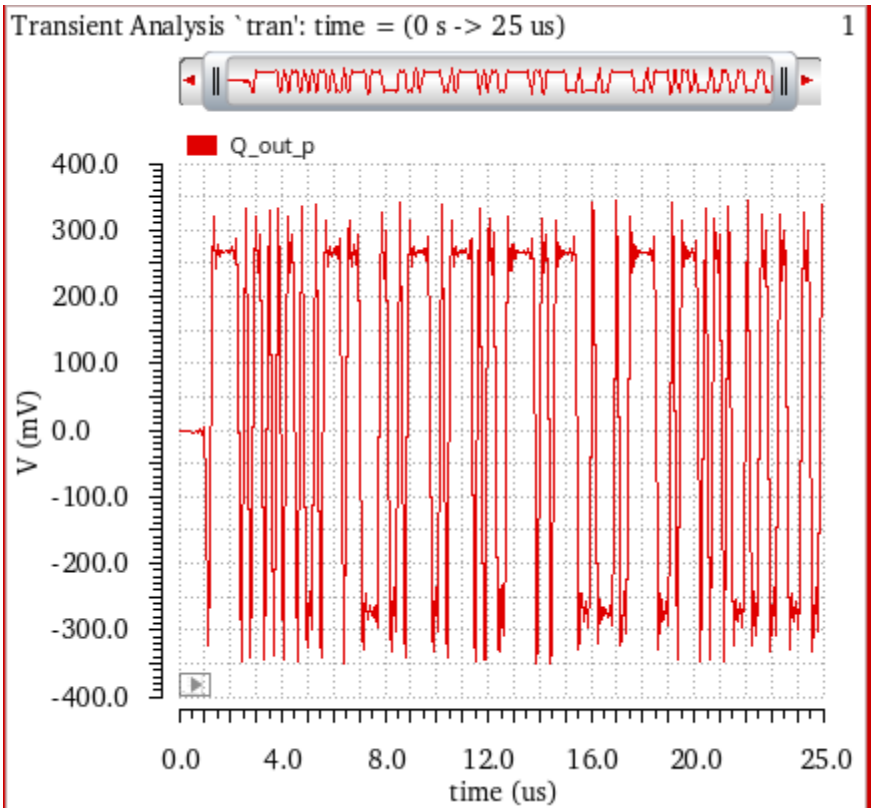
When the *Recommended* option is selected in the *Eye Settings* field, by default the centered eye diagram is created.

When the Custom option is selected, select the *Center Eye* check box to create the centered eye diagram. The resultant eye diagram has the clock trigger period same as that of the eye period.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following example in which the simulation results from transient analysis is used to create a eye diagram in the center of the graph. Firstly, plot signal `Q_out_p`.

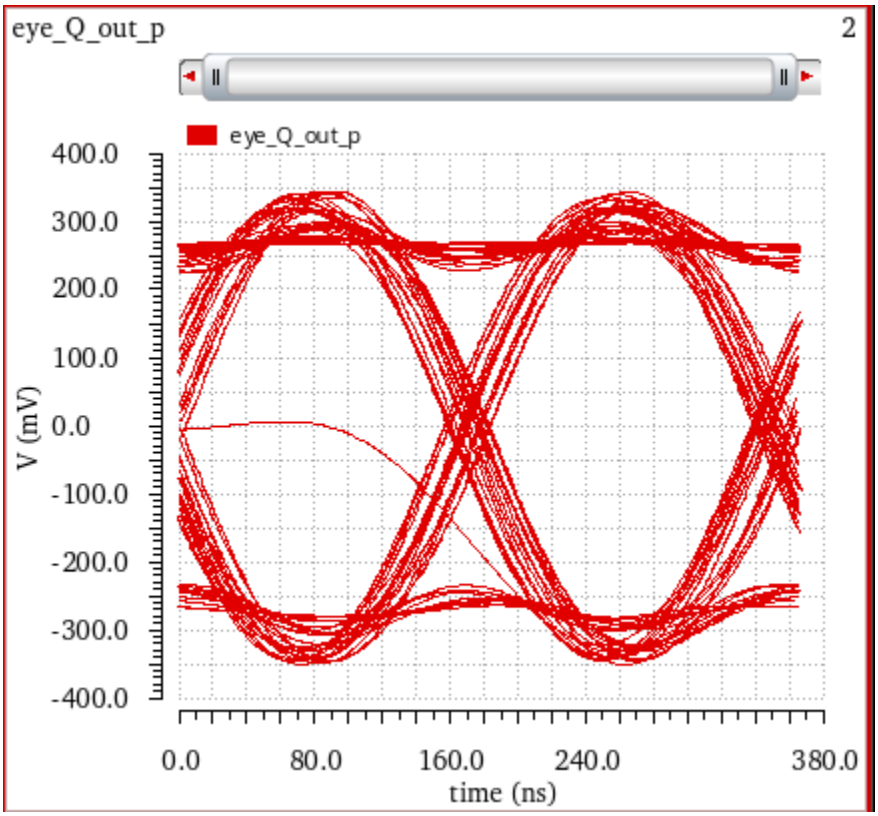


Perform the following steps to generate a centered eye diagram:

1. Open the Eye Diagram assistant and plot an eye diagram for this signal with the following field values:
   - Select `Q_out_p` in the *Signal/Expr Names* field.
   - *Start*—`8.3e-07`
   - *Stop*—`2.4e-05`
   - *Eye Period*—`3.632e-07`
   - *Offset*—`0`

© 2023 Cadence Design Systems, Inc.

2. Click the *Plot Eye* button to plot the eye diagram. The following eye diagram appears:

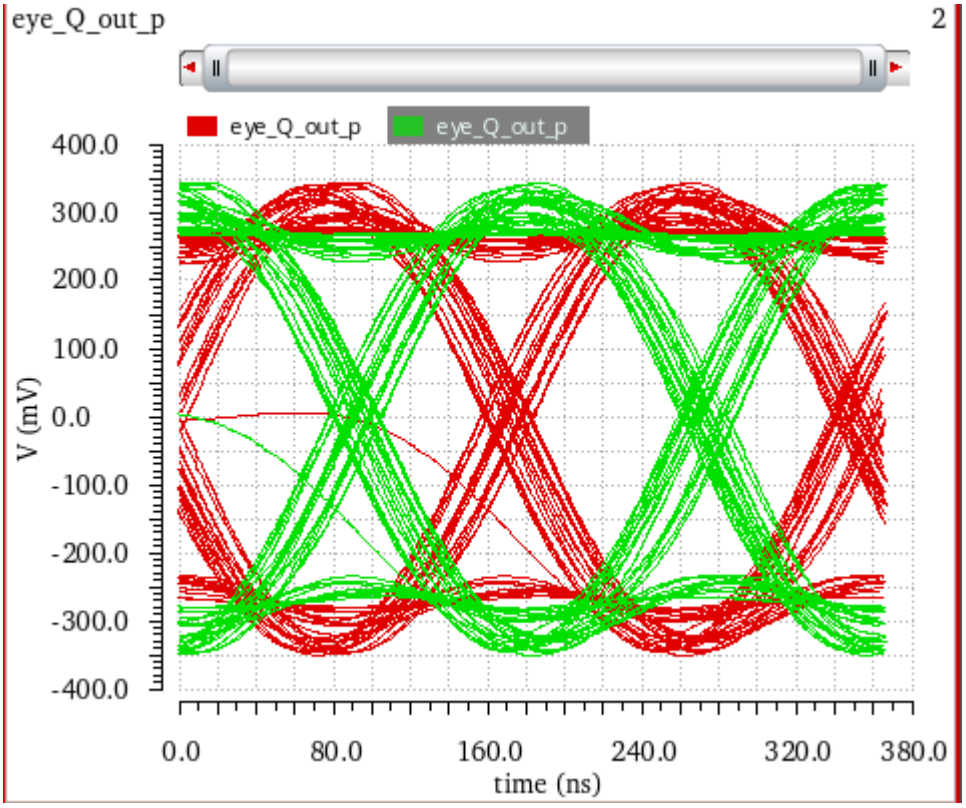


3. Now, select the *Center Eye* check box. When you select this check box, notice that the *Offset* field is disabled.

4. Select *Plot Mode* as `Append`.

© 2023 Cadence Design Systems, Inc.

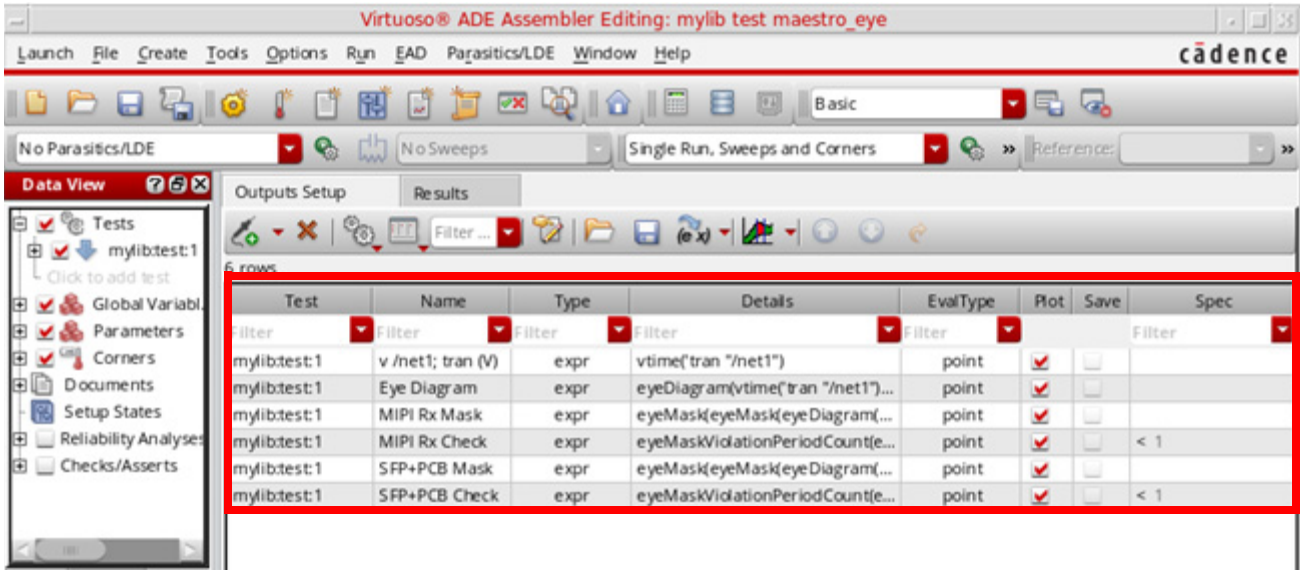
5. Click the *Plot Eye* button. The following eye diagram is generated. The new eye generated (in green) is placed in the center of the graph.



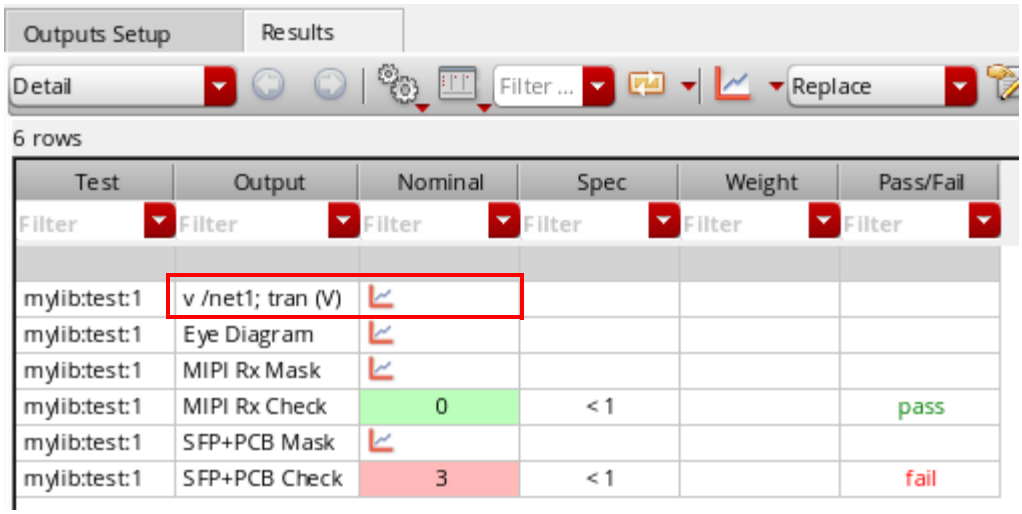
© 2023 Cadence Design Systems, Inc.

Example: Creating an Eye Diagram

Consider the following setup in ADE Assembler in which you run transient analysis with the given output expressions.



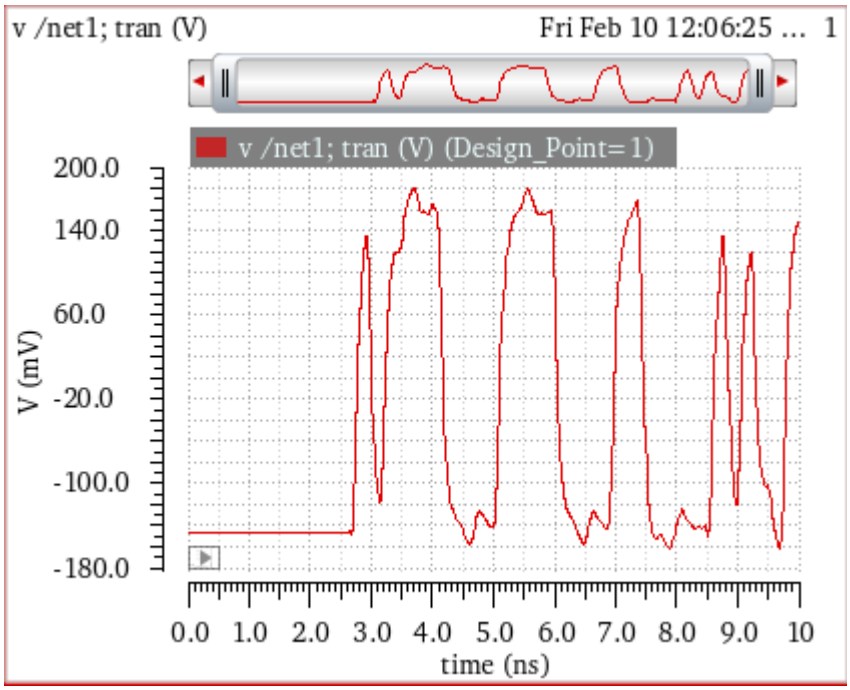
When you run the simulation for this setup, the output expressions are evaluated and the following output is displayed:



Perform the following steps to plot eye diagram:

© 2023 Cadence Design Systems, Inc.

1. Plot the `v/net1; tran (V)` output, as shown in the figure below.

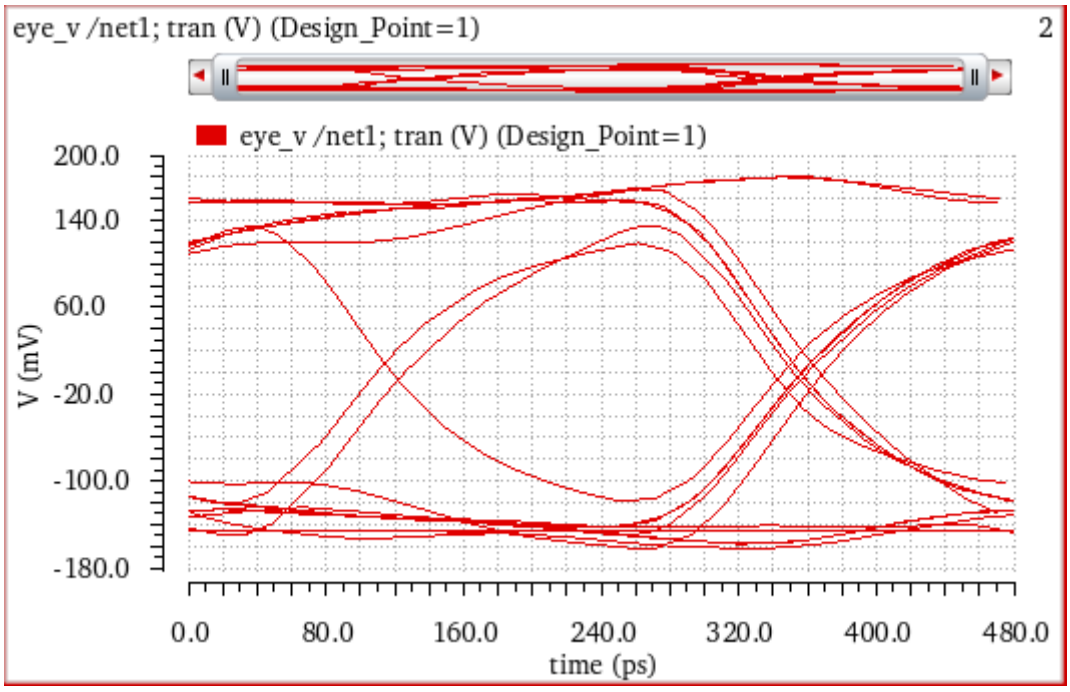


2. Now, create the eye diagram for this signal by choosing *Measurements – Eye Diagram*. Select `v/net1; tran (V) (Design_Point=1)` waveform and specify the following values in the Eye diagram fields:

   - *Start*=`0.0`, *Stop*=`10.00n`
   - *Eye Period*=`468.9p`
   - Select the *Center Eye* check box to create a centered eye diagram. This helps in carrying out the measurements accurately.
   - Click the *Plot Eye* button to plot the eye diagram for this signal.

© 2023 Cadence Design Systems, Inc.

The following eye diagram is plotted:



Setting Eye Diagram Properties

To change the trace properties of an eye diagram, right-click the eye diagram plot and choose *Trace Properties*. The *Eye Trace Properties for <eye_diagram_name>* form appears. It includes the following fields:

- *Name*—Specify the name of the eye diagram trace. If you select the *Default* check box, it displays the default trace name.
- *Expression*—Specify an expression that you can associate with the selected eye diagram.
- *Color*—Specify the foreground color of the trace.

© 2023 Cadence Design Systems, Inc.

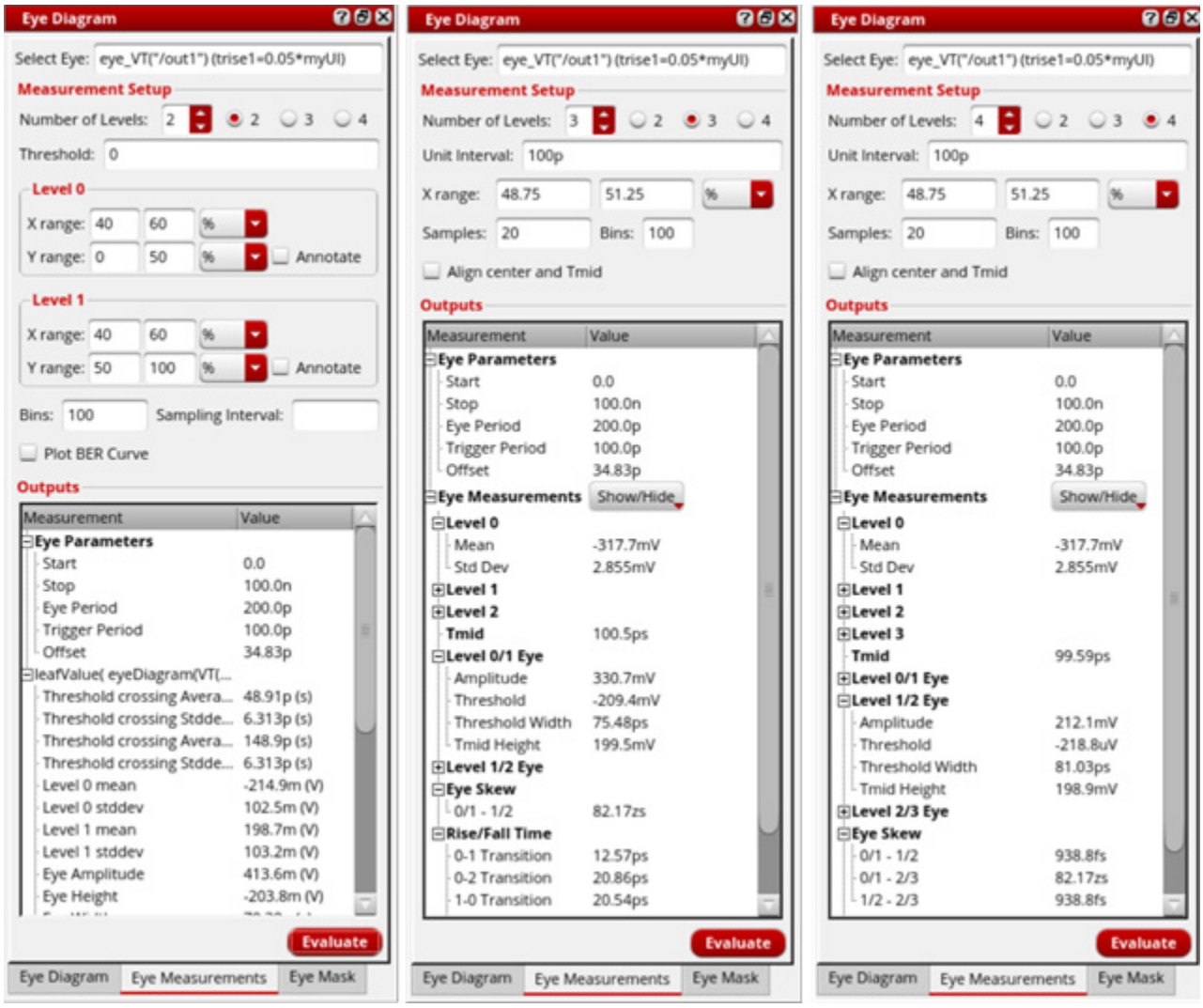
Working with the Eye Measurements Tab

After you plot the eye diagram, you can use the *Eye Measurement*s tab to calculate the measurements of the plotted eye and to plot the measurement values.

Click the *Eye Measurements* tab at the bottom of the Eye Diagram assistant. The fields displayed in the Eye Measurements tab depend on the number of voltage levels you select in the Number of Levels field.

The following figure shows the fields displayed for non-return-to-zero (NRZ), Pulse-Amplitude Modulation 3-Level (PAM3), and Pulse-Amplitude Modulation 4-Level (PAM4).

NRZ PAM3 PAM4



- *Select Eye*—Select the eye diagram on which you want to perform measurements.

© 2023 Cadence Design Systems, Inc.

- *Number of Levels*—Select the number of voltage levels, which also indicate the modulation schemes:
  - *2*:Non-Return-to-Zero (NRZ). This is the default modulation scheme.
  - *3*: Pulse-Amplitude Modulation 3-Level (PAM3).
  - *4*: Pulse-Amplitude Modulation 4-Level (PAM4).
- Depending upon the modulation scheme you selected, specify the following fields:

© 2023 Cadence Design Systems, Inc.

Section and Fields		Modulation Scheme NRZ Description	PAM3	PAM4
Measurement Setup	*Threshold*	Specify the threshold value. The Y-axis value, such as voltage, represents the switching threshold of the signal and is half the value of the signal range. This is used to compute the statistical information about the times (relative to the beginning of the eye diagram) at which the signal crosses the threshold level.	N/A	N/A
	Unit Interval	N/A	Specify the symbol period of the signal.	Specify the symbol period of the signal.
	X range	N/A	Specify the start and the end values of X-axis values in either % or time to determine the levels of the eye.	Specify the start and the end values of X-axis values in either % or time to determine the levels of the eye.

© 2023 Cadence Design Systems, Inc.

Section and Fields		Modulation Scheme NRZ Description	PAM3	PAM4
Measurement Setup	*Bins*	Specify the signal bins you want to display in the eye diagram plot. These signals bins are used to form the horizontal (threshold crossing times) and vertical (amplitude variation) histograms.	Specify the signal bins you want to display in the eye diagram plot. These signals bins are used to form the horizontal (threshold crossing times) and vertical (amplitude variation) histograms.	Specify the signal bins you want to display in the eye diagram plot. These signals bins are used to form the horizontal (threshold crossing times) and vertical (amplitude variation) histograms.
	Samples	N/A	Specify the number of samples to collect per period within the specified X range.	Specify the number of samples to collect per period within the specified X range.
	Align center and Tmid	N/A	Adjusts the offset of the eye so that T_{mid} becomes the midpoint of the eye period. T_{mid} is the midpoint of the maximum eye width.	Adjusts the offset of the eye so that T_{mid} becomes the midpoint of the eye period. T_{mid} is the midpoint of the maximum eye width.

© 2023 Cadence Design Systems, Inc.

Section and Fields		Modulation Scheme NRZ Description	PAM3	PAM4
Level 0	X range	Specify the X-range to plot the vertical histograms for level 0 measurements. Select the type as % or xval.	N/A	N/A
	Y range	Specify the Y-range to plot the vertical histograms for level 0 measurements. Select the type as % or yval.	N/A	N/A
	Annotate	Select this check box if you want to annotate level 0 on the specified eye.	N/A	N/A
Level 1	X range	Specify X-range to plot the vertical histograms for level 1 measurements. Select the type as % or xval.	N/A	N/A
	Y range	Specify Y-range to plot the vertical histograms for level 1 measurements. Select the type as % or yval.	N/A	N/A
Level 1	Annotate	Select this check box if you want to annotate level 1 on the specified eye.	N/A	N/A

© 2023 Cadence Design Systems, Inc.

	Modulation Scheme		
Section and Fields	**NRZ**	**PAM3**	**PAM4**
	Description		
Sampling Interval	Specify the time interval after which the signals are divided in the eye diagram plot. If this field is left blank, the data within the level 1 and level 0 regions are used to analyze the amplitude variation of the signal. This means there is some sensitivity to the original spacing between the data points in the signal, which is caused by the variable time steps in the simulator. If the points are clustered in the curve portion, the distribution can be skewed. To perform the analysis, the sampling interval you specify in this field is divided into even time points.	N/A	N/A

© 2023 Cadence Design Systems, Inc.

Section and Fields	Modulation Scheme NRZ Description	PAM3	PAM4
Plot BER Curve	Select this check box if you want to plot the Bit Error Rate (BER) curve. **Note:** When you export the BER expression in the VCSV format and re-plot it, the re-plotted trace is different because VCSV format assigns a Y unit to the Y-axis. Therefore, new trace is not plotted on the same axis as compared to the trace plotted from the measurements. To make the trace look the same, set the new Y-axis to use log-scale and have the same range as the axis of the original plots.	N/A	N/A

- *Evaluate*—Click thi*s* button to evaluate the eye measurements.

  After the evaluation is complete, depending on the modulation scheme you selected, the following eye measurements are displayed in the *Output*s section:

© 2023 Cadence Design Systems, Inc.

NRZ	PAM3 and PAM4
Eye Parameters: ■ Start ■ Stop ■ Eye Period ■ Trigger Period ■ Offset	**Eye Parameters:** ■ Start ■ Stop ■ Eye Period ■ Trigger Period ■ Offset

© 2023 Cadence Design Systems, Inc.

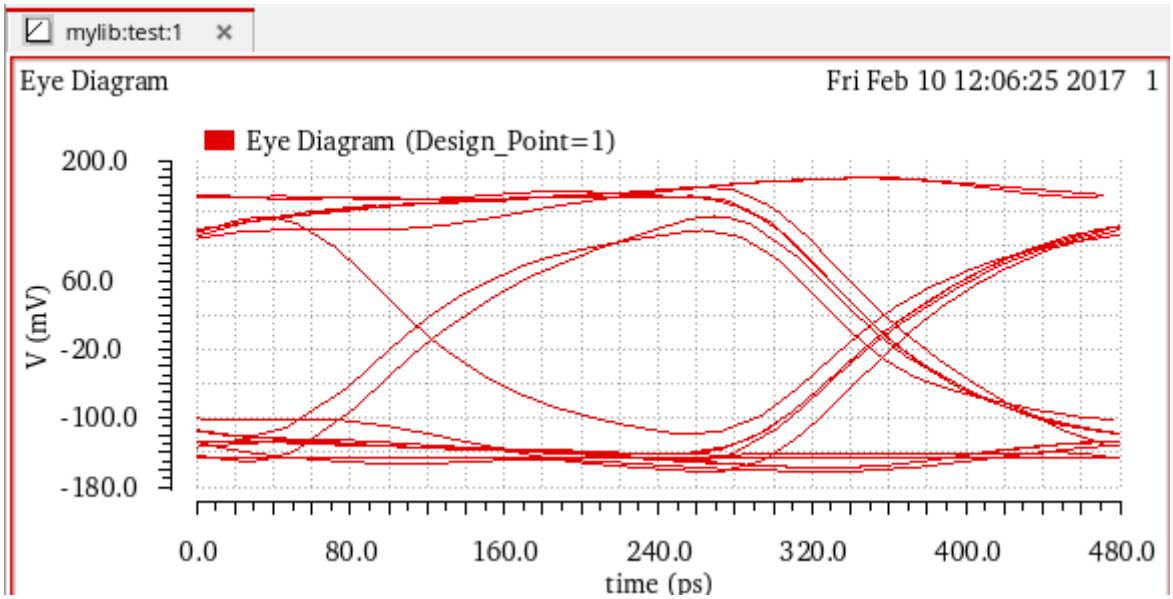
NRZ	PAM3 and PAM4
Eye measurements: Threshold crossing Average (left) Threshold crossing Stddev (left) Threshold crossing Average (right) Threshold crossing Stddev (right) Level 0 mean Level 0 stddev Level 1 mean Level 1 stddev Eye Amplitude Eye Height Eye Width Max Eye Height Time Max Eye Height Max Eye Width Threshold Max Eye Width Eye S/N Eye Rise Time Eye Fall Time Random Jitter (left) Peak-to-peak Jitter (left) Random Jitter (right) Peak-to-peak Jitter (right) Deterministic Jitter	**Eye measurements:** ■ Mean ❑ For PAM3: Mean for levels 0, level 1, and level 2 ❑ Form PAM4: Mean for levels 0, level 1, level 2, and level 3 ■ Std Dev ❑ For PAM3: Std Dev for levels 0, level 1, and level 2 ❑ Form PAM4: Std Dev for levels 0, level 1, level 2, and level 3 ■ T_{mid} ■ Amplitude, Threshold, Threshold Width, and T_{mid} Height ❑ For level 0/1 eye and level 1/2 eye in PAM3 ❑ For level 0/1 eye, level 1/2 eye, and level 2/3 eye in PAM4 ■ Eye Skew ❑ For PAM3: 0/1-1/2 ❑ For PAM4: 0/1-1/2, 0/1/-2/3, and 1/2-2/3 ■ Level Separation Mismatch Ratio ■ Rise/Fall Time

See the NRZ Eye Measurements and PAM4 Eye Measurements, and Eye Measurements for Different Levels sections for more information.

© 2023 Cadence Design Systems, Inc.

Example 1: Calculating Measurements and Plotting the BER Curve

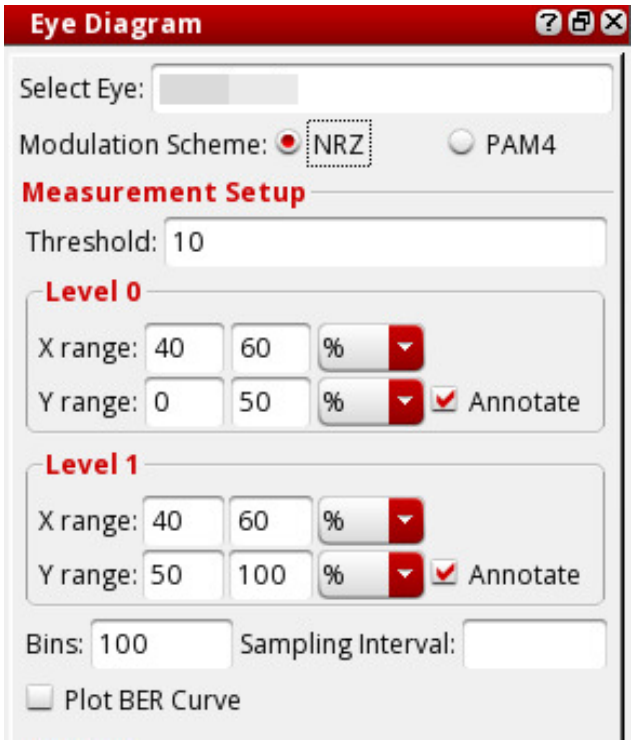
Consider the following eye diagram from the setup shown in the Example: Creating an Eye Diagram section. Ensure that the eye diagram is centered so that measurements are calculated accurately.



Perform the following steps to plot the eye diagram, calculate the measurement values, and plot the BER curve for the eye diagram shown in the figure below:

© 2023 Cadence Design Systems, Inc.

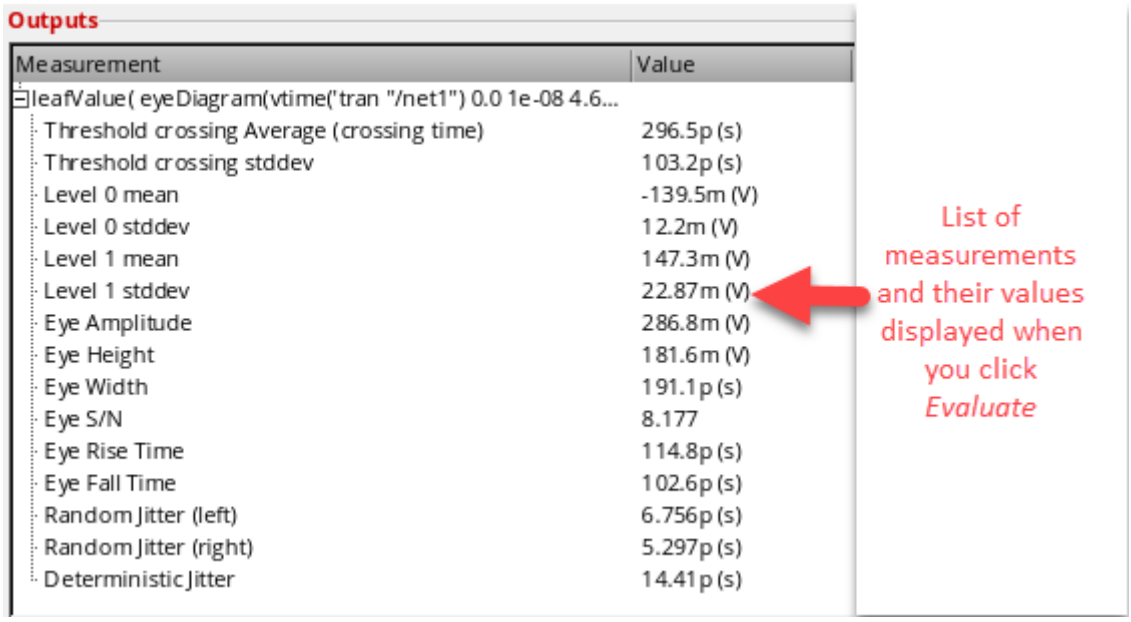
1. Click the *Eye Measurements* tab displayed at the bottom of this assistant. The fields displayed in the figure below appear:



2. Specify the following field values:
   - Select the eye diagram waveform in the graph window.
   - Select `NRZ` as the modulation scheme.
   - Specify threshold value as `0`.
   - Select the *Plot BER Curve* check box.

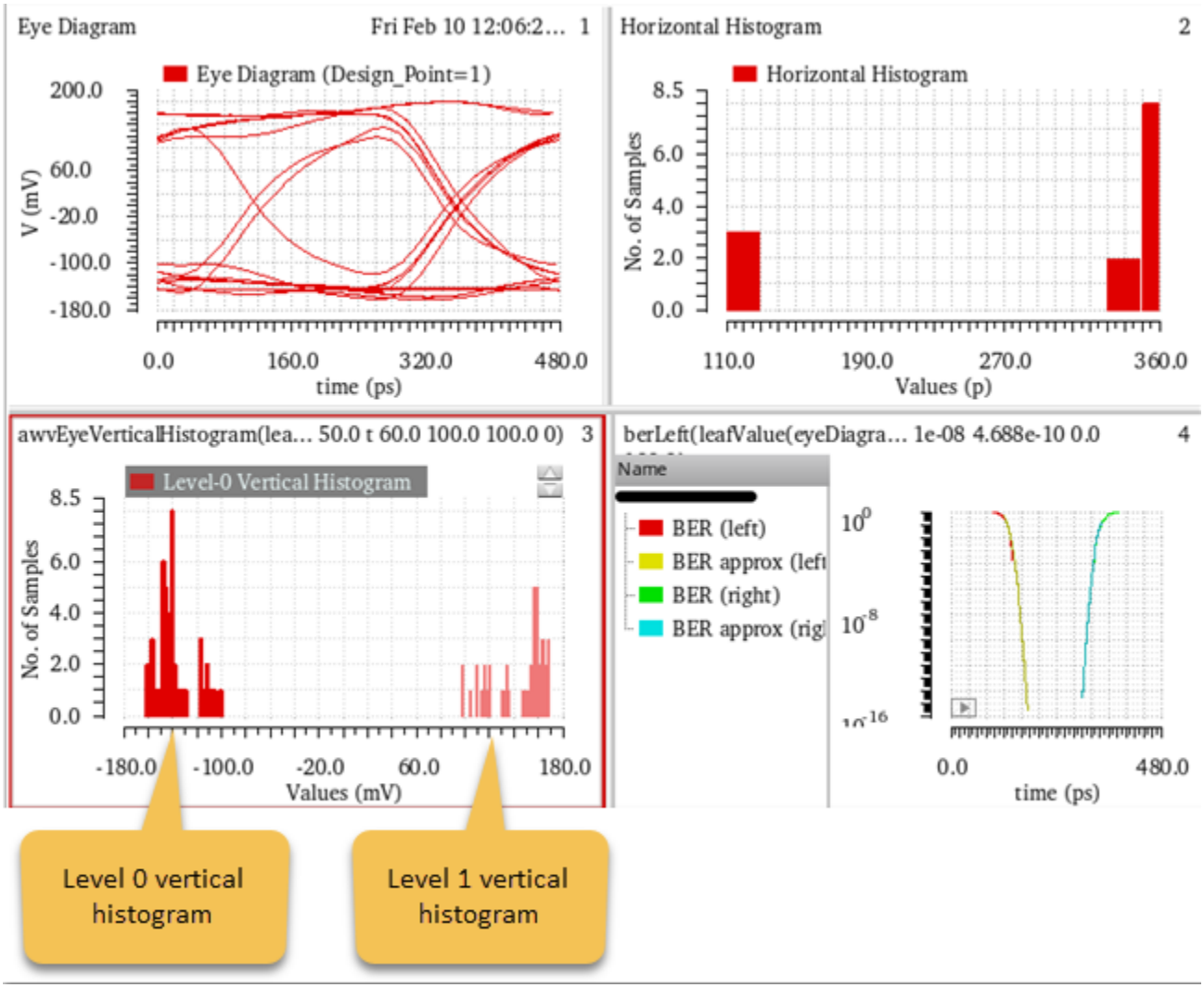
© 2023 Cadence Design Systems, Inc.

The *Outputs* section at the bottom displays the measurement values that are calculated when you press the *Evaluate* button.



© 2023 Cadence Design Systems, Inc.

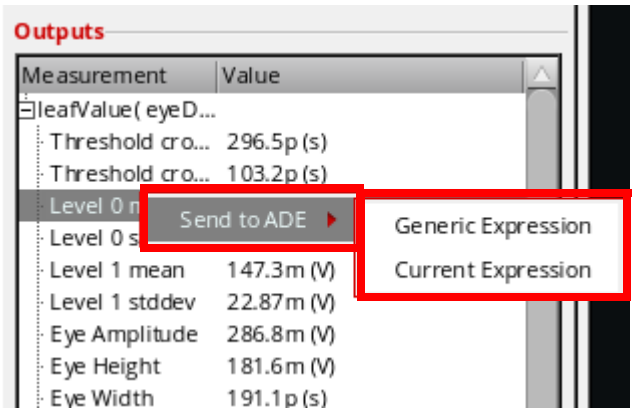
The plots for the horizontal histogram, level 0 and level 1vertical histograms, and BER curve appear in the graph window, as shown in the figure below.



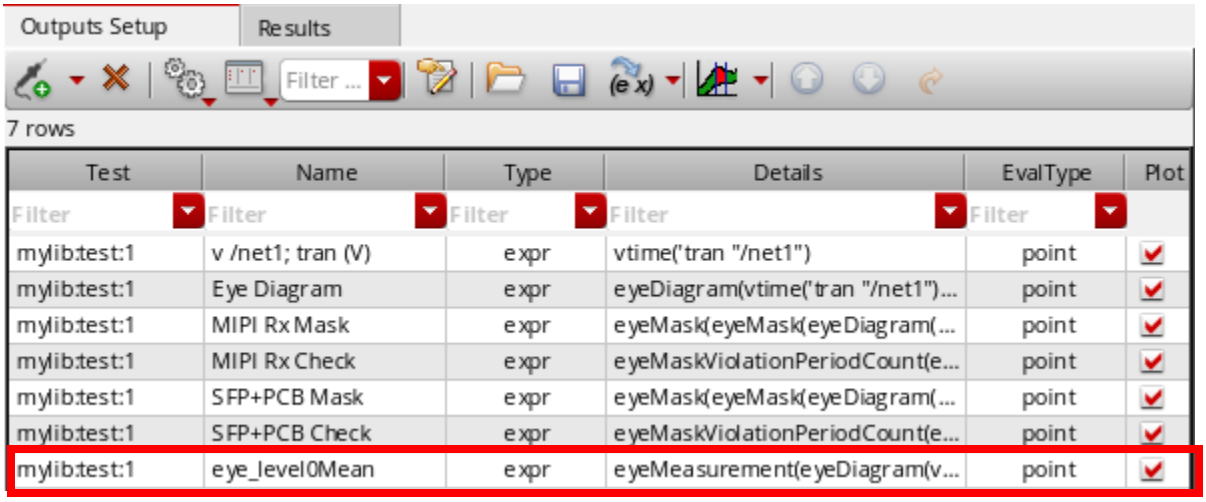
Note: The eye width and jitter measurements cannot be calculated and an error message is displayed when more than two peaks are detected in the horizontal histogram for the specified eye. To get the best measurements, it is recommended to select a centered eye diagram with an eye period of 2 UI.

© 2023 Cadence Design Systems, Inc.

You can also send these advanced measurement values to the ADE window. To do this, right-click a measurement value and choose *Send To ADE – Current Expression* / *Generic Expression*.



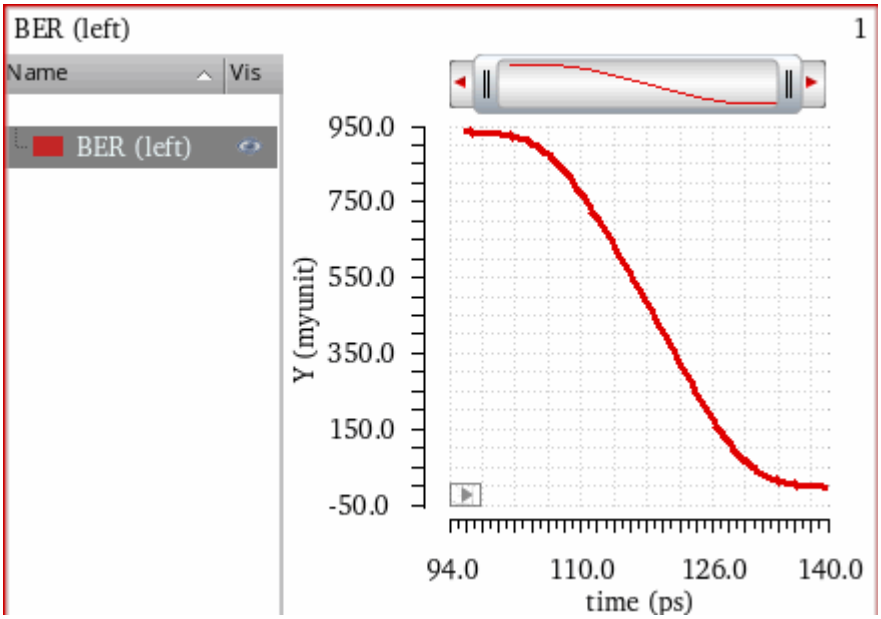
The expression for the selected measurement appears in the Virtuoso ADE Assembler as shown in the figure below:



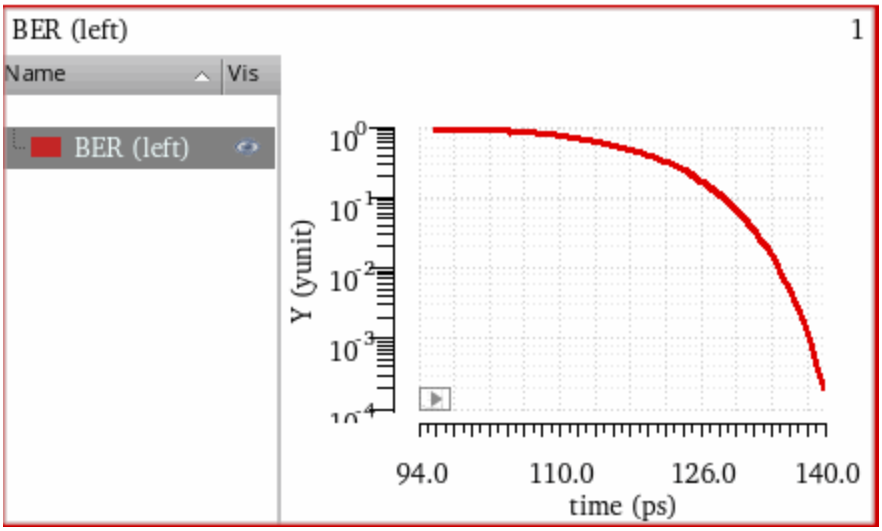
If you save the waveform for the BER expression in the VCSV format and the reload the waveform, the VCSV format assigns a `Y (myunit)` axis to Y-axis, as shown in the figure

© 2023 Cadence Design Systems, Inc.

below. Therefore, the new trace looks different as compared to the original trace plotted from the measurements.



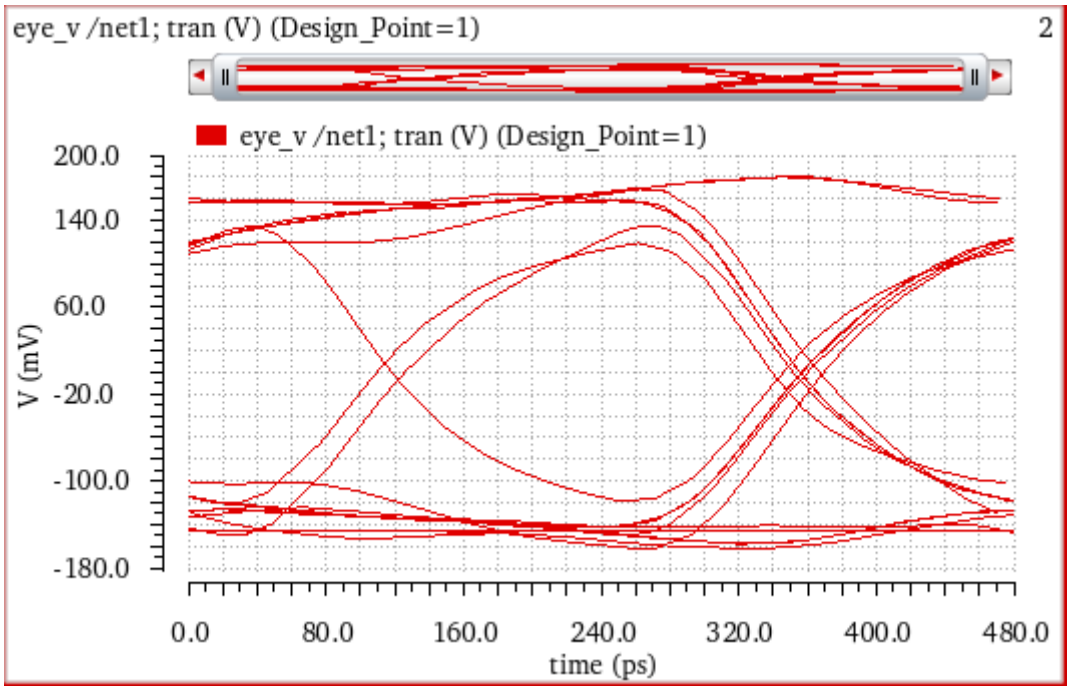
If you want the new trace to look same as that of the original trace, right-click the Y-axis and choose scale as *Log Scale*, and specify the same range as given in the original plot.



© 2023 Cadence Design Systems, Inc.

Example 2: Viewing Annotations for Level 0 and Level 1

Consider the following eye diagram plotted for `v/net1; tran (V)` signal, as shown in the Example: Creating an Eye Diagram section. Ensure that the eye diagram is centered so that measurements are calculated accurately.

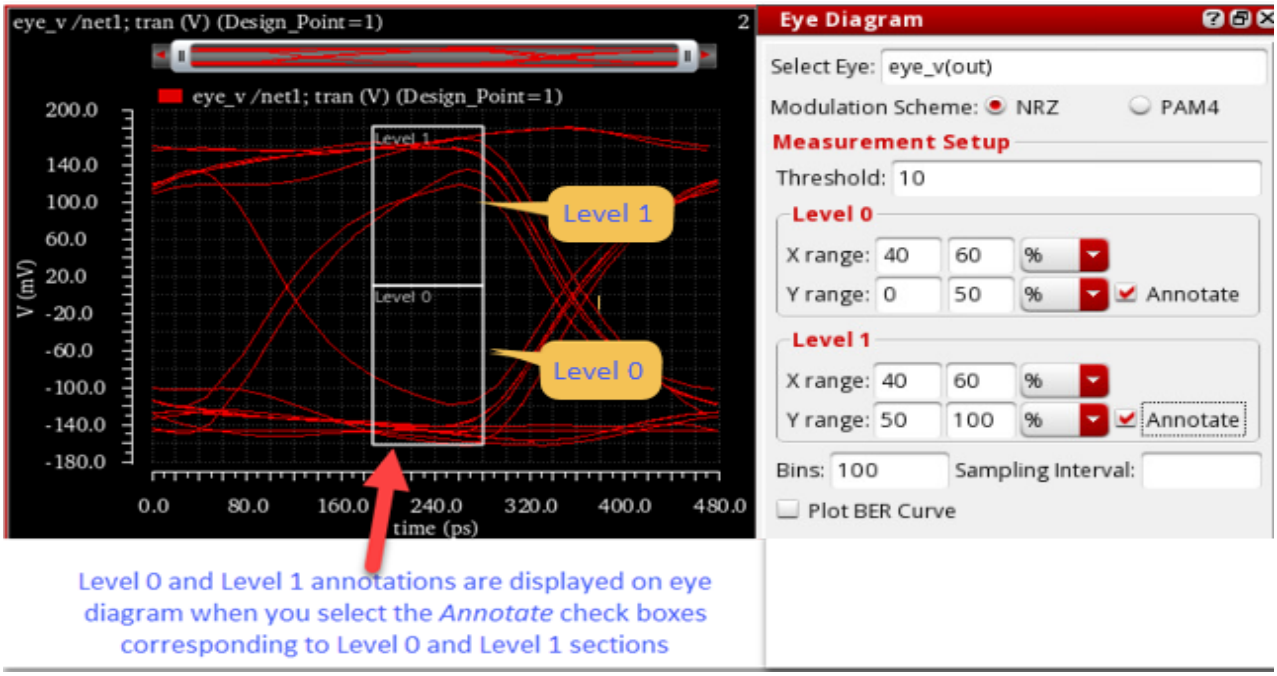


Perform the following steps to plot the eye diagram, calculate the measurement values, and view level 0 and level 1 annotations on the eye diagram:

1. Open the *Eye Measurements* tab.
2. Select the eye diagram you plotted and specify the threshold as `0`.

© 2023 Cadence Design Systems, Inc.

3. Click the *Annotate* check boxes corresponding to *Level 0* and *Level 1* sections to view annotations on the eye diagrams (as shown in the figure below).



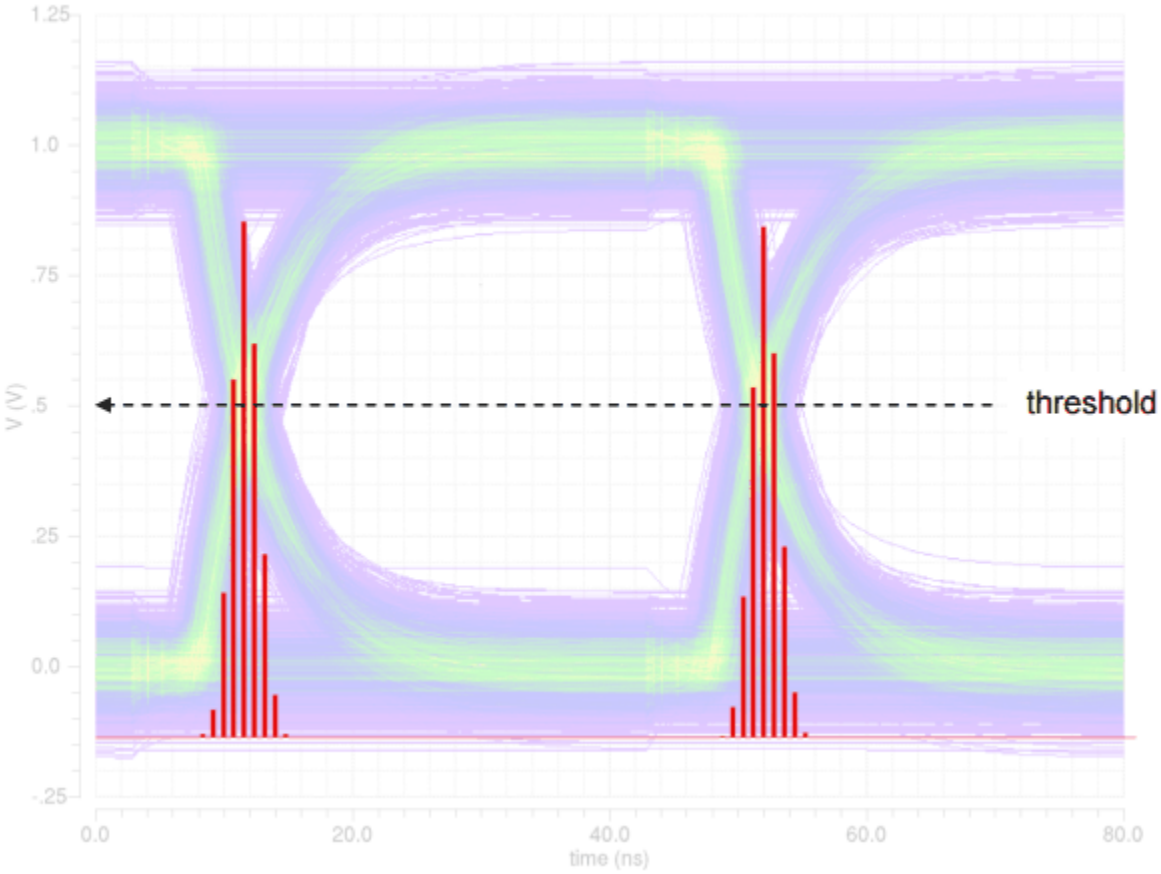
Computing Advanced Measurements

The Advanced Analysis is computed based on the following two measurement groups—horizontal histograms and vertical histograms. In the figure below, 3 sigma is used for the extent of the data as this represents 99.7% of the values in a normal distribution.

© 2023 Cadence Design Systems, Inc.

Measurements based on Horizontal Histograms

A horizontal histogram is generated based on the analysis of variation of the times at which the signal crosses the specified threshold level, as shown in the diagram below:



When you click the *Evaluate* button, the resulting horizontal histogram is plotted. By analyzing the mean and standard deviation of the transition times relative to the beginning of each period, various metrics are computed. The results for the following computations are displayed:

- Threshold crossing stddev—Measures the jitter of the signal. It is accurately computed only when there is a single transition region in the eye diagram because it is analyzed over the entire period, and therefore, it gives the overall standard deviation of both transition regions (as shown in the figure above).
- Threshold crossing average—Similar to the threshold crossing stddev, it is computed over the entire period.

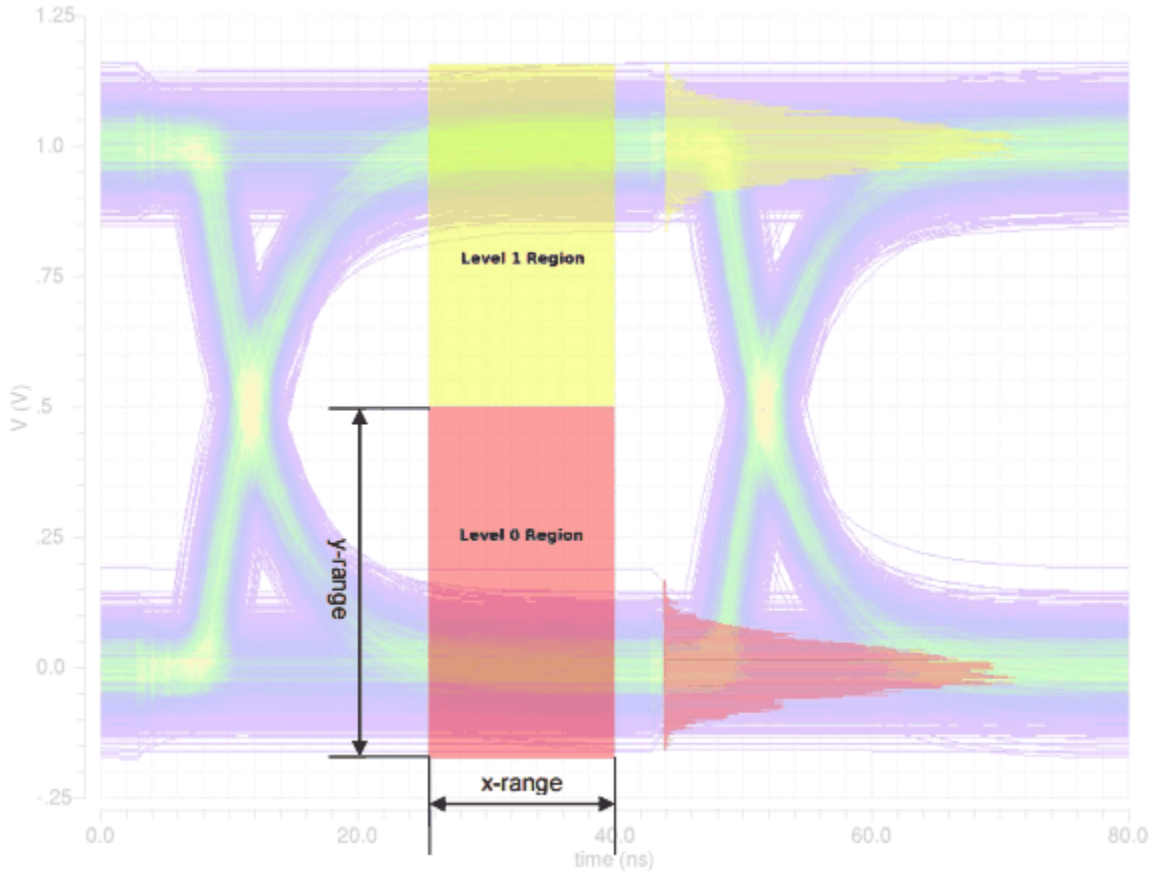
© 2023 Cadence Design Systems, Inc.

- Eye width—Represents the opening of the eye in X direction. It is computed as:

$$(Mean_{transition2} - 3\sigma_{transition2}) - (Mean_{transition1} + 3\sigma_{transition1})$$

Measurements based on Vertical Histograms

The vertical histogram is created based on the analysis of variation of the signal value in the level 0 and level 1 regions specified in the assistant, as shown in the diagram below:



The level 0 and level 1 regions are those specified by the X and Y ranges in the Eye Diagram assistant. All the Y-points in the waveform within these regions are used to perform the statistical analysis, unless they have been sampled first according to the Sampling Interval field. In the plotted vertical histograms in the above diagram, X-axis represents the amplitude variation.

© 2023 Cadence Design Systems, Inc.

Advanced Options Results

The calculations for the threshold crossing assume that only a single transition is used; therefore, one additional metric, eye risetime, is computed. In the eye risetime, two thresholds are created at 20% and 80% points between level 0 mean and level 1 mean. At each of these two thresholds, a horizontal histogram is computed (that indicates an analysis of the times of the crossing points of these two thresholds), and the resulting risetime is the difference in the mean crossing time at each of these two thresholds.

Horizontal and Vertical Eye Width

The eye width represents the opening of the eye in the X direction and is computed as:

```
(Meantransition2 - 3*std(transition2)) - (Meantransition1 + 3*std(transition1))
```

Consider a waveform that is folded on the X-axis by the length `n_period` between the start time `n_start` and stop time `n_stop`. The following steps are performed to calculate the horizontal eye opening:

- Calculates all the X points where the folded waveform intersects the line `y = yMid`

  where, `yMid = ymax(o_waveform)+ min(o_waveform))/2`
- From these calculated X points, returns the two consecutive X points that have the maximum distance between them as the horizontal eye opening.

  `X[k]` and `X[k-1]` for which the `(X[k]-X[k-1])` value is maximum.

The function performs the following steps to calculate the vertical eye width:

- Calculates the horizontal eye width to find the consecutive X points, `X[k]` and `X[k-1]` having maximum distance between them.
- Calculates all the Y points where the folded waveform intersects the following line: `x = (X[k]-X[k-1])/2`
- From these calculated Y points, returns two consecutive Y points that have the maximum distance between them as the vertical eye opening.

  `Y[k]` and `Y[k-1]` for which `(Y[k]-Y[k-1])` is maximum.

Assumptions

The following assumptions have been made while calculating the advanced option values:

- The opening of an eye approximately lies on the following:

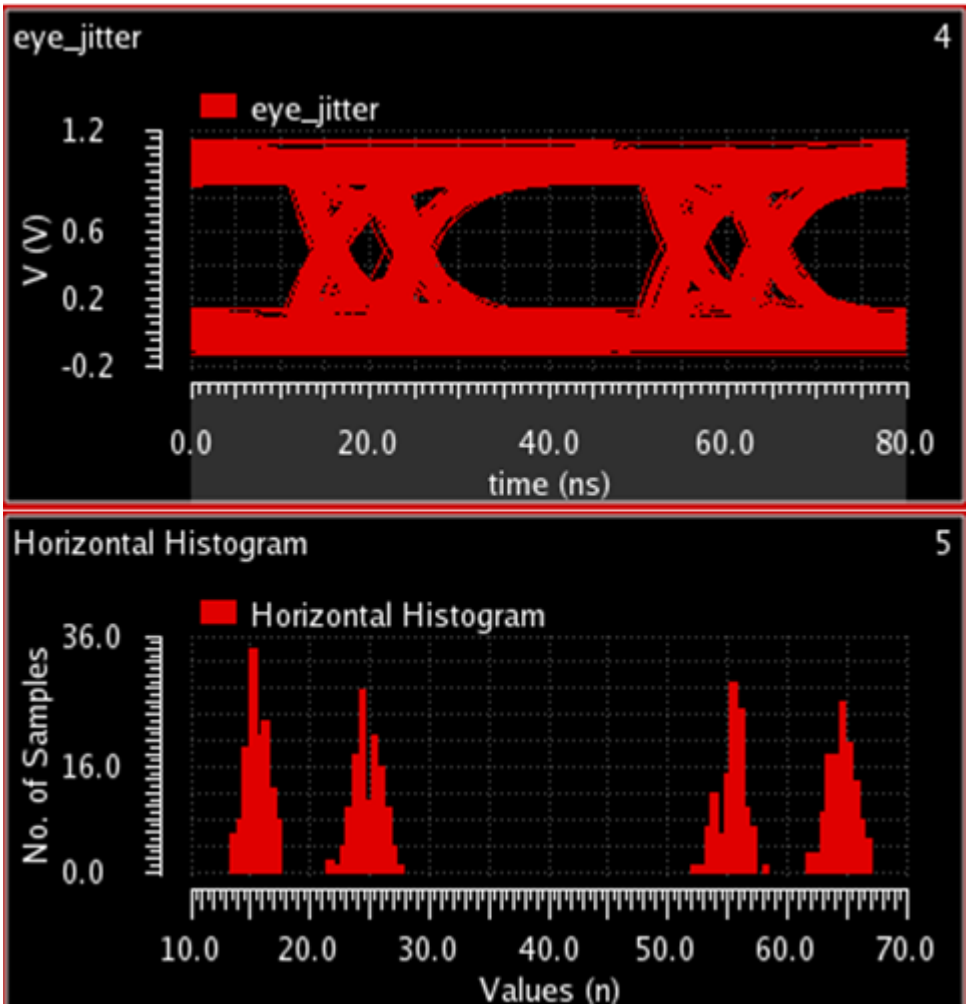
© 2023 Cadence Design Systems, Inc.

```
(ymax(o_waveform)+ymin(o_waveform))/2
```

- Only one eye opening exists in the area mentioned above in which the waveform is folded.

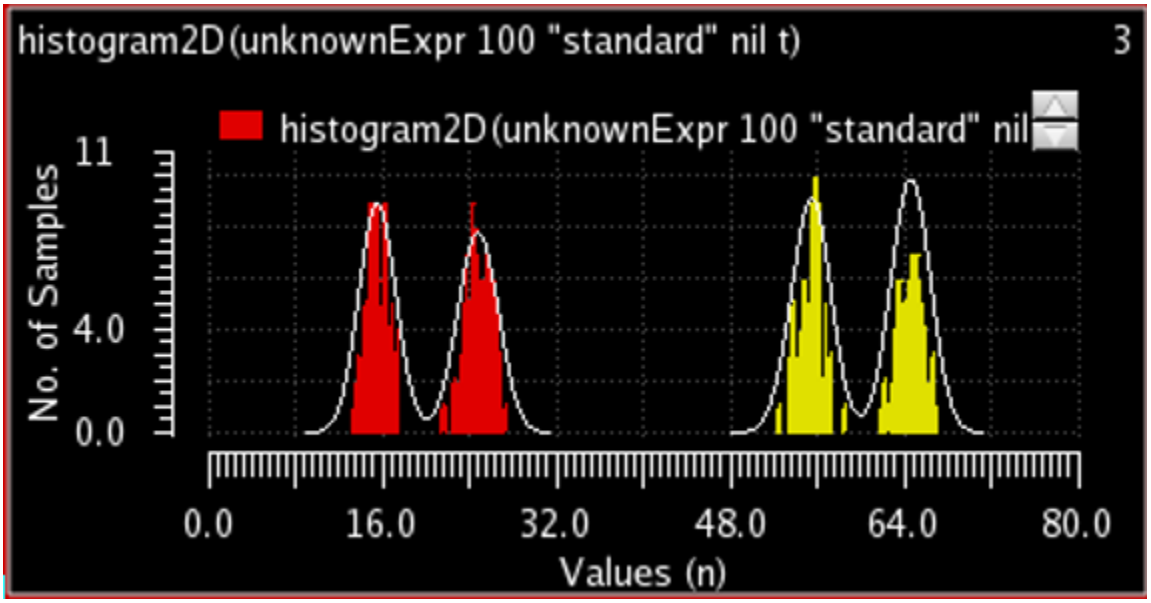
Computing the Jitter and BER Values

Consider the following histograms created for the `eye_jitter` waveform to detect the crossing areas.



© 2023 Cadence Design Systems, Inc.

The following histograms are created for each side and density estimation is applied to fit each histogram to a distribution J(x), as shown in the figure below:



- If deterministic jitter is present, each side's distribution is expected to have two peaks, and the deterministic jitter is the time difference of the peaks.
- Random jitter is the standard deviation of the tails of the distribution.
- BER curves are calculated as follows:
  - Left BER plot

    Integrate left distribution from x to infinity

    $$BER_L(x) = \int_x^{\infty} J_L(x')\,dx'$$

    The mean and standard deviation of the right-side tail of the left distribution is used to estimate beyond the range of the histogram.
  - Right BER plot

    Integrate right distribution -infinity to x

    $$BER_R(x) = \int_{-\infty}^{x} J_R(x)\,dx$$

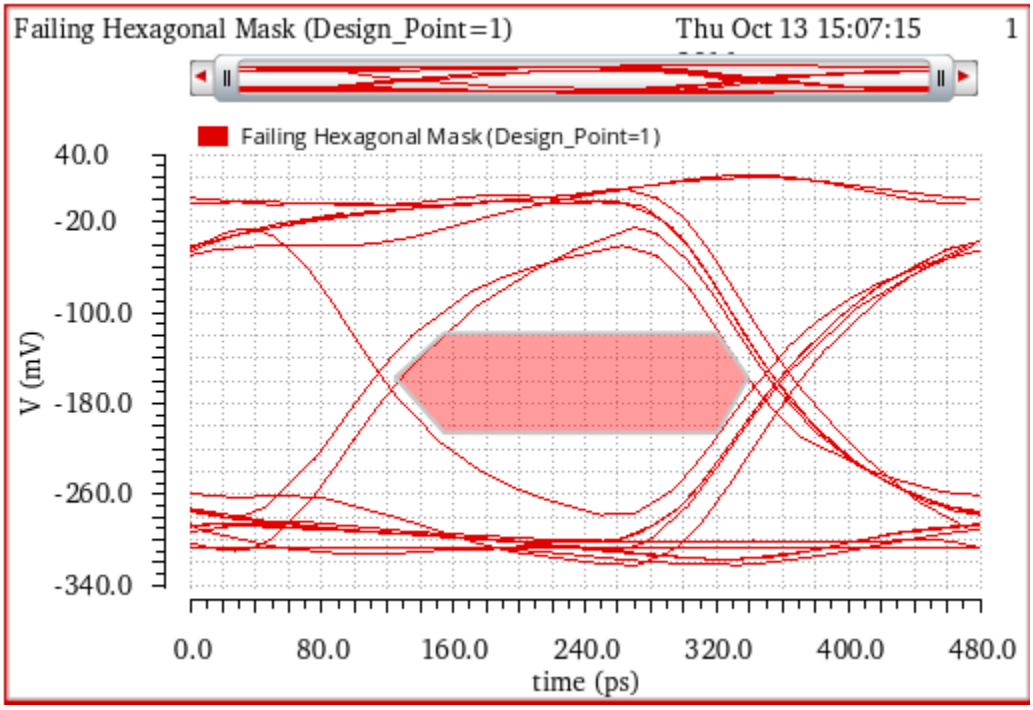
© 2023 Cadence Design Systems, Inc.

The mean and standard deviation of the left-side tail of the right distribution is used to estimate beyond the range of the histogram.

© 2023 Cadence Design Systems, Inc.

Working with the Eye Mask Tab

You can evaluate the eye diagram against any violation by using a compliance mask overlaid on the eye diagram waveform. To create an eye mask for the selected eye diagram, you can use the *Eye Mask* tab available in the Eye Diagram assistant. The eye mask can be of the following geometries: rectangular, hexagonal, diamond, and custom shaped defined in the VCSV file. For example, the figure below shows a hexagonal eye mask.



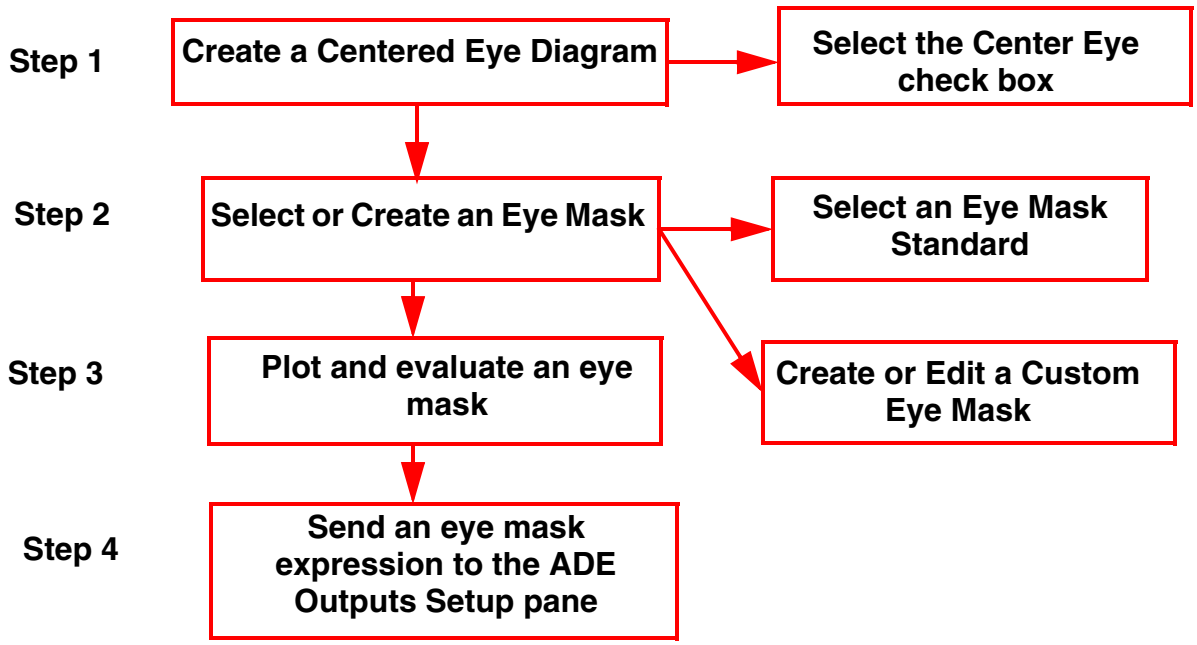
Video

To know more about how to use eye masks, view the ViVA - Using Eye Masks video available on Cadence Online Support.

In addition, you can read an interesting blog about eye masks, see Virtuoso Video Diary: Eye Masks.

© 2023 Cadence Design Systems, Inc.

The flow chart below describes how you can create and plot the eye mask for an eye diagram.



This section includes the following topics:

- Creating a Centered Eye Diagram
- Selecting or Creating an Eye Mask
  - Method 1: Creating an Eye Mask Using Existing Eye Mask Standards
  - Method 2: Creating an Eye Mask Using the Given Shape
  - Method 3: Creating an Eye Mask Based on the Existing Eye Mask
- Plotting and Evaluating an Eye Mask
- Sending an Eye Mask Expression to ADE
- Editing an Eye Mask
- Performing Various Tasks on an Eye Mask
- Example of Saving and Importing an Eye Mask in VCSV Format

Creating a Centered Eye Diagram

Perform the following steps to select or create the centered eye diagram:

© 2023 Cadence Design Systems, Inc.

1. Choose *Measurements – Eye Diagram* to open the Eye Diagram assistant.

   Two tabs appear at the bottom—*Eye Diagram* and *Eye Mask*.

2. Click the *Eye Diagram* tab to first create an eye diagram. For more information about how to use the fields, see Eye Diagram Assistant on page 242.

Note: The original eye diagram should be centered before adding the eye mask. For details about how to shift the eye diagram to the center, see Shifting Eye Diagram to Center on page 258.

Selecting or Creating an Eye Mask

There are three available methods that you can use to create an eye mask. In the first method, you can create a new eye mask based on the given shape standards. To do this, select an eye mask from the given eye mask standards, add the updated mask to the list, and save it in the VCSV file if required.

In the second method, you can select a mask geometry (shape) and then edit the X- and Y-range values and vertex points (corners of the eye mask). Then, add it to the mask list and save it in the VCSV file if required.

In the third method, you can create an eye mask using an existing eye mask, then edit the eye mask shape or coordinates, add the eye mask to the mask list, and save it in the VCSV file if required.

For detailed information about these three methods of creating an eye mask, see the topics below:

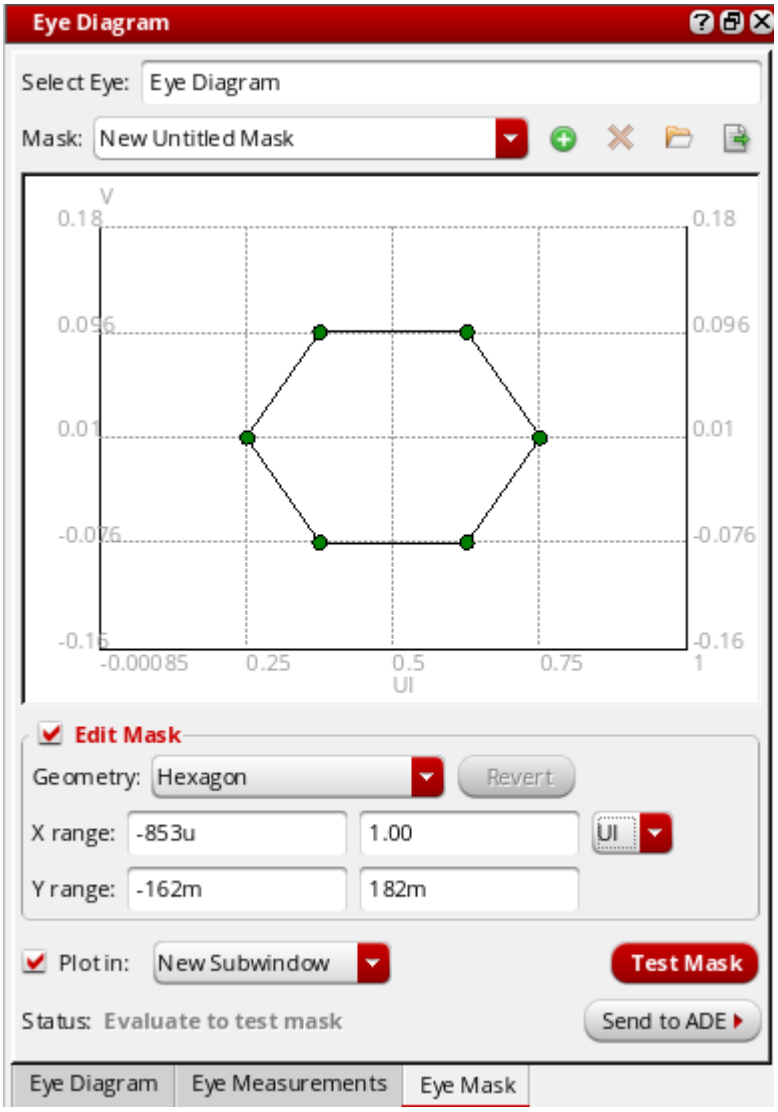
- Method 1: Creating an Eye Mask Using Existing Eye Mask Standards
- Method 2: Creating an Eye Mask Using the Given Shape
- Method 3: Creating an Eye Mask Based on the Existing Eye Mask

Method 1: Creating an Eye Mask Using Existing Eye Mask Standards

You can select a given eye mask standard to create a new eye mask. To do this, perform the following steps:

© 2023 Cadence Design Systems, Inc.

1. Click the *Eye Mask* tab to view the eye mask options. The figure below shows the fields available under the *Eye Mask* tab.

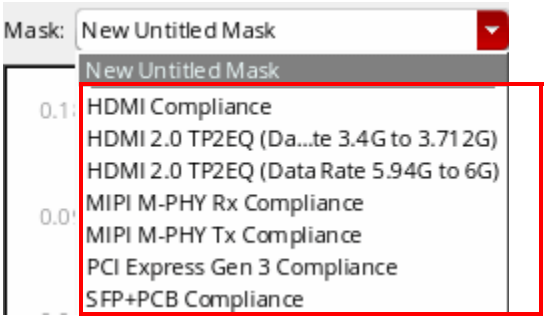


2. Now, select an eye diagram in the graph window on which you want to create the eye mask. The selected eye diagram is displayed in the *Select Eye* field.

3. In the *Mask* drop-down list, select an industry standard eye mask template that you want to use.The available options are: `HDMI Compliance`, `HDMI 2.0 TP2EQ (Data Rate 3.4G to 3.712G)`, `HDMI 2.0 TP2EQ (Data Rate 5.94G to 6G)`, `MIPI M-PHY Rx Compliance`, `MIPI M-PHY Tx Compliance`, `PCI Express Gen 3 Compliance`, and `SFP+PCB Compliance`.

© 2023 Cadence Design Systems, Inc.

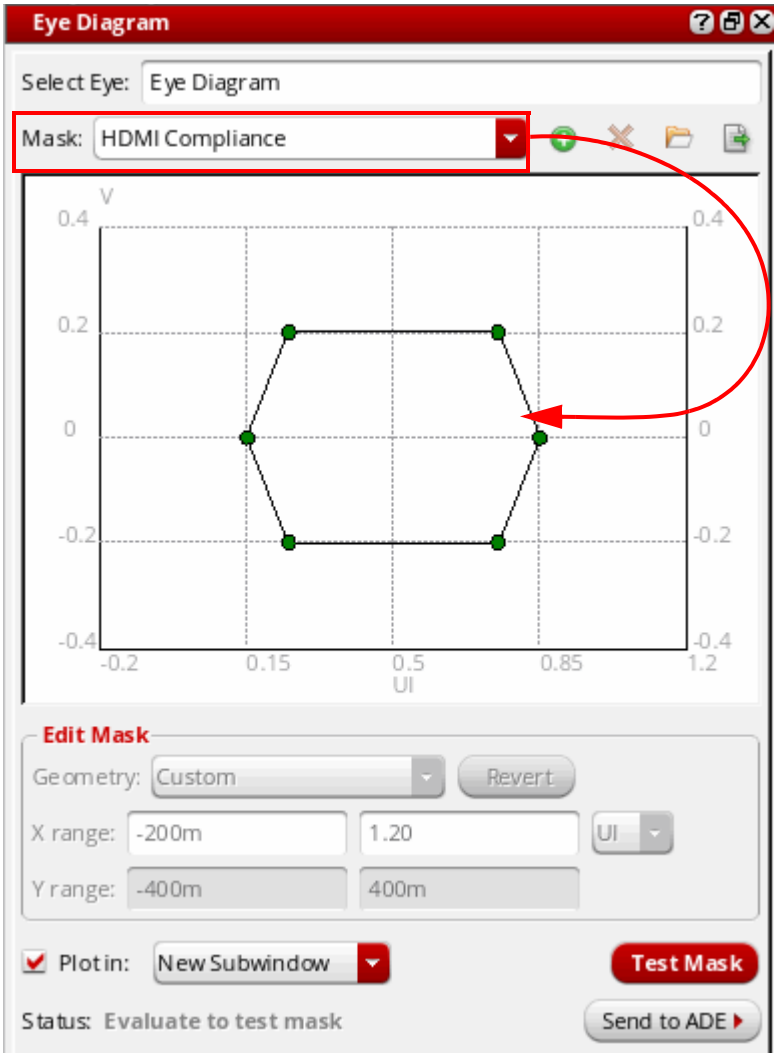
By default, this drop-down displays the `New Untitled Mask` text.



The selected eye mask is displayed in the mask editing area, as shown in the figure below. The X-axis of this eye mask displays the unit intervals (UI) and the Y-axis displays

© 2023 Cadence Design Systems, Inc.

voltage (V). In addition, the eye mask shape and coordinates vary according to the template you select.



4. After selecting the eye mask, you can plot and evaluate the eye mask with the eye diagram and send the expression to ADE outputs. For more information, see Plotting and Evaluating an Eye Mask on page 296.

5. Edit the eye mask if required. The Edit Mask section is disabled when you select an industry standard eye mask, which means standard eye masks cannot be edited directly in the drop-down, but you can create a copy of the standard eye mask and edit it. To do this, select a standard eye mask from the Mask drop-down list and click the Add current mask to list button to create a copy of the selected mask. Specify a name for this new mask in the Mask drop-down list and then edit this mask. For more information, see Editing an Eye Mask on page 300.

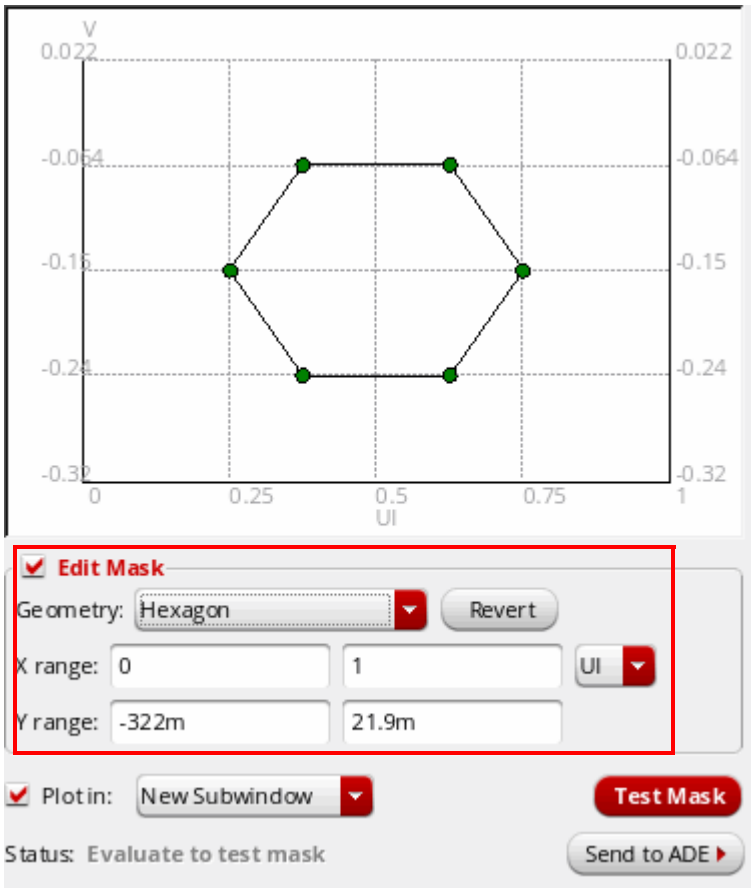
© 2023 Cadence Design Systems, Inc.

6. After editing, you can rename the eye mask and add it to the *Mask* drop-down list. You can also save the eye mask in the VCSV file. For more information, see Performing Various Tasks on an Eye Mask on page 302.

Method 2: Creating an Eye Mask Using the Given Shape

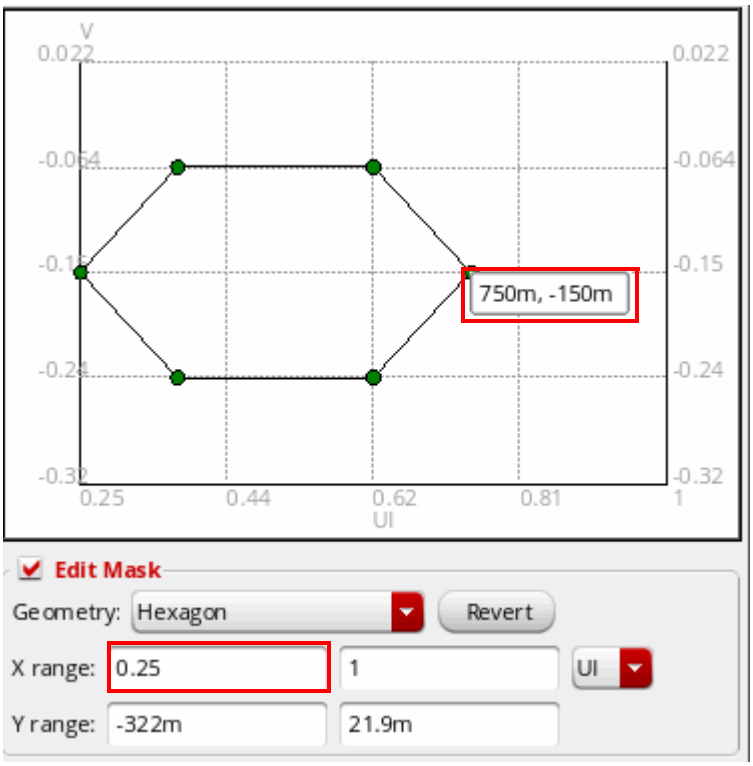
Perform the following steps to create an eye mask using the existing shape:

1. Select `Untitled Mask` from the *Mask* drop-down. If you want to create an eye mask using the standard eye mask shape, select the standard mask from the *Mask* drop-down and add it to the list.
2. On the *Eye Mask* tab, enable the *Edit Mask* check box.
3. Select a shape from the *Mask Geometry* drop-down and then change the X Range and Y Range coordinates if required. For more information about how to use the edit mask fields, see Editing an Eye Mask on page 300.
4. Now, click the vertex points in the mask editing area and edit the values. For example, consider you select a hexagon shape as shown in the figure below:



© 2023 Cadence Design Systems, Inc.

Now, change the X Range value from `0.25` to `1`. The mask position is changed as shown in the figure below. To edit the vertex point, click any of the vertices and type new values in the label. For example, in the figure below, the vertex value is changed from `750m, -150m` to `700m, -150m`.



5. Now, save the edited eye mask if required. For information about how to add the eye mask to the Mask drop-down list, and how to save the eye mask, see Performing Various Tasks on an Eye Mask on page 302.
6. After the eye mask is created, you can click the *Test Mask* button to test the eye mask by plotting it on the specified eye diagram in the graph window.
7. You can also plot and evaluate the eye mask with the eye diagram and send expression to ADE outputs. To know about how to plot the eye mask, see Plotting and Evaluating an Eye Mask on page 296.

Method 3: Creating an Eye Mask Based on the Existing Eye Mask

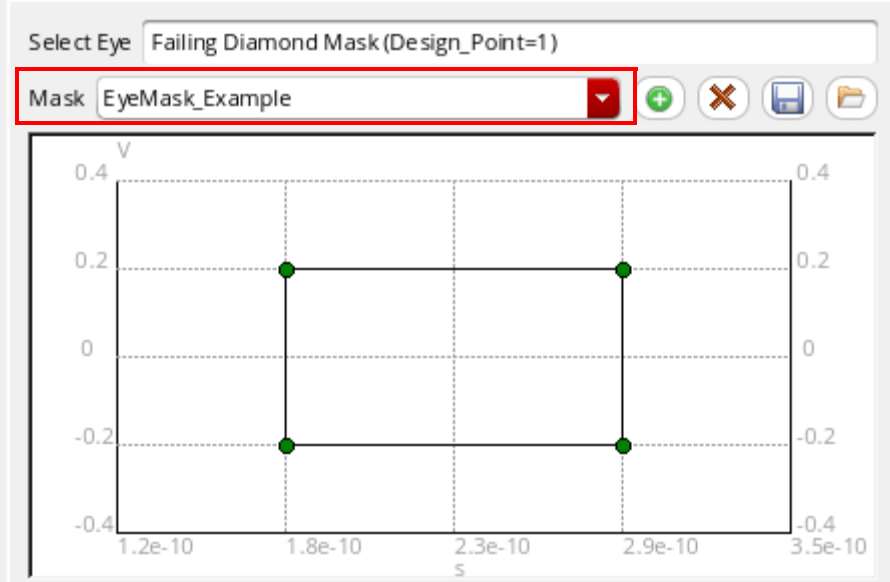
Perform the following steps to create an eye mask using an already existing eye mask:

1. Select an already existing eye mask from the *Mask* drop-down or import an eye mask from the VCSV file. For more information about how to import an eye mask from the

© 2023 Cadence Design Systems, Inc.

VCSV file, see Example of Saving and Importing an Eye Mask in VCSV Format. The selected or imported eye mask is displayed in the mask editing area.

2. Now, edit the mask shape, range, or coordinates. For information about editing the eye mask, see Editing an Eye Mask on page 300.
3. After the eye mask is created, you can click the *Test Mask* button to test the eye mask by plotting it on the specified eye diagram in the graph window.
4. Then, add this edited eye mask with a new name in the *Mask* drop-down. For example, consider the eye mask named `EyeMask_example` available in the *Mask* drop-down. It is of type square. You can select this eye mask and then edit the mask.



You can then rename this eye mask and add it to the *Mask* drop-down list. For information about how to add the eye mask to the *Mask* drop-down list, and how to save the eye mask in the VCSV file, see Performing Various Tasks on an Eye Mask on page 302.

5. After creating the eye mask, you can plot and evaluate the eye mask with the eye diagram and send the expression to ADE outputs. To know about how to plot the eye mask, see Plotting and Evaluating an Eye Mask on page 296.

Plotting and Evaluating an Eye Mask

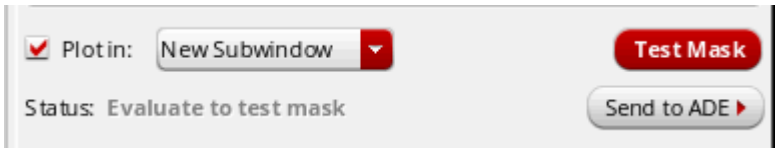
After creating or selecting an eye mask, you can plot and evaluate the eye mask.

To plot the selected eye mask,

© 2023 Cadence Design Systems, Inc.

➡ Click the *Plot in* check box. Then, in the drop-down, select the plot destination window. The available options are `Append`, `Replace`, `New Subwindow`, and `New Window`.

To test whether the eye mask passes or fails, click the *Test Mask* button to plot the eye mask on the specified eye diagram in the graph window.

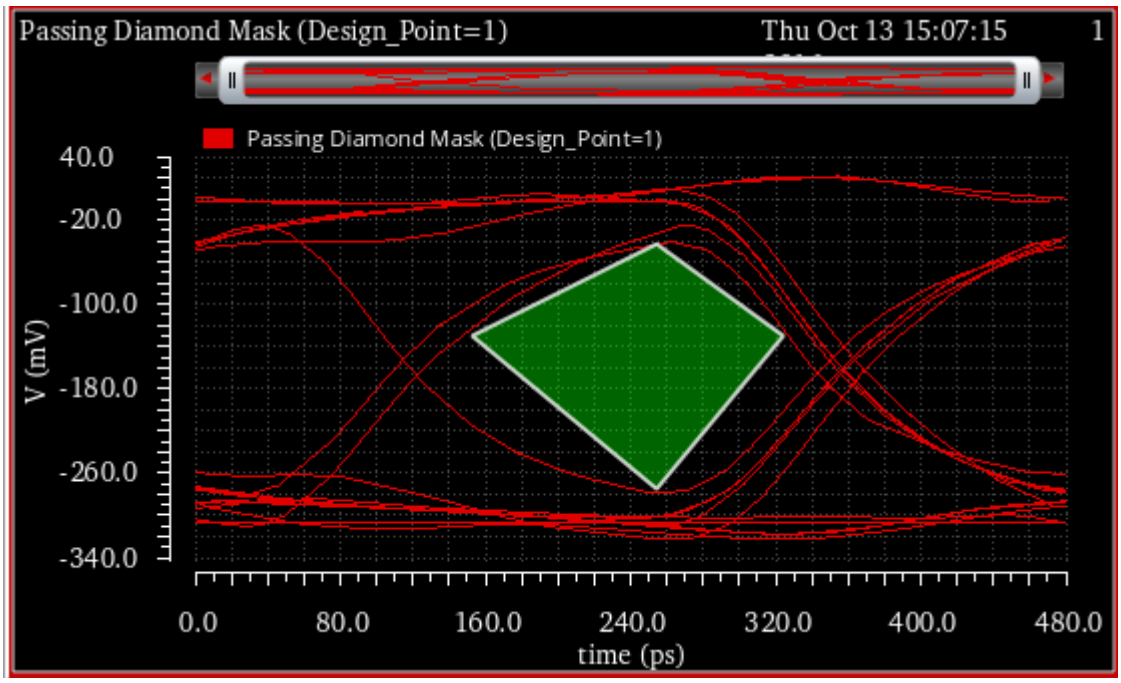


Note: The *Status* label at the bottom displays the current status of the eye mask and shows whether the eye mask has passed or failed. By default, the status is `Evaluate to test mask`.

To annotate and evaluate the eye mask on the selected eye diagram,

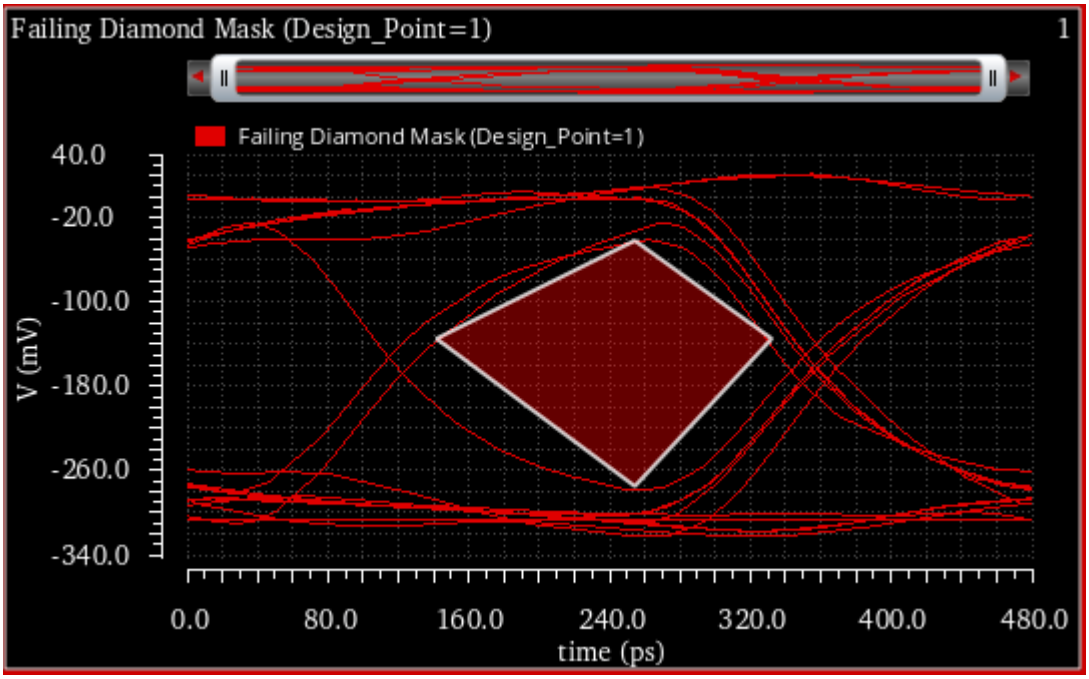
➡ Click the *Evaluate* button.

If the eye mask is displayed in green and none of the eye touches the mask, it indicates that the eye mask has passed the check. The status is also displayed as `PASS`.



© 2023 Cadence Design Systems, Inc.

If the eye mask is displayed in red and the eye touches the mask, it indicates that the mask has failed the check. The status is also displayed as `FAIL`.



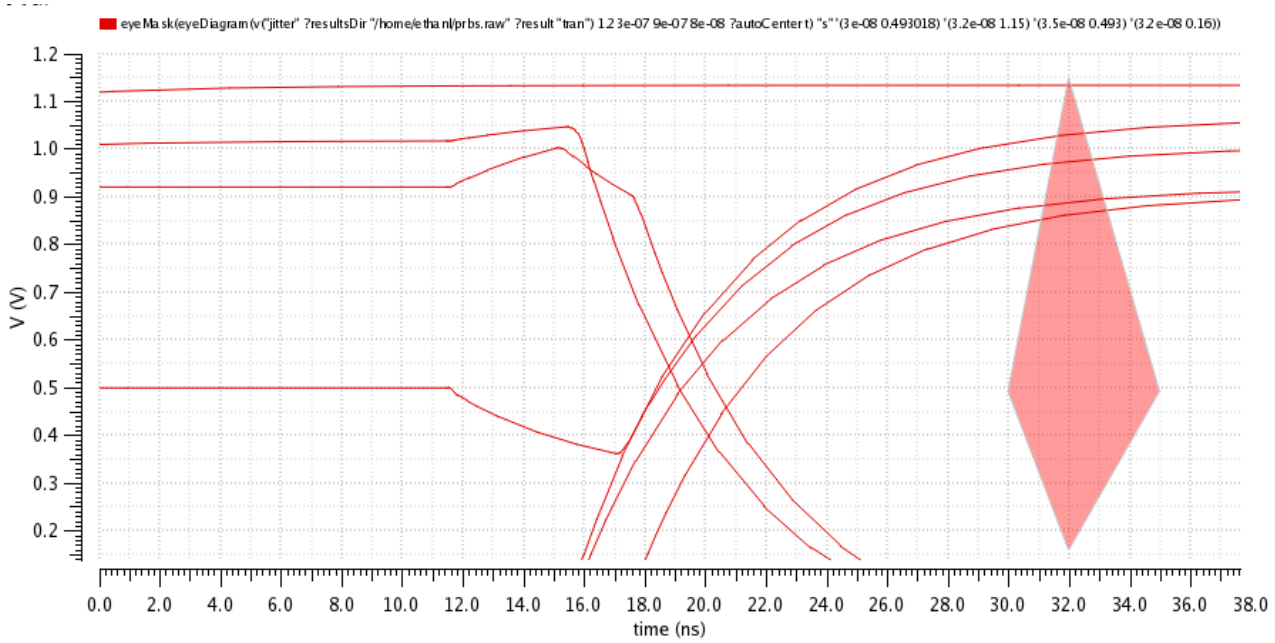
Sending an Eye Mask Expression to ADE

To send the eye mask to the ADE Outputs setup pane, choose one of the following options in the *Send to ADE* drop-down list:

- *Plot Expression*—Select this option to send the expression for the eye mask to the ADE Outputs Setup.
- *Test Expression*—Select this option to send an `eyeMaskViolationPeriodCount` expression to the ADE Outputs setup regardless of whether there is a violation. You can specify the specification conditions in this expression. The expression returns a numerical value. Each strand of the eye diagram obtained from slicing the input waveform by the eye period or triggering. The value returned by this expression is the

© 2023 Cadence Design Systems, Inc.

number of periods that contain a violation. For example, *eyeMaskViolationPeriodCount* returns `5` in the case below.



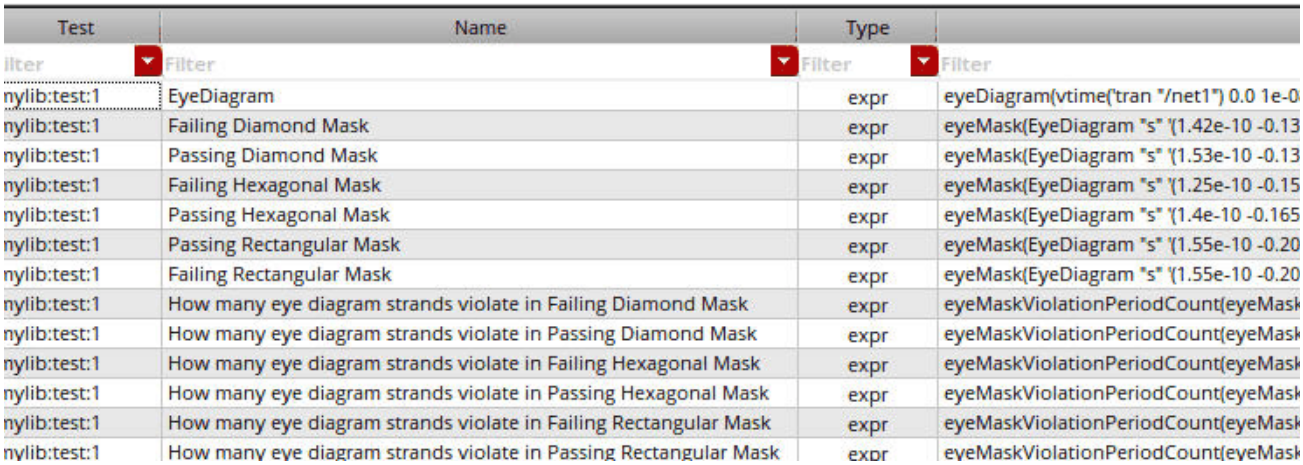
If you send a named mask, the name of the eye mask waveform expression is `eye_<signal name>_<mask name>`

The name of the named eye mask test expression is `eye_<signal name>_<mask name>_test`

The unnamed eye masks shows the full expression name.

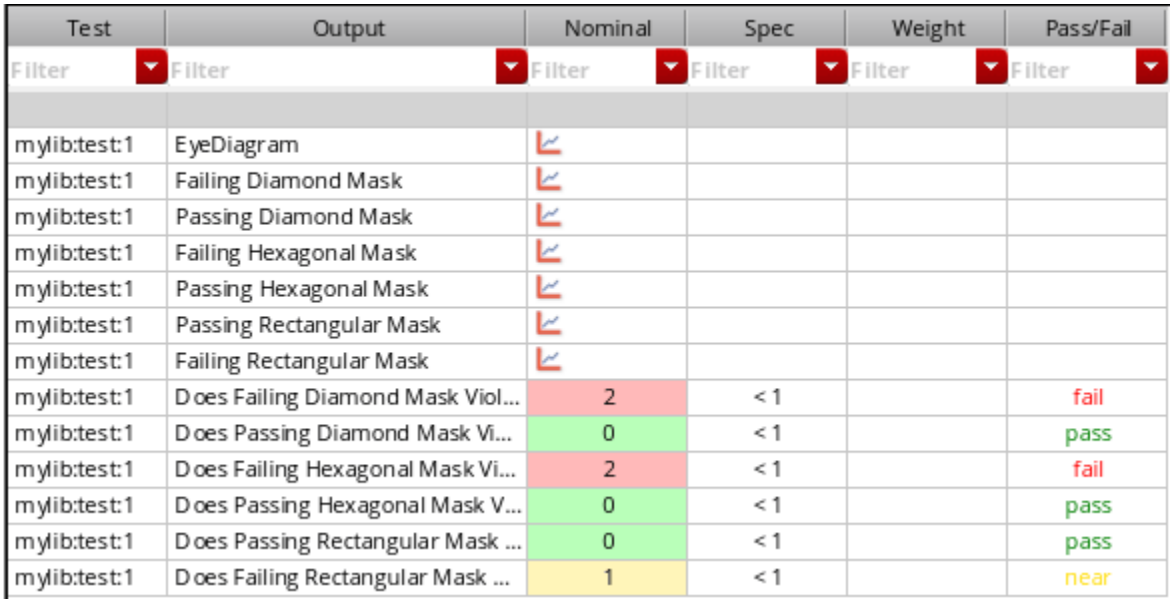
© 2023 Cadence Design Systems, Inc.

The figure below shows the eye mask expressions added to the ADE Assembler Outputs setup tab.



Test	Name	Type	
Filter	Filter	Filter	Filter
mylib:test:1	EyeDiagram	expr	eyeDiagram(vtime('tran "/net1") 0.0 1e-0
mylib:test:1	Failing Diamond Mask	expr	eyeMask(EyeDiagram "s" '(1.42e-10 -0.13
mylib:test:1	Passing Diamond Mask	expr	eyeMask(EyeDiagram "s" '(1.53e-10 -0.13
mylib:test:1	Failing Hexagonal Mask	expr	eyeMask(EyeDiagram "s" '(1.25e-10 -0.15
mylib:test:1	Passing Hexagonal Mask	expr	eyeMask(EyeDiagram "s" '(1.4e-10 -0.165
mylib:test:1	Passing Rectangular Mask	expr	eyeMask(EyeDiagram "s" '(1.55e-10 -0.20
mylib:test:1	Failing Rectangular Mask	expr	eyeMask(EyeDiagram "s" '(1.55e-10 -0.20
mylib:test:1	How many eye diagram strands violate in Failing Diamond Mask	expr	eyeMaskViolationPeriodCount(eyeMask
mylib:test:1	How many eye diagram strands violate in Passing Diamond Mask	expr	eyeMaskViolationPeriodCount(eyeMask
mylib:test:1	How many eye diagram strands violate in Failing Hexagonal Mask	expr	eyeMaskViolationPeriodCount(eyeMask
mylib:test:1	How many eye diagram strands violate in Passing Hexagonal Mask	expr	eyeMaskViolationPeriodCount(eyeMask
mylib:test:1	How many eye diagram strands violate in Failing Rectangular Mask	expr	eyeMaskViolationPeriodCount(eyeMask
mylib:test:1	How many eye diagram strands violate in Passing Rectangular Mask	expr	eyeMaskViolationPeriodCount(eyeMask

Now, when you run the simulation, these expressions are evaluated and the *Results* tab displays the pass and fail results. The Nominal value indicates the number of periods by which the eye mask is violated.



Test	Output	Nominal	Spec	Weight	Pass/Fail
Filter	Filter	Filter	Filter	Filter	Filter
mylib:test:1	EyeDiagram				
mylib:test:1	Failing Diamond Mask				
mylib:test:1	Passing Diamond Mask				
mylib:test:1	Failing Hexagonal Mask				
mylib:test:1	Passing Hexagonal Mask				
mylib:test:1	Passing Rectangular Mask				
mylib:test:1	Failing Rectangular Mask				
mylib:test:1	Does Failing Diamond Mask Viol...	2	< 1		fail
mylib:test:1	Does Passing Diamond Mask Vi...	0	< 1		pass
mylib:test:1	Does Failing Hexagonal Mask Vi...	2	< 1		fail
mylib:test:1	Does Passing Hexagonal Mask V...	0	< 1		pass
mylib:test:1	Does Passing Rectangular Mask ...	0	< 1		pass
mylib:test:1	Does Failing Rectangular Mask ...	1	< 1		near

Editing an Eye Mask

To edit an existing eye mask, select the eye mask from the Mask drop-down list and add it to to the mask list.

© 2023 Cadence Design Systems, Inc.

To edit the properties of the given mask, select the *Edit Mask* check box. The options in this group box become available for editing.

Note: When the *Edit Mask* check box is selected, you can also edit the mask coordinates directly in the mask editing area. Double-click the label and type the new coordinate values.

- In the *Mask Geometry* field, select the geometry you want to apply on the mask. The available geometries are `Hexagon`, `Diamond`, and `Rectangle`. By default, this field displays `Custom`, which is the default geometry value used by the selected template. The geometry of the displayed eye mask changes according to the type you select.
- Click the *Revert* button to set the fields back to their original values, clear the *Edit Mask* check box, and center the eye mask. The Revert button is disabled when you select the mask and becomes available after the mask is edited.
- In the *X Range* field, specify the minimum (xmin) and maximum (xmax) eye mask setup range for the X-axis. In the drop-down list next to this field, select the unit for the X-axis coordinates. The X range can be specified in unit intervals (`UI`) or seconds (`s`). The default unit is `UI`, where each UI=0.5 * eyePeriod. The X-axis coordinates are relative to 0.5*UI.
- In the *Y Range* field, specify the minimum (ymin) and maximum (ymax) range for the eye mask for the Y-axis.

  The range fields are used to control the limits of the mask editing area. If you change a range, it does not change the coordinates of the mask. However, when you change the mask shape, the mask is placed in the center according to the current axis limits.
- Double-click the vertices (corners) of the mask in the setup to directly change the X and Y coordinates.

After the eye mask is edited, you can add the eye mask to the *Mask* drop-down list. See the next section to know how to perform various operations on the eye mask, such as adding the eye mask to the list, removing an eye mask, saving the eye mask, and importing the eye mask.

© 2023 Cadence Design Systems, Inc.

Performing Various Tasks on an Eye Mask

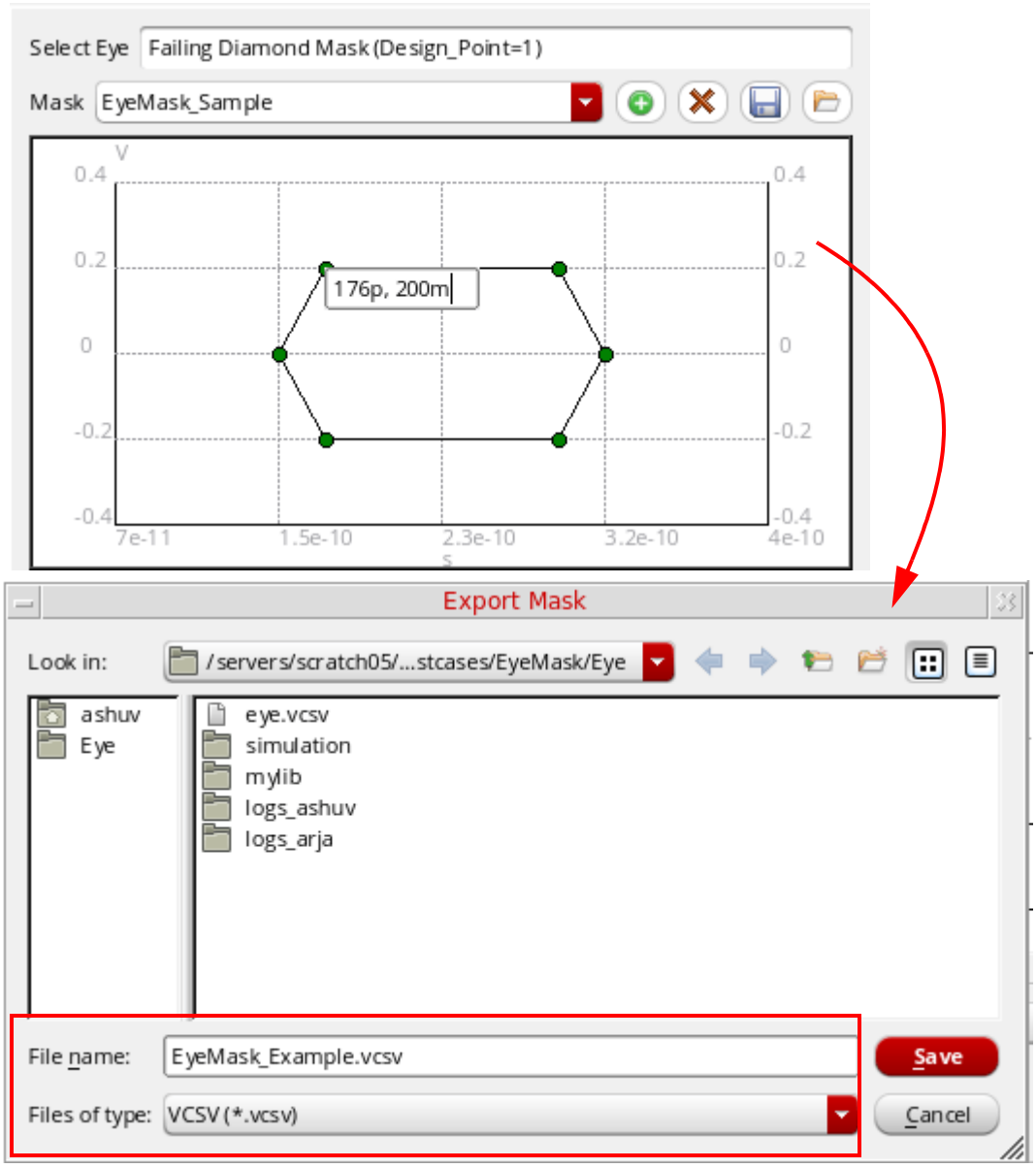
After selecting, creating, or editing the eye mask, you can click the following buttons to perform various tasks:

- —Adds the selected eye mask to the template list. After editing, if you want to add the updated eye mask to the template list, click the add button. The *Mask* field displays the `Untitled Mask` text. Then, specify a name for this eye mask. This mask is added at the top of the drop-down list of available templates.
- —Removes the selected eye mask from the list of templates.
- —Saves the selected eye mask to a file. When you click this button, the Export Mask form is displayed.
- —Imports an eye mask from a file and adds it to the list of available templates. When you click this button, the Import Mask form is displayed. In this form, specify the filename and location from where you want to import the eye mask. When you click *Open*, the selected eye mask is added to the *Mask* drop-down list and also displayed in the mask editing area. The file type that you select to save eye mask should be VCSV.

© 2023 Cadence Design Systems, Inc.

Example of Saving and Importing an Eye Mask in VCSV Format

Consider the example below in which the eye mask named `EyeMask_Example` is saved in the `EyeMask_Example.vcsv` file. To save this eye mask, click the *Save* button. Then, specify the mask filename in the VCSV format, as shown in the figure below.



Note: Ensure that you have the required write permissions on the VCSV file in which you save the eye mask, otherwise an error message is displayed.

After the mask is saved, you can import it later. The coordinates of the eye mask are saved in the VCSV file. For example, the `EyeMask_Example.vcsv` file has the following contents:

© 2023 Cadence Design Systems, Inc.

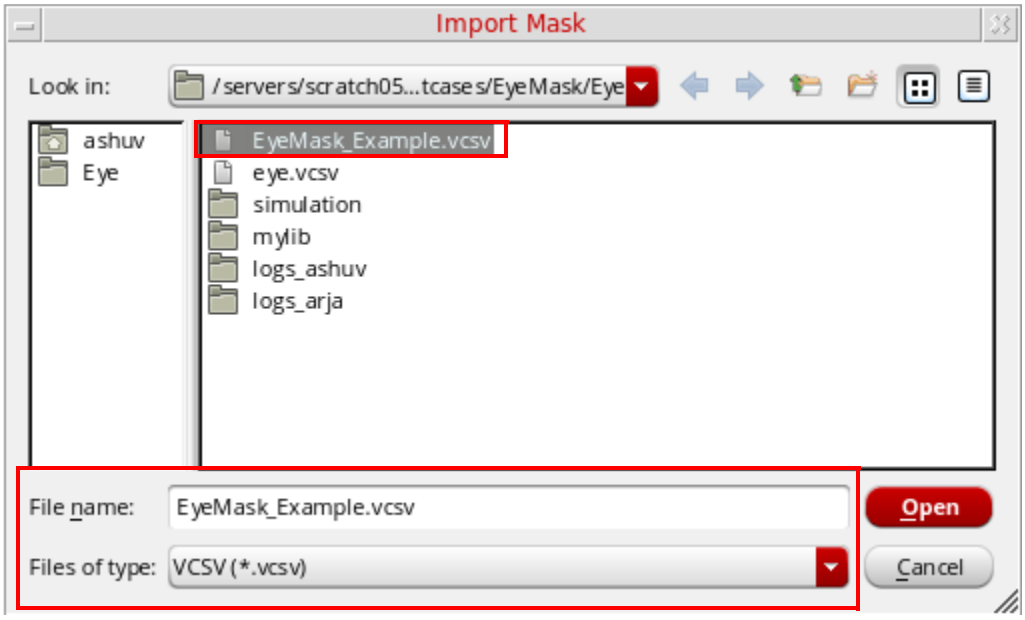
```
;EyeMask_Example
;X, Y
;Re, Re
;time, V
;s, V
1.758e-10, -0.2
1.5236e-10, 0
1.758e-10, 0.2
2.93e-10, 0.2
3.1644e-10, 0
2.93e-10, -0.2
```

You can directly change the coordinates in this file to support arbitrary mask shapes. Now, when you import this file, the changes are reflected in the mask shape. For example, change the file coordinates as shown below to specify a square mask shape:

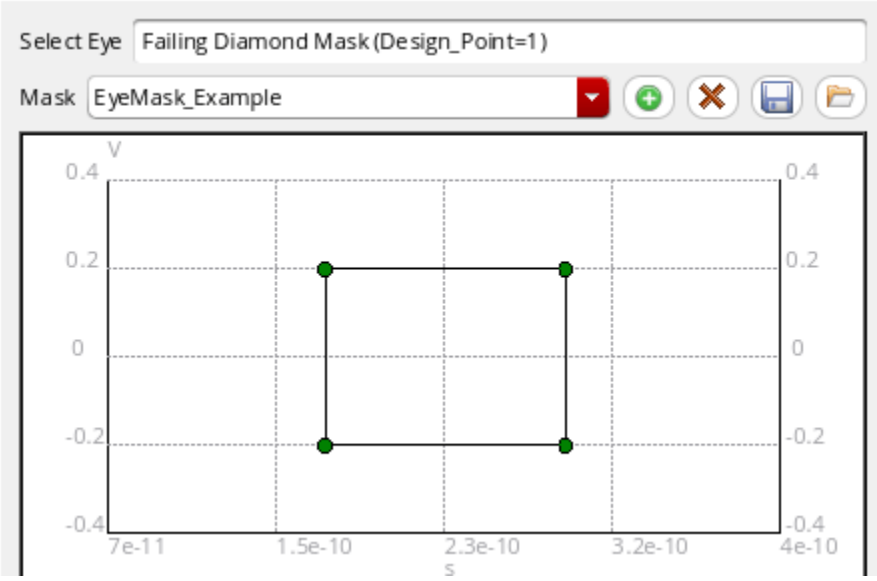
```
;EyeMask_Example
;X, Y
;Re, Re
;time, V
;s, V
1.758e-10, -0.0
1.758e-10, 0.2
2.93e-10, 0.2
2.93e-10, -0.2
```


© 2023 Cadence Design Systems, Inc.

Now, to import this updated file, click the Import button and select the saved mask filename, `EyeMask_Example.vcsv`, in the Import Mask form, as shown in the figure below.



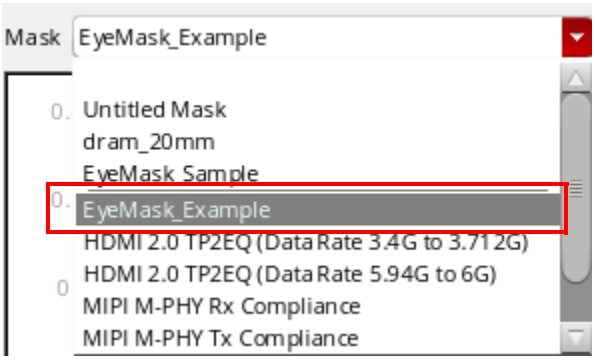
The selected eye mask is displayed in the mask editing area as shown in the figure below. Notice that the mask shape has changed now.



The eye mask name, `EyeMask_Example`, is also listed in the *Mask* drop-down.

© 2023 Cadence Design Systems, Inc.

Note: The added or imported eye mask is displayed in the *Mask* drop-down list only for the current session. You need to import it again for all future sessions.



You can then customize this eye mask by changing the mask shape and coordinates, and save it in a different VCSV file.

You can also use the `awvLoadEyeMask` SKILL function to load eye masks saved in a VCSV file and add them to the *Mask* drop-down list of the *Eye Mask* tab. Each mask is saved into a separate VCSV file.

Note: When you create a custom eye mask, it is available only for the current session. To load the custom eye mask automatically during startup and to display it in the *Mask* drop-down list, either export the custom mask into a VCSV file or import it from the VCSV file.

© 2023 Cadence Design Systems, Inc.

Horiz Marker Table Assistant

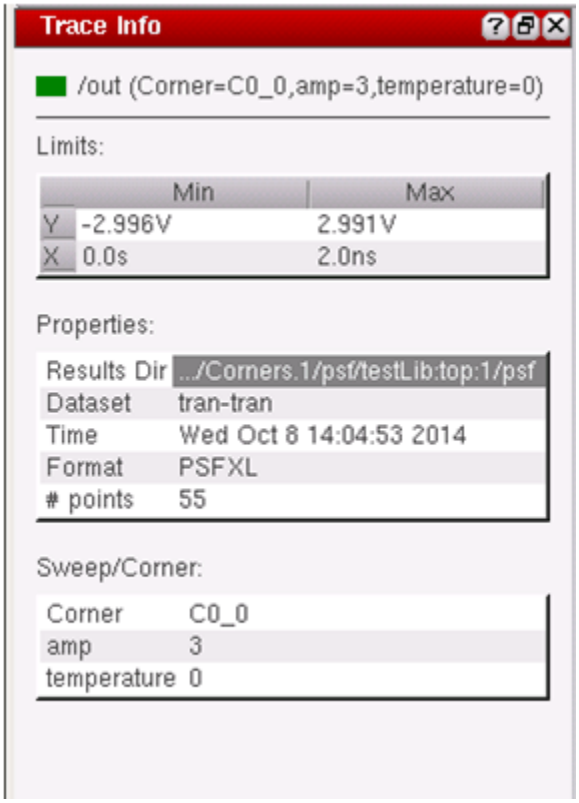
You use this assistant to view the interception data for horizontal markers in a table. For more information, see Displaying Intercept Data for Markers in Marker Tables on page 516.

Trace Info Assistant

You use this assistant to view information about the selected trace. To open the Trace Info, select a trace and choose *Window – Assistants – Trace Info*. The *Trace Info* assistant appears, displaying the information, such as *trace name and color, Y Min, Y Max, X Min, X Max, Time, Results dir, Dataset, Time, Data Format,* and *Number of Data points,* about the selected signal. It also displays information about the sweep and corner conditions for the selected trace.

To copy the trace properties, right-click the property and choose *Copy*.

For more information about the data displayed in the trace panel, see Working with Traces on page 368.



© 2023 Cadence Design Systems, Inc.

Vert Marker Table Assistant

You use this assistant to view the interception data for vertical markers in a table. For more information, see Displaying Intercept Data for Markers in Marker Tables on page 516.

Transient Measurement Assistant

The Transient Measurement assistant displays the calculated measurements for the transient markers on specific edges. The measurements can be falltime, risetime, overshoot, undershoot, and slewrate values. You can also generate the derived plots for rising and falling edges by using this assistant.

To hide the transient measurement, set the following environment variable in CIW or .cdsinit file:

```
envSetVal("viva.rectGraph" "enableEdgeMeasurement" 'string "false")
```

Note: This variable is by default set to `true`.

Before you start using the Transient Measurement assistant, refer to the following topics to understand some basic concepts:

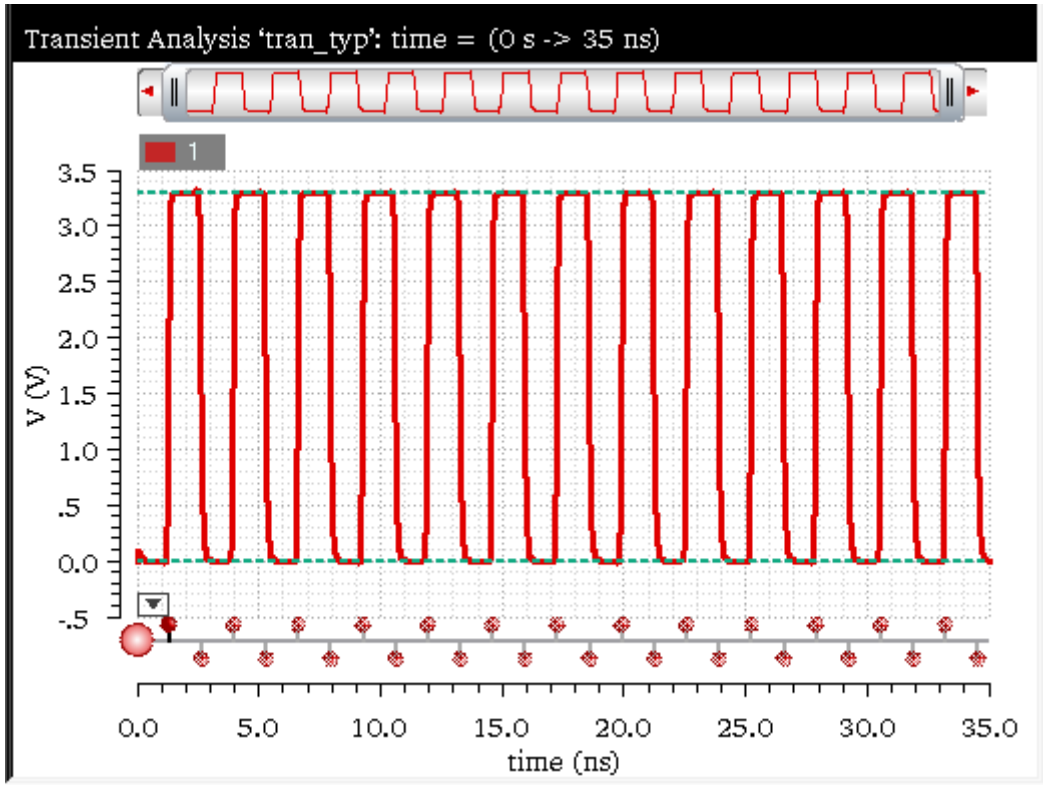
- Overview on page 308
- Specifying Edge Settings on page 309
- Using Edge Browser on page 312
- Context-Sensitive Menus on page 313
- Using the Transient Measurement Assistant on page 316
- Using Markers on page 319

Overview

When you plot a trace and load the waveform, Virtuoso Visualization and Analysis XL analyses the waveform and generates a set of edges, which are referred as threshold (crossing) points. An edge traverses from a low threshold point to a high threshold point

© 2023 Cadence Design Systems, Inc.

(rising edge) or from a high threshold point to a low threshold point (falling edge). Refer to the next section to know more about how to specify the properties of an edge.



Specifying Edge Settings

The *Edge Settings* form lets you change the edge properties and recalculate the edges.The measurements are displayed in the Transient Measurements assistant.

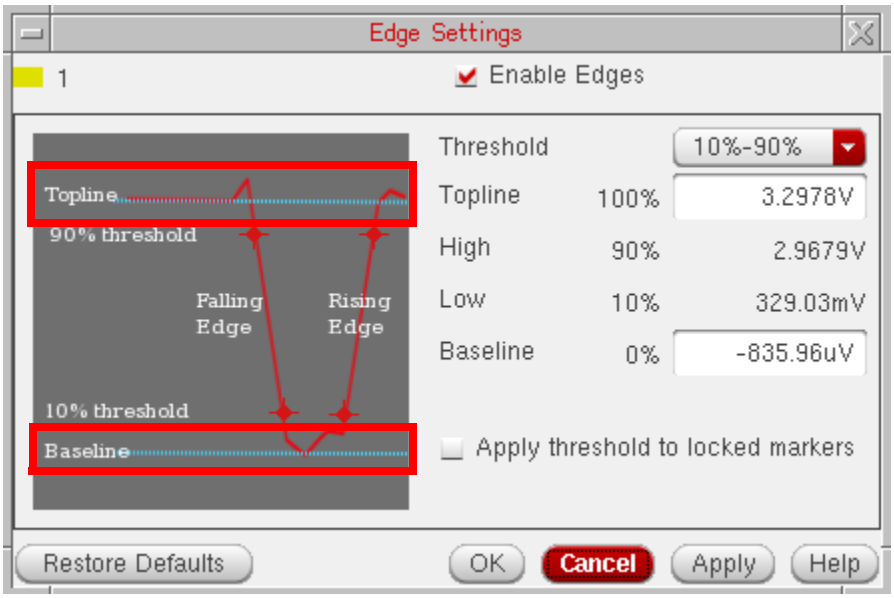
Do one of the following to open the Edge Settings form:

- Click the *Settings* button on the Transient Measurement assistant.
- Right-click the Edge Browser and choose *Edge Settings*.

The *Edge Settings* form displays a model depicting how the tool calculates edges of the selected trace. The blue reference lines in the model, as highlighted in the figure below,

© 2023 Cadence Design Systems, Inc.

indicate the topline and baseline reference values. You can drag these blue reference lines on the graph to set the topline and baseline values interactively.



This form includes the following fields that you can set to modify settings of the edges for a selected trace:

- *Enable Edges*—Disable this check box if you want to disable the edge calculation for the selected graph. By default, this check box is selected. When this check box is not selected, all fields in Transient Measurement assistant are disabled and the following message is displayed:

  ```
  Edge calculation for current graph disabled.
  To enable, check the Enable Edges checkbox in the Edge Settings dialog.
  ```

- *Threshold*—Select the threshold value from the drop-down. The default threshold value is `20%-80%`.
- *Topline*—Specify the topline (maximum) value. For more information about how to calculate the topline value, see Calculating Topline and Baseline Values on page 311.
- *High*—Displays the maximum threshold value.
- *Low*—Displays the minimum threshold value.
- *Baseline*—Specify the baseline (minimum) value. For more information about how to calculate the baseline value, see Calculating Topline and Baseline Values on page 311.
- *Restore Defaults*—Click this button to fill default values in the fields in the Edge Settings form.

© 2023 Cadence Design Systems, Inc.

- Click *OK*.

  The specified settings are applied to the selected trace and the trace edges are recalculated. Measurements and edge markers are updated.

Important

The edge settings that you define by using this form are applied only to the selected traces. Edge settings for other traces in the graph follow the settings defined under the following environment variables:

- `viva.trace threshold string "20_80"`
- `viva.trace baseAndToplineReferenceHint string "0.0,3.5"`
- `viva.trace autoReferenceLines string "true"`

Calculating Topline and Baseline Values

A threshold point is where the waveform intersects the Y threshold value. A threshold Y value is defined relative to the distance between the baseline and topline values. For example, for the threshold level of 20% and 80%, low and high threshold Y values are calculated as below:

```
Low threshold Y value == (0.20*(topline-baseline))+baseline

High threshold Y value == (0.80*(topline-baseline))+baseline
```

The topline (maximum) value is the most frequent value in the upper half of the entire range of trace Y values. Similarly, the baseline (minimum) value will be the most frequent value in the lower half of the entire range of Y values.

By default, the baseline and topline values are calculated automatically using a simple statistical analysis. These values are not original values and only indicate an estimate. Therefore, there might be cases where these values do not result in generating edges. In such cases, you need to specify these values manually. The following environment variables control the topline and baseline reference values:

```
viva.trace autoReferenceLines string "true"
```

By default this variable is set to `true`, which means Virtuoso Visualization and Analysis XL calculates the topline and baseline reference values automatically. If set to `false`, the values defined by the following environment variable is returned:

```
viva.trace baseAndToplineReferenceHint string "0.0,3.5"
```

Note: The edge analysis is performed on transient time waveforms only.

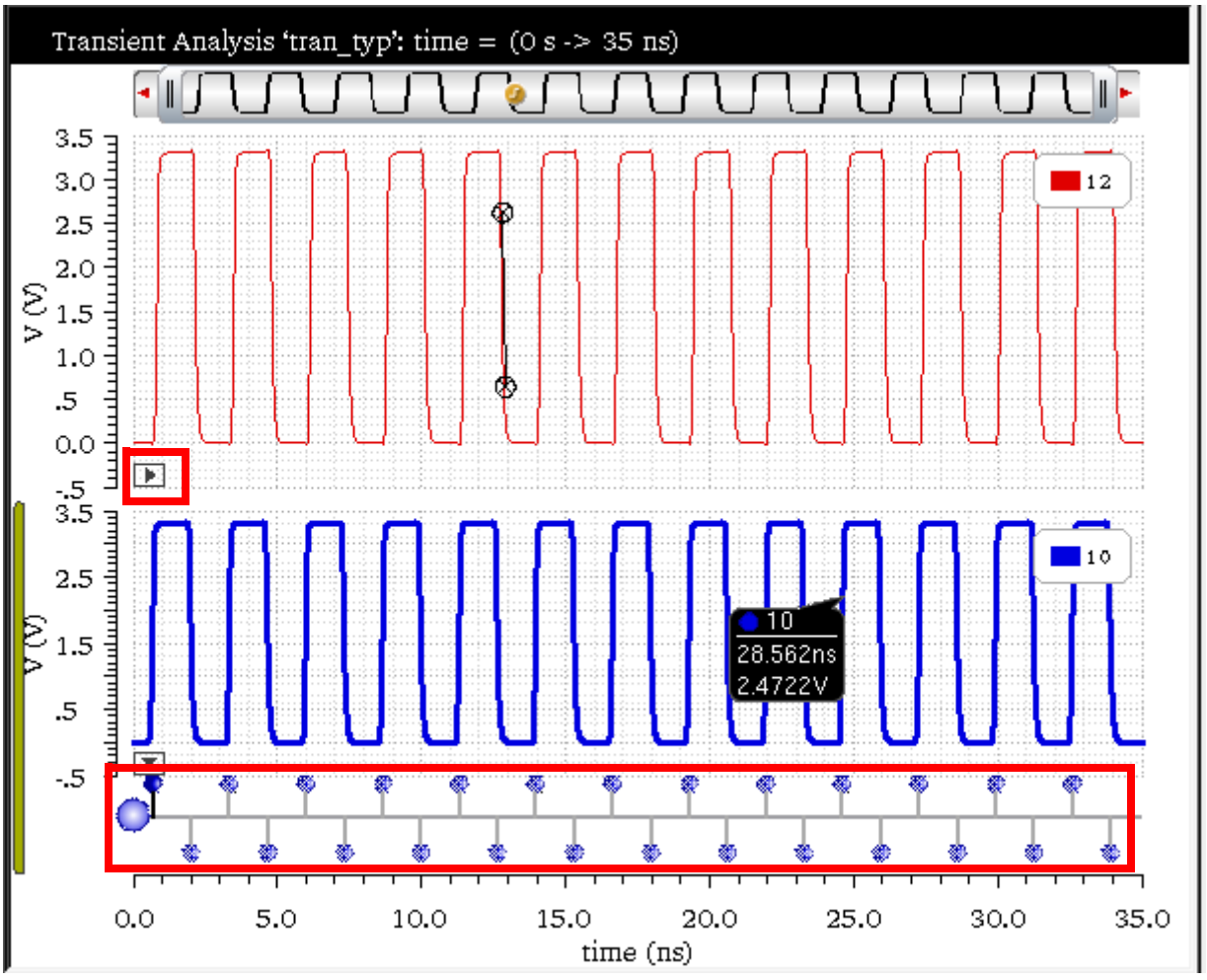
© 2023 Cadence Design Systems, Inc.

Using Edge Browser

You can use the edge browser to view and analyze the entire set of trace edges, zoom into a smaller range of edges, or zoom into a specific edge. You can use the edge context menu options to analyze a specific edge and annotate the edge with different marker types, such as edge, dx/dy, or period markers. Edge Browser also includes an individual context-menu options that you can use to generate risetime or falltime waveforms derived from the entire set of edges.

Each trace is associated with an individual edge browser, which can be displayed and hidden by clicking the Edge Browser button, as shown in the figure below. The edge browser is hidden in the first strip and displayed in the second strip.

Note: The edges in the graph are calculated only when the Edge Browser is open.



Do one of the following to display the edge browser on the graph:

© 2023 Cadence Design Systems, Inc.

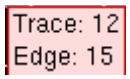
- Click the *Edge Browser* button as displayed in the above figure.
- Choose *Graph – Edge Browser*. This toggles the display of edge browsers for all strips in the graph.
- Right-click the *Edge Browser* button on the strip and choose *Edge Browser.*
- Set the following environment variable to `true`:

  ```
  envSetVal("viva.rectGraph showEdgeBrowser" 'string "true")
  ```

  If this variable is set to `true`, the edge browsers are displayed in all the strips. By default, this variable is set to `false`, which means all the edge browsers are hidden in all the strips.

Note the following:

- When you hover pointer on an edge in the Edge Browser, the tooltip displays the trace and edge number as shown in the figure below:

- When the mouse pointer focus is on edge browser, you cannot create horizontal and point markers because these markers require a Y value to be specified. As a result, the bindkeys `m` and `h` do not work when mouse pointer is on edge browser. However, you can create vertical markers by selecting an edge in the edge browser.
- All the edge controls are disabled and the measurement values are cleared in the Transient Measurement assistant to indicate that the edge calculation is in progress.

Context-Sensitive Menus

The two context-sensitive menus are available for edge and edge browser respectively.

Edge Context-Sensitive Menu

The following context-sensitive (shortcut) menu appears when you right-click the edge browser button:

© 2023 Cadence Design Systems, Inc.

This menu includes the following options:

- *Edge Browser*—Displays the Edge Browser for the selected trace.
- *Plot*—Plots the risetime or falltime waveforms.
- *Edge Settings*—Opens the Edge Settings form that you can use to modify the settings of the selected edge. For more information, see Specifying Edge Settings on page 309.

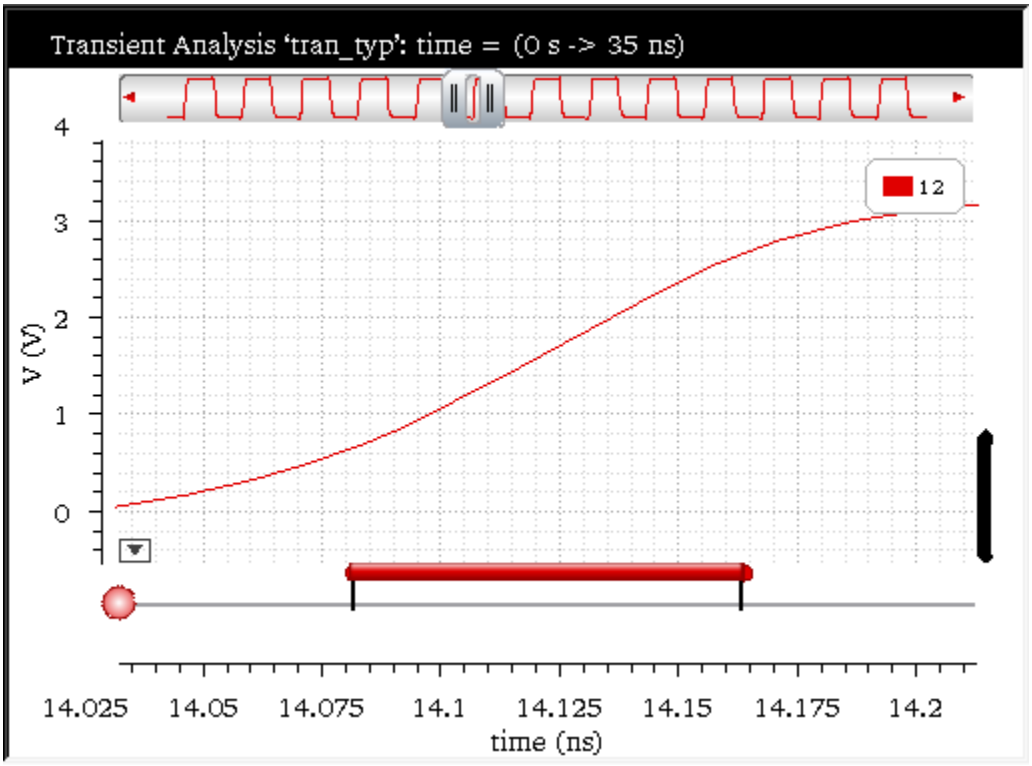
Edge Browser Context-Sensitive Menu

When you right-click an individual edge in the Edge Browser, the following context-sensitive (shortcut) menu appears:

- *Create Edge Marker*—Creates an edge marker on the selected edge. For more information about edge markers, see Working With Edge Markers on page 531
- *Label with Dx/Dy*—Creates a delta marker by adding two point markers at the lower and upper thresholds and a delta line between them.
- *Create Period Marker*—Creates a period marker on the selected edge.

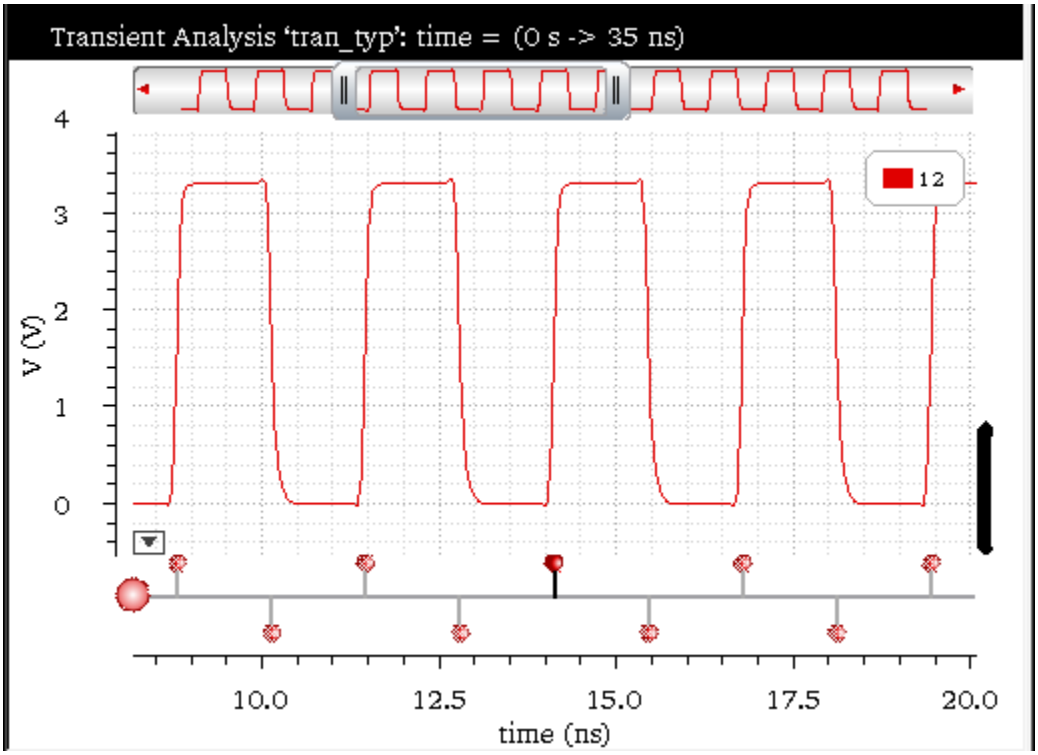
© 2023 Cadence Design Systems, Inc.

- *Zoom to Edge*—Zooms in the selected edge to view the details of the edge on which it is placed. The following figure illustrates the zoomed-in graph.



© 2023 Cadence Design Systems, Inc.

- *Zoom to Edge Context*—Zooms in the selected edge to the context of the edge on which it is placed. The following figure illustrates the zoomed-in graph.



Using the Transient Measurement Assistant

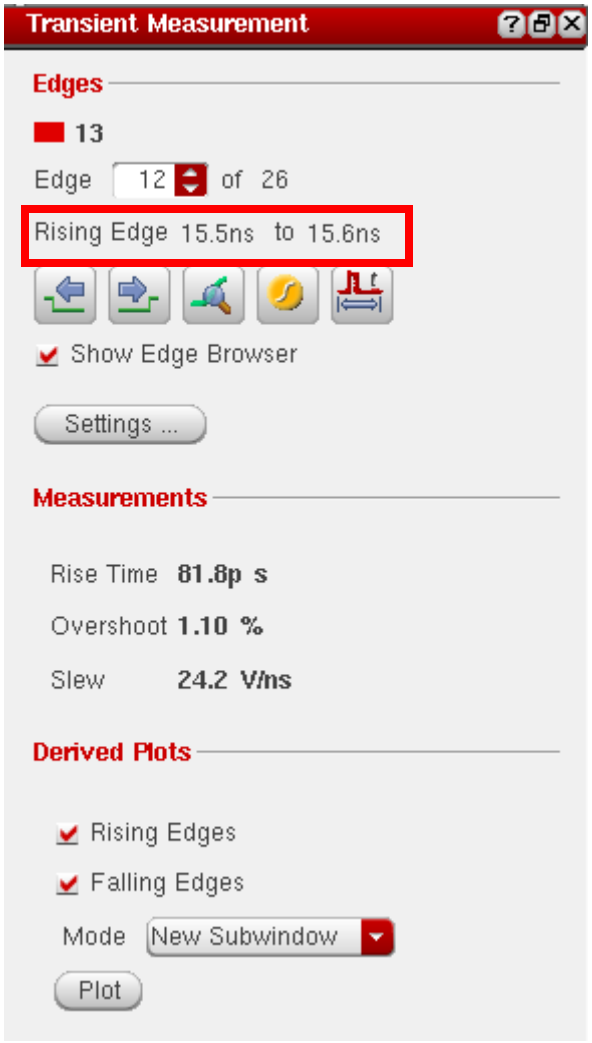
Perform one of the following steps to open the Transient Measurement assistant:

- Choose *Measurements — Transient Measurement*.
- Choose *Window – Assistants – Transient Measurement*.

  Alternatively, you can open the tool in the `TM` workspace to display the Transient Measurement assistant. The workspace can be set using the Workspace toolbar.

© 2023 Cadence Design Systems, Inc.

The *Transient Measurement* assistant is shown in the figure below:



This assistant includes the following fields:

- *Edges*—This panel lets you navigate between edges and displays the information for the current edge. It also helps you to perform certain operations on the selected edge, such as placing an edge or a period marker, zoom into the edge and so on. This includes the following fields:
  - *Edge*—Select the edge from the combo box for which you want to view the measurements. This also displays the number of total edges in the selected trace. The edge calculation is displayed below the *Edge* combo box, as highlighted in the figure above.

© 2023 Cadence Design Systems, Inc.

- The table below lists the icons that you can click to perform certain tasks on the graph:

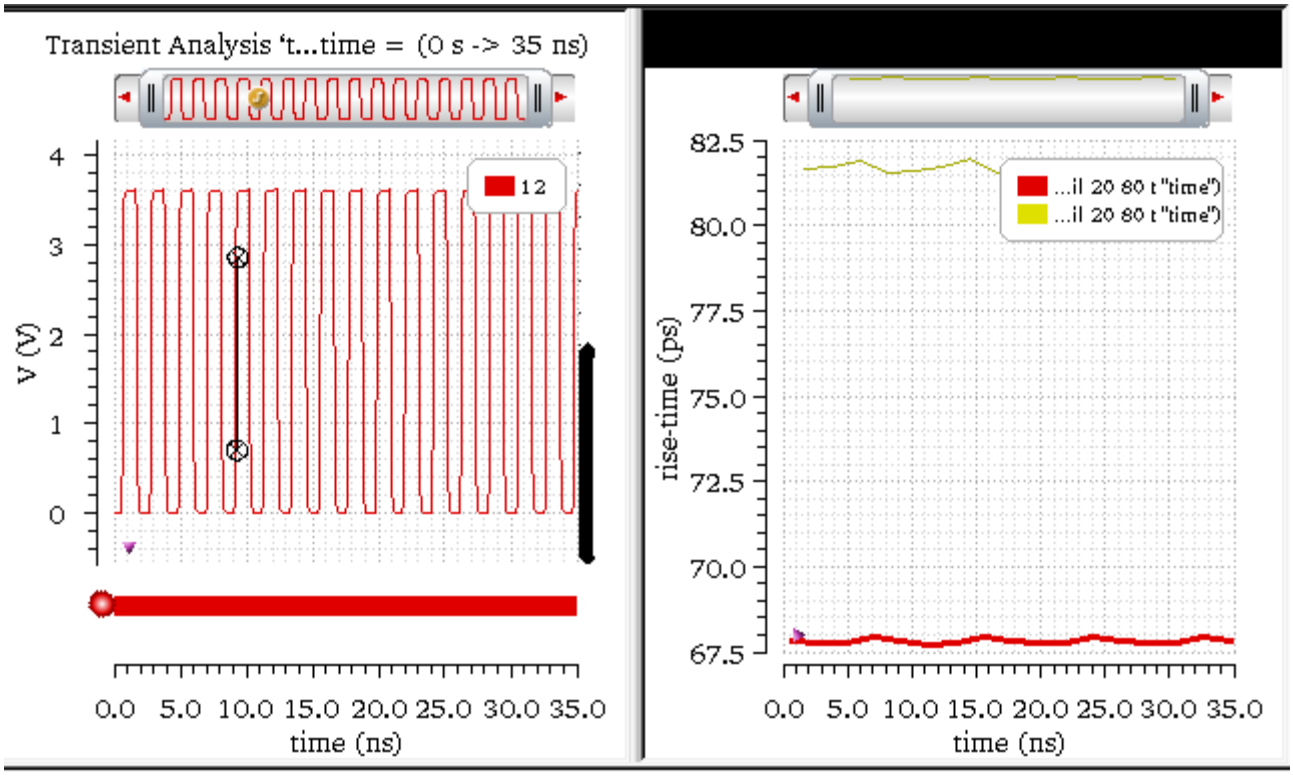
Button	Description
	Click this icon to move to the pervious edge on the selected trace.
	Click this icon to move to the next edge on the selected trace.
	Click this icon to zoom the graph to view the area of the edge.
	Click this icon to place an edge marker on the selected edge in the graph
	Click this icon to place a period marker on the current edge to the next edge in the same direction. For example, rising-to rising and falling-to-falling.

  - *Show Edge Browser*—Select this check box to show or hide the edge browser for the selected trace, which is displayed at the bottom of the graph or strip. For more information about edge browser, see Using Edge Browser on page 312.
  - *Settings*—Click this button to view and modify the edge settings of the selected trace. For more information, see Specifying Edge Settings on page 309.

- *Measurements*—Calculates and displays the following measurement values for the selected edge:
  - *Rise Time*
  - *Fall Time*
  - *Overshoot*
  - *Undershoot*
  - *Slew*
- *Derived Plots*—Generates the derived plots for the selected edges in the graph. Derived plots are the risetime or falltime waveforms derived from the entire set of edges and plotted against time.

© 2023 Cadence Design Systems, Inc.

- *Rising Edges*—Click this check box if you want to generate derived plots for all the rising edges in the graph.
- *Falling Edges*—Click this check box if you want to generate derived plots for all the falling edges in the graph.
- *Mode*—Specify the plotting mode that you want to use for the derived plots. The available plotting modes are—*Append*, *Replace*, *New Window*, and *New SubWindow.*
- *Plot*—Click this button to plot the specified derived plots. The figure below displays the derived plots for falling as well as rising edges in a new subwindow:



Using Markers

You can use the following markers to view and examine a specific edge:

Edge Marker

An Edge Marker annotates an edge graphically by using the current edge information. If you update the edges by using the *Edge Settings* form, the edge markers are also updated to

© 2023 Cadence Design Systems, Inc.

reflect the new edge data. In general, an edge marker shows the top and bottom threshold points of the edge and the risetime or falltime values of the selected edge. When you zoom into the marker, the additional information is displayed to show the baseline and topline values and other details. You can change the threshold value that is displayed by an edge marker. This does not change the edge and only changes the annotation points of the edge marker. If you lock the marker, the intercepts points are not updated when the edges are changed.

For more information about edge markers, see Working With Edge Markers on page 531.

Video

The Working with Transient Measurement Assistant video shows how you can use Transient Measurement assistant, create edge markers and change their properties.

© 2023 Cadence Design Systems, Inc.

Direct Measurements Assistant

You can use the *Direct Measurements* assistant to quickly perform amplitude, time-domain, and frequency-domain measurements without creating expressions. Using the assistant, you can also annotate these measurements.

This section includes the following topics:

- Opening the Direct Measurements Assistant
- Overview of the Direct Measurements Assistant
- The Amplitude Tab
- The Time Tab
- The Freq Tab
- The Compare Tab

Video

You can watch a video demonstration on how to work with the Direct Measurements assistant at Direct Measurements Assistant. You can also read the related blog at Virtuoso Video Diary: Direct Measurements Assistant in Virtuoso Visualization and Analysis XL.

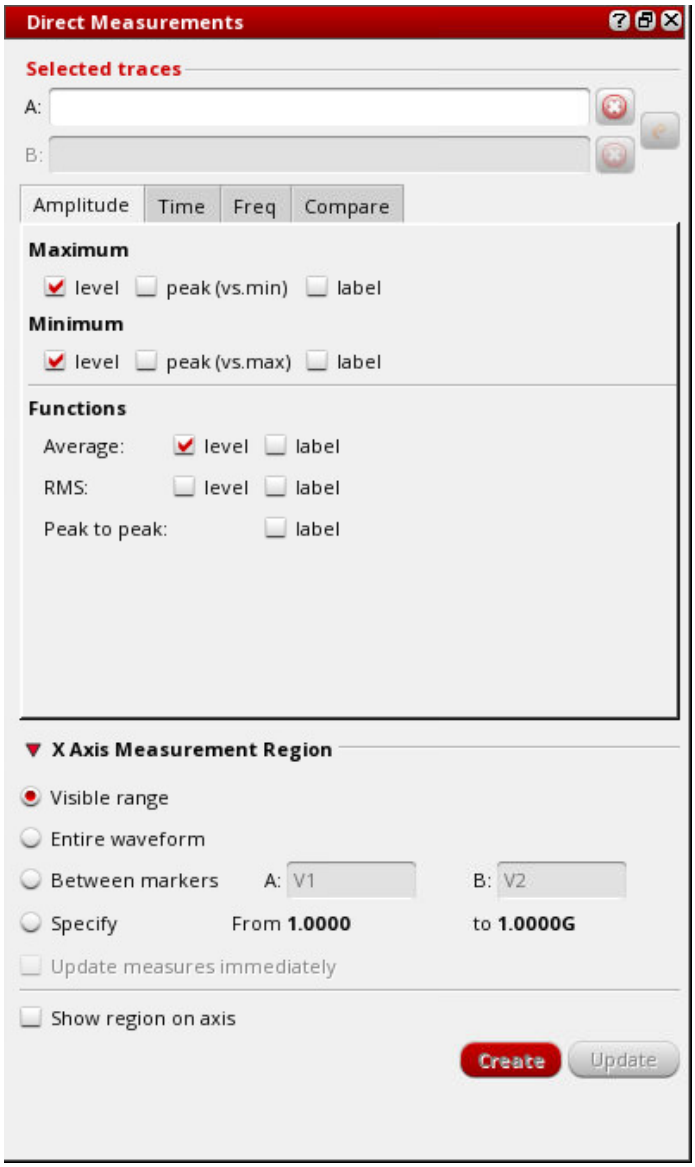
Opening the Direct Measurements Assistant

To open the *Direct Measurements* assistant:

➡ From the menu bar of Virtuoso Visualization and Analysis XL, choose *Window – Assistants – Direct Measurements*.

© 2023 Cadence Design Systems, Inc.

The *Direct Measurements* assistant opens.



Overview of the Direct Measurements Assistant

The *Direct Measurements* assistant contains the following tabs:

- *Amplitude*

  This tab is available for both ac and transient signals. You can use this tab to perform the following amplitude measurements:

© 2023 Cadence Design Systems, Inc.

- *Maximum value*
- *Minimum value*
- *Average value*
- *RMS value*
- *Peak-to-Peak value*

- *Time*

  This tab is available only for transient signals. You can use this tab to perform the following time-domain measurements:

  - *Duty cycle*
  - *Frequency*
  - *Pulse width*
  - *Period*
  - *Rise time*
  - *Fall time*
  - *Delay*

- *Freq*

  This tab is available only for ac signals. You can use this tab to perform the following frequency-domain measurements:

  - *Phase margin*
  - *Gain margin*
  - *Bandwidth*

- Compare

  This tab is available for both ac and transient signals. You can use this tab to compare two waveforms based on absolute and relative tolerances, and offset values.

The Amplitude Tab

This section describes how to measure the amplitude of a signal by using the *Amplitude* tab of the *Direct Measurements* assistant:

© 2023 Cadence Design Systems, Inc.

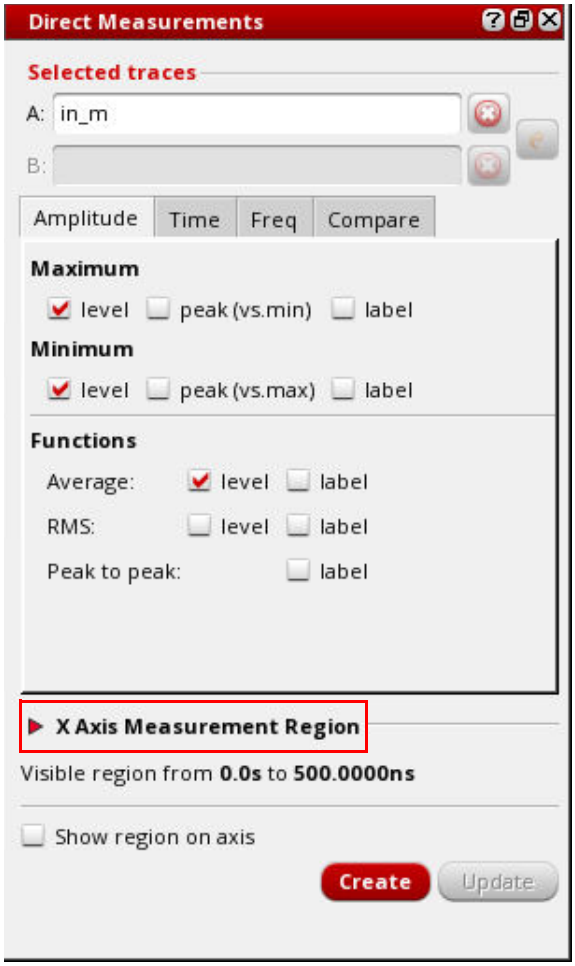
- Specifying the Measurement Region
- Measuring the Amplitude of a Signal

Specifying the Measurement Region

You can specify the measurement region on the X-axis that you want to consider when performing amplitude measurements of a signal.

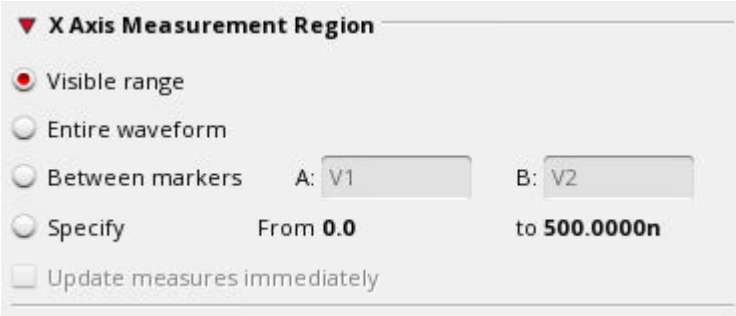
To specify the measurement region:

1. Click the arrow next to *X Axis Measurement Region* to expand the available options.



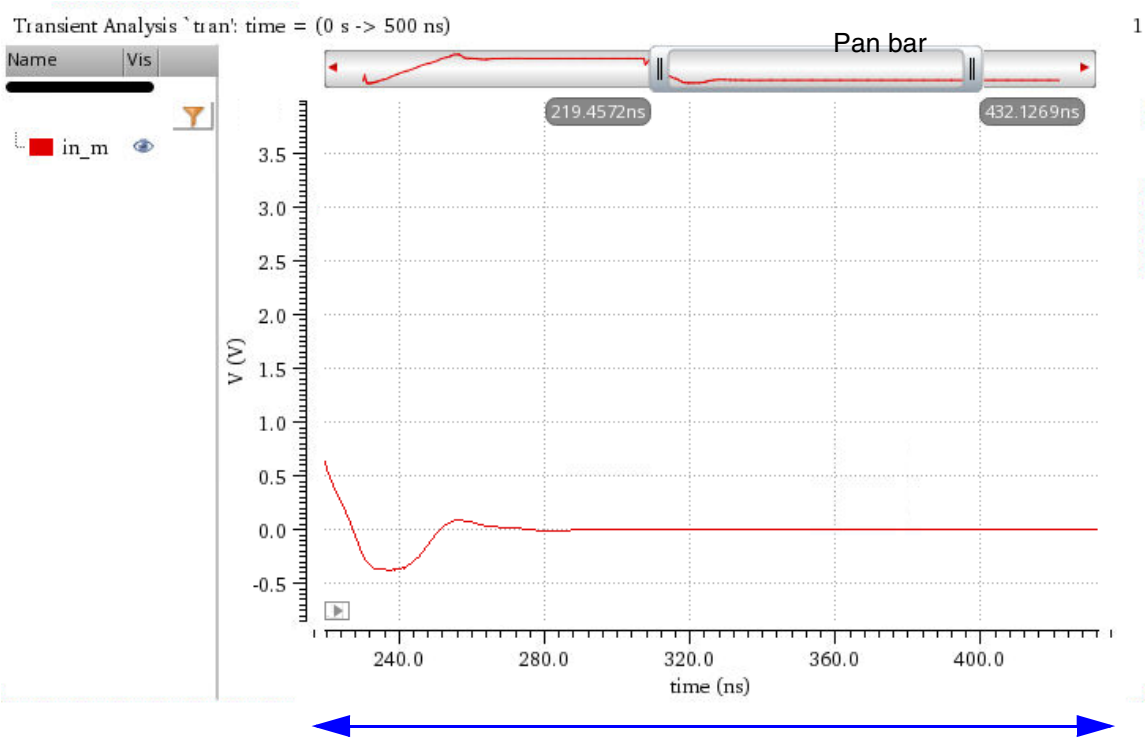
© 2023 Cadence Design Systems, Inc.

The options are displayed, as follows:



2. Select one of the following options:

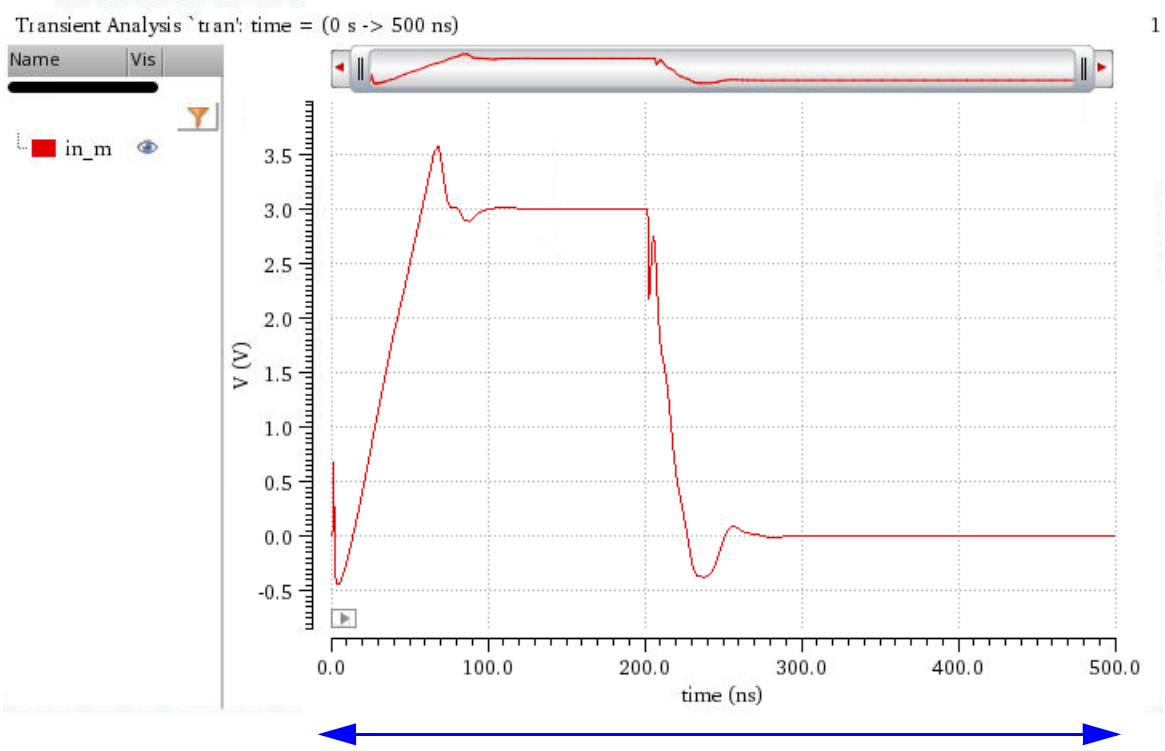
   - *Visible range*: Measurements are performed only on the part of the waveform that is currently visible in the graph window. You can change the visible range by adjusting the pan bar.



Visible range: `219.4572ns-432.1269ns`

© 2023 Cadence Design Systems, Inc.

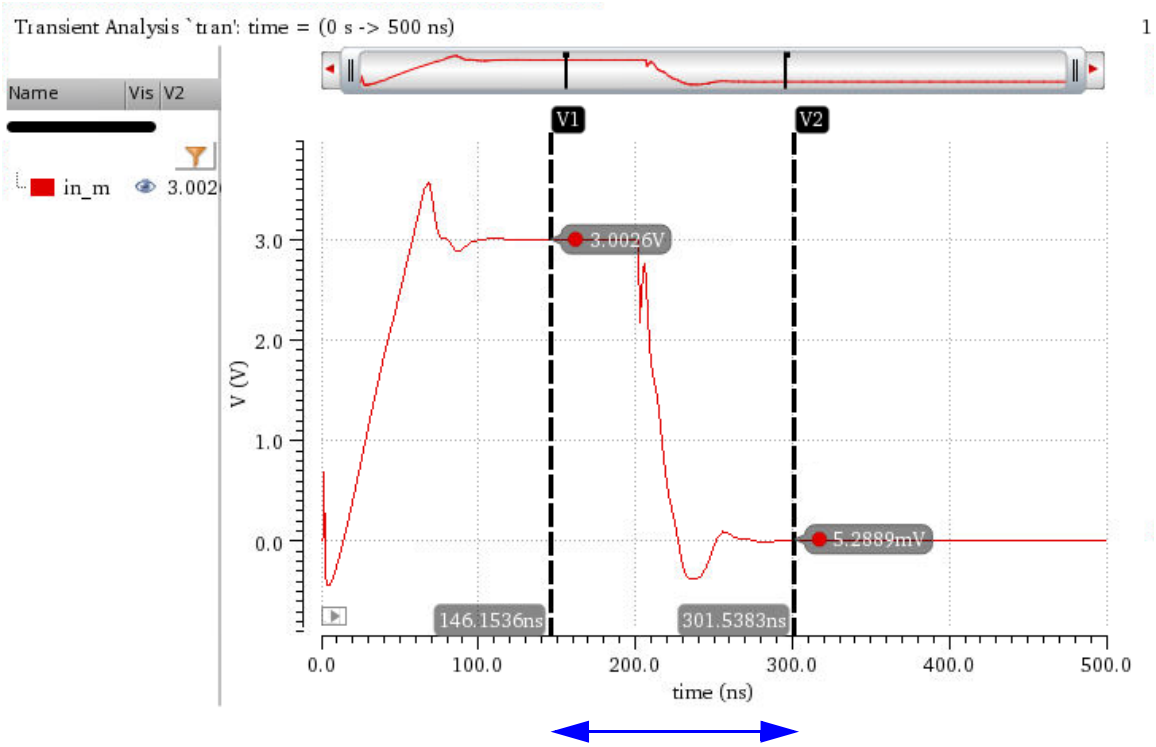
- *Entire waveform*: Measurements are performed on the entire waveform of the signal.



Entire waveform: `0.0ns-500.0ns`

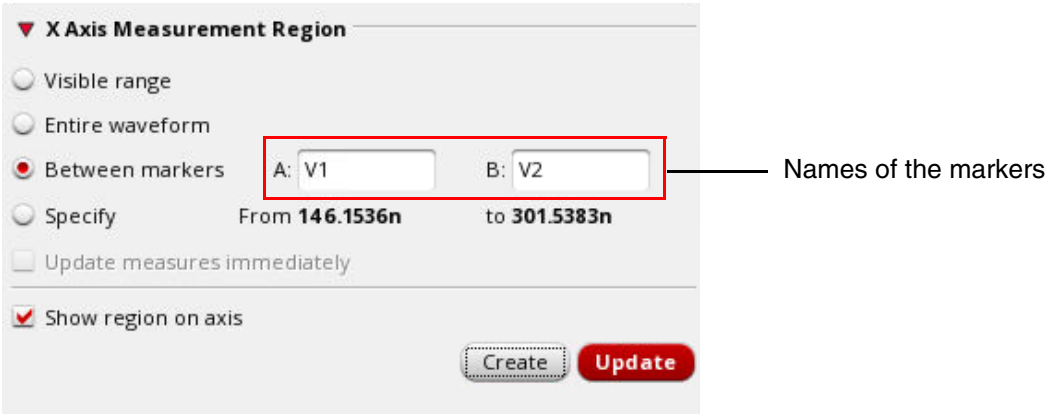
© 2023 Cadence Design Systems, Inc.

- *Between markers*: Measurements are performed on the part of the signal that is between the vertical markers or point markers.



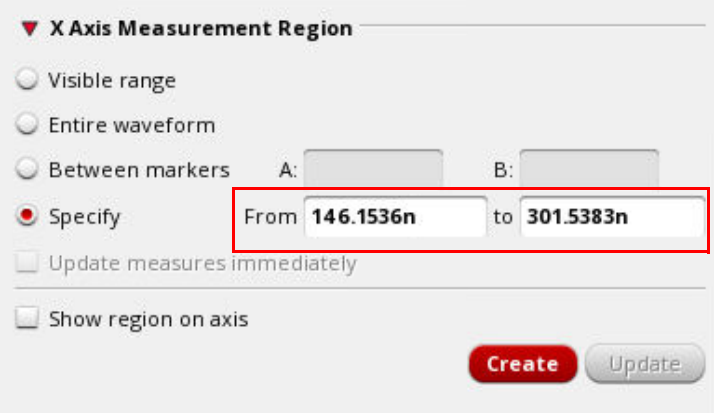
Between markers: `V1-V2`

In this case, also specify the names of the vertical or point markers in the *A* and *B* fields, as follows:



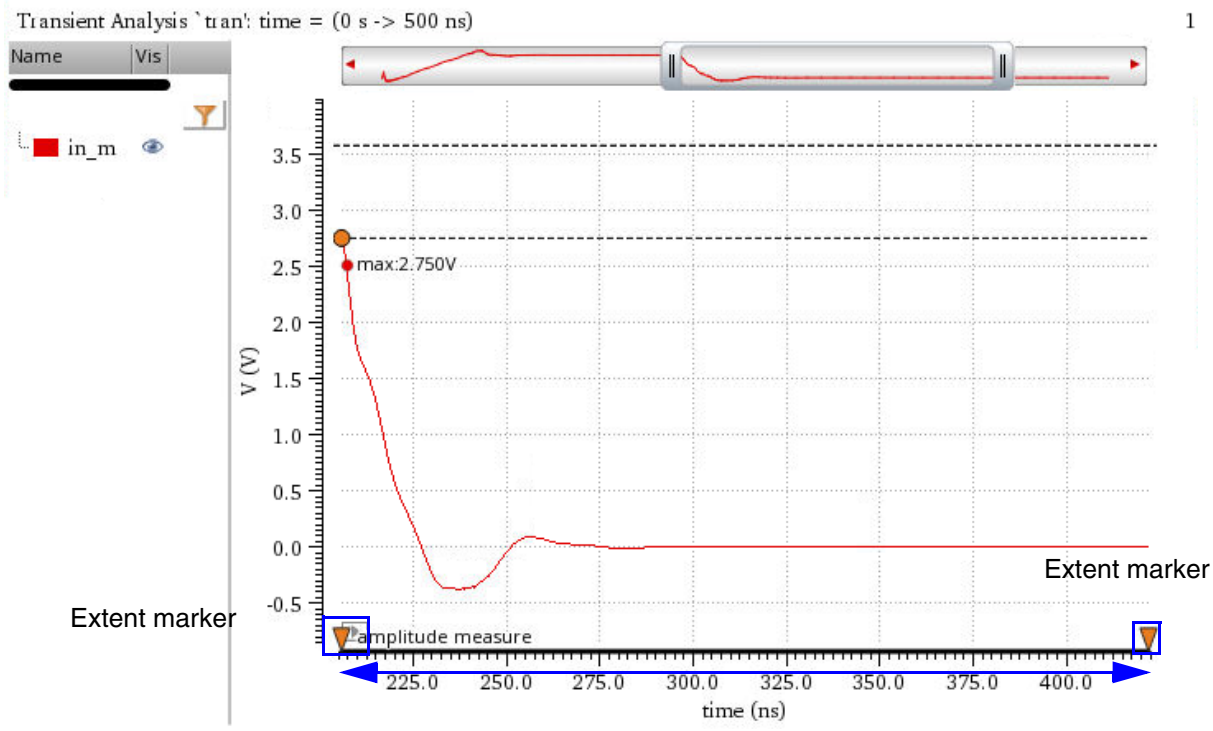
© 2023 Cadence Design Systems, Inc.

- *Specify*: Measurements are performed on the part of the signal that falls between the specified range on the X axis. Specify the range manually in the *From* and *to* fields.



3. Select *Show region on axis* if you want to display an extent marker over the X axis, highlighting the region of measurement.

   The measurement region is highlighted, as shown in the following figure:



Extent markers over X-axis highlighting the region of measurement

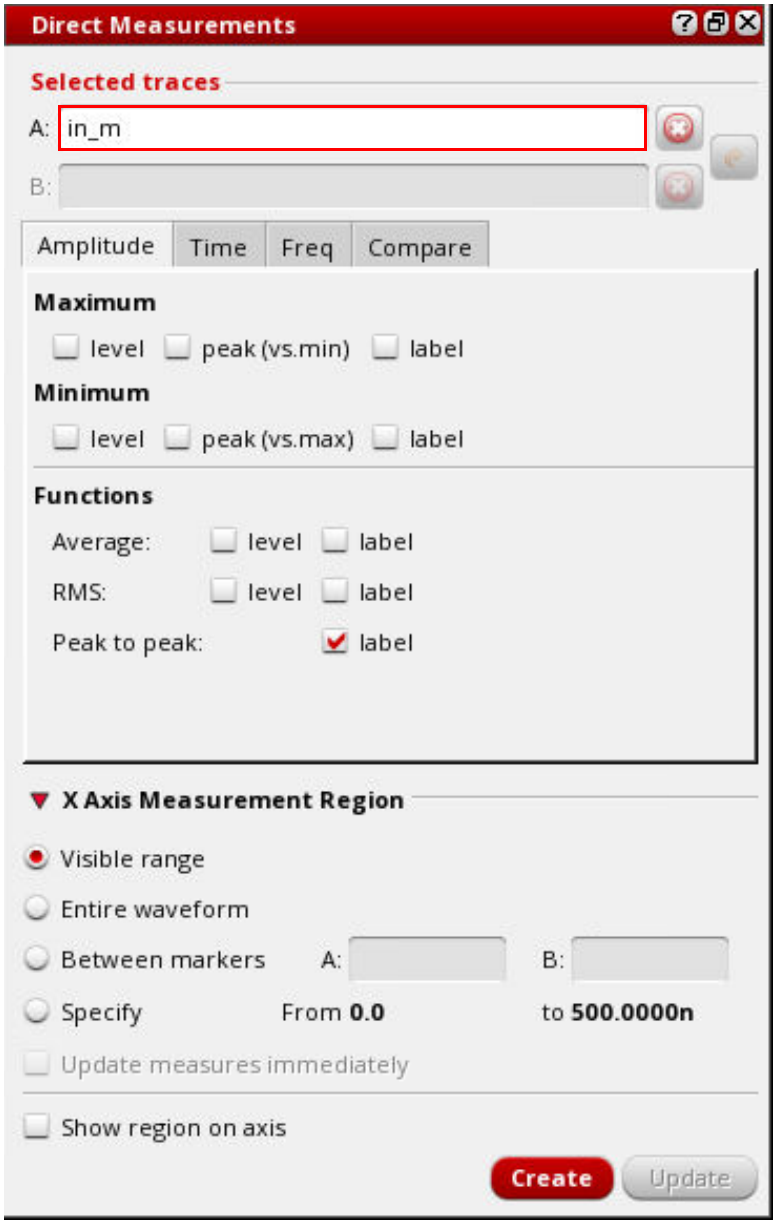
© 2023 Cadence Design Systems, Inc.

Measuring the Amplitude of a Signal

To measure the amplitude of a signal:

1. In a graph window, select the signal whose amplitude you want to measure.
2. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Window – Assistants – Direct Measurements*.

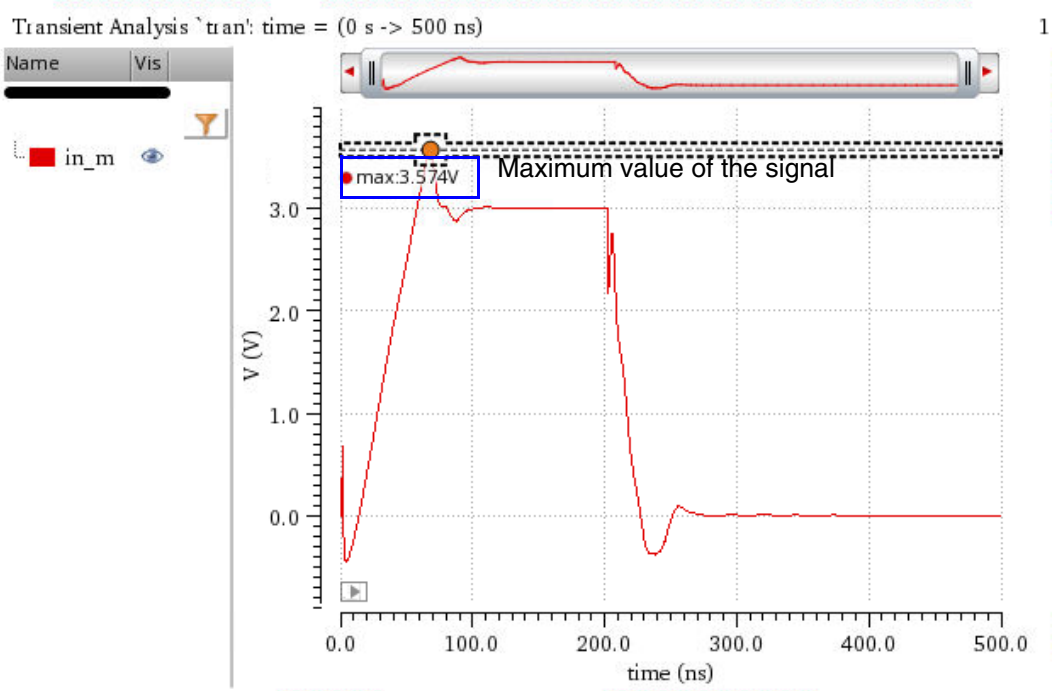
   The *Direct Measurements* assistant opens, displaying the selected signal.



© 2023 Cadence Design Systems, Inc.

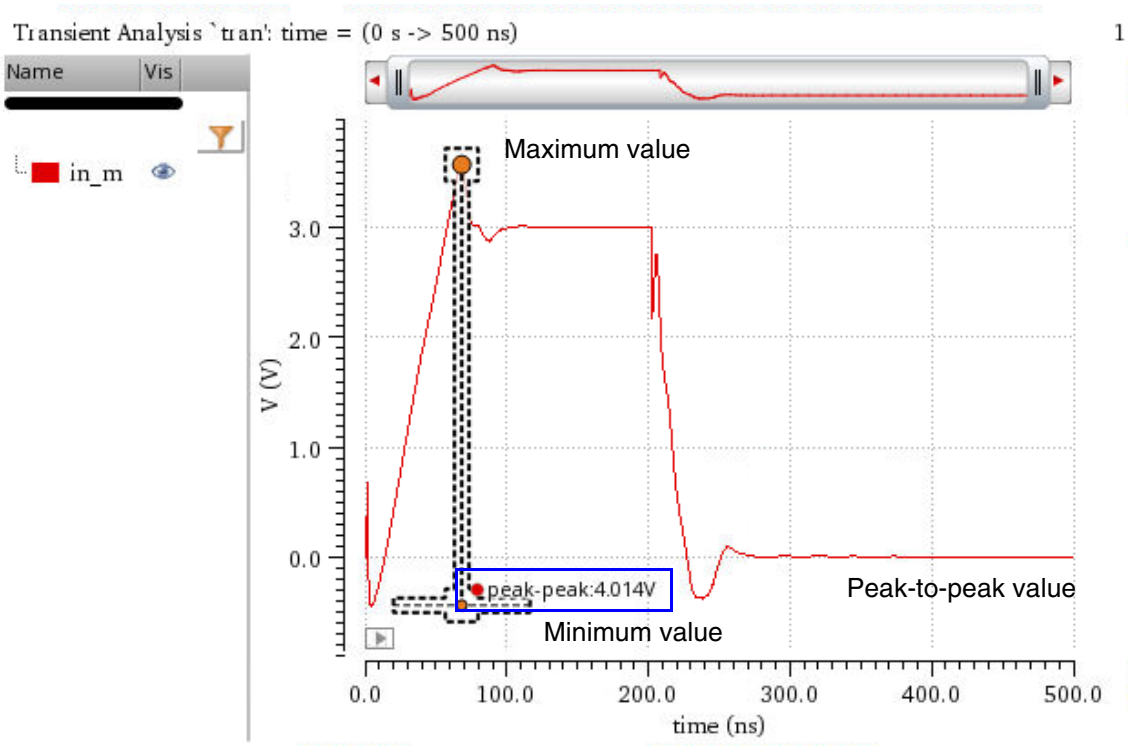
3. In the *Maximum* section on the *Amplitude* tab, select one or more of these options:

- *level*: To display the location of the maximum value and to highlight the maximum level with dotted lines.



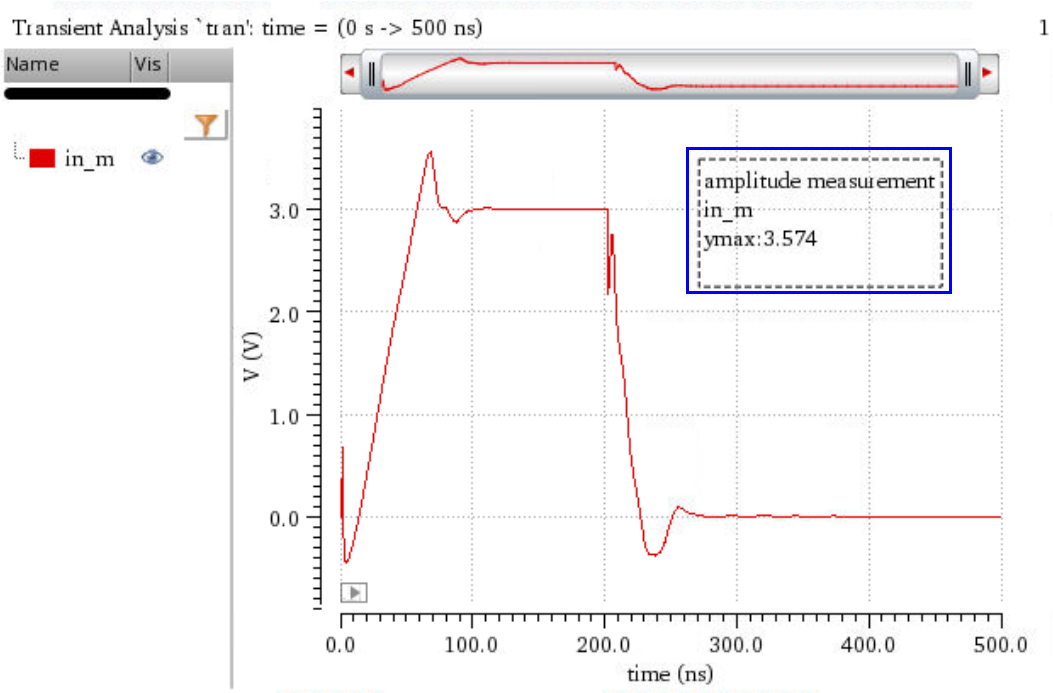
© 2023 Cadence Design Systems, Inc.

- *peak (vs. min)*: To display the location of the maximum value and to highlight the peak-to-peak value of the signal.



© 2023 Cadence Design Systems, Inc.

- *label*: To display the maximum value of the signal, name of the signal, and type of the measurement in a label.



4. In the *Minimum* section, select one or more of the following fields:
   - *level*: To display the location of the minimum value and to highlight the minimum level with dotted lines.
   - *peak (vs. min)*: To display the location of the minimum value and to highlight the peak-to-peak value of the signal.
   - *label*: To display the minimum value and name of the signal, and type of the measurement in a label.
5. In the *Functions* section, select one or more of the following fields:
   - *Average*
     - *level*: To display the location of the average value and to highlight the average level with dotted lines.
     - *label*: To display the average value and name of the signal, and the type of the measurement in a label.
   - *RMS*

© 2023 Cadence Design Systems, Inc.

- *level*: To display the location of the RMS value and to highlight the RMS level with dotted lines.
- *label*: To display the RMS value and name of the signal, and type of the measurement in a label.

- For *Peak to peak* function, select *label* to display the peak-to-peak value and name of the signal, and type of the measurement in a label.

6. In the *X Axis Measurement Region* section, specify the measurement region on the X axis.

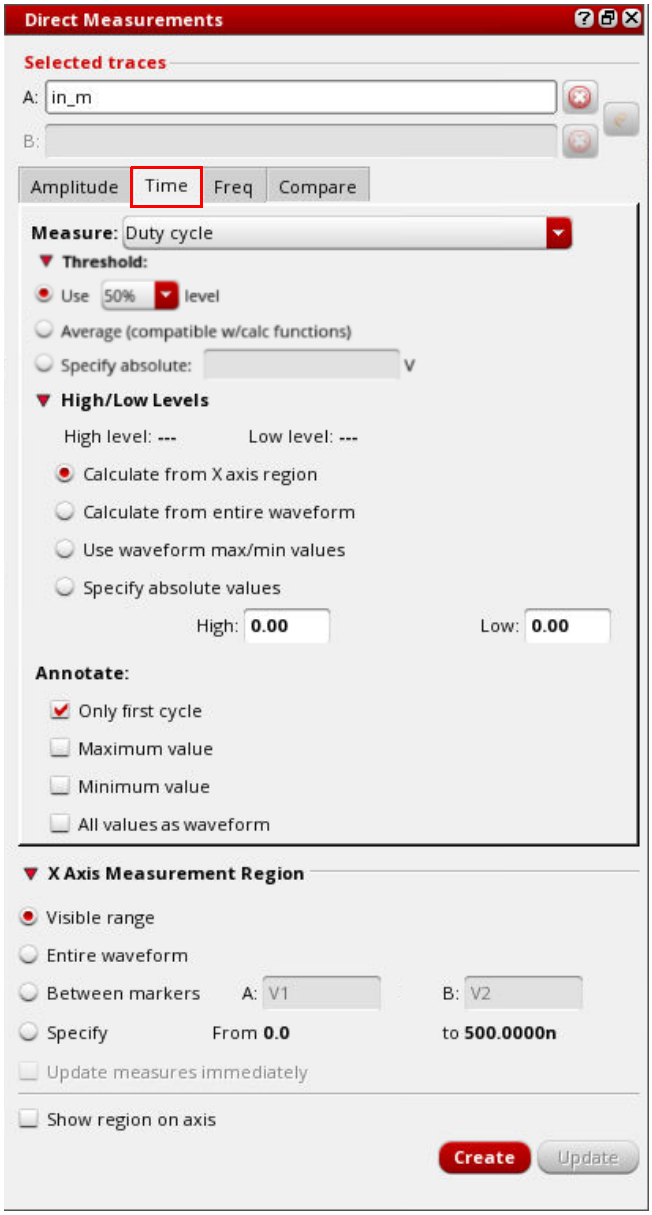
   See the Specifying the Measurement Region section for more information.

7. Click *Create* to perform the selected amplitude measurements.

© 2023 Cadence Design Systems, Inc.

The Time Tab

The following figure shows the various sections of the *Time* tab.



The following topics describe how to specify various options to perform time-domain measurements using the *Time* tab:

- Specifying Time-Domain Measurements
- Specifying Threshold Values

© 2023 Cadence Design Systems, Inc.

- Specifying Edge Threshold Levels
- Specifying the Method to Calculate High and Low Levels
- Specifying Annotate Options
- Measuring Delay between Signals

Specifying Time-Domain Measurements

To specify the time-domain parameter that you want to measure:

1. From the *Measure* drop-down list, select one of the following time-domain parameters:
   - *Duty cycle*
   - *Frequency*
   - *Pulse width*
   - *Period*
   - *Rise time*
   - *Fall time*
   - *Delay*
2. For *Frequency*, *Pulse width*, and *Period* measurements, select either *Positive pulse* or *Negative pulse*.

Specifying Threshold Values

➡ In the *Threshold* section, select one of the following options to specify threshold values of the signal to be used in time-domain measurements:

- *Use*: To specify the threshold level in percentage, select a percentage value from the drop-down list.
- *Average*: To specify the average value of the signal as the threshold value, which is the integral of the waveform divided by the x range.
- *Specify absolute*: To specify an absolute threshold value, enter the absolute threshold value in the text field.

© 2023 Cadence Design Systems, Inc.

Specifying Edge Threshold Levels

Note: You need to specify the edge threshold levels to measure rise time and fall time.

To specify the edge threshold levels:

1. From the *Measure* drop-down list on the *Time* tab, select either *Rise time* or *Fall time*.
2. In the *Edge Threshold Levels* section, select one of the following options:
   - *90/10%*: The rise time or fall time is calculated, as follows:
     - Rise time: Time taken to reach from A (`10% of the peak-to-peak value + initial value`) to B (`90% of the peak-to-peak value + initial value`).
     - Fall time: Time taken to reach from A (`90% of the peak-to-peak value + initial value`) to B (`10% of the peak-to-peak value + initial value`).
   - *80/20%*: The rise time or fall time is calculated, as follows:
     - Rise time: Time taken to reach from A (`20% of the peak-to-peak value + initial value`) to B (`80% of the peak-to-peak value + initial value`).
     - Fall time: Time taken to reach from A (`80% of the peak-to-peak value + initial value`) to B (`20% of the peak-to-peak value + initial value`).
   - *Specify absolute*: Specify the absolute threshold values in the *High* and *Low* fields.

     The rise time is calculated as the time taken by the signal to reach from the specified low value to the specified high value.

     The fall time is calculated as the time taken to reach from the specified high value to the specified low value.

Specifying the Method to Calculate High and Low Levels

In the *High/Low Levels* section of the `Time` tab, specify how you want to calculate the high and low levels of the signal:

- *Calculate from X axis region*: To automatically calculate high and low levels for the part of the waveform that is currently visible in the graph window. You can change the visible range by adjusting the pan bar.

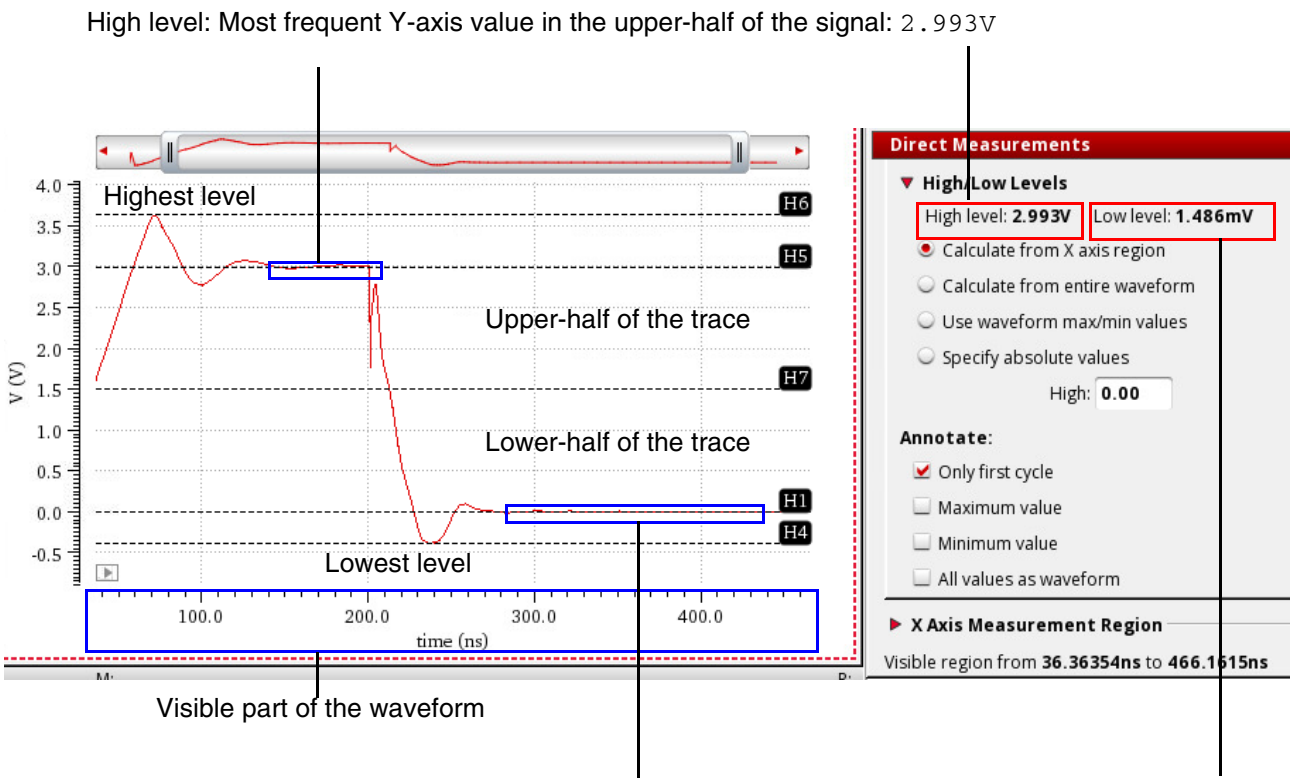
© 2023 Cadence Design Systems, Inc.

- *Calculate from entire waveform*: To automatically calculate high and low levels for the entire waveform of the signal.
- *Use waveform max/min values*: To use the maximum and minimum values of the signal as the high and low levels, respectively.
- *Specify absolute values*: To manually specify the absolute values for high and low levels.

Important points:

- High level of the signal: Most frequent Y-axis value in the upper-half of the signal.
- Low level of the signal: Most frequent Y-axis value in the lower-half of the signal.

  The following figure shows the high and low levels of the signal being automatically calculated for the currently visible range of the waveform:

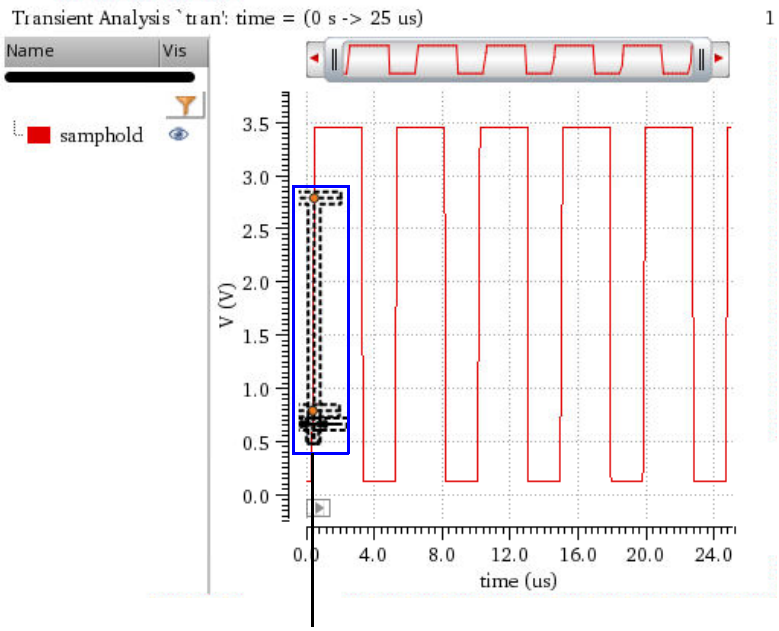


© 2023 Cadence Design Systems, Inc.

Specifying Annotate Options

In the *Annotate* section of the *Time* tab, specify how you want to annotate the measured values:

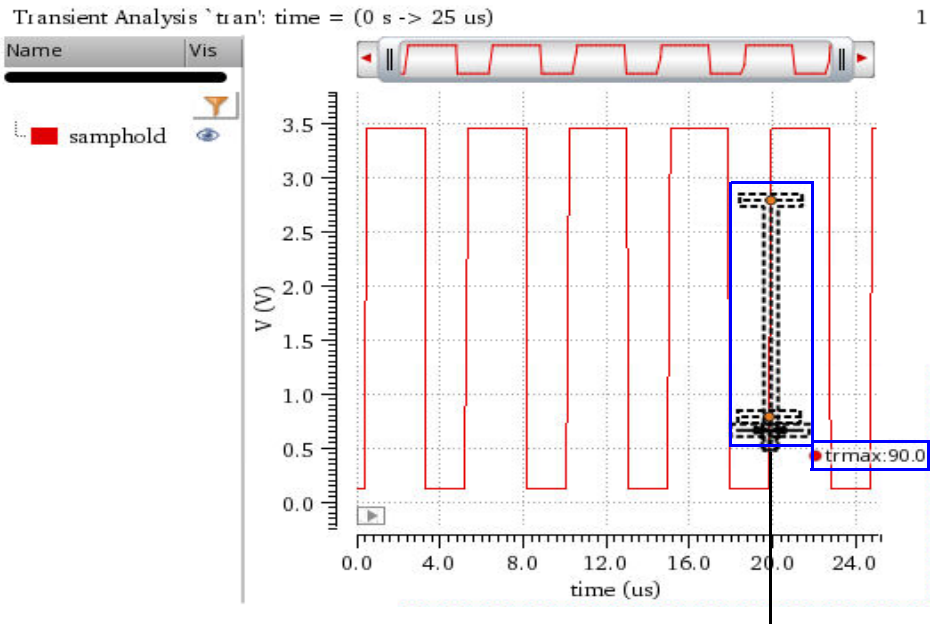
- *Only first cycle*: Annotates the measurement only for the first cycle of the signal.



Rise time measurement is annotated for the first cycle

© 2023 Cadence Design Systems, Inc.

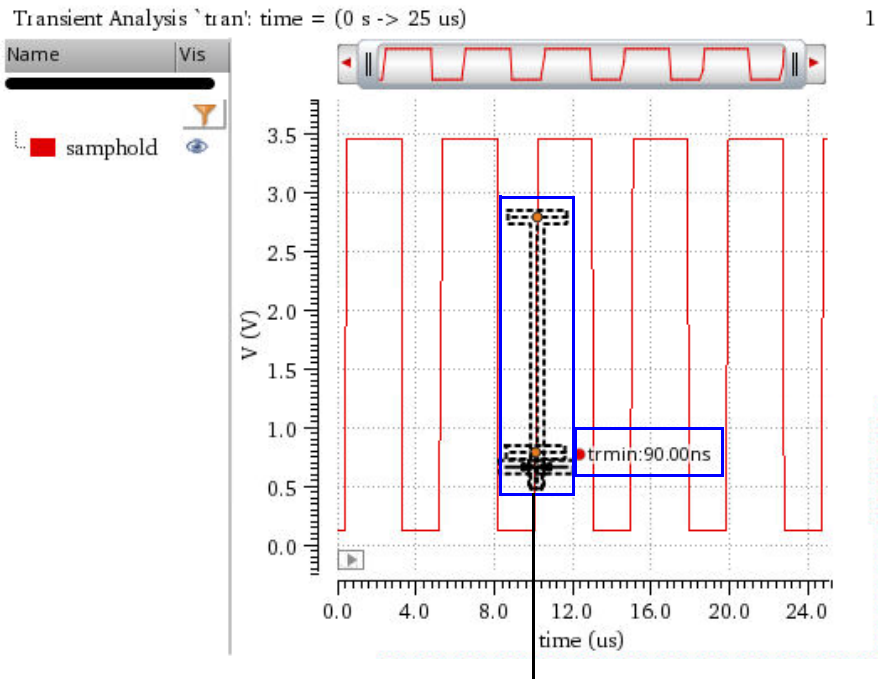
- *Maximum value*: Annotates the maximum value of the measurement.



The maximum value of rise time annotated

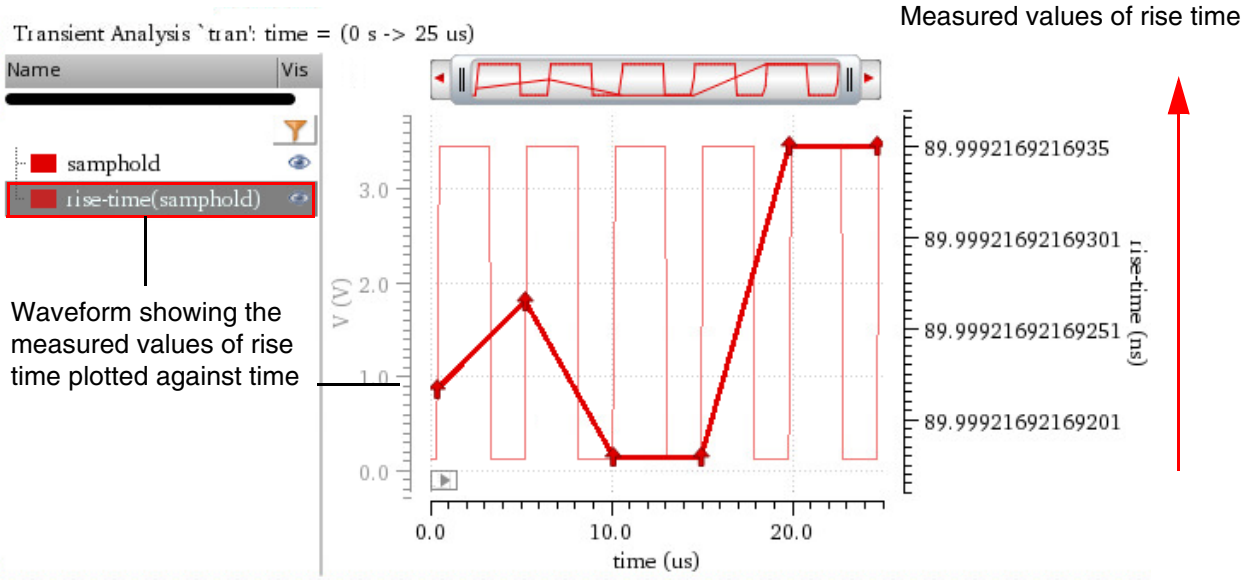
© 2023 Cadence Design Systems, Inc.

- *Minimum value*: To annotate the minimum value of the measurement.



The minimum value of rise time annotated

- *All values as waveform*: All the measurement values are plotted against time, thereby constituting a waveform.

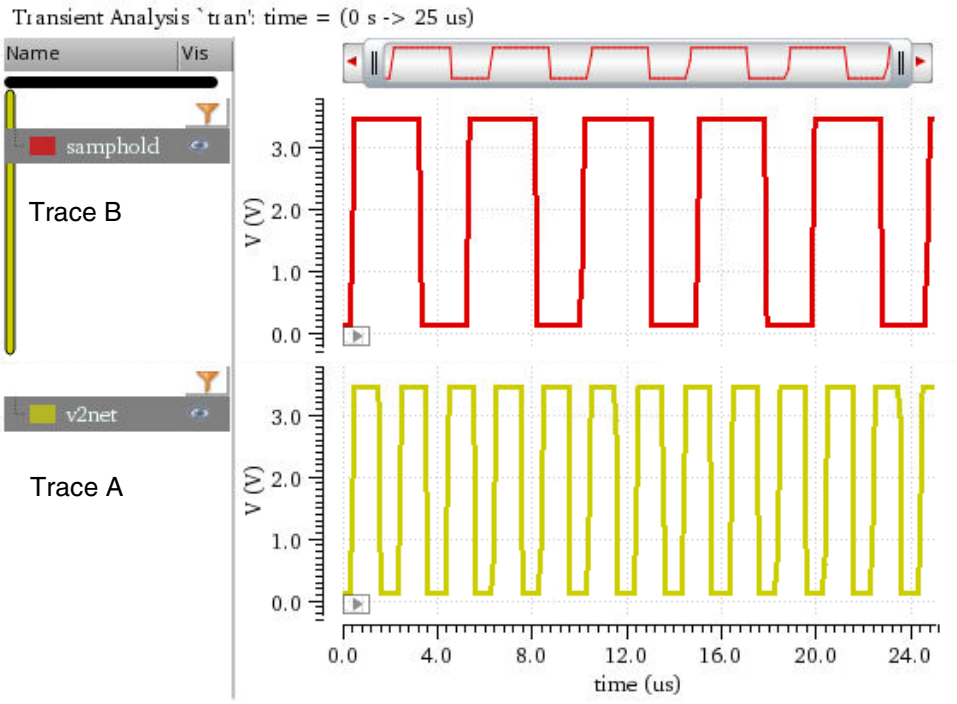


© 2023 Cadence Design Systems, Inc.

Measuring Delay between Signals

To measure the delay between two signals:

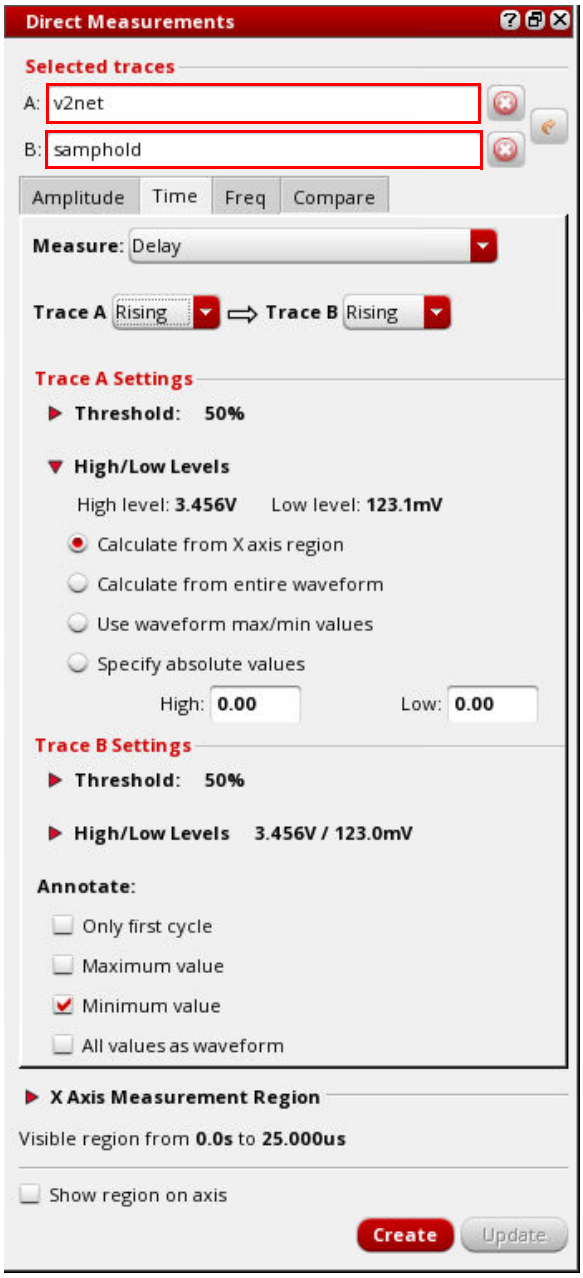
1. Plot the signals in a graph window.



2. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Window – Assistants – Direct Measurements*.

© 2023 Cadence Design Systems, Inc.

The *Direct Measurements* assistant opens, displaying the selected signals.



3. From the *Trace A* and *Trace B* drop-down lists, select the edge type of the signals.
4. In the *Trace A Settings* and *Trace B Settings* sections, specify the threshold, high, and low-level values of trace A and trace B, respectively.

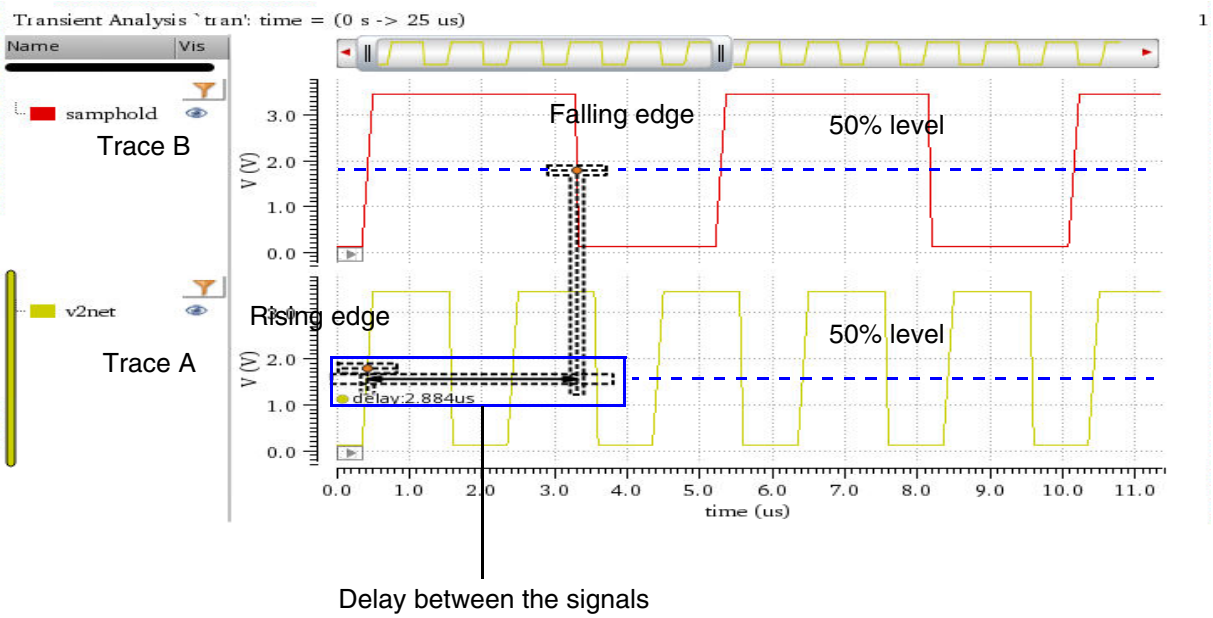
   Refer to the Specifying Threshold Values and Specifying the Method to Calculate High and Low Levels sections for more information.

© 2023 Cadence Design Systems, Inc.

5. In the *Annotate* section, select one of the following options:
   - *Only first cycle*: Annotates the measurement only for the first cycle of the signal.
   - *Maximum value*: Annotates the maximum value of the measurement.
   - *Minimum value*: To annotate the minimum value of the measurement.
   - *All values as waveform*: All measurement values are plotted against the time, thereby constituting a waveform.
6. Click *Create*.

   The delay between signals is measured and annotated in the graph window.

   The following example shows the delay measured for the 50% levels of the signals between the first rising edge of trace A and the first falling edge of trace B.



The Freq Tab

This section describes how to use the *Freq* tab of the *Direct Measurements* assistant to measure frequency-domain parameters:

- Measuring the Bandwidth of an AC Signal
- Measuring Stability Margins

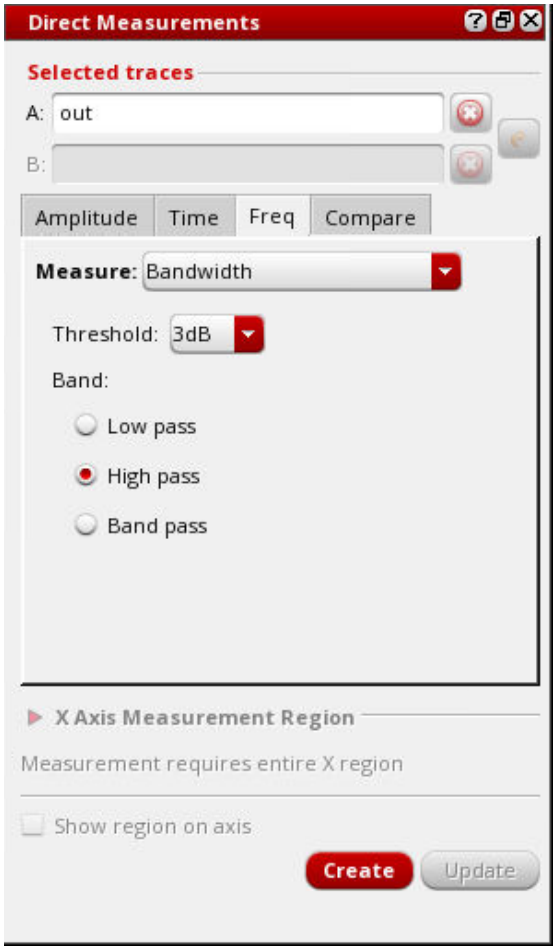
© 2023 Cadence Design Systems, Inc.

Measuring the Bandwidth of an AC Signal

To measure the bandwidth of an ac signal:

1. In a graph window, select an ac signal whose bandwidth you want to measure.
2. From the menu bar, choose *Window – Assistants – Direct Measurements*.

   The *Direct Measurements* assistant opens, displaying the selected signal.



3. On the *Freq* tab, from the *Measure* drop-down list, select *Bandwidth*.
4. From the *Threshold* drop-down, select either *3dB* or *6dB*.
5. From the Band options, select one of the following response types: *Low pass*, *High pass*, and *Band pass*.

   For more information about these options, refer to the bandwidth Calculator function.
6. Click *Create*.

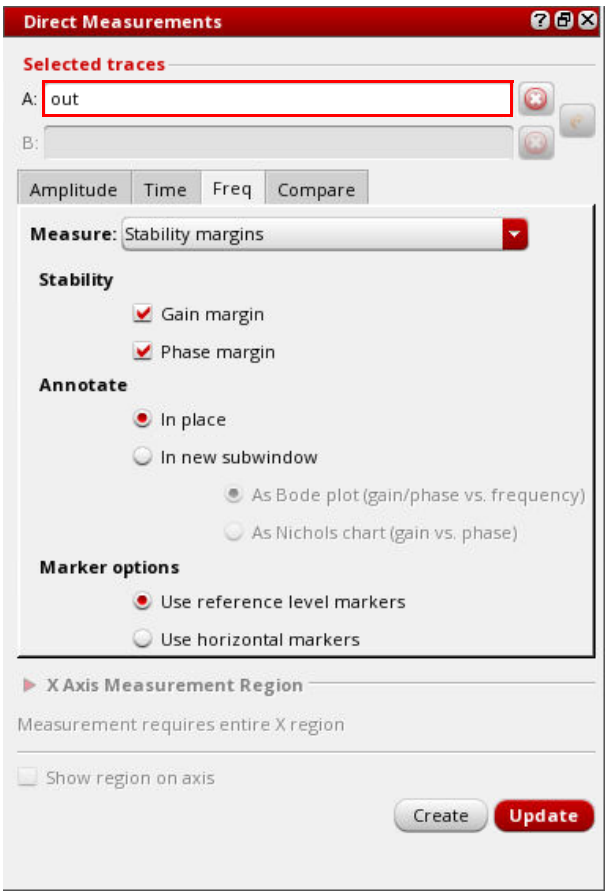
© 2023 Cadence Design Systems, Inc.

Measuring Stability Margins

To measure stability margins of an ac signal:

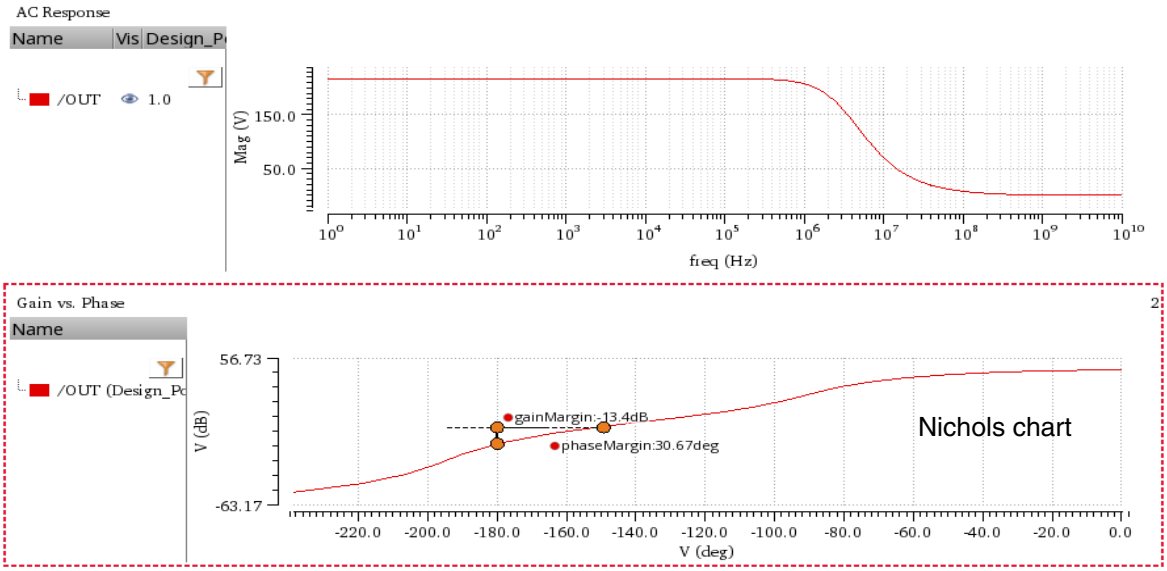
1. In a graph window, select an ac signal whose stability margins you want to measure.
2. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Window – Assistants – Direct Measurements*.

   The *Direct Measurements* assistant opens, displaying the selected signal.



3. In the *Freq* tab, from the *Measure* drop-down list, select *Stability margins*.
4. In the *Stability* section, select the stability measures by clicking the following check boxes:
   - *Gain margin*: To measure the gain margin. Refer to gainMargin for the definition.
   - *Phase margin*: To measure phase margin. Refer to phaseMargin for the definition.
5. In the *Annotate* section, select one of the following options:

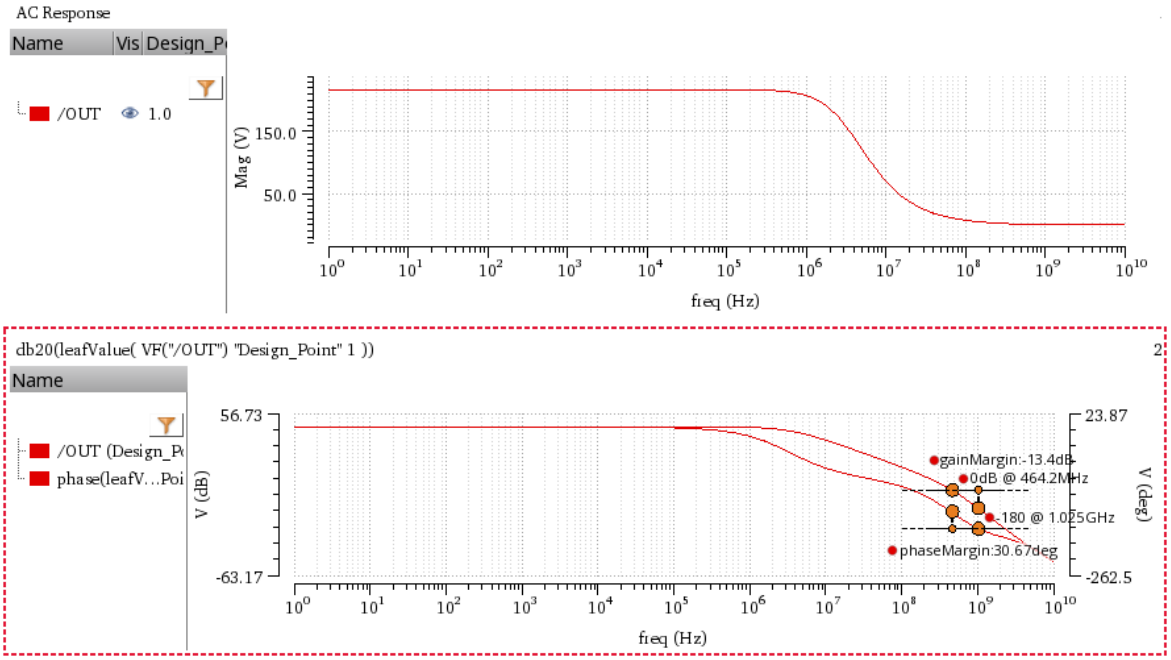
- *In place*: The gain and phase margin measurements are annotated in the same subwindow where the trace is present.
- *In new subwindow*: The gain and phase margin measurements are annotated in a new subwindow. Select one of the following options:
  - *As Bode plot (gain/phase vs. frequency)*: The gain in `dBs` and phase in `degrees` are plotted against the frequency in Hertz.
  - *As Nichols chart (gain vs. phase)*: The gain in `dBs` is plotted against phase in `degrees`. Using this chart, measures of stability (phase margin and gain margin) can be read directly.



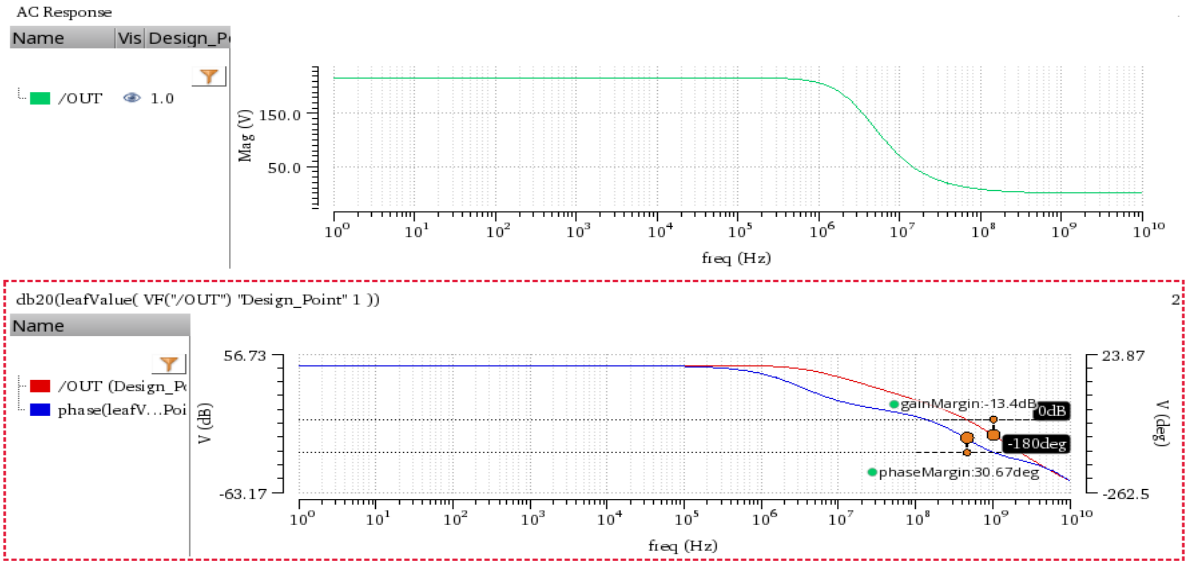
6. In the *Marker Options* section, select one of the following:

© 2023 Cadence Design Systems, Inc.

- *Use reference level markers*: The stability margins in the Bode plot are annotated using reference level markers, as shown in the following figure:



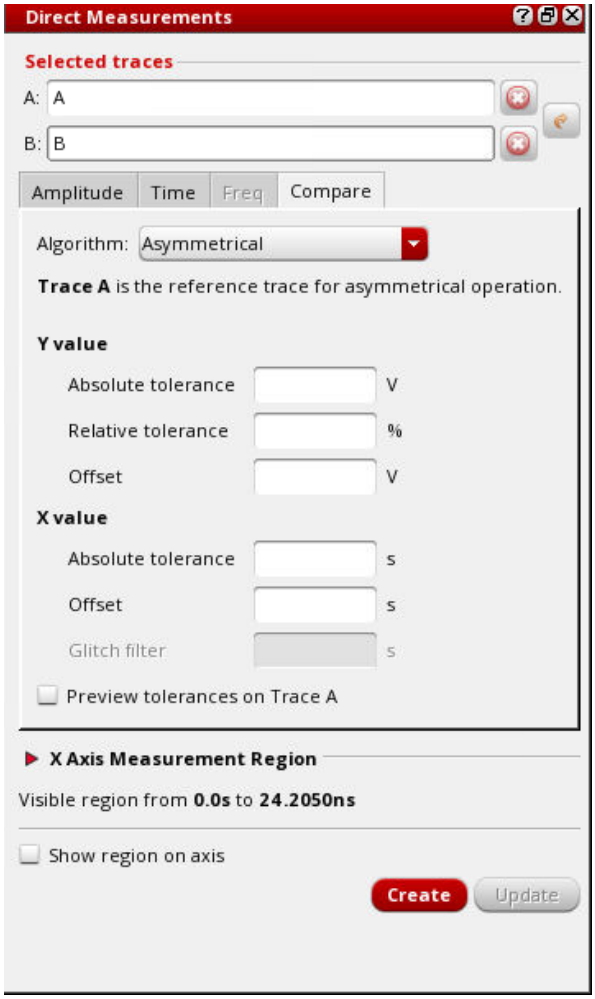
- *Use horizontal markers*: The stability margins in the Bode plot are annotated using horizontal markers, as shown in the following figure:



© 2023 Cadence Design Systems, Inc.

The Compare Tab

The following figure shows the various sections on the *Compare* tab of the Direct Measurements assistant.



Using the *Compare* tab, you can compare two waveforms, say ‘A’ and ‘B’. The first waveform ‘A’ is referred to as the reference waveform and the second waveform ‘B’ is referred to as the compared waveform.

For each data point of the reference waveform ‘A’, an acceptance or tolerance region is marked based on the absolute and relative tolerances and the offset values. If the corresponding data point on the compared waveform ‘B’ falls within the acceptance region, it is marked as pass. The data point on the compared waveform ‘B’ that falls outside the acceptance region is marked as fail. This process is repeated for all data points.

© 2023 Cadence Design Systems, Inc.

Comparison Algorithms

The Direct Measurements assistant uses the following algorithms for waveform comparison:

- *Asymmetrical*

  This is the default algorithm used for waveform comparison. In this algorithm, comparison of waveform is done only once. The first waveform is considered as the reference waveform and the second waveform is considered as the waveform to be compared.

  Note: This shows a better performance as compared to the other algorithms and is sufficient for most of the cases.

- *Conservative (symmetrical)*

  In this algorithm, the comparison of waveforms is done twice—first, the reference waveform is compared with the compared waveform and then, the compared waveform is compared with the reference waveform. Data points that fail in any of the comparisons are considered as failed.

Comparisons	Failed data points	Final result of comparisons
■ A as reference waveform ■ B as compared waveform	1, 3, 5	1, 2, 3, 4, 5, 6
■ B as reference waveform ■ A as compared waveform	2, 4, 6	

- *Liberal (symmetrical)*

  Similar to the conservative algorithm, in this algorithm too, the comparison of waveforms is done twice. Data points that fail in both the comparisons are considered as failed.

Comparisons	Failed data points	Final result of comparisons
■ A as reference waveform ■ B as compared waveform	1, 2, 3, 4, 5	1, 4
■ B as reference waveform ■ A as compared waveform	1, 4, 7	

For more information about these comparison algorithms, refer to the Waveform Compare Algorithms section the Virtuoso ADE Assembler User Guide.

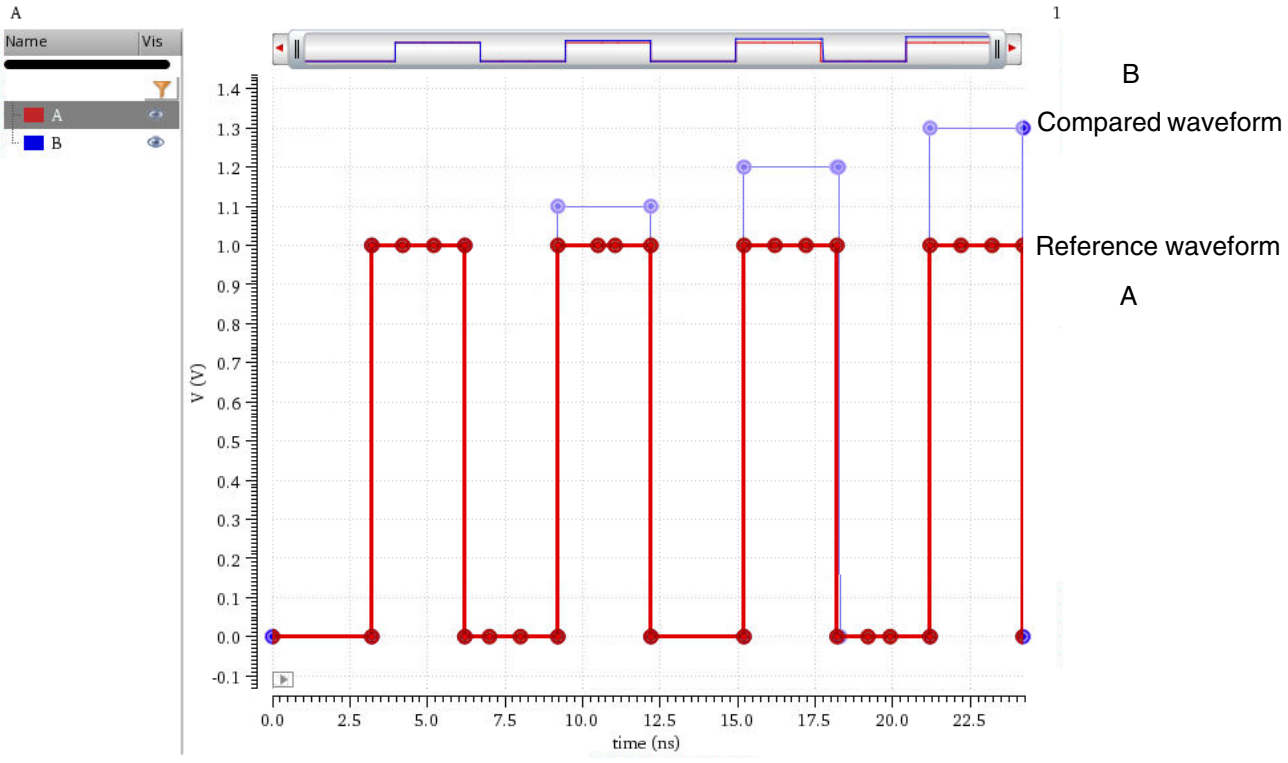
© 2023 Cadence Design Systems, Inc.

Acceptance or Tolerance Region

Acceptance or tolerance region is calculated based on the following values:

- *Absolute tolerance*: The tolerance value of the absolute deviation allowed between the waveforms.
- *Relative tolerance*: The % tolerance value of the relative deviation allowed between the waveforms.
- *Offset*: The offset value by which the acceptance or tolerance region is shifted along y axis or x axis. It is an integer value. The positive offset shifts the acceptance region upwards on the y axis or towards right on the x axis. The negative offset shifts the acceptance region downwards on the y axis or towards left on the x axis.

Consider the following waveforms, 'A' and 'B', where 'A' is the reference waveform and 'B' is the waveform to be compared.



The following examples describe how tolerance or acceptance region is calculated based on the different values of *Absolute tolerance*, *Relative tolerance*, and *Offset*.

© 2023 Cadence Design Systems, Inc.

Example 1

This example describes the calculation of the acceptance region based on the following values:

	Y value	X value
Absolute tolerance	`0.15V`	`0ns`
Relative tolerance	`10%`	N/A
Offset	`0V`	`0ns`

Consider the data point 1 at `1V` on the reference waveform '`A`'.

y-deviation range based on Absolute tolerance: `1±0.15V` = `0.85V–1.15V`

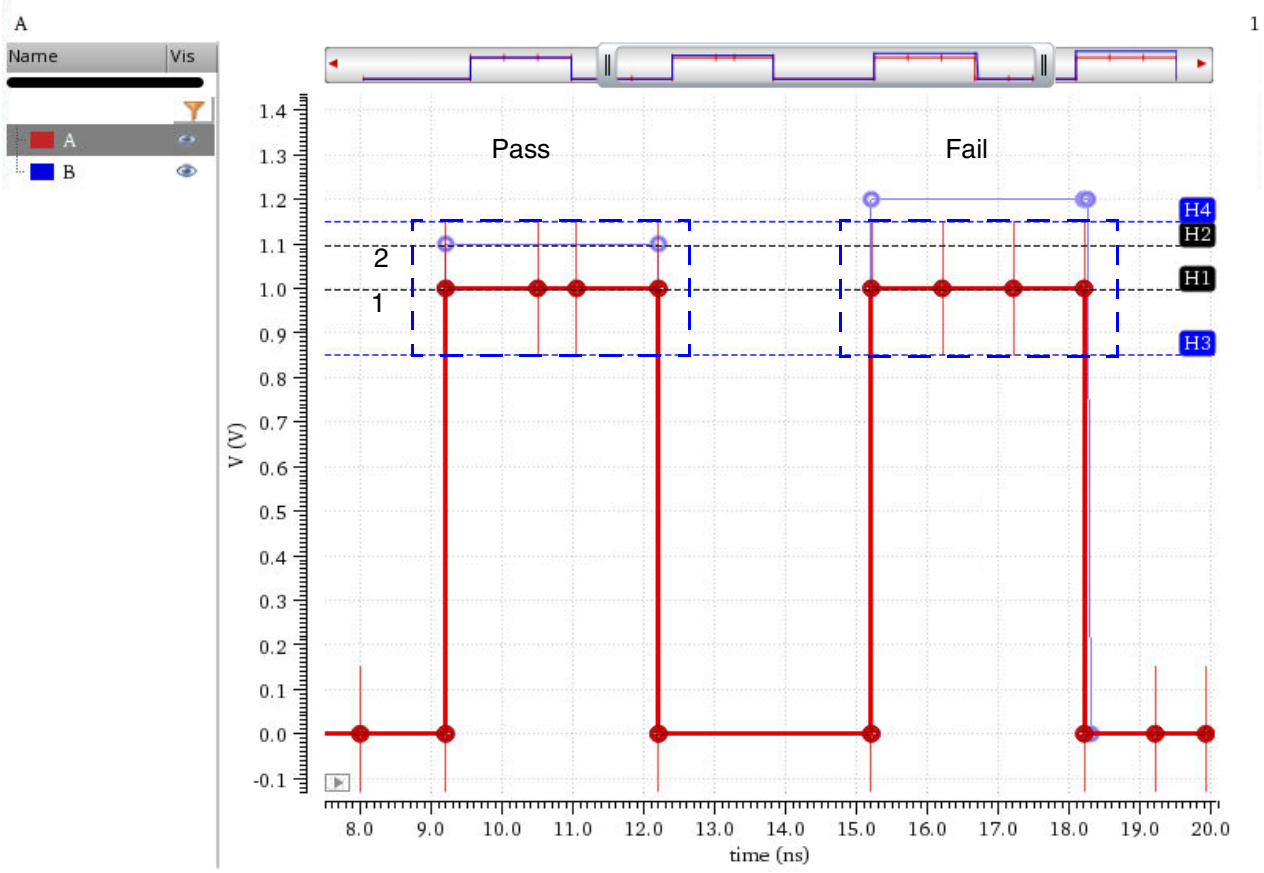
y-deviation range based on Relative tolerance: `1±(10%` of `1V)` = `1±0.1` = `0.9V–1.1V`

The maximum deviation range of the two is considered as the acceptance or tolerance region: `0.85V–1.15V`

Note: There is no shift in the acceptance region because the *Offset* value is not specified

© 2023 Cadence Design Systems, Inc.

Observe that the points that fall within the acceptance region are marked as `pass`, while those fall outside the acceptance region are marked as `fail`.



Example 2

This example describes the calculation of the acceptance region based on the following values:

	Y value	X value
Absolute tolerance	`0.15V`	`0ns`
Relative tolerance	`10%`	N/A
Offset	`0.1V`	`0ns`

© 2023 Cadence Design Systems, Inc.

Consider the data point 1 at `1V` on the reference waveform ‘A’.

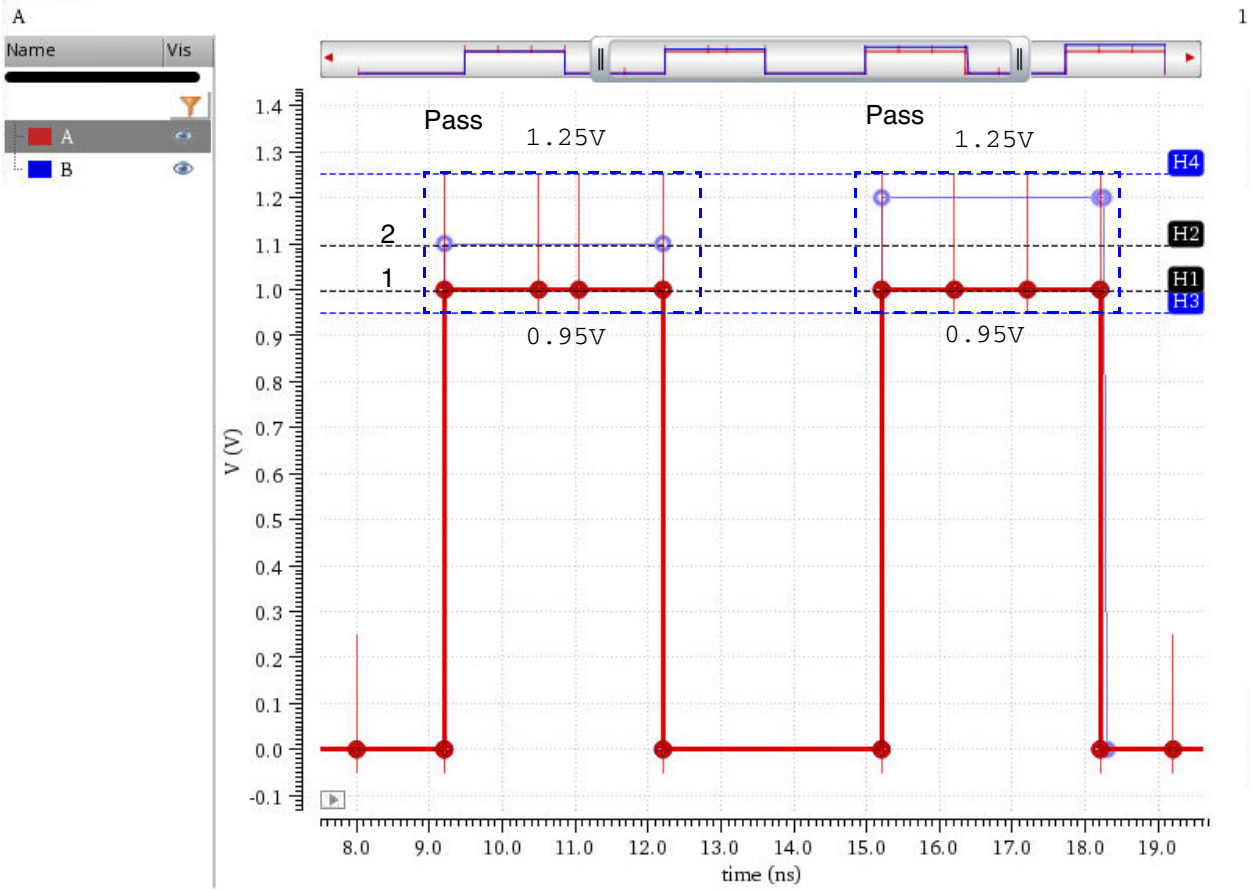
y-deviation range based on *Absolute tolerance*: `1±0.15V` = `0.85V–1.15V`

y-deviation range based on *Relative tolerance*: `1±(10%` of `1V)` = `1±0.1` = `0.9V–1.1V`

The maximum deviation range of the two is `0.85V–1.15V`

The acceptance or tolerance after considering the *Offset* value of 0.1V: `(0.85+0.1)V–(1.15+0.1)V = 0.95V–1.25V`

All the data points fall within the acceptance region, therefore no points of failure are reported.



© 2023 Cadence Design Systems, Inc.

Example 3

This example describes the calculation of the acceptance region based on the following values:

	Y value	X value
Absolute tolerance	`0.15V`	`0.2ns`
Relative tolerance	`0%`	N/A
Offset	`0.1V`	`0ns`

Consider the data point ‘1’ at `1V` on the reference waveform ‘A’.

y-deviation range based on *Absolute tolerance*: `1±0.15V` = `0.85V–1.15V`

y-deviation range based on *Relative tolerance*: `1±(10% of 1V)` = `1±0.1` = `0.9V–1.1V`

The maximum deviation range of the two is: `0.85V—1.15V`

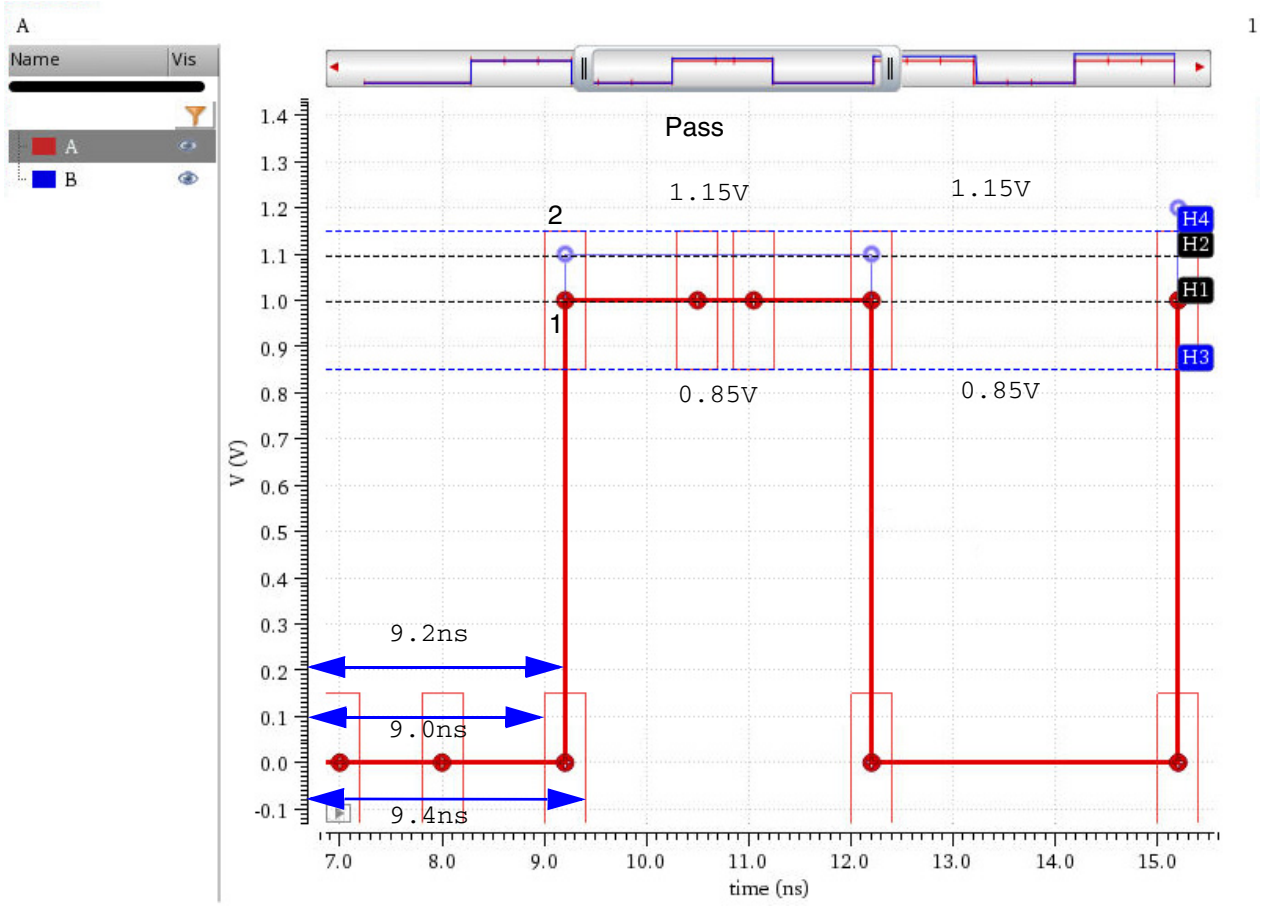
The acceptance or tolerance after considering the *Offset* value of 0.1V: `(0.85+0.1)V–(1.15+0.1)V = 0.95V—1.25V`

x-deviation range based on the *Absolute tolerance* (time tolerance): `9.2±0.2ns` = `9ns–9.4ns`

Observe that if you specify absolute tolerance values for both x and y axes, the acceptance region is marked as rectangular boxes. Unlike the vertical lines that appear when you specify tolerance values only for the y axis.

© 2023 Cadence Design Systems, Inc.

All the data points fall within the acceptance region, therefore no points of failure are reported.



Comparing Waveforms

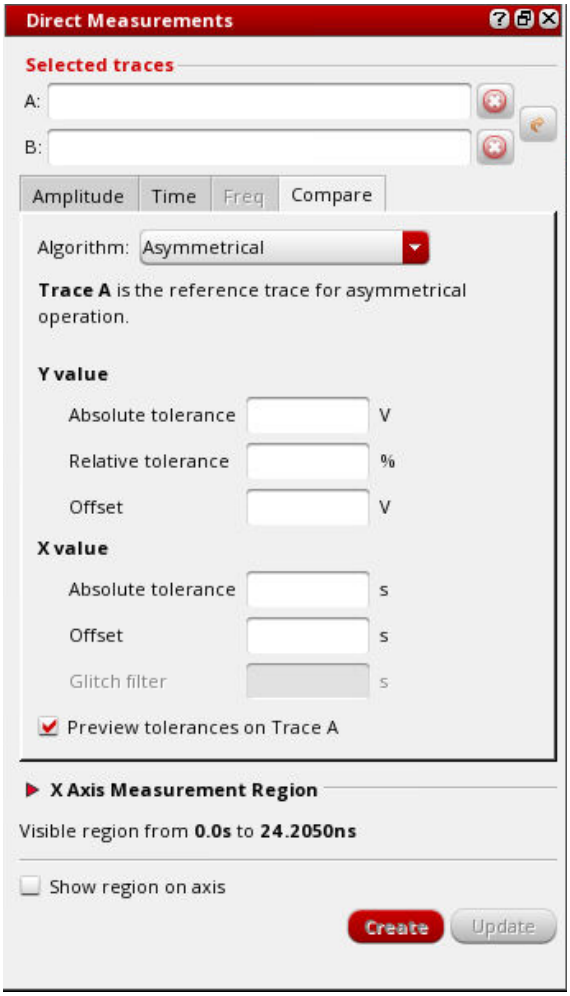
To compare waveforms:

1. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Window – Assistants – Direct Measurements*.

   The *Direct Measurements* assistant opens.

© 2023 Cadence Design Systems, Inc.

2. Click the *Compare* tab.



3. Click the *A* field, and then select the first waveform in the graph window or the subwindow. This waveform is considered as the reference waveform.

4. Click the *B* field, and then select the second waveform in the graph window or the subwindow. This waveform is considered as the compared waveform.

5. From the *Algorithm* drop-down list, select a comparison algorithm.

   See the Comparison Algorithms section for more information.

6. In the *Y value* group, specify the *Absolute tolerance*, *Relative tolerance*, and *Offset* values for y axis.

   See the Acceptance or Tolerance Region section for more information.

© 2023 Cadence Design Systems, Inc.

7. In the *X value* group, specify the *Absolute tolerance*, *Offset, and Glitch filter* values for x axis.
8. In the *Glitch filte*r field, specify the maximum glitch size (in seconds) up to which the glitches in digital signals are acceptable and ignored while comparing waveforms.

   Note: This field appears dim and becomes available only when you compare digital signals.
9. Select the *Preview tolerances on Trace A* check box if you want to preview the tolerance or acceptance regions.
10. In the *X Axis Measurement Region* section, specify the measurement region on the x axis.

    See the Specifying the Measurement Region section for more information.
11. Select the *Show region on axis* check box to display an extent marker over the x axis, highlighting the region of measurement.
12. Click *Create* to compare the selected waveforms.

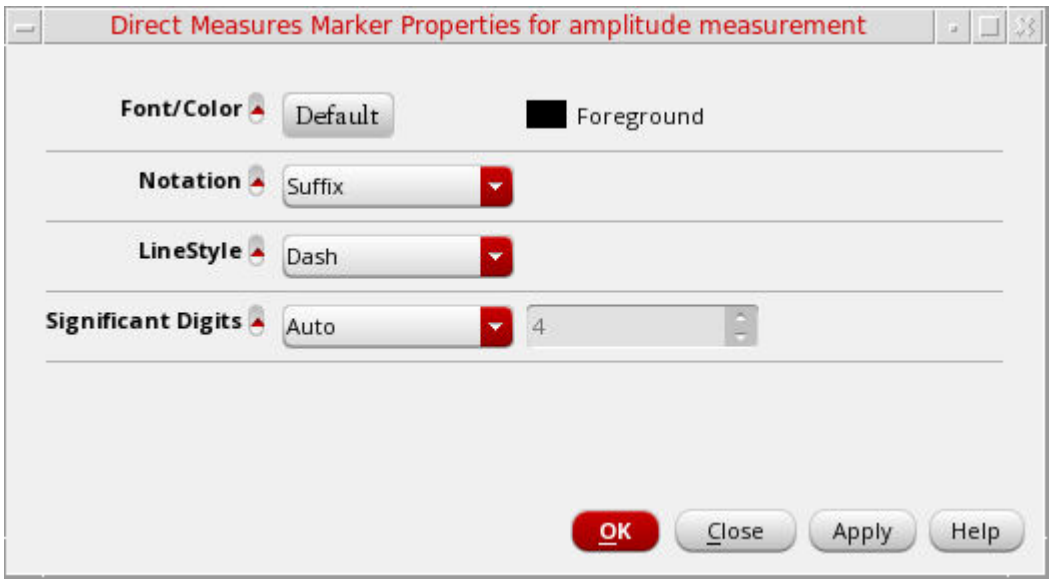
© 2023 Cadence Design Systems, Inc.

Editing Properties of Direct Measurements

To edit properties of a direct measurement:

1. Right-click a direct measurement marker, label, or bookmark and then choose *Direct Measurement Properties*.

   The Direct Measures Marker Properties for <*measurement name*> form appears.



2. Edit the following properties on the form, as required:

   a. *Font/Color* — Specifies the font and color of the direct measurement marker, label, or bookmark.

   b. *Notation* — Specifies the notation of the direct measurement. Possible values are *Scientific*, *Engineering*, and *Suffix*. The default value is *Suffix*.

   c. *LineStyle* — Specifies the line style of the direct measurement marker, label, or bookmark. Possible values are *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. The default value is *Solid*.

   d. *Significant Digits* — Specifies the number of significant digits to be displayed in the measured values. The default value is *4*. To change this value, you need to first select *Manual* from the drop-down list.

3. Click *OK* to save the changes and close the form.

© 2023 Cadence Design Systems, Inc.

Deleting Direct Measurements

To delete the annotations of direct measurements, right-click a direct measurement marker, label, or bookmark, and then choose *Delete Direct Measurements*.

Customize Trace Groups Assistant

This assistant is used to customize the trace settings for the traces that belong to a common family.

Perform the following step to open the Customize Trace Group assistant:

➡ Choose *Window – Assistants – Customize Trace Groups.*

The traces in the graphs are updated based on the filter value you select in each field. The filter values can be: `none`, `trace`, `leaf`, `family`, `Corner`, and sweep variables.

- *Strip*—Specify how you want to display the traces in a strip. Traces that belong to the filter value you select are displayed in the same strip. For example, if you select `family`, all the traces that belong to a common family are displayed in the same strip.
- *Color*—Specify the color of the traces. The traces that belong to the specified filter value you select are displayed in the same color. Traces can be displayed in 18 different colors and after that the colors are repeated.
- *Symbol*—Specify how symbols are to be shown on traces. The traces that belong to the specified filter value you select display the same type of symbols. By default, the symbol

© 2023 Cadence Design Systems, Inc.

type is plus. To change the symbol type, right-click the trace and select the symbol type from the *Symbol* menu.

- *Linestyle*—Specify how you want to set the line style of traces. The traces that belong to the specified filter are displayed using the same line style.
- *All Subwindows*—Select this check box to update the traces in all subwindows based on the options you select in this form.
- Click *Apply*.

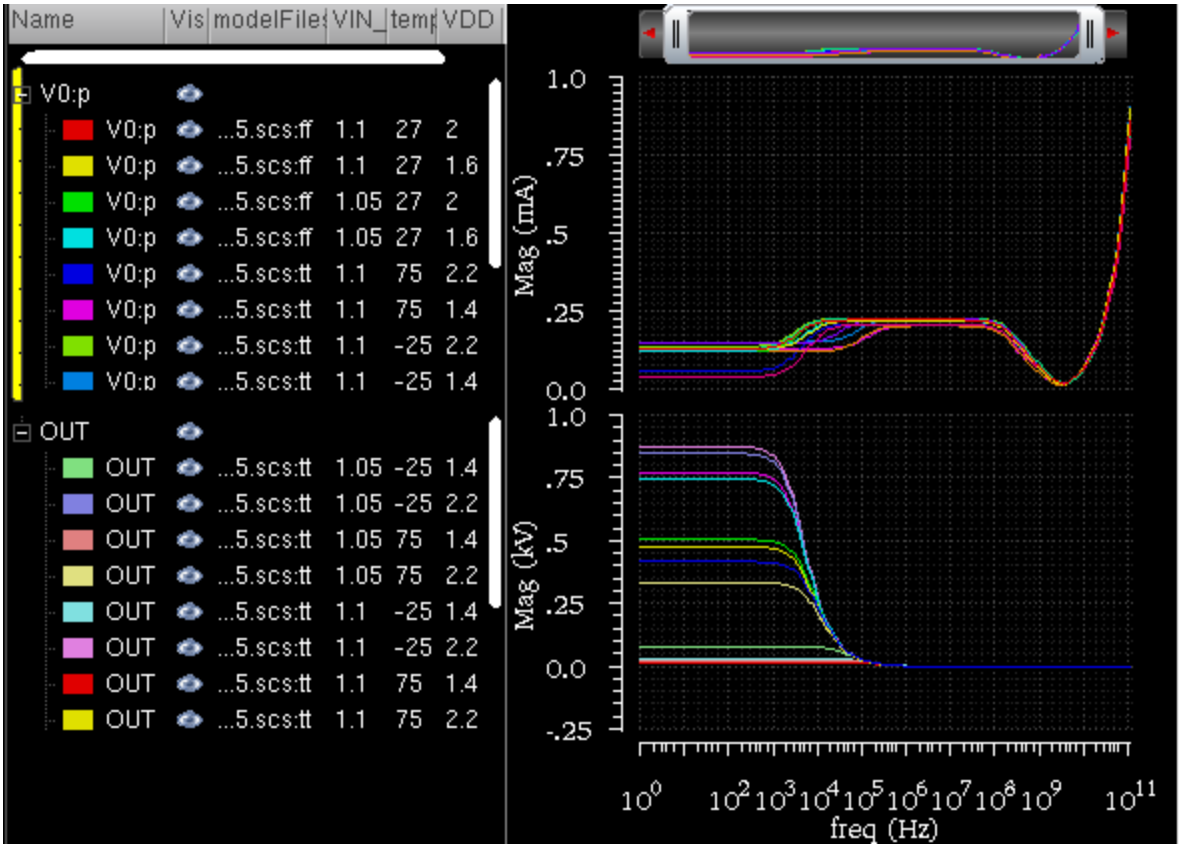
Note: You can also view a related blog post to know more about how to organize waveform families.

Example1: Displaying traces belonging to a common family in one strip:

Set the fields in the Customize Trace Settings assistant as shown in the figure below:

© 2023 Cadence Design Systems, Inc.

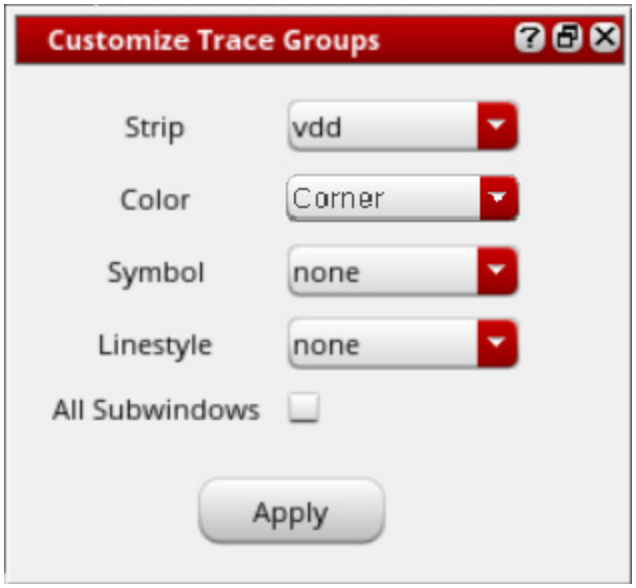
When you click the *Apply* button, the traces in the graph are updated as shown in the figure below. Notice that traces belonging to a common family (`V0:p` and `OUT`) are displayed in the same strip.

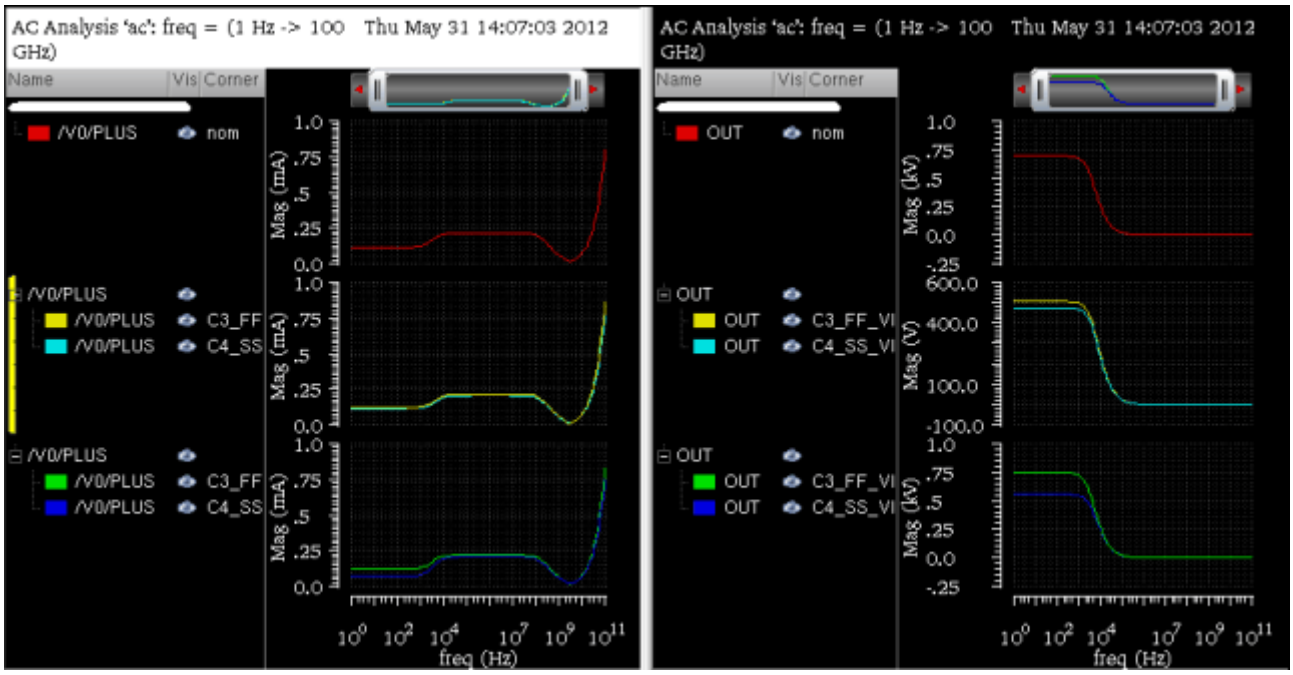


© 2023 Cadence Design Systems, Inc.

Example2*: Displaying traces that belong to a common VDD value in one strip and common corners in same color:*

Set the fields in the Customize Trace Settings assistant as shown in the figure below:



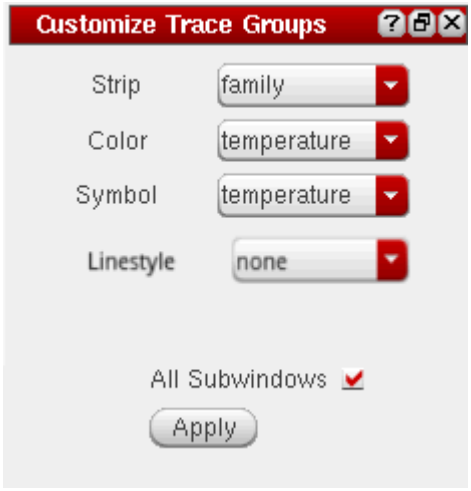


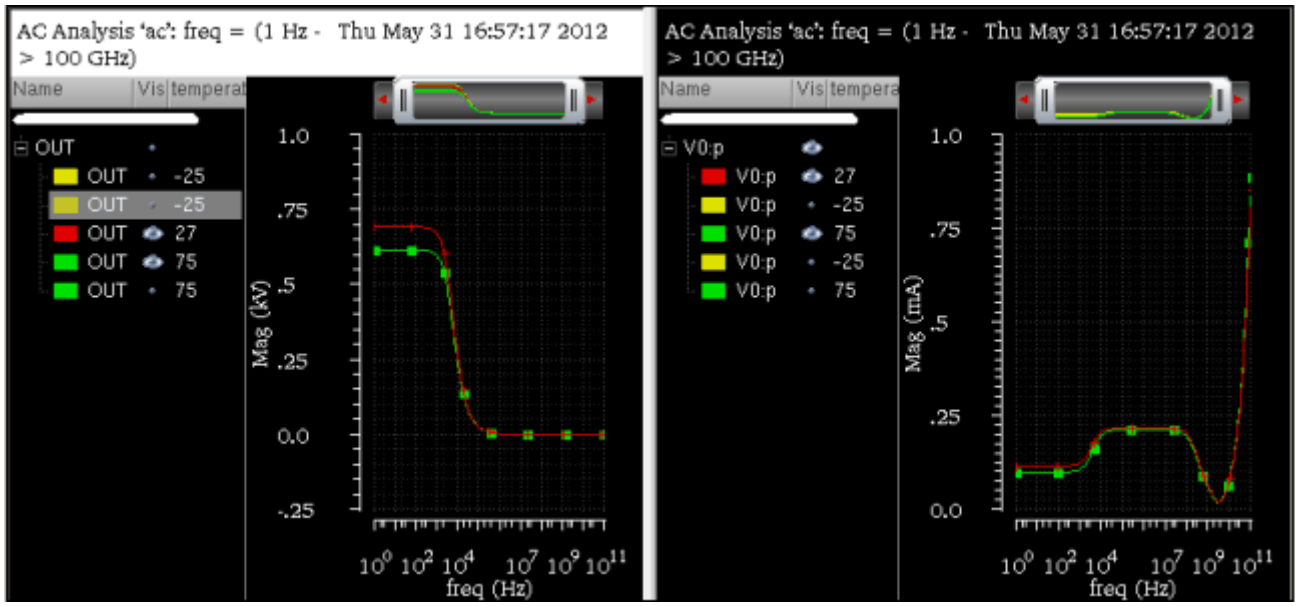
In this example, notice that the traces with similar VDD value are displayed in the same strip and traces that belong to similar corners are displayed in same color.

© 2023 Cadence Design Systems, Inc.

Example3: Displaying traces that belong to a common temperature value with same type of symbols displayed on them

Set the fields in the Customize Trace Settings assistant as shown in the figure below:





In this example, notice that the traces with similar temperature value are displayed in the same color and also display the similar types of symbols on them.

Displaying Traces having Same Values for a Sweep Variable in the Same Linestyle

To display traces that have same values for a sweep variable using the same line style:

© 2023 Cadence Design Systems, Inc.

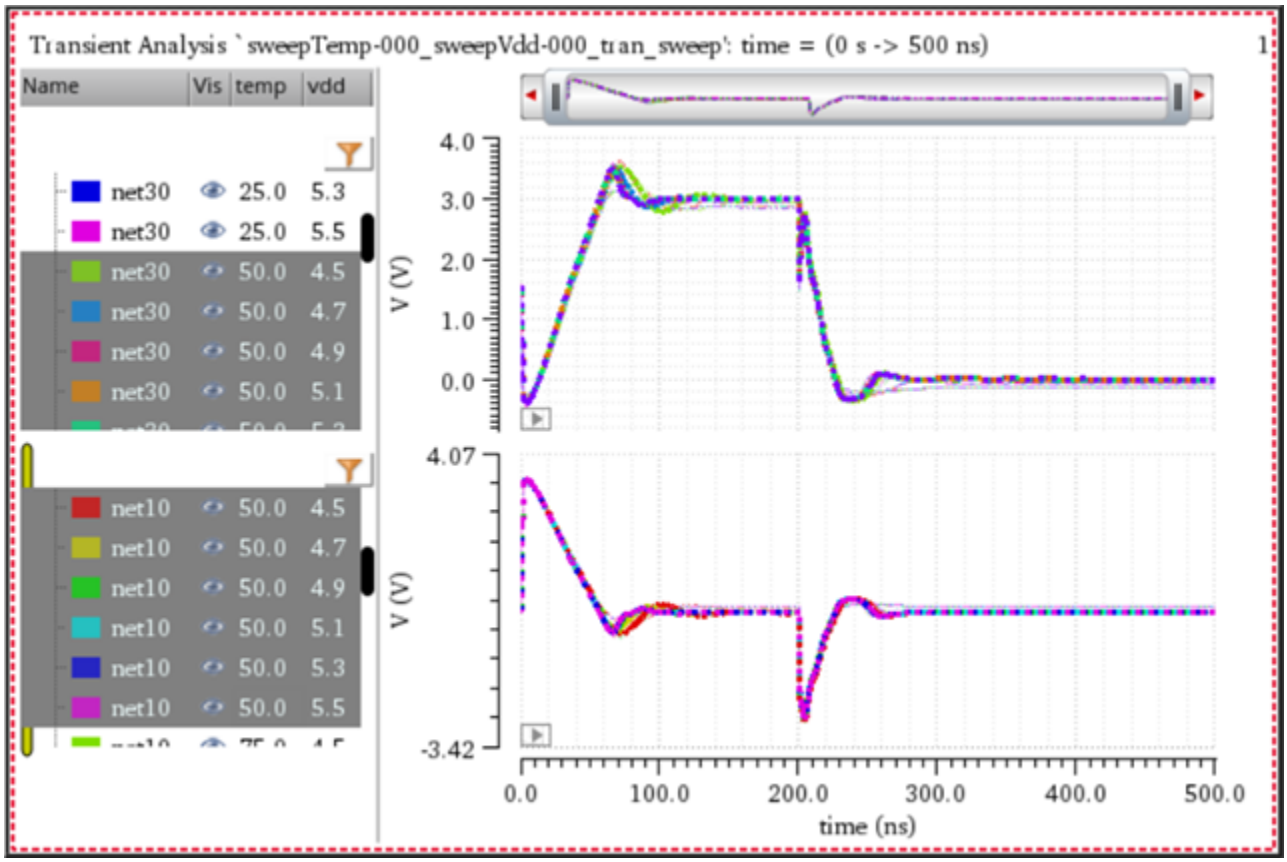
1. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Window—Assistants—Customize Trace Groups*.

   The Customize Trace Groups assistant opens.

2. In the Customize Trace Groups assistant, set the fields as follows:
   - *Strip*: *Family*
   - *Color*: *None*
   - *Symbol*: *None*
   - *Linestyle*: *temp*
3. Click *Apply*.

The traces that have the same value for the sweep variable `temp` are displayed in the same line style.

In the following example, all the traces of family `net30` and `net10` that have same value `50` for the sweep variable `temp`, are displayed using the same line style, `dot`.



© 2023 Cadence Design Systems, Inc.

Working with Workspaces

A workspace is the arrangement of various assistants and the window settings that you specify while working with a graph. You can either use the available workspaces or create your own workspace while working in the window.

Workspace Types

The available workspaces are of four types:

- **Basic**—This workspace displays the following dockable assistants:
  - Subwindows
  - Results Browser
  - Graph
- **Browser**—This workspace displays the following dockable assistants:
  - Results Browser
  - Graph
- **Classic**—This workspace displays only the graph window.
- **MarkerTable**—This workspace displays the following dockable assistants:
  - Subwindows
  - Results Browser
  - Graph
  - Marker Table
- **TM**—This workspace displays the following dockable assistants:
  - Subwindows
  - Results Browser
  - Graph
  - Transient Measurement Assiatant

If you open the tool in stand-alone mode, it is opened in the `Basic` workspace because in stand-alone mode you work on the saved simulation results.

© 2023 Cadence Design Systems, Inc.

If you open the tool from within ADE, it is opened in the `Classic` workspace because in ADE you work on the simulation results for the current run.

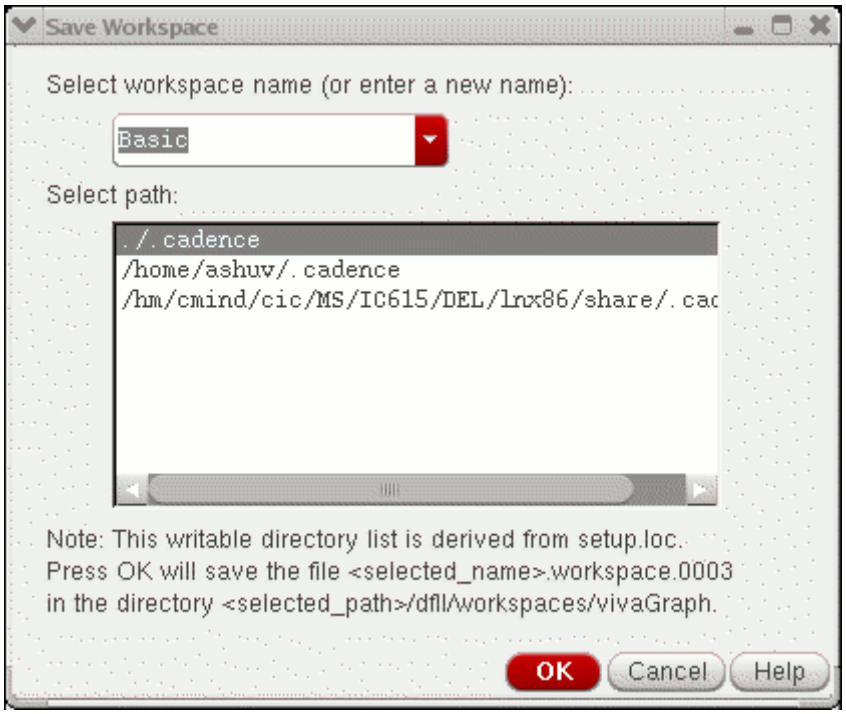
When you open a new Virtuoso Visualization and Analysis XL session from the same Virtuoso window, the workspace that you specified in the previous session is available.

Saving a Workspace

You can customize a workspace by selecting the assistants that you want to display from the *Window – Assistants* menu. You can then save the customized workspace by doing one of the following:

- Choose *Window – Workspaces – Save As.*
- On the Workspace toolbar, select the option.

  The *Save Workspace* form appears.



In this form, specify the name with which you want to save the workspace and select the path where you want to save the workspace. You can specify a new name or can make changes to an existing workspace.

© 2023 Cadence Design Systems, Inc.

If you do not want to save tWorking With Assistants on page 225he changes you made to the existing workspace, choose *Windows – Workspaces – Revert to Saved* to revert to the factory settings.

Loading a Workspace

To load a workspace, do one of the following:

- Choose *Windows – Workspaces – Load.*

  The *Load Workspace* form appears. In this form, select the workspace you want to load.

- On the Workspace toolbar, select the required workspace from the *Workspace* drop-down list box.

Deleting a Workspace

To delete a workspace, choose *Windows – Workspaces – Delete.*

The *Delete Workspace* form appears. Select the name of the workspace that you want to delete.

Setting the Default Workspace

To set a workspace as the default workspace, choose *Windows – Workspaces – Set Default.*

The *Default Workspace* form appears. Select the name of the workspace that you want to set as the default.

Showing and Hiding Assistants

To show or hide the assistants in the workspace, do one of the following:

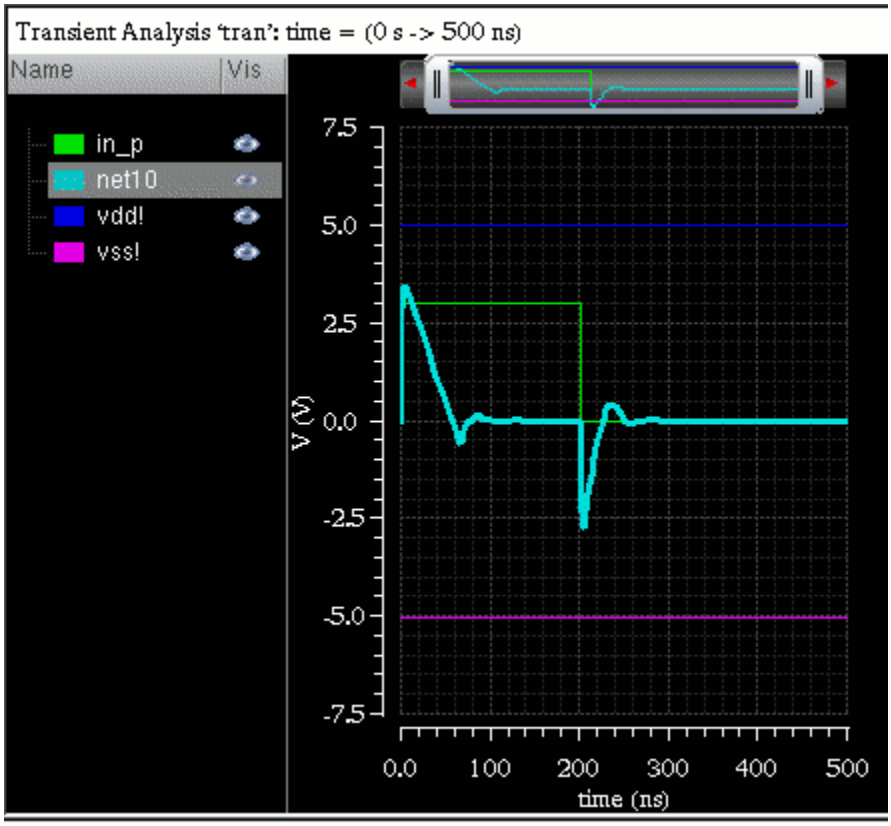
- Choose *Windows – Workspaces – Show/Hide Assistants*
- Press the `F11` key to hide or show the assistants.
- On the Workspace toolbar, select the option.

© 2023 Cadence Design Systems, Inc.

Working with Traces

A signal when plotted in the window is called a trace. Each trace in the window is displayed in a different color. The graph supports 18 unique colors. Information, such as, *Name*, *Vis* (visibility of the trace), *Spec*, *Region*, *History*, *Test*, and *Design Point*, can be displayed as columns in the area to the left of the trace is called the trace legend. A splitter line separates the trace and the trace legend area. You can double-click the splitter line to adjust the width of the trace legend area.

To highlight a trace in the window, select the legend corresponding to the trace or select the trace in the graph.



The trace legend area displays information in the following columns:

- *Name*: Displays the name and color of all the traces plotted in a graph.
- *Vis*: Displays the icon to toggle the visibility of the traces.
- *Spec*: Displays the icon to toggle the visibility of the specifications of the traces.
- *History*: Displays the name of the histories.

© 2023 Cadence Design Systems, Inc.

- *Test*: Displays the name of the tests.

The corner names and sweep parameters from ADE in separate columns that can be sorted.

The trace families, based on sweep parameters, in a hierarchical order. To view the analog traces in a family or the digital traces in a bus, click the + symbol in the trace legend area.

© 2023 Cadence Design Systems, Inc.

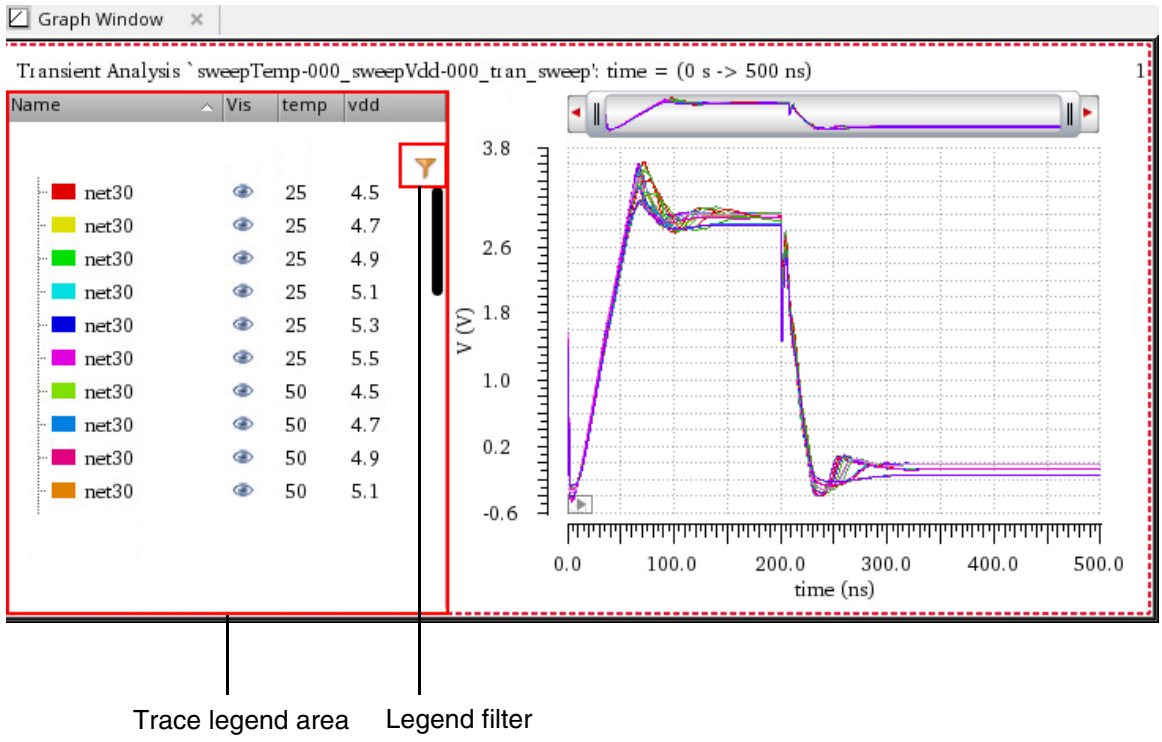
Filtering Trace Legends

You can use the *Legend Filter* form to:

- Apply filters to selectively view the traces that you want to work upon
- View only the filtered traces in the graph window or subwindow
- View information, such as history, test, and corner, about the traces plotted in a graph

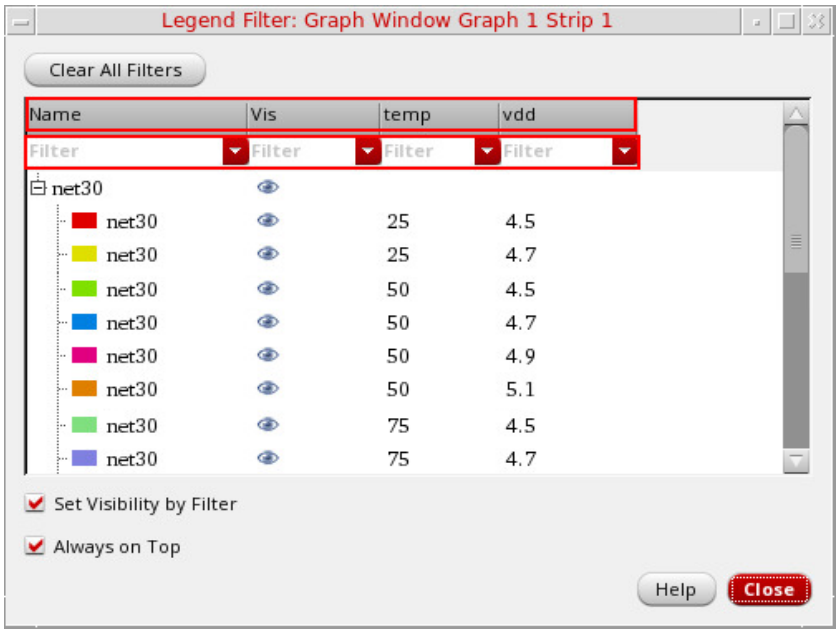
Opening the Legend Filter Form

➡ In the trace legend area, click the legend filter icon, , to open the *Legend Filter* form, as shown in the following figure.



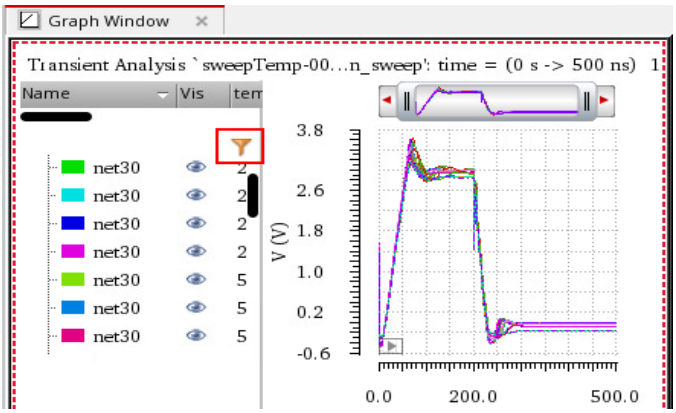
© 2023 Cadence Design Systems, Inc.

The *Legend Filter* form opens.



Note: The title of the *Legend Filter* form is `Legend Filter: <window name> Graph <graph number> Strip <strip number>`. For example, `Graph Window Graph 1 Strip 1`.

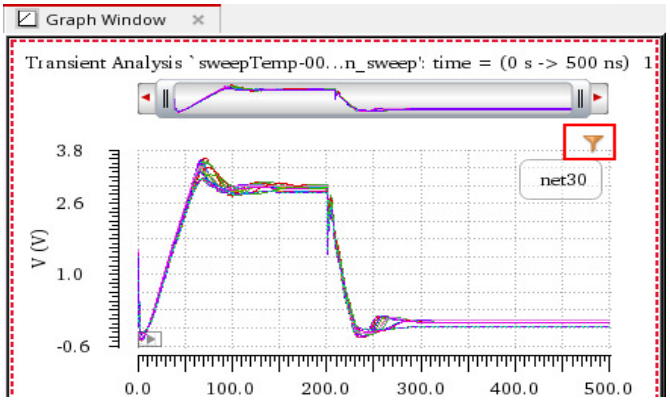
The location of the legend filter icon, , depends on the legend position specified on the *Graph Options* tab of the *Graph Properties* form. The following table shows the location of legend filter icon for different legend positions.

Legend Position	Location of the Legend Filter Icon
Left	

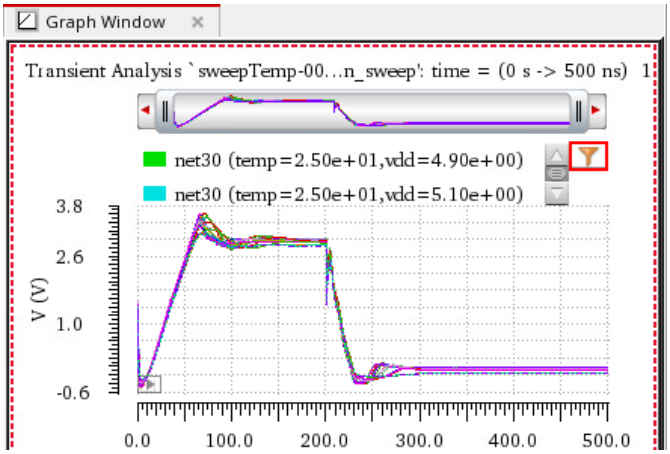
© 2023 Cadence Design Systems, Inc.

Legend Position **Location of the Legend Filter Icon**

Inside



Above



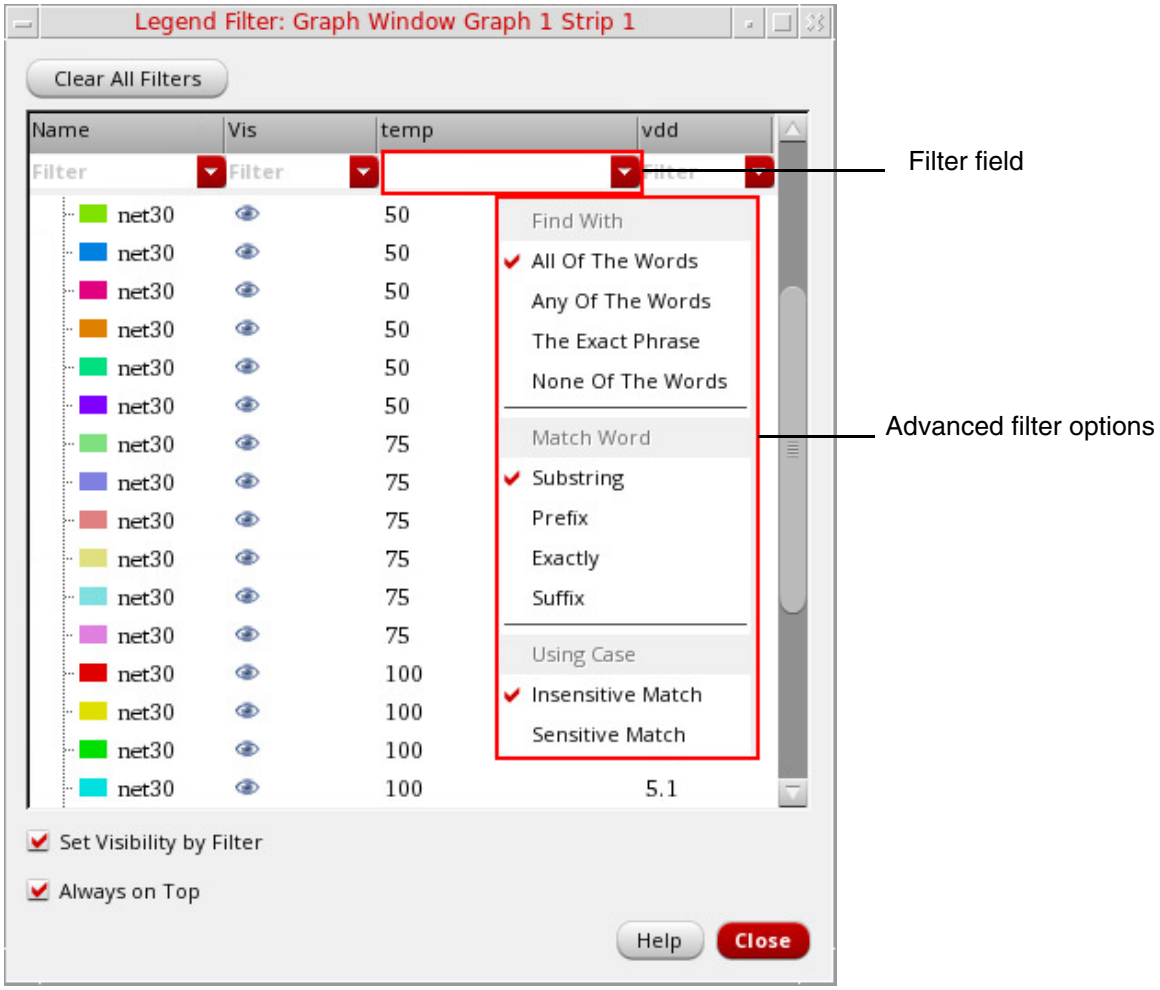
© 2023 Cadence Design Systems, Inc.

Applying Filters to Traces

You can apply filters to the traces in the *Legend Filter* form to selectively view the traces that you want to work upon.

Perform the following steps to apply filters to the traces:

1. Right-click anywhere in a filter field to specify the advanced filter options, as shown in the following figure.



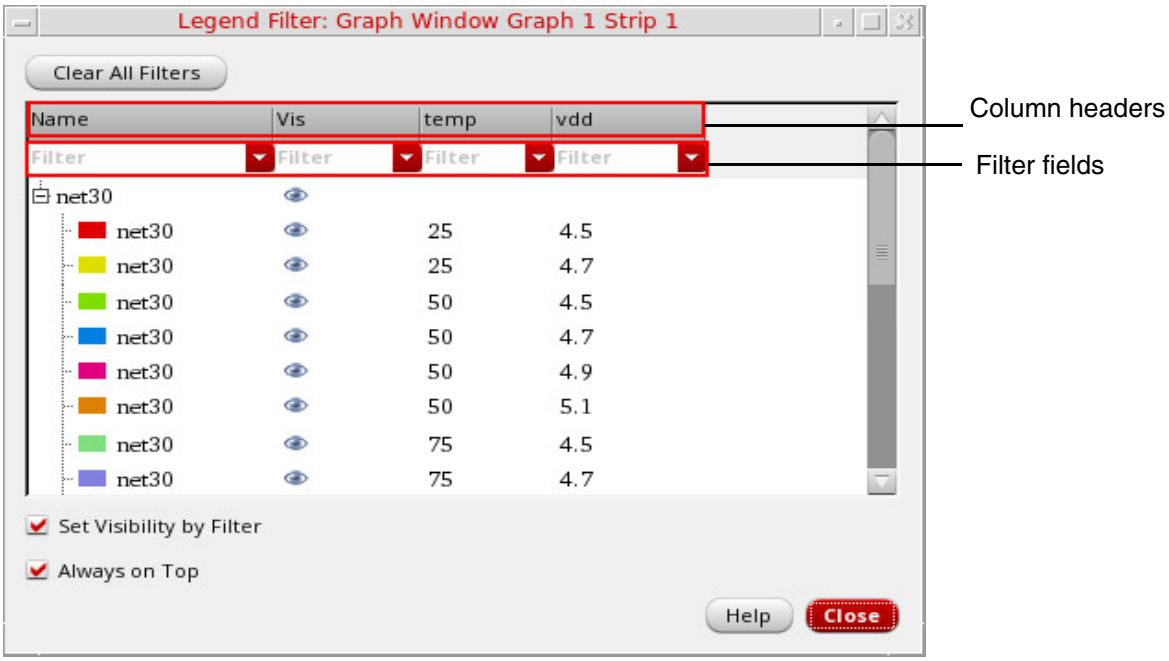
© 2023 Cadence Design Systems, Inc.

The following table describes the various advanced filter options.

Advanced Filter Options	Description
Find With	Using this option, you can set the filter criteria to search: ■ All of the words specified in the filter field ■ Any of the words specified in the filter field ■ The exact phrase specified in the filter field ■ Results that do not contain the words specified in the filter field
Match Word	Using this option, you can set the filter criteria to match: ■ *Substring*: displays the column with the values containing the filter criteria ■ *Prefix*: displays the columns with values starting with the filter criteria ■ *Exactly*: displays the columns with values matching exactly with the filter criteria ■ *Suffix*: displays the columns with values ending with the filter criteria
Using Case	Using this option, you can set the filter criteria to: ■ *Insensitive Match*: filters the results regardless of the case specified in the filter field ■ *Sensitive Match*: filters the results matching the case specified in the filter field

© 2023 Cadence Design Systems, Inc.

2. Enter the filter criteria in the filter field below the required column header.

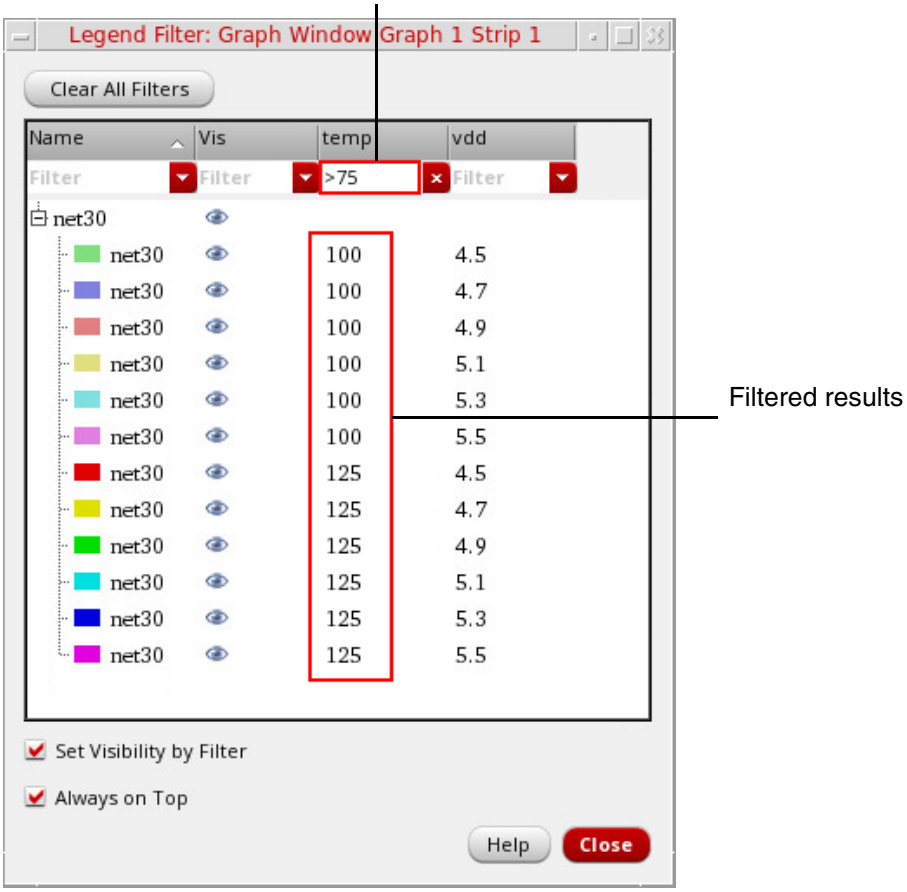


© 2023 Cadence Design Systems, Inc.

The following examples show how you can set the filter criteria:

- Filter values that are greater than the specified value

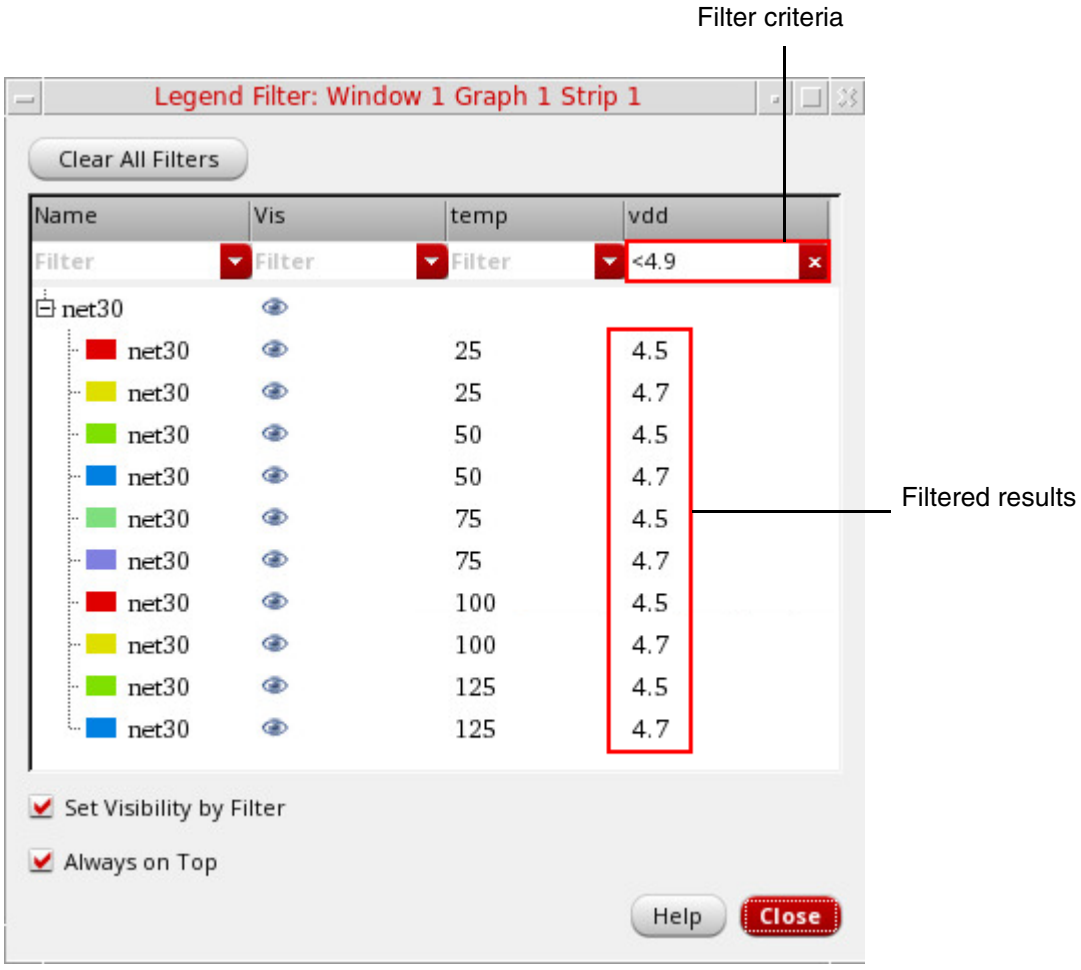
  If you want to view only the traces for which temperature value is greater than `75`, type `>75` in the filter field below the *temp* column, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

- Filter values that are less than the specified value:

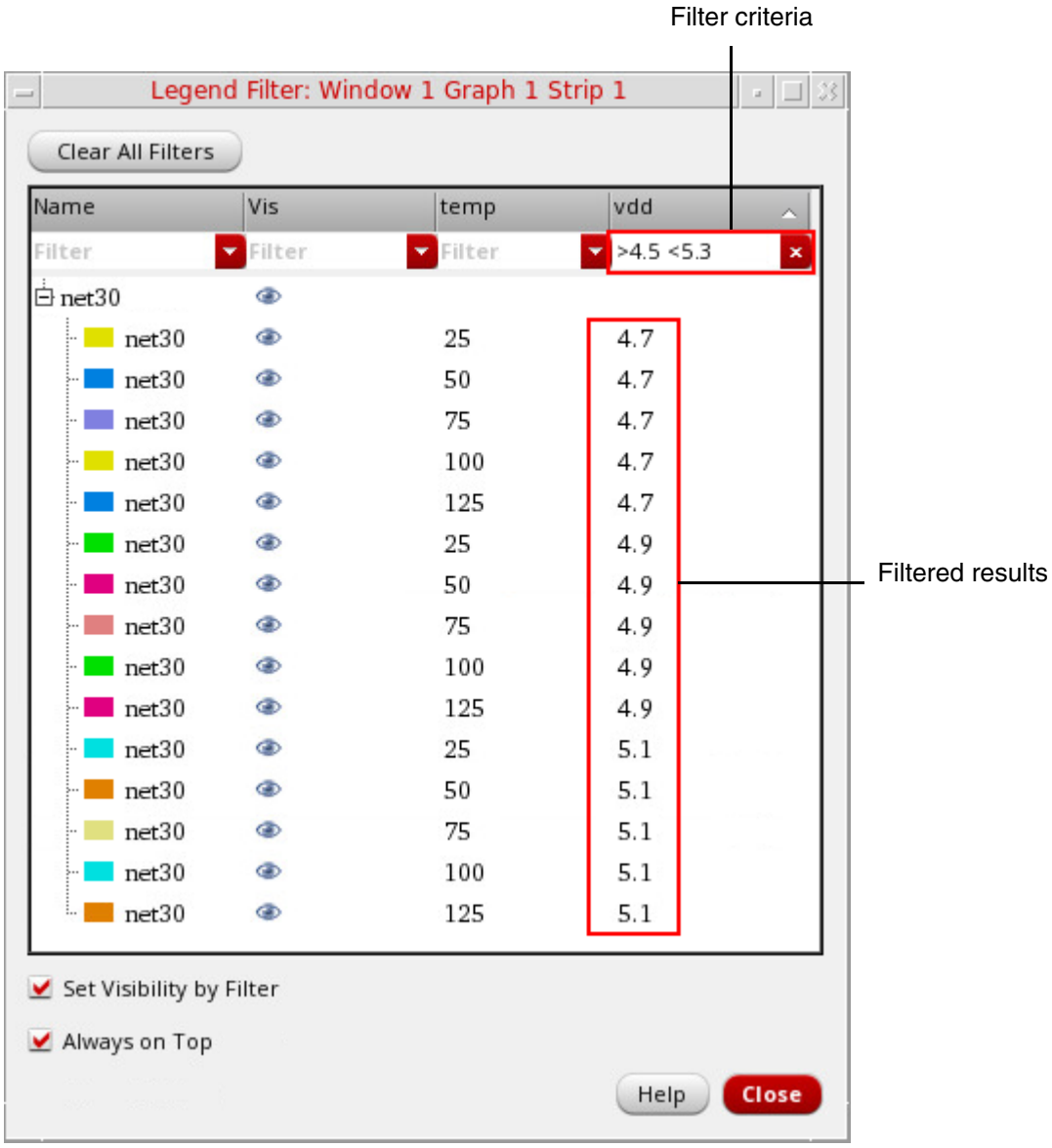
  If you want to view only the traces for which vdd value is less than `4.9`, type `<4.9` in the filter field below the *vdd* column, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

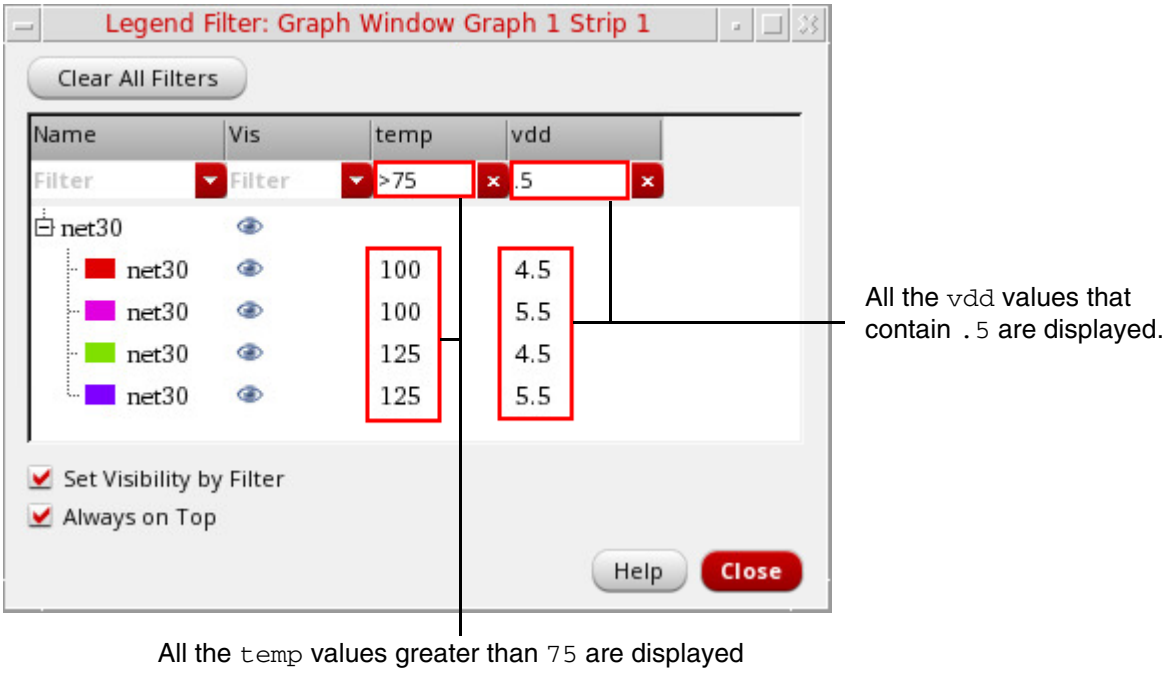
- Filter values that are in the specified range:

  If you want to view only the traces for which *vdd* value is greater than `4.5` and less than `5.3`, type `>4.5 <5.3` in the filter field below the *vdd* column, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

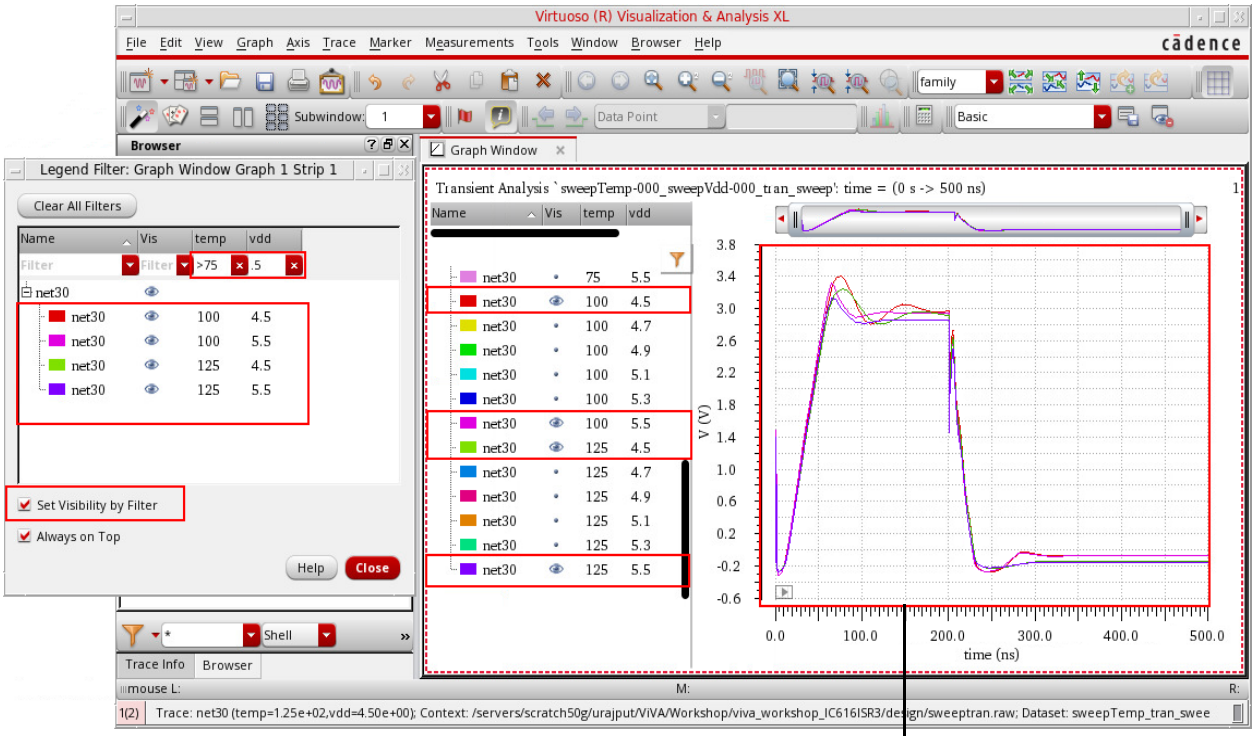
3. You can apply multiple filters in the *Legend Filter* form. The following figure shows an example where more than one filters are applied to the traces.



4. Select the *Set Visibility by Filter* check box if you want to view only those traces in the graph that meet the filtering criteria. The following figure shows an example of how the

© 2023 Cadence Design Systems, Inc.

visibility of the traces in the graph is automatically set based on the applied filters when you select the *Set Visibility by Filter* check box.

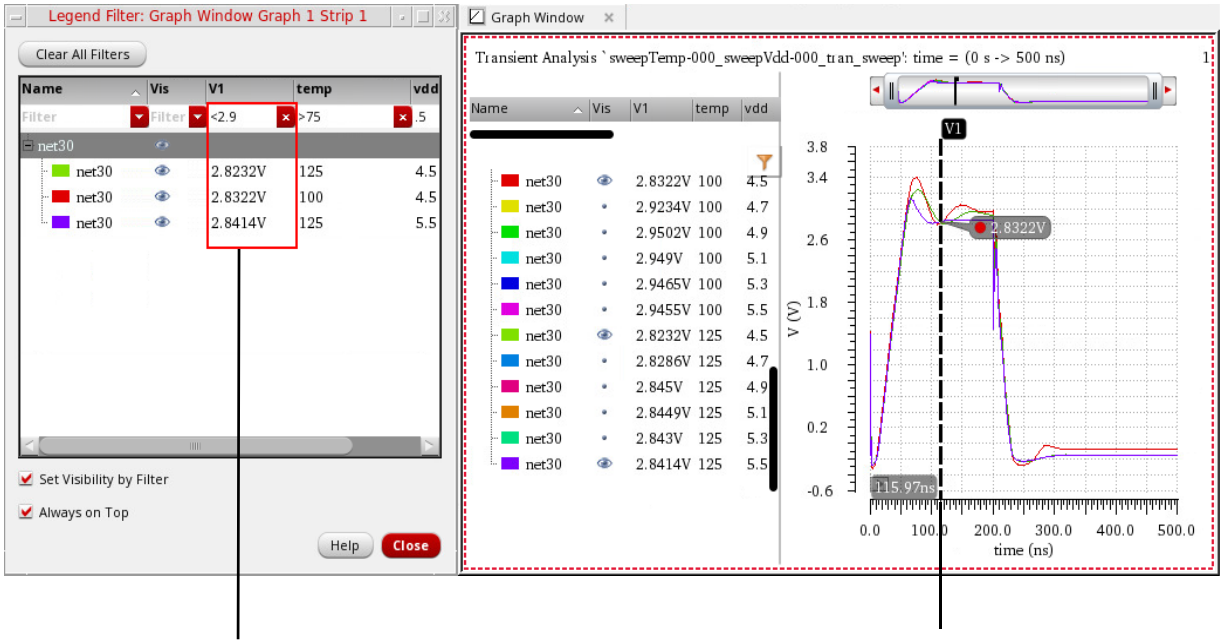


Only the filtered traces are visible in the graph window if the *Set Visibility by Filte*r check box is selected.

© 2023 Cadence Design Systems, Inc.

Important Points to Note

- You can also set vertical marker values as the filtering criteria if a vertical marker is attached to a trace. The following figure shows an example where a vertical marker, `V1`, is attached to the trace and a filter, `<2.9`, is applied.



Filter *<2.9* is applied to the trace for the vertical marker, '*V1*'

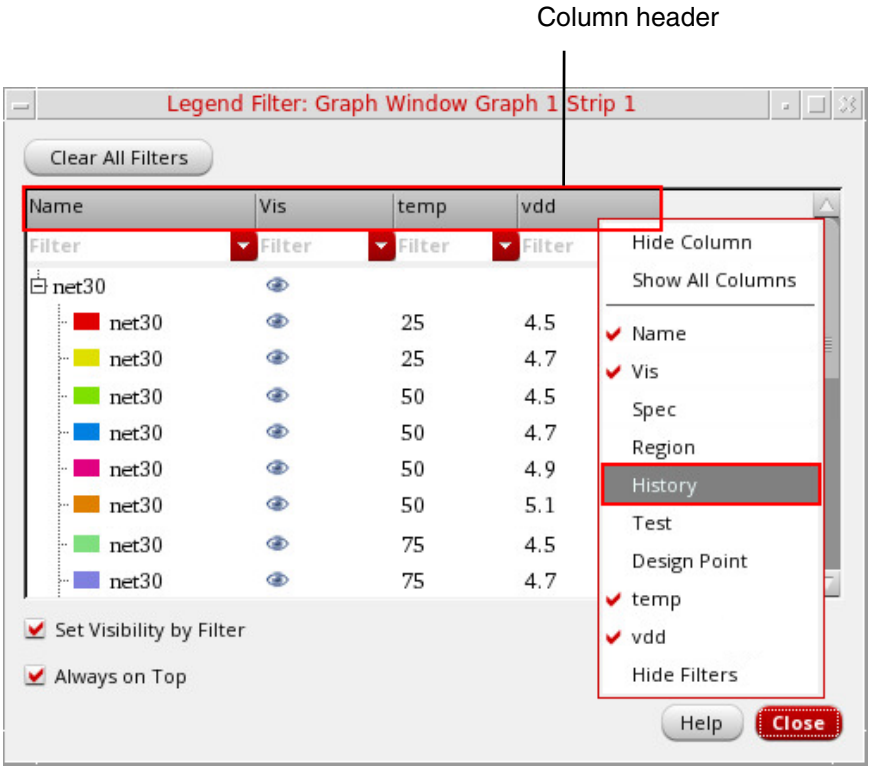
Vertical marker '*V1*'

- Click the column headers to sort the values in the ascending or descending order.
- Click *Clear All Filters* to clear all the applied filters.
- Select the *Always on Top* check box to keep the *Legend Filter* form on top of all the Virtuoso windows regardless of which window you have currently selected. You can resize and move the *Legend Filter* form to keep it side-by side of the Virtuoso Visualization & Analysis XL window for better analysis and visibility of the graphs.
- You can also use the trace menu commands from the *Legend Filter* form. To use the trace menu commands, select a trace and then right-click to choose the command that you want to use.
- When you toggle the visibility of a trace by clicking the visibility icon, , the *Set Visibility by Filter* check box is deselected.
- You can filter legends on both rectangular and circular graphs.

© 2023 Cadence Design Systems, Inc.

Viewing Information about the Traces

➡ To view information about the traces plotted in the graph, right-click anywhere in the column header, and then choose the required column. For example, if you want to view history name, choose *History*, as shown in the following figure.



- ❑ Choose *Show All Columns* to view information about all the columns.
- ❑ Choose *Hide Column* to hide the information about the selected column.
- ❑ Choose *Hide Filters* to hide the filter fields displayed below the column headers.

Some Useful Features

- ■ Each trace in the trace legend area displays a Visibility button that you can use to show or hide traces.
- ■ You can resize the trace legend area by dragging the dynamic splitter on the right. You can also adjust the width of each column in the trace legend area.

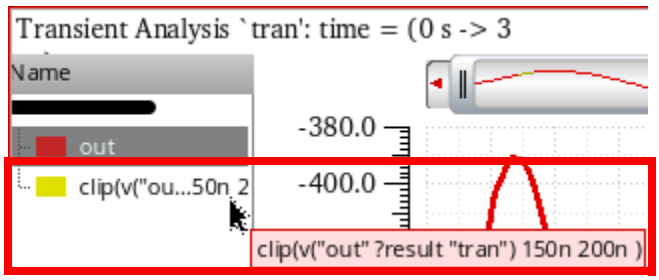
  By default, the trace legend is displayed on the left of the graph. You can also move the trace legend inside the graph or to the top of each strip by using the *Graph Properties* form. For more information, see Editing Graph Properties on page 196.

© 2023 Cadence Design Systems, Inc.

Following are the situations in which the legend positions—left, top, or inside—are used:

- The trace legend on the left is suitable for the plots containing sweeps, corners, and families. When you plot only a few signals per strip, in some cases, you may notice that keeping trace legend at left results in the wastage of a lot of graph space.
- The legend on the top is suitable in cases when a few signals are plotted per strip and not suitable for family plots.
- The inside legend takes the minimum space, thereby, giving the maximum space to original plots. The inside legend does not show additional information about the plots, such as sweep, corner. In addition, unlike other legend positions, the inside position does not support cross-selection of signals.

- If a trace name is long, it is elided from the middle to show the starting and ending text of the trace name, as shown in the figure below. The tooltip displays the full trace name.

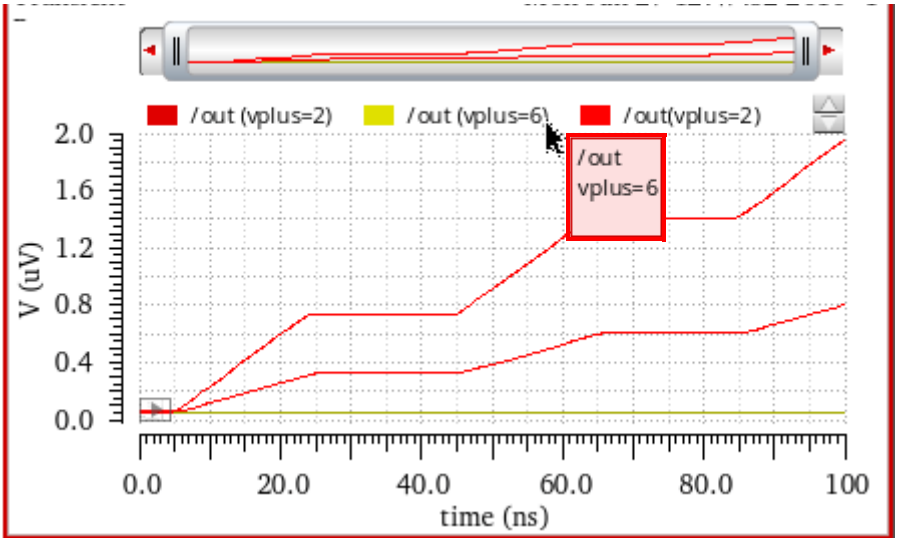


- When you select multiple traces in trace legend or select all the traces by using the *Trace – Select All – In Graph* menu command, then only the first group, which has some traces selected in the trace legend, is expanded and the other groups are displayed in the non-expanded form for performance reasons. Also note, after the selection is made, all the traces remain selected irrespective of the expanded and non-expanded groups. You can manually expand the groups to view the selected traces.
- When you plot traces belonging to a sweep data, do the following to show or hide sweep variables in the trace legend:
  - Right-click the trace legend header and choose the sweep variable you want to hide or show.

    A red tick mark appears with the selected sweep variable, which indicates that the variable is displayed in the trace legend.
- When you work with live simulation results where traces are generated through real-time tuning (RTT) or the setup has embedded variables, the trace legend displays the parameter values along with the trace name.The tooltip on the trace name in legend shows each parameter name and its value in a new line.

© 2023 Cadence Design Systems, Inc.

The trace legend for the waveform info-balloon on schematic also works in the similar way. When the legend is positioned at left or top, the tooltips on the trace names displays the trace name and the parameter names and values as shown in the figure below.



Displaying History, Test, and Design Point Information in Graphs

When you plot waveform results for a particular history in Virtuoso Visualization and Analysis XL, the related history, test, and design point information for the plotted waveform is displayed at the following locations:

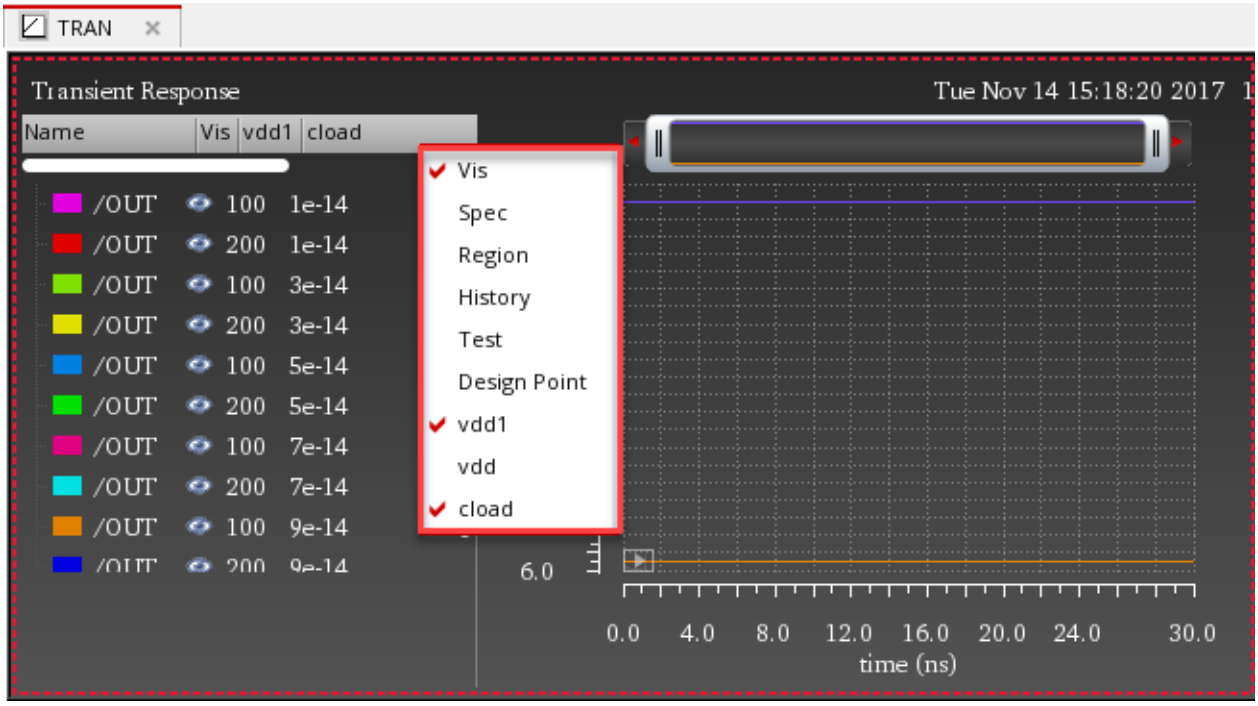
- Trace Legend
- Trace Info Assistant
- Status Bar
- Markers

Trace Legend

The trace legend of the graph includes columns to show the *History*, *Test*, and *Design Point* information. By default, these columns are not displayed in the trace legend. To display them,

© 2023 Cadence Design Systems, Inc.

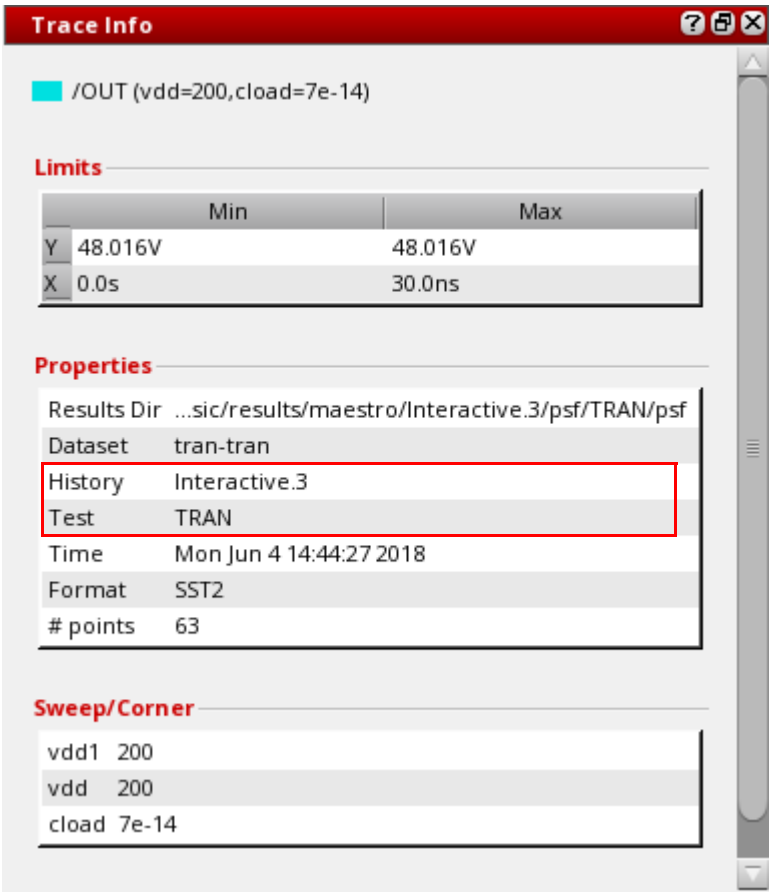
right-click anywhere in the column header and choose the required column, as shown in the figure below.



© 2023 Cadence Design Systems, Inc.

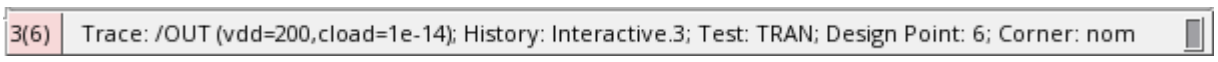
Trace Info Assistant

The Trace Info assistant displays the *History* and *Test* information about the trace that is currently selected in the graph window, as shown in the figure below:



Status Bar

When you move the mouse pointer over a trace in the graph window, the status bar displays its history, test, and design point information, as shown in the figure below:



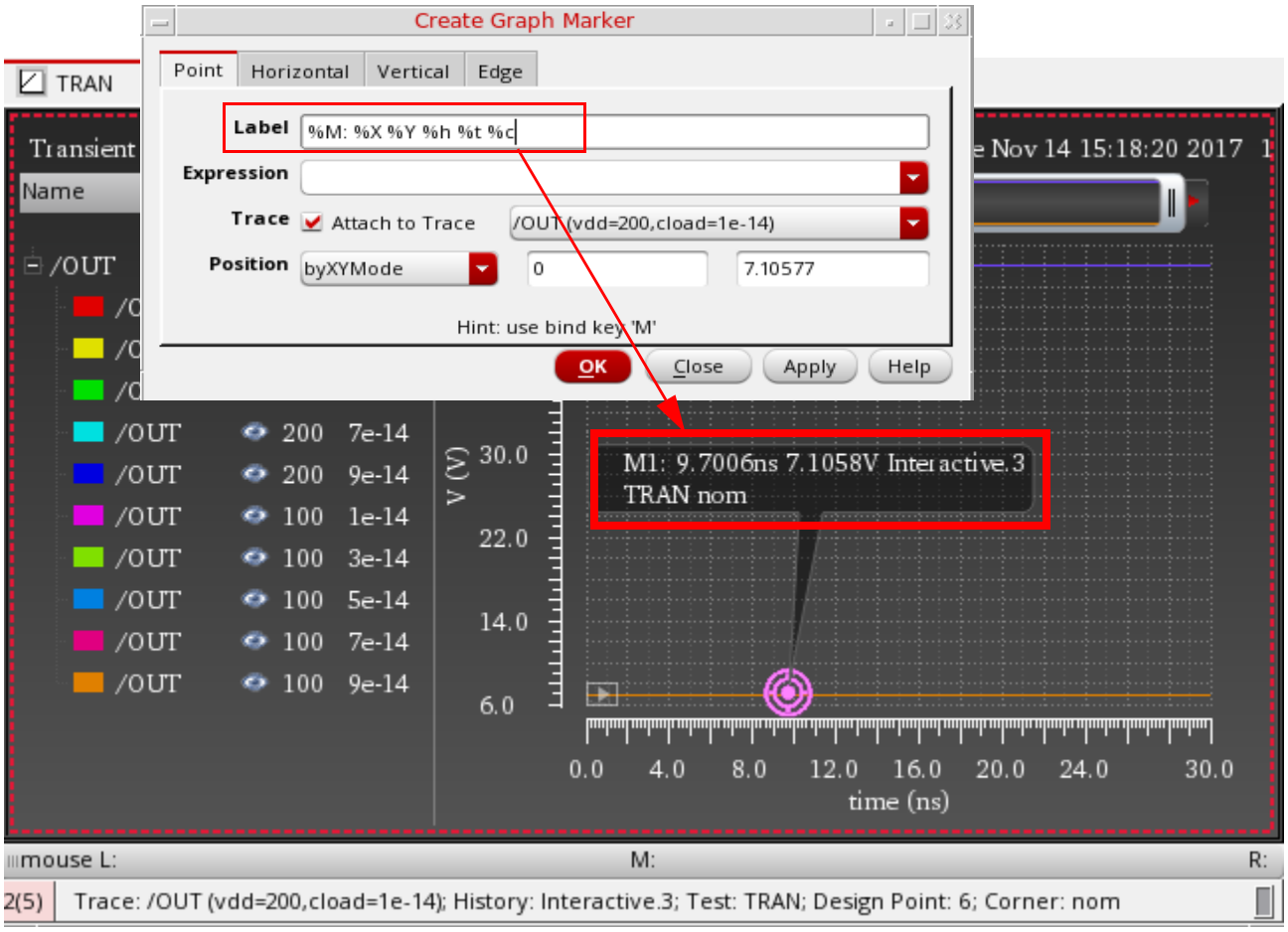
Markers

You can add the following marker labels to show information about history, test, and design point using the Create Graph Marker form.

© 2023 Cadence Design Systems, Inc.

- `%t`—Test Name
- `%h`—History
- `%c`—Corner Name

The following figure shows the marker labels added on the point marker `M1`.



Note: You can also read the Virtuosity: Identifying Those Traces blog for more information about these marker labels.

Selecting Traces

You can select one or more traces in a graph by doing the following:

- To select a trace, click the trace in the graph or click the trace name in the trace legend area.

© 2023 Cadence Design Systems, Inc.

- To select all traces in a graph or a subwindow, press `Ctrl+Shift+A`.
- To select all traces in a strip, press `Ctrl+A`.

Note: The active strip is determined by a yellow bar displayed to the left of the trace legend area.

Note: When you select multiple traces in the trace legend or select all the traces by using the *Trace - Select All - In Graph* menu command, then only the topmost group in the trace legend is expanded and the other groups are displayed in the non-expanded form. Also note, after the selection is made, all the traces remain selected irrespective of the expanded and non-expanded groups.

Using Trace Menu Commands

You can select a trace and choose one of the following *Trace* menu commands to manipulate traces in a window:

- *Symbols On*—Displays the symbols on the individual data points for the selected trace.
- *Select By Family*—Selects the traces from the parametric sweeps by family, instead of selecting an individual leaf.
- *Strip By Family*—Adds all the traces belonging to a family to a strip.
- *Select All*—Selects all the traces in a graph or in a strip in the active window or subwindow.
- *Delete All*—Deletes all the traces in the active window or subwindow. You can also use bindkey `Ctrl+E` to delete all the traces in a window.
- *Move to*—Moves the selected trace to the following locations:
  - *New Window*—Moves the selected trace in a graph to a new window.
  - *New Subwindow*—Moves the selected trace in a graph to a new subwindow.
  - *New Strip*—Moves the selected trace in a graph to a new strip.
- *Copy to*—Copies the selected trace to the following locations:
  - *New Window*—Copies the selected trace to a new window.
  - *New Subwindow*—Copies the selected trace in a graph to a new subwindow.
  - *New Strip*—Copies the selected trace in a graph to a new graph strip. This is especially useful when you want to study a single trace separate from a set of

© 2023 Cadence Design Systems, Inc.

parametric leaf waveforms. You can alter trace selection to select signals by family, rather than by selecting individual traces, by choosing *Trace - Select by Family*.

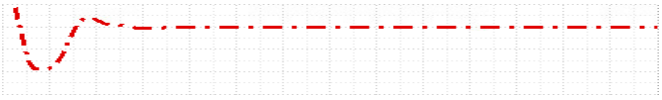
To show or hide traces, right-click the trace and choose *Visible*. The red check mark is displayed with this option indicating that the trace is visible.

Note: When you move a trace to a new window, the moved trace is plotted with the same x-axis scale in the new window.

Setting the Style of the Traces

To set the style of a trace, right-click anywhere on a trace to open the trace menu commands, and then choose *Style* – `options`.

The following table describes the different styles that you can apply to a trace.

Style Options	Example
Solid	
Dot	
Dash	
DashDot	
DashDotDot	

© 2023 Cadence Design Systems, Inc.

You can also change the styles of the trace by setting the following environment variable:

```
envSetVal("viva.trace" "lineStyle" 'string "solid")
```

Valid values are: `solid`, `dot`, `dash`, `dashdot`, `dashdotdot`

Setting the Thickness of the Traces

To set the width of a trace, right-click anywhere on a trace to open the trace menu commands, and then choose *Width* – *`options`*.

The following table describes the different width that you can apply to a trace.

Width Options	Example
Thin	
Medium	
Thick	
ExtraThick	

You can also change the thickness of the trace by setting the following environment variable:

```
envSetVal("viva.trace" "lineThickness" 'string "fine")
```

Valid values are: `fine`, `medium`, `thick`, `extrathick`

© 2023 Cadence Design Systems, Inc.

Setting the Color of the Traces

To set the color of a trace, right-click anywhere on a trace to open the trace menu commands, and then choose *Color – `options`*.

You can also change the color of the trace by setting the following environment variable:

```
envSetVal("viva.trace" "hiliteColor" 'string "lime")
```

Dragging Traces

Traces support the following drag-and-drop operations:

- You can drag traces from one window to another.
- You can drag traces from one subwindow to another. The window to which you drag a trace becomes the active window.
- You can drag traces from one strip to another.

Tracking Cursor

The tracking cursor displays the color and name of the trace and its X- and Y-axis values when you drag a trace. Do one of the following to display tracking information on the tracking cursor:

- Choose *Markers – Tracking Cursor*.
- Click the Tracking Cursor button on the Marker toolbar.

Hiding and Showing Traces

To show or hide a trace, do the following:

- Right-click a trace and select *Visible*.
- In the trace legend area, click the Visibility button ().

Deleting Traces

To delete a trace from a graph, subwindow, or strip, do one of the following:

- Select the trace and choose *Edit – Delete*, or press the `Delete` key.
- Right-click the trace and choose *Delete*.

© 2023 Cadence Design Systems, Inc.

To delete all traces in a subwindow, select a trace and do one the following:

- Choose *Trace – Delete All*
- Press `Shift+E`.

Setting Trace Properties

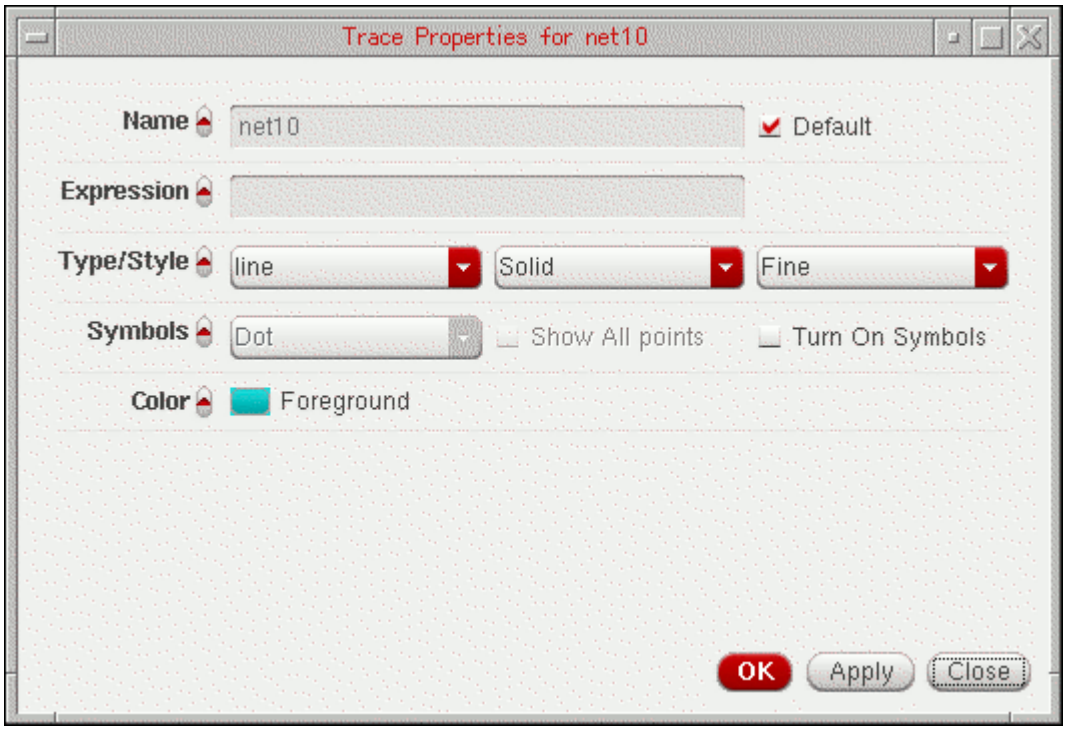
The default trace properties are controlled by the values assigned to variables in the `.cdsenv` file. For more information, see Appendix A, "Virtuoso Visualization and Analysis XL Tool Environment Variables."

Follow these steps to set the trace properties for a trace:

1. Select a trace and choose *Trace – Properties*.

   Alternatively, double-click a trace in the window.

   The *Trace Properties for <trace-name>* form appears.



2. In this form, set the trace properties:

   a. In the *Name* field, type the name for the trace or select the *Default* check box to display the default trace name. When you select the *Default* check box, the *Name*

© 2023 Cadence Design Systems, Inc.

field becomes unavailable. The *Name* and *Default* fields are not available if you select more than one trace.

b. In the *Expression* field*,* type the expression associated with the selected trace or get the expression from the Calculator Buffer.

c. In the *Type/Style* fields, do the following:

- Specify whether you want to represent the trace by a *line*, *points*, *histogram*, *bar*, *spectral,* or *sampleHold*.
- Specify whether you want the trace style to be *Solid*, *Dashed*, *Dotted*, or *DotDashed,* or *DashDotDot*.

Note: The trace style option does not work for *Bars* and *Spectrum*.

- Specify whether you want the trace to be *Fine*, *Medium*, *Thick* or *ExtraThick*.

Note: The trace thickness option does not work for *Bars* and *Spectrum*.

d. In the *Symbols* field, select the *Show All Points* check box to display data points on the trace and specify the number of points to be displayed.

e. In the *Points per Symbol* field, specify whether you want data points to be displayed on the trace as symbols of *Point*, *Dot*, *Plus*, *Square*, *Box*, *X*, *Circle,* and so on*.*

f. In the *Color* field, select the foreground color for the trace. Alternatively, you can also set the trace color by right-clicking the trace and selecting *Color.*

g. Click *OK*.

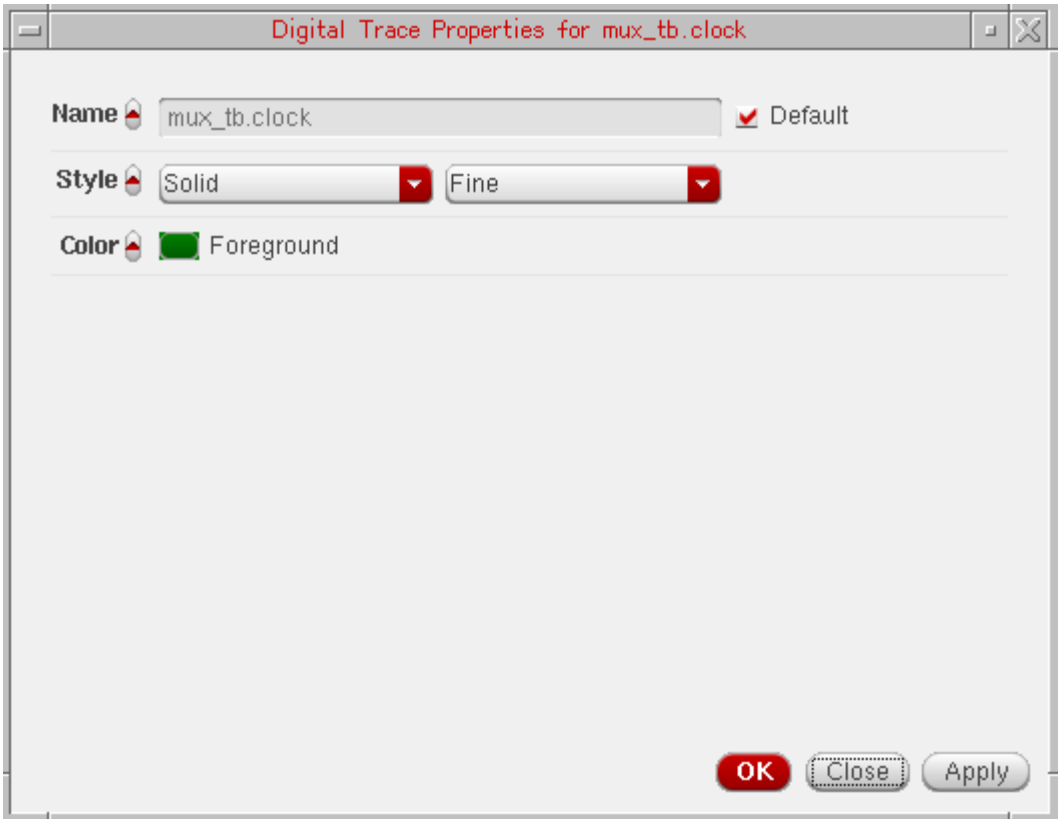
Note: You can also select these properties by right-clicking a trace.

For more information about the selected trace, choose *Window – Assistants – Trace Info Panel*. The Trace Info Panel assistant appears, displaying the *Name, Max value, Min value,* and *Data points* of the selected trace.

© 2023 Cadence Design Systems, Inc.

Setting Digital Trace Properties

To change the properties of a digital signal, right-click the digital signal and choose *Digital Trace Properties*.



The *Digital Trace Properties for <signal name>* form appears. This form includes the following fields:

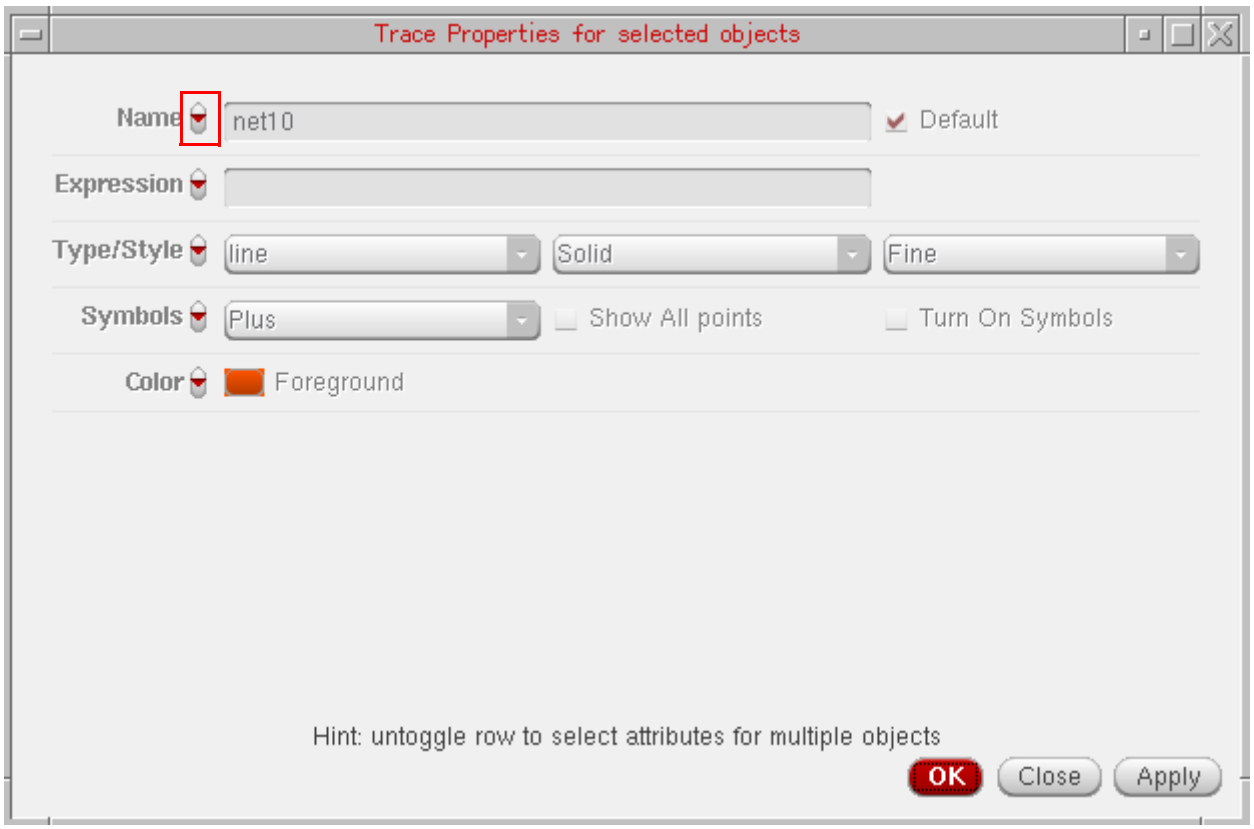
1. *Name*—Specify the name for the trace or select the *Default* check box to display the default trace name. When you select the *Default* check box, the *Name* field becomes unavailable. The *Name* and *Default* fields are not available if you select more than one trace.
2. *Style*—Specify whether you want the trace style to be *Solid*, *Dashed*, *Dotted*, or *DotDashed,* or *DashDotDot*. And, specify whether you want the trace width to be *Fine*, *Medium*, *Thick* or *ExtraThick*. Alternatively, you can set the trace style and width by right-clicking the trace and selecting *Style* and *Width* respectively.
3. *Color*—Select the foreground color for the trace. Alternatively, you can set the trace color by right-clicking the trace and selecting *Color.* By default, the trace is displayed in green.
4. Click *OK*.

© 2023 Cadence Design Systems, Inc.

Setting Properties for Multiple Traces

You can set the properties for more than one trace at a time. To perform this, select the traces by using the `Ctrl` key and choose *Trace – Properties*.

The *Trace Properties for selected objects* form appears.



You can change the form fields by clicking the arrow button adjacent to each field name (as highlighted in the figure above).

Note: You can also use this method to change the properties of markers of similar marker type. For example, select markers of similar marker type by using the `Ctrl` key, such as two or more vertical markers, and choose *Marker – Properties*. The *<Marker-name> Properties for selected objects* form appears. You can then use this form to change the properties for selected markers in a single step.

Displaying Symbols on a Trace

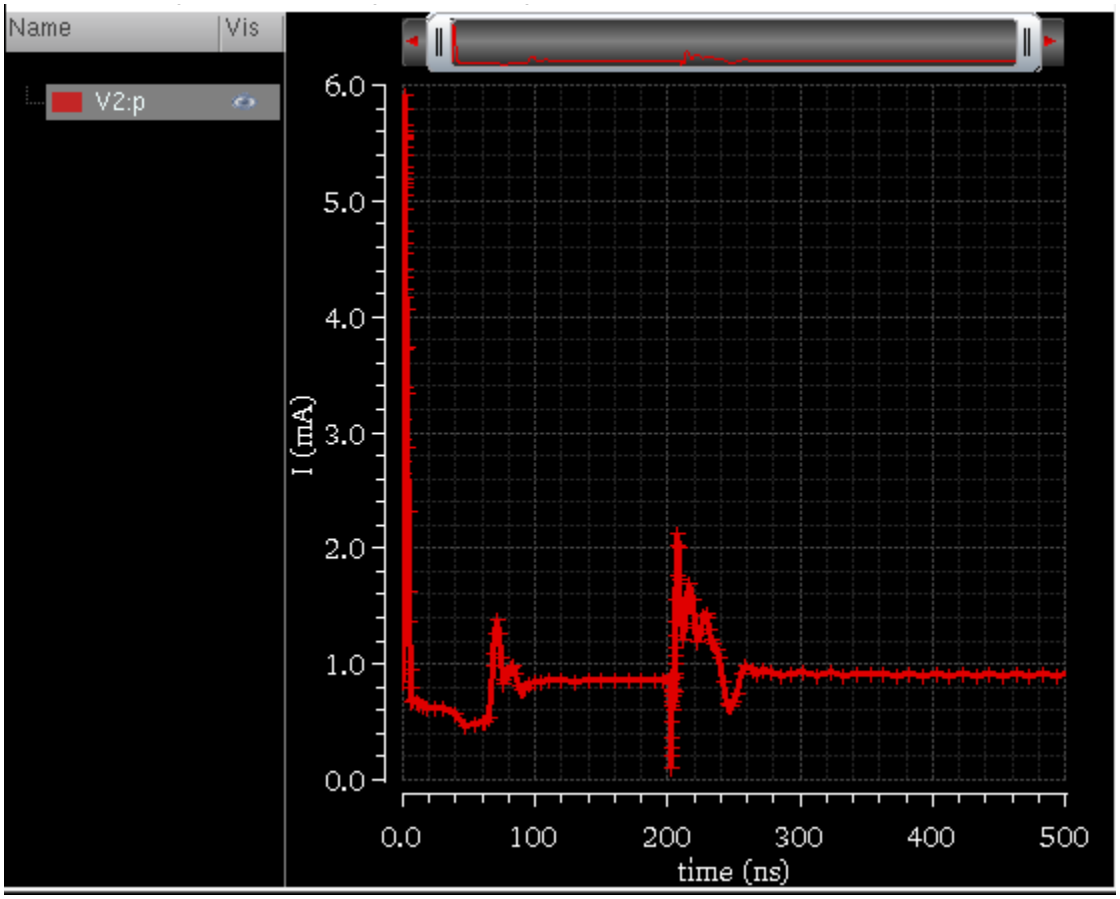
To display symbols for the data points on a trace, do one of the following:

➤ Select a trace and choose *Trace – Symbols On*.

© 2023 Cadence Design Systems, Inc.

You can control the symbol type and the number of data points that can be identified by the symbols. The symbol used for the trace is displayed next to the trace name.

The following figure illustrates how two traces can be distinguished by using symbols for the data points.



Exporting a Trace

To export a trace from the window in a variety of formats and later load it in the required application. You can also save a clipped part of the dataset by specifying the start and end values, or interpolate the data before saving it. By exporting a trace, you can also save the expressions associated with the trace.

To export the trace, do one of the following:

- Select the trace and choose *Trace — Export*.
- Right-click the trace and choose *Send To – Export*.

© 2023 Cadence Design Systems, Inc.

The *Export Waveforms* form appears. For detailed information about the fields in this form and how to save a trace using this form, see Exporting Waveforms from a Single Results Directory.

Sending Trace Expressions to Calculator

To send the expression associated with a trace in the window to the Calculator, select the trace for which you want to send the expression to the Calculator and do one of the following:

- Choose *Tools – Calculator*.
- Right-click the trace and choose *Send To – Calculator*.

  The Calculator window appears with the expression for the selected signal displayed in the Buffer.

  You can select more than one trace by holding down the `Shift` or `Ctrl` key and clicking the traces you want to select. The most recently selected trace appears in the Buffer and the remaining traces are added to the Stack with the recently selected trace at the top of the stack.

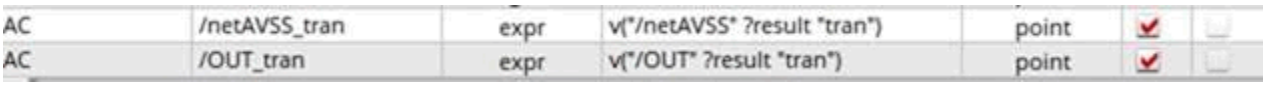
Sending Traces to ADE

To include the expressions for the traces displayed in a graph directly to ADE Outputs Setup tab, right-click a trace and choose *Send To – ADE*. The expression for the selected trace is added to ADE as a new output and evaluated when you run the simulation.

You use this option if you want to evaluate the expression for the trace in the current simulation run in ADE.

When you send the measurement values obtained from assistants to ADE, such as Eye Diagram and Spectrum toolbox, the alias name in the ADE displays the assistant name and the measurement name. For example, if you send the DC Power value from Spectrum toolbox to ADE, it is displayed with the `spectrum_dcpower` alias name. Similarly, the `Level0 Mean` value calculated from Eye Diagram is displayed with the `eye_level0Mean` alias name.

When you plot a signal from Results Browser, then right-click the plotted signal in the graph window and choose *Send To – ADE*, the expression for the plotted signal is added in the ADE Explorer or ADE Assembler Outputs Setup pane, as shown in the figure below:



© 2023 Cadence Design Systems, Inc.

Note: If you are using the Refresh feature in ADE Explorer or Assembler, the traces that you include into ADE Explorer or Assembler from the graph are also updated with the new simulation data.

Working with Strips

You can append multiple traces to a graph. If you want to view the individual traces, you can split the graph into strips that are arranged vertically. Each strip has its own Y-axis and shares the X-axis with the other strips. The window displays the trace legend separately for each individual strip.

The active strip is determined by a yellow bar displayed on the left of the strip. If you want to change the order in which the strips are displayed, drag the strips. You can drag a strip to any of the following locations:

- To a strip—the trace is appended to the strip.
- To an area outside the strip—the trace is placed in a new strip below the strip closest to the point to where you drag the strip.

You can also resize the active analog strip by dragging the strip splitter.

This section includes the following topics:

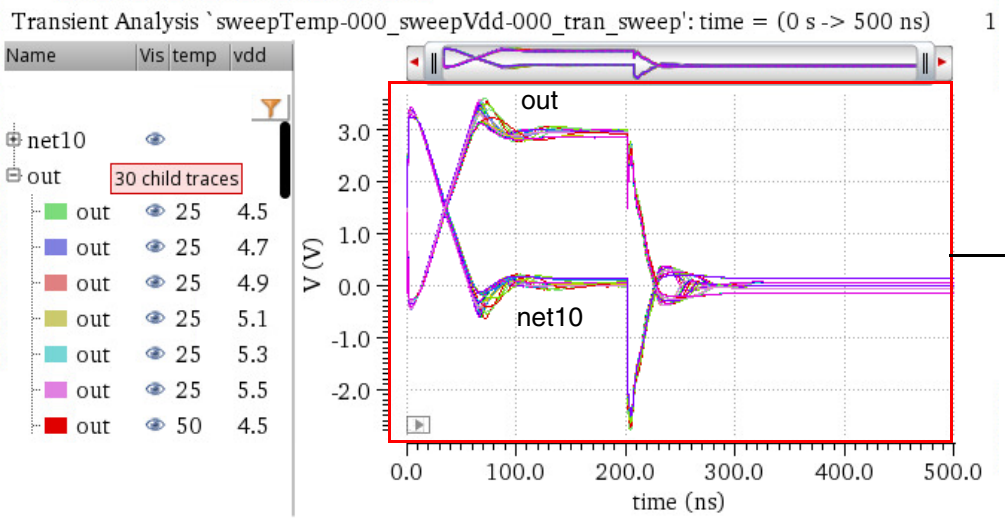
- Splitting Strips on page 398
- Combining Graph Strips on page 405
- Moving Traces on page 406
- Copying Traces on page 407

Splitting Strips

In the following figure, the signals `net10` and `out` are displayed in a single strip. Both the signals constitute a family of waveforms, containing 30 child traces grouped together. Each

© 2023 Cadence Design Systems, Inc.

of these child traces represents a waveform of the signal plotted for different set of `temp` and `vdd` values.



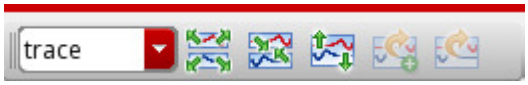
Note: `temp` and `vdd` are the sweep variables used in this example.

You can split the strip in any of the following ways:

- Split by *trace*

  To display all the traces individually in different strips, do one of the following:

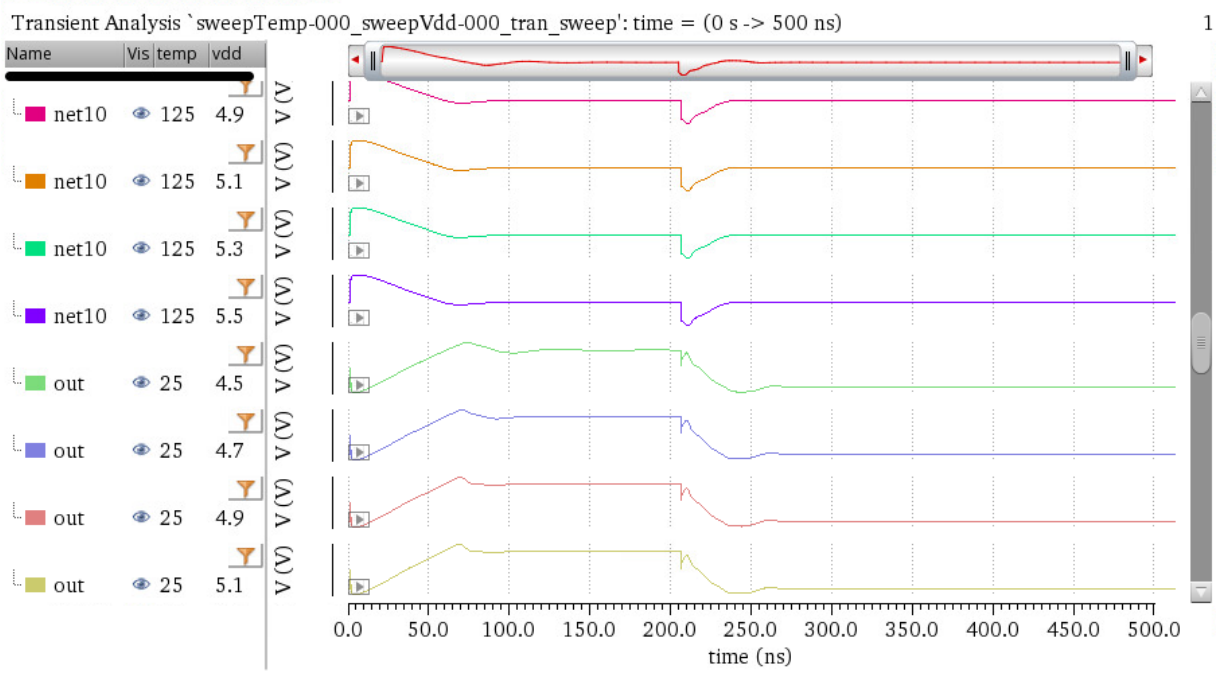
  - From the *Set Strip By Parameter* drop-down list, select *trace*, and then click either *Split All Strips* or *Split Current Strip*.

  - In the Customize Trace Groups assistant, from the *Strip* drop-down list, select *trace*, and then click *Apply*.

© 2023 Cadence Design Systems, Inc.

The strip is split to display all the traces individually in different strips:



- Split by *leaf*

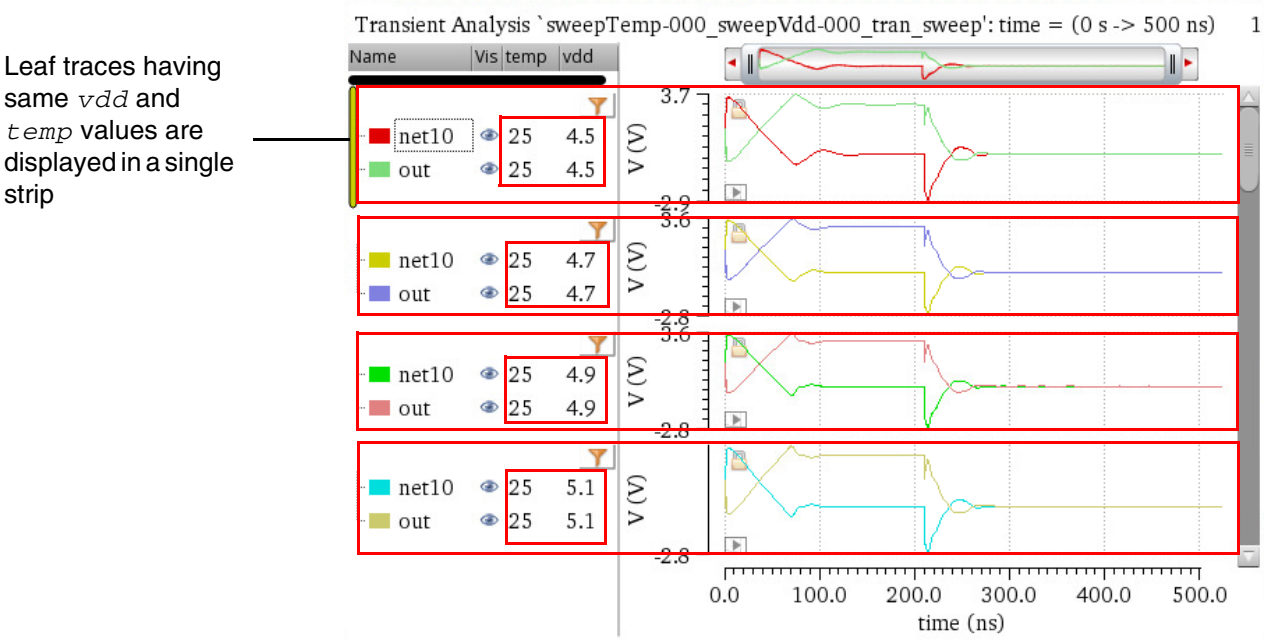
  To display the leaf traces that have the same set of `temp` and `vdd` values in a single strip, do one of the following:

  - From the *Set Strip By Parameter* drop-down list, select *leaf*, and then click either *Split All Strips* or *Split Current Strip*.

  - In the Customize Trace Groups assistant, from the *Strip* drop-down list, select *leaf*, and then click *Apply*.

© 2023 Cadence Design Systems, Inc.

The strip is split to display all the leaf traces in a single strip:



- Split by *family*

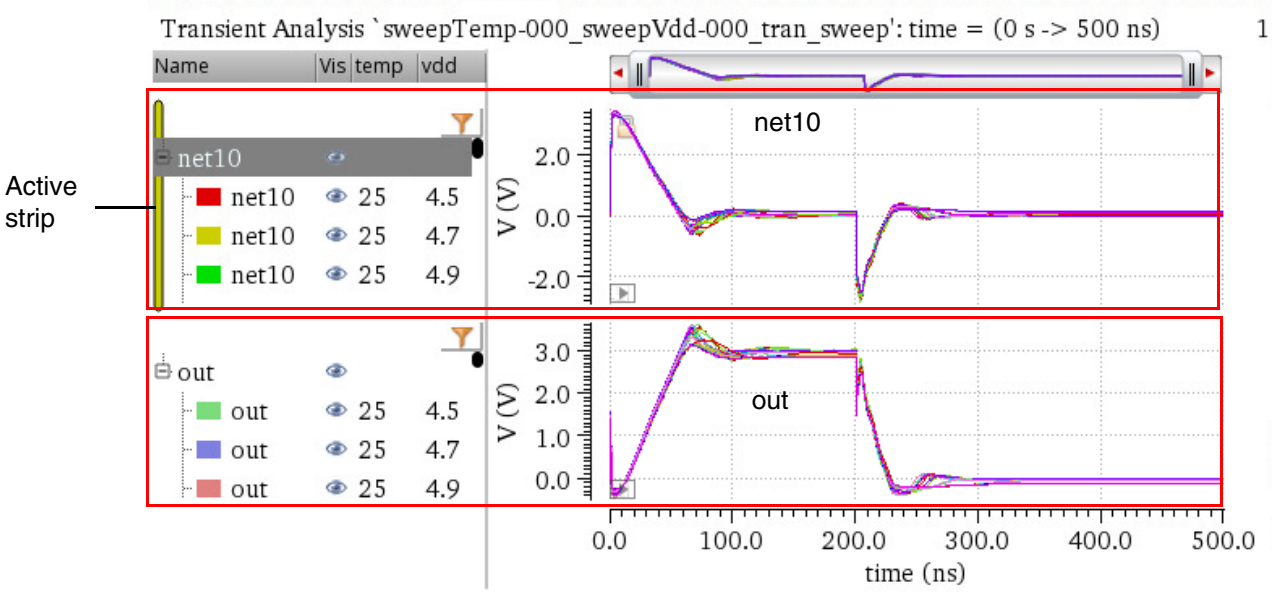
  To display the traces that belong to a common family in a single strip, do one of the following:

  - From the *Set Strip By Parameter* drop-down list, select *family*, and then click either *Split All Strips* or *Split Current Strip*.

  - In the Customize Trace Groups assistant, from the *Strip* drop-down list, select *family*, and then click *Apply*.

© 2023 Cadence Design Systems, Inc.

The strip is split to display the traces belonging to a common family in an individual strip:

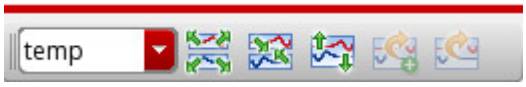


Note: The active strip is determined by a yellow bar displayed on the left of the strip. In this example, `net10` is the active strip.

- Split by *`temp`*

  To display the traces that have the same *`temp`* values in a single strip, do one of the following:

  - From the *Set Strip By Parameter* drop-down list, select *`temp`*, and then click either *Split All Strips* or *Split Current Strip*.

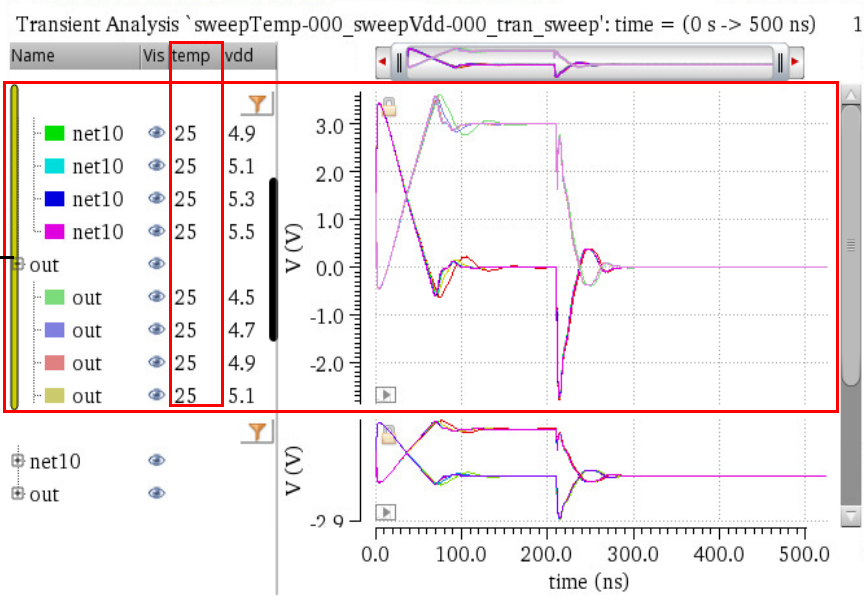
    

  - In the Customize Trace Groups assistant, from the *Strip* drop-down list, select *`temp`*, and then click *Apply*.

© 2023 Cadence Design Systems, Inc.

The strip is split to display all the traces that have the same temp values are displayed in a single strip.

All the traces having the same `temp` values are displayed in a single strip



Note: The `temp` and `vdd` options represent the sweep variables used in this example. You may see different options if you have other sweep variables.

- Split by `vdd`

  To display the traces that have the same `vdd` values in a single strip, do one of the following:

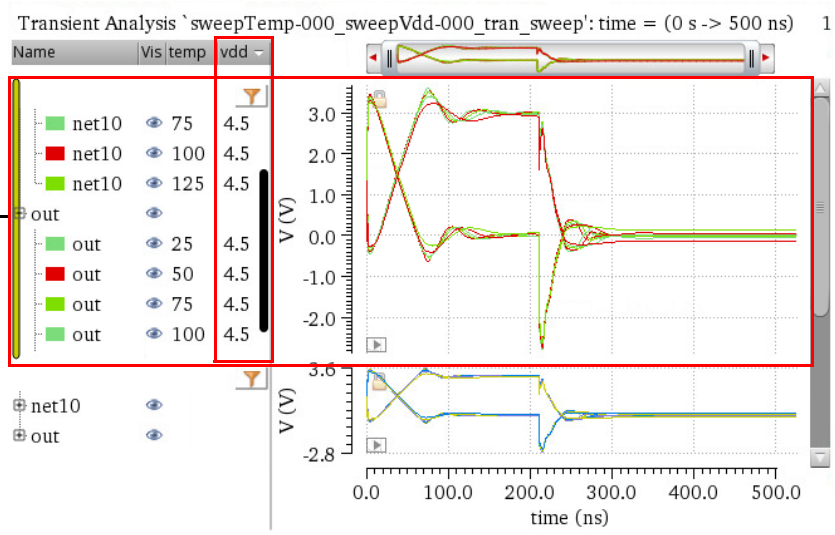
  - From the *Set Strip By Parameter* drop-down list, select `vdd`, and then click either *Split All Strips* or *Split Current Strip*.

  - In the Customize Trace Groups assistant, from the *Strip* drop-down list, select `vdd`, and then click *Apply*.

© 2023 Cadence Design Systems, Inc.

The strip is split to display all the traces that have the same *vdd* values are displayed in a single strip.

All the traces having the same *vdd* values are displayed in a single strip



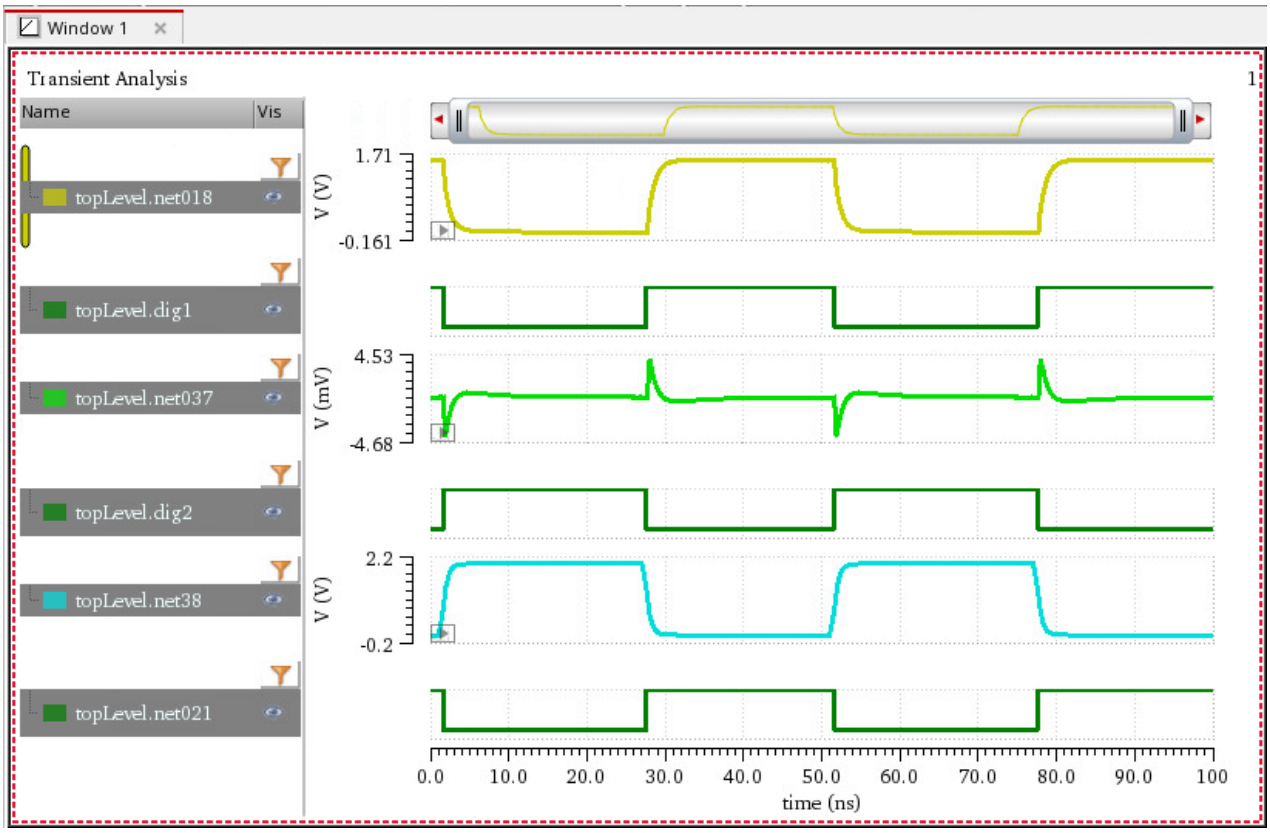
Note: The *temp* and *vdd* options represent the sweep variables used in this example. You may see different options if you have other sweep variables.

© 2023 Cadence Design Systems, Inc.

Combining Graph Strips

You can combine one or more strips by dragging them to a single strip.

Consider the following graph window, where various analog and digital traces are plotted in different strips.

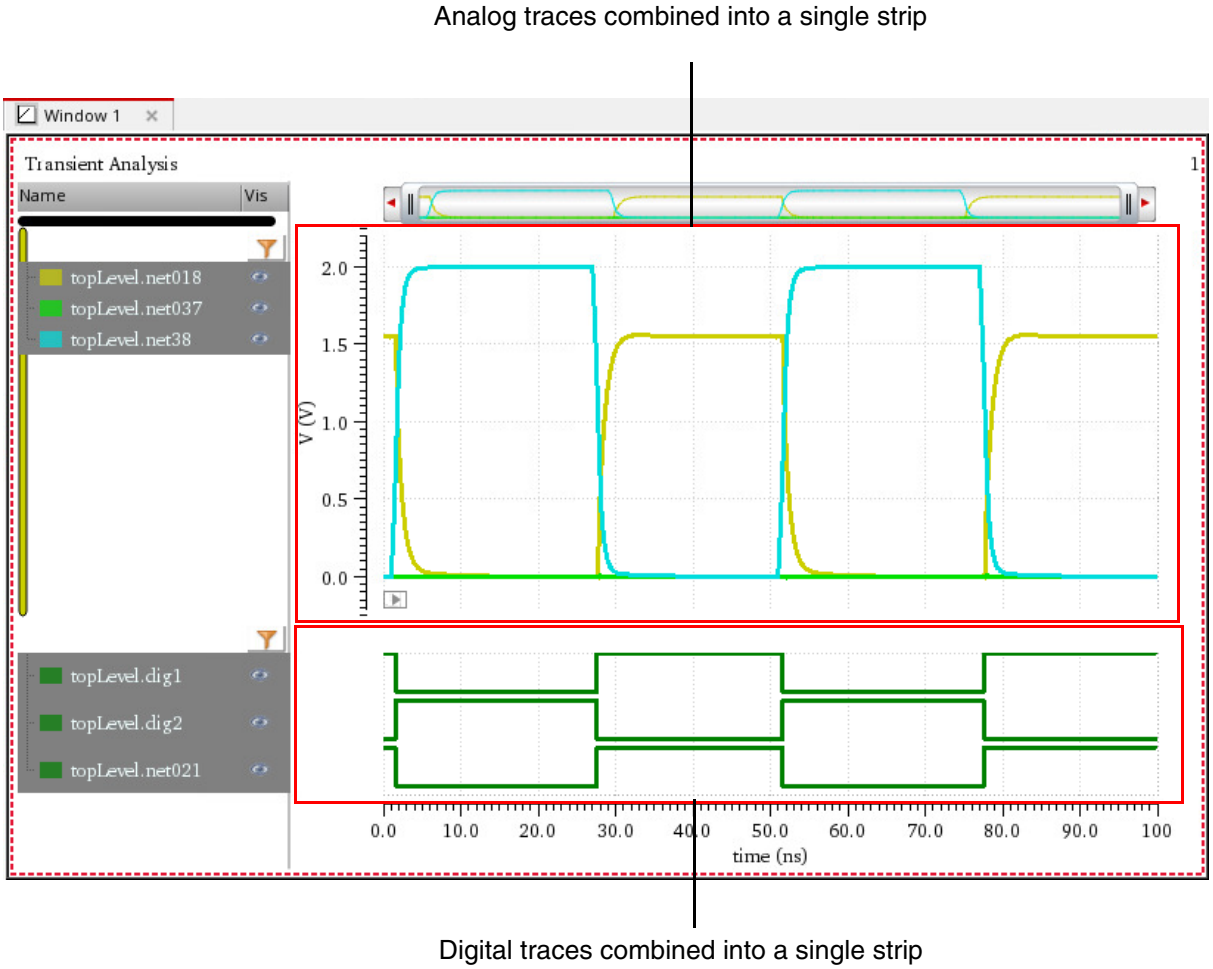


To combine digital signals into one strip and to combine analog signals into another strip, do one of the following:

- Choose *Graph – Combine All Analog Traces*.
- Right-click anywhere in the window and choose *Combine All Analog Traces*.
- Click the button on the Strip toolbar.

© 2023 Cadence Design Systems, Inc.

The analog traces displayed in the various strips are combined into a single strip. Similarly, all the digital traces are combined into another strip.



Moving Traces

Do one of the following to move the selected traces in a graph to a new window, subwindow, or strip:

- Select a trace and choose *Move to – Move Selected Traces to a New Window/Move Selected Traces to a New Subwindow/Move Selected Traces to a New Strip*.
- Select the traces that you want to move. Right-click the selection and choose *Move to – New Window/New Subwindow/New Strip*.
- Select the traces and click the button on the Strip toolbar.

© 2023 Cadence Design Systems, Inc.

Copying Traces

Do one of the following to copy the selected traces in a graph to a new window, subwindow, or strip:

- Select a trace and choose *Copy to – Move Selected Traces to a New Window/Move Selected Traces to a New Subwindow/Move Selected Traces to a New Strip*.
- Select the traces that you want to move. Right-click the selection and choose *Copy to – New Window/New Subwindow/New Strip*
- Select the traces and click the button on the Strip toolbar.

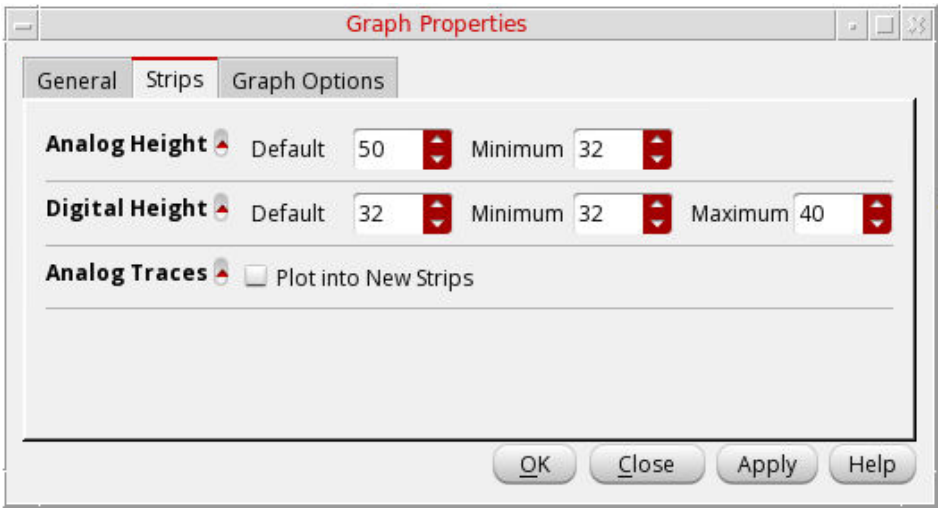
Setting Strip Properties

You can set the heights of strips by using the *Strips* tab on the Graph Properties form.

To set the heights of the strips:

1. Do one of the following to open the Graph Properties form:
   - Choose *Graph – Properties.*
   - Right-click anywhere in the graph window and choose *Graph Properties*.

   The *Graph Properties* form appears.

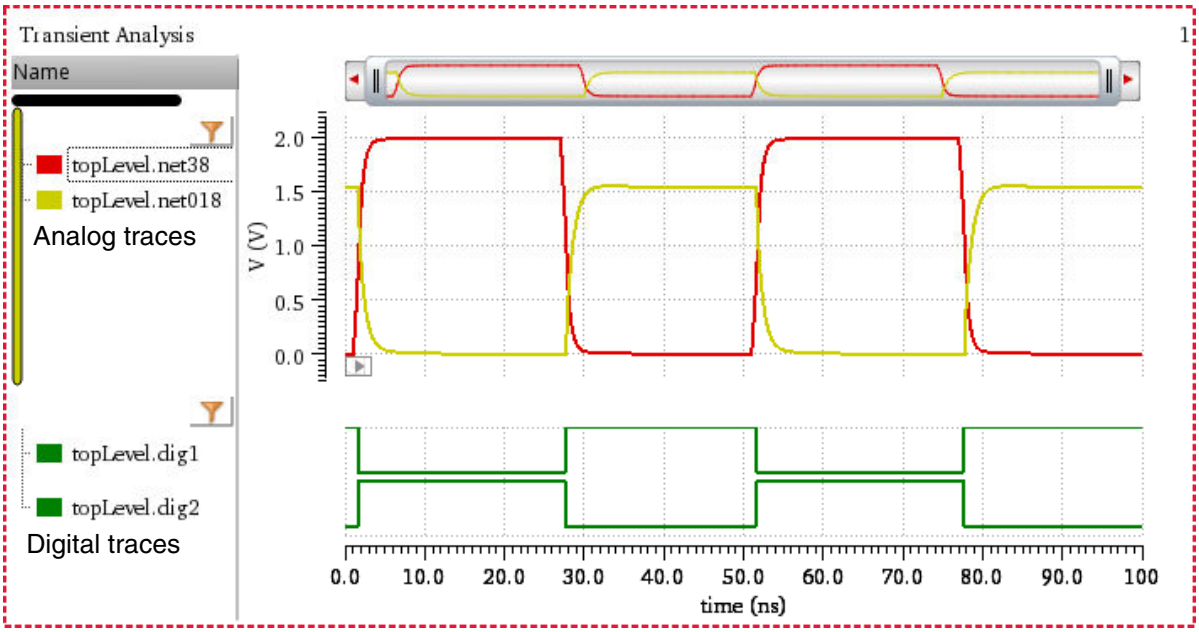


2. On the *Strips* tab, set the *Minimum* values of *Analog Height* and *Digital Height* of the strips of analog and digital signals, respectively.
3. Click *Plot into New Strips* check box if you want to plot analog traces into new strips.

© 2023 Cadence Design Systems, Inc.

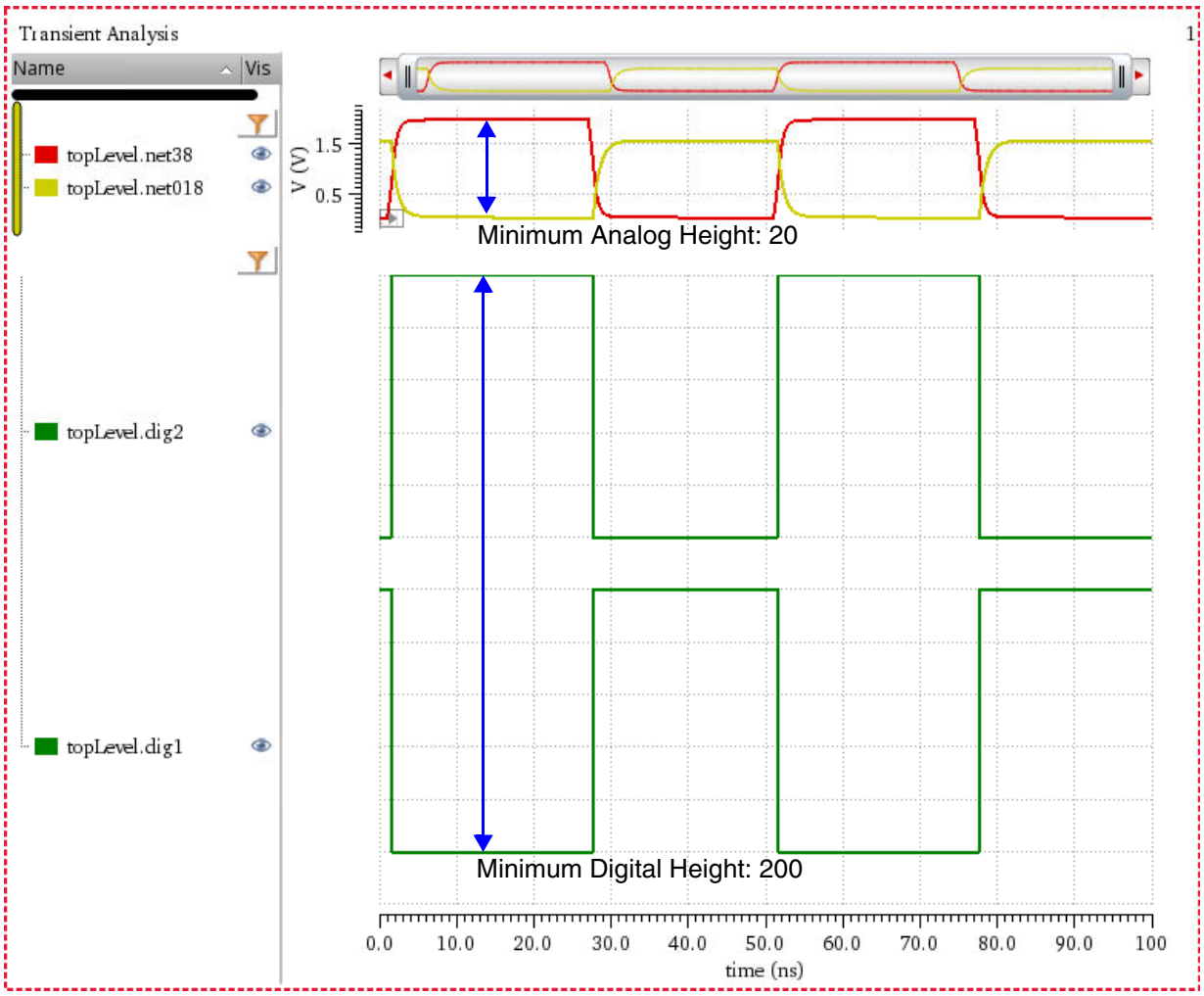
4. Click *Apply,* and then click *OK*.

   Consider the following graph window in which analog signals (`net38` and `net018`) and digital signals (`dig1` and `dig2`) are plotted in strips with their default heights.



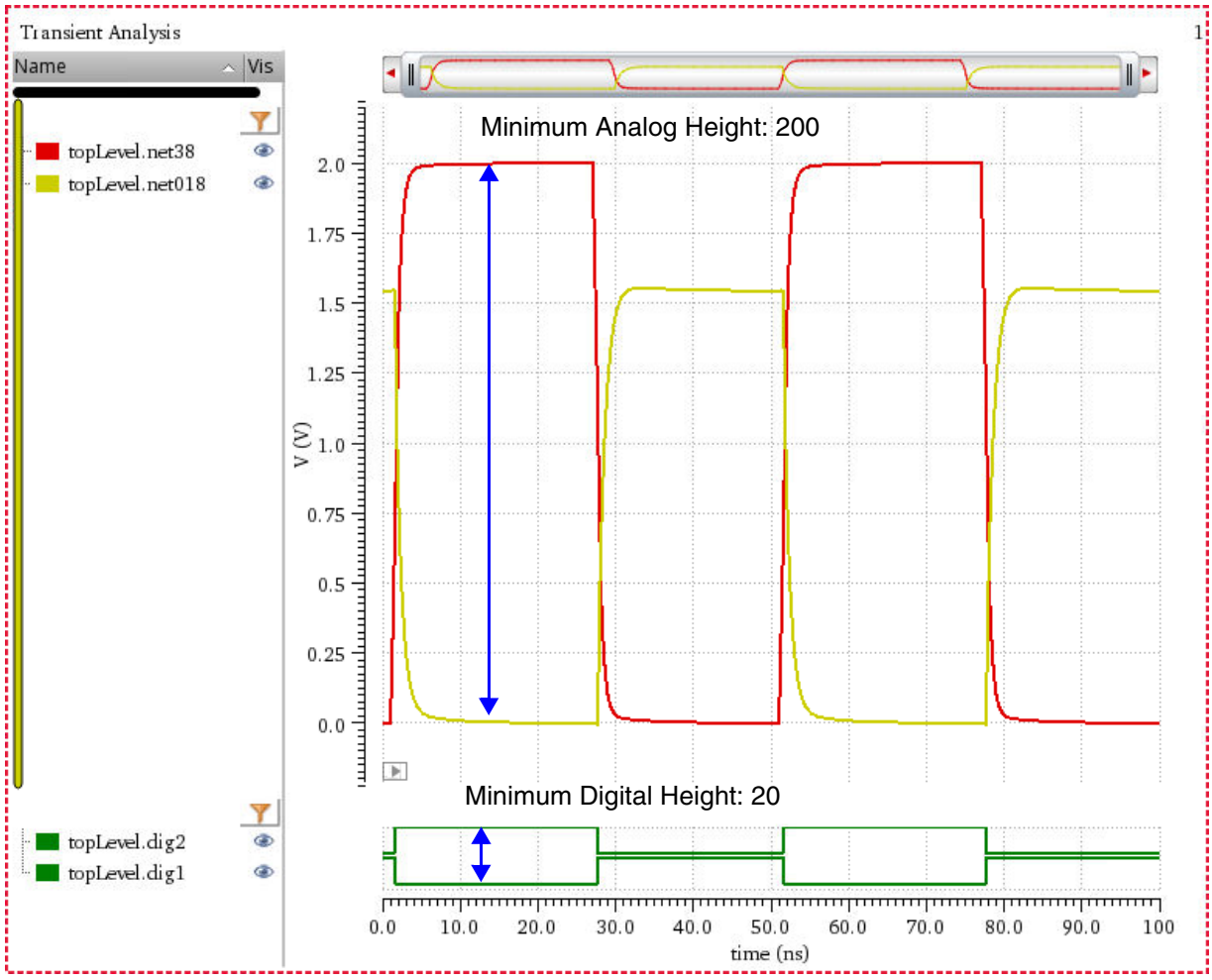
© 2023 Cadence Design Systems, Inc.

The following figure shows how the strip heights of the analog and digital signals change when you set the *Minimum* value of *Analog Height* to `20` and *Minimum* value of *Digital Height* to `200`.



© 2023 Cadence Design Systems, Inc.

The following figure shows how the strip heights of the analog and digital signals change when you set the *Minimum* value of *Analog Height* to `200` and *Minimum* value of *Digital Height* to `20`.



Locking and Unlocking Strips

You can lock and unlock a strip while splitting the traces into different strips.

To lock a strip, right-click anywhere in the window and choose *Lock Strip Size*. A red check mark is displayed before this command and a lock icon appears at the top right corner of the strip, displaying the strip in the locked mode. If you change the size of any other strip in the window, it does not change the size of the locked strip.

© 2023 Cadence Design Systems, Inc.

To unlock the strip, right-click anywhere in the window and choose *Lock Strip Size*. The lock icon is no longer displayed and you can resize the strip now.

Video

The Using Strip Charts in Qt Graph video demonstrates how to split and combine analog traces into strips, resize strips, plot the trace to a new strip, and split and combine traces from sweep data.

Working with Sweeps

To display the sweep data for a family in the same strip, choose *Trace – Strip by Family* before plotting the sweep data. Now, if you select to display traces in strips, each trace family is displayed in a separate strip.The trace legend area displays the traces in the family. Click the + sign to view all traces.

- To select an individual trace in the family, click the trace in the trace legend area.
- To select all the traces in the family, choose *Trace – Select By Family*. Now, when you select a signal in the trace legend area, the entire family is selected. Alternatively, you can select all signals in the family by using the `Ctrl` key.

To display the traces from the sweep data in individual strips, ensure that the *Trace – Strip By Family* is not selected. Then, right-click anywhere in the graph and choose *Split Current Strip*.

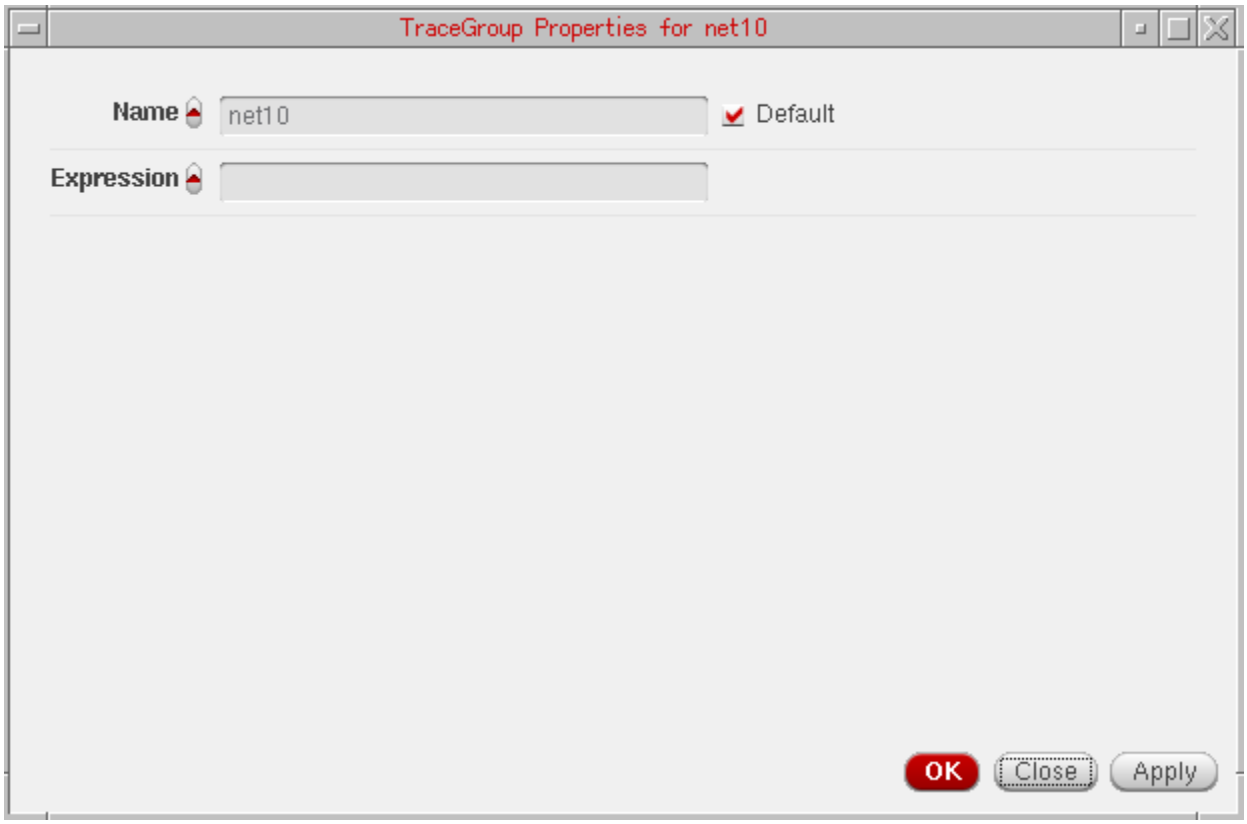
Changing Trace Properties for Family

To change the properties of a trace in the family, right-click the trace in the trace legend and choose *Trace Properties*. For more information about how to use the Trace Properties form, see Setting Trace Properties on page 392.

To change the properties of all traces in the family, select the traces by using the `Ctrl` key and choose *Trace Properties*. The *Trace Properties for selected objects* form appears. For more information about this form, see Setting Properties for Multiple Traces on page 395.

© 2023 Cadence Design Systems, Inc.

To change the trace group properties for a family of traces, right-click the family header in the trace legend and choose *Trace Group Properties*. The *TraceGroup properties for family-name* form appears.



Swapping Sweep Variables

The sweep data can include multiple sweep variables; however, you can plot sweep data analysis results by using only two variables at a time. If you want to plot the sweep data results with another variable, you can swap the X-axis variable with this variable.

To swap the sweep variables in the graph:

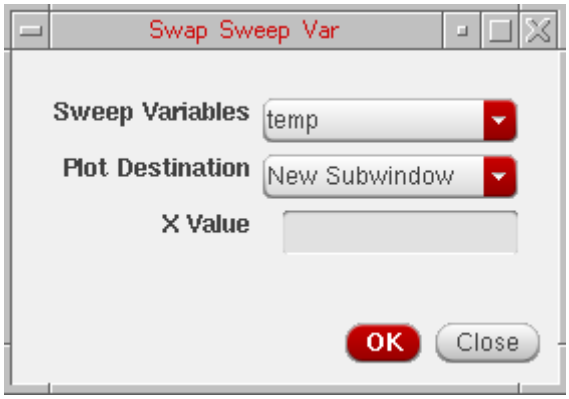
➡ Right-click the X-axis and choose *Swap Sweep Var.*

The *Swap Sweep Var* form appears. You cannot perform any action in the Virtuoso Visualization and Analysis XL window when the Swap Sweep Var form is open.

Note: The *Sweep Swap Var* option is enabled only if the simulation involves more than

© 2023 Cadence Design Systems, Inc.

one sweep variable.



The *Sweep Variables* drop-down list box in this form displays all the sweep variables. Now, perform the following steps:

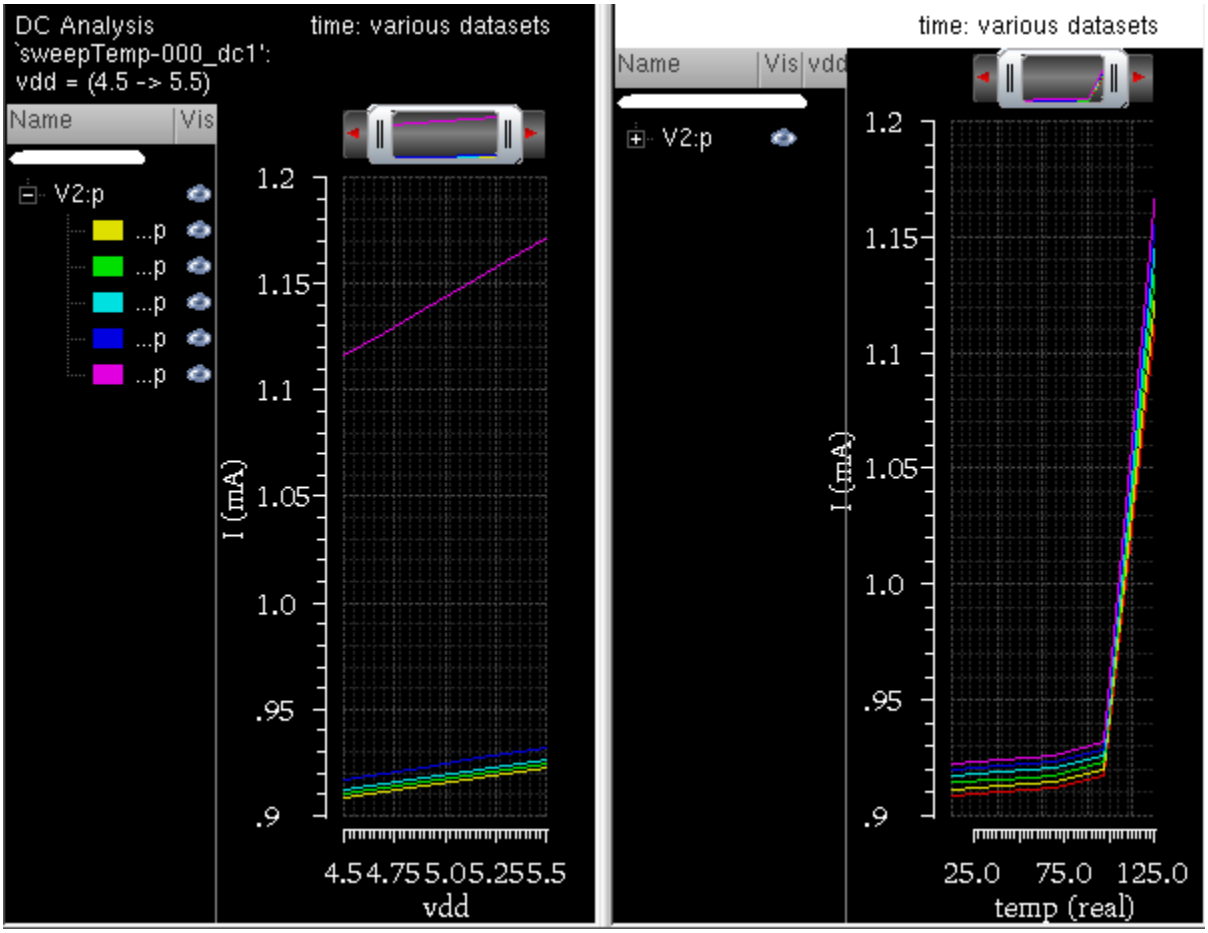
- Select from the drop-down list the sweep variable that you want to swap and specify the plot destination that can be a new window or a subwindow.

  Note: The *X Value* field is not available in this form. This field is used to plot data for signals for a particular X-axis value of frequency (AC response) and time (transient response). When the graph includes the plots against frequency or time, this field is enabled.

- Click *OK*.

© 2023 Cadence Design Systems, Inc.

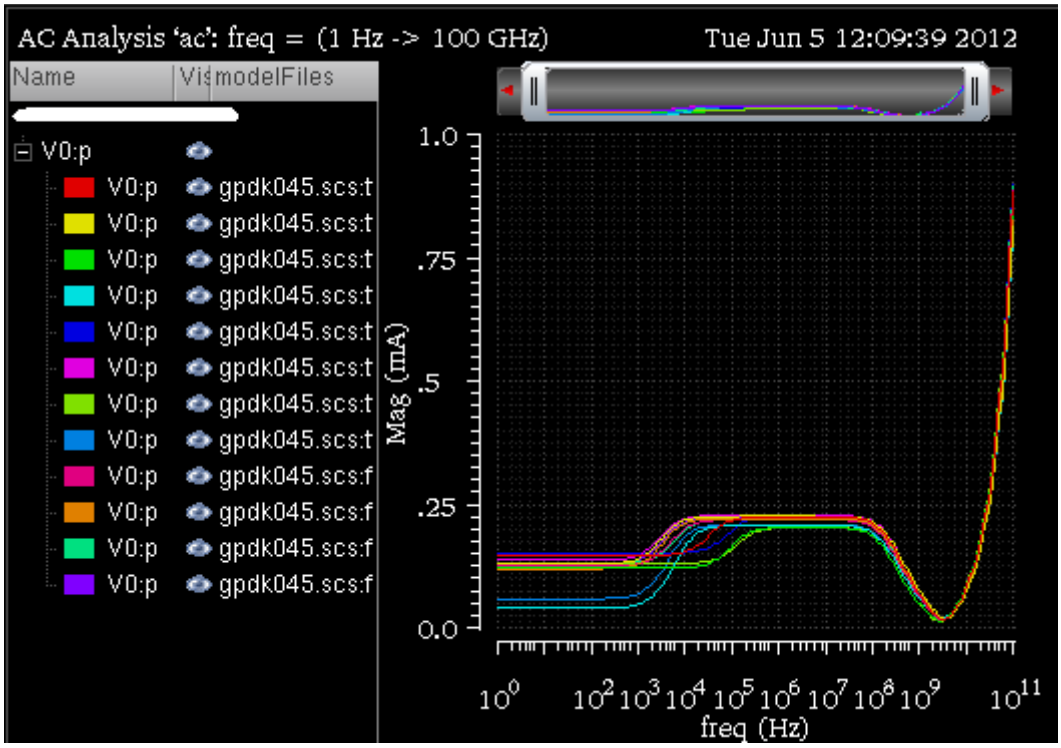
A new graph is created according to the swapped variable for the X-axis.



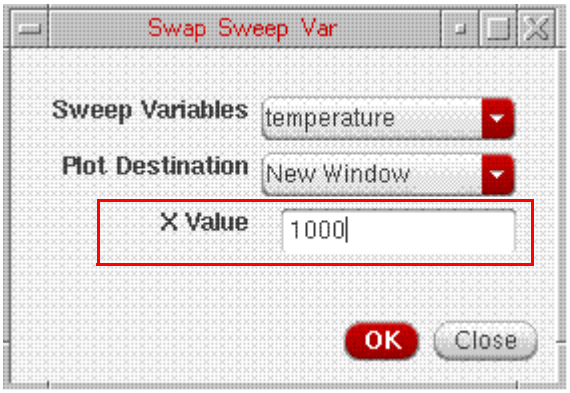
© 2023 Cadence Design Systems, Inc.

Plotting Graph across Fixed Frequency

The graph shown in the figure below is obtained from an AC analysis run on sweep data. In this graph, the signal `V0:p` is plotted against `frequency` on X-axis.

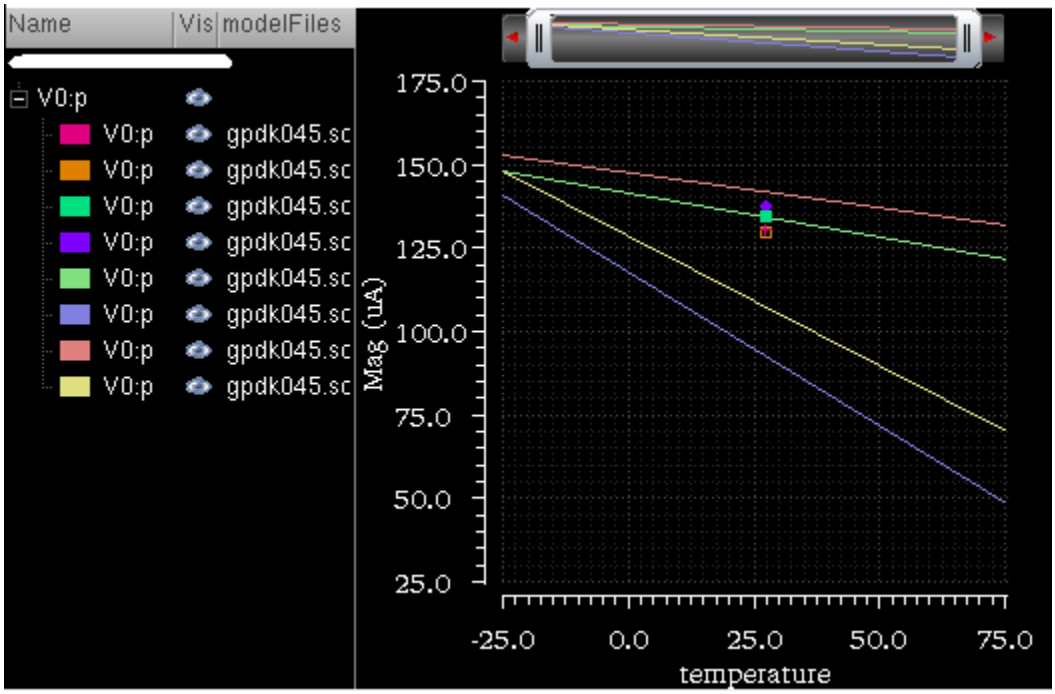


If you want to plot this graph against another sweep variable, `temperature`, and analyze the plot at a particular frequency, specify the frequency value in the *X Value* field in the Swap Sweep Var form (as shown in the figure below).



© 2023 Cadence Design Systems, Inc.

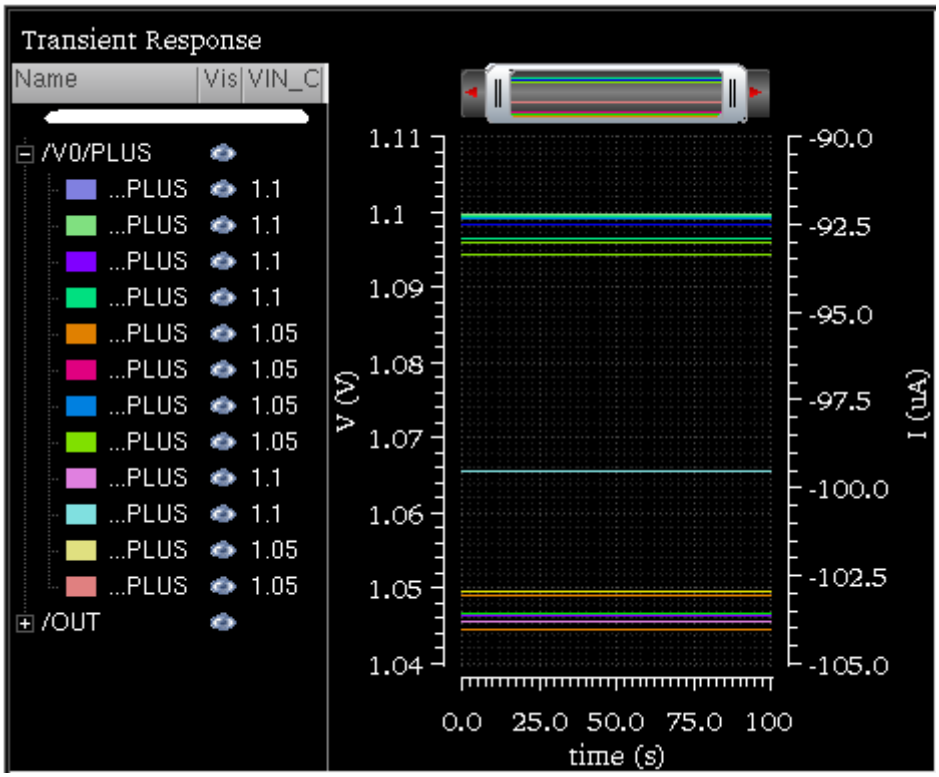
The graph is now plotted against temperature for frequency=1000 Hz (as shown in the figure below).



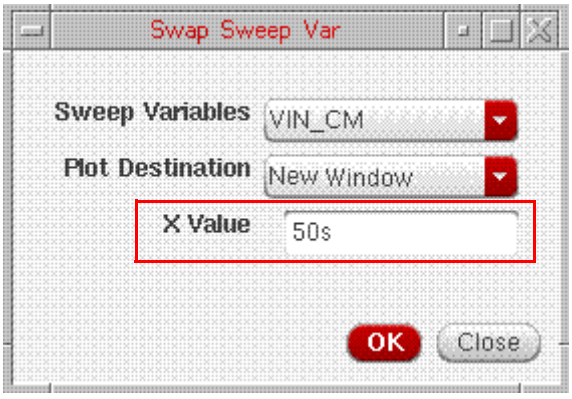
Plotting Graph across Fixed Time

The graph shown in the figure below is obtained from a transient analysis run on sweep data. In this graph, the signal `V0/PLUS` and `OUT` are plotted against `time` on X-axis.

© 2023 Cadence Design Systems, Inc.

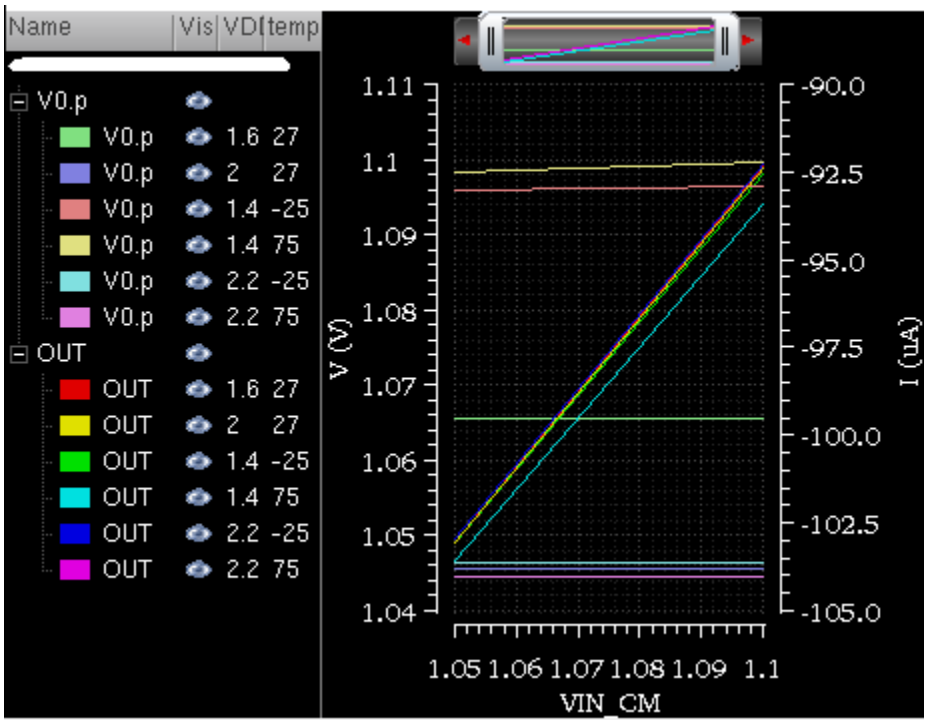


If you want to plot this graph against another sweep variable, `VIN_CM`, and analyze the plot at a particular time value, specify a time value in the *X Value* field in the Swap Sweep Var form (as shown in the figure below).



© 2023 Cadence Design Systems, Inc.

The graph is now plotted against `VIN_CM` for time=50s (as shown in the figure below).



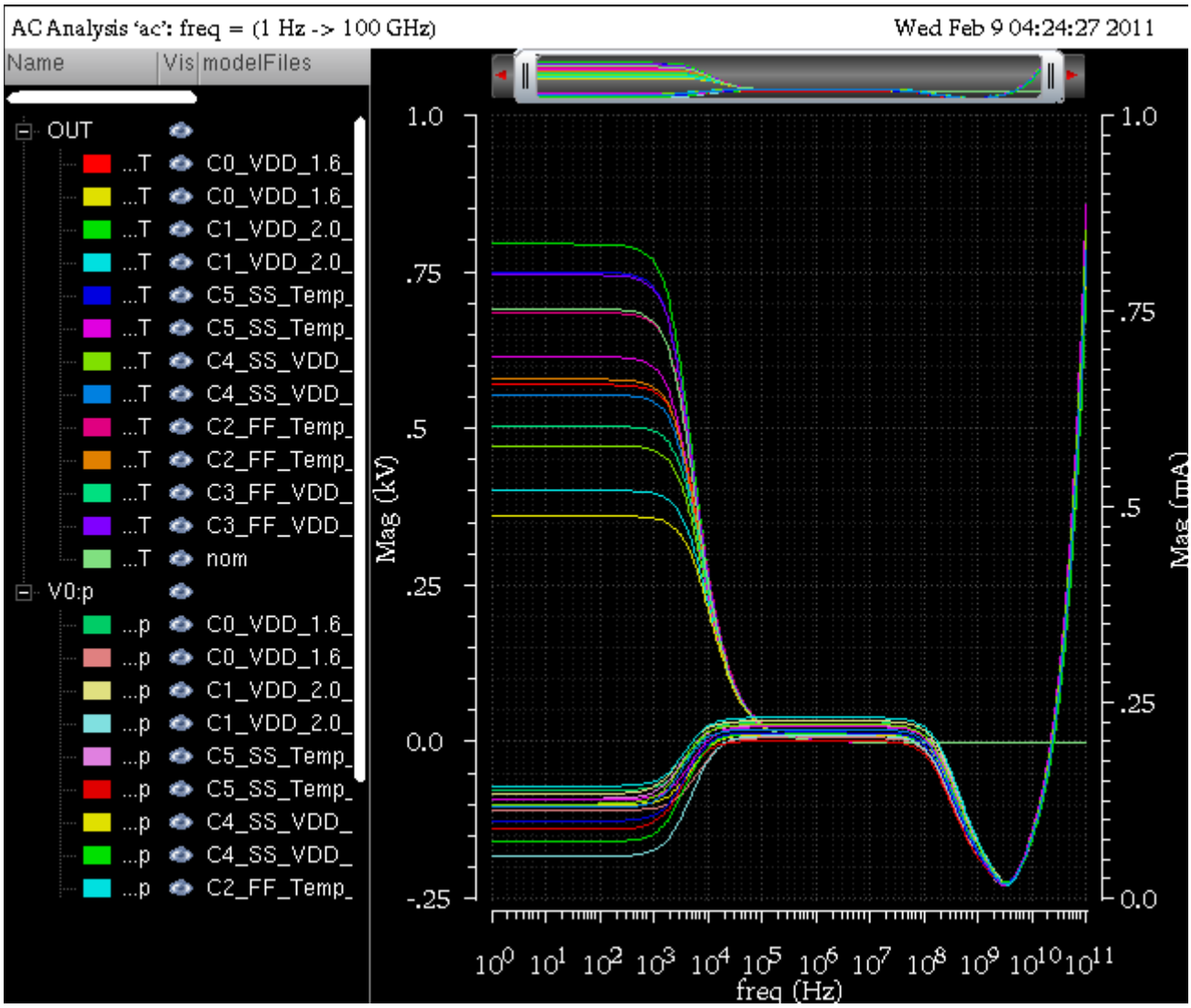
Filtering Traces Using Sweep Visibility Filter

To display in a graph specific traces that belong to a selected sweep data range, you can filter the traces by using the sweep visibility filter option. Filtering helps you analyze the simulation data in a specific sweep range.

The figure below shows the `OUT` and `V0:P` traces that are plotted after running a simulation is run in ADE XL for sweep data. The X-axis sweep variables for this simulation are—

© 2023 Cadence Design Systems, Inc.

modelFiles, *VDD*, *temperature*, *freq*, and *Corner*. The graph below is plotted with `freq` as the sweep variable on X-axis.



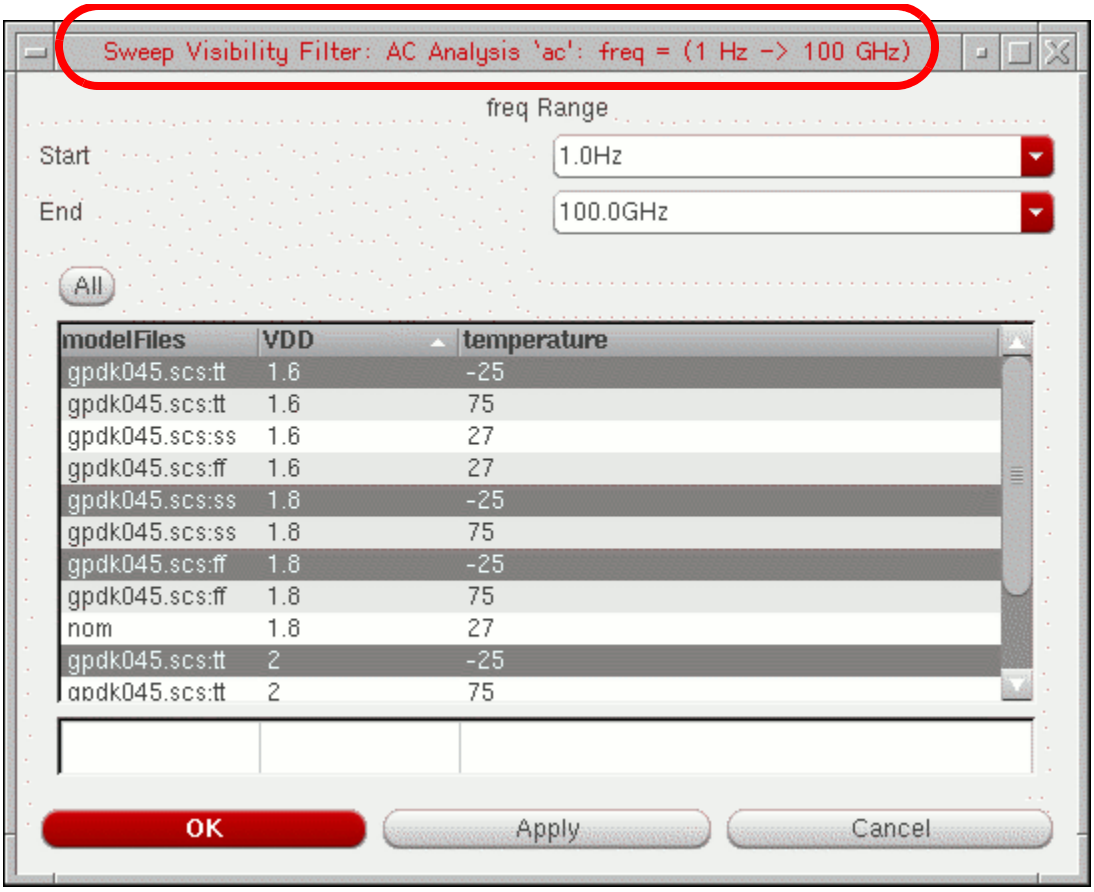
You can select different combinations of sweep variables to filter the visibility of traces that you want to display in the graph. For example, in the graph shown above, you can select a specific range of other sweep variables—*modelfiles*, *VDD*, and *temperature*—to filter traces. The traces that fall in the range you have selected are visible in the graph and the visibility of remaining traces is turned off.

To filter traces from sweep data, perform the following steps:

1. Choose *Graph – Filter By Sweep Var*.

© 2023 Cadence Design Systems, Inc.

The *Sweep Visibility Filter* form appears. The form name includes the subwindow name, as shown below.



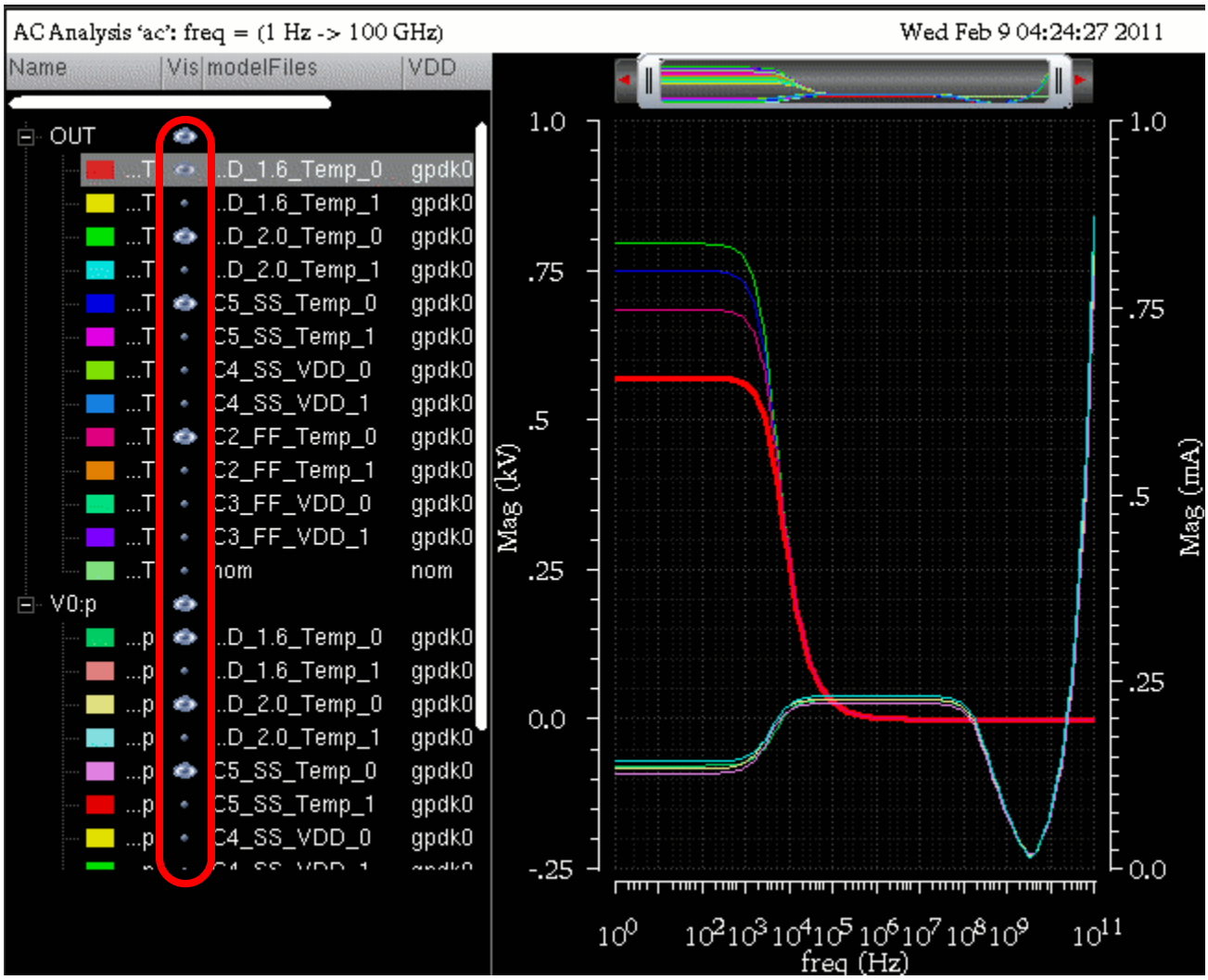
2. Select the variable values for which you want to display traces in the graph. For example, the form displayed in the figure above shows the following four traces selected for the *modelFiles*:*VDD*:*temperature* combination:

```
gpdk045.scs:tt, 1.6, -25
gpdk045.scs:ss,1.8, -25
gpdk045.scs:ff, 1.8, -25
gpdk045.scs.tt, 2, -25
```

3. Click *Apply*.

© 2023 Cadence Design Systems, Inc.

The traces for the selected combination of *modelFiles*, *VDD* and *temperature* values are displayed in the graph.



Note: The visibility icon in the trace legend area is ON only for the selected combination of sweep variables, as shown in the figure above.

You can also filter traces by selecting the *Set Sweep Ranges* option in the Results Browser. For more information, see Plotting a Transient Signal Over a Time Range and Plotting an AC Signal over a Frequency Range.

© 2023 Cadence Design Systems, Inc.

Working with Graph Labels

You can add labels in a graph to display information about the graph or a trace. You can also attach labels with markers. For information about how to attach and edit the labels on marker, see Adding Markers on page 495.

To add a graph label in a graph, do one of the following:

- Select a graph and choose *Graph – Add Label.*
- Right-click anywhere in the window and choose *Create Graph Label*.

A label is added to the graph you selected. By default, the label displays the string `Graph Label1`. You can change the graph label by double-clicking the displayed string.

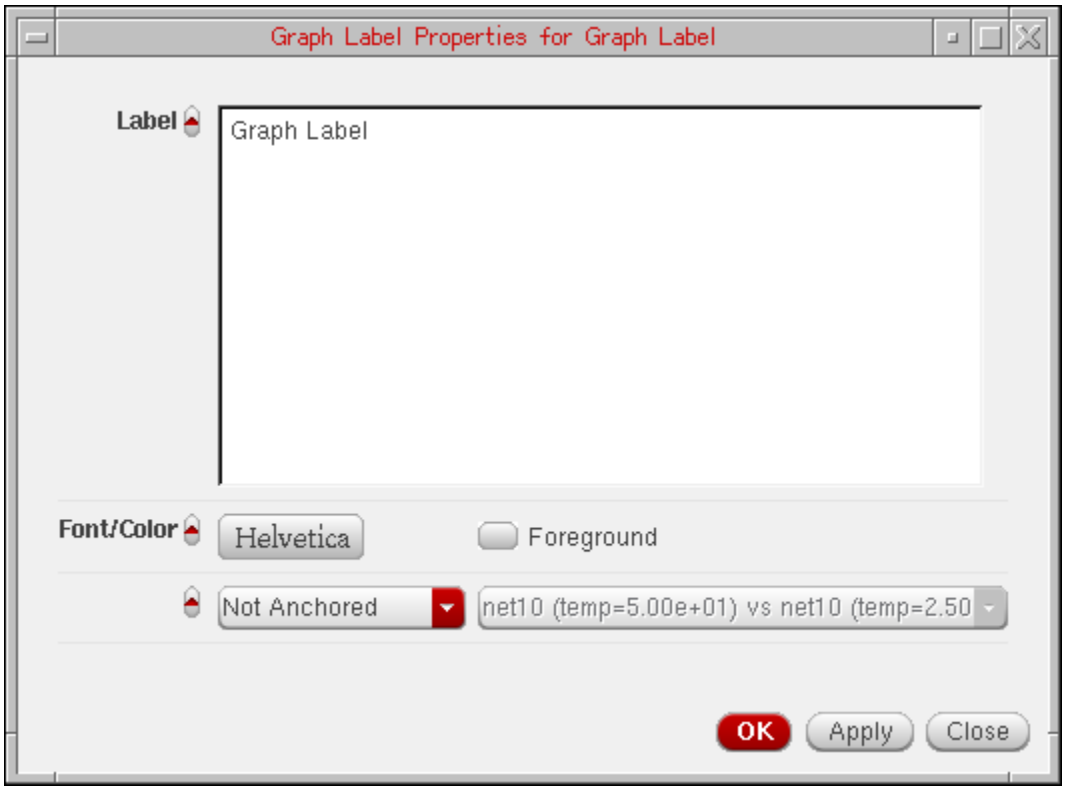
Note: To add a new line in the graph label, press `Enter`. The new line and carriage return characters are also supported in the graph labels.

To change the graph label properties, do one of the following:

- Right-click the graph label and choose *Graph Label Properties*.
- Select the label and press `Q`.

© 2023 Cadence Design Systems, Inc.

The *Graph Label properties for <Graph Label Name>* form appears.



This form includes the following fields:

- *Label*—Displays the default graph label name. You can provide a new name for the label.
- *Font*—Specifies the font type.
- *Foreground*—Specifies the foreground and background color of the graph label.
- Drop-down list box to select whether the label is an anchored frame or attached to a trace. You can select the trace name in the drop-down.

■ Click *OK* to save the changes you made.

Note: You can change the position of a graph label by dragging the graph label to a new location.

Deleting Graph Labels

To delete a graph label, do one of the following:

© 2023 Cadence Design Systems, Inc.

- Select a label and choose *Edit – Delete*, or press the `Delete` key.
- Right-click the label and choose *Delete*.

Creating Multiple Graph labels

To create multiple graph labels, right-click a label in the graph and choose *Copy*. Then, right-click any where in the window and choose *Paste*. A copy of the graph label is created. Drag one of the labels to the required new position. Using this method, you can create as many labels as you want.

Note: You cannot move a label that is attached to a marker.

Plotting WREAL Signals

Virtuoso Visualization and Analysis XL supports the plotting of WREAL (wire-real) signals, where the WREAL signals are by default plotted in the sample and hold plot type. The WREAL signals are displayed with the symbol in the Results Browser.

Important

If you want to disable the WREAL plotting, set the `vivaWrealSupport` environment variable to `false`:

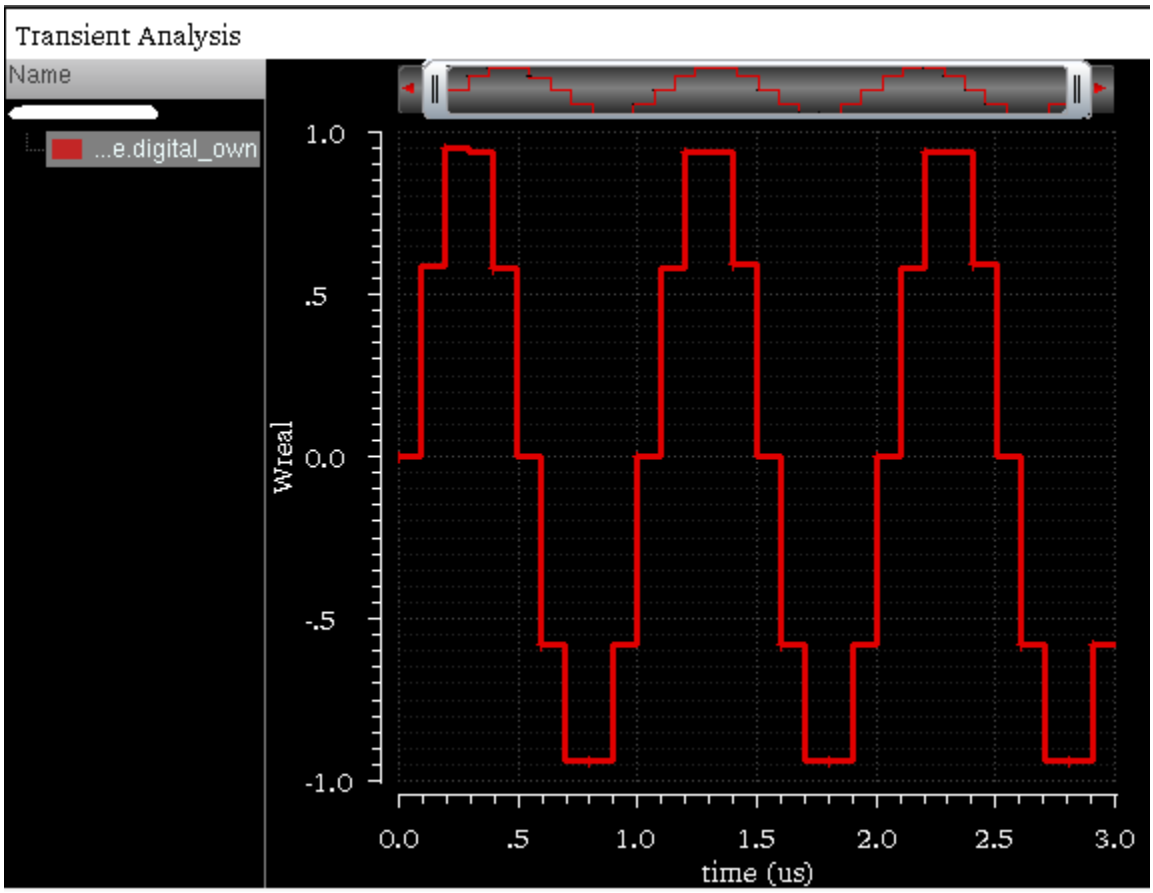
```
setenv vivaWrealSupport false
```

Perform the following step to set the plot type of a trace to sample and hold:

➥ Right-click the trace for a WREAL signal and choose *Type – Sample Hold*.

© 2023 Cadence Design Systems, Inc.

The following figure displays a WREAL signal plotted in the sample and hold plot type.

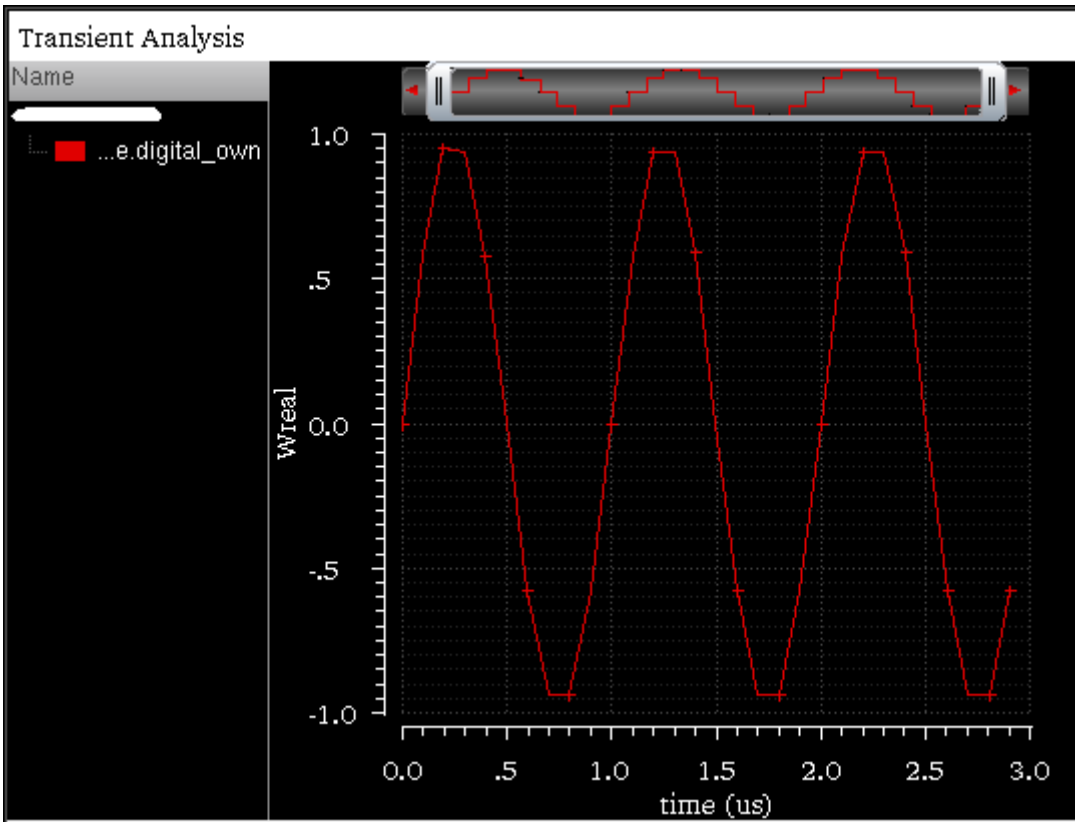


By default the data points are visible in the plots for the WREAL signals. To hide the data points, do one of the following:

- Choose *Trace – Symbols On*.
- Right-click the trace and deselect *Symbols On*.
- Right-click the trace and choose *Symbol – Symbols Off*.

© 2023 Cadence Design Systems, Inc.

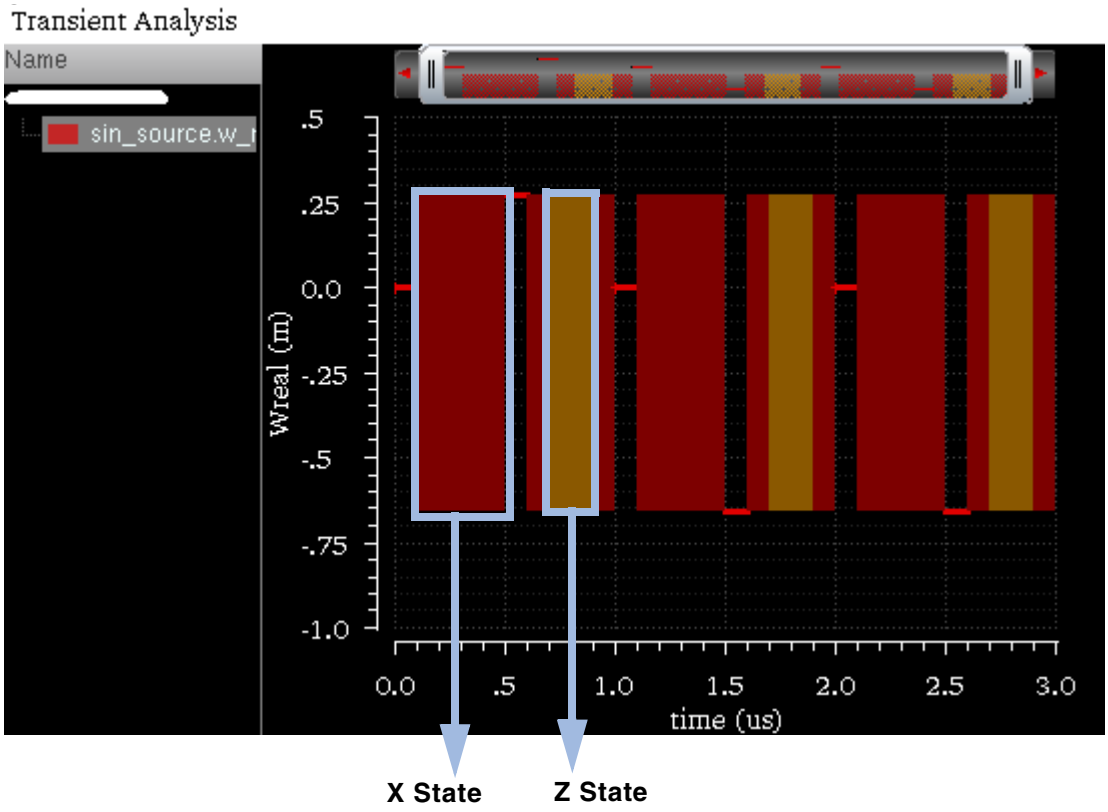
To change the plot type to a continuous line, right-click the trace and choose *Type – Continuous line*.



To change the properties of a WREAL trace, right-click the trace and choose *Trace Properties*. The *Trace Properties for <WREAL trace name>* form appears. This form includes the similar fields as that of the rectangular trace. For more infomation about the fom fields, see Setting Trace Properties on page 392.

© 2023 Cadence Design Systems, Inc.

WREAL plots includes a special depiction style to display the X and Z states. In the figure below, the red blocks indicate the X state and the yellow blocks indicate the Z state.



Limitations

Following are the limitations while plotting the WREAL signals:

- The X and Z states are not visible when the trace is exported into a table.
- The X and Z states are not plotted when you plot the WREAL signal from ADE and Calculator.
- Single point WREAL signals are not displayed as a line.
- WREAL array data is not displayed as a group and each member of the array is displayed separately.
- Zooming a WREAL signal does not work properly.

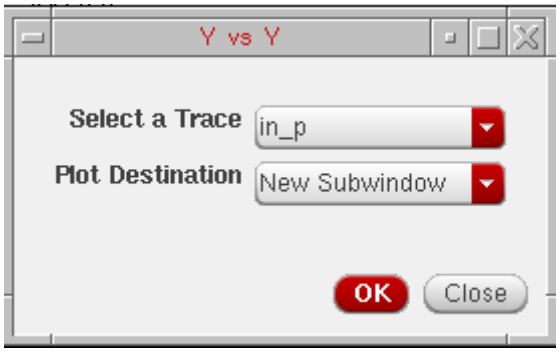
© 2023 Cadence Design Systems, Inc.

Plotting YvsY Graph

To plot a YvsY graph for the sweep data, do the following:

- Right-click the X-axis and choose *YvsY.*

  The *YvsY* form appears. You cannot perform any action in the Virtuoso Visualization and Analysis XL window when the YvsY form is open.

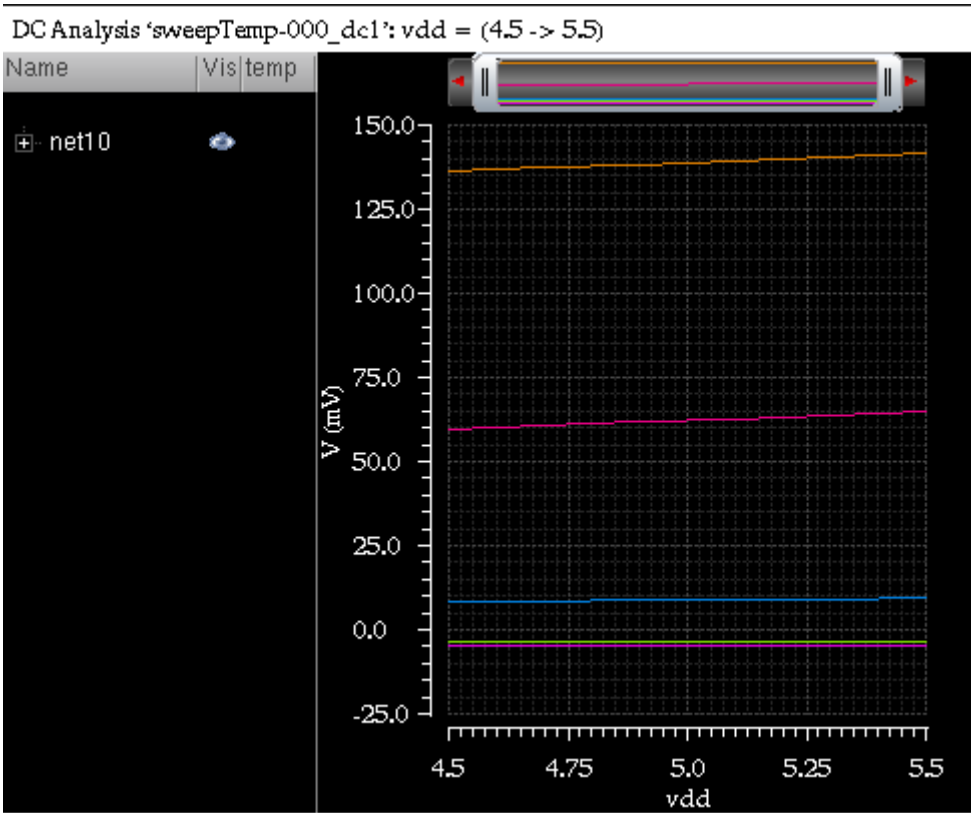


- Select a trace in this form and specify the plot destination, which can be a new window or a subwindow.
- Click *OK.*

  The YvsY plot is created in the destination window that you select. For example, the YvsY plot for `net10` is displayed below.

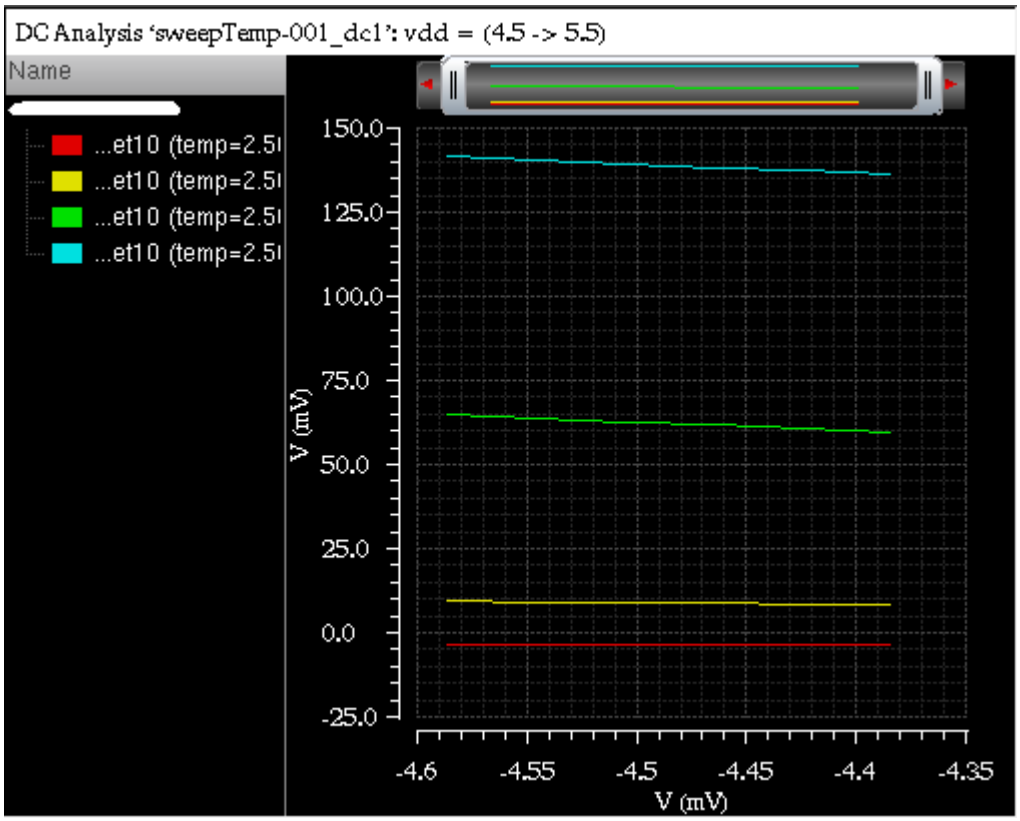
© 2023 Cadence Design Systems, Inc.

Signal `net10`:



© 2023 Cadence Design Systems, Inc.

YvsY of `net10`:



Plotting Normal Quantile Graphs

Currently, you cannot plot normal quantile graphs using the stand-alone mode of the Virtuoso Visualization and Analysis XL tool. You can plot normal quantile graphs only for the Monte Carlo results from the *Results* tab of the ADE XL window.

Saving and Loading Graphs

You can save a graph to a file for future use. When you save a graph, the graph settings, such as zooming and panning, changing font type and font color, setting labels for X and Y axis, or changing the trace color are also saved with the graph file. As a result, you save on the effort required to customize the graph when you display the same graph again.

When you load the saved graph in a window and append a trace to that window, the new trace is displayed with the same window attributes.

© 2023 Cadence Design Systems, Inc.

Note: If you open a new graph in a subwindow, the graph appears in the default graph attributes.

This section includes the following topics:

- Saving a Graph on page 431
- Loading a Graph on page 433
- Saving a Graph as an Image on page 434

Saving a Graph

The graph is saved as an XML file with the `.grf` extension. The following information is saved with the graph:

- The location of the data—data directory, data set, and trace name—and not the original data. Therefore, if your simulation data changes between sessions, the graph reflects those changes.
- All graph objects and attributes, such as grids, background and foreground color, labels, and markers.

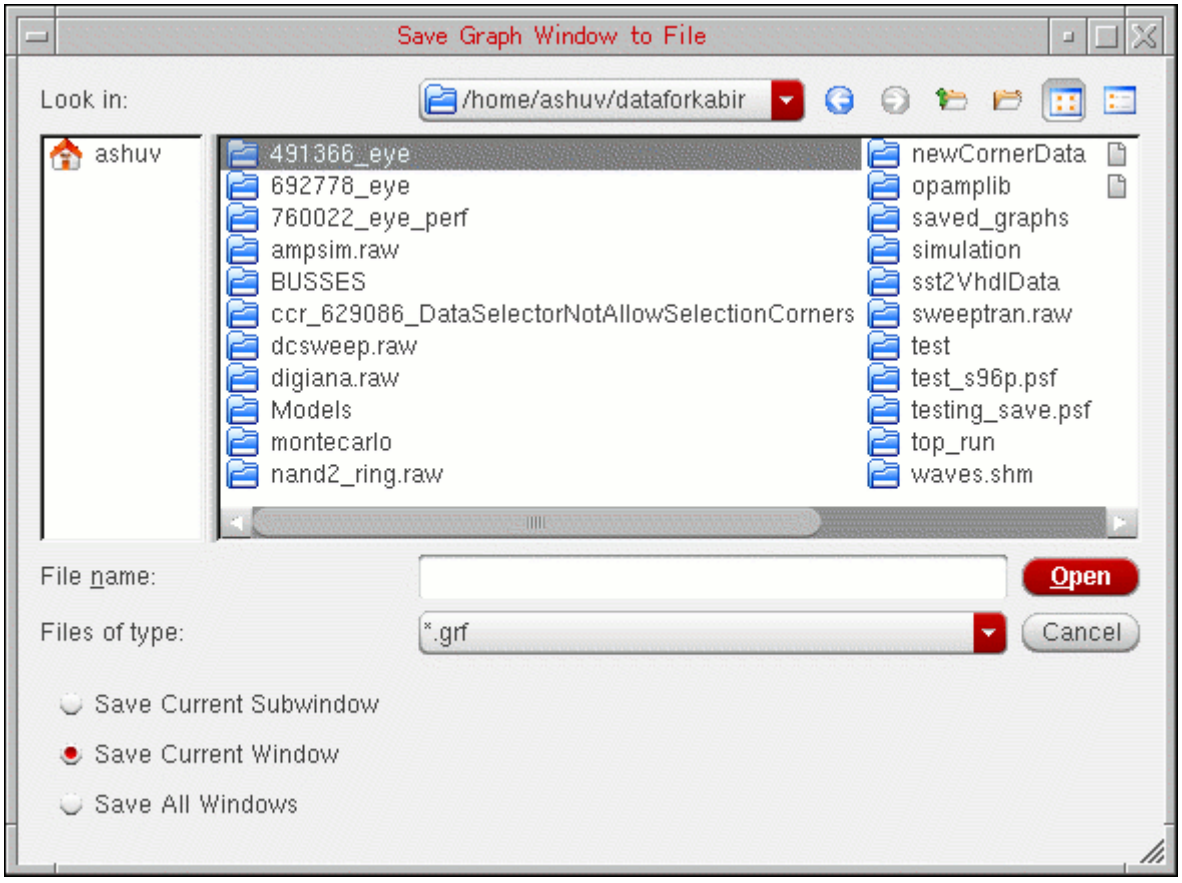
Save a group of windows with the file extension `.grf.group`.

To save a window, plot a signal in the window and do the following:

- Choose *File – Save Window As*, or press `Ctrl+S`.

© 2023 Cadence Design Systems, Inc.

The *Save Graph Window to File* form appears.



- In the *Look in* field, select the directory where you want to save the graph. The file extension is displayed as `.grf` in the *Files of Type* field.
- Now, do one of the following:
  - If you want to overwrite an existing graph file, select that graph file from the list box below the *Look in* field.
  - In the *File name* field, type a name for the graph file to which you want to save the graph.
- Select one of the following options:
  - *Save Current Subwindow*
  - *Save Current Window*
  - *Save All Windows*

© 2023 Cadence Design Systems, Inc.

When you select the *Save All Windows* option, the graphs are saved in a `.grf.group` file.

- Click *Save*.

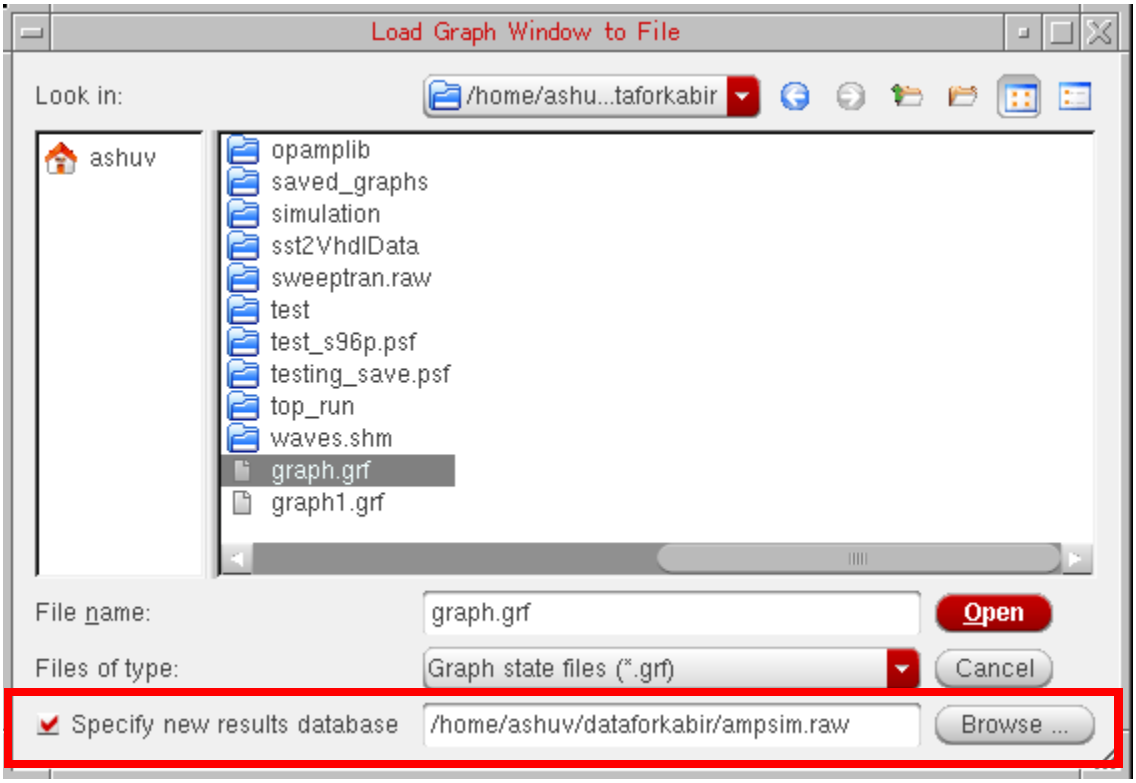
  The window that you selected is saved at the specified location.

Loading a Graph

Perform the following steps to load a graph that you have saved.

1. Choose *File – Load Window,* or press `Ctrl+L`.

   The *Load Graph Window to File* form appears.



   In the *Look in* field, select the directory in which the `.grf` file that you want to open exists.

   Select the *Specify new results database* check box and click the *Browse* button to load the same `.grf` file, either with the original data or different data.

2. Now, do one of the following:

© 2023 Cadence Design Systems, Inc.

- Select the graph file you want to open from the list box below the *Look in* field.
- In the *File name* field, type the name of the file you want to open.
- Select the `.grf` extension to display the graph files and `grf.group` file extension in the *Files of type* drop-down list box to select a graph file group. The extension for graph files is specified by the `filesuffix` variable in the `.cdsenv` file.
- Select the *Specify new results database* check box to plot the saved graph from a new results directory. Also, specify the name of the results directory from which you want to plot the graph. Ensure that the signal plotted in the saved graph exists in the results directory you specified.

  When you use this option, the saved graph is updated with the data from the new results directory and all the trace settings that you have applied to the saved graph are also retained. Hence, saved graphs can be used as a template when you reload a graph.

3. Click *Open*.

   The saved graphs are displayed in a new window. This graph has all the attributes that you saved with the graph.

   Note: If you have multiple windows open, loading a new window does not affect these windows.

You can also type the following command in the terminal window to load the graph saved in a graph state file (.grf).

```
viva -load_graph myGraphs.grf
```

Saving a Graph as an Image

If you save the graph as an image, the active graph or subwindow is saved with the trace legend area. You can insert the graph image into a document or print it.

Note: You cannot load a graph image in a window.

The Virtuoso Visualization and Analysis XL tool provides several image formats to support a variety of applications and environments. While all image formats are functionally equivalent to binary storage formats, the size of a typical file varies greatly according to the format chosen. For example, a simple graph saved in the PNG format is typically less than 100 KB, while the same file saved in the TIFF or BMP format may exceed 1MB.

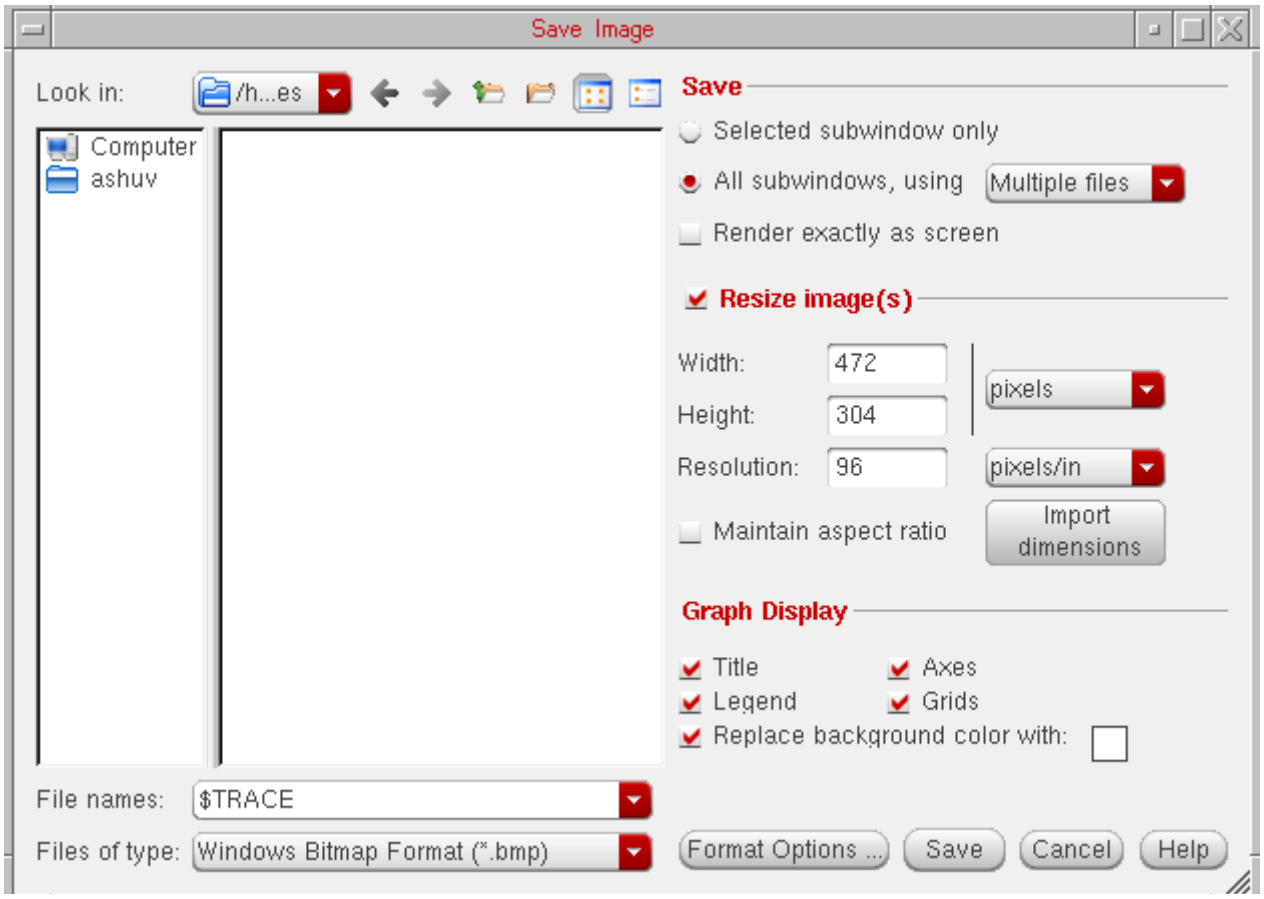
Perform the following steps to save a graph as an image file.

1. Plot a signal in a graph.

© 2023 Cadence Design Systems, Inc.

2. Choose *File – Save Image*

   The *Save Image* form appears.



3. In the *Look in* field, browse to locate the directory where you want to save the image file.

4. In the *File name* field, type a name for the image file. You can also specify macros, `$TRACE` and `$SUBWIN`, in the filename. These are used to save image files with the trace names and subwindows names.

5. In the *Files of type* drop-down list box, select the format in which you want to save the image file.

   By default, the file is saved in a format based on the file extension you specify. For example, if you type `output.png`, the file is saved in the PNG format. The image file can be saved with `.png`, `.bmp`, and `.tiff` (or `.tif`) file extensions. On the other hand, if you select *PNG (Best compressed)* and type `output.tiff`, the file is saved as `output.png`. Though PNG and TIFF files are compressed, there is no loss in image quality with these image formats.

   The graph image file can be saved in the following formats:

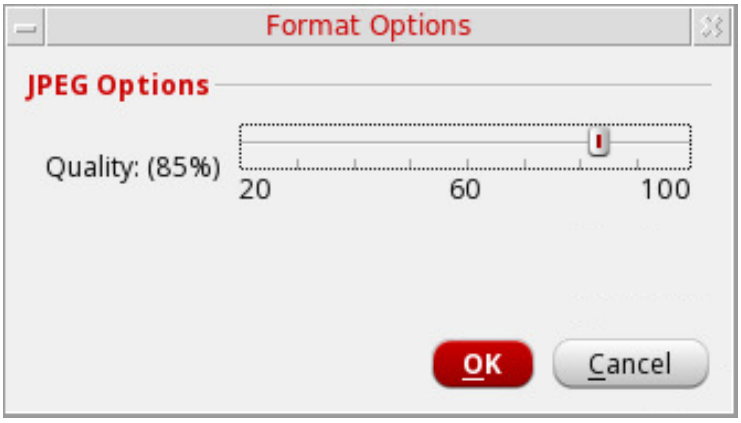
© 2023 Cadence Design Systems, Inc.

- Windows Bitmap Format (*.bmp)
- JPEG Format (*.jpg)
- Portable Network Graphics (*.png)
- Portable Pixmap Format (*.ppm)
- Tagged Image File Format (*.tif)
- X Pixmap Format (*.xpm)
- Adobe PDF (*.pdf)
- Scalable Vector Graphics (*.svg)

The default file type is `JPEG`.

6. To specify the format properties, click the *Format Option* button.

   The *Format Option*s form appears. This form includes the following fields:



   A quality bar, ranging from 20–100, is displayed under the *JPEG Options* section. Move the slider to the right or to the left to increase or decrease the quality of the graph image being saved. This quality bar is activated only if the file type is `JPEG`.

7. In the *Save* section, select the following fields:

   - *Selected subwindow only*—Select this button if you want to save only the selected subwindow.
   - *All subwindows, using*—Select this button if you want to save all the subwindows. You can use the drop-down list to specify whether you want to save subwindows in a single file or multiple files.

     Note: If you choose to save subwindows separately in multiple files, ensure that you specify the following macros: $TRACE or $SUBWIN. Depending on the specified

© 2023 Cadence Design Systems, Inc.

macro, the graphs in the subwindows are saved in different files named either by their trace names or subwindow numbers.

- *Render exactly as screen*—Select this check box if you want to save the exact copy of graph as it is visible on the screen. This also saves all attributes of the graph as well as zooming and panning properties. When you select this option, the *Graph Display* section is disabled.

This option saves only those strips that are visible on the active graph. If you do not want to save a strip, you can adjust the graph window size to hide the strip.

8. *Resize Image(s)*—Select this check box if you want to resize the image. In this section, you can change the following image attributes:

   - *Width*—Specify the width of the image.
   - *Height*—Specify the height of the image
   - You can select the units for the height and width from the drop-down list displayed at the right. The available units are `pixels`, `inch`, `cm`, `mm`, `picas`, and `points`. When you change the image units, the height and width values you have specified are automatically changed as per the selected units.
   - *Resolution*—Specify the resolution of the image. This field is unavailable if you select the image type in the vector format. Select the resolution type as `pixels/cm` or `pixels/in` from the drop-down list displayed at the right. When you change the resolution type, the height and width fields also change accordingly.
   - *Maintain Aspect Ratio*—The ratio of the width of the image to its height. Select this check box if you want to maintain the aspect ratio while modifying the height or width of the image.
   - *Import Dimensions*—Click this button to specify the default values for height, width and resolution.

9. *Graph Display*—In the this section, specify the following fields:

   - *Title*—Select this check box if you want to display the graph title in the graph image
   - *Legend*—Select this check box if you want to display trace legend in the graph image.
   - *Axes*—Select this check box if you want to display axes in the graph image.
   - *Grids*—Select this check box if you want to display grids in the graph image.
   - *Replace background color with:*—Select this check box if you want to save the graph image in a different background color. You can specify a new color by clicking

© 2023 Cadence Design Systems, Inc.

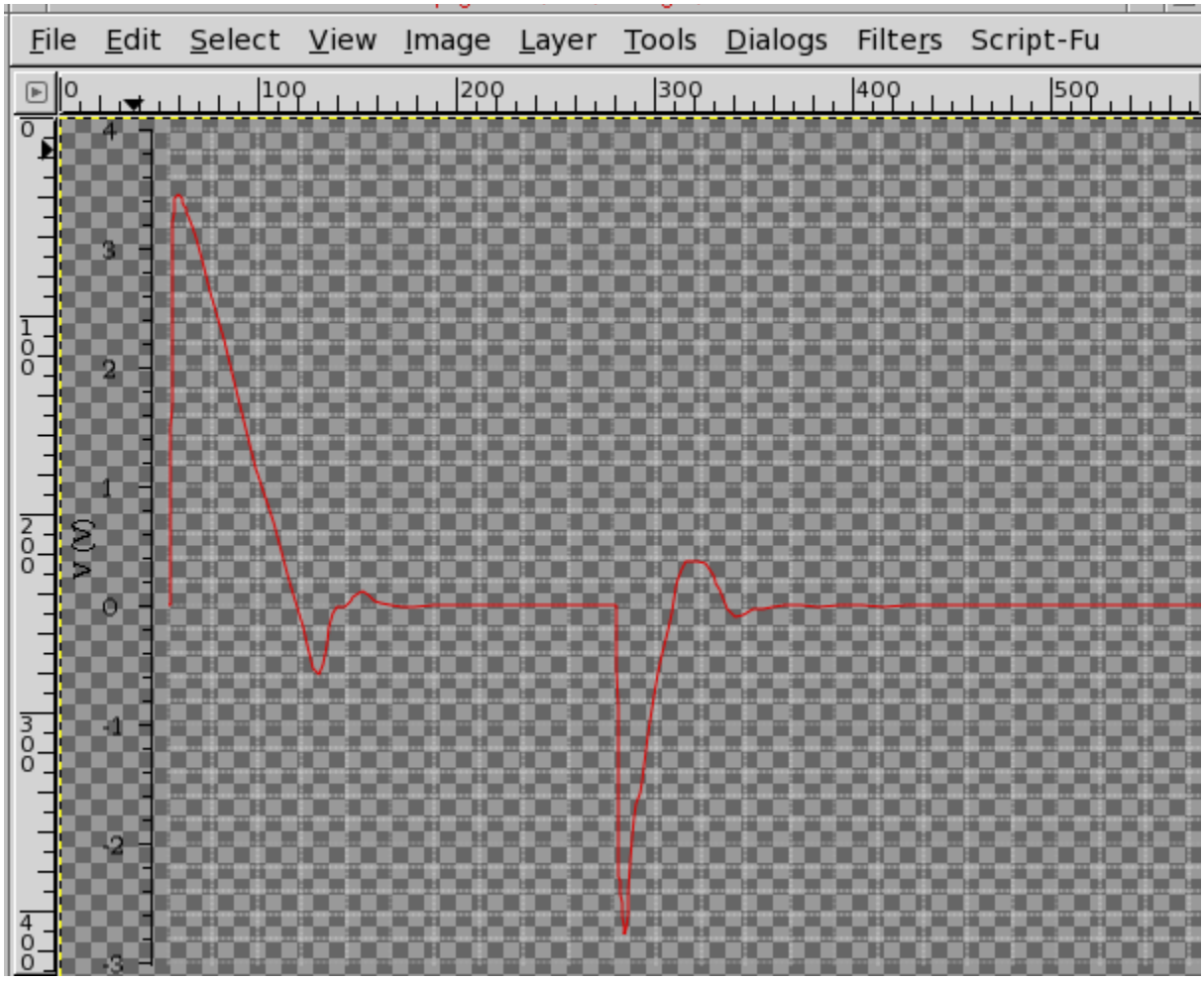
the button provided with this option. This option is not available when the file type is `pdf` and `svg`.

Note: All the above options are selected by default.

10. Click *Save*.

    The graph is saved as an image file with the specified attributes.

When you do not specify a background color and save the graph as an image, the saved image may not be in readable form. For example, when you open the saved image by using an image viewer, such as GIMP, the graph appears as shown in the figure below:

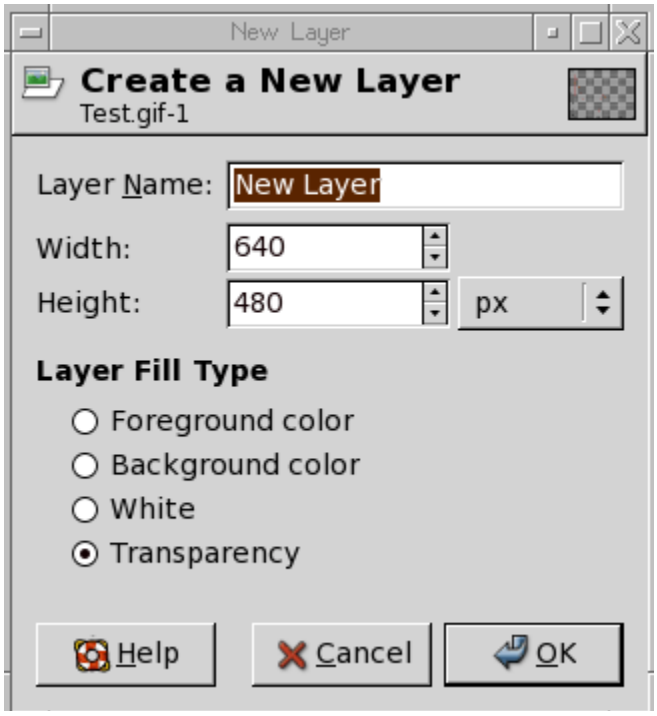


You can use the GIMP image viewer and editor to convert the image into a readable format by performing the following steps:

1. Choose *Layer – New Layer*.

© 2023 Cadence Design Systems, Inc.

The *New Layer* form appears.

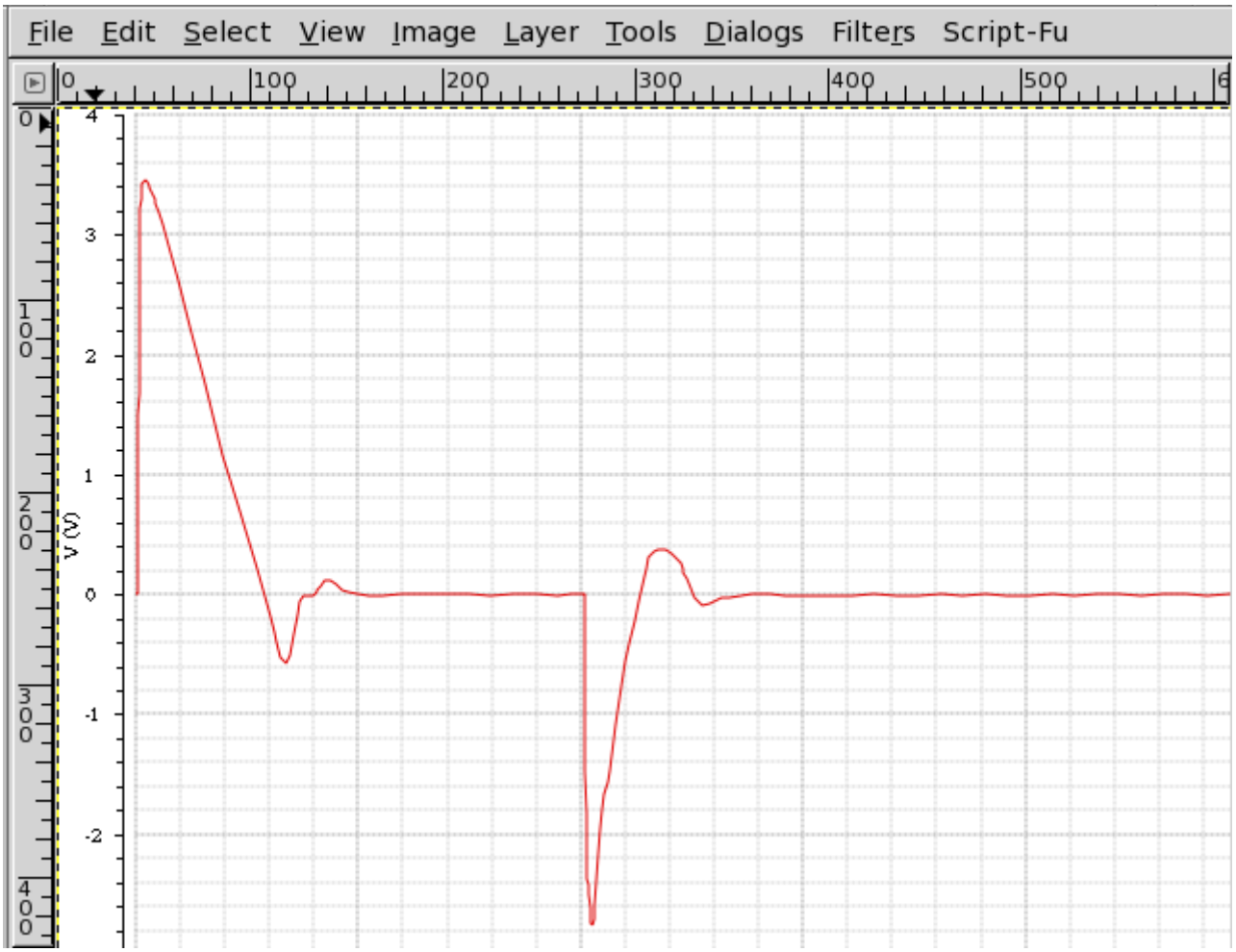


2. In this form, select the *Layer Fill Type* field as *Background color* and click *OK*.

   The image is filled with the white background color.

© 2023 Cadence Design Systems, Inc.

3. Choose *Layer – Stack – Layer to Bottom*. The image is now in readable format.



Reloading Graphs

Reloading a graph updates the already plotted traces with the latest simulation results based on the current in-context results directory. When you reload a graph, the settings that you have applied to a trace are also applied to the reloaded traces, such as, background color and font. For more information about the trace settings that are retained during reloading a graph, see Graph Settings Retained During Reloading on page 448.

You can reload a graph when you open the Virtuoso Visualization and Analysis XL tool in the stand-alone mode and also when you run the tool from ADE.

This section includes the following topics:

© 2023 Cadence Design Systems, Inc.

- Reloading Graph When Opened in Stand-Alone Mode on page 441
- Reloading Graph When Opened From Within ADE Explorer, or ADE Assembler on page 447
- Disabling Trace Reload on page 447
- Graph Settings Retained During Reloading on page 448

Reloading Graph When Opened in Stand-Alone Mode

In the stand-alone mode, you can reload an already open graph with the simulation results based on the current in-context results directory selected in the Results Browser.

Following are the examples that describe how reloading works in the stand-alone mode:

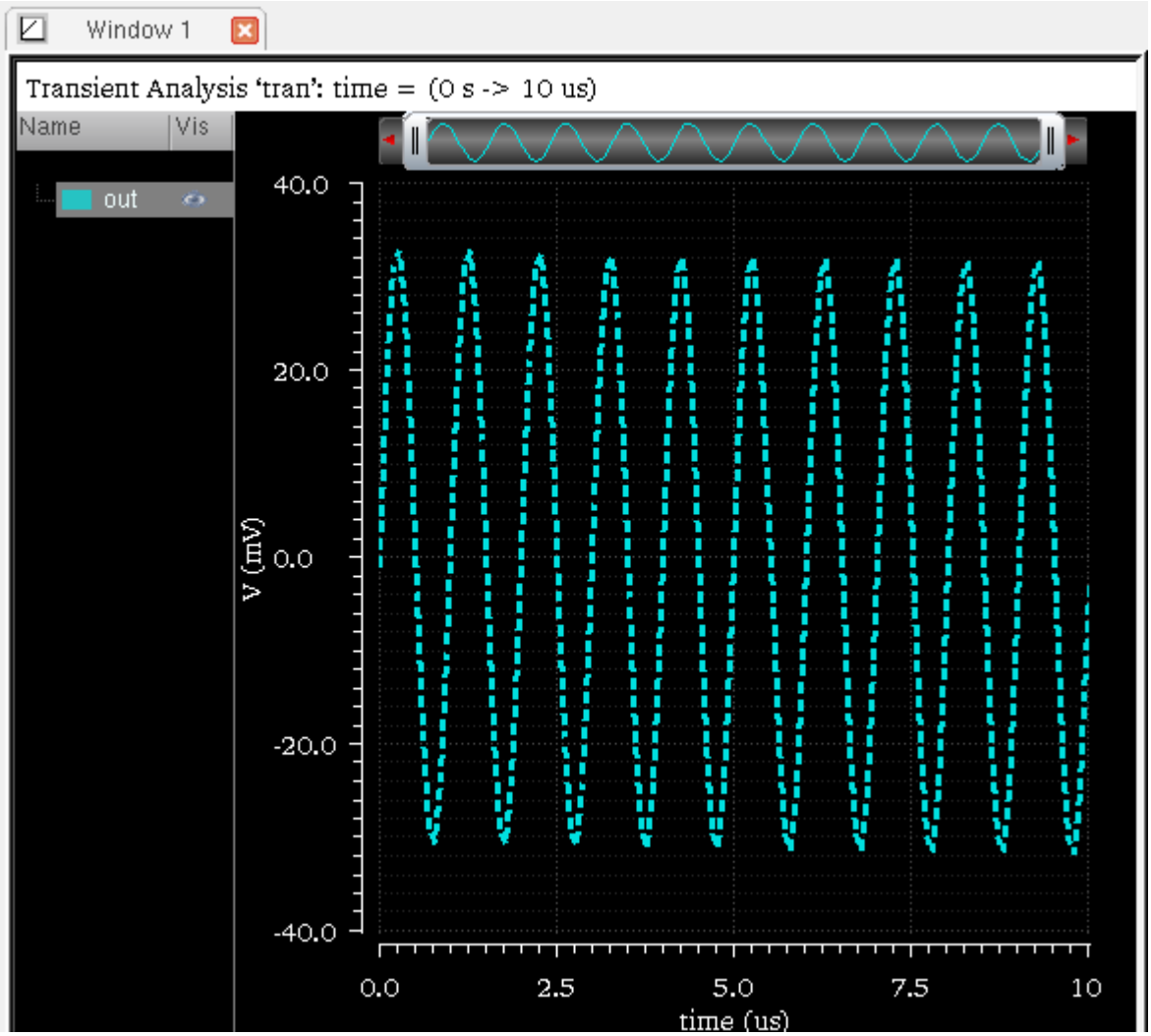
Example 1

Consider the following scenario in which you plot a trace from a results directory, change the in-context results directory, and reload the trace with the data from a new in-context results directory.

1. Plot a signal, `out`, from an in-context results directory (`simulation1`) in a new window.

© 2023 Cadence Design Systems, Inc.

2. Set the trace *Style* to `Dot` (as shown in the figure below). Notice that the time range of the `out` signal varies from 0.0 to 10 microseconds.



3. Open another results directory, `simulation2`, which contains the simulation results for the same design.

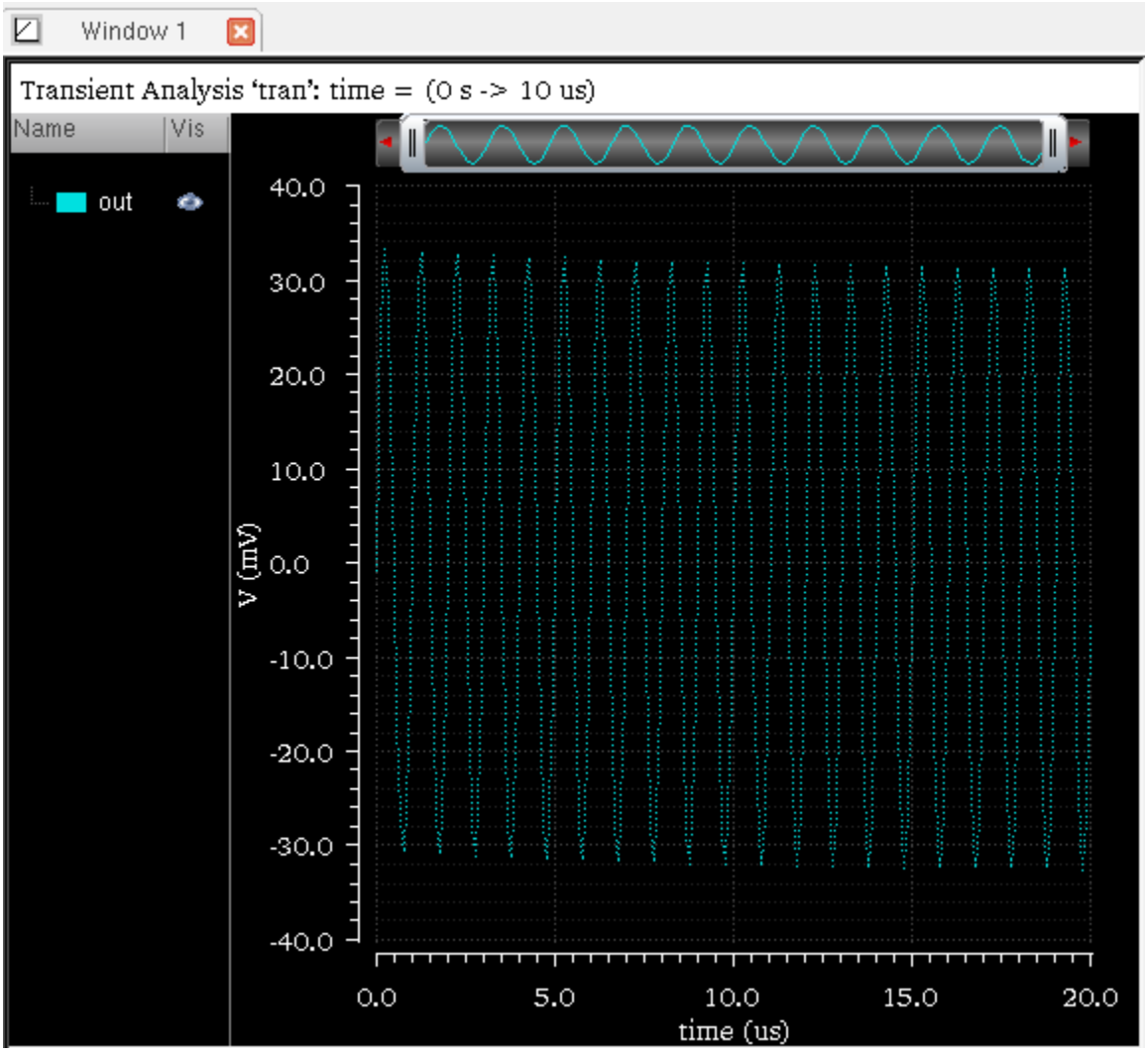
4. Set the database context to this new results directory, `simulation2`. For more information about how to change the in-context results directory, see Setting the In-Context Results Directory.

5. Now, to reload the already plotted trace for the `out` signal from the previous results directory (`simulation1`) with the data from the new in-context results directory (`simulation2`), choose *File – Reload – Current Subwindow.*

   Note: Alternatively, you can press `Ctrl+R` to reload graphs in the current window.

© 2023 Cadence Design Systems, Inc.

The trace for the `out` signal is reloaded using data from the latest in-context results directory, `simulation2`, and all the trace settings are retained (as shown in the figure below). Notice that the time range of the reloaded trace now varies from 0.0 to 20 microseconds.



Note: You can select the *Reload using current context* check box available on the Graph properties form to reload signals according to the data from the current in-context results directory. If this check box is not selected, signals are reloaded based on their individual databases. This check box is selected by default.

To reload the traces in all the subwindows, choose *File – Reload – All Subwindows*. All the traces for the common signals are updated with data from the current in-context results directory and all the trace settings are retained.

© 2023 Cadence Design Systems, Inc.

To reload the traces in all the windows in Virtuoso Visualization and Analysis XL, choose *File – Reload – All Windows*.

Example 2

You can also reload the traces for expressions created in the Calculator. Consider the following scenario in which you plot a trace for an expression, change the in-context results directory in Results Browser, and then reload the plotted trace with the data from the new in-context results directory

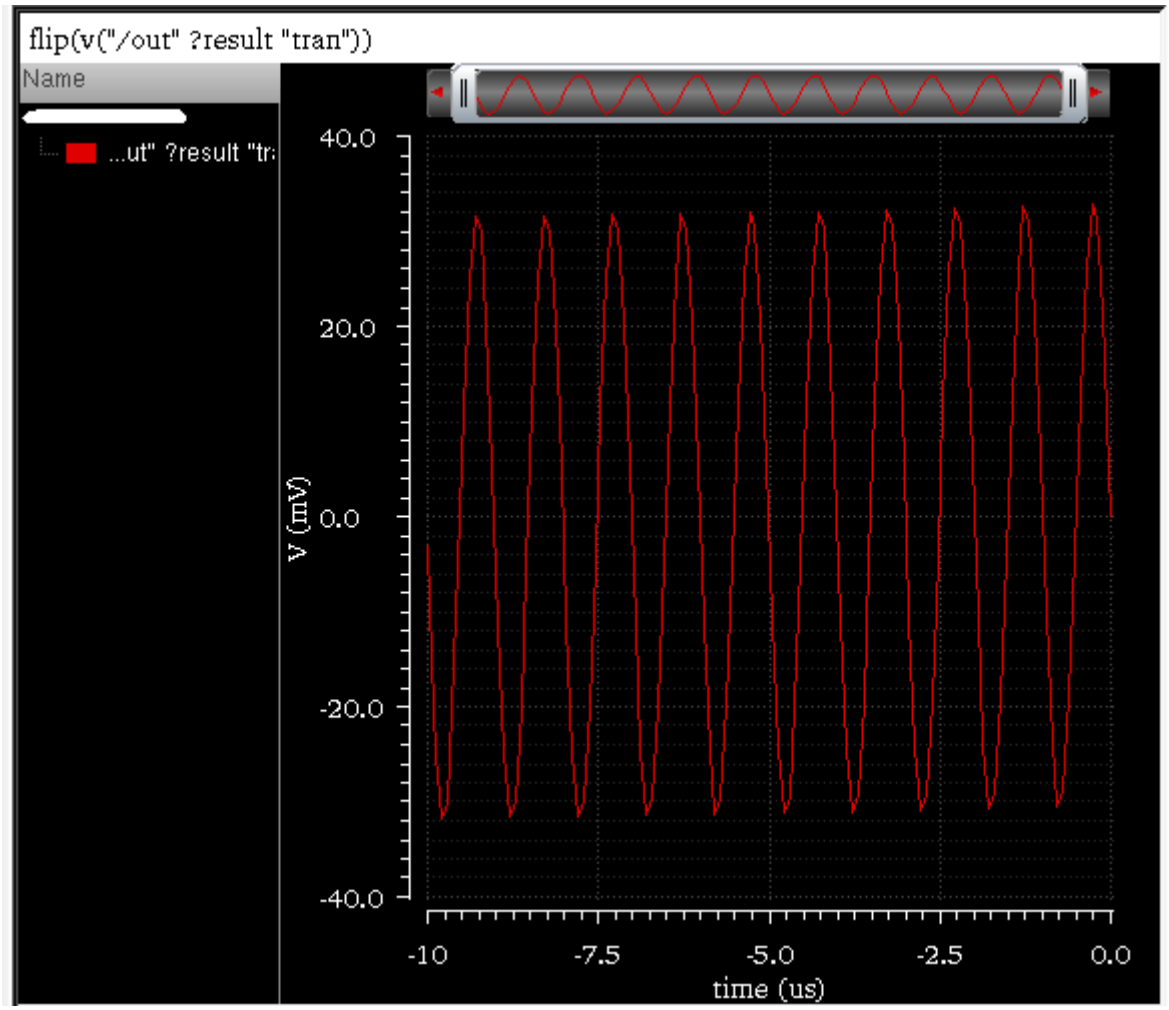
1. In the Calculator, create an expression for the signal, `out`, from `simulation1` results directory. The following expression is displayed: `v("/out" ?result "tran")`.

   Note: `Simulation1` is the in-context results directory selected in the Results Browser.

2. Apply `flip` function to this expression. The expression changes to: `flip(v("/out" ?result "tran"))`

© 2023 Cadence Design Systems, Inc.

3. Now, evaluate this expression and plot the output in a new window. The following output trace is displayed:



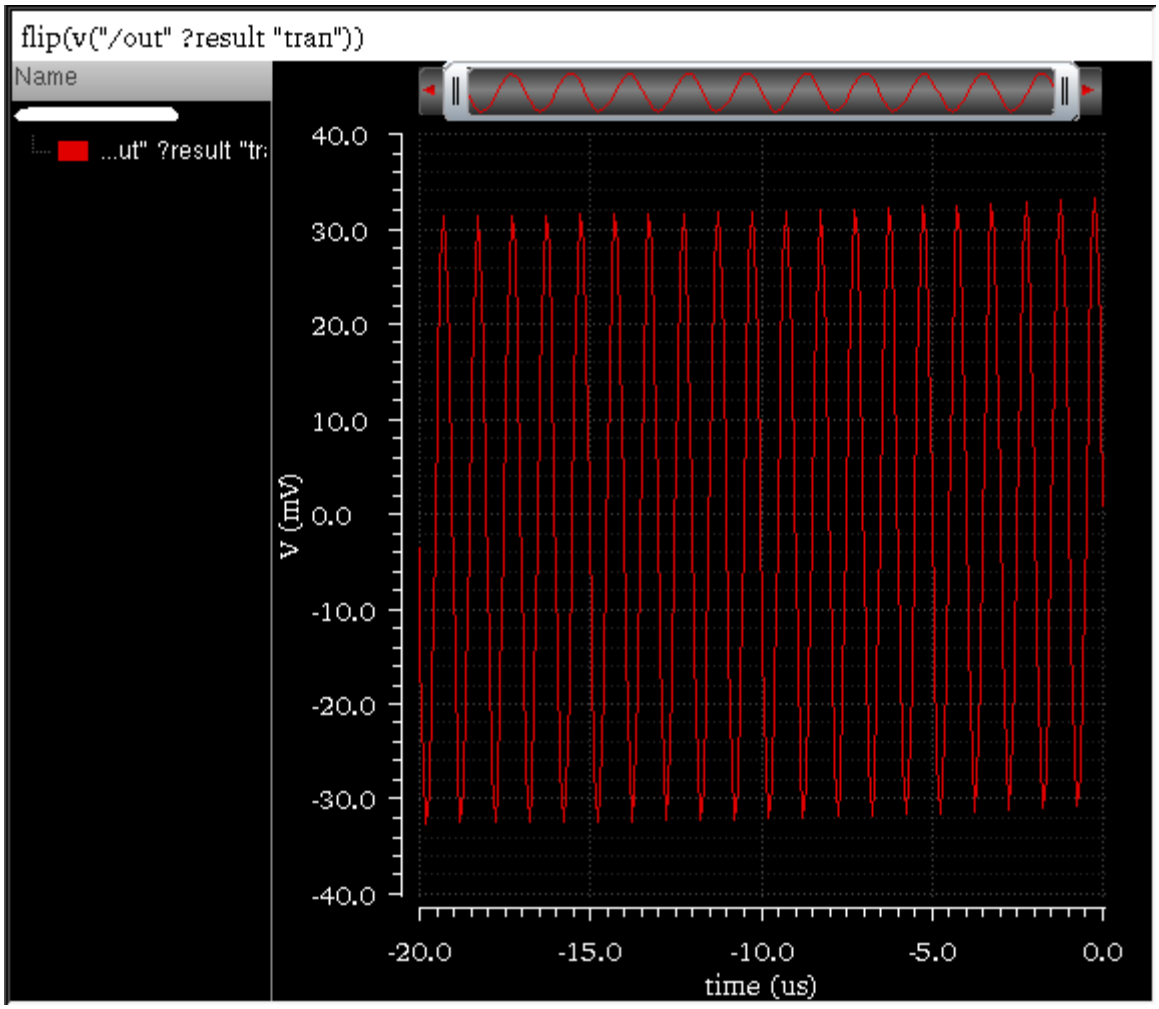
Notice that the time range of the output trace varies from -10 to 0.0 microseconds.

4. In the Results Browser, change the in-context results directory to `simulation2`, which contains the simulation results for the same design.

5. Choose *File – Reload – Current Window*.

© 2023 Cadence Design Systems, Inc.

The trace for the expression you plotted earlier is updated with the data from the new in-context results directory (see figure below). The time range of the updated trace now varies from -20 to 0.0 microseconds.



If you create an expression for a signal from a results directory that is not set as the in-context results directory in Results Browser, the expression for this signal also displays the path of the results directory. For example,

```
v("/out" ?result "tran" ?resultsDir "./simulation_10/ampTest/
spectre/config/psf")
```

This indicates that the database context of this signal is specified within the expression. When you plot the expression, it is always plotted with the specified results directory. The context for this expression does not change with the change in the in-context results directory in the

© 2023 Cadence Design Systems, Inc.

Results Browser; and therefore, when you perform *File – Reload* on this trace, it is not updated with the new data.

Reloading Graph When Opened From Within ADE Explorer, or ADE Assembler

While working in ADE Explorer, or ADE Assembler, you can use the Refresh plotting option to update already open graphs with current simulation results. This option retains all the trace settings that you have applied to the traces in open graphs.

Important points to note:

- Only those open graphs are updated that are common in the current simulation run.
- You can also use *File – Reload* to reload the graphs plotted from within ADE Explorer, or ADE Assembler based on the results from a new in-context results directory selected in the Results Browser. This helps you review graphs across different simulation runs when you change the context. For example, in a test `ACTestBench`, you want to run multiple simulations with varying corners and values of the global variable `VDD`. In the first run, you specify the value `2.2` for the variable `VDD` in corner `C0` and plot the results. After plotting the simulation results in the graph, in the next run, you sweep `VDD` for three values: `1.8`, `2.0`, and `2.4` and add corners `C1`, and `C2`. After simulation, when you choose *Reload* and set the context to a new results directory, the graph window updates to display traces for `VDD` = `1.8`, `2.0`, `2.2` and `2.4`.
- If you rerun the simulation even without changing the corner settings, *Reload* plots traces from all the corners in the graph.

To know more about refreshing graphs through ADE Assembler, see Refreshing Graphs in *Virtuoso ADE Assembler User Guide*.

To know more about refreshing graphs through ADE Explorer, see Refreshing Graphs in *Virtuoso ADE Explorer User Guide*.

Disabling Trace Reload

You can lock the database context of a trace to disable the trace reloading. You use this feature if you do not want to update the trace while reloading graphs with data from the in-context result directory.

Note: The trace for which you lock the database context are not updated when you do *File – Reload.*

© 2023 Cadence Design Systems, Inc.

To disable the reloading of a trace, do one of the following:

- Select the trace and choose *Trace – Disable Reload*.
- Right-click a trace and choose *Disable Reload*.

  A lock appears on the trace color symbol displayed in the subwindow title, which indicates that the context of the selected trace is locked and it is not updated when you reload traces with data from the in-context results directory.

Graph Settings Retained During Reloading

The Virtuoso Visualization and Analysis XL graph window saves and maintains the following settings for the graphs when you reload the graphs with the data from the latest in-context results directory:

- Trace color, type, style, width, or symbols
- Visibility status of graphs
- Axes settings
- Pan and zoom settings
- Graph layout
- Strip layout
- Markers and marker locations

How Marker Locations Change After Graph Reloading?

- Point Marker—If a point marker is attached to a trace and the trace remains plotted after you reload graphs, the point marker stays attached to the trace on the same X-axis value. However, the marker is snapped to the nearest Y-axis value on the trace.

  If a point marker was detached from the trace before reloading, the point markers remain detached and continues to exist at the same XY location.

- AB Marker—If an AB marker is attached to a trace and the trace remains plotted after you reload graphs, the AB marker stays attached to the trace on the same X-axis value; however, the marker is snapped to the nearest Y-axis value on the trace. If the trace is deleted after refreshing, the corresponding AB marker is also deleted. This results in the display of only A and B markers on the graph without a delta line between them.
- Delta Marker—If a delta marker is attached to a trace and the trace remains plotted after you reload graphs, the delta value is updated based on the new point marker locations.

© 2023 Cadence Design Systems, Inc.

If the point markers do not exist after reloading the graph, the corresponding delta marker is also deleted.

Printing Graphs

You can save a selected graph or all the graphs in the selected window to a file in PDF format, and then print the graphs on a network printer that you have installed on your computer.

Before you print, you can use the print preview mode to see how the graph will look after it is printed.

Perform the following steps to print a graph:

- In the window, choose *File – Print*.

  The *Print* form appears.

  In the *Printer* group box, do the following:

  - From the *Name* drop-down list, choose a destination. The destination can be a printer or a PDF file.

  If you select *Print to file (PDF)*, the *Location* field displays *Local file* and the *Type* field displays *Write PDF file.* If you select a printer, the *Location* field displays the

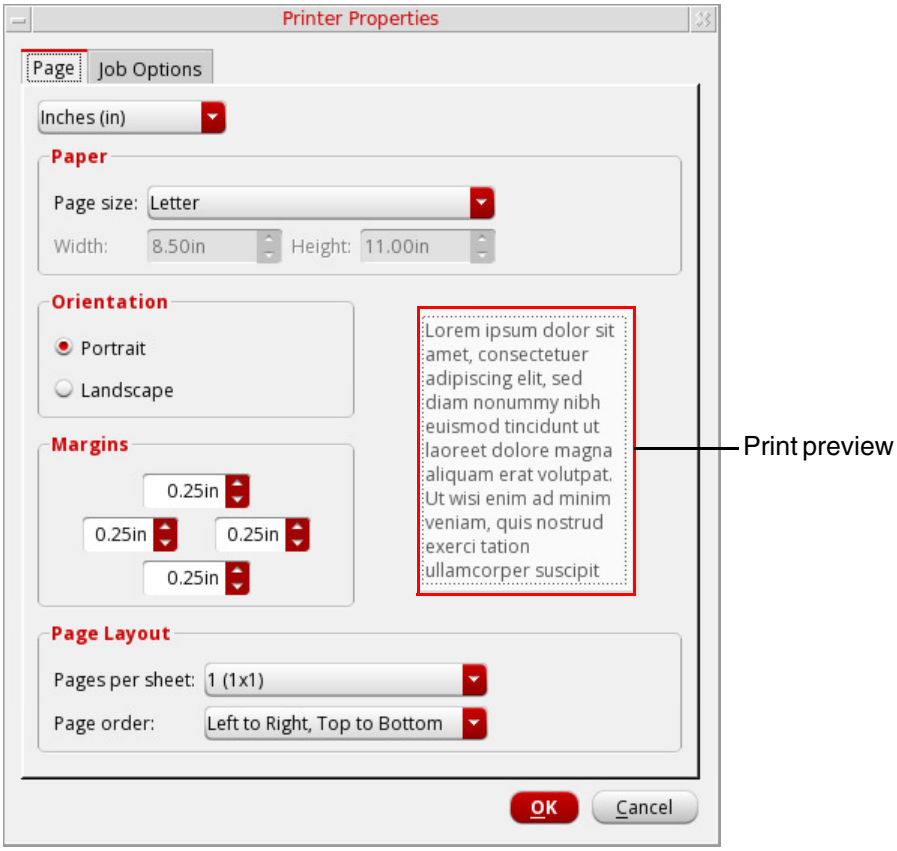
© 2023 Cadence Design Systems, Inc.

location and the *Type* field displays the model number of the selected printer, as shown in the following figure:

- To set properties for the print job, click the *Properties* button next to the *Name* field.

© 2023 Cadence Design Systems, Inc.

The *Printer Properties* form appears. This form has two tabs—*Page* and *Job Options*.



- On the *Page* tab, specify the following fields:
  - Select the units, such as *Inches (in),* or *Millimeters (mm)*, in which you want the paper size to be displayed.
  - In the *Paper* group box, select the required page size in the *Page size* drop-down list box. The *Width* and *Height* fields display the default paper settings for that page size in the specified units.

  Note: The default settings are defined by the `LC_ALL` environment variable on your computer.

  - In the *Orientation* group box, select the print orientation as *Portrait* or *Landscape*.
  - In the *Margin* group box, select the top, left, right, and bottom margins.

© 2023 Cadence Design Systems, Inc.

- In the *Page Layout* group box, select how many pages you want to print per sheet from the *Pages per sheet* drop-down list. You can also control the order in which pages are printed from the *Page order* drop-down list. By default, pages are printed from left to right and then from the top down.

- On the *Job Options* tab, specify the following fields:

  - In the *Job Control* group, from the *Scheduled printing* drop-down list, you can choose when do you want to schedule the printing job. You can also specify the billing information and the job priority in the respective fields.

  - In the *Banner Pages* group box, from the *Start* and *End* drop-down lists, select the level of the information sensitivity that you want to print at the start and the end of the banner page, respectively.

- In the *Output file* field, specify the name and location of the output file. To specify a file at a location other than the default, click the browse button. By default, the name of the output PDF file is `print.pdf`.

- Click the *Options* button to view the other printing options. When you click this button, two tabs are displayed in the *Print* form—C*opies* and *Options.*

  - On the *Copies* tab, do the following:

    - In the *Print range* group box, specify the range of pages that you want to print. By default, all pages are printed.

    - On the *Output Settings* tab, specify the number of copies you want to print. In addition, specify the type in which you want the pages to be printed—*Collate* or *Reverse*.

  - On the *Options* tab, do the following:

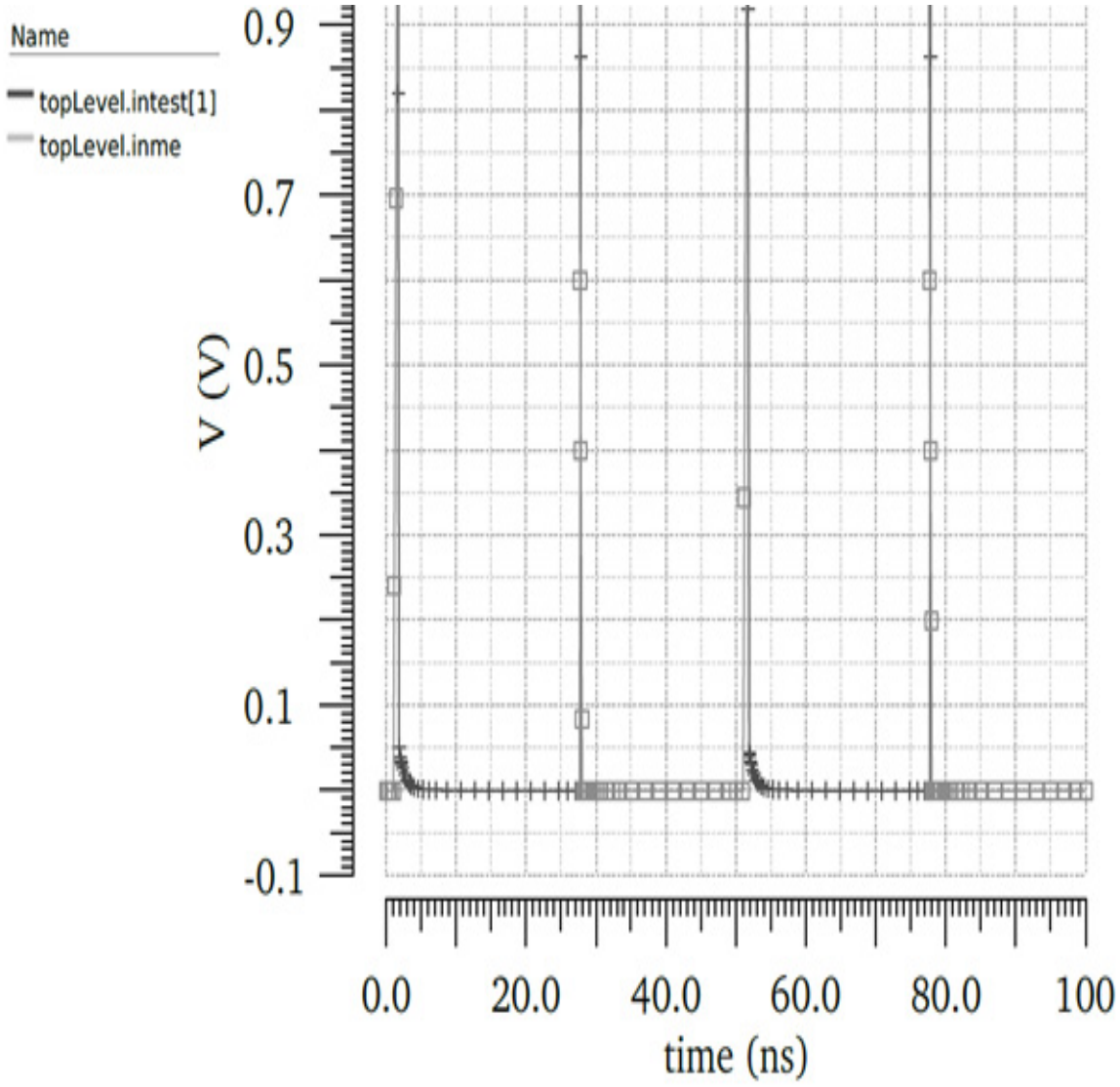
    - Specify *Duplex Printing* as *None, Long side*, or *Short side*.

    - Specify *Color Mode* as *Color* or *Grayscale*.

    Note: When you print in grayscale, you do not have colors (except the shades of gray) by which you can distinguish between different signals. To overcome this, when a graph is printed using the *Grayscale* option, different symbols are automatically added to each signal to help you distinguish between them. For example, see the figure below, where two signals are printed in grayscale. Signal `intest[1]` can be identified by the symbol '+', and the signal `inme` can be

© 2023 Cadence Design Systems, Inc.

identified by the symbol ' '.



- On the *Graph Options* tab, do the following:
  - In the *Graphs/page* section, specify how many graphs you want to print on a page.

    If you want to print all the subwindows in a PDF file in the same order in which they are arranged in the graph, select the *Match window* check box. However, if you do not select this check box, the visibility settings of columns in the trace legend are reflected in the output PDF file. Also, if you adjust the width of the trace legend area and hide few columns, only visible columns are printed in the output file.

© 2023 Cadence Design Systems, Inc.

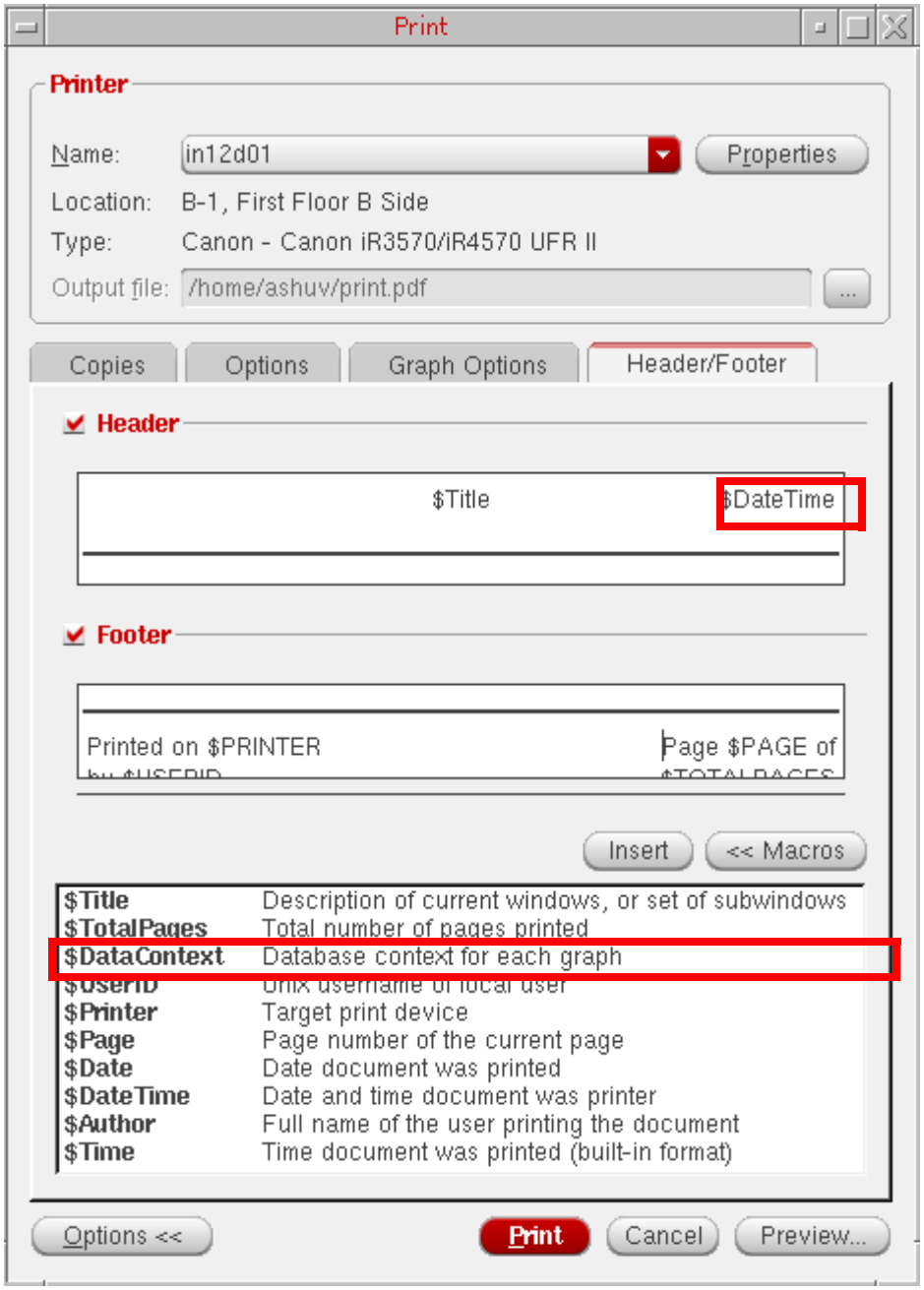
- In the *Strip Options* section, specify which strips you want to print. To print only active strips that are currently visible in the graph, select the *Visible only* option. To print all the strips from all the subwindows, select the *All Strips (multiple pages)* option.The *All Strips* option will print multiple pages.
- Select the *Print Marker Table* check box to print marker tables for the graphs. In addition, specify how you want the marker tables to be printed—*Underneath graph* or *on separate page*.

■ On the *Header/Footer* tab, do the following:

- Click the *Header* and *Footer* check boxes if you want to display header and footer in the output file.
- Specify the information that you want to display in header and footer of the output file. Click the *Macro* button at the bottom of the form, a table listing all the available macros appears. Place the pointer in one of the boxes provided below the *Header* and *Footer* check boxes where you want to insert the macro. Select a macro in the table and click the *Insert* button.

© 2023 Cadence Design Systems, Inc.

The selected macro is inserted at the specified location. For example, if you want to display date and time in the output file header in the right-most corner, you can insert the `$DateTime` macro, as highlighted in the figure below.



© 2023 Cadence Design Systems, Inc.

The table below contains a list of macro that you can specify:

Macro	Description
$Title	Displays the window or subwindow title.
$TotalPages	Displays the total number of pages that are printed.
$DataContext	Displays the results directory for the plotted expressions and signals in the graph. This macro does not display the current in-context results directory in the Results Browser. If the signals from multiple results directories are plotted and then printed, specifying this macro in header or footer prints the name of all the databases that are used for plotting.
$UserID	Displays the user's UNIX login name.
$Printer	Displays the name of the printer.
$Page	Displays the current page number.
$Date	Displays the date when the document is printed.
$DateTime	Displays the date and time when the document is printed.
$Author	Displays the full name user name.
$Time	Displays the time when the document is printed.

- Click *Preview* to generate the print preview.
- Click *Print*.

Note: The upper-right corner of the PDF displays the time and date when the PDF is created.

Supporting Mixed Signals

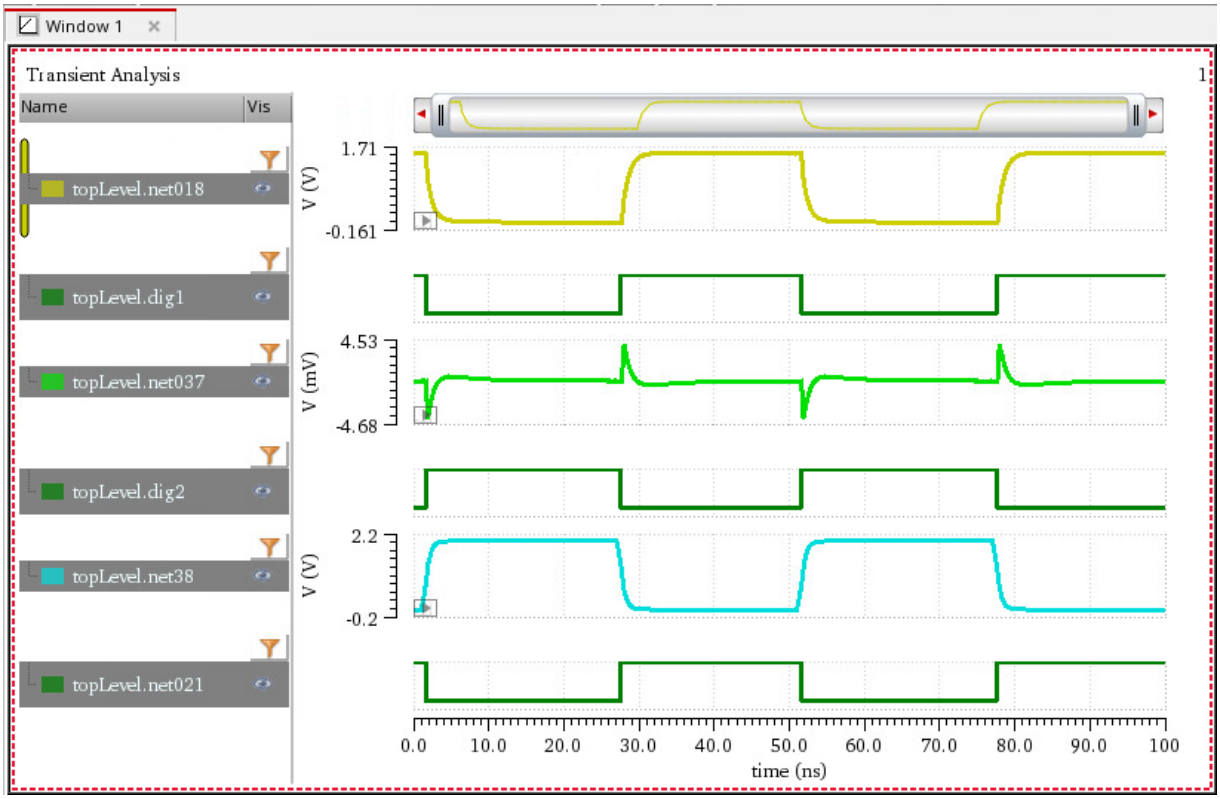
You can use the Virtuoso Visualization and Analysis XL tool to plot analog and digital signals together in one window. Like analog traces, multiple digital traces can also be combined into a single strip. The digital signals are displayed in green by default.

To split the analog and digital signals into different strips, see Working with Strips on page 398.

© 2023 Cadence Design Systems, Inc.

If you want to work with analog and digital signals at the same time, you can plot analog and digital signals in the same window. The signals are plotted in the order they are selected in the Results Browser or ADE.

You can also drag an analog signal to place it in between two digital signals. The analog signal that you drag is displayed in a different strip. You can also impose an analog signal on a digital signal. In this case, the strip height for a digital signal is adjusted according to the height of the analog signal.



You can convert analog signals into digital signals and digital signals into analog signals, if required.

Note: The Verilog and VHDL states are also displayed in specific colors to denote their strength and condition.

© 2023 Cadence Design Systems, Inc.

Loading Vector Files in Virtuoso Visualization and Analysis XL

Most digital simulators store the signal information in text files, such as VCD or VEC files.

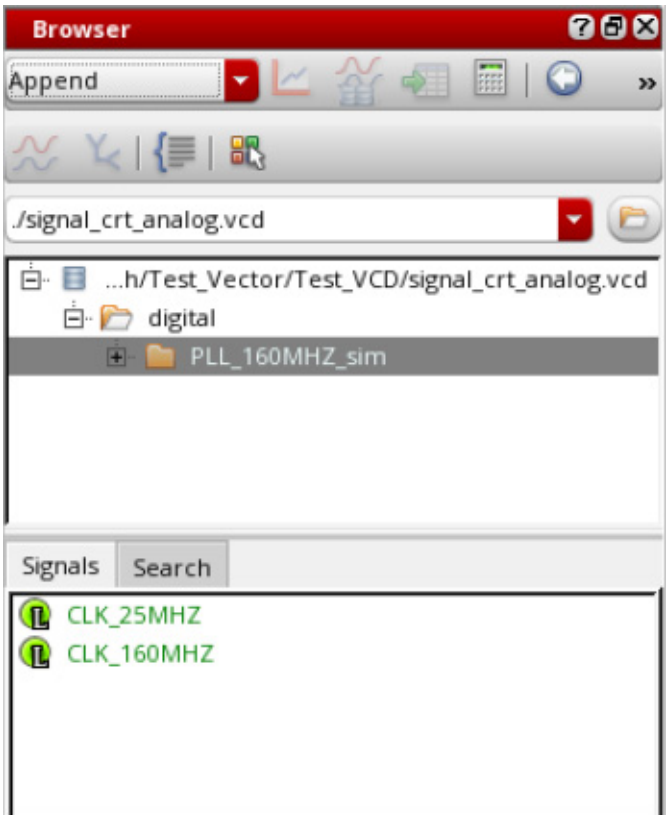
- **Value Change Dump (VCD) File**: VCD files are the most commonly used textual representations for digital signals. They are in ASCII format and contain the value changes for selected signals in the circuit design. VCD files also require information from the .info file, which contains the risetime and falltime and voltage high/low values. VCD files contain signal definitions, timescale data, and waveform data.
- **Digital Vector (VEC) File**: VEC files include a table of times and values at those times. The VEC file can include more than one vector definition. You can directly load the digital stimuli files (VEC) into Virtuoso Visualization and Analysis XL, plot the digital waveforms, and automatically generate the analog waveforms with the stimuli applied.

  Note: You can also read the 'Virtuosity: Reading Vector Files in Virtuoso Visualization and Analysis' blog for more information about reading the vector files.

© 2023 Cadence Design Systems, Inc.

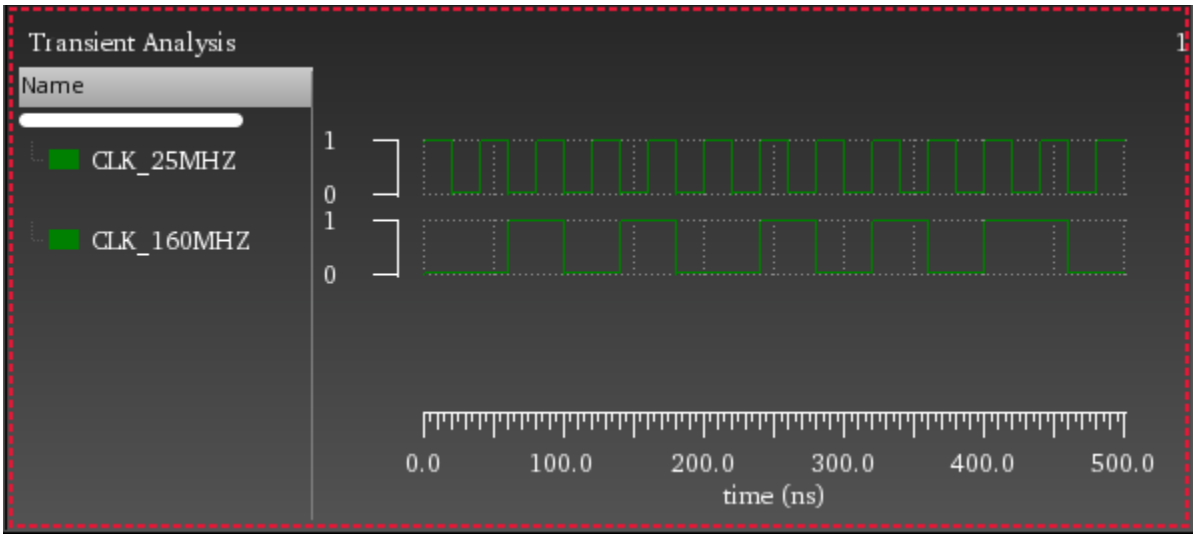
Loading a VCD File

You can load a VCD file in Virtuoso Visualization and Analysis XL using the Select Waveform Database form. The selected file is loaded in the Results Browser and displays a folder, `digital`, as shown in the figure below:



© 2023 Cadence Design Systems, Inc.

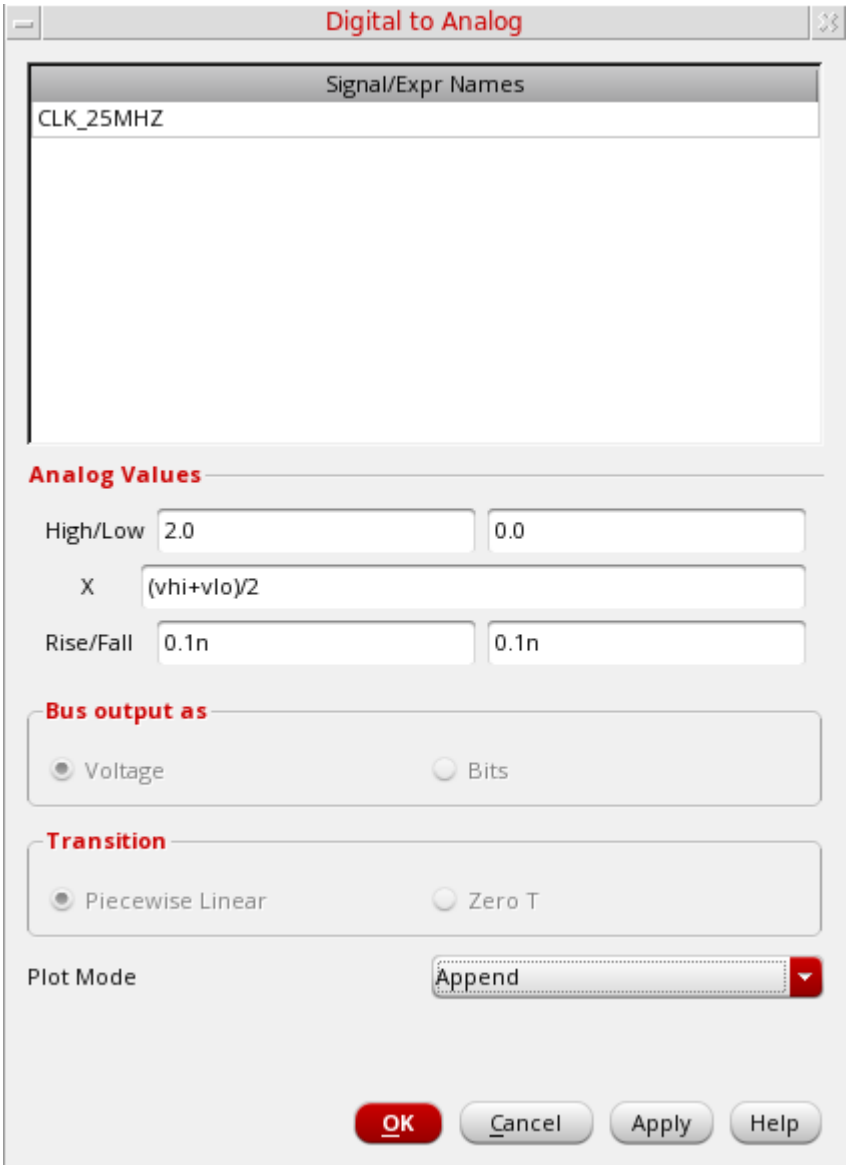
The digital signals are plotted in the Results Browser as shown in the figure below.



To generate analog stimuli for this digital signal, you need to convert the digital signal into analog signal by using the Digital to Analog assistant that appears when you choose

© 2023 Cadence Design Systems, Inc.

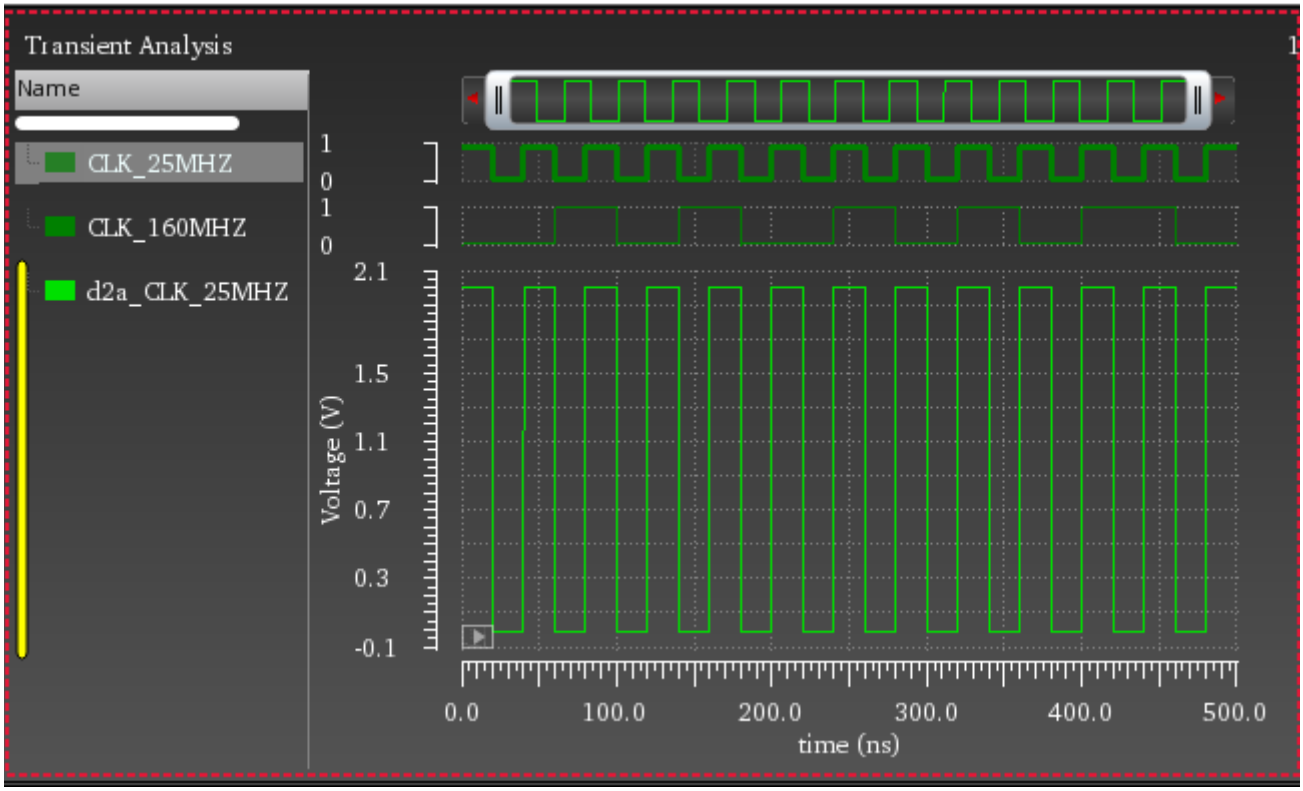
Measurements – Digital to Analog. Specify the form field values as shown in the figure below. For more information, see Converting a Digital Signal to an Analog Signal.



Note: The .info file contains voltages, risetime, and falltime values and the VCD file contains the signal changes and the unit for time.

© 2023 Cadence Design Systems, Inc.

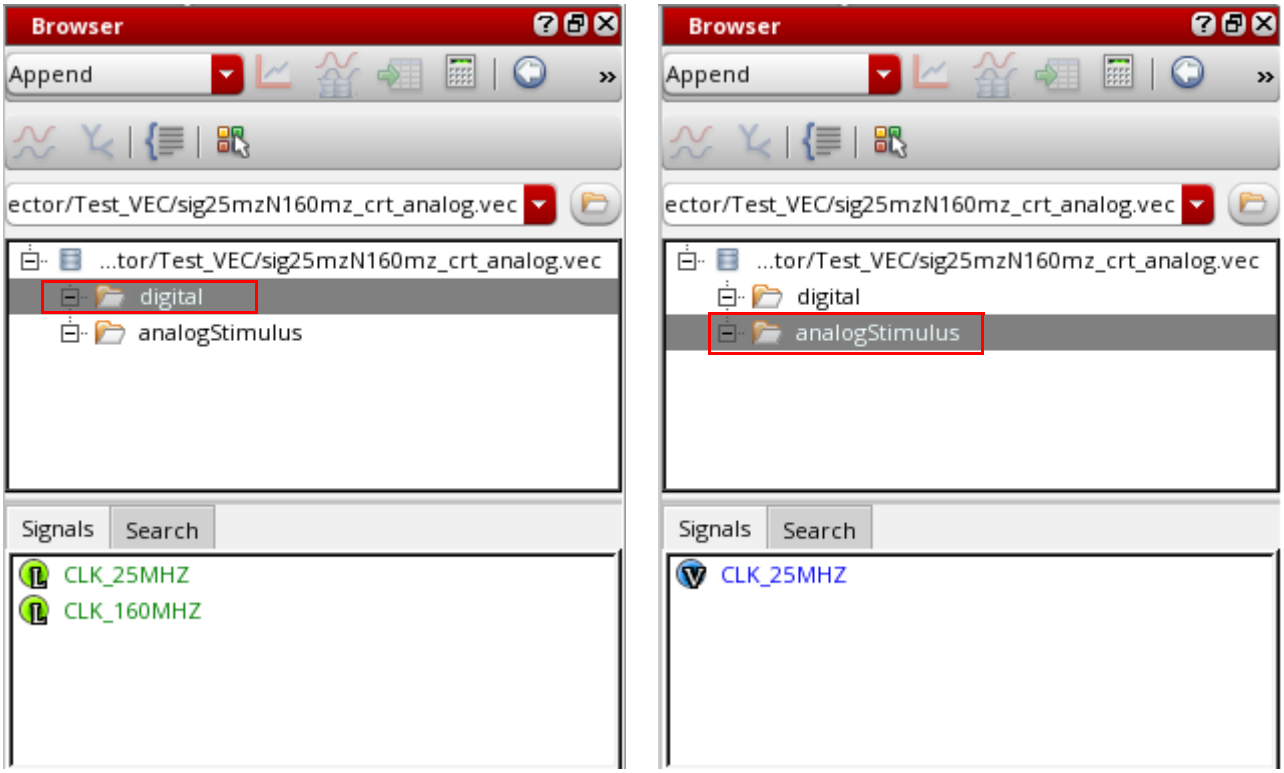
When you click *OK*, the `CLK_25MHZ` digital signal is converted into an analog stimuli signal, `d2a_CLK_25MHZ`.



© 2023 Cadence Design Systems, Inc.

Loading a VEC File

When you load a VEC file in the Results Browser, two folders, `digital` and `analogStimulus` are displayed, as shown in the figure below:



The analog stimuli files are automatically generated when you load a VEC file in Virtuoso Visualization and Analysis XL, whereas for VCD file, you need to perform an additional step to convert the digital signal to generate analog stimuli.

Loading Vector Files Using Command-Line Convertor

You can also use the command-line convertor, `cdsConvertDigitalVector`, to convert a VEC or VCD file into SST2 and PSF XL database that can be loaded into Virtuoso Visualization and Analysis XL.

- Digital waveforms are written to SST2 dataset named `digital`.
- If the VCD file includes variables of type real, they were written to PSF XL dataset named `analog`.
- If the timing characteristics are available, then the digital to analog waveform conversions are written to PSF XL dataset named `analogStimulus`.

© 2023 Cadence Design Systems, Inc.

Example: Converting a VEC File

Run the following command to call the convertor:

```
cdsConvertDigitalVector -i sig25mzN160mz_crt_analog.vec -o
sig25mzN160mz_convert
```

where,

- `sig25mzN160mz_crt_analog.vec` is the name of the VEC file
- `sig25mzN160mz_convert` is the name of the converted output database

Now, when you open this converted database in the Results Browser, the `digital` and `analogStimulus` results directories are displayed. You can then plot the waveforms from these directories.

Example: Converting a VCD and .info File

Run the following command to call the convertor:

```
cdsConvertDigitalVector -i signal_crt_analog.vcd -vcd-info
signal_analog.info -o stimuliDir
```

where,

- `signal_crt_analog.vcd` is the name of the VCD file
- `signal_analog.info` is the info file
- `stimuliDir` is the name of the converted output database

Now, when you open this converted database in the Results Browser, the digital and `analogStimulus` results directories are displayed. You can then plot the waveforms from these directories.

Command-Line Options for File Convertor

The following options are available for the command-line convertor:

Option	Description
`-i [<vector-file>]`	The VEC/VCD/EVCD vector file.
`-o <data-directory>`	The data directory for SST2 and PSF XL files.
`[-vcd-info <info-file>]`	A VCD signal .info file describing timing characteristics.

© 2023 Cadence Design Systems, Inc.

Option	Description
`[-log <log-file>]`	Write errors or warnings to named log file.
`[-expand-bus]`	Expand each bus into individual bits when writing to the SST2 database. By default, buses are not expanded.
`[-bi-signals <in/out>]`	Define bidirectional signals as input (default) or output.
`[-exclude-stimulus-dataset]`	Do not generate the analog stimulus dataset.
`[-report <file>]`	Generate a CSV report file describing each digital vector variable (analog or digital), full path, and direction, <rise>, <fall> ,<vil> , <vih>.

Converting an Analog Signal into a Digital Signal

In the stand-alone SKILL mode, you can create a digital representation of an analog signal.

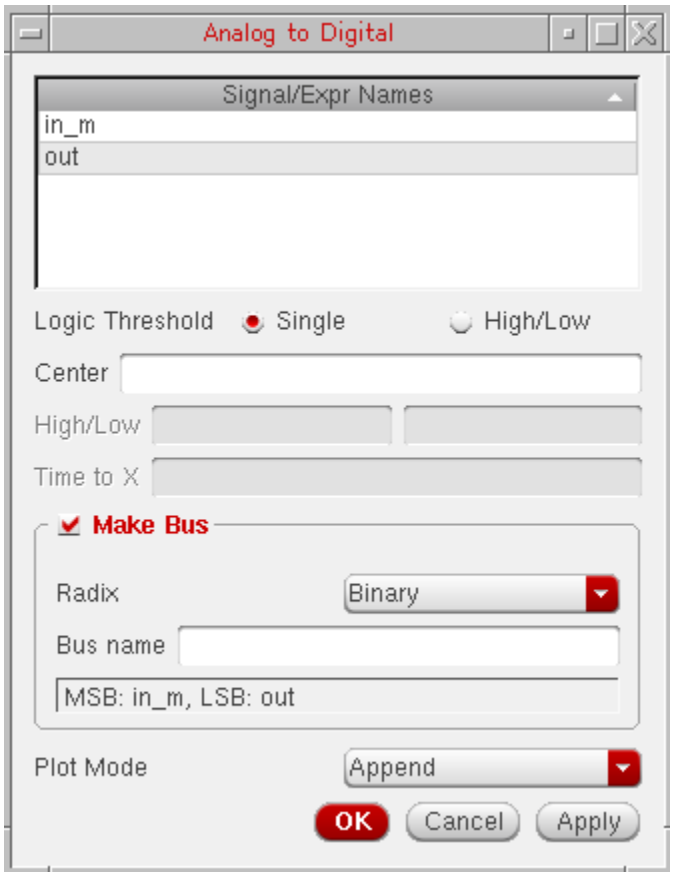
To convert an analog signal into a digital signal, do the following:

1. In the graph, select a trace and choose *Measurements – Analog To Digital*. You can select more than one analog trace at a time.

   The *Analog to Digital* conversion form appears. This form displays the name of the selected analog signals. The signals are displayed on the basis of their selection order;

© 2023 Cadence Design Systems, Inc.

however, you can rearrange the order either by clicking the column header or by using the drag operation. The assistant has the following fields



2. In the *Logic Threshold* field, select *Single* or *High/Low*.

   - If you select *Single*, you need to specify a *Center* value. Analog values equal to or greater than the specified center value are mapped to a digital value of 1. Analog values less than the center value are mapped to a logical value of 0.
   - If you select *High/Low*, you need to specify a high and a low threshold value. All analog values equal to or greater than the high threshold value are mapped to a digital value of 1. All analog values equal to or less than the low threshold value are mapped to a logical value of 0.
     The value *Time to X* puts a time limit on the interval that the signal may remain between the high and low threshold values before the signal is assigned a value of *X*.

3. If you want to make a bus of digital signals from the analog signal, select the *Make Bus* check box.

© 2023 Cadence Design Systems, Inc.

4. Then, select the radix type in the drop-down list and provide a bus name.

   Note: The MSB (topmost signal) and LSB (signal at the bottom) values for the bus are indicated below the *Bus name* field.

5. In the *Plot Mode* drop-down list box, select whether you want to *append* the digital trace to an existing graph, *replace* an existing graph with the digital graph, or add the digital trace to a *new subwindow* or *new window*.

6. Click *OK*.

Converting a Digital Signal to an Analog Signal

In the stand-alone SKILL mode, you can also create an analog representation of a digital signal.

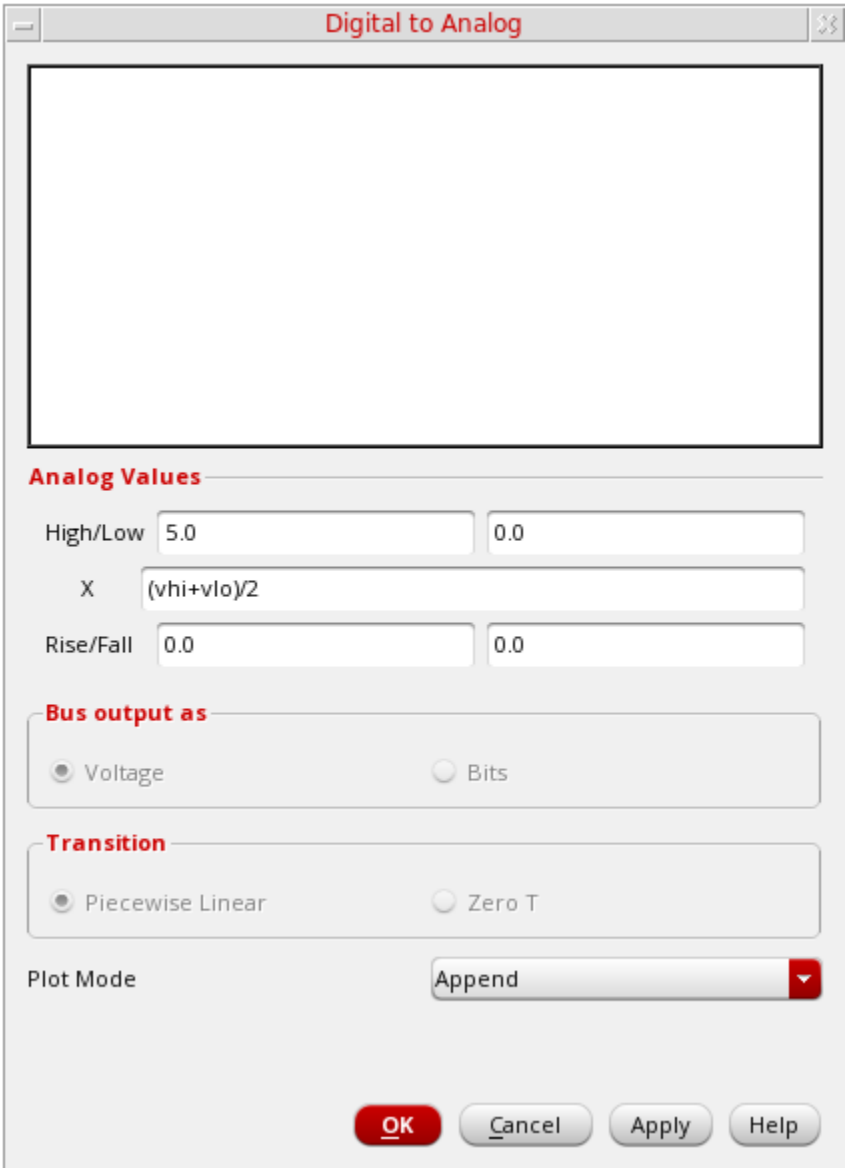
To convert a digital signal into an analog signal, do the following:

1. In the window, select a digital trace and choose *Measurements – Digital To Analog*. You can select more than one digital trace at a time.

   The *Digital to Analog* Conversion form appears. This form displays the digital signals you select. The signals are displayed on the basis of their selection order; however, you

© 2023 Cadence Design Systems, Inc.

can rearrange the order either by clicking the column header or by using the drag operation. The assistant has the following fields:



2. In the *Analog Values section*,

   a. In the *High/Low* field,

      - Specify the high analog value to which the digital value 1 (for single bit waveforms) or the maximum bus value is to be converted to. The default value is controlled by the `dToAHiVoltage` variable in the `.cdsenv` file. For more information, see Appendix A, "Virtuoso Visualization and Analysis XL Tool Environment Variables."

© 2023 Cadence Design Systems, Inc.

- Specify the low analog value to which the digital value 0 (for single bit waveforms) or the minimum bus value is to be converted to. The default value is controlled by the `dToALoVoltage` variable in the `.cdsenv` file. For more information, see Appendix A, "Virtuoso Visualization and Analysis XL Tool Environment Variables."

b. In the *X* field, specify the value to which state *X* of the digital wave is converted to. The *X* value may be given as a:

- Number
- Keyword `vhi`, `vlo`, or `vprev`, where

  `vhi` substitutes the X value with the value in the *Analog Hi Voltage* field.

  `vlo` substitutes the X value with the value in the *Analog Low Voltage* field.

  `vprev` implies that the previous (non-X) state, either `vhi` or `vlo` is used.
- Simple expression, such as `(vhi + vlo)/2`

c. In the *Rise/Fall* fields, specify the risetime and falltime.

3. In the *Bus Output as* group box, select *Voltage* to return the selected bus as a single analog signal or select *Bits* to return the selected bus as a wave list, which is one analog wave for each bus bit. The field is available only if you select a bus.

4. In the *Transition* group box, select *Piece-wise-Linear* to join the points in the analog waveform with straight lines or select *Zero-T* for voltage transitions in zero time.

The *Transition* field is available only if the *Bus Output in* field is selected.

5. In the *Plot Mode* drop-down list box, specify whether you want to *append* the analog trace to an existing graph, *replace* an existing graph with the analog graph, or add the analog trace to a *new subwindow* or *new window*.

6. Click *OK*.

Generating Derived Plots

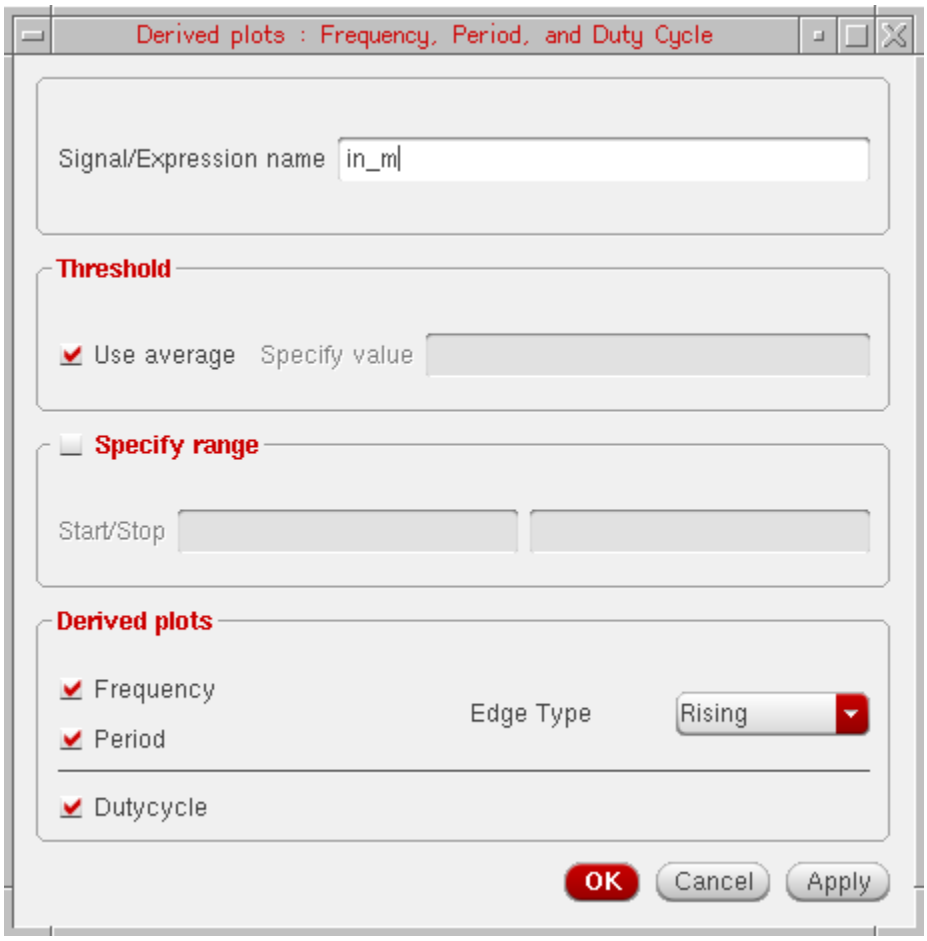
For transient periodic signals, you can generate the derived plots, such as frequency, duty cycle, and period that can be plotted against time.

Do the following to generate a derived plot:

- Choose *Measurements – Derived Plots*.

© 2023 Cadence Design Systems, Inc.

The *Derived plots: Frequency, Period, and Duty Cycle* form appears.



- The *Signal/Expression name* field displays the name of the selected trace or expression in the graph. Select a trace for which you want to generate the derived plot, the selected trace is displayed in this field.
- In the *Threshold* field, select the *Use average* check box if you want to use the average threshold value calculated by Virtuoso Visualization and Analysis XL. Otherwise, specify the threshold value in the *Specify value* field. The *Use average* option is selected by default.
- Select the *Specify range* check box if you want to provide the start and stop values, which indicate the range of the derived plots.
- In the *Derived plots* field, select the derived plot you want to generate, such as *Frequency*, *Period*, and *Dutycycle*.
  - Select *Frequency* to plot a waveform representing the frequency of a signal versus time. For more information about *Frequency*, see frequency on page 1163.

© 2023 Cadence Design Systems, Inc.

- Select *Period* to plot a waveform representing the time period of a signal versus time.
- Select *Dutycycle* to plot a waveform representing the calculated ratio of the time, for which signal remains high, to the period of the signal. For more information about *Dutycycle*, see dutyCycle on page 1093.

- Select the *Edge Type* as `Falling` or `Rising` for Frequency and Period.
- Click *Apply* and then click *OK*.

  The selected derived plots are plotted in different strips in the same subwindow.

Limitations

- Derived plots are currently not supported for family data.
- Derived plots are supported only for periodic signals and if you try to generate a derived plot for the non-periodic signals or buses, an error is displayed in CIW.

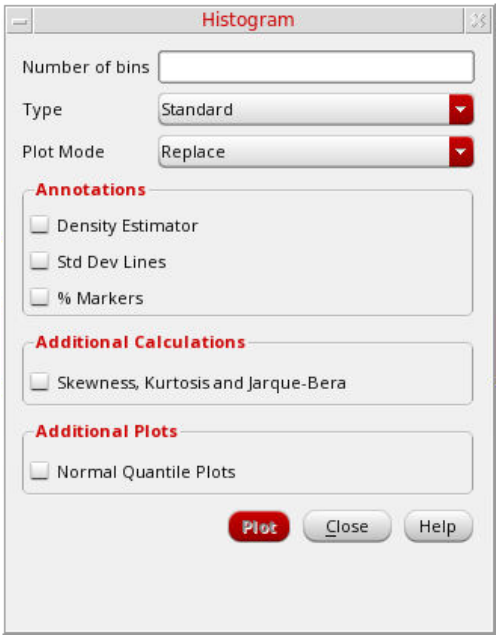
Plotting Histogram

To generate the histogram plot directly on a graph, do one of the following:

- Select the trace for which you want to create a histogram plot and choose *Measurement – Histogram*.
- Right-click the trace and choose *Measurement – Histogram*.

© 2023 Cadence Design Systems, Inc.

The *Histogram* form appears.



You can specify the following fields in this form:

- *Number of Bins*—Specify the number of bars to plot in histogram. The default value is `10`.
- *Type*—Select the histogram type. You can select one of the following:
  - *Standard*
  - *Cumulative Box*
- *Plot Mode*—Specify the plotting mode. You can specify the following plot modes:
  - *Replace*—This plotting mode is selected by default.
  - *New Subwindow*
  - *New Window*
- In the *Annotations* section, you can select the following annotations that you want to display on the histogram plot:
  - *Density Estimator*—Plots a curve that estimates the distribution concentration.

    Note: The Gaussian Kernel Density Estimator is used to calculate the density estimation in histogram plots.

© 2023 Cadence Design Systems, Inc.

- *Std Dev Lines*—Shows the standard deviation lines in the graph indicating the mean, (mean - standard deviation), and (mean + standard deviation) values. The standard deviation is a sample standard deviation.
- *% Markers*—Displays the markers associated with the histogram trace.

- In the *Additional Calculations* section, select *Skewness, Kurtosis and Jarque-Bera* to show these calculations in the graph label of the histogram trace.

  Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or data set, is symmetric if it looks the same to the left and right of the center point.

  Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. That is, data sets with high kurtosis tend to have heavy tails, or outliers. Data sets with low kurtosis tend to have light tails, or lack of outliers. A uniform distribution would be the extreme case.

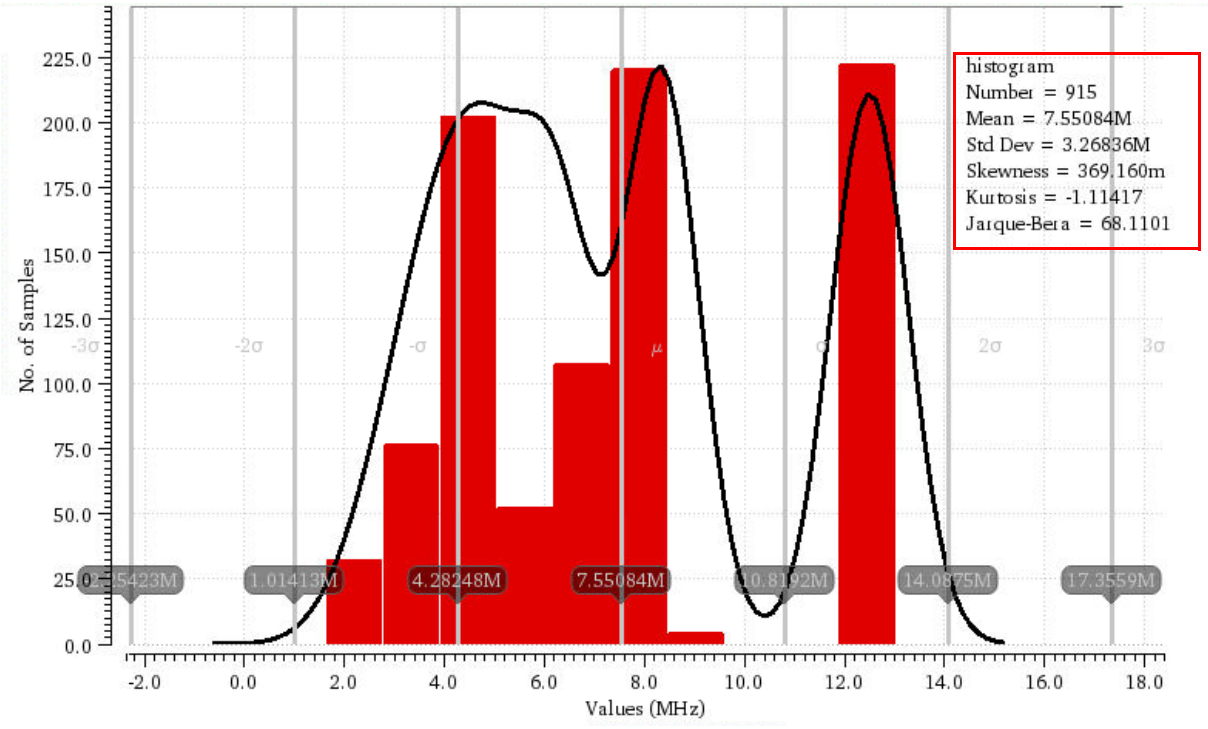
- In the *Additional Plots* section, select the following field:
  - *Normal Quantile Plots*—Generates the quantile plots.

    A normal quantile plot is an indication of how close the data is to being normally distributed. The closer the plot is to a straight line, the closer it is to being normally distributed.

- Click the *Plot* button.

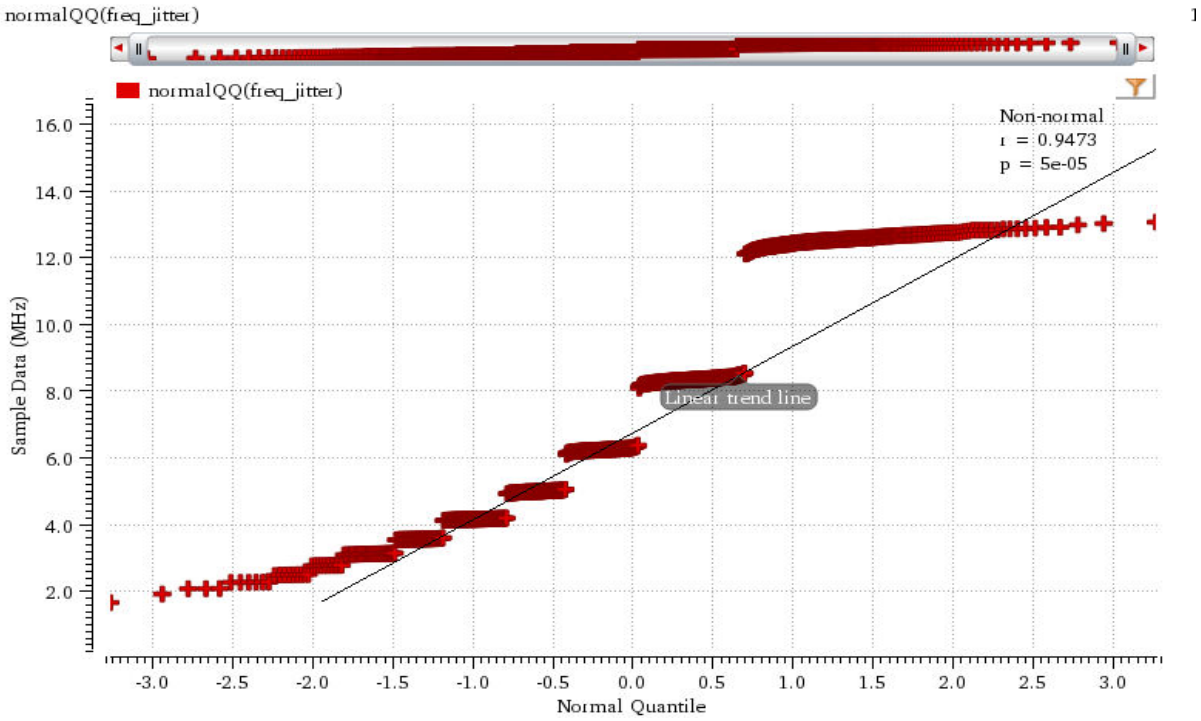
© 2023 Cadence Design Systems, Inc.

The histogram plot is generated with the specified properties. The figure below displays a histogram plot with *Density Estimator*, *Std Dev Lines,* and *Additional Calculations* fields enabled.



© 2023 Cadence Design Systems, Inc.

The figure below displays a normal quantile plot:

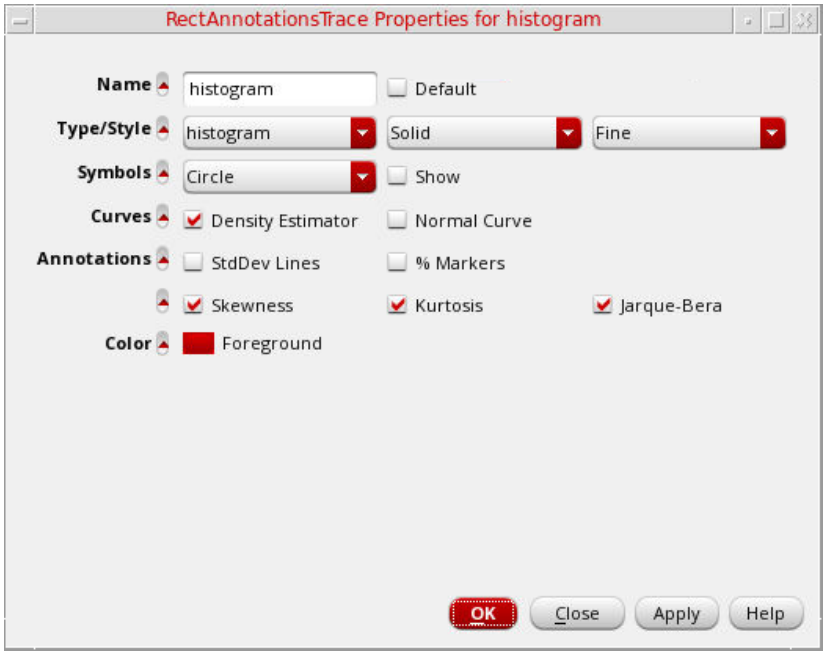


Changing Histogram Trace Properties

To change the properties of the histogram plot, right-click the histogram trace and choose *Trace Properties*.

© 2023 Cadence Design Systems, Inc.

The *RectAnnotationsTrace Properties for <histogram_trace_name>* form appears.



Specify the following fields in this form:

1. In the *Name* field, type the name for the trace or select the *Default* check box to display the default trace name. When you select the *Default* check box, the *Name* field becomes unavailable. The *Name* and *Default* fields are not available if you select more than one trace.
2. In the *Type/Style* fields, do the following:
   - Specify whether you want to represent the trace by a *line*, *points*, *histogram*, *bar*, *spectral,* or *sampleHold*.
   - Specify whether you want the trace style to be *Solid*, *Dashed*, *Dotted*, or *DotDashed,* or *DashDotDot*.

     Note: The trace style option does not work for *Bars* and *Spectrum*.
   - Specify whether you want the trace to be *Fine*, *Medium*, *Thick* or *ExtraThick*.

     Note: The trace thickness option does not work for *Bars* and *Spectrum*.
   - In the *Symbols* field, select the *Show* check box to display data points on the trace and select the symbol type from the drop-.down list.
   - In the *Annotations* field, select the following check boxes to display annotations:

© 2023 Cadence Design Systems, Inc.

- *Density Estimator*— Select to display density estimator curve.
- *StdDev Lines*— Select to display standard deviation lines.
- *% Markers*— Select to display markers.
- *Skewness*— Select to display skewness calculation in the graph label of the histogram trace.
- *Kurtosis*— Select to display kurtosis calculation in the graph label of the histogram trace.
- *Jarque-Bera*— Select to display normality calculation based on Jarque-Bera test in the graph label of the histogram trace.

3. In the *Color* field, select the foreground color for the trace. Alternatively, you can also set the trace color by right-clicking the trace and selecting *Color.*

Working with Buses

A group of digital signals can be converted to a create a bus. You can expand a bus to view its component signals.

If you want to create a bus of analog signals, you need to convert the analog signals to the corresponding digital signals. For information about how to convert an analog signal to the corresponding digital signal, see Converting an Analog Signal into a Digital Signal on page 465. After the conversion is complete, you can create bus from the digital signals.

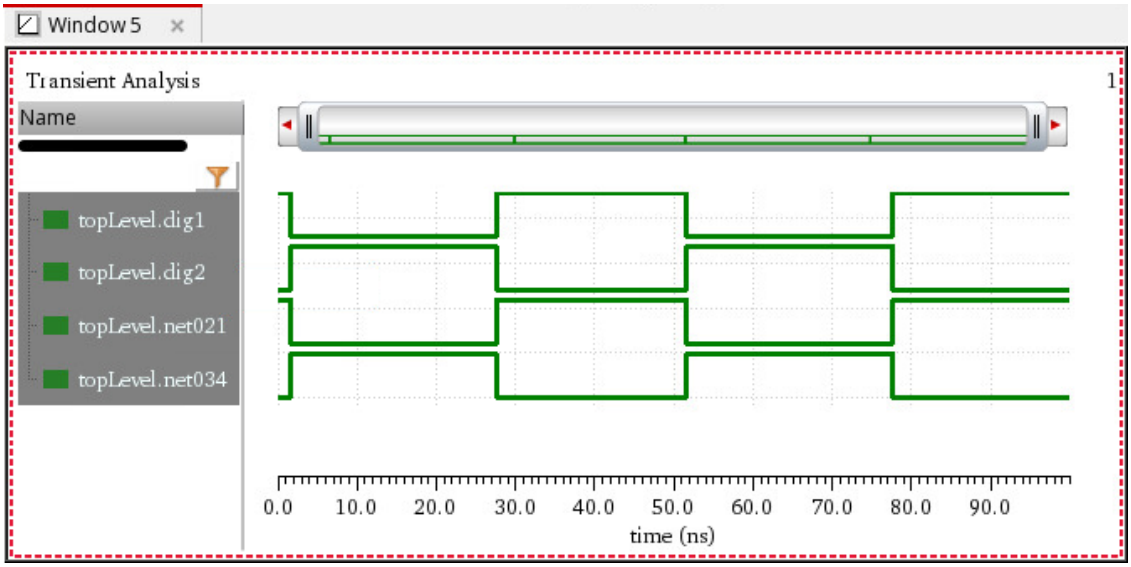
To combine the signal conversion and bus creation processes in a single step, you can select the *Make Bus* check box in the *Analog to Digital* conversion form, and specify a bus name while converting the analog signal to a digital signal.

Creating a Bus

To create a bus:

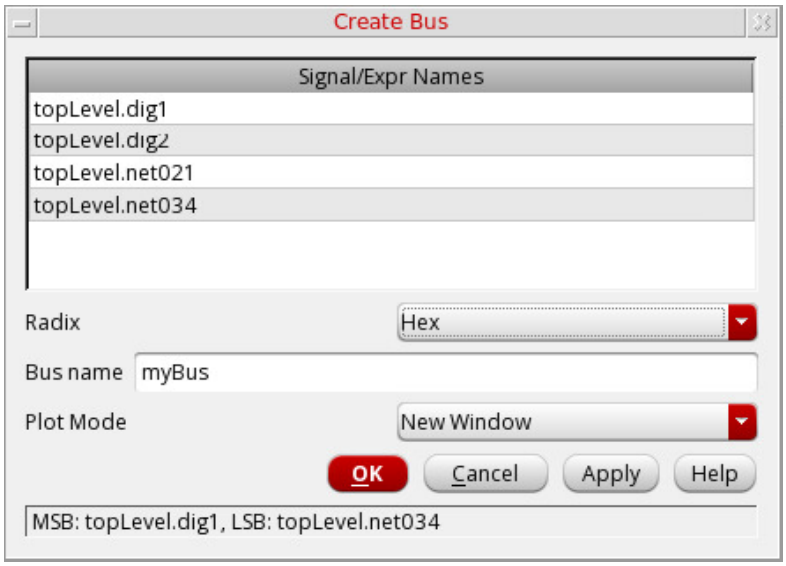
© 2023 Cadence Design Systems, Inc.

1. In a graph window, select the digital traces that you want to use to create a bus.



2. Choose *Trace – Bus – Create.*

   The *Create Bus* form appears.



   The *Signal/Expr Names* section in this form displays the selected traces, which you use to create the bus, in the order of significance (top to bottom—from the least significant bit (LSB) to the most significant bit (MSB)). However, you can rearrange the trace order either by clicking the column header or by using the drag operation.

3. Now, in the *Create Bus* form, do the following:

© 2023 Cadence Design Systems, Inc.

a. Type a name for the bus in the *Bus Name* field.

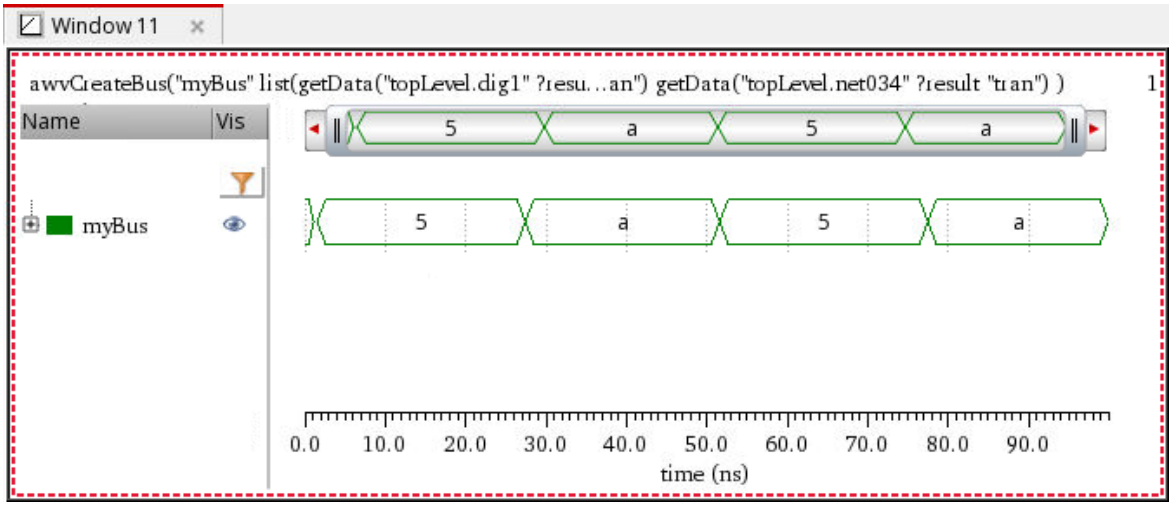
b. Select the *Radix* for the bus, such as `Ascii`, `Binary`, `Hex`. Alternatively, to change the radix type after a bus is created, right-click the bus and choose *Radix*.

c. Specify whether you want the bus to be appended to the digital traces in the graph or to replace the selected traces in the *Plot Mode* field.

d. Click *OK*.

Note: The MSB and LSB values are indicated at the end of the *Create Bus* form.

The bus is created from the selected digital traces.



There are three states in which a signal can exist in the bus—Hi, Lo, and XZ, where X is a transition from Lo to Hi and Z is a transition from Hi to Lo. Following is the color pattern for the traces in a bus that belong to a particular state:

- If the bus is all Z values, it is displayed in orange.
- If the bus is all X values, it is displayed in red color rectangles.
- if the bus includes both X and Z values, it is displayed in yellow colored rectangles.

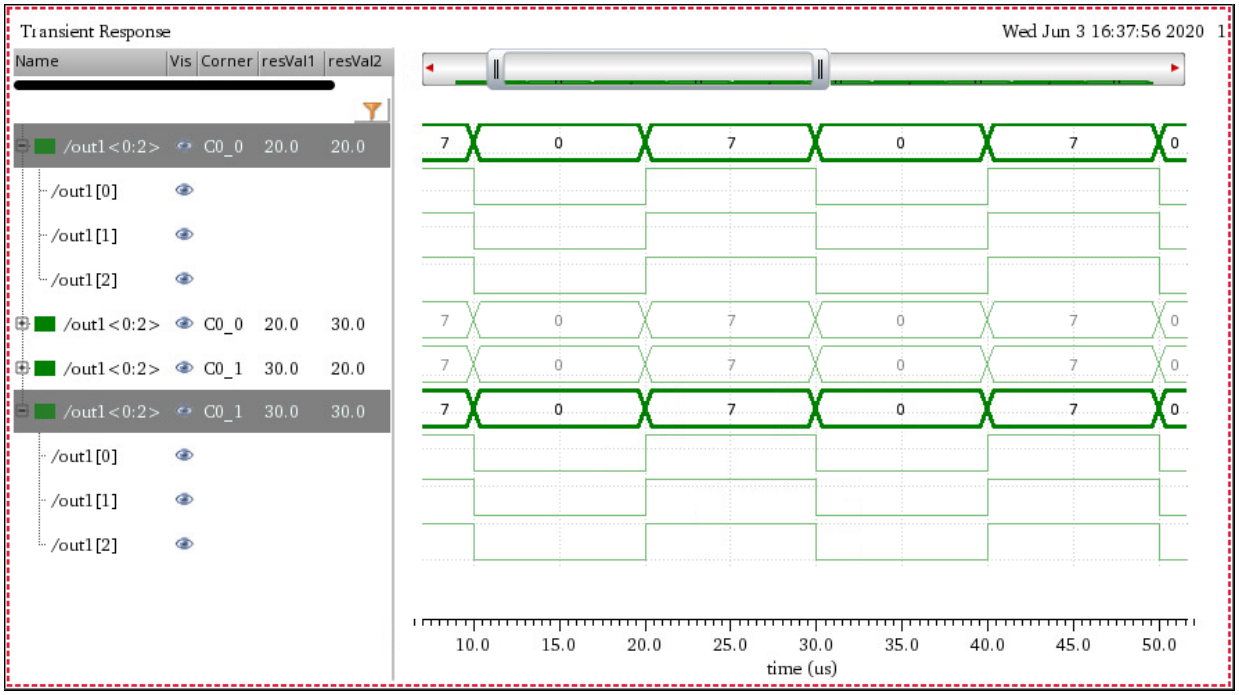
If you want to create a bus of analog signals, convert them to digital signals by using the Analog to Digital assistant. Then, follow the above mentioned steps to create the bus.

© 2023 Cadence Design Systems, Inc.

Plotting Parametric Buses

You can plot parametric digital buses or bus bundles, which are families of buses containing sweeps or corners, using the Results Browser or CIW. The digital buses can also be plotted from ADE when you right-click the signal icon in the *Results* tab and choose *Plot* or *Plot Across Corners*.

The figure below displays a parametric bus, `/out1<0:2>`, where all leaf waveforms (for each sweep or corner) are plotted in a single strip. Analog buses are always plotted in a tree structure with the root displayed at the top, and when you expand the root, the leaf waveforms appear in the legend and graph.



Notice that the legend of a digital parametric bus lists the corner and design variables. When you send the bus waveforms from the Calculator or graph to ADE, the root expression is generated and displayed in the Outputs Setup pane of ADE.

You can also perform calculations on bus waveforms using the following Calculator functions:

- numConv to convert bus value strings to other formats
- busTransition to find the time when a digital bus changes from one value to another
- value to find the value of a bus at a given time
- cross to find the time when a digital bus value is equal to a specified number

© 2023 Cadence Design Systems, Inc.

- average to calculate the average of a bus waveform
- clip to clip the bus waveform from time `t1` to `t2`
- frequency to calculate the frequency of a bus waveform.

Note: You can also set the useD2A environment variable to determine how the calculation results are represented for digital buses and whether output results are to be converted from digital to analog.

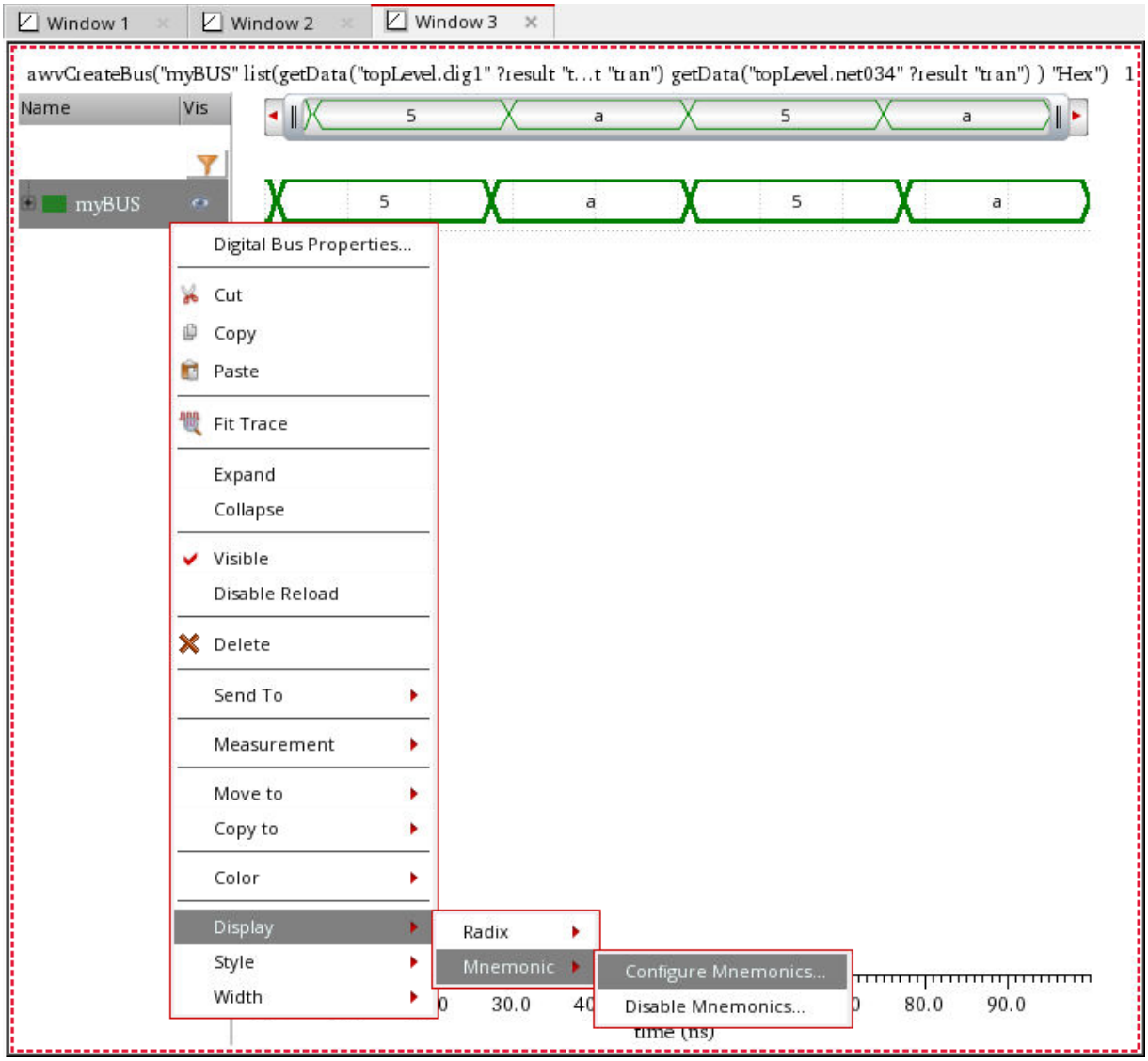
© 2023 Cadence Design Systems, Inc.

Creating a Mnemonic Map

You can create a mnemonic map and add rules to set up how different bit patterns and values of a bus are mapped with specific colors and formats by using the Configure Mnemonics form.

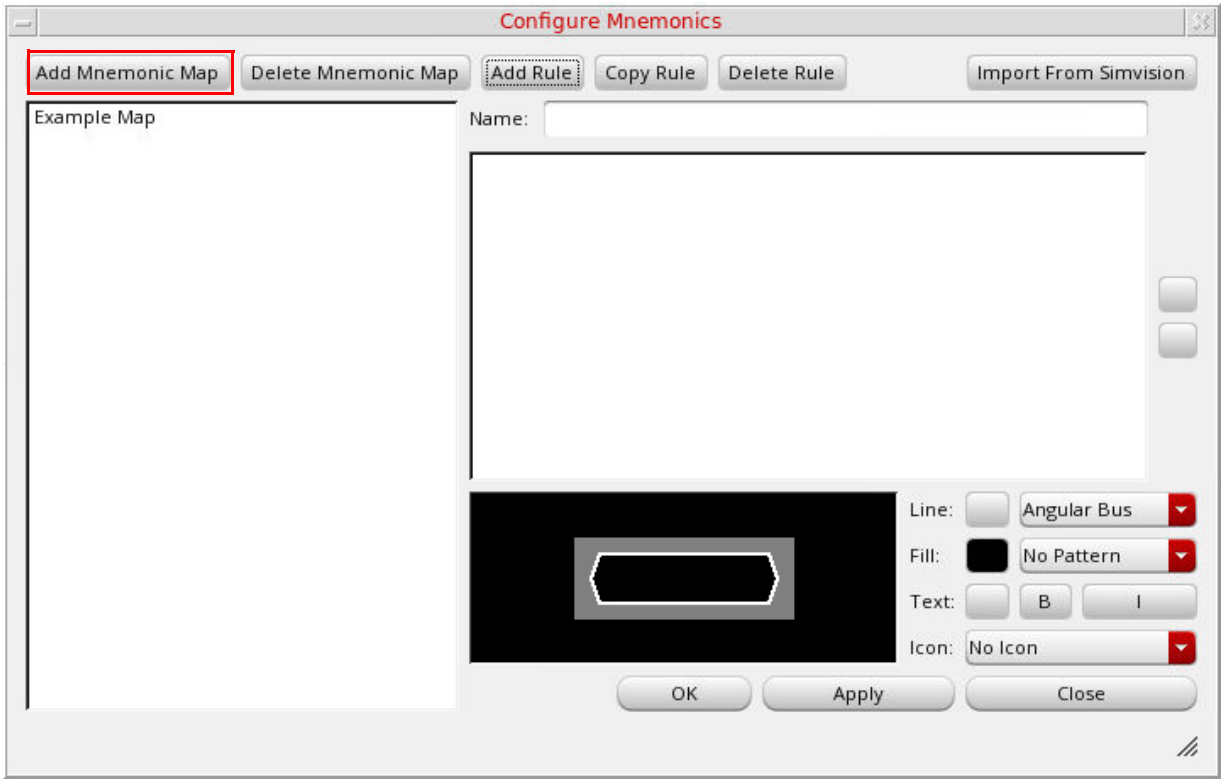
To create a mnemonic map:

1. In the trace legend area, right-click a bus and then choose *Display – Mnemonic – Configure Mnemonics*.



© 2023 Cadence Design Systems, Inc.

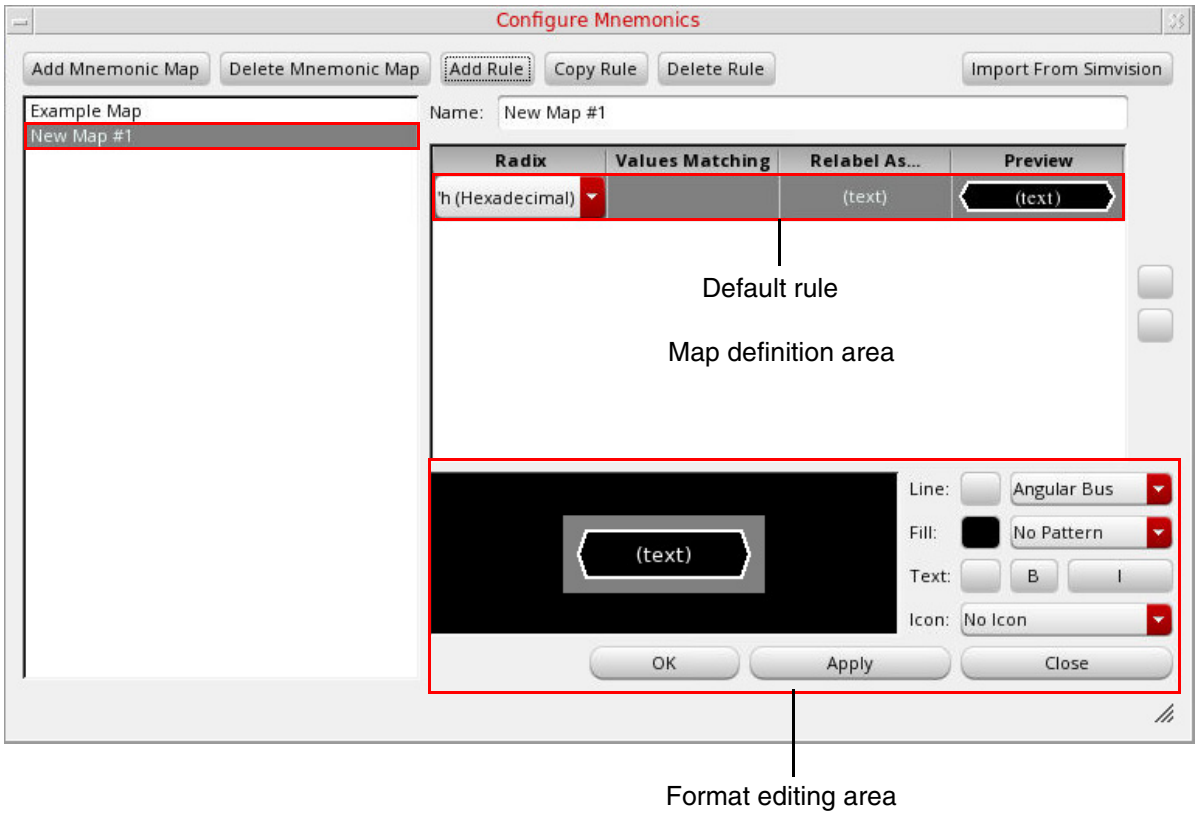
The *Configure Mnemonics* form opens.



2. Click *Add Mnemonic Map* to create a mnemonic map.

© 2023 Cadence Design Systems, Inc.

A mnemonic map is created and displayed in the left pane of the form. By default, mnemonic maps are named as `New map #1`, `New map #2`, `New map #3`, and so on. You can assign a different name to the map by editing the *Name* field.



A default rule is also added to the mnemonic map.

3. Edit the default rule by specifying the following information:

   a. In the *Radix* column select a radix. For example, `'h (Hexadecimal)`.

   b. In the *Values Matching* column, enter a value that you want to map in the format of the selected radix. For example, `5`.

   c. In the *Relabel As* field, enter the ASCII string you want to associate with the value. For example, `Power`.

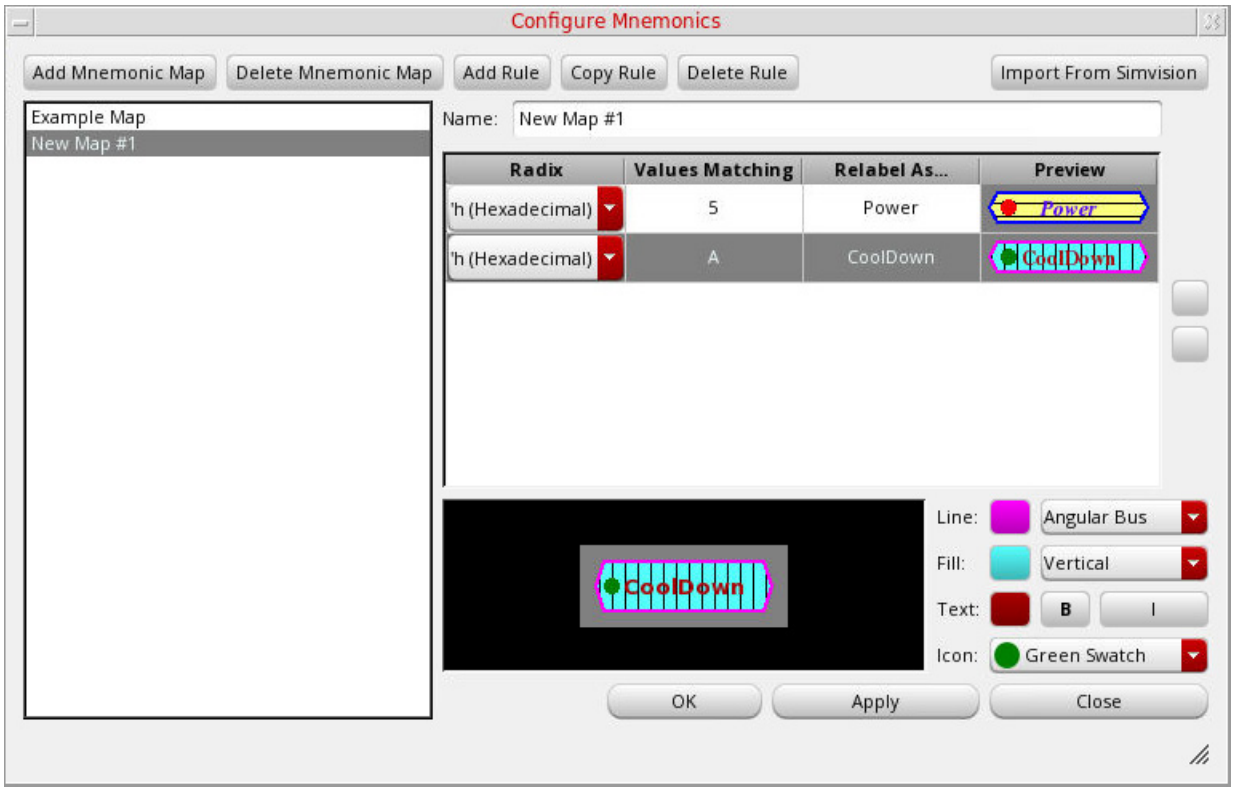
   d. The *Preview* field shows how the specified bit pattern or value will look like in the bus. You can change its appearance by using the format editing area below the mnemonic map definition, as follows:

      - *Line*: Specify the color and shape of the lines drawn around the value.

© 2023 Cadence Design Systems, Inc.

- *Fill*: Specify the color and pattern of the filled area within the lines.
- *Text*: Specify the color and font style of the mnemonic text.
- *Icon*: Specifies the icon to display to the left of the mnemonic text.

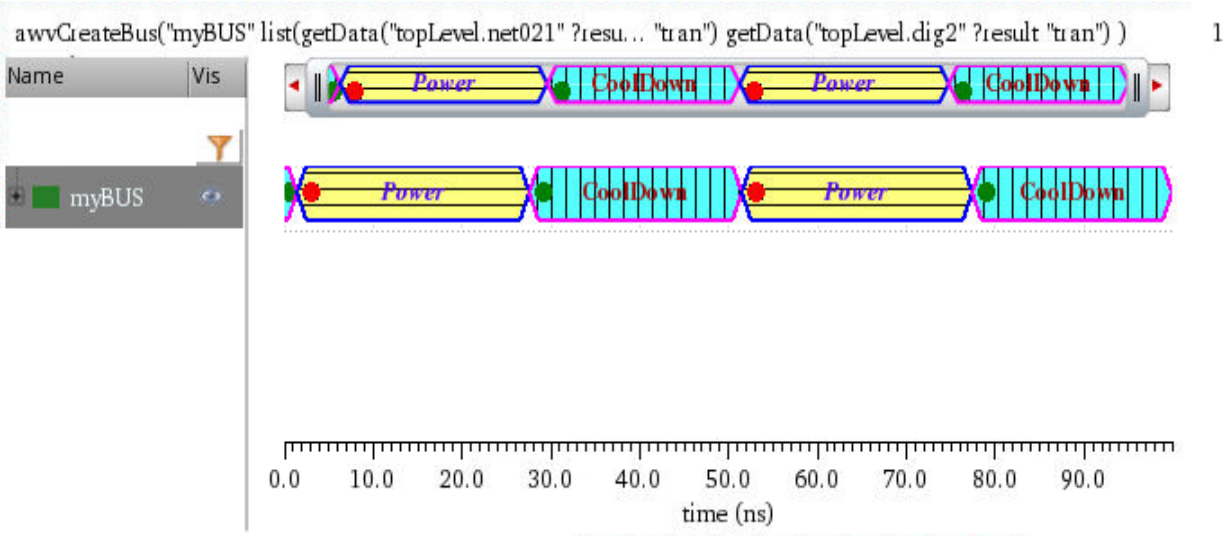
4. Click *Add Rule* to add another rule. Edit the rule by specifying the *Radix*, *Values Matching*, *Relabel As*, *Line*, *Fill*, *Text*, and *Icon* fields.



5. Click *Save Rules* to save the mnemonic map.
6. Right-click the bus and then choose *Display* – *Mnemonic* – `Name of the Mnemonic Map` to apply the mnemonic map to the bus.

© 2023 Cadence Design Systems, Inc.

The mnemonics are applied to the bit patterns or values according to the rules defined in the mnemonic map.



Video

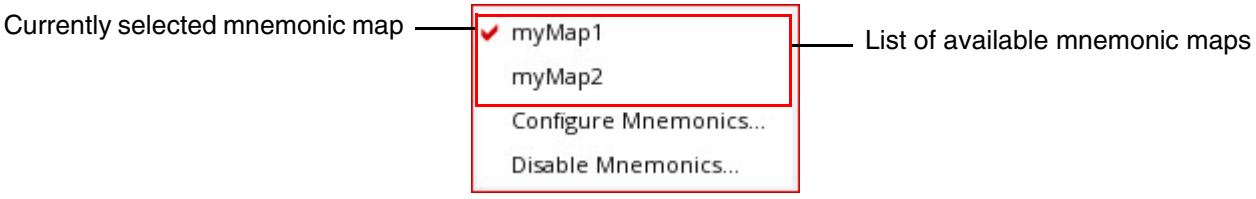
You can watch a video demonstration on how to work with mnemonic maps at Usability Enhancements in Digital Signals. You can also read the related blog at Virtuoso Video Diary: Usability Enhancements in Digital Signals.

Applying a Mnemonic Map to a Bus

To apply a mnemonic map to a bus:

1. In the trace legend area, right-click a bus and then choose *Display – Mnemonic* to open the context menu command.

   A list of available mnemonic maps is displayed.



2. Select the mnemonic map that you want to apply to the bus.

   The mnemonics are applied to the bit patterns or values according to the rules defined in the mnemonic map.

© 2023 Cadence Design Systems, Inc.

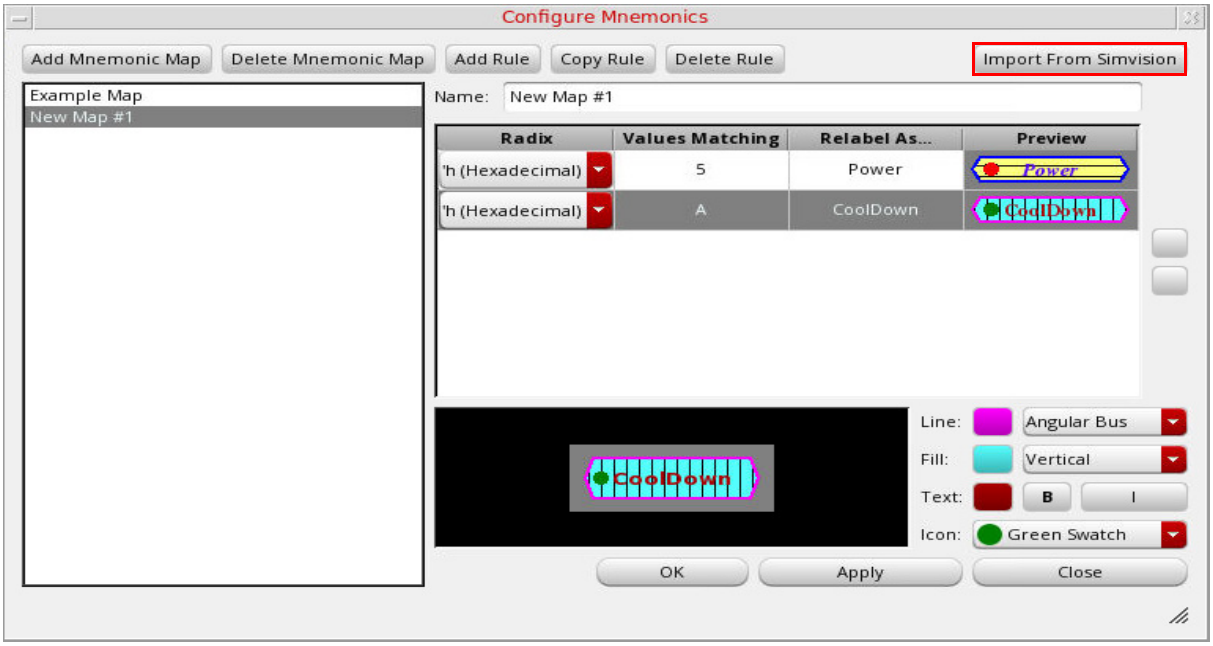
Note: To disable the mnemonic map, right-click the bus and choose *Display – Mnemonic – Disable Mnemonics*.

Importing Mnemonic Maps from SimVision

If you have created a mnemonic map in SimVision, you can import it using the *Import From SimVision* command on the Configure Mnemonics form.

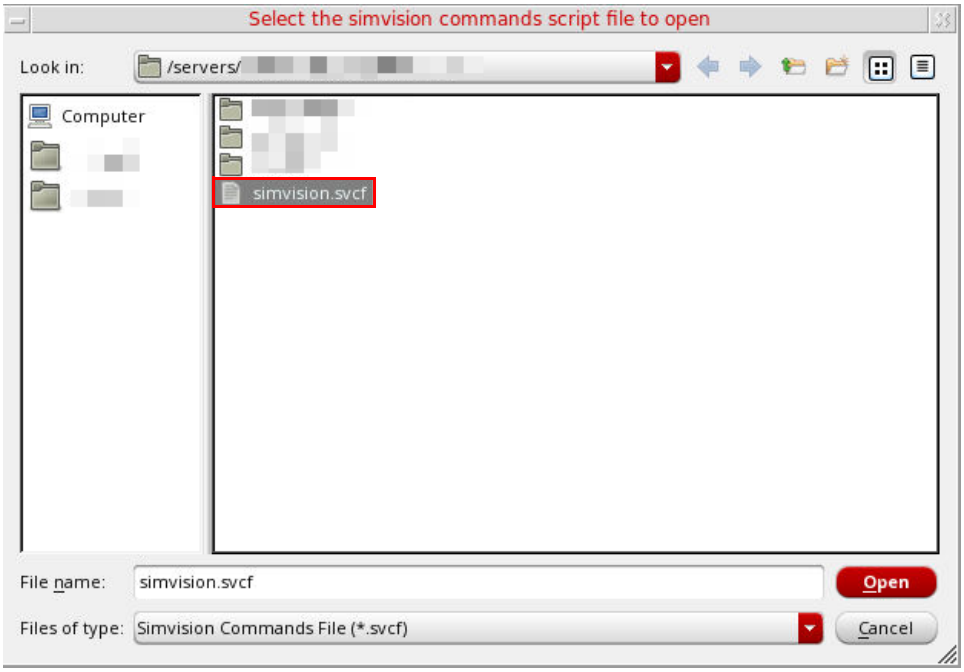
To import a mnemonic map from SimVision:

1. In the top-right corner of the Configure Mnemonics form, click *Import From SimVision*.



© 2023 Cadence Design Systems, Inc.

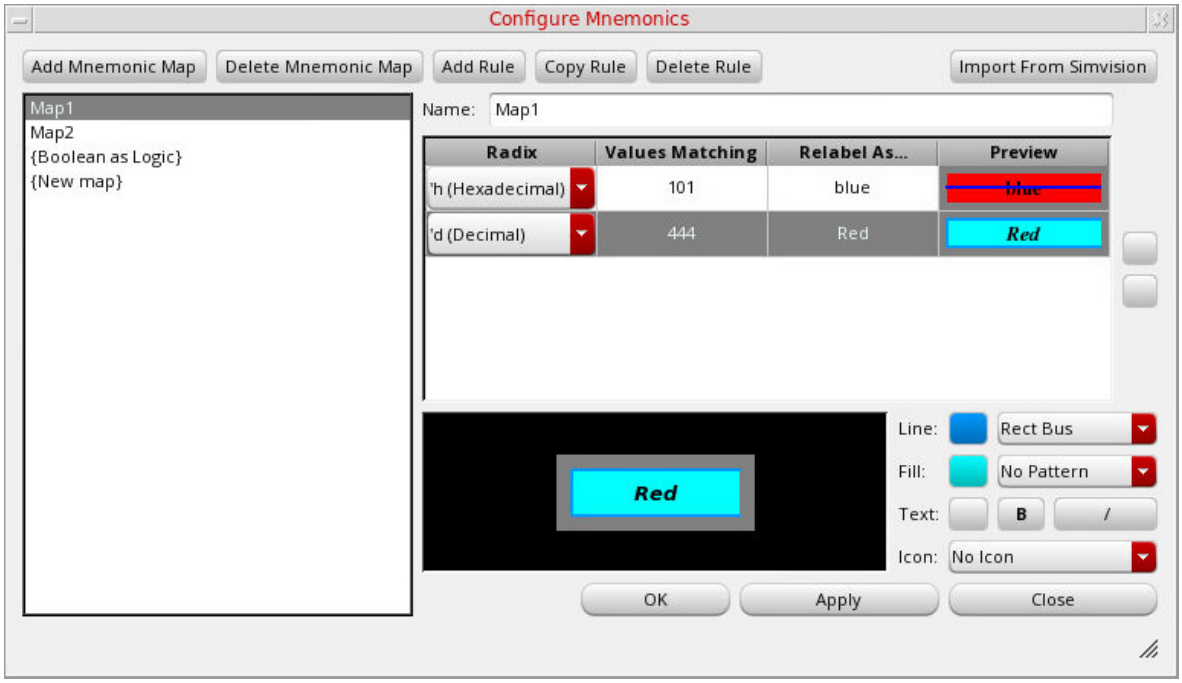
The *Select the SimVision commands script file to open* form appears.



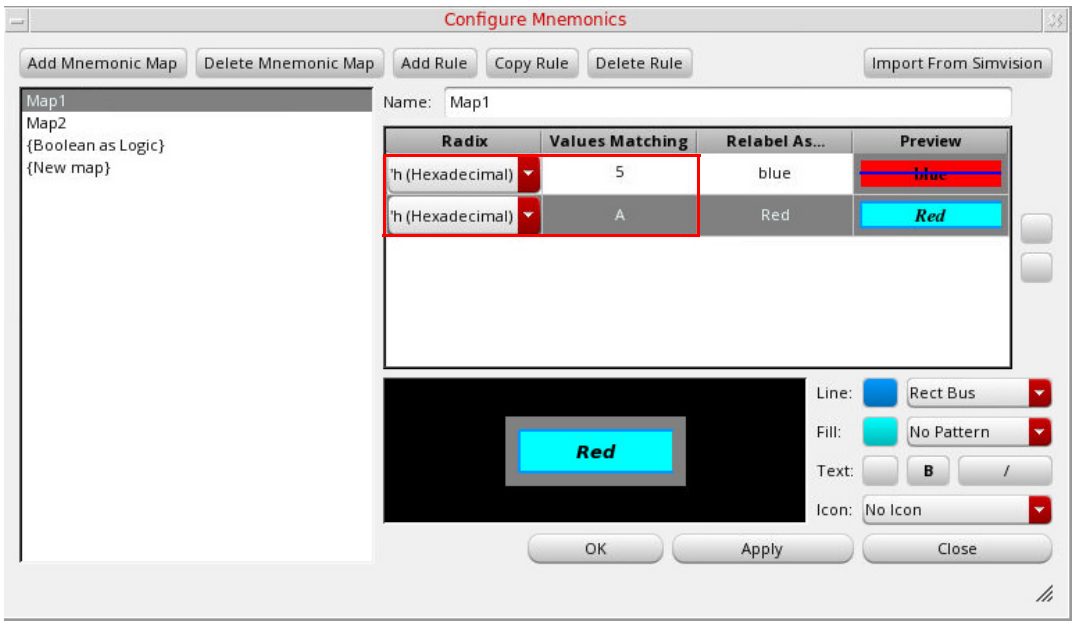
2. Select the SimVision commands file of `.svcf` format from which you want to import the mnemonic map created in SimVision.
3. Click *Open*.

© 2023 Cadence Design Systems, Inc.

The Configure Mnemonics form is updated to show the mnemonic maps defined in the SimVision commands file.



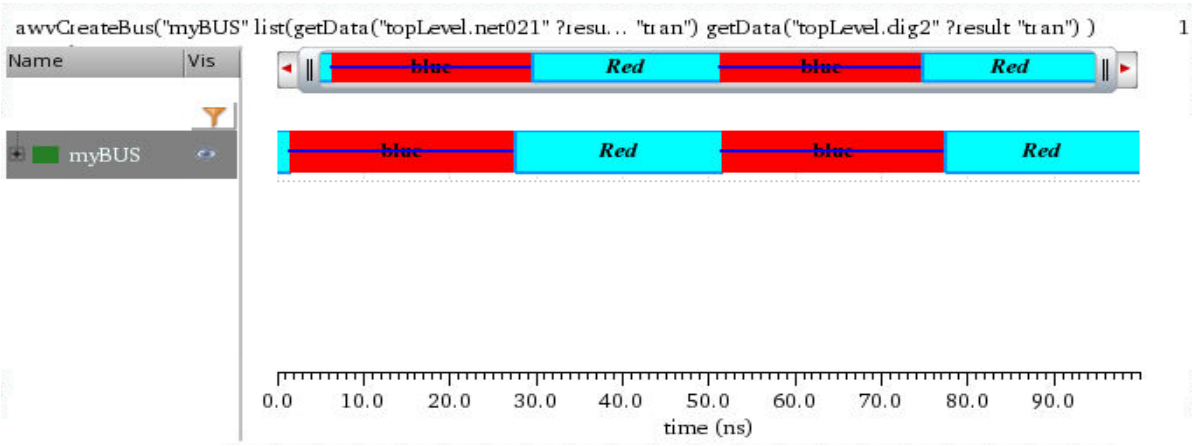
4. Select a mnemonic map that you want to apply. You can also edit rules defined in the selected map. For example, you can edit the values of *Radix* and *Values Matching* according to data in the bus to which you are applying mnemonic map.



© 2023 Cadence Design Systems, Inc.

5. Click *Apply* and then click *OK*.

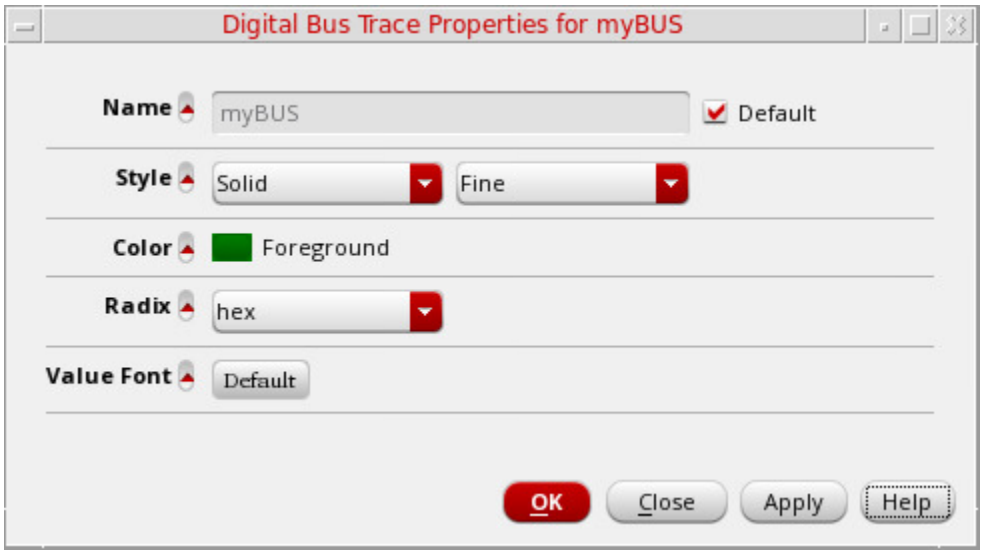
   The bus is updated according to the rules defined in the mnemonic map.



Setting Bus Properties

To set the properties of a digital bus, right-click the bus name and choose *Digital Bus Properties*.

The *Digital Bus Trace Properties for* `Bus_Name` form appears.



This form displays the following information:

© 2023 Cadence Design Systems, Inc.

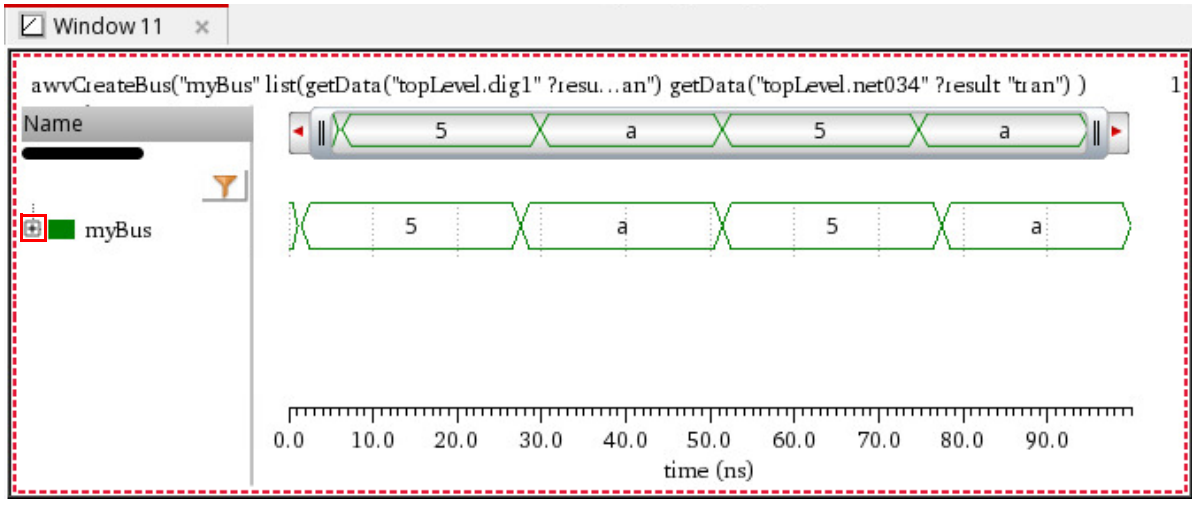
- *Name*—Displays the default name of the bus. You can change the bus name if the Default check box is not selected.
- *Style*—Specifies the bus signal properties, such as *Solid* and *Dash*, and specifies the thickness, such as *Fine* and *Medium.*
- *Color*—Specifies the foreground color of the bus.
- *Radix*—Specifies the radix type, such as *hex*, *binary*, and *octal*. You can also set the radix type by right-clicking the bus and choosing *Display – Radix*.
- In the *Value Font* field, click *Default* to open the Define Font form where you can select the font, font style, and font size that you want to set for the bus values. You can also choose to apply *Strikeout* or *Underline* effects on the selected font.

Expanding a Bus

After you add a bus trace to the window, you can expand the bus to display the digital traces contained in the bus into individual strips.

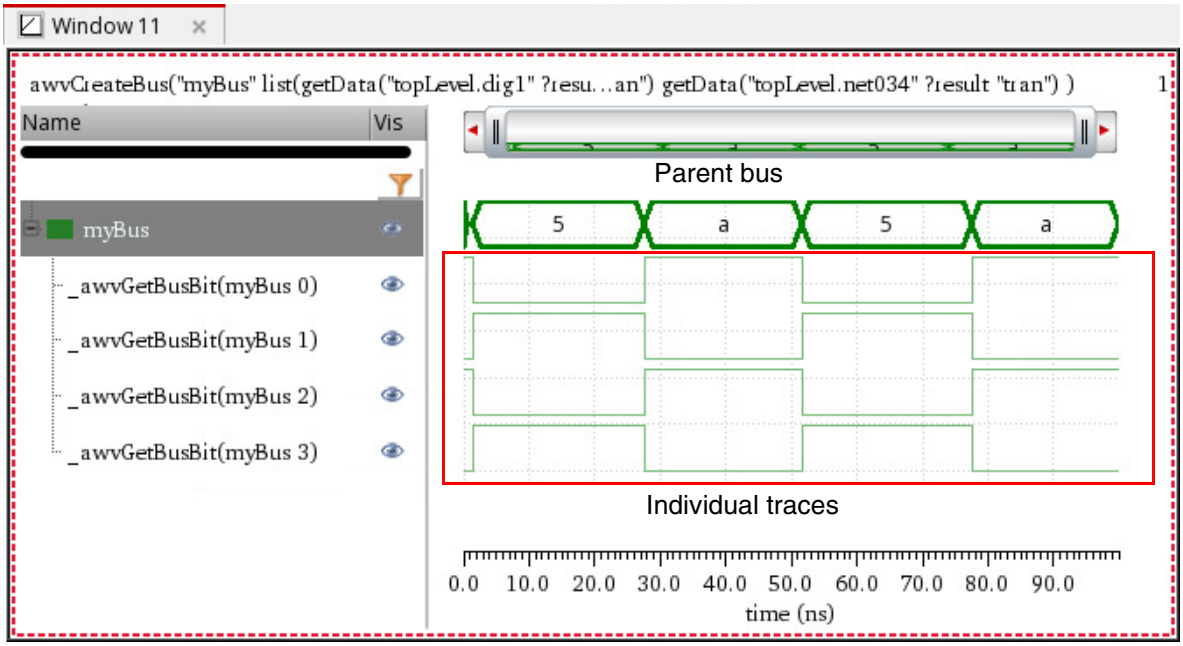
To expand a bus, select the bus you want to expand and do one the following:

- Choose *Trace – Bus – Expand.*
- Right-click the bus or the bus name in trace legend area and choose *Expand*.
- Click the '+' icon next to the bus name in the trace legend area.



© 2023 Cadence Design Systems, Inc.

The window displays the individual digital traces in the selected bus and also displays the parent bus.



Note: After a bus is expanded, you cannot move, cut, drag, or delete the parent bus and the individual bits in the parent bus.

Collapsing a Bus

To collapse an expanded bus, select the bus you want to expand by selecting the bus name in the trace legend area and do one of the following:

- Choose *Trace – Bus – Collapse.*
- Right-click the bus or the bus name in trace legend area and choose *Expand*.

  The window displays only the bus of digital signals.

Exporting a Bus

To export the data from a bus signal into a CSV file, do the following:

- Select the bus and choose *Trace – Export*.
- Right-click the bus and choose *Send to – Export*.

© 2023 Cadence Design Systems, Inc.

The *Export Waveforms* form appears. For more information about this form, see Exporting Waveforms from a Single Results Directory.

Sending a Bus Signal to Calculator

To send a bus signal to Calculator, do the following:

- Right-click the bus signal in graph or in the trace legend and choose *Send To – Calculator*.
- Select the bus and choose *Tools – Calculator*.

  The selected bus signal data is displayed in the Calculator.

Consider an example in which you convert an analog signal to digital using the Analog To Digital assistant, then create the bus from the converted digital signal, and finally send the bus signal to Calculator, the following expression is created in the Buffer:

```
awvCreateBus("Test_bus" list(awvAnalog2Digital(v("net10" ?result "tran") nil nil 2
nil "centre") ) "Binary")
```

Here, `awvAnalog2Digital(v("net10" ?result "tran") nil nil 2 nil "centre"` indicates that the `net10` analog signal has been converted to a digital signal.

Sending a Bus Signal to Table

To send the bus signal to Table, do the following:

➡ Right-click the bus signal in graph or in the trace legend and choose *Send To – Table.*

The selected signal is displayed in Table.

Important

If you send the bus that you created after converting an analog signal to digital, the bus data may not appear in Table. To display the bus data in Table, first send the bus signal to Calculator and then send the corresponding Calculator expression to Table.

Working with Markers

A marker attaches a description to a point on the graph. The default label for a marker displays the X and Y coordinates of its intersection with the trace—if it is attached to the trace—or the coordinates of the point location of the marker. You can associate an expression

© 2023 Cadence Design Systems, Inc.

with a marker label. The expression is evaluated when you place the marker on the graph and updated when you choose *File – Reload*.

If you use the replace mode to plot signals obtained from the simulation runs on the same design, the signals are updated with the new data and the expressions are re-evaluated.

Markers are of the following types:

- Point
- Fixed-y
- Reference Point
- Vertical
- Horizontal
- Circular
- Delta
- AB Delta
- Spec
- Edge
- Period
- dx/dy

This section includes the following topics:

- Adding Markers on page 495
- Setting Marker Properties on page 509
- Snapping Markers on page 514
- Customizing Markers on page 515
- Working with Delta Markers on page 521
- Working With Edge Markers on page 531

Video

To lean the various tips, tricks, and shortcuts that you can apply while working with markers in Virtuoso Visualization and Analysis XL, see the **Tips and Tricks: Markers** video on Cadence Online Support.

© 2023 Cadence Design Systems, Inc.

In addition, you can read the Tips and Tricks on Virtuoso Visualization and Analysis XL Markers blog.

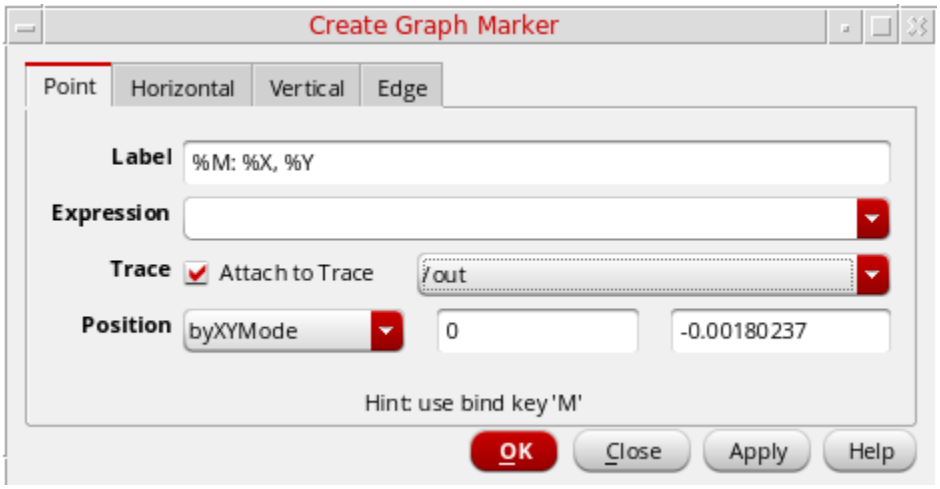
Adding Markers

You can add point, vertical, horizontal, and delta markers to a trace. The circular markers can be added to circular graphs that are obtained from AC analysis, Smith Charts, and polar plots.

To add markers to a trace, do the following:

➡ Select a point on the trace and choose *Marker – Create Marker*.

The *Create Graph Marker* form appears. This form includes various tabs that help to create rectangular markers—point, horizontal, vertical, and edge.



Refer to the below sections to know how to add different markers:

- Adding a Point Marker on page 501
- Adding a Horizontal Marker on page 504
- Adding a Fixed-Y Marker on page 502
- Adding a Vertical Marker on page 505
- Adding an Edge Marker on page 506
- Adding a Period Marker on page 508
- Adding a Dx/Dy Marker on page 508
- Adding a Delta Marker on page 508

© 2023 Cadence Design Systems, Inc.

Adding Markers with Marker Toolbox

To add a vertical, horizontal, point, or reference point (ARefPoint or BRefPoint) marker with the help of the Marker Toolbox, do the following:

➡ Choose *Window – Assistants – Marker Toolbox*.

The *Marker Toolbox* assistant appears to the left of the window. It includes tools that you can use to add the required type of markers. To create a marker, drag the desired marker to the specific location where it needs to be placed.

© 2023 Cadence Design Systems, Inc.

Marker Labels

Depending on the information you want to display in the marker labels, you can specify different format strings in the *Label* field. These formats are evaluated and inserted into the string when you place or edit a marker. As a result, labels can reference properties, such as marker coordinates, trace slope, trace name, and so on, or the result of a scalar expression.

Note: Each marker label displays the default value set in the `defaultLabel` .cdsenv variable.

The following table describes the various format strings that can be specified using the *Label* field.

Marker Label	Description
`%M`	Marker name
`%X`	X-coordinate
`%Y`	Y-coordinate
`%h`	History name
`%t`	Test name
`%c`	Corner name
`%d`	Dataset name
`%x`	Second X-coordinate for delta markers
`%y`	Second Y-coordinate for delta markers
`%W`	Delta value on X-axis (Δx)
`%H`	Delta value in Y-axis (Δy)
`%S`	Slope ($\Delta y/\Delta x$)
`%N`	Name of the trace
`%E`	Evaluated value of the expression
`%F`	Frequency value
`%D`	Absolute path to the results directory from where the signal is plotted
`%i`	Index value of the nearest data point on the trace
For Circular Graphs	
`%C`	Real and imaginary Cartesian values

© 2023 Cadence Design Systems, Inc.

%Z	Impedance values, such as resistance and reactance
%A	Admittance values, such as conductance and susceptance
%R	Reflection coefficients, such as mag and angle
%P	Polar values, such as mag and angle
%F	Frequency value, which includes the independent axis data
%D	Absolute path to the results directory from where the signal is plotted

© 2023 Cadence Design Systems, Inc.

Editing the Marker Label of a Point Marker

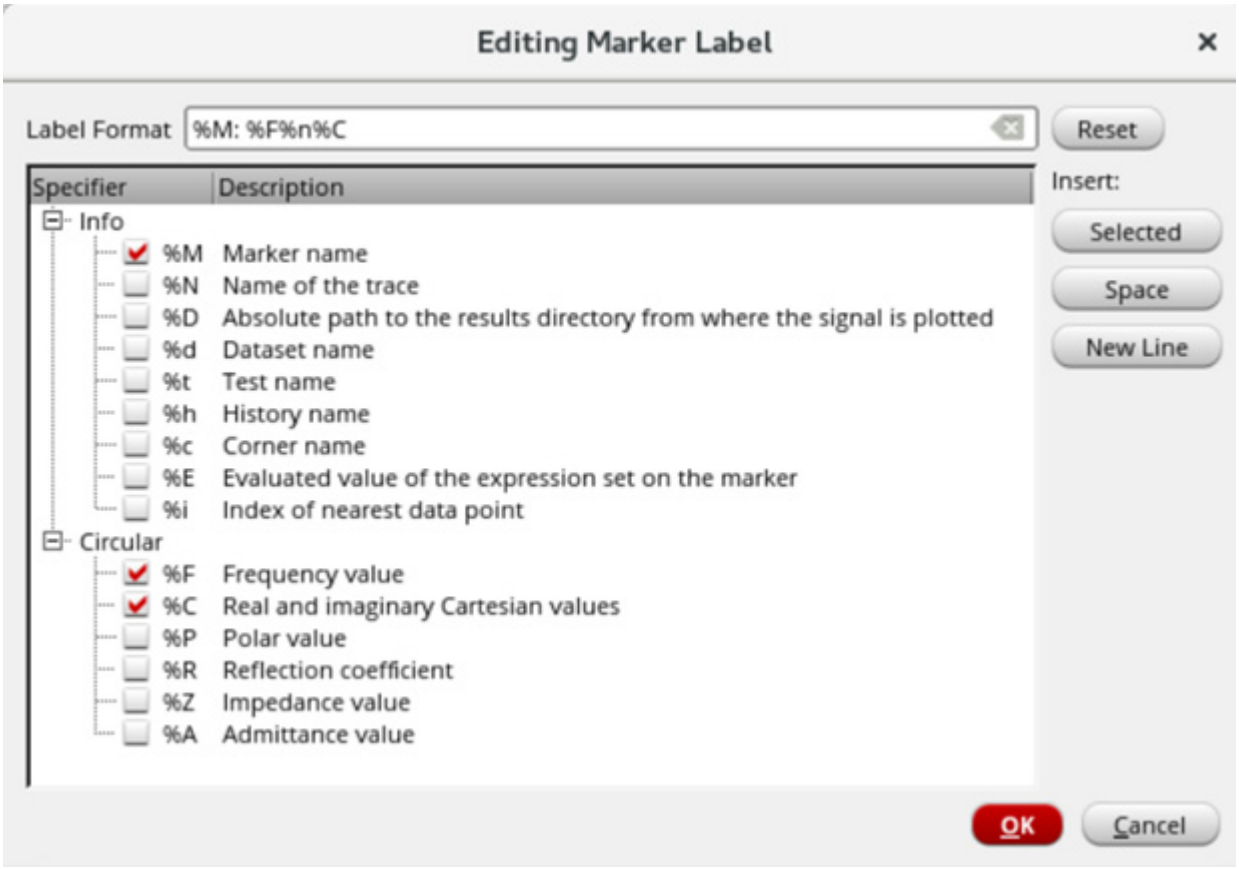
To edit the marker label of a point marker:

1. Right-click a point marker and choose *Point Marker Properties*.

   The Point Marker Properties for `Marker Name` form opens.

2. Click the *Edit* button, which is displayed next to the *Label* field.

   The Editing Marker Label form opens. The format specifiers displayed in this form depend on the type of graph and marker.



3. In the *Label Format* field, type the format specifiers that you want to display in the marker label.

   Alternatively, you can select or deselect the check boxes next to the format specifiers that you want to add or remove from the marker label.

   Note: You can also click the icon to clear the values set in the *Label Format* field. You can click *Reset* to reset the *Label Format* field to the last saved value.

© 2023 Cadence Design Systems, Inc.

4. From the *Insert* options:
   a. Click *Selected* to add the currently selected format specifier to the *Label Format* field.
   b. Click *Space* to add a space between two format specifiers.
   c. Click *New Line* to add a newline character (`%n`) between two format specifiers.
5. Click *OK* to save the changes and close the Editing Marker Label form.

   The values set in the *Label Format* field are automatically updated in the *Label* field on the Point Marker Properties for *Marker Name* form.

Related Topics

Marker Labels

Adding a Point Marker

© 2023 Cadence Design Systems, Inc.

Adding a Point Marker

- To add a point marker, click the *Point* tab. The following fields appear:
  - *Label*—Specify a label for the marker. You cannot insert multiline text in the marker labels.

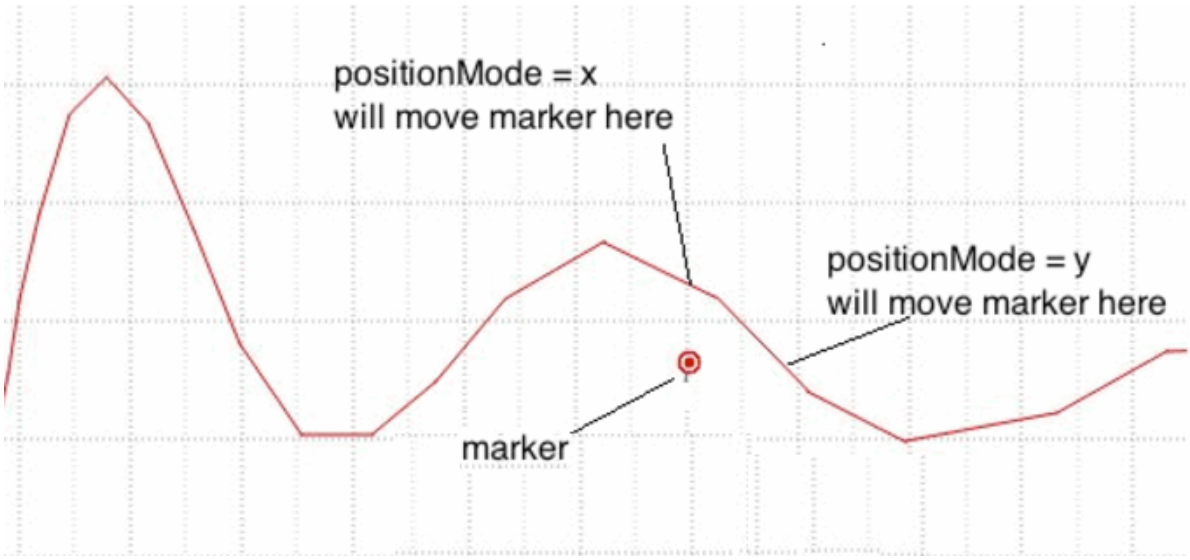
    See the Marker Labels section for more information.

    Note: If you do not enter text in the *Label* field, the X- and Y-coordinates of the marker are displayed.
  - *Expression*—Select the expression you want to display in the marker label. Click the *Expression* arrow to view the *Buffer Contents* and all defined memories (in SKILL mode). If you entered `%E` in the *Label* field, you can choose the variable you want to use in your expression, or you can choose *Calculator Buffer.* If you choose a variable, the expression associated with the selected variable appears in the *Expression* field. If you choose *Calculator Buffer*, the expression in the Calculator Buffer appears in the *Expression* field.
  - *Trace*—Select the *Attach to Trace* check box to attach the marker to the closest interpolated point on the trace. When multiple traces are appended to a graph, in the drop-down list, select the trace to which you want to attach the marker.

    Note: When this check box is selected, the marker color remains same as that of the trace color. Now, if you change the color of the trace, the marker color also changes.
  - *Position*—Select the X and/or Y position from the drop-down list box where you want to place the point marker. You can specify the marker position as:
    `byXMode`—Specify only the X-axis value. Marker is created at the specified X-axis value and the nearest Y-axis value on the trace. (See the image below)
    `byYMode`—Specify only the Y-axis value. Marker is created at the specified Y-axis value and the nearest X-axis value on the trace. (See the image below)
    `byXYMode`—Specify the X- and Y- axes values. Marker is created at the specified X-

© 2023 Cadence Design Systems, Inc.

and Y- axes values on the trace.
The default value is `byXYMode`.

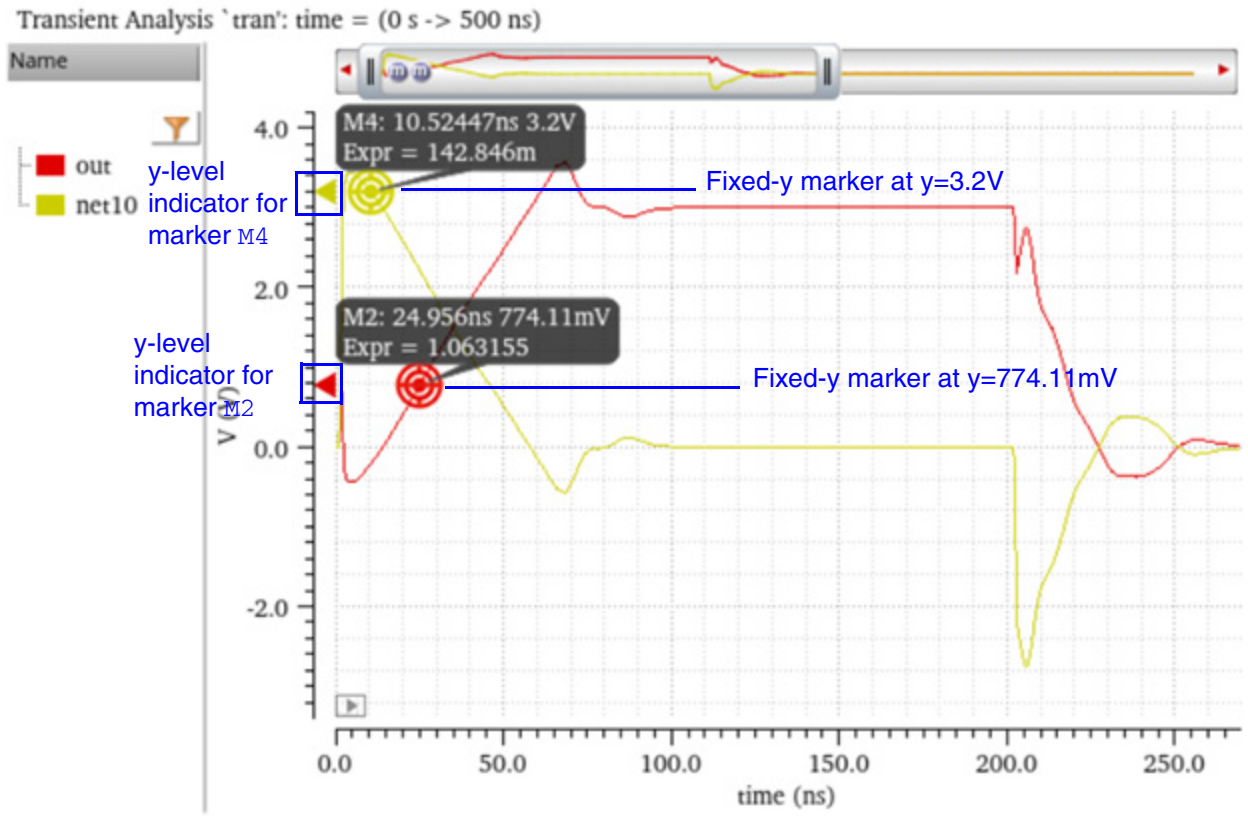


Adding a Fixed-Y Marker

A fixed-y marker is a special type of point marker that is attached to the trace at the specified y-axis value. You can drag this marker to other traces in the same graph window or subwindow to create another fixed-y level marker at the same y-axis value.

© 2023 Cadence Design Systems, Inc.

You can also choose to display a y-level indicator for a fixed-y marker when it is selected. Use this indicator to drag the fixed-y marker vertically, which makes it easy to change the y position of the marker.



To add a fixed-y marker:

1. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Marker – Create Marker*.

   The Create Graph Marker form opens.

2. Click the *FixedY* tab and specify the following fields:

   - In the *Label* field, specify the format specifiers whose corresponding information you want to display in the label of the fixed-y marker.
   - In the *Expression* field, specify the expression whose evaluated value you want to display in the marker label. For example:

     ```
     average(v("out" ?result "tran-tran") )
     ```

   - From the *Trace* drop-down list, select the trace to which you want top attach the fixed-y marker.

© 2023 Cadence Design Systems, Inc.

- Select the *Show Marker Level when Selected* check box to display a y-level indicator when the FixedY marker is selected. You can drag this y-level indicator to drag the marker vertically.
- In the *Y Position* field, specify the y-axis value at which you want to create the fixed-y marker.

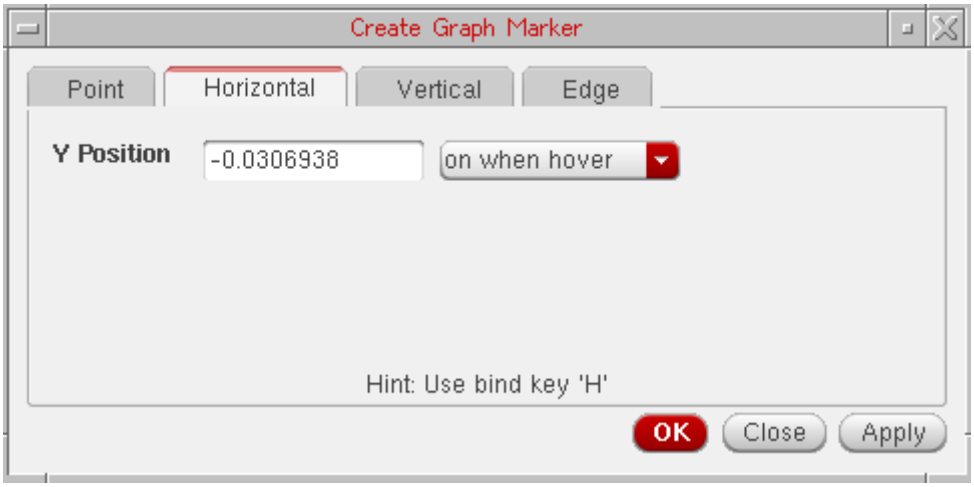
You can also use the bindkey `Shift + Y` to create a fixed-y marker.

Related Topics

Create Graph Marker Form

Adding a Horizontal Marker

- To add a horizontal marker, click the *Horizontal* tab.



The following field appears:

- *Y Position*—Specify the position on the Y-axis where you want to create the marker. You can select a point on the trace; the Y-axis value of that point is displayed in this field.
- In the drop-down list box, select the event for which you want to display the horizontal marker.

You can create multiple horizontal markers at the specified locations in one step by providing a set of Y Position values. For example, `10n 20n 30n`.

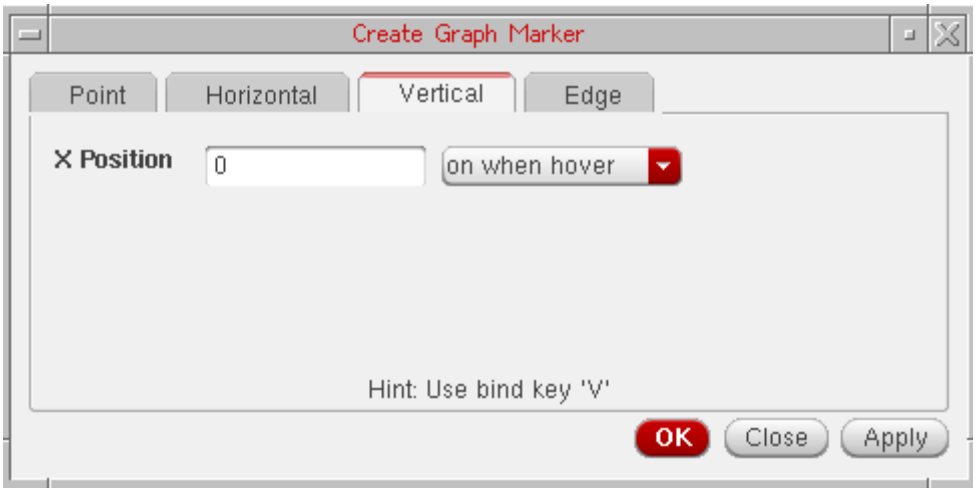
© 2023 Cadence Design Systems, Inc.

To optimize the performance of the tool, the horizontal marker displays 10 intercepts at the maximum. If you have multiple traces plotted in a graph, you need to turn off the visibility of other traces to view the intercepts of a given trace. Alternatively, to view all the horizontal marker intercepts, open the horizontal marker table. For information about horizontal marker table, see Horiz Marker Table Assistant on page 307.

The horizontal marker appears in a dash line style. To change the line style, right-click the marker and choose *Horizontal Marker Properties*. For more information see, Setting Marker Properties on page 509.

Adding a Vertical Marker

- To add a vertical marker, click the *Vertical* tab.



The following field appears:

- *X Position*—Specify the position on the X-axis where you want to create the marker. You can select a point on the trace; the X-axis value of that point is displayed in this field.
- In the drop-down list box, select the event for which you want to display the vertical marker.

You can create multiple vertical markers at the specified locations in one step by providing a set of X Position values. For example, `10n 20n 30n`.

The vertical marker appears in a dash line style. To change the line style, right-click the marker and choose *Vertical Marker Properties*. For more information see, Setting Marker Properties on page 509.

© 2023 Cadence Design Systems, Inc.

The horizontal and vertical markers show all intercepts on all traces across all strips in a subwindow. To display the horizontal and vertical marker intercepts, right-click a horizontal or vertical marker and choose one of the following options in the *Intercepts* menu – *Off*, *On When Hover*, or *On*.

Note: By default, maximum 20 vertical marker intercepts are displayed on the graph. To display the intercepts of particular traces, you need to select those traces. The intercepts of only selected traces will become visible.

Adding Vertical and Horizontal Markers on Point Marker Location

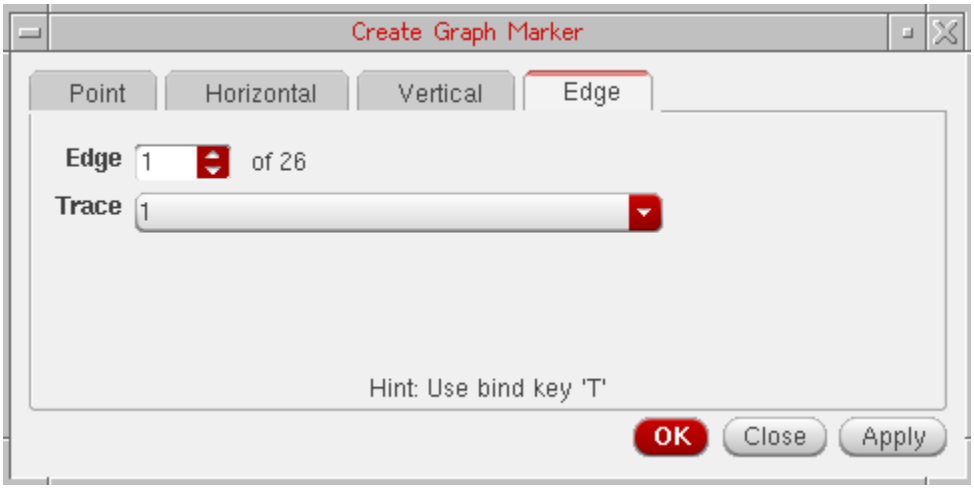
You can add a vertical or a horizontal marker at the same location where a point marker is placed.

To add a vertical marker, right-click the point marker and choose *Create Vertical Marker*.

To add a horizontal marker, right-click the point marker and choose *Create Horizontal Marker*.

Adding an Edge Marker

- To add an edge marker, click the *Edge* tab.



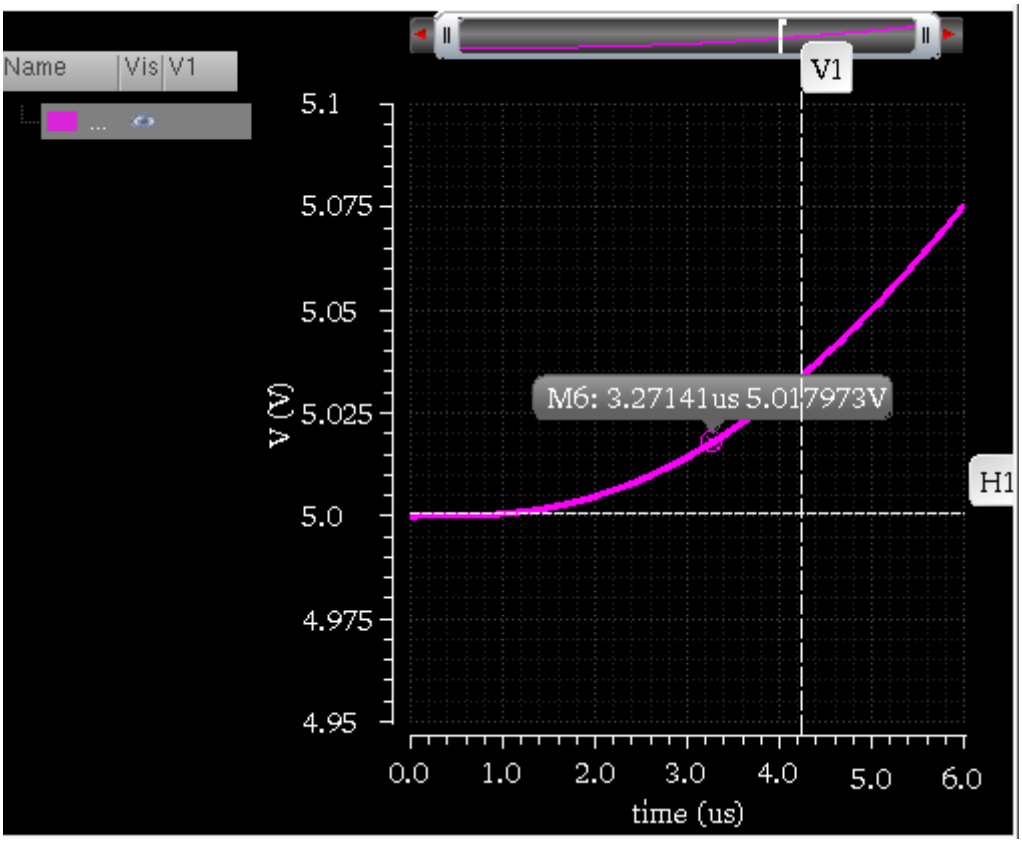
The following fields appear:

- *Edge*—Specify the edge number where you want to create an edge marker. The total number of edges are displayed next to this field.

© 2023 Cadence Design Systems, Inc.

- *Trace*—Select a trace from the drop-down on which you want to place the edge marker. This drop-down lists the name of all traces that are plotted in the selected window.
- For more information about how to create edge markers, see Creating an Edge Marker on page 532.

The following figure shows a point, vertical, and horizontal marker.



Adding Markers with Bindkeys

To add a marker with the help of bindkeys, do the following:

1. Click a point on the graph where you want to place a marker.
2. Press one of the following keys:
   - `M`—Adds a point marker
   - `H`—Adds a horizontal marker
   - `V`—Adds a vertical marker

© 2023 Cadence Design Systems, Inc.

A marker is placed on the trace based on the bindkey you use.

Adding a Period Marker

The Period Marker measures the distance between the selected edge and its adjacent rising or falling edge. By default, this measures the threshold mid-point distance between the two edges. You can use the marker context menu options to change the start and end measurement points for each edge. For example, you can measure from the high threshold of the first edge to the low threshold of the end edge. When you create a period marker, if required, the edge markers are added and connected with a standard delta marker. For more information about this marker, see Transient Measurement Assistant on page 308.

Adding a Dx/Dy Marker

The Dx/Dy marker is a standard delta marker with end points initially placed at the low and high threshold intercept points of a single edge. After this marker is created, you can perform all the delta marker operations on it. The dx/dy marker does not change with the change in the edges. For more information about this marker, see Transient Measurement Assistant on page 308.

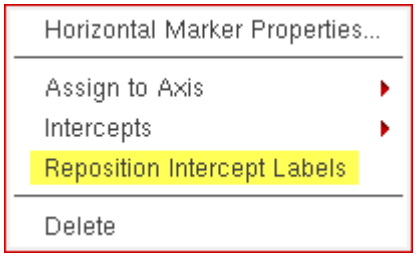
Adding a Delta Marker

Delta markers are used to mark the difference between two points in a graph. A delta marker joins two point markers in the same or different traces. To place a delta marker you must first place a point marker or select one. Delta markers can be moved or deleted independent of their point markers. For information about how to work with a delta marker, see Working with Delta Markers on page 521.

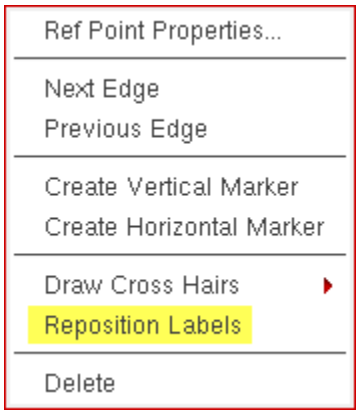
© 2023 Cadence Design Systems, Inc.

Repositioning Marker Labels

You can drag and move the labels attached to the horizontal and vertical markers. To bring the marker labels back to their original position, right-click the horizontal, or vertical markers and choose *Reposition Intercept Labels*.



After moving labels attached to the point, reference points (A/B marker), delta, and circle markers, you can bring the marker labels back to their original position by right-clicking the point marker and choosing *Reposition Labels.*



Setting Marker Properties

To view or change the properties of markers, do one of the following.

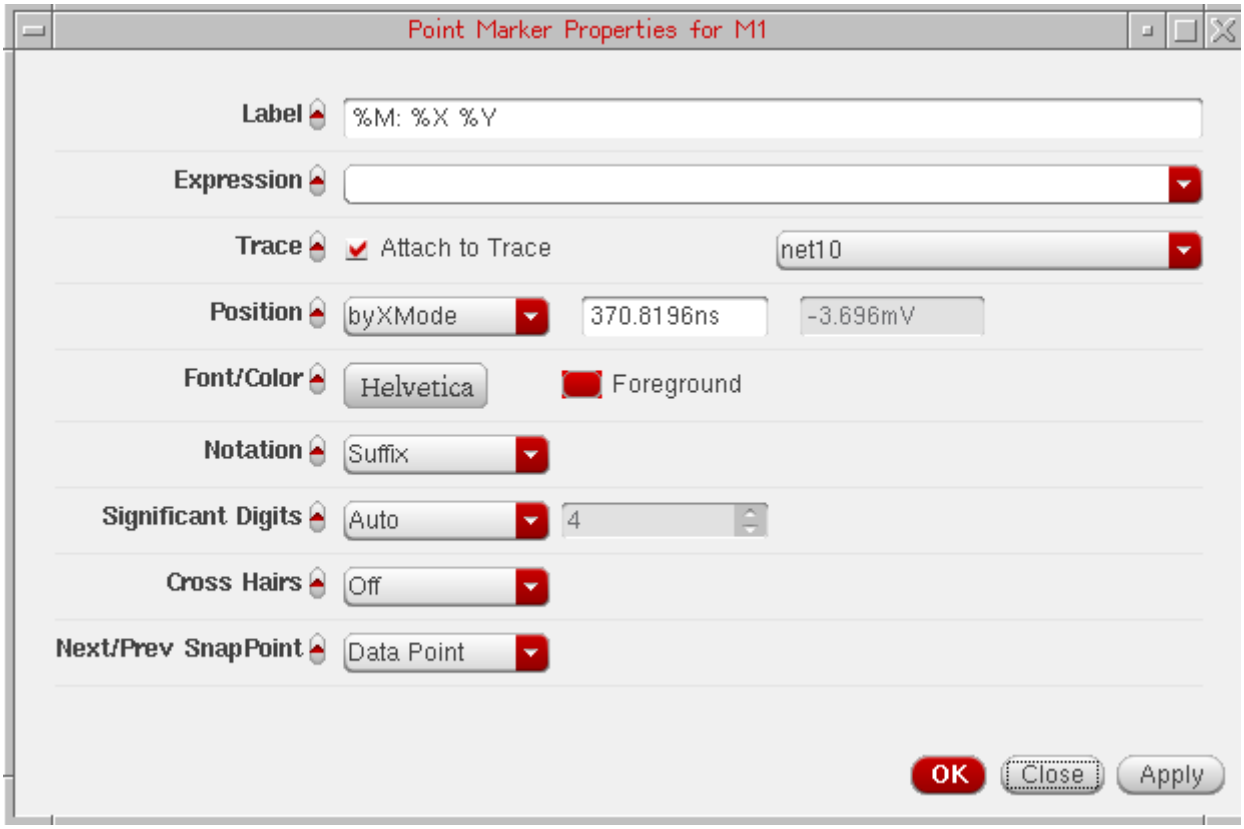
- Double-click the marker.
- Right-click the marker and choose *Marker Properties*.

  The *Marker Attribute Properties* form appears. The properties form displays different fields based on the type of marker selected.

© 2023 Cadence Design Systems, Inc.

Setting Properties for Point Markers

The following fields are displayed for setting the properties of a point marker:



- *Label*—Specify the label for the marker. See Marker Labels for more information.
- *Expression*—Specify the expression associated with the marker. For more information about expressions, see Adding Markers on page 495.
- *Trace*—Specify the name of the trace to which you want to attach the marker.
- *Position*—Specify X-axis and Y-axis position for point marker. By default, the position is `byXMode`.
- *Font/Color*—Specify the font name, style, and size for the label.
- *Notation*—Specify the notation to be displayed on labels. The available options are—`Suffix`, `Engineering`, and `Scientific`. Default value: `Suffix`.
- *Significant Digits*—Specify the number of significant digits if you select `manual` from the drop-down. The other option is `auto`.

© 2023 Cadence Design Systems, Inc.

- *Cross Hairs*—Specify whether you want to display the cross hairs for point marker or delta marker. If you select *Dynamic* from the *Cross Hairs* drop-down list box and click the reference point marker, the cross hairs are displayed.

  Alternatively, to turn on the cross hairs for a delta marker, right-click one of the reference point markers and choose *Draw Cross Hairs – On/Off/Dynamic*.

- *Next/Prev SnapPoint*—Specify the criterion based on which the selected reference point marker should be snapped.

  The available options are—`Local Maxima`, `Local Minima`, `Local Max or Min`, `Specific Y Value`, `Specific X Value`, `Data Point`, `Global Maxima`, and `Global Minima`.

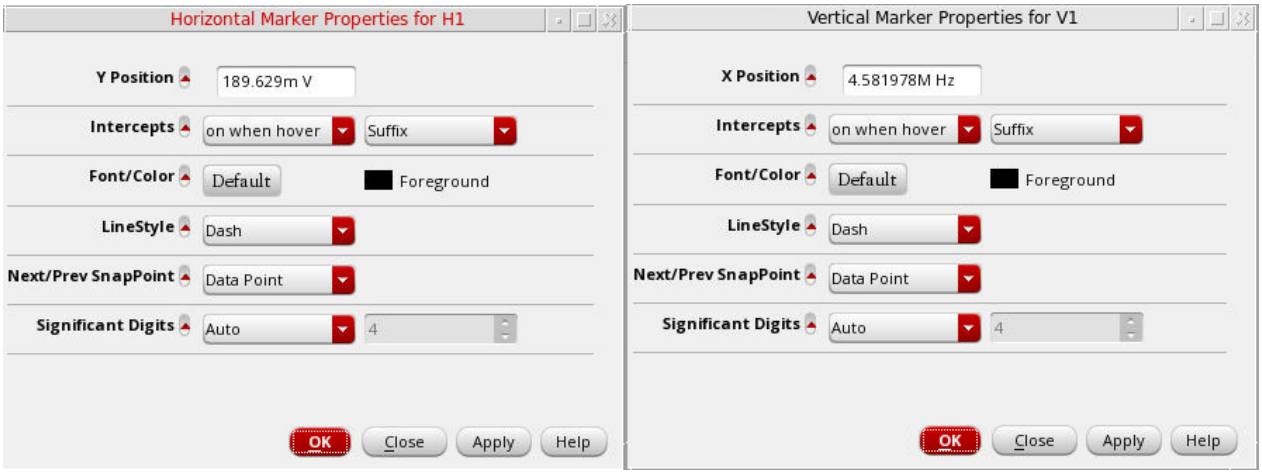
  Default value: `Data Point`.

Setting Properties for Horizontal and Vertical Markers

To change the properties for horizontal or vertical marker, do one of the following:

- Select the marker and choose *Marker – Properties*.
- Right-click the marker and choose *Vertical Marker Properties* for vertical marker and *Horizontal Marker Properties* for horizontal marker.
- Right-click the marker column in the Vertical or Horizontal Marker Table and choose *Vertical Marker Properties* for vertical marker and *Horizontal Marker Properties* for horizontal marker. For more information about marker tables, see Displaying Intercept Data for Markers in Marker Tables on page 516.

The following fields are displayed for setting the properties of a horizontal or vertical marker:



© 2023 Cadence Design Systems, Inc.

- *Y Position*—Specify the Y-coordinates where you want to place the horizontal marker. You can specify multiple Y-coordinates to place multiple horizontal markers at a time.
- *X Position*—Specify the X-coordinates of the point where you want to place the vertical marker. You can specify multiple X-coordinates to place multiple vertical markers at a time.
- *Intercepts*—Specify the event for the marker intercept label display. Alternatively, right-click the marker and choose *Intercepts – Off/On When Hover/On*.
- *Font/Color*—Specify the font name, style, and size for the label.
- *Foreground*—Select the foreground color for the symbol, the arrow and the label.
- *Line Style*—Specify the marker line style, such as dotted.
- *Significant Digits*—Specify the number of significant digits if you select `manual` from the drop-down. The other option is `auto`.
- *Next/Prev SnapPoint*—Specify the criterion based on which the selected reference point marker should be snapped.

  The available options are—`Local Maxima`, `Local Minima`, `Local Max or Min`, `Specific Y Value`, `Specific X Value`, `Data Point`, `Global Maxima`, and `Global Minima`.

  The `Specific X Value` option is not available for horizontal markers.

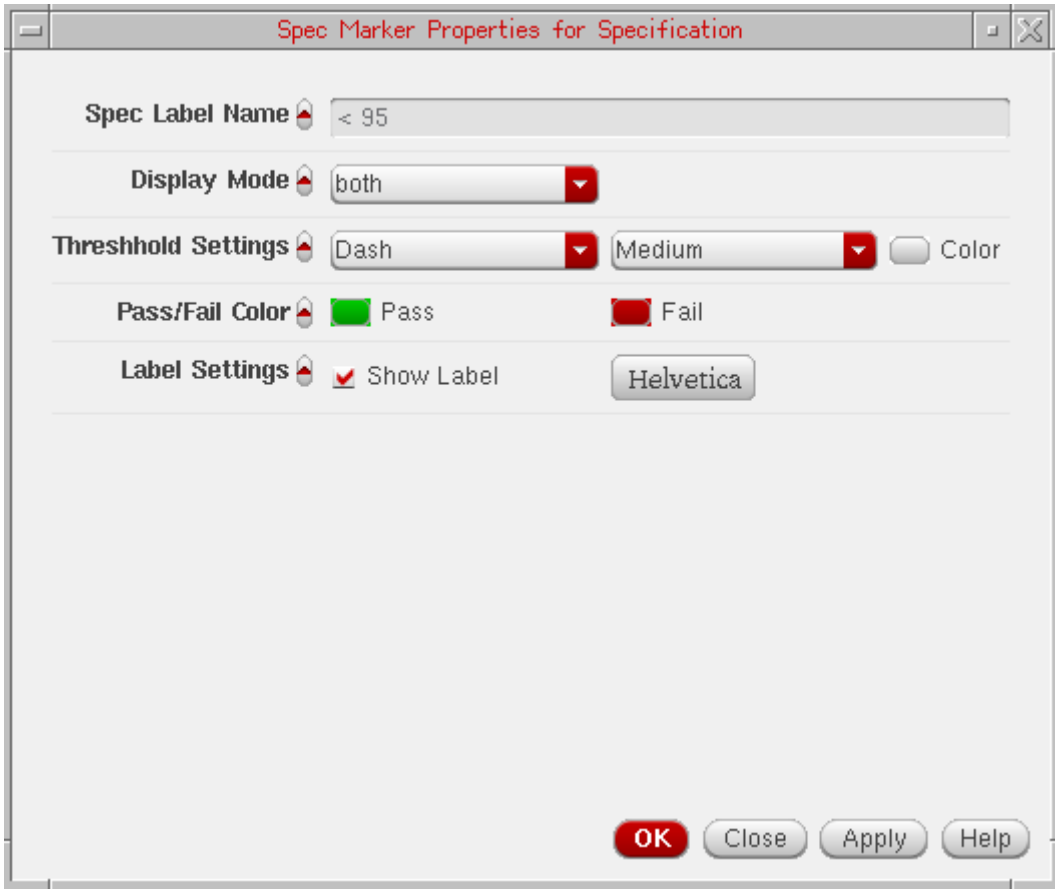
  Default value: `Data Point`.

Setting Properties for Spec Markers

To view or change the properties of spec markers, right click the corresponding trace and choose *Spec Properties*.

© 2023 Cadence Design Systems, Inc.

The *Spec Marker Properties for Specification* form is displayed, as shown in the following figure:



In this form, you can edit the following properties:

- *Spec Label Name*—Shows the spec name that is displayed on the label.
- *Display Mode*—Specify the spec marker pass/fail display style. The available options are `none`, `pass`, `fail`, `both`, `thresholdOnly`, and `pointsOnly`.
- *Threshold Settings*—Specify the line style, line thickness, and line color for the spec marker.
- *Pass/Fail Color*—Specify the colors to shade pass or fail regions.
- *Label Settings*—Specify the font and display settings for spec marker labels.

If a graph contains multiple traces, spec marker properties cannot be changed for an individual trace. Changes in spec marker properties are applicable for all the traces of a graph.

© 2023 Cadence Design Systems, Inc.

Snapping Markers

You can snap markers to analog and digital traces. In analog traces, you can also set the criteria based on which you want to snap markers, where as digital markers can be snapped only to the edge transitions, low to high and high to low. If the marker extends beyond the display area, the marker is panned automatically.

For digital traces or buses, the vertical, delta, and point markers can be snapped and for analog traces, the horizontal, vertical, delta, and point markers can be snapped based on the snapping criterion.

To set the snapping criteria to snap the markers to analog traces in the window, do the following:

1. Select a maker and choose a snapping criterion from the drop-down list box displayed on the snap toolbar based on which you want to snap the marker. For example, local maxima, local minima, and so on. By default, `Data Point` is selected in the drop-down list box.

   When you add a vertical or horizontal marker on a trace, the *Value* field displays the X-axis location of the selected marker. This field is updated automatically if you move the marker.

2. Specify a value for the selected snap criterion by which you want to snap the marker.
3. Then, to snap the marker to the next and previous snap points, do one of the following:
   - On the Snap toolbar, click the *Next Edge* and *Previous Edge* buttons.
   - Right-click the selected marker and choose *Next Edge* and *Previous Edge* respectively. These options are not available in Horizontal marker context menu.
   - Press the `N` or `P` bindkeys to move to the next or previous edges, respectively.

   The selected marker is snapped based on the snap criterion you selected. For example, if you select *Local Maxima* as the snap criterion, the marker is shifted to the maxima value (peak) local to the curve when you click the *Next Edge* button.

   You can select any one of the following snap criterion:

   - **Local Maxima**—Defines the transition point when there is a change in the slope from the rising to falling edge starting from the marker's current position. The local maxima is calculated as the change in slope from rising to falling edge starting from the current marker position and the transition point is known as the local maxima.

© 2023 Cadence Design Systems, Inc.

- **Local Minima**—Defines the transition point when there is a change in the slope from the falling to rising edge starting from the marker's current position.

  Note: At any point, the double derivative of a waveform should be zero to find the local maxima or the local minima. **Local Max or Min**—Defines that the snap point can be either local maxima or local minima.

- **Specific Y Value**—Defines the snap point of the marker to a specific Y-axis value.
- **Specific X Value**—Defines the snap point of the marker to a specific X-axis value.

  Note: This snap criterion works for all types of markers except the horizontal markers.

- **Data Point**—Defines that a specific data point on the curve should be considered as the snap point. This is the default snap criterion.
- **Global Maxima**—This snap criterion is similar to local maxima. The only difference is that it applies the snap settings to the global maxima or positive peak.
- **Global Minima**—This snap criterion is similar to local minima. The only difference is that it applies the snap settings to the global minimum value.

If you want to use the same snap criterion to snap a marker to more than one analog trace, select the maker and then hold down the `Ctrl` key and click the analog traces. Next, click the *Next Edge* or *Previous Edge* button to snap the marker to the snap points on the selected traces.

To snap digital markers, select the digital marker and click the *Next Edge* and *Previous Edge* buttons on the Snap toolbar.

For more information about snapping point markers and reference point markers on circular graphs, see Snapping Markers on Circular Graphs on page 554.

Customizing Markers

This section includes the following topics:

- Displaying Intercept Data for Markers in Marker Tables on page 516
- Deleting a Marker on page 519
- Editing a Marker on page 520
- Moving a Marker on page 520

© 2023 Cadence Design Systems, Inc.

- Zooming a Marker on page 520

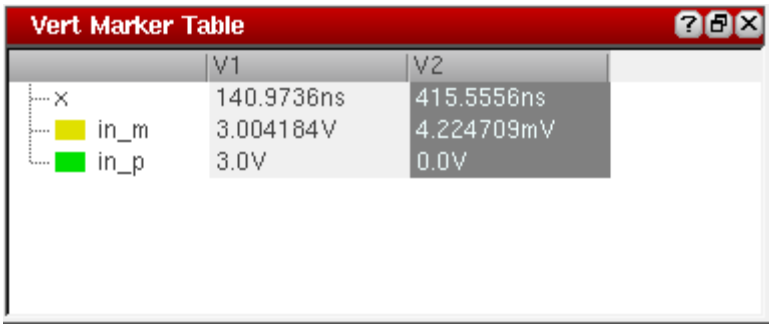
Displaying Intercept Data for Markers in Marker Tables

Vertical Marker Table

To display the vertical marker intercepts, do the following:

➥ Choose *Window – Assistants – Vertical Marker Table*.

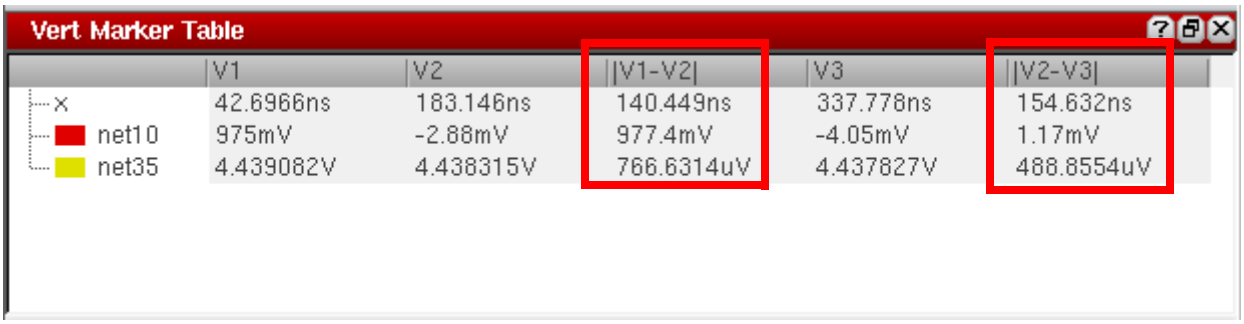
The *Vert Marker Table* assistant appears at the bottom of the window, displaying all vertical marker intercepts for each trace. When you add a vertical marker on a trace, the vertical marker intercepts for all the traces are displayed in the marker table.



In the vertical marker table, rows display the trace names and columns display the intercept points of each vertical marker.

Note: The active vertical marker intercepts for each trace are also displayed in the trace legend area.

If you create a delta marker between two or more vertical markers, the vertical marker table includes an additional column to display the vertical marker delta values on traces (as shown in the figure below).



© 2023 Cadence Design Systems, Inc.

You can create a delta marker between two vertical markers by using one of the following methods:

- Create a vertical marker by using the bindkey `V`. Keeping the marker selected, place the mouse pointer at a point on the trace where you want to create the other vertical marker. Press the bindkey `D`. A delta marker is created between the two vertical markers.
- Select all vertical markers by using the `Ctrl` key and choose *Marker – Create Delta Marker*.
- Select all vertical markers by using the `Ctrl` key and press bindkey `Shift+D`.

Note: The delta value displayed in the vertical marker table is always an absolute value.

If you do not want to display delta values in the vertical marker table, right-click the delta line joining two vertical markers and choose *Diff Visible*.

You use the horizontal marker table to view the trace intercepts for all the horizontal markers in a table.

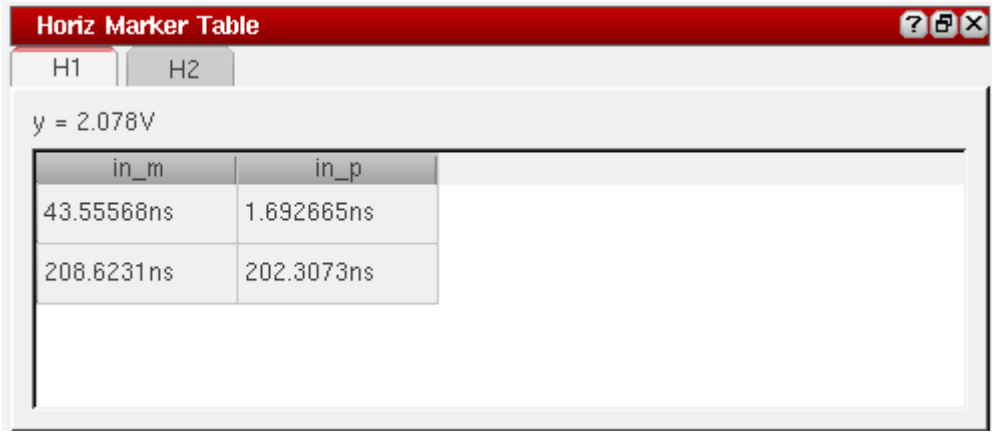
Horizontal Marker Table

To display the horizontal marker intercepts in a table, do the following:

➥ Choose *Window – Assistants – Horizontal Marker Table*.

The *Horiz Marker Table* form appears at the bottom of the window and displays the intercepts where horizontal marker intersects traces in the graph.

The horizontal marker table includes a separate tab for each horizontal marker. The marker table for the active horizontal marker is displayed. If you change the marker name, the tab name in the table is updated automatically.



© 2023 Cadence Design Systems, Inc.

Note: When you zoom-in a graph, the horizontal marker table lists only those intercepts that are visible in the zoomed-in portion of the graph. The horizontal marker table is updated only when you move the marker.

If the graph includes multiple Y-axes, do the following to change the axis of the horizontal marker:

➡ Right-click the marker and choose *Assign to Axis – axis-name*.

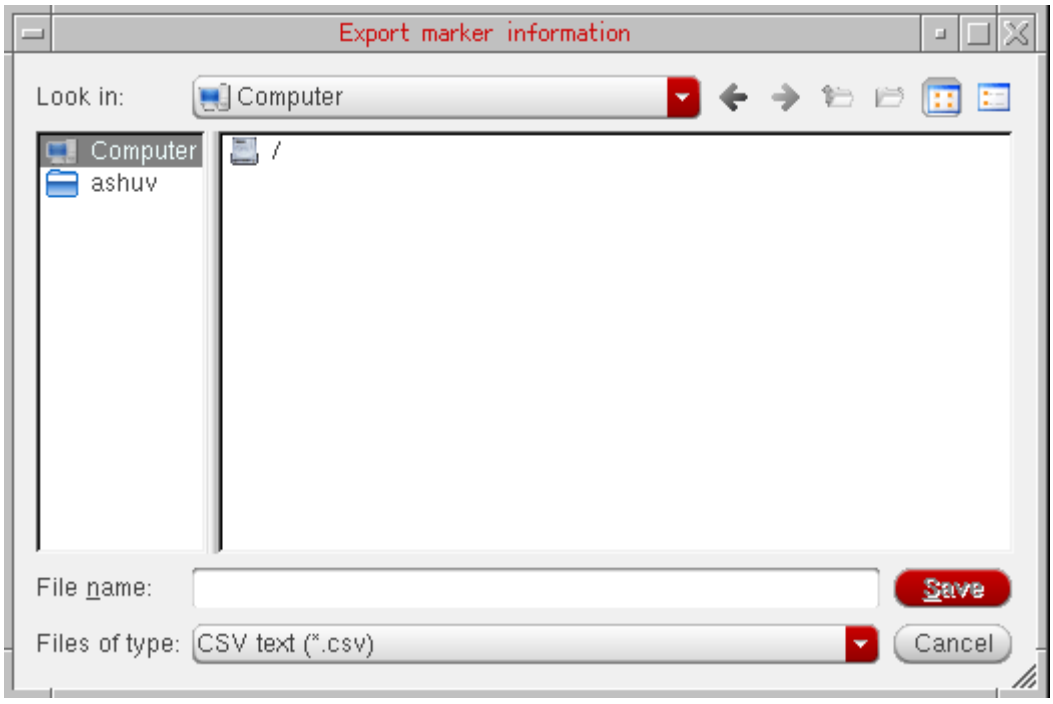
Horizontal marker now shows intercepts for the traces that are attached to the axis you select. The marker table is also updated with the new intercepts.

Exporting Markers

To export the vertical marker intercept data in a CSV file, do the following:

➡ Choose *Marker – Export – Vertical Marker*.

The *Export marker information* form appears. In this form, specify the name and location of the CSV file in which you want to save the vertical marker information and then click *Save*.

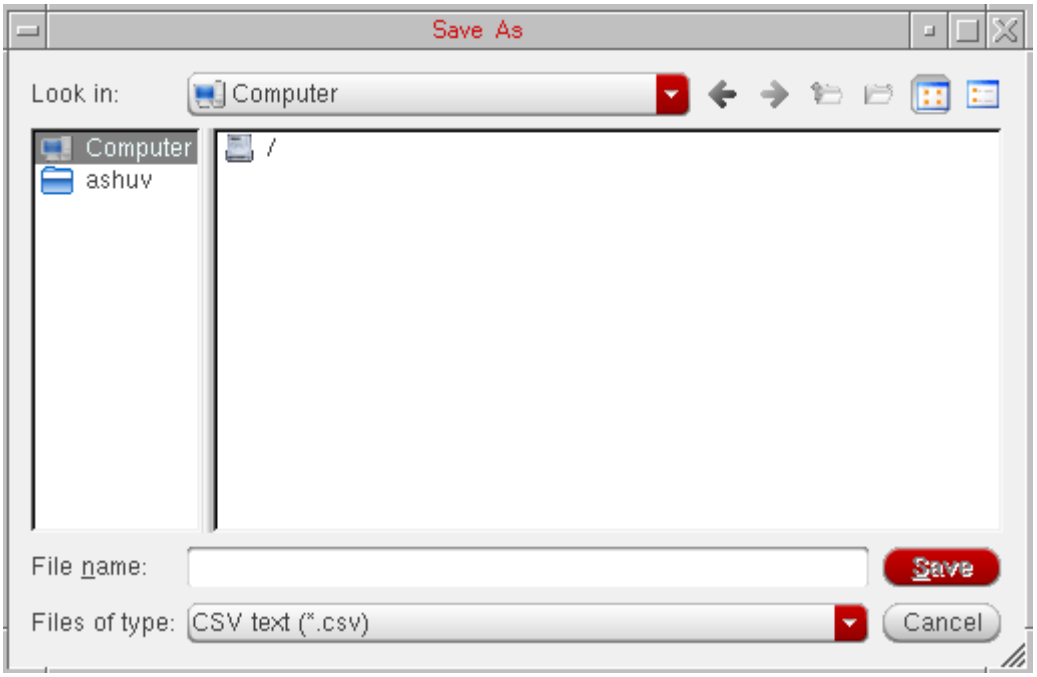


To export the horizontal marker intercept data in a CSV file, do the following:

➡ Choose *Marker – Export – Horizontal Marker.*

© 2023 Cadence Design Systems, Inc.

The *Save As* form appears. In this form, specify the name and location of the CSV file in which you want to save the horizontal marker information and then click *Save*.



Deleting a Marker

To delete a marker, do one of the following:

- Select a marker and choose *Edit – Delete*, or press the `Delete` key.
- Right-click a marker and choose *Delete*.

To delete all markers on a trace, select a marker and do one of the following:

- Choose *Edit – Delete All*
- Choose *Trace – Delete All*
- Press `Ctrl+E`

To delete a AB delta marker, you can right-click any of the two point markers or the delta marker line and choose *Delete*. The A and B markers in the delta marker are deleted.

© 2023 Cadence Design Systems, Inc.

Editing a Marker

The default marker attributes are controlled by the values assigned to variables in the `.cdsenv` file. For more information, see Appendix A, "Virtuoso Visualization and Analysis XL Tool Environment Variables."

To edit a marker, double-click the marker. The *Marker Properties* form appears. Edit the required fields in this form. For more information about the fields, see Setting AB Delta Marker Reference Point Properties on page 525.

Moving a Marker

To move a point marker, drag the point marker anywhere on the trace.

To move a vertical marker, place the pointer on the vertical marker. When the pointer becomes a bidirectional arrow, drag the pointer along the X-axis to move the marker. Similarly, drag a horizontal marker along the Y-axis to move the marker.

Zooming a Marker

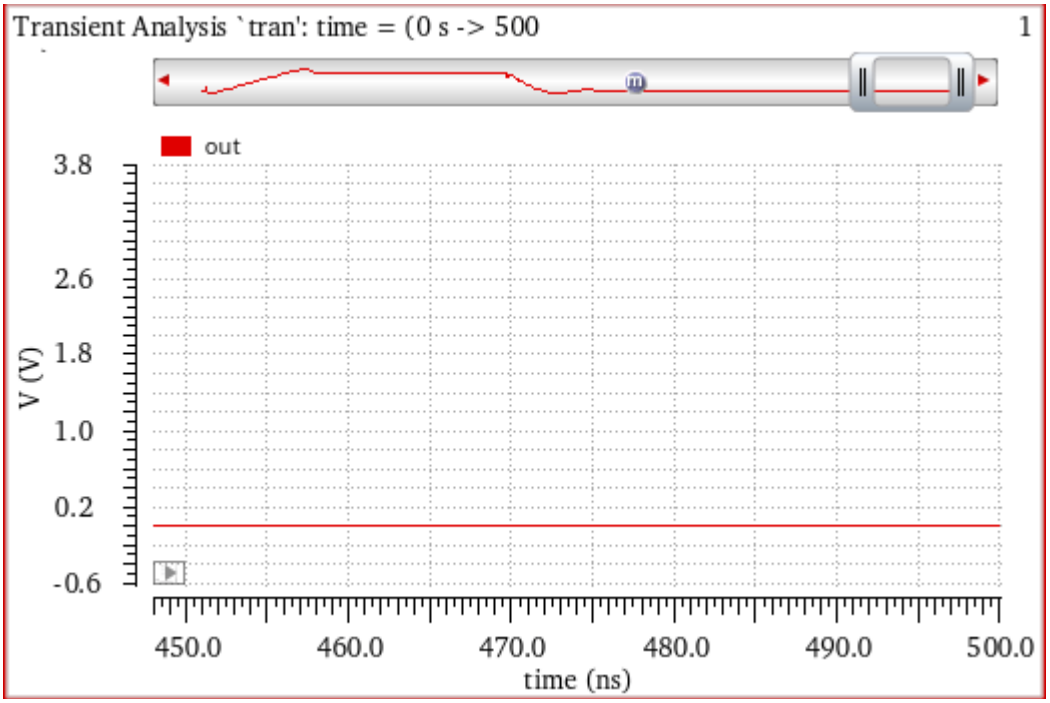
You can zoom in a graph and add a point marker on the zoomed in region of the graph. Now, when you zoom out, if the data point where you created a marker is out of the current screen, the marker may not place correctly and move out of the screen range.

The data point which is out of the current screen is always closer to a data point inside the current screen, because the X range is much smaller than the Y range. Virtuoso Visualization and Analysis XL looks for the closest data point among all data and not the data on the current screen.

Consider the following example in which you zoom in to the range from 450ns to 500ns and then move the pointer to (480ns, 0.4V) and then add a point marker. This marker is placed at (274ns, 0.0081V) because data points on screen are (482ns, 0.00464V) (472s, 0.00453V),

© 2023 Cadence Design Systems, Inc.

and so on. Notice (480ns, 0.4V) is closer to (274ns, 0.0081V) than the data points on the screen.



Working with Delta Markers

Delta markers are used to mark the difference between two points in a graph. A delta marker joins two point markers in the same or different traces. To place a delta marker you must first place a point marker or select one. Delta markers can be moved or deleted independent of their point markers.

You can move either end of a delta marker; X and Y coordinates are updated accordingly. You can use delta markers to measure delays or use them with the min and max functions to measure peak-to-peak values.

This section includes the following topics:

- Adding Delta Markers on page 522
- Adding AB Marker on page 524
- Setting AB Delta Marker Reference Point Properties on page 525
- Deleting Delta Markers on page 527
- Editing Delta Marker Properties on page 527

© 2023 Cadence Design Systems, Inc.

Adding Delta Markers

You can create delta markers between two or more point, vertical, and horizontal markers on one or more traces. You can also create delta markers between two different marker types. For example, you can create delta marker between point markers and vertical markers.

Following are the two methods that you can use to create multiple delta markers on a trace:

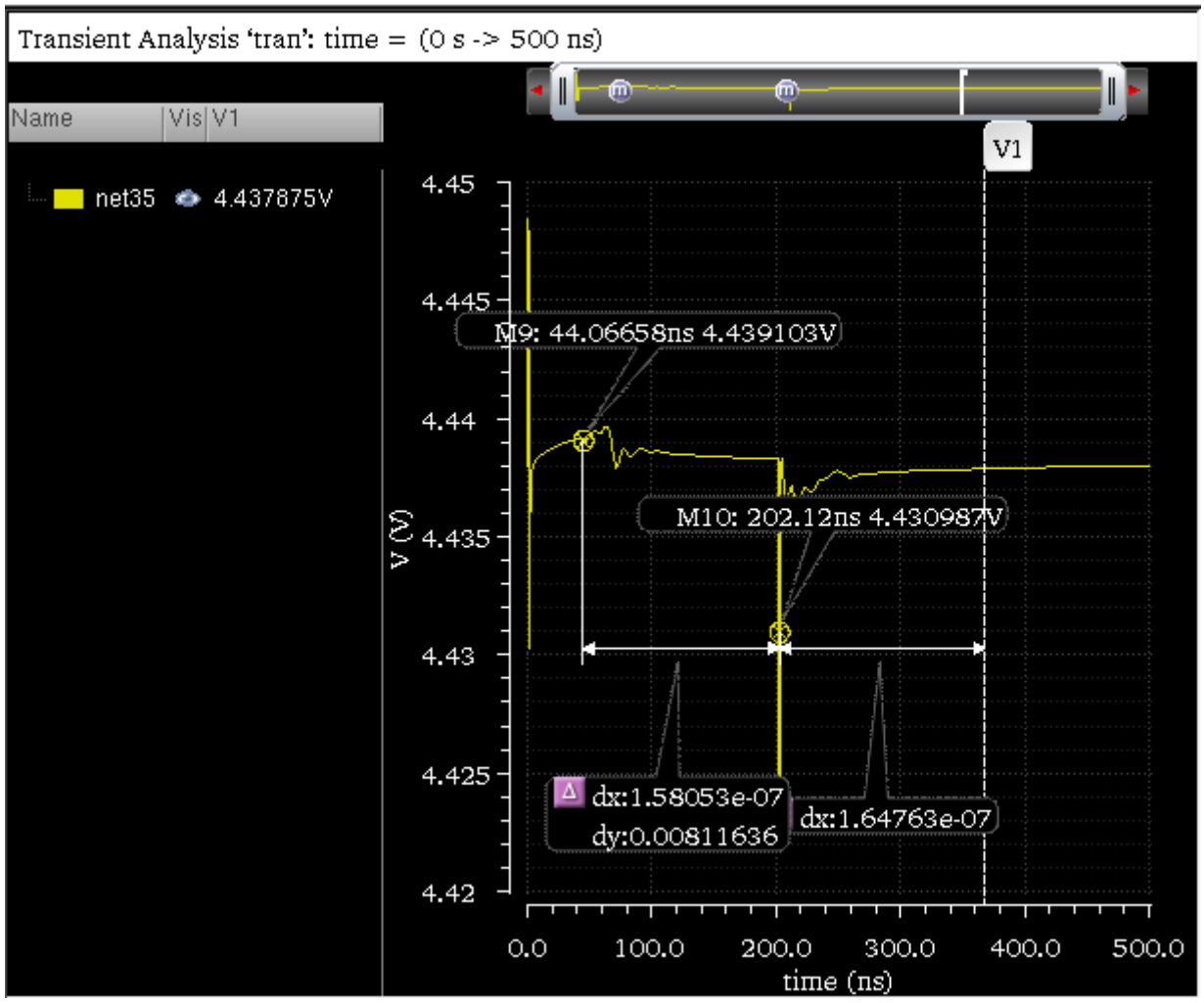
Method 1:

1. Add two or more point markers. To know how to create a point marker, see Adding a Point Marker on page 552.
2. Select all the point markers by holding down the `Ctrl` key and do one of the following:
   - Press the bindkey `Shift+D`.
   - Choose *Marker – Create Delta Marker*.

   The delta markers are created between all the selected point markers. The method can be applied to create delta markers between any combinations of point, vertical, and horizontal markers.

© 2023 Cadence Design Systems, Inc.

Note: You can add point, vertical, horizontal, delta, and AB markers in the eye diagram and spectrum plots. However, when you add a vertical or a horizontal marker in the eye diagram, the intercepts are not displayed in the plot.



Method 2:

1. Create a point marker. To know how to create a point marker, see Adding a Point Marker on page 552.

   The point marker you created remains selected.

2. Place the mouse pointer on the trace where you want to create the second marker. You can create delta markers on multiple traces.

3. Press the bindkey `D`.

© 2023 Cadence Design Systems, Inc.

A new point marker is created at the same point where you placed the mouse pointer and a delta between this new marker and the previously created point marker is also created. This new marker is of the same type as the marker type of the previously selected marker. For example, if you created a point marker in step1, the new maker created after step2 is also a point marker.

You can repeat this method to create delta markers between multiple point markers. The last marker that you create remains selected.

Repeat steps 2-3 to create delta marker between two or more vertical or horizontal markers.

Note: If the two markers in a delta marker have the same Y-axis units, the delta marker label displays the dx, dy and slope values. On the other hand, if the Y-axis units are different, the delta marker label displays only the dx value.

Moving a Delta Marker

To move a delta marker, you can set the snap criterion on a point marker in the delta marker. You can then use the *Next Edge* and *Previous Edge* buttons to move the selected point marker in the delta marker.

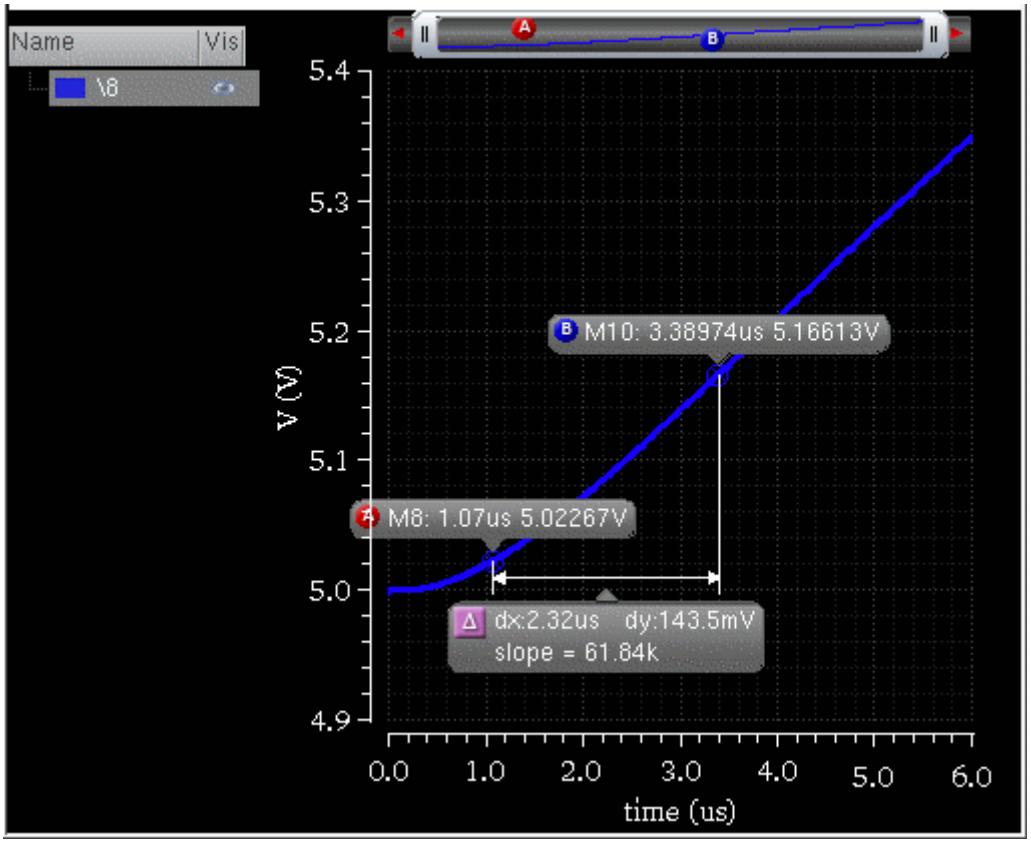
Adding AB Marker

AB marker is a delta marker of XY type and displays the `dx`, `dy`, and the `slope` values. Do the following to add an AB marker to the trace with the help of bindkeys:

1. Move the mouse pointer to a location on the trace where you want to create an AB marker.
2. Press `A`.
3. Move the mouse pointer to another point on the trace to specify the second location for the delta marker.
4. Press `B`.

© 2023 Cadence Design Systems, Inc.

A delta marker of XY type appears on the graph. If one of the traces is a digital trace, the delta marker label displays only the `dx` value.



You can add multiple AB markers by repeating these steps.

Note: To convert an AB delta marker into a delta marker, right-click the delta marker line and choose *Convert A/B Marker to Delta*.

Displaying Marker Labels in Delta Markers

To show or hide the marker labels for the point markers in a delta marker, do one of the following:

- Select the delta line and choose *Marker – Show Child Labels*.
- Right-click the delta line and choose *Show Child Labels.*

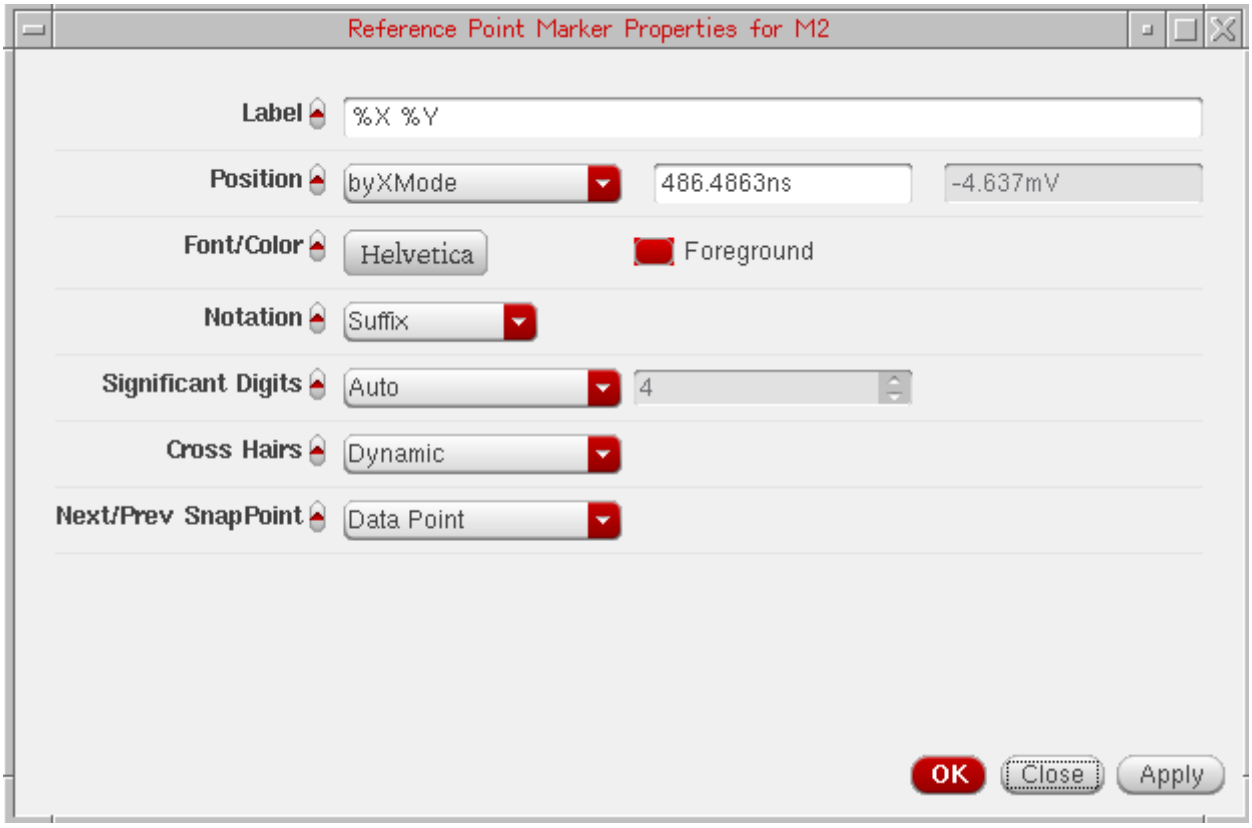
Setting AB Delta Marker Reference Point Properties

To view or change the properties of an AB reference point marker, do one of the following:

© 2023 Cadence Design Systems, Inc.

- Double-click any of the two point markers that compose the delta marker.
- Right-click a point marker in the delta marker and choose *Marker – Properties*.

  The *Reference Point Marker Properties for <marker name>* form appears.



This form includes the following fields:

- *Label*—Specify the label for the marker. See Marker Labels for more information.
- *Position*—Specify the X and Y coordinates of the reference point marker. You can specify the position by XY, X, and Y modes. By default, the position is set to `byXMode`.
- *Font/Color*—Specify the font type and font color for the marker label.
- *Foreground*—Select the foreground color for the symbol, the arrow and the label.
- *Notation*—Specify the numerical format (notation) to be displayed on labels. The available options are—`Suffix`, `Engineering`, and `Scientific`. Default value: `Suffix`.

© 2023 Cadence Design Systems, Inc.

- *Significant Digits*—Specify the number of significant digits if you select `manual` from the drop-down. The other option is `auto`.
- *Cross Hairs*—Specify whether you want to display the cross hairs for point marker or delta marker. If you select *Dynamic* from the *Cross Hairs* drop-down list box and click the reference point marker, the cross hairs are displayed.

  Alternatively, to turn on the cross hairs for a delta marker, right-click one of the reference point markers and choose *Draw Cross Hairs – On/Off/Dynamic*.
- *Next/Prev SnapPoint*—Specify the criterion based on which the selected reference point marker should be snapped.

  The available options are—`Local Maxima`, `Local Minima`, `Local Max or Min`, `Specific Y Value`, `Specific X Value`, `Data Point`, `Global Maxima`, and `Global Minima`.

  The `Specific X Value` option is not available for horizontal markers.

  Default value: `Data Point`.

Deleting Delta Markers

To delete the delta marker, right-click the line joining two point markers and choose *Delete*. All the delta markers are deleted.

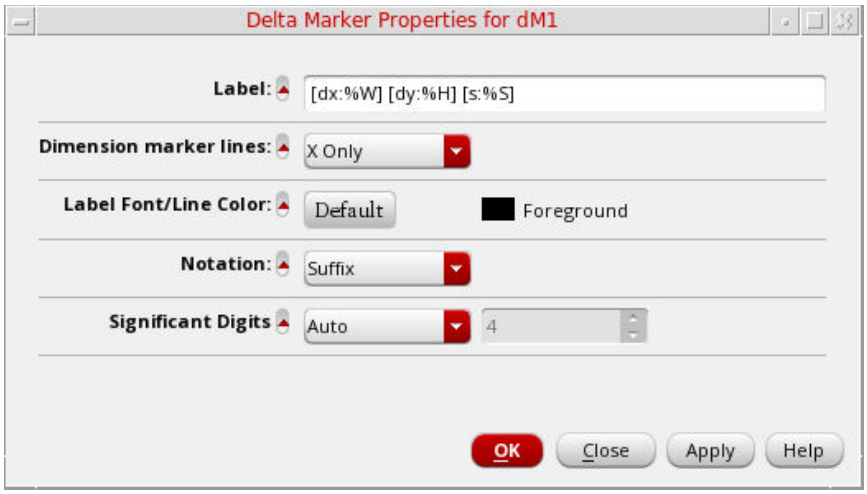
To delete a particular set of delta marker, right-click the point marker which you want to delete and choose *Delete*. The selected point marker and the delta marker joining the point marker are deleted.

Editing Delta Marker Properties

To view or change the delta marker properties, do the following:

© 2023 Cadence Design Systems, Inc.

➡ Right-click the delta marker line joining two or more point markers and choose *Delta Marker Properties*. The *Delta Marker Properties for <delta marker name>* form appears.



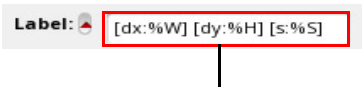
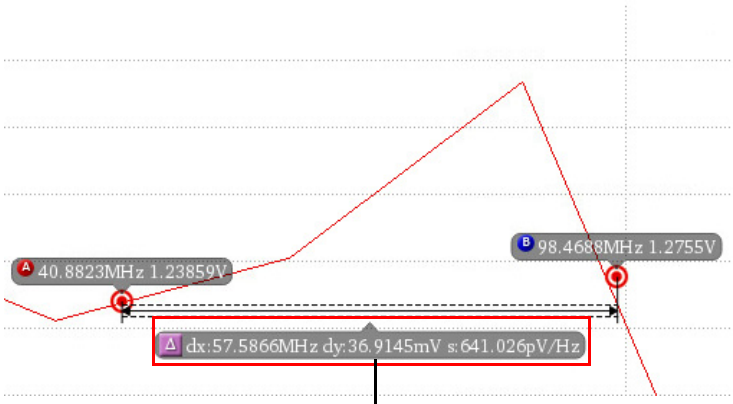
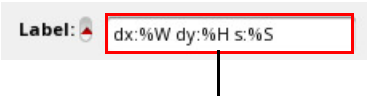
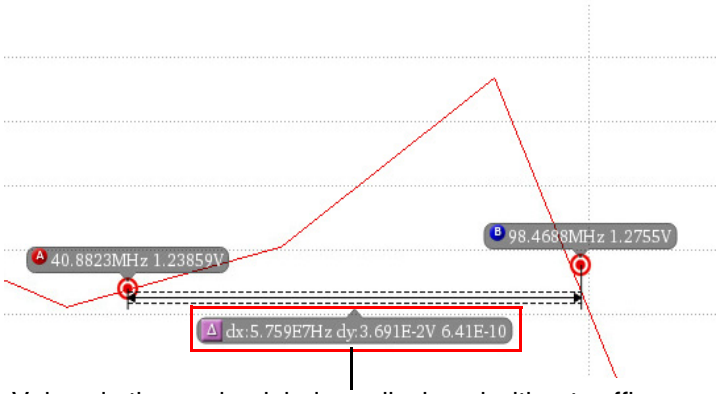
The form includes the following fields:

- *Label*—Specify the label for delta marker. The default is `[dx: %W] [dy: %H] [s: %S]`, where,
  - `dx:%W`—Delta value on X-axis
  - `dy:%H`—Delta value on Y-axis
  - `s:%S`—Slope (dy/dx)

  Note: If you do not enclose these labels in parenthesis `[ ]`, the corresponding information is displayed without suffix notations in the marker label.

  The following table shows how values are displayed in the marker labels of a delta marker when format strings are specified with and without parenthesis, `[]`.

© 2023 Cadence Design Systems, Inc.

Format String	Marker Label
 Format strings with `[]`	 Values in the marker label are displayed with suffix notation.
 Format strings without `[]`	 Values in the marker label are displayed without suffix notation.

For more information about the values that you can use for marker labels, see Marker Labels.

- *Dimension Marker Lines*—Specify the X and Y-axis dimension lines for markers. To display the X-axis dimension line, select `X Only`. To display the Y-axis dimension line, select `Y Only`. To display both the X and Y-axis dimension lines, select `X and Y`.
- *Label Font/Line Color*—Specify the font properties for the label and also specify the dimension line color.

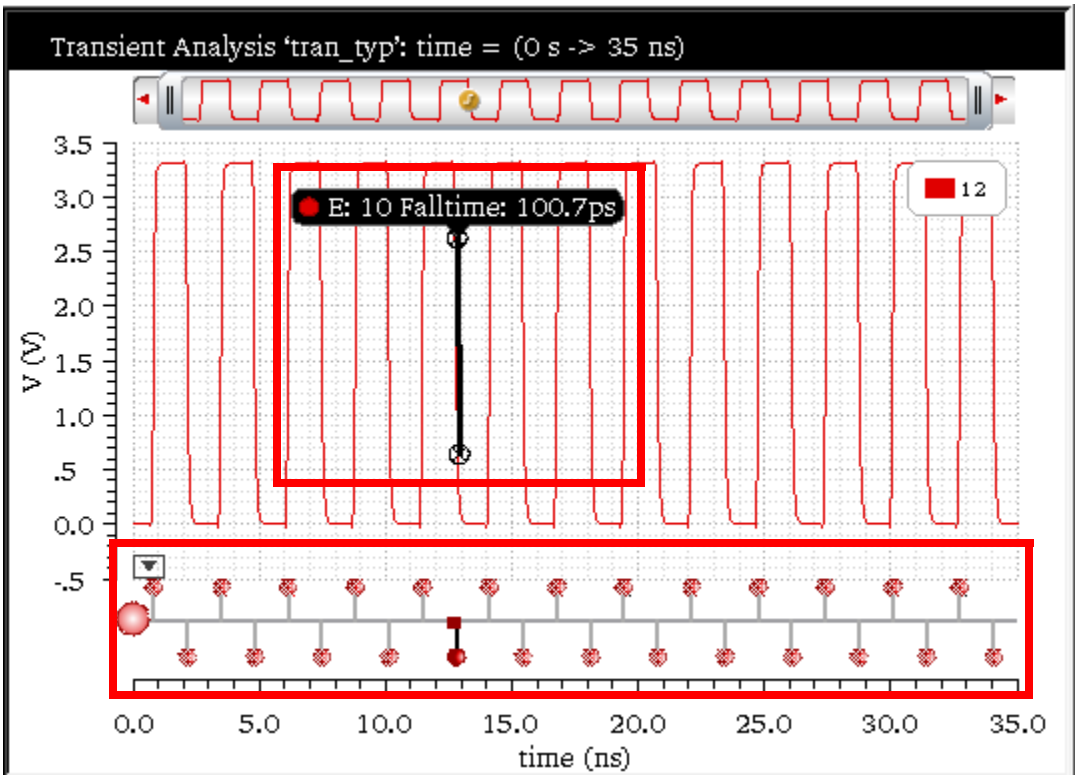
© 2023 Cadence Design Systems, Inc.

- *Notation*—Specify the notation to be displayed for the delta values on labels. The available options are—`Suffix`, `Engineering`, and `Scientific`.
- *Significant Digits*—Specifies the number of significant digits to be displayed in the calculated values. The default value is `4`.

© 2023 Cadence Design Systems, Inc.

Working With Edge Markers

Edge Markers are special markers that can be attached to the rising or falling edges of a trace to measure the transient properties of the selected edge. The figure below displays an edge marker placed on the rising edge of a trace. The marker label displays the risetime value for this edge. If you place the edge marker on the falling edge of the trace, the marker label displays the falltime value for that edge.



The figure above also displays the Edge Browser at the bottom of the strip that you can use to view and analyze the various edges in the trace. By default, the Edge Browser is hidden in the graph. For more information about how to use Edge Browser, see Using Edge Browser on page 312.

This section includes the following topics:

- Creating an Edge Marker on page 532
- Setting Edge Marker Properties on page 533
- Edge Marker Context-Sensitive Menu on page 535
- Creating a Delta Marker between Edge Markers on page 537

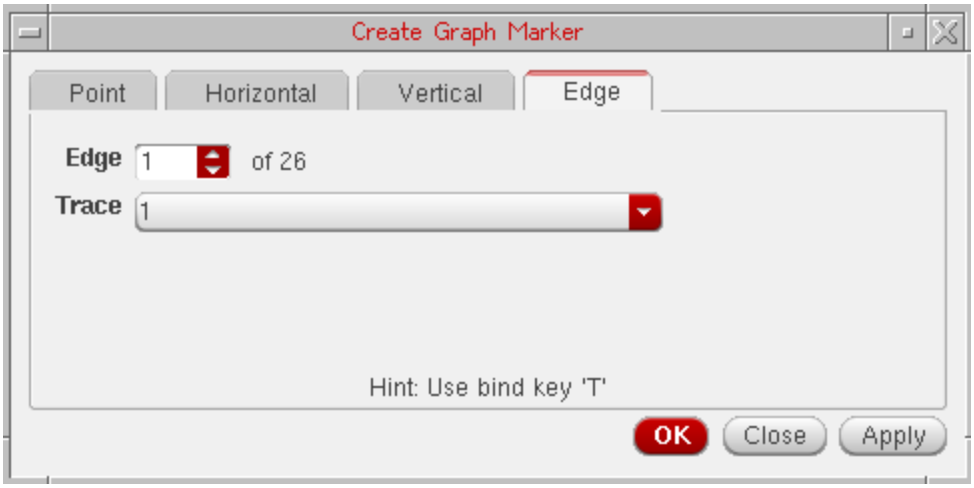
© 2023 Cadence Design Systems, Inc.

Creating an Edge Marker

You can create an edge marker by using one of the following methods:

- Choose *Marker – Create Marke*r.
- Click the button on the Marker toolbar.
- Use the Transient Measurement assistant to create an edge marker. For more information about transient measurement, see Transient Measurement Assistant on page 308
- Right-click the Edge Browser and choose *Create Edge Marker*.

  The *Create Graph Marker* form appears, as displayed in the figure below.



  In this form, on the *Edge* tab, specify the following fields:

  - *Edge*—Specify the edge number where you want to create an edge marker. The total number of edges are displayed next to this field.
  - *Trace*—Select a trace from the drop-down on which you want to place the edge marker. This drop-down lists the name of all traces that are plotted in the selected window.
  - Click *OK*.

  An edge marker is placed at the specified edge on the selected trace.

- Place the mouse pointer on the edge where you want to create an edge marker and press the bindkey `T`.
- Right-click an edge in the Edge Browser and choose *Create Edge Marker*. The edge marker is placed at the selected edge of the trace in the graph window.

© 2023 Cadence Design Systems, Inc.

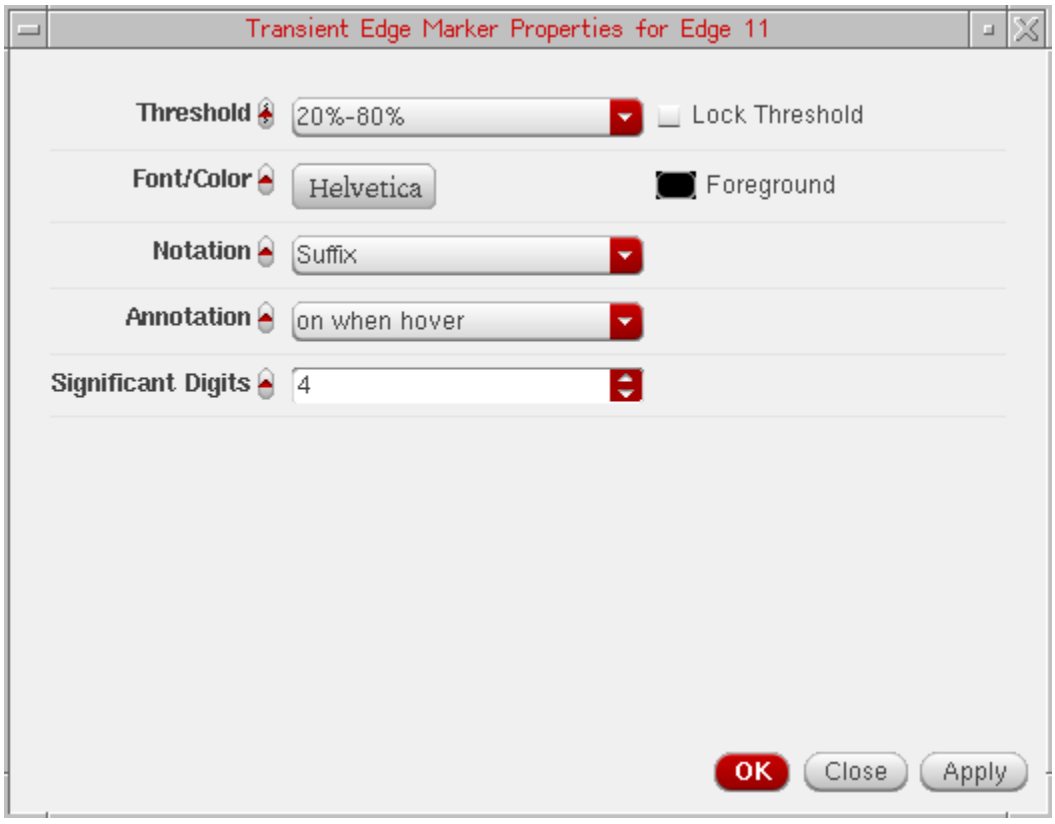
- In the Transient Measurement assistant, select the edge on which you want to place the edge marker from the *Edge* drop-down and click the button. For more details, see Transient Measurement Assistant on page 308.

Setting Edge Marker Properties

Do the following to set the properties for the edge markers:

- Right-click the edge marker and choose *Edge Marker Properties*.
- Select the edge marker and choose *Marker – Properties.*
- Select the edge marker and press the bindkey `Q`.

  The *Transient Edge Marker Properties for Edge <edge number>* form appears



This form includes the following fields:

- *Threshold*—Specify the threshold value. The available options are: *10%-90%*, `20%-80%`, and `30%-70%`. The default value is *20%-80%*. The threshold value is controlled by the following environment variable:

© 2023 Cadence Design Systems, Inc.

```
envSetVal("viva.trace" "threshold" 'string "20_80")
```

- *Lock Threshold*—Select this check box to lock the threshold value so that the threshold setting for an individual marker is not overridden if you change the settings of the trace.
- *Font/Color*—Specifies the font and color of the edge marker.
- *Notation*—Specifies the notation of the edge marker. The valid values are `Scientific`, `Engineering`, and `Suffix`. The default value is `Suffix`.
- *Annotation*—Specifies the event when to display the risetime and falltime values on the marker labels. The available values are `off`, `on when hover`, and `on`. The default value is `on when hover`.
- *Significant Digits*—Specifies the number of significant digits to be displayed in the calculated values. The default value is `4`.

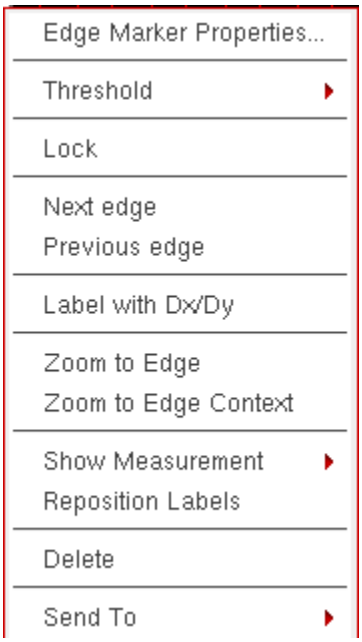
Important

When you are saving a trace with edge markers as an image file by using the *File – Save Image* option, ensure that the *Annonation* field is set to `on`. Otherwise, the edge marker labels will not be displayed in the saved image file.
You can also

© 2023 Cadence Design Systems, Inc.

Edge Marker Context-Sensitive Menu

Right-click the edge marker to use the various options listed in the context-sensitive (shortcut) menu. The following figure displays the available menu options:



The shortcut menu options are explained as below:

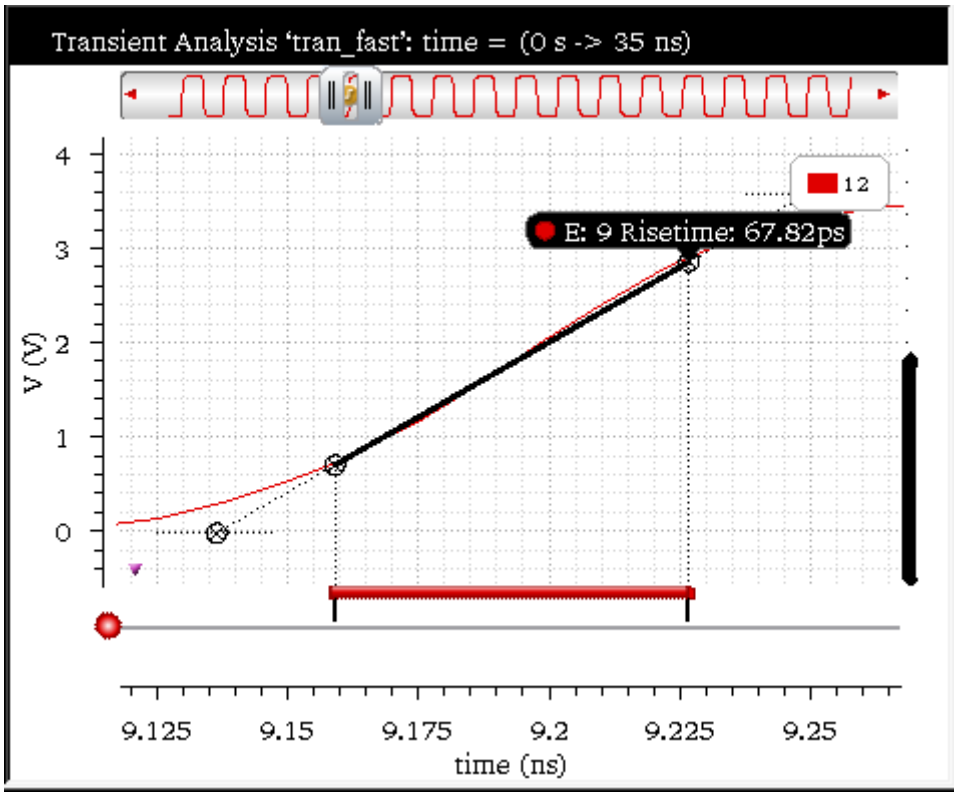
- *Edge Marker Properties*—Opens the Edge Marker properties form, where you can specify the marker properties for the selected marker. For more information on marker properties, see Setting Edge Marker Properties on page 533.
- *Threshold*—Specifies the threshold value for the selected edge marker. The available options are: *10%-90%*, `20%-80%`, `30%-70%`, and `default`. The default value is *20%-80%*. The threshold value is controlled by the following environment variable:

  ```
  envSetVal("viva.trace" "threshold" 'string "20_80")
  ```

- *Lock*—Locks the threshold value so that the threshold setting for an individual edge marker is not overridden if you change the settings of the trace.
- *Next Edge*—Moves the selected edge marker to the next edge on the trace.
- *Previous Edge*—Moves the selected edge marker to the previous edge on the trace.
- *Label with Dx/Dy*—Creates a delta marker by adding two point markers and a delta line between them.

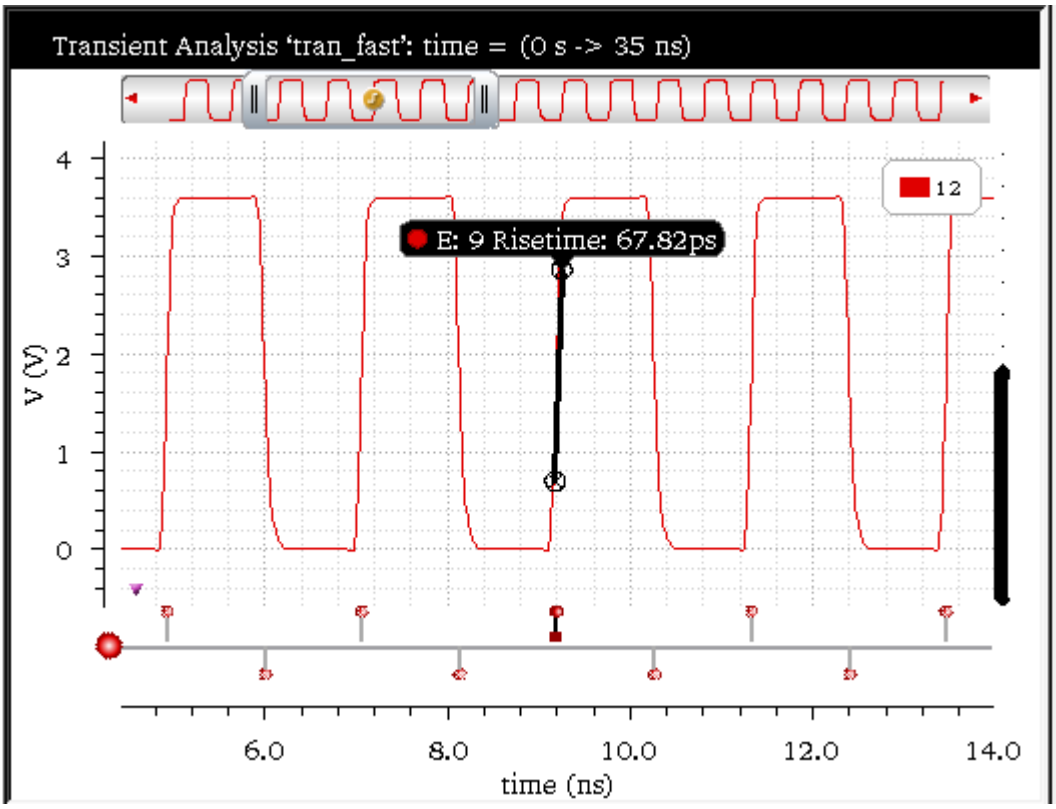
© 2023 Cadence Design Systems, Inc.

- *Zoom to Edge*—Zooms in the selected edge marker to view the details of the edge on which it is placed. The following figure illustrates the zoomed-in graph.



© 2023 Cadence Design Systems, Inc.

- *Zoom to Edge Context*—Zooms in the selected edge marker to the context of the edge on which it is placed. The following figure illustrates the zoomed-in graph.



- *Show Measurement*—Specifies the event on which you want to show the measurement label for the selected edge marker. The available options are `off`, `on when hover`, and `on`. You can also set the following environment variable to specify an event:

  ```
  envSetVal("viva.transEdgeMarker""showMeasSummary"'string "on when hover")
  ```

- *Reposition Labels*—Moves the selected marker label back to its default position.
- *Delete*—Deletes the selected edge marker.
- *Send To*—Sends the edge marker measurements, such as slewRate, overshoot, undershoot, risetime and falltime to ADE or Calculator.

Creating a Delta Marker between Edge Markers

To create delta markers between two or more edge markers on a single or multiple traces, perform the following steps:

© 2023 Cadence Design Systems, Inc.

1. Create an edge marker using the bindkey `t` and then select this edge marker.
2. Place the pointer close to a rising or falling edge on the same or another trace and press the bindkey `d`.

   A new edge maker is created at the selected point and a delta marker is created between the two edge makers. You can also select more than one markers and repeat these steps to create further edge markers and delta markers.

Alternatively, to create delta markers between two or more edge markers on a single or multiple traces, perform the following steps:

1. Create an edge marker on the first trace using the bindkey `t`.
2. Create an edge marker on the second trace using the bindkey `t`.
3. Now, select both the edge markers using `Ctrl+click` and press `Shift+D`.

   A delta maker is created at between the selected edge markers.

© 2023 Cadence Design Systems, Inc.

Working with Circular Graphs

You can display complex data values from AC analysis in the form of Smith charts and polar plots. The Smith chart shows the unity circle that is R = 1 circle, the resistance circles, and the reactance circles.

This section includes the following topics:

- Creating a Circular Graph on page 539
- Adding Markers on Circular Graphs on page 548
- Snapping Markers on Circular Graphs on page 554
- Zooming Circular Graphs on page 557
- Editing Circular Graph Properties on page 557
- Filtering Data Using the Frequency Slider Bar on page 559
- Setting Smith Grid Properties on page 562

Creating a Circular Graph

To create a circular graph, do the following:

1. Open a results directory in the Results Browser, and then open the *ac-ac* analysis directory.

   The *Graph Type* drop-down list appears on the Results Browser toolbar.

2. Select a graph type from this list. The available graph types are—`Default`, `Rectangular`, `Polar`, `Impedance`, `Admittance`, `Immittance`, and `RealvsImag`.

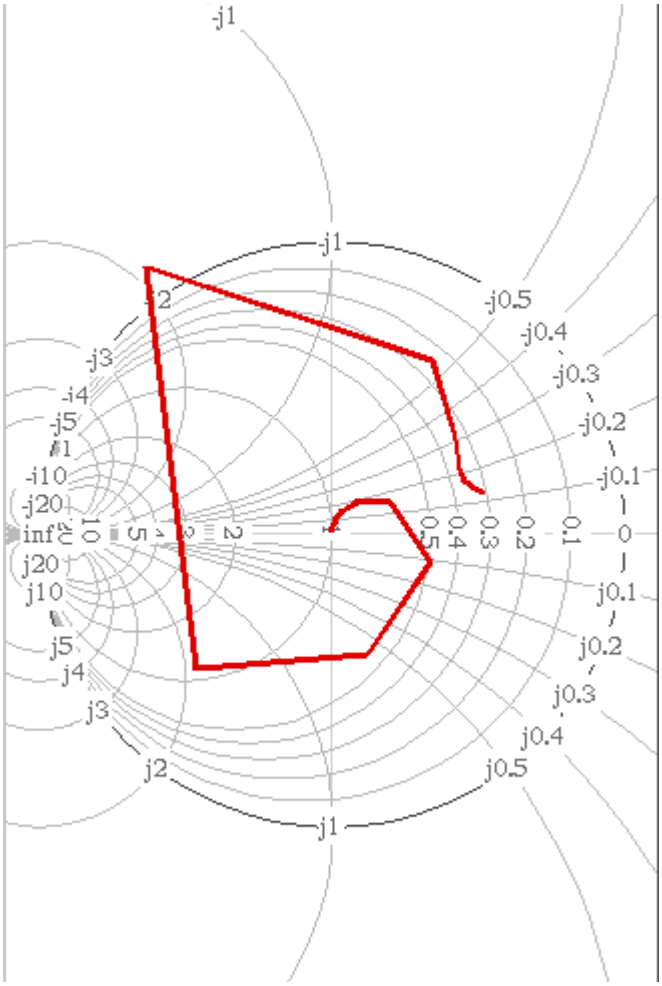
   - To create a polar plot, select the `Polar` graph type.
   - To create a Smith chart, select either `Impedance`, `Admittance`, or `Immittance` graph type.

3. Plot a signal in the selected destination. For more information about how to plot a signal, see Plotting a Signal from the Results Browser.

   The circular graphs can be of two types—Smith charts and polar graphs. The Smith chart can further be of three types—Impedance Smith Charts (Z Smith), Admittance Smith

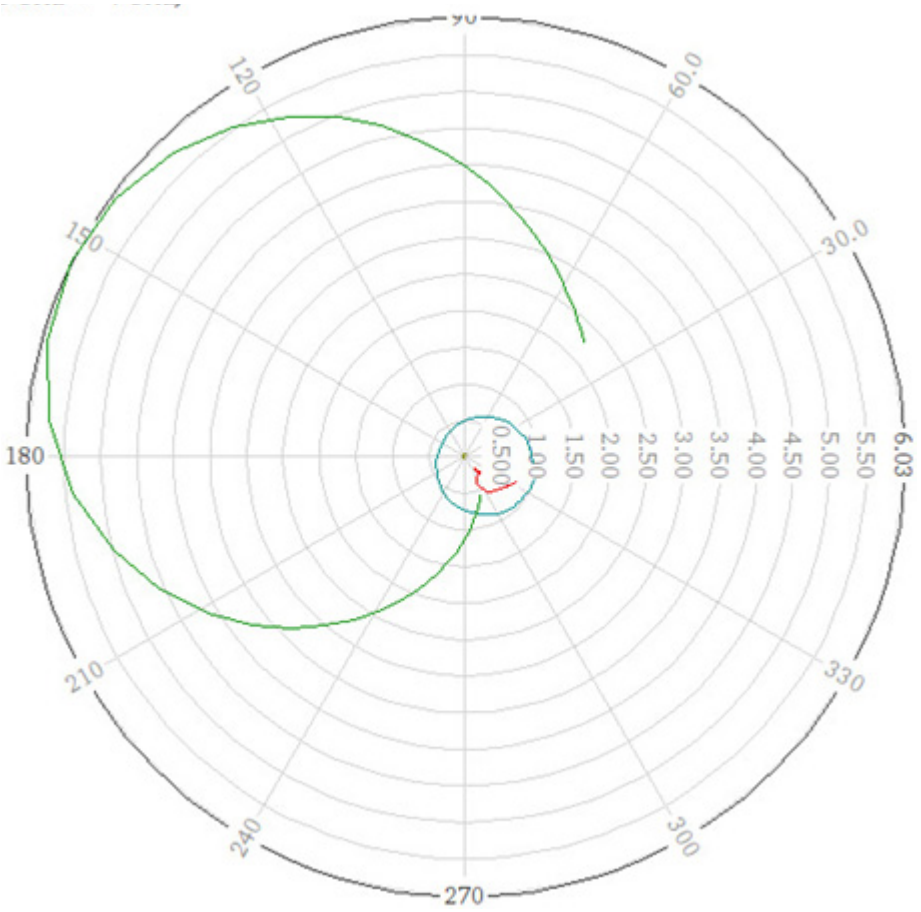
© 2023 Cadence Design Systems, Inc.

Charts (Y Smith), and Immittance Smith Charts (Combined Z Smith and Y Smith). The circular graph of type Z Smith is displayed in the figure below.



© 2023 Cadence Design Systems, Inc.

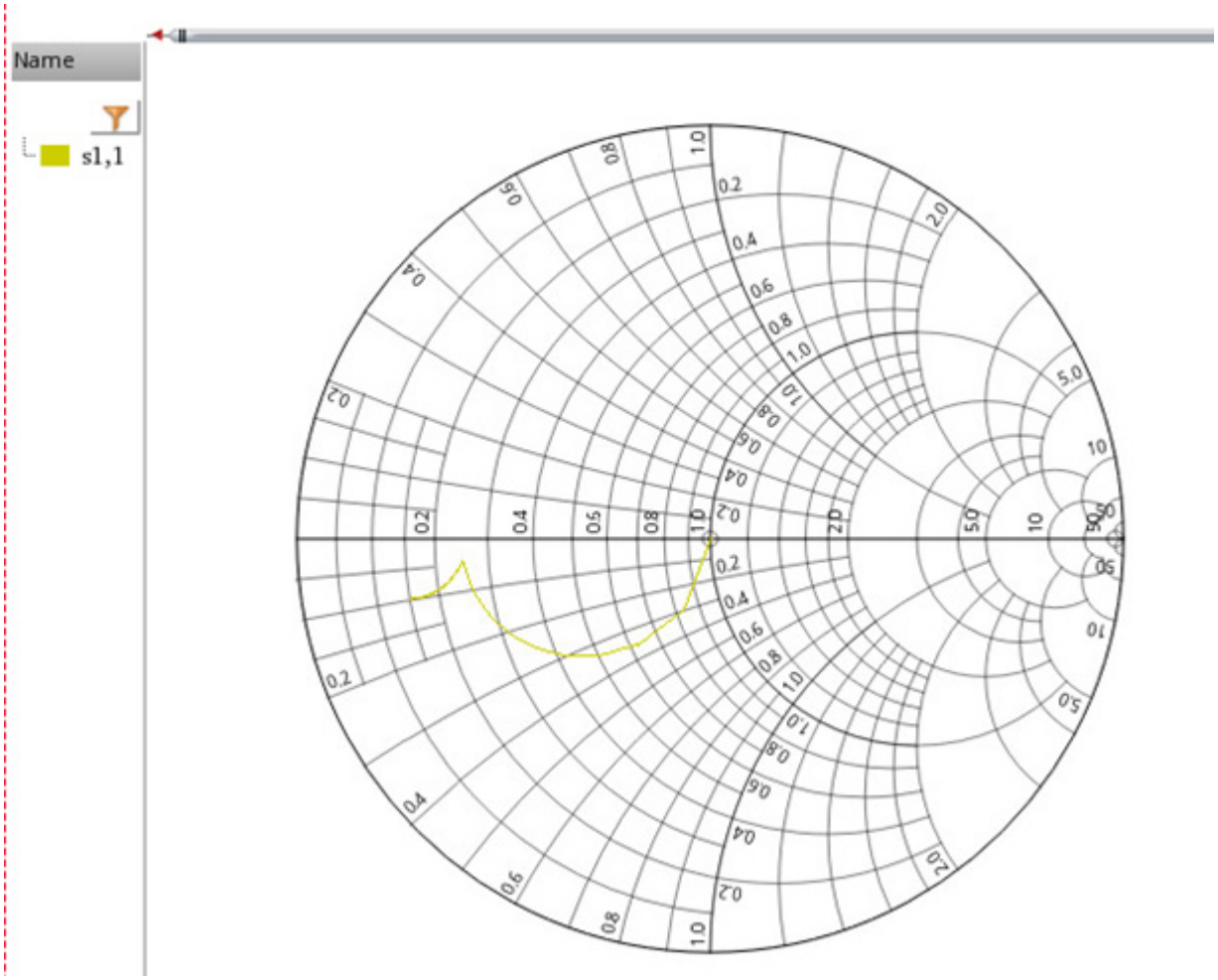
Polar Plot:



© 2023 Cadence Design Systems, Inc.

Type of Smith Charts

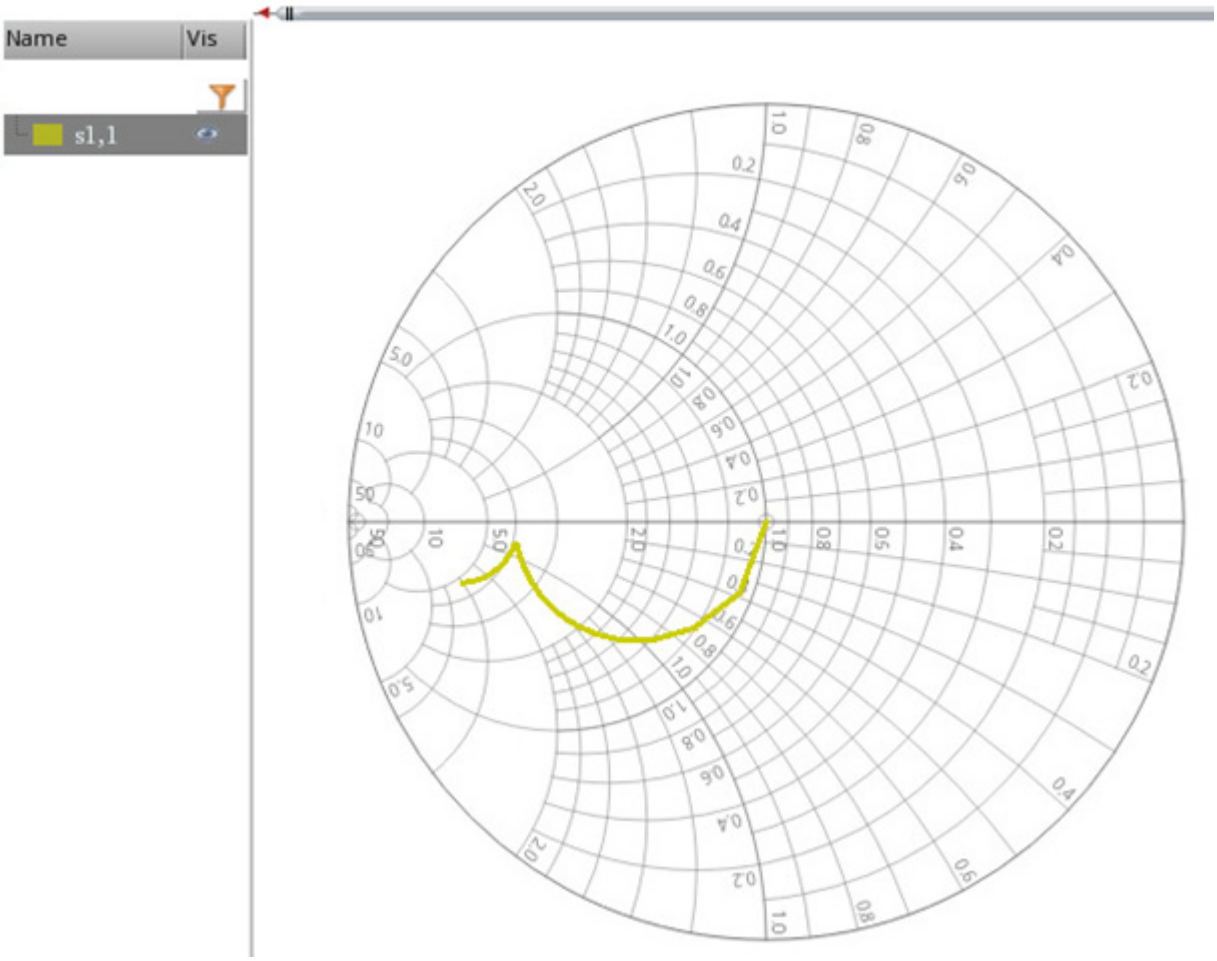
- Impedance or Z Smith: In the Z Smith chart, the region above the x axis represents inductive impedances (positive imaginary parts) and the region below the x axis represents capacitive impedances (negative imaginary parts).



- Admittance or Y Smith: In the Y Smith chart, the region above the x axis represents capacitive admittances and the region below the x axis represents inductive admittances. Capacitive admittances have positive imaginary parts and inductive admittances have negative imaginary parts.

© 2023 Cadence Design Systems, Inc.

The Y Smith chart appears like a normalized Impedance chart, but with the graphic scaling rotated through 180° and the numeric scaling remains unchanged.

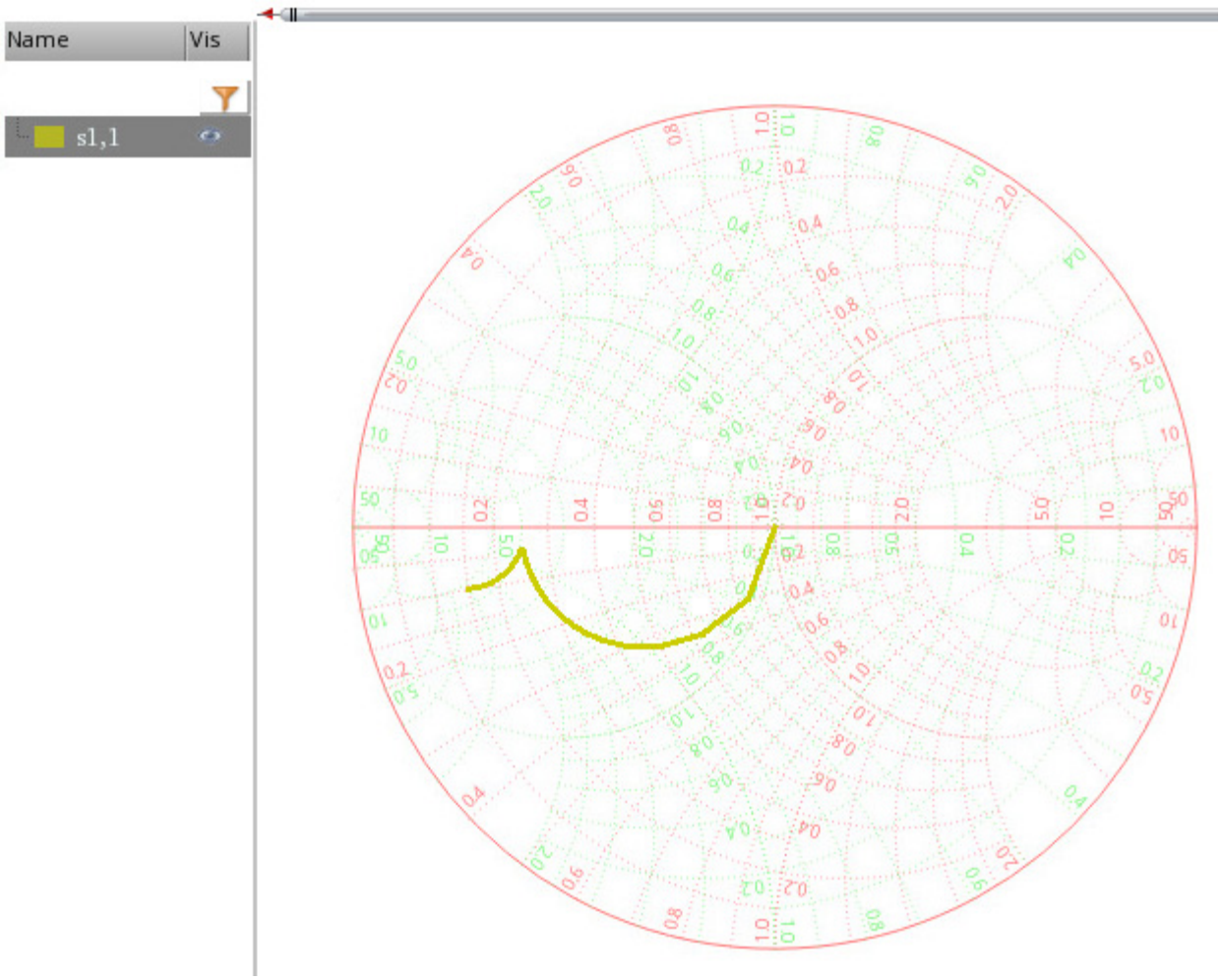


- Immittance: In the Immittance Smith chart (ZY Smith chart), both Z Smith and Y Smith grids are plotted on the same chart. The Immittance Smith chart has two grids: red and green. The red grid represents the Impedance or Z Smith grid, whereas the green grid represents the Admittance or Y Smith grid.

  On the red grid, the region above the x axis represents inductive impedances (positive imaginary parts) and the region below the x axis represents capacitive impedances (negative imaginary parts).

© 2023 Cadence Design Systems, Inc.

On the green grid, the region above the X-axis represents capacitive admittances (positive imaginary parts) and the region below the X-axis represents inductive admittances (negative imaginary parts).



Note: You can use the bindkey O to switch between the Impedance, Admittance, and Immittance Smith Charts.

After a Smith chart is plotted, the following sections are displayed in the trace legend area on the left:

- *Name*—Displays the trace name and trace color.
- *Visibility*—Controls the display of the trace. You can show or hide the trace by clicking the visibility button ( ).
- *Tracking Info*—Displays the following tracking information for the points that you click or point to:

© 2023 Cadence Design Systems, Inc.

- *Zo*—Displays the characteristic impedance (`Zo`). The default `Zo` value is 50.
- *Normalize*—Normalizes the readings. When you select the *Normalize* check box, the Smith reference values are multiplied with the impedance value that you specify in the `Zo` field.

  Note: When you change the `Zo` value and the *Normalize* check box is not selected, press `Tab` or `Enter` to update the reference values on the Smith chart.
- *Real + Imag*—Displays the real and the imaginary values of the point selected on the trace
- *Gamma* —Displays the magnitude and the angle of the selected point on the trace.
- *Gamma dB20*—Displays the magnitude (`dB20`) and the angle of the selected point on the trace.
- *Z*—Displays the impedance of the point selected on the trace.
- *Y*—Displays the admittance of the point selected on the trace.
- *VSWR*—Displays the Voltage Standing Wave Ratio (VSWR). The VSWR is a scalar value.
- *Frequency*—Displays the frequency of the point selected on the trace.
- *Data Index*—Displays the index value of the nearest data point on the trace.

■ *Reference point values*—Displays the reference marker readout. The reference point readout includes *Real + Imag*, *Gamma*, *Zd* and *Yd* values of the reference point, and the Voltage Standing Wave Ratio (*VSWR*).

The resistance and reactance circles, along with the reflection coefficient and VSWR circle, are displayed on the graph. You can toggle these values by clicking the respective *visible* check boxes next to each field.

The default scale attributes for circular graphs are controlled by the values assigned to the variables in the `.cdsenv` file. For more information, see Appendix A, “Virtuoso Visualization and Analysis XL Tool Environment Variables.”

If you plot circular graphs that have the same plot type, the circular graphs are plotted in the same subwindow. For example, two Y-Smith or two Z- Smith can be plotted in the same window. However, if you plot graphs from different plot types, the graphs are plotted in a new subwindow. For example, Y-Smith and Z-Smith are plotted in two different subwindows.

© 2023 Cadence Design Systems, Inc.

Tracking Cursor

The tracking cursor displays the trace name, trace color, and frequency of the point you that you select or point to on the circular graph. For a Smith chart, the tracking cursor also displays the real and imaginary values. However, for polar plots, the tracking cursor displays the magnitude and angle, instead of the real and imaginary values.

The rest of the circular graph values are displayed dynamically in the various sections in the trace legend area.

To show or hide the tracking cursor:

1. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Marker – Tracking Cursor*.
2. Right-click anywhere in the graph and choose *Trace Marker Always Visible*.

You can also set the `traceMarkerDisplay` .cdsenv variable to display values for the tracking cursor, which is also called a trace marker. The following formats are supported to display the trace marker values:

- `%C` —Displays the real and imaginary Cartesian values.
- `%Z`—Displays the impedance values, such as resistance and reactance.
- `%A`—Displays the admittance values, such as conductance and susceptance.
- `%R`—Displays the reflection coefficients, such as mag and angle.
- `%P`—Displays the polar values, such as mag and angle.
- `%F`—Displays the frequency value, which includes the independent axis data.

The trace marker or tacking cursor is always visible on the graph only if both the *Tracking Cursor* and the *Trace Marker Always Visible* commands are enabled.

When the *Tracking Cursor* command is enabled but the *Trace Markers Always Visible* command is disabled, the trace marker is displayed only when you point the mouse pointer onto the trace.

With the *Tracking Cursor* command enabled, if you enable the *Trace Marker Always Visible* command again, the trace marker remains visible at its last location on the trace even if the mouse pointer is away from the trace.

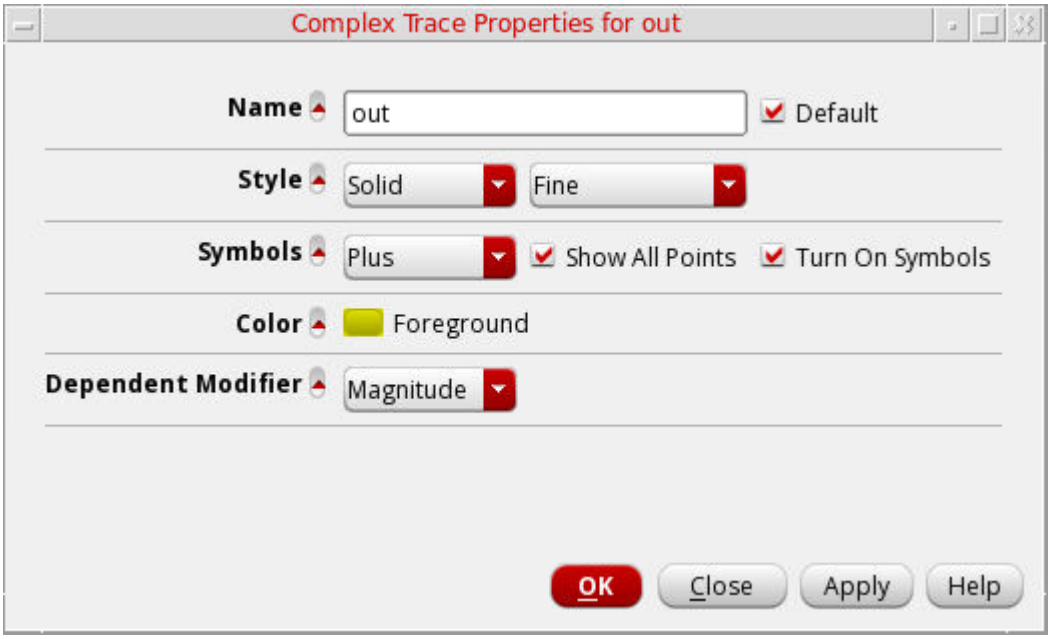
Displaying Symbols on Circular Traces

To display symbols on a circular trace, do one of the following:

© 2023 Cadence Design Systems, Inc.

- Choose *Trace – Symbols On.*
- Right-click the trace and choose *Symbols On*.
- Right-click the trace and choose *Trace Properties*.

  The *Complex Trace Properties for <trace-name>* form appears.



In this form, select the *Turn On Symbols* check box to display symbols on the trace. By default, the symbols are displayed for all trace points. To display symbols for a given number of trace points, specify the count in the *Points per Symbol* field.

After turning on the symbols, you can select the symbol type by doing one of the following:

- Right-click the trace and choose *Symbols,* and then select the symbol type that you want to apply.
- Right-click the trace and choose *Trace Properties*.

  The *Complex Trace Properties for <trace-name>* form appears. In this form, select the symbol that you want to display for the data points on the trace from the *Symbols* drop-down list.
- In the *Color* field, select the foreground color for the trace. Alternatively, you can also set the trace color by right-clicking the trace and selecting *Color*.

© 2023 Cadence Design Systems, Inc.

You can also set the dependent modifiers for an AC or a complex dataset, such as `Magnitude`, `dB10`, `dBm`, and `dB20`. See the Setting Dependent Modifiers for a Complex Trace for more information.

Adding Markers on Circular Graphs

You can add reference, point, circular, and delta markers on circular graphs. By default, the markers in the circular graphs are always visible. To hide the markers, you can set the `tracemarkeralwaysvisible .cdsenv` variable to false.

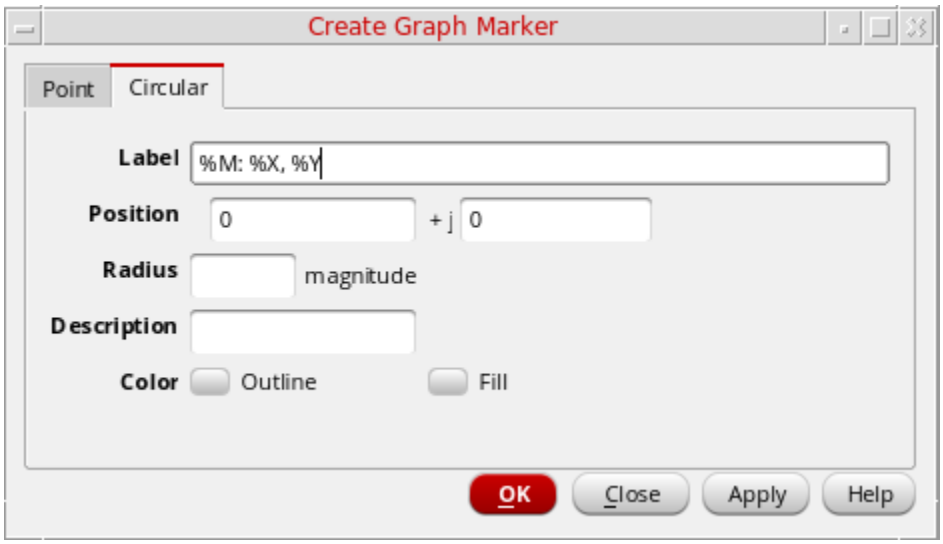
Adding a Circular Marker

To add a circular marker on polar plots or Smith graphs,

1. Choose *Marker — Create Marker*.

   The Create Graph Marker form appears.

2. Open the *Circular* tab. The form fields corresponding to circular marker appear, as shown in the figure below.



3. Specify the following fields:

   - *Label*—Specify a label for the marker. See Marker Labels for more information.

     Note: If you do not enter text in the *Label* field, the X and Y coordinates of the marker are displayed.

© 2023 Cadence Design Systems, Inc.

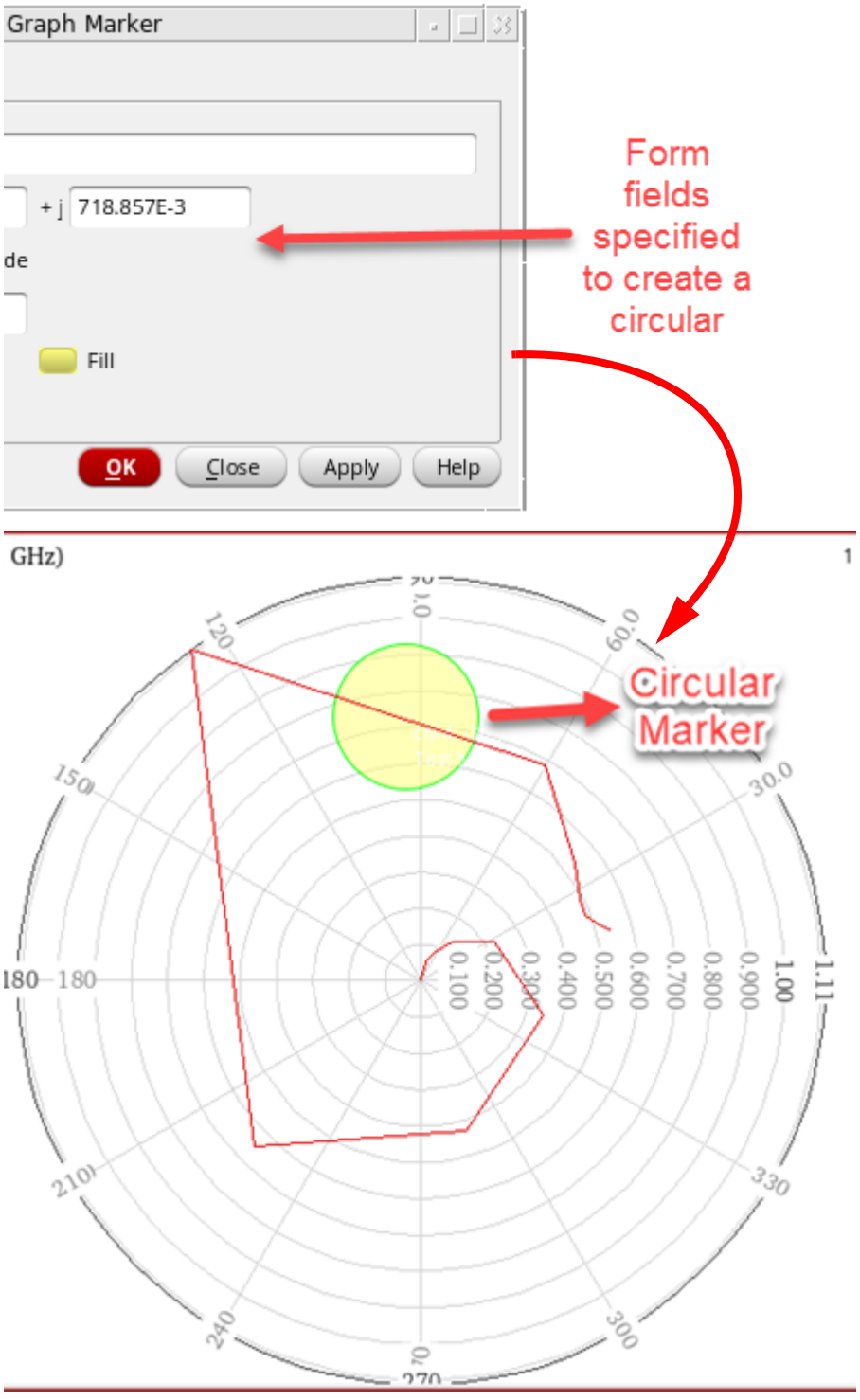
- *Position*—Type the real and imaginary values (`x+ij`) where you want to add the circular maker. Alternatively, you can select a point on the graph where you want to add the circular marker. The real and imaginary values of the point you select are automatically added in the *Position* field.
- *Radius*—Specify the radius (magnitude) of the circle.
- *Description*—Specify an additional information about the circular marker. The description text is displayed when you move the pointer on the marker.
- *Color*—Specify the marker outline and fill color.

4. Click *OK*.

   The circular marker according to the specified values is created on the circular graph.

© 2023 Cadence Design Systems, Inc.

The figure below shows an example of the circular graph added on a polar plot.



© 2023 Cadence Design Systems, Inc.

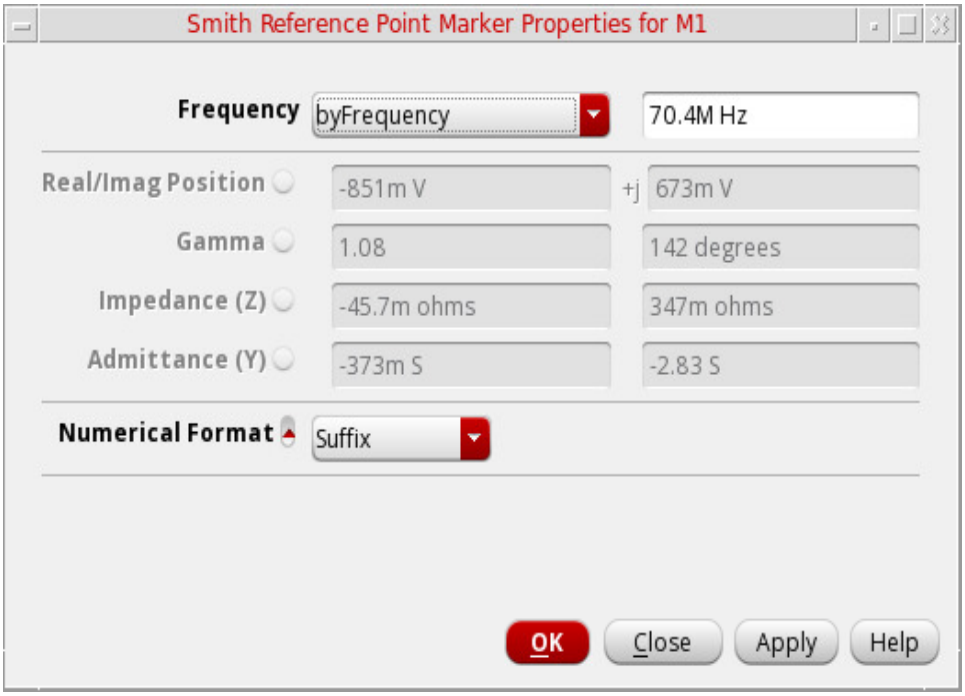
Adding a Smith Reference Point Marker

To add a Smith reference point maker on a circular graph, select a point on the trace and press the bindkey `R`. The reference maker is created from the center to the selected point. The readout for the reference point marker—`Real + Imag`, `Gamma`, `Zd`, `Yd`, and `VSWR`—is displayed in the *Reference point values* section in the trace legend area.

Note: You can add multiple reference markers on the circular graph. When you select a reference marker, the reference point values and graphical measurements change based on the circles and annotation for the selected reference point.

Setting Smith Reference Point Marker Properties

To set the reference marker properties, right-click a reference marker and choose *Smith Ref Point Properties*. The *Smith Reference Point Marker Properties for <marker_name>* form opens.



The *Smith Reference Point Marker Properties for <marker_name>* form includes the following fields:

- *Frequency*—Specify the frequency to move the marker by frequency. If the data point for the specified frequency exists on the graph, the marker will be moved to that data point.

  Note: See the Snapping Markers on Circular Graphs section for more information.

© 2023 Cadence Design Systems, Inc.

- *Real/Imag Position*—Displays the real and imaginary values of the data point on the trace where marker is currently attached. This is a read-only field.
- *Gamma*: Displays the magnitude and the angle of the data point on the trace where marker is currently attached. This is a read-only field.
- *Impedance (Z)*: Displays the impedance of the data point on the trace where marker is currently attached. This is a read-only field.
- *Admittance (Y)*: Displays the admittance of data point on the trace where marker is currently attached. This is a read-only field.
- *Numerical Format*—Select the numerical format as `Scientific`, `Engineering`, and `Suffix`.

Adding a Point Marker

To add a point marker, do one of the following:

- Select the point on the trace where you want to add the point marker and press `M`.
- Press the bindkey `M`.

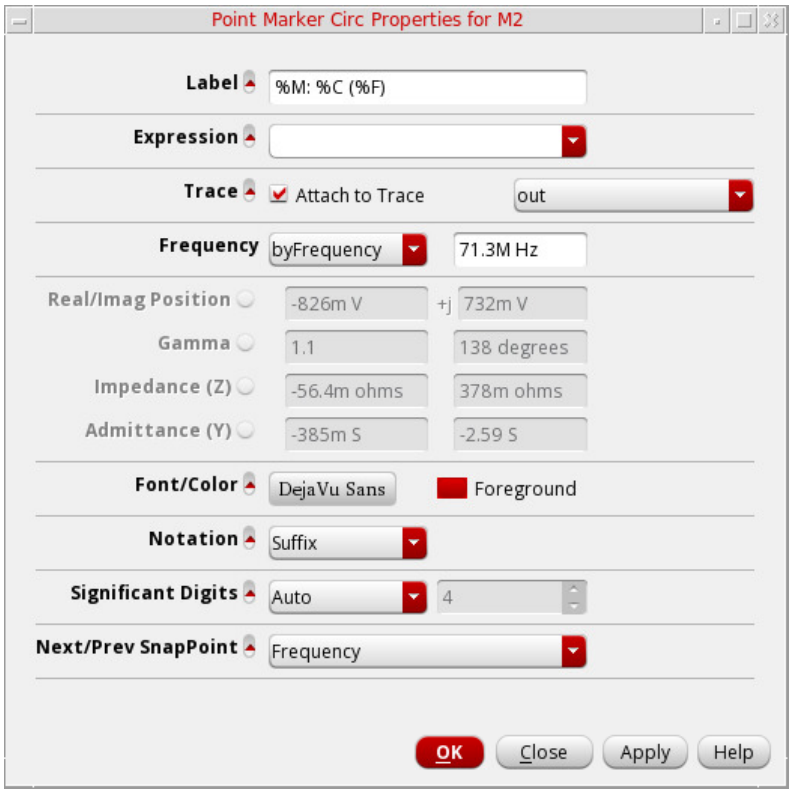
  The point marker is created on the point you click on the trace.

  To change the point marker properties, right-click the point marker and choose *Marker Properties*. The *Point Marker Properties for <marker-name>* form appears.

© 2023 Cadence Design Systems, Inc.

Setting Point Marker Properties for a Circular Graph

The following fields are displayed for setting the properties of a point marker for a circular graph:



- *Label*—Specify the label for the marker. See Marker Labels for more information.
- *Expression*—Specify the expression associated with the marker. For more information about expressions, see Adding Markers.
- *Trace*—Specify the name of the trace to which you want to attach the marker.
- *Frequency*—Specify the frequency to move the marker by frequency. If the data point for the specified frequency exists on the graph, the marker will be moved to that data point.

  Note: See the Snapping Markers on Circular Graphs section for more information.
- *Real/Imag Position*: Displays the real and imaginary values of the data point on the trace where marker is currently attached. This is a read-only field.
- *Gamma*: Displays the magnitude and the angle of the data point on the trace where marker is currently attached. This is a read-only field.

© 2023 Cadence Design Systems, Inc.

- *Impedance (Z)*: Displays the impedance of the data point on the trace where marker is currently attached. This is a read-only field.
- *Admittance (Y)*: Displays the admittance of data point on the trace where marker is currently attached. This is a read-only field.
- *Font/Color*—Specify the font name, style, and size for the label.
- *Notation*—Specify the notation to be displayed on labels. The available options are—`Suffix`, `Engineering`, and `Scientific`. Default value: `Suffix`.
- *Significant Digits*—Specify the number of significant digits if you select `manual` from the drop-down. The other option is `auto`.
- *Next/Prev SnapPoint*—Specify the criterion based on which the selected marker is to be snapped. See the Snapping Markers on Circular Graphs section for more information.

  The available options are—`Frequency` and `Specific Frequency`.

  Default value: `Frequency`.

Adding a Circular Marker

To add a circular marker, do the following:

➥ Choose *Marker – Create Marker*.

The *Create Graph Marker* form appears. On the *Circular* tab, in the *Position* field, specify the real and imaginary values for the circular marker. In the *Radius* field, specify the radius. For more information about the circular marker fields displayed in the *Create Graph Marker* form, see Adding Markers on page 495.

Deleting Markers

To delete a marker, see Deleting a Marker on page 519.

Snapping Markers on Circular Graphs

You can snap point markers and reference point markers on circular graphs based on the following point of interest (POI) criteria:

- *Frequency*: Snaps the marker on the trace to the data point where there is a change in frequency on the circular graph.

© 2023 Cadence Design Systems, Inc.

- *Specific Frequency*: Snaps the marker based on the specified frequency:
  - If the data point for the specified frequency exists on the graph, the marker will be moved to that data point.
  - If the data point for the specified frequency does not exist on the graph, and the frequency value is within the range of frequencies for the waveform, the marker will be moved to a point that is obtained using linear interpolation of the following values:
    - The greatest frequency value from the dataset which is lower than the specified frequency value
    - The smallest frequency value from the dataset which is greater than the specified frequency value
  - If the specified frequency is out of range of frequencies for the waveform, the marker will not be moved and an error message will be displayed on the CIW.

POI criteria defines the data point on the trace where the marker is moved when you snap it.

© 2023 Cadence Design Systems, Inc.

Follow these steps to snap a point marker or a reference point marker on a circular graph:

1. Select the marker and choose either *Frequency* or *Specific Frequency* as the POI criteria from the POI criteria drop-down list displayed on the snap toolbar, as shown in the following figure.

   Note: The text field displayed next to the POI criteria drop-down list appears dimmed, and becomes available only when you select *Specific Frequency* as the POI criteria.

2. Depending upon the POI criteria that you selected in step 1, do the following:

   - If you selected *Frequency*, click *Next POI* or *Previous POI* on the snap toolbar to move the marker to the next or previous data points, respectively.

     You can also use the `N` or `P` bindkeys to snap the marker to the next or previous data points, respectively.

     - Clicking *Next POI* or pressing the `N` bindkey snaps the marker on the trace to a data point where the next higher frequency is available.
     - Clicking *Previous POI* or pressing the `P` bindkey snaps the marker on the trace to a data point where the next lower frequency is available.

   Note: *Next POI* and *Previous POI* commands are disabled when you set the POI criteria to *Specific Frequency*.

   - If you select *Specific Frequency*, the text field next to the POI criteria drop-down list becomes available. Specify the frequency value in the field to snap the marker to that location.

You can also snap a marker by clicking the *Next POI* and *Previous POI* commands from the context menus of the *Point Marker Properties* and *Smith Reference Point Marker Properties* forms.

Alternatively, you can specify the frequency in the *Frequency* fields on these forms to snap the markers. See the Setting Point Marker Properties for a Circular Graph and Setting Smith Reference Point Marker Properties sections for more information.

© 2023 Cadence Design Systems, Inc.

Plotting Circular Graphs with Different Axis Units

By default, the circular graphs with the matching dependent axis units are plotted in the append mode in the same window.

To append data with different dependent axis units to an existing polar or smith graph,

➡ Right-click the graph and choose *Axis Allows Any Units*.

The circular graphs with any axis units can now be plotted in the same window.

Zooming Circular Graphs

You can zoom in or out the circular graph by doing the following:

- To fit the circular graph to the data values, choose *View – Fit* or right-click anywhere on the circular graph and choose *Fit.*
- To fit the trace in the circular graph to the window, do one of the following:
  - Right-click the trace in the circular graph and choose *Fit Trace*.
  - Select the trace in the circular graph and choose *View – Fit Trace*.
  - Select the trace in the circular graph and choose *Trace – Fit Trace*.
- To fit the circular graph to the Smith values, do one of the following:
  - Right-click anywhere on the circular graph and choose *Smith Fit*.
  - Choose *View – Fit Smith*.

Note: When you zoom in a graph, the labels for the zoomed-in area are displayed outside the standard unit circle.

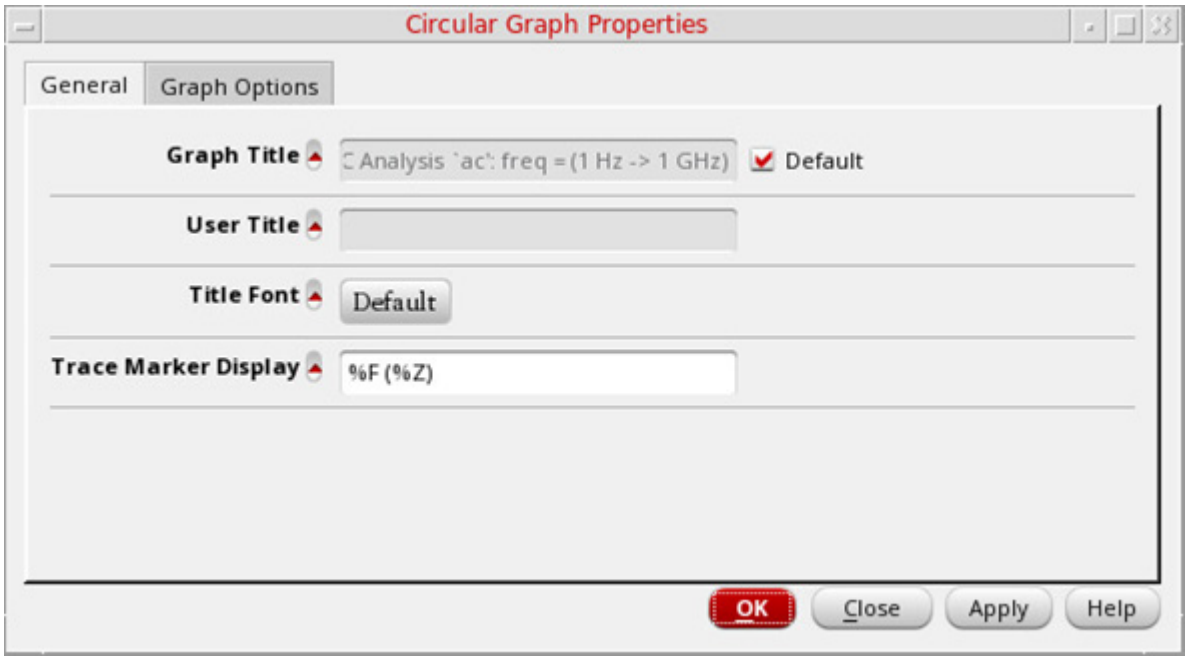
Editing Circular Graph Properties

To set the properties of the circular graph, do one of the following:

- Choose *Graph – Properties*.
- Right-click anywhere in the circular graph and choose *Smith Graph Properties* for a Smith chart and *Polar Graph Properties* for a polar graph.

© 2023 Cadence Design Systems, Inc.

The *Circular Graph Properties* form appears.



This form includes the following two tabs:

- *General*—For information about the fields displayed on the *General* tab, see Editing Graph Properties on page 196.
  - *Trace Marker Display*—Specify the following formats to display the trace marker values:
    - `%C`—Displays the real and imaginary Cartesian values
    - `%P`—Displays the polar values, such as mag and angle
    - `%Z`—Displays the impedance values, such as resistance and reactance
    - `%A`—Displays the admittance values, such as conductance and susceptance
    - `%R`—Displays the reflection coefficients, such as mag and angle
    - `%F`—Displays the frequency value, which includes the independent axis data
- *Graph Options*—Includes the following fields:
  - *Font*—Select the font properties of the Smith chart.
  - *Notation*—Select the graph notation as *Scientific*, *Engineering*, or *Suffix*. Default value: *Suffix*.

© 2023 Cadence Design Systems, Inc.

- *Reload using Current Context*—When this check box is selected, signals in the graph are reloaded according to the data from the current in-context results directory. If this check box is not selected, signals are reloaded based on their individual databases. This check box is selected by default.
- *Grid Type*—Select the grid type as *Polar*, *Impedance*, *Admittance*, or *Immittance*.
- *Legend Position*—Sets the trace legend position as *left*, *inside*, or *above*. Default value: *left*.
- *Show Frequency Filter*—When this check box is selected, a slider bar is displayed on the circular graph. You can use this slider bar as a frequency filter to selectively display data between the specified frequencies on the circular graph. See the Filtering Data Using the Frequency Slider Bar section for more information.
- *Frequency Filter*—Select the minimum and maximum frequencies between which you want to display data on a circular graph. You can also use the slider bar as a frequency filter.
- *Smith Data*—Specify the characteristic impedance Zo. Select the *Normalize Smith Value* check box if you want to normalize the Smith data values according to the impedance you specify.

■ Click *OK.*

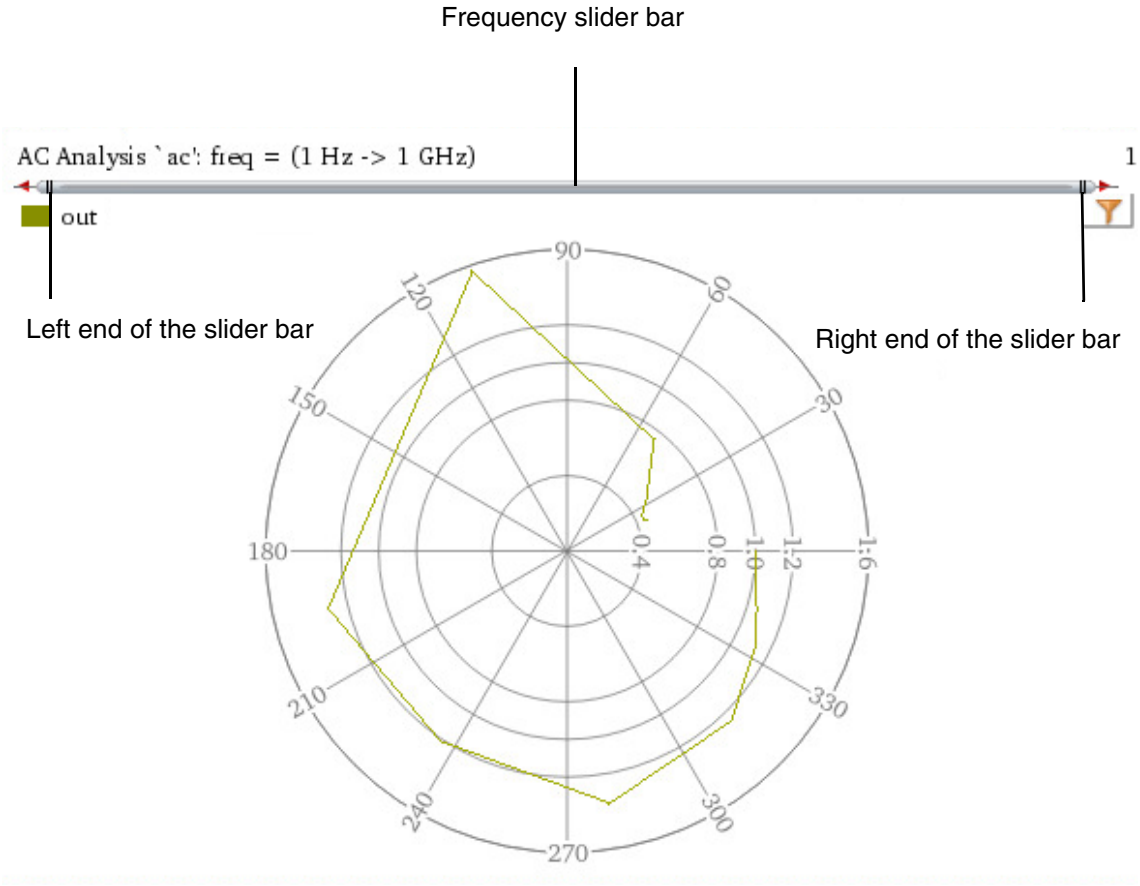
Filtering Data Using the Frequency Slider Bar

You can use the frequency slider bar as a frequency filter to set a range of frequencies. Data on the circular graph is filtered and selectively displayed based on the specified frequency range.

Note: The frequency slider bar is hidden by default. To view the frequency slider bar, set the environment variable `viva.circGraph showFreqFilterBar` to `true`.

© 2023 Cadence Design Systems, Inc.

The following figure shows the frequency slider bar that appears on top of a circular graph.

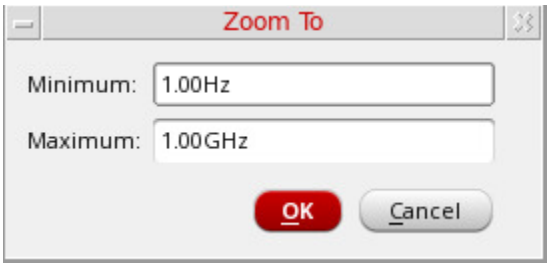


The default frequency range of the slider bar varies depending on the simulation data.

To set the minimum and the maximum frequencies, do the following:

1. Move the left end of the slider bar to set the minimum frequency.
2. Move the right end of the slider bar to set the maximum frequency.
3. Move the slider bar to the left or right of the graph to change both the minimum and the maximum frequencies simultaneously.
4. Double-click the slider bar to open the Zoom To form, where you can set the values of *Minimum* and *Maximum* frequencies.

© 2023 Cadence Design Systems, Inc.



You can also use suffix notation to specify the minimum and maximum frequency values. For example, `KHz`, `MHz`, and `GHz`.

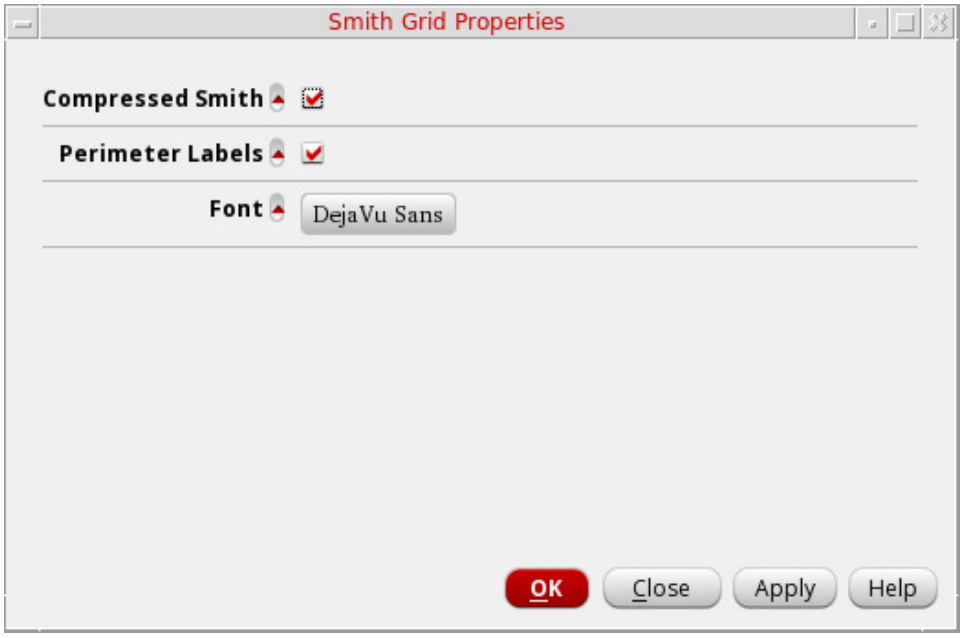
5. Right-click the slider bar and select one of the following options to reset the applied frequency filters:

   - *Reset Frequency Limits*: Resets the frequency filter to its default range.
   - *Reset Minimum Frequency*: Resets the minimum frequency limit to its default value.
   - *Reset Maximum Frequency*: Resets the maximum frequency limit to its default value.

© 2023 Cadence Design Systems, Inc.

Setting Smith Grid Properties

To set the Smith grid properties, right-click a Smith chart and choose *Smith Grid Properties*. The *Smith Grid Properties* form appears.



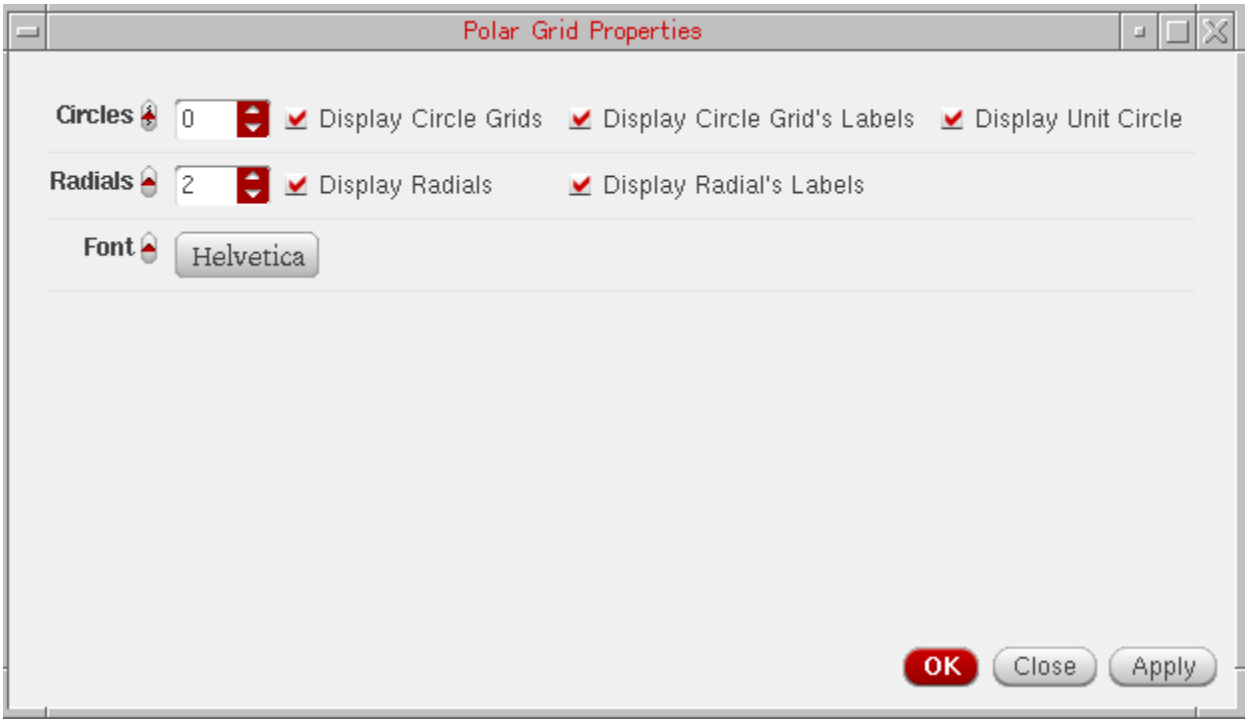
The form includes the following fields:

- *Compressed Smith*—Select this check box if you want to display the extra horizontal grid lines (arcs) outside the Smith chart boundary.
- *Perimeter Labels*—Select this check box if you want to display labels at arc intersections on the Smith unity circle.
- *Font*—Specify the font properties for the Smith chart labels.

© 2023 Cadence Design Systems, Inc.

Setting Polar Grid Properties

To set the polar grid properties, right-click a polar plot and choose *Polar Grid Properties*. The *Polar Grid Properties* form appears.



This form includes the following fields:

- *Circles*—Specify the number of circles you want to draw on the polar grid. Default value is `0`.
  - Select the *Display Circular Grid's Labels* check box if you want to display the labels for circular grids.
  - Select the *Display Unit Circle* check box if you want to display a unit circle, which means circle with R=1.
- *Radials*—Specify the number of radials you want to draw in each quarter of the polar grid. Default value is `2`.
  - Select the *Display Radials* check box if you want to display the radials in the polar graph.
  - Select the *Display Radial's Labels* check box if you want to display labels for radials in the polar graph
- *Font*—Select the font properties for the polar grid labels.

© 2023 Cadence Design Systems, Inc.

Setting Dependent Modifiers for a Complex Trace

You can set the dependent modifiers for an AC or a complex dataset, such as Mag, dB10, dBm, and dB20. You can also calculate these modifiers based on the resulting eye diagram and the spectrum waveform.

To change the modifier for an AC or a complex dataset, do one of the following:

- Right-click the AC waveform and choose *Dependent Modifier – Magnitude/Phase/dB10/dB20/dBm/WrapPhase/Real/Imag*.

  The selected modifier is plotted in the window in which the graph was plotted.

- Right-click the AC waveform and choose *Trace Properties*.

  The *Complex Trace Properties* form appears. In this form, from the *Dependent Modifier* drop-down list, select one of the following modifiers that you want to apply to the trace:

  - *Magnitude*—Plots magnitude versus frequency.
  - *dB10*—Plots *dB10* value of the dependent data versus frequency.
  - *dB20*—Plots *dB20* value of the dependent data versus frequency.
  - *dBm*—Plots mili dB value of the dependent data versus frequency.
  - *Phase*—Plots phase versus frequency.
  - *WPhase*—Plots wrapped phase versus frequency.
  - *Real*—Plots real value of the dependent data versus frequency.
  - *Imaginary*—Plots imaginary value of the dependent data versus frequency.

  The selected modifier is plotted in the window in which the circular graph was plotted.

© 2023 Cadence Design Systems, Inc.

Setting Bindkeys

A bindkey is a key or a sequence of key press events linked (bound) to a task. When you press the key or the sequence of keys, the associated task is performed. The Virtuoso Visualization and Analysis XL too provides a set of default bindkeys, which are displayed next to the relevant commands on the menus. These bindkeys can be overwritten or modified by a customized bindkey file.

The sample bindkey files are found at the following locations:

```
$CDSHOME/tools/dfII/samples/local/vivaBindKeys.il
```

```
$CDSHOME/tools/dfII/samples/local/vivaJavaBindKeys.il
```

To view all the bindkeys for the Virtuoso Visualization and Analysis XL tool, in CIW, choose *Options Bindkey*. The *Bindkey Editor* appears that includes all the bindkeys for all applications, such as *vivaBrowser*, *vivaCalculator*, and *vivaGraph*.

You can map keystrokes to the tasks that you perform in the window. However, you cannot bind mouse actions to tasks. A task is defined as follows:

```
Task = { graph_task | menu_item_task | skill_function }
```

`graph_task`	Specifies the task that the bindkey is linked to. You can link a bindkey to any of the following tasks: `marker, vertmarker, horizmarker, deltamarker, vertcursor, horizcursor, deltacursor, tracecopywin, tracemovewin, tracecopysubwin, tracemovesubwin, logscale, cut, copy, paste, delete, cancel, undo, cancel, traceinfo, pandown, panup, panleft, panright, zoom, zoomx, zoomy, zoomin, and zoomfit`
`menu_item_task`	Specifies the item name in the following format `graph.`*`menu.submenu1.submenu2`* For example, panning to the right is defined as follows: `graph.zoom.pan.panright.`
`skill_function`	Specifies the SKILL function call, such as `awvFitMenuCB`(). For information about SKILL functions, see the *SKILL Language Reference*.

A keystroke is defined as follows:

© 2023 Cadence Design Systems, Inc.

```
keystroke = simple_keystroke | composite_keystroke
```

`simple_keystroke`	A single letter, number, symbol, or key name. Examples: `a`, `2`, `@`, `Up`, `F1`
`composite_keystroke`	The format is *`modifiers`*`<key>`*`simple_keystroke`* where *`modifiers`* = `alt`, `ctrl`, `meta`, `shift`, `control`, `super`, `hyper`, `mod1`, `mod2`, `mod3`, `mod4`, `mod5` Examples: `meta<key>Right`, `AltShift<key>F2`

The below table lists the bindkeys available in Virtuoso Visualization and Analysis XL:

Command	**Bindkey**
Results Browser	
Open Results Browser using the Select Waveform form	`Alt+F+O`
Plot the selected signal(s) in the Results Browser	`Ctrl+P`
Reload the results in Browser	`Ctrl+R`
Create a table view of selected signals	`Ctrl+T`
Remove traces from the graph window	`Ctrl+X`
Calculator	
Close Calculator	`Ctrl+Q`
Graph	
Add an A type marker	`A`
Add a B type marker	`B`
Toggle the tracking cursor	`C`
Add a delta marker between two markers	Use `Shift+D` when two or more markers selected
Pan down	`Down`
Delete selected object	`Delete`
Delete All	`E`
Stop listening for signal selection in the Function Assistant	`Escape`
Zoom Fit	`F`

© 2023 Cadence Design Systems, Inc.

Command	Bindkey
Toggle Assistants	F11
Toggle Toolbars	Ctrl+F11
Toggle Toolbars and Assistants	Shift+F11
Toggle Minor Grids	G
Toggle Major Grids	Shift+G
Toggle Minor and Major Grids	
Add a horizontal marker	H
Pan left	Left arrow
Add a marker	M
Jump to next edge	N
Jump to previous edge	P
Edit attributes of the selected object	Q
Add a reference point marker	R
Pan right	Right arrow
Add a transient edge marker	T
Undo	U
Add a vertical marker	V
Zoom in X	X (RMB+drag), Shift+RMB+drag or Shift+ mouse scroll
Zoom in Y	Y (RMB+drag), Ctrl+RMB+drag, or Ctrl+ mouse scroll
Zoom in XY	Z (RMB+drag box) or RMB+drag box
Zoom Out	[
Zoom in	]
Select all traces in a strip	Ctrl+A
Select all traces	Ctrl+Shift+A

© 2023 Cadence Design Systems, Inc.

Command	Bindkey
Redo	`Shift+U`
Paste	`Ctrl+V`
Save graph	`Ctrl+S`
Load graph	`Ctrl+V`
Fit Trace	`Ctrl+F`
Fir Smith	`Shift+F`
Edit graph properties	`Shift+Q`
Cut	`Ctrl+X`
Copy	`Shift+C`
Create new plot window	`Ctrl+N`
Close all windows	`Ctrl+Q`
Delete all traces	`Shift+E`
Delete all markers	`Ctrl+E`
Update/Reload Graphs	`Ctrl+R`

© 2023 Cadence Design Systems, Inc.

Working with Tabular Graph

You can use tabular graph to display waveform data in a table. Tabular graphs offer a convenient way to analyze waveform data, especially for S-Parameter data.

This section included the following topics:

- Creating a Tabular Graph
- Editing Tabular Graph Properties
- Using Tabular Trace Properties Commands for a Column
- Using Tabular Trace Properties Commands for a Row
- Customizing Display of S-Parameter Data

Video

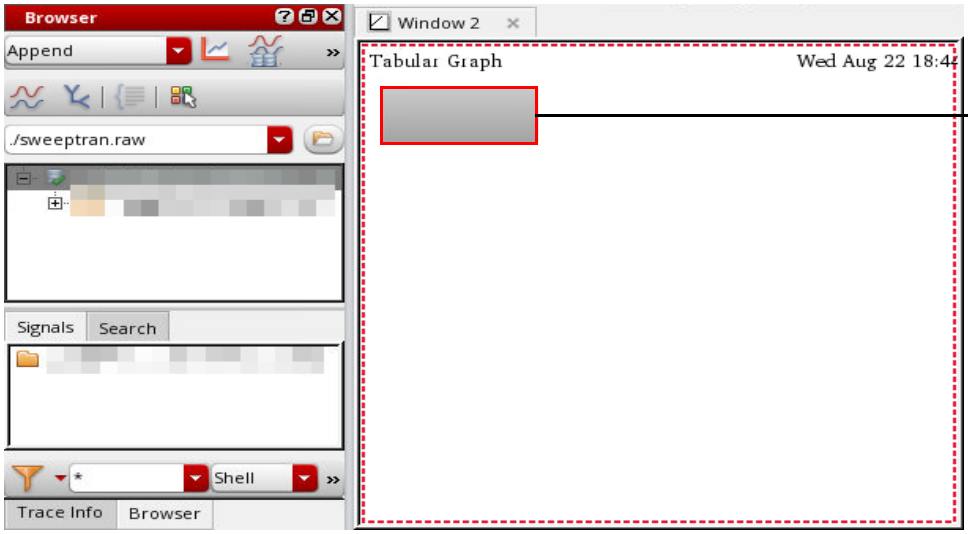
You can watch a video demonstration for this feature. Also, read the related blog.

Creating a Tabular Graph

To create a tabular graph:

1. From the File tool bar of Virtuoso Visualization and Analysis XL, choose *Create New Window – Tabular*.

   An empty tabular graph window opens.

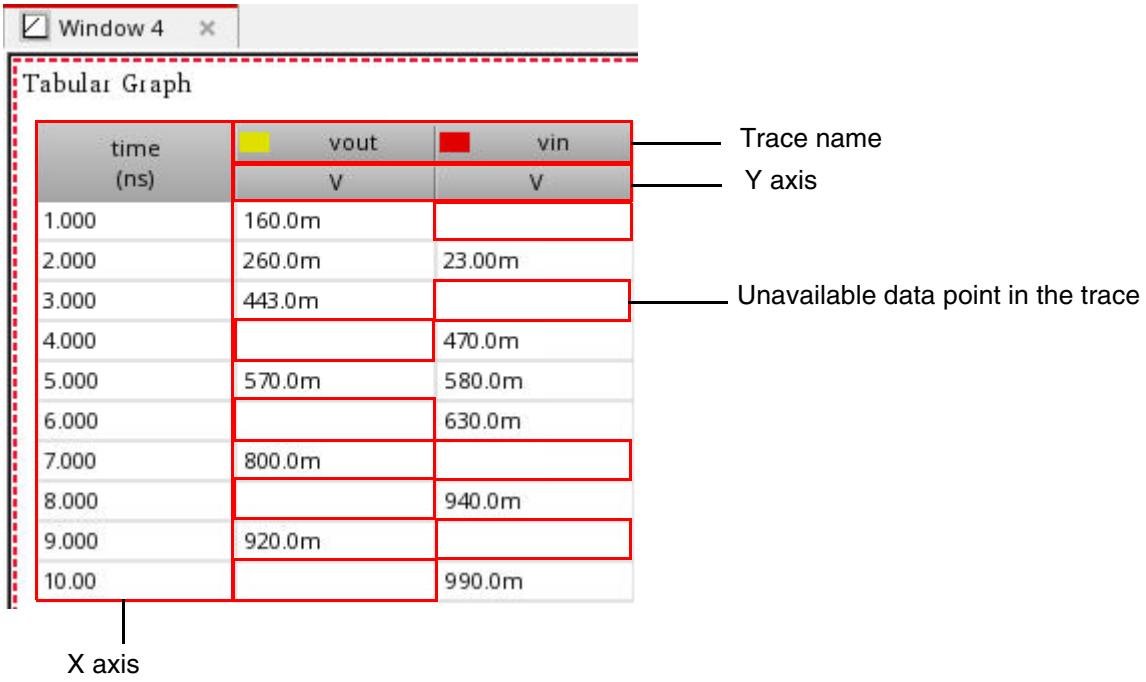


© 2023 Cadence Design Systems, Inc.

2. From the Results Browser, right-click the signals, and then select *Plot Signal*.

   Alternatively, you can drag the signals onto the tabular graph window.

   The waveform data of the selected signals is displayed in a table in the tabular graph window.

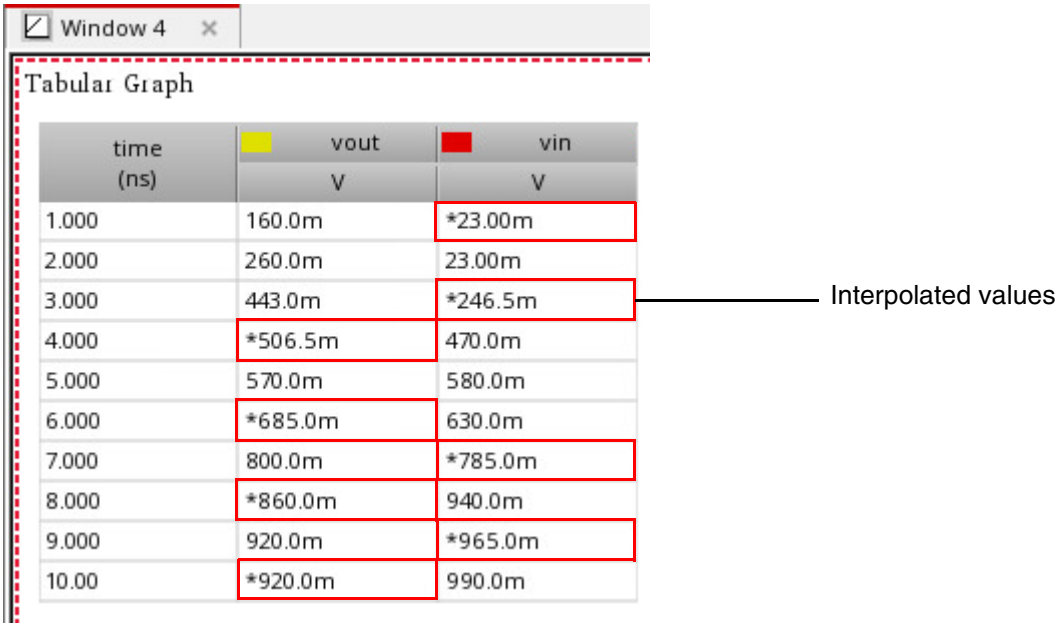


   The left-most column in the table represents the X-axis values. The multi-level headers represent the trace names and their Y-axis values for the corresponding X-axis values.

   The empty cells in the table indicate that the corresponding data points are not available in the trace. However, you can obtain these values using interpolation method by right-clicking any cell and then choosing *Tabular Trace Properties – Interpolate*. The

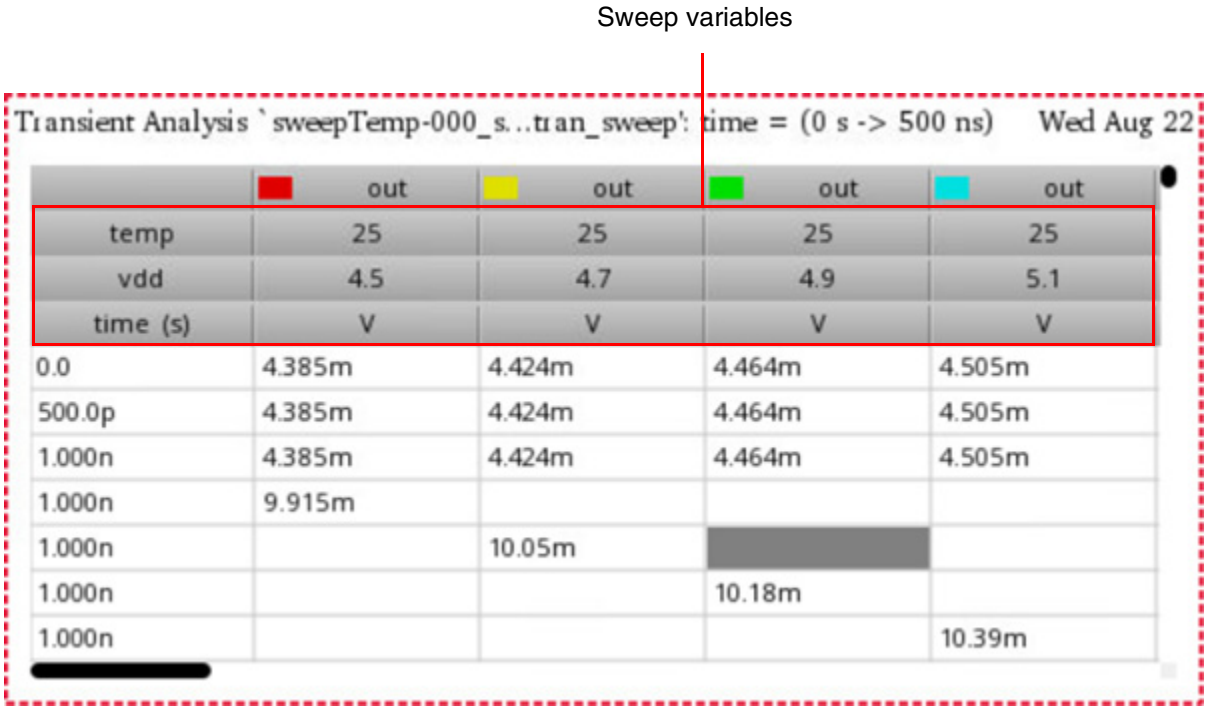
© 2023 Cadence Design Systems, Inc.

interpolated values are updated in the tabular graphs. These values are marked with an asterisk (*).



time (ns)	vout V	vin V
1.000	160.0m	*23.00m
2.000	260.0m	23.00m
3.000	443.0m	*246.5m
4.000	*506.5m	470.0m
5.000	570.0m	580.0m
6.000	*685.0m	630.0m
7.000	800.0m	*785.0m
8.000	*860.0m	940.0m
9.000	920.0m	*965.0m
10.00	*920.0m	990.0m

The corner names and sweep parameters are displayed as headers in the tabular graph, as shown in the following figure.



	out	out	out	out
temp	25	25	25	25
vdd	4.5	4.7	4.9	5.1
time (s)	V	V	V	V
0.0	4.385m	4.424m	4.464m	4.505m
500.0p	4.385m	4.424m	4.464m	4.505m
1.000n	4.385m	4.424m	4.464m	4.505m
1.000n	9.915m			
1.000n		10.05m		
1.000n			10.18m	
1.000n				10.39m

© 2023 Cadence Design Systems, Inc.

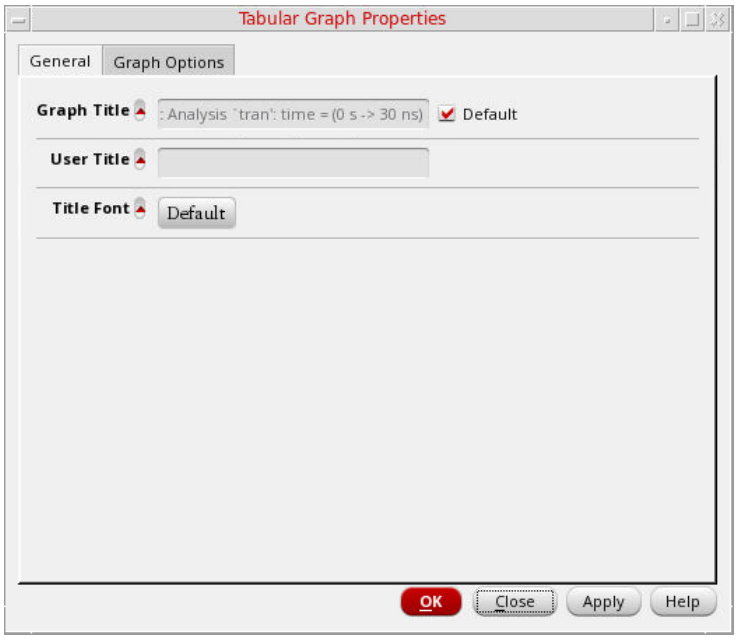
Editing Tabular Graph Properties

To edit the properties of a tabular graph:

1. From the menu bar of Virtuoso Visualization and Analysis XL, choose *Graph – Properties*.

   Alternatively, right-click on the tabular graph window, and then select *Tabular Graph Properties*.

   The *Tabular Graph Properties* form is displayed.



2. On the *General* tab, specify the following properties:

   - *Graph Title*— Title of the graph that is displayed at the top of the graph. When you select the *Default* check box next to this field, you cannot edit the graph title or provide a new graph title. The default graph name includes the name of the analysis and the Y-axis name.
   - *User Title*— Title of the graph window that you want to set. You must clear the *Default* check box to edit this field.
   - *Title Font*— Font properties of the graph title.

3. On the *Graph Options* tab, specify the following fields:

© 2023 Cadence Design Systems, Inc.

- *Interpolation*— Interpolated values of the signal for data points that are not available in the trace are shown in the tabular graph. These values are marked with an asterisk (*).
- *Font*— The font properties for the graph and its components, such as labels and axes.
- *Use Common Format*— Common format of notation is used for all values in the tabular graph.
- *Notation*—Specifies the notation type to be used for displaying data. You can specify one of the following three notation types:
  - *Engineering*— Displays data by using the engineering notation.
  - *Suffix*— Displays data by using the suffix notation.
  - *Scientific*— Displays data by using the scientific notation.
- *Significant Digits*— Number of significant digits to be displayed in the calculated values. The default value is 4. You must select *Manual* to edit this value.
- *Reload using Current Context*— When this check box is selected, signals in the graph are reloaded according to the data from the current in-context results directory. If this check box is not selected, signals are reloaded based on their individual databases. This check box is selected by default.

4. Click *Apply* and then click *OK* to save the changes.

Using Tabular Trace Properties Commands for a Column

Right-click a column header in the tabular graph, and then select one of the following *Tabular Trace Properties* commands:

- *Interpolate*— Displays the interpolated values of the signal for data points that are not available in the trace. These values are marked with an asterisk (*).
- *Hide*— Hides the selected column.
- *Reveal Columns*— Reveals all hidden columns.
- *Format Column*— Formats the values displayed in the selected columns. In the *Format Attributes* form, select a notation from the Scale Format drop-down list. Specify the number of significant digits.
- *Reveal All*— Reveals all hidden columns and rows.
- *Sort*— Sorts the values in ascending or descending order.

© 2023 Cadence Design Systems, Inc.

- *Resize to Contents*— Resizes the width of columns to ensure that the contents displayed in them are fully visible.
- *Resize to Defaults*— Restores the default width of columns.

In addition to these commands, you can also use the following context-menu commands upon right-clicking a column header of a tabular graph:

- *Trace Properties*— Opens the Trace Properties form for the selected trace.
- *Cut*— Moves the selected trace (column) to the clipboard. You can then paste the trace into a new graph window or subwindow.
- *Copy*— Copies the selected trace (column) to the clipboard. You can then paste the trace into a new graph window or subwindow.
- *Paste*— Pastes the contents of the clipboard to the selected graph window or subwindow.
- *Delete*—Deletes the selected trace (column) from the tabular graph.
- Send to— Sends the selected data to Virtuoso Visualization and Analysis XL table, Calculator, or ADE.
- *Move to*— Moves the selected column to a new window or subwindow.
- *Copy to*— Copies the selected column to a new window or a subwindow.
- *Color*— Changes the color of the selected trace.
- *Customize Display*— Customizes the display of S-Parameter data. See the Customizing Display of S-Parameter Data section for more information.

  Note: This option is available only if you have plotted S-Parameter data in the tabular graph.

Using Tabular Trace Properties Commands for a Row

Right-click a row in the tabular graph, and then select one of the following *Tabular Trace Properties* commands:

- *Interpolate*— Displays the interpolated values of the signal for data points that are not available in the trace. These values are marked with an asterisk (*).
- *Hide*— Hides the selected row.
- *Reveal Rows*— Reveals all hidden rows.

© 2023 Cadence Design Systems, Inc.

- *Format Table*— Formats the values displayed in the selected rows. In the *Format Attributes* form, select a notation from the Scale Format drop-down list. Specify the number of significant digits.
- *Reveal All*— Reveals all hidden rows and columns.

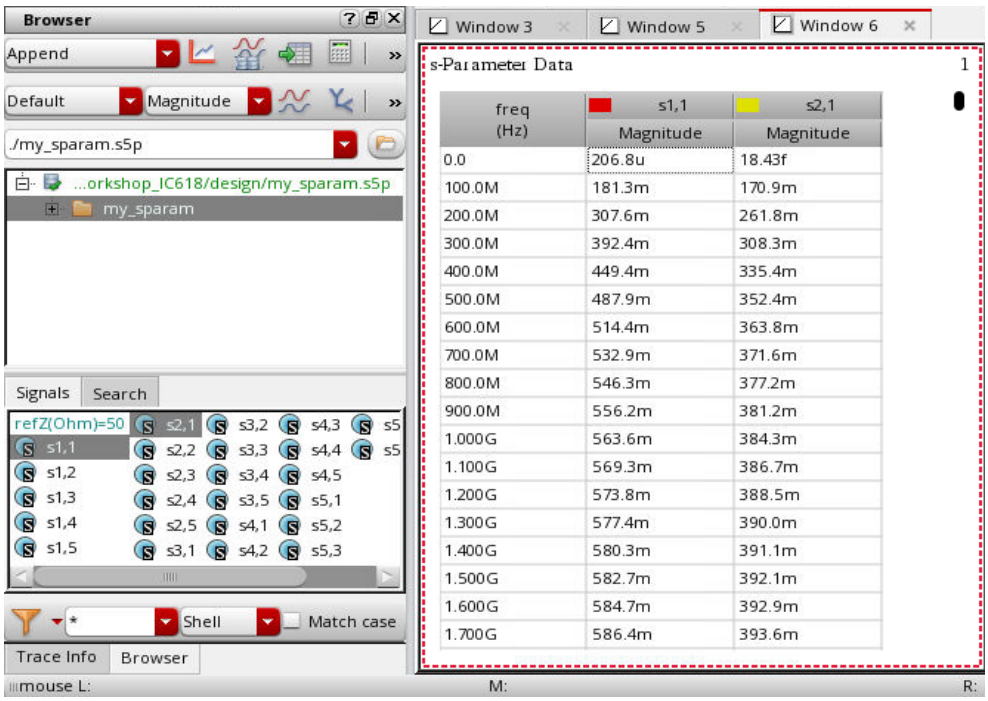
Customizing Display of S-Parameter Data

Tabular graphs offer a convenient way of analyzing S-Parameter data for different modifiers.

To customize display of S-Parameter data for different modifiers:

1. Plot S-Parameter data in a tabular graph window. See the Creating a Tabular Graph section for more information.

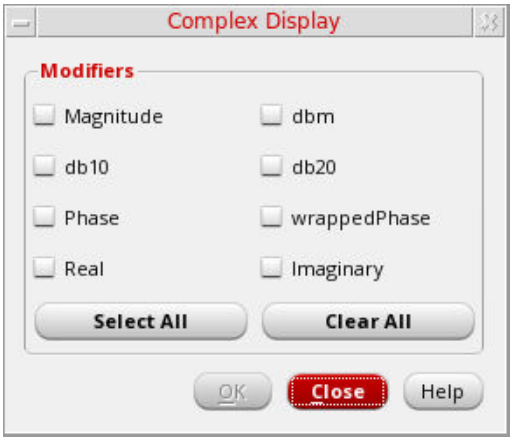
   The following figure shows an example of S-Parameter data plotted as a tabular graph.



2. Select a column header for which you want to customize the display.
3. Right-click the selected column and then select *Customize Display*.

© 2023 Cadence Design Systems, Inc.

The *Complex Display* form opens.



4. From the Modifiers list, select the modifiers for which you want to display the values of S-Parameters. For example, *Real* and *Imaginary*, and then click *OK*.

   The S-Parameter values for the selected modifiers are displayed in the tabular graph.

s-Parameter Data 1

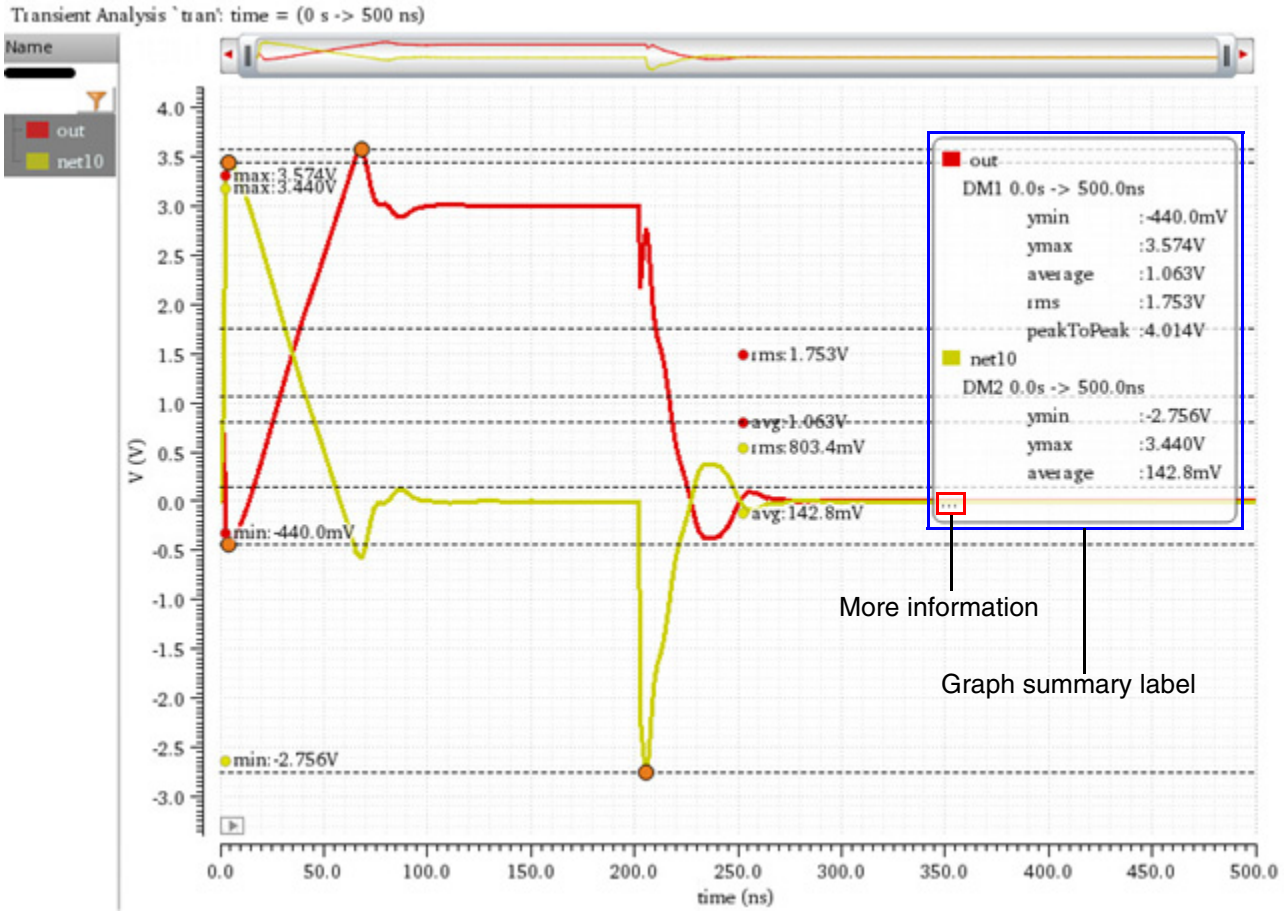
freq (Hz)	s1,1		s2,1
	Real	Imaginary	Magnitude
0.0	206.8u	146.9a	18.43f
100.0M	-63.20m	-169.9m	170.9m
200.0M	-173.2m	-254.2m	261.8m
300.0M	-272.1m	-282.7m	308.3m
400.0M	-349.7m	-282.2m	335.4m
500.0M	-407.3m	-268.6m	352.4m
600.0M	-449.3m	-250.5m	363.8m
700.0M	-479.9m	-231.7m	371.6m
800.0M	-502.6m	-214.1m	377.2m
900.0M	-519.7m	-198.1m	381.2m
1.000G	-532.7m	-184.0m	384.3m
1.100G	-542.9m	-171.4m	386.7m
1.200G	-551.0m	-160.4m	388.5m
1.300G	-557.4m	-150.7m	390.0m
1.400G	-562.7m	-142.1m	391.1m
1.500G	-567.0m	-134.5m	392.1m
1.600G	-570.6m	-127.7m	392.9m
1.700G	-573.6m	-121.6m	393.6m

© 2023 Cadence Design Systems, Inc.

Graph Summary Label

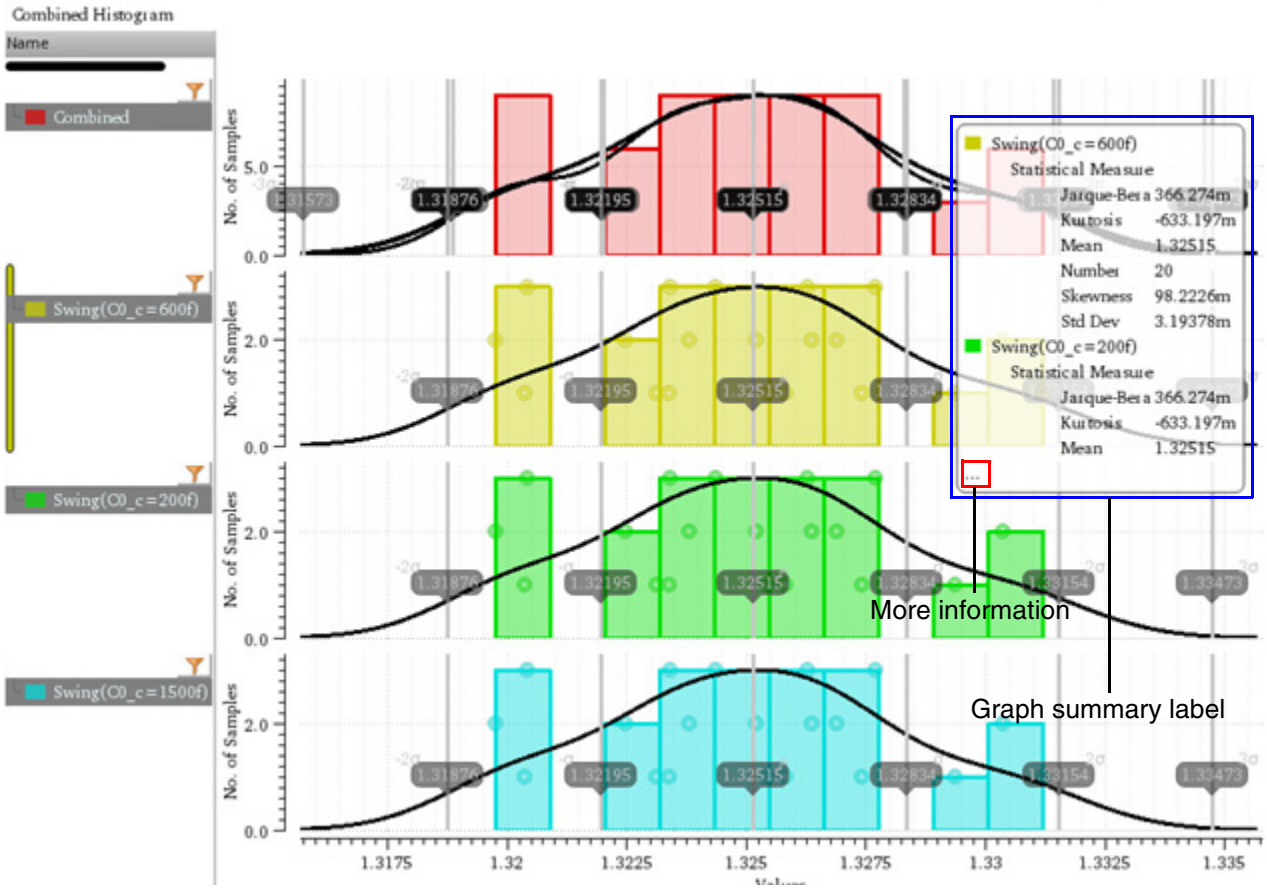
A graph summary label provides summarized information of graph measurements for histograms, direct measurements, and normal quantile plot (QQ plot) in an easy-to-view layout.

The following figure shows how various direct measurements for multiple signals are shown in a graph summary label.



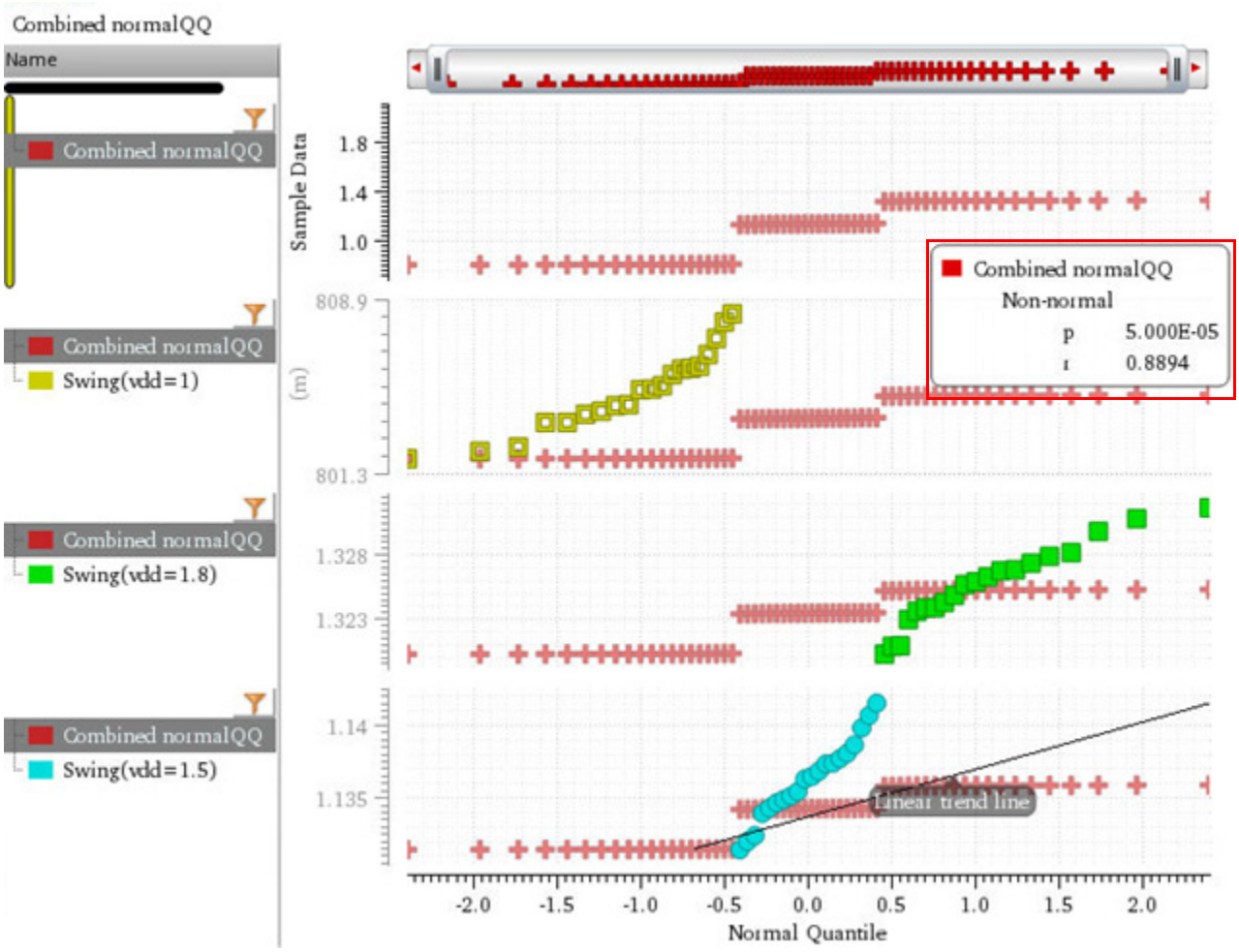
© 2023 Cadence Design Systems, Inc.

The following figure shows how various graph measurements for multiple histograms are shown in a graph summary label.



© 2023 Cadence Design Systems, Inc.

The following figure shows how graph measurements p-value (*p*) and correlation coefficient (*r*) for normal quantile plot (QQ plot) are shown in a graph summary label.



The ellipsis (…) displayed at the bottom-left corner of a graph summary label indicates that there are more graph measurements to view. You can choose to view the summary of these measurements out of the graph window or subwindow in a separate Summary Information form.

Video

You can watch a video demonstration for this feature at Graph Summary Label. Also, read the related blog at Virtuoso Video Diary: Usability of the Graph Summary Label in Virtuoso Visualization and Analysis XL Levels Up.

© 2023 Cadence Design Systems, Inc.

Related Topics

Hiding a Graph Summary Label

Viewing Graph Summary Information

Editing the Properties of a Graph Summary Label

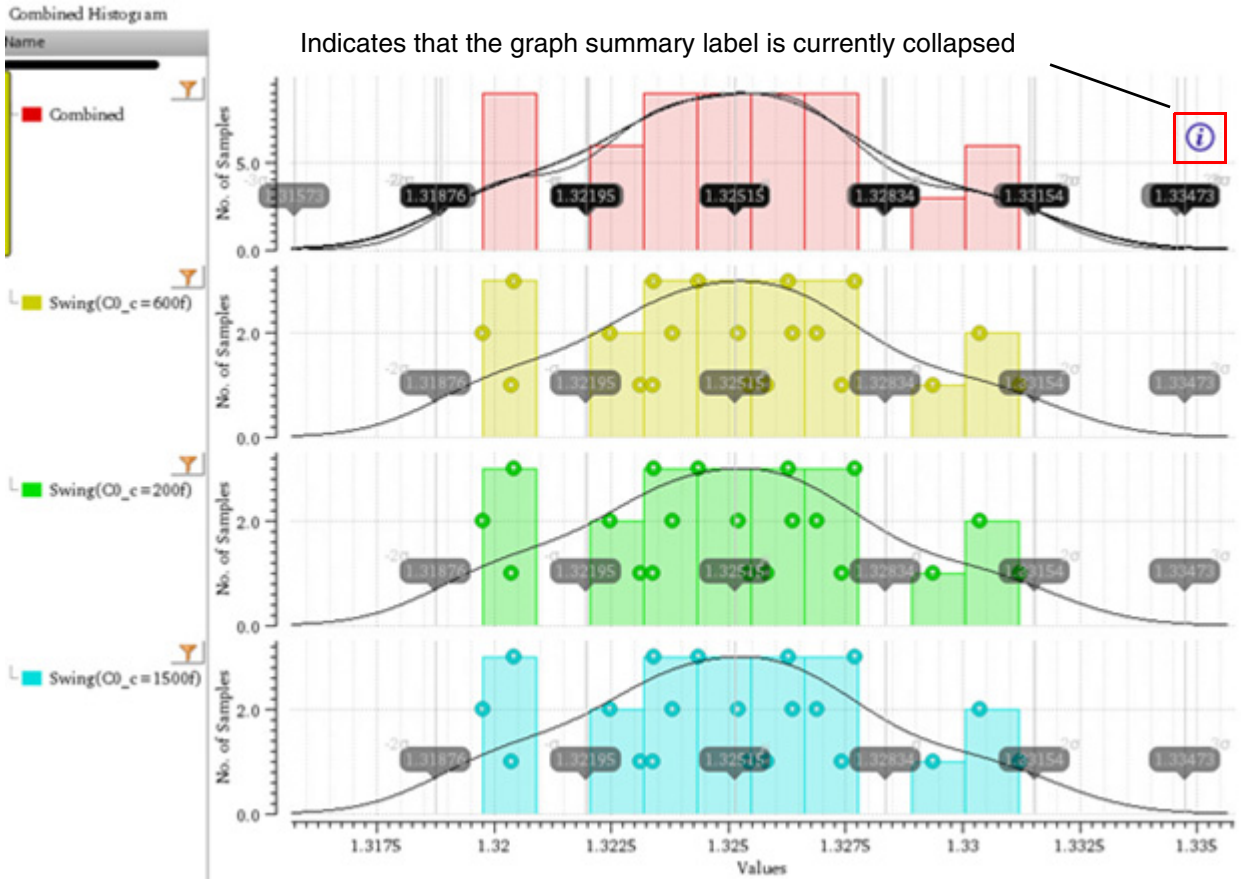
© 2023 Cadence Design Systems, Inc.

Hiding a Graph Summary Label

You can hide the graph summary label if you want to focus only on the graph. To do this:

1. Right-click anywhere on the graph summary label and choose *Expand/Collapse*.

   The graph summary label is collapsed and an icon is displayed.



2. To bring the graph summary label back, right-click the '*i*' icon and choose *Expand/Collapse*.

Related Topics

Graph Summary Label

Viewing Graph Summary Information

Editing the Properties of a Graph Summary Label

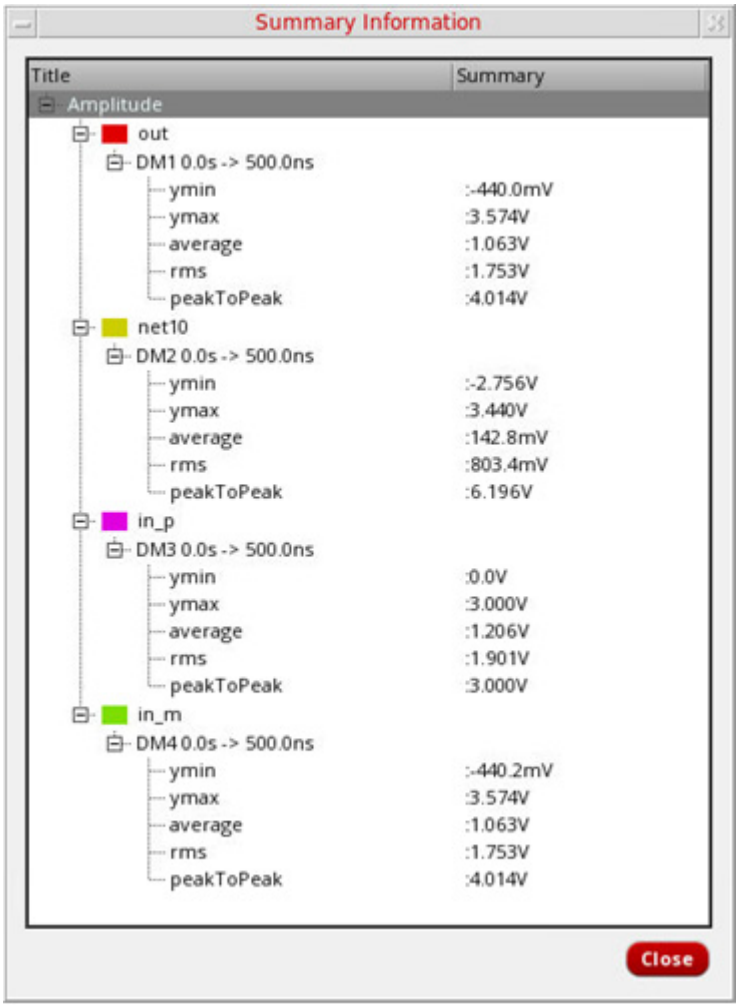
© 2023 Cadence Design Systems, Inc.

Viewing Graph Summary Information

To view the graph summary information:

1. Right-click anywhere on a graph label and choose *Summary Information*.

   The Summary Information form opens, displaying the graph measurements for all the signals of the graph window or subwindow.



2. Click *Close* to close the Summary Information form.

You can also copy the graph summary information to the clipboard. To do so, right-click the graph summary label and select *Copy*.

© 2023 Cadence Design Systems, Inc.

Related Topics

Graph Summary Label

Hiding a Graph Summary Label

Editing the Properties of a Graph Summary Label

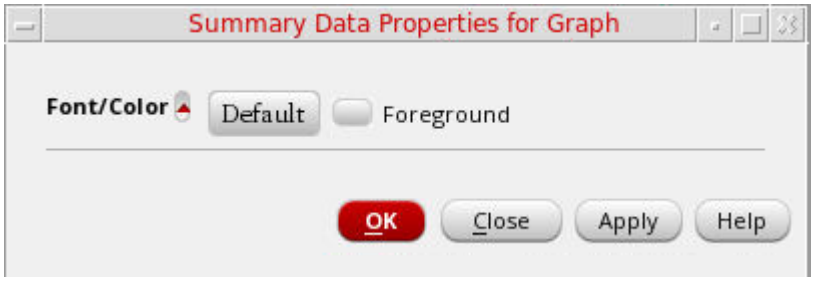
© 2023 Cadence Design Systems, Inc.

Editing the Properties of a Graph Summary Label

To edit the properties of a graph summary label:

1. Right-click anywhere on a graph label or on the icon and choose *Summary Properties*.

   The Summary Data Properties for Graph form opens.



2. In the *Font/Color* field, click *Default* to open the Define Font form where you can select the font, font style, and font size. You can also choose to apply *Strikeout* or *Underline* effects on the selected font.
3. Click *OK* to apply the changes and return to the Summary Data Properties for Graph form.
4. Click *Foreground* to choose the font color for the graph summary label.
5. Click *OK* to apply the settings.

Related Topics

Graph Summary Label

Hiding a Graph Summary Label

Viewing Graph Summary Information

© 2023 Cadence Design Systems, Inc.

Annotating Scalar Outputs for Single-Point Simulation

You can annotate scalar outputs in the graph window from the results of a single-point simulation. To annotate scalar outputs:

1. Depending on whether you are using ADE Explorer or ADE Assembler, do one of the following:

   - From the menu bar of ADE Assembler, choose *Options – Plotting/Printing*.

     The ADE Assembler Plotting/Printing Options form opens.

   - From the menu bar of ADE Explorer, choose *Results – Plotting/Printing Options*.

     The ADE Explorer Plotting/Printing Options form opens.

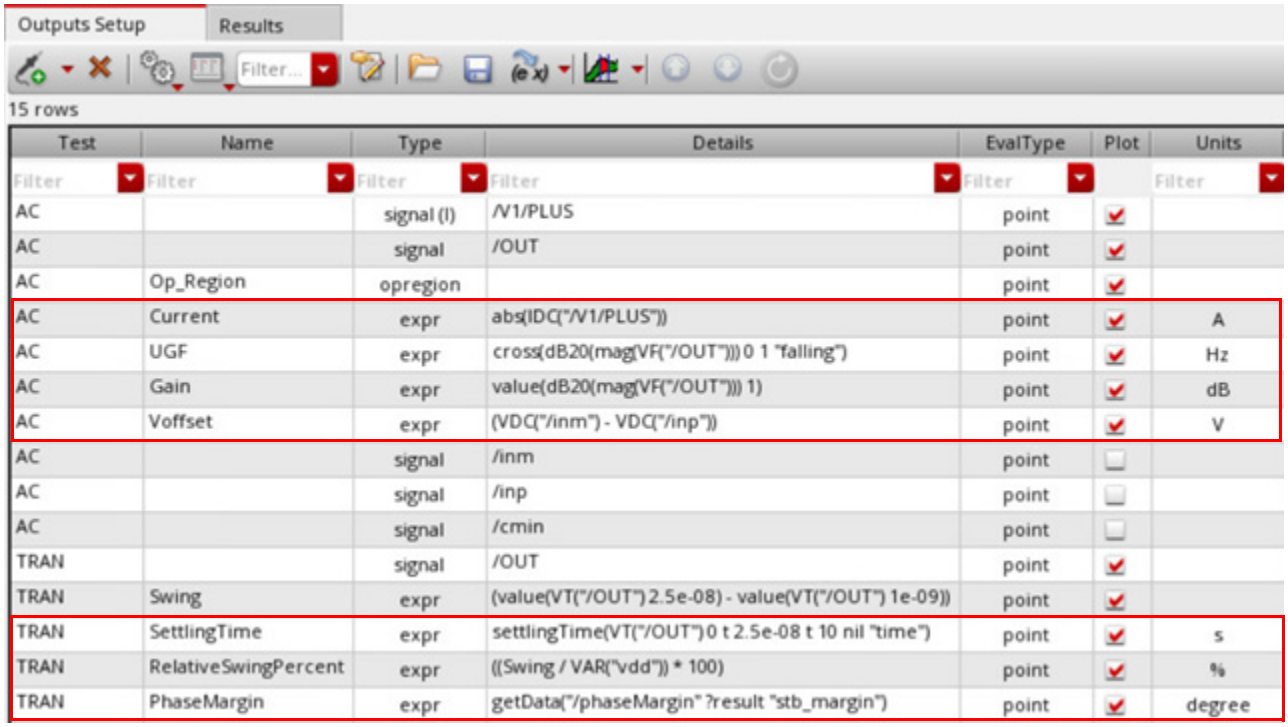
2. In the *Graph Annotations* group of the *Plot* section, select the *Scalar Outputs for Single-Point Simulation* check box.

3. Click *Run Simulation*.

   Simulation results are displayed in the *Results* tab. When you plot simulation results in Virtuoso Visualization and Analysis XL, the outputs that returned scalar values are annotated in the graph window.

© 2023 Cadence Design Systems, Inc.

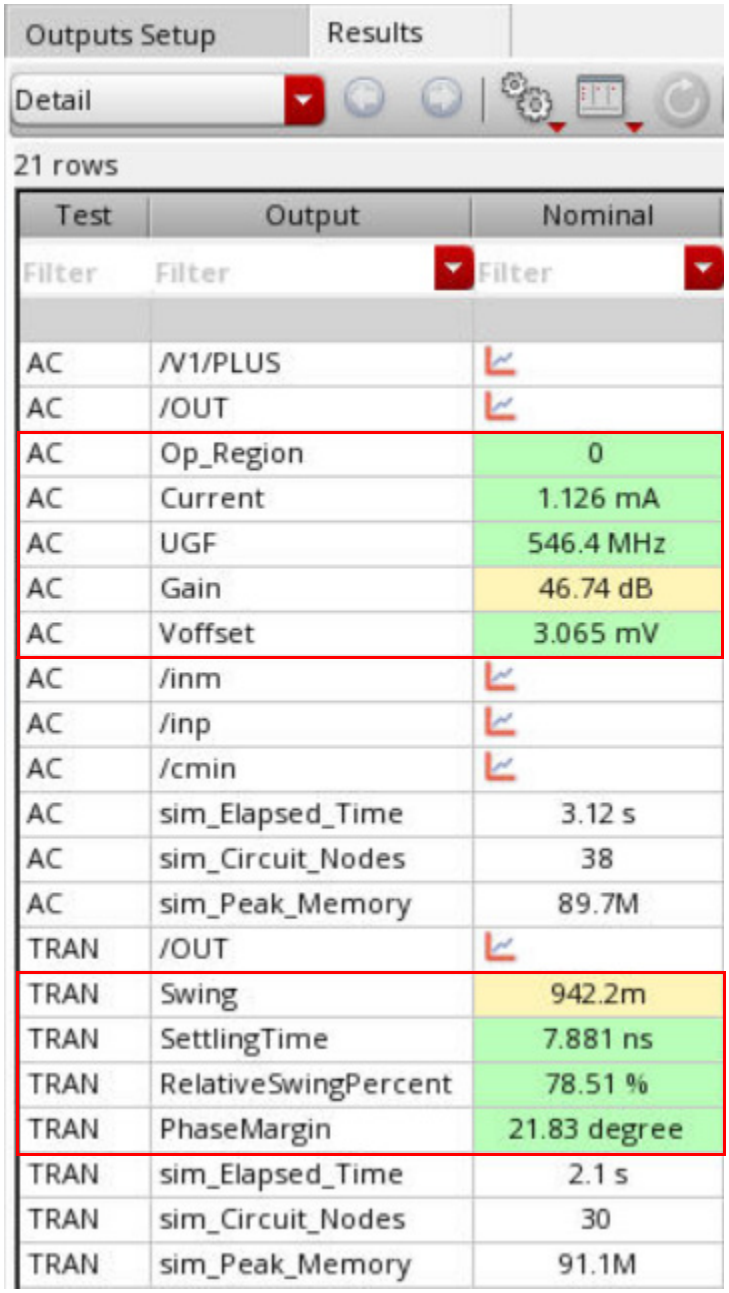
Consider the following test setup in ADE Assembler. The setup has two tests, `AC` and `TRAN`. Both the tests contain scalar outputs for which units are defined in the *Units* column.



Test	Name	Type	Details	EvalType	Plot	Units
Filter	Filter	Filter	Filter	Filter		Filter
AC		signal (I)	/V1/PLUS	point	☑	
AC		signal	/OUT	point	☑	
AC	Op_Region	opregion		point	☑	
AC	Current	expr	abs(IDC("/V1/PLUS"))	point	☑	A
AC	UGF	expr	cross(dB20(mag(VF("/OUT"))) 0 1 "falling")	point	☑	Hz
AC	Gain	expr	value(dB20(mag(VF("/OUT"))) 1)	point	☑	dB
AC	Voffset	expr	(VDC("/inm") - VDC("/inp"))	point	☑	V
AC		signal	/inm	point	☐	
AC		signal	/inp	point	☐	
AC		signal	/cmin	point	☐	
TRAN		signal	/OUT	point	☑	
TRAN	Swing	expr	(value(VT("/OUT") 2.5e-08) - value(VT("/OUT") 1e-09))	point	☑	
TRAN	SettlingTime	expr	settlingTime(VT("/OUT") 0 t 2.5e-08 t 10 nil "time")	point	☑	s
TRAN	RelativeSwingPercent	expr	((Swing / VAR("vdd")) * 100)	point	☑	%
TRAN	PhaseMargin	expr	getData("/phaseMargin" ?result "stb_margin")	point	☑	degree

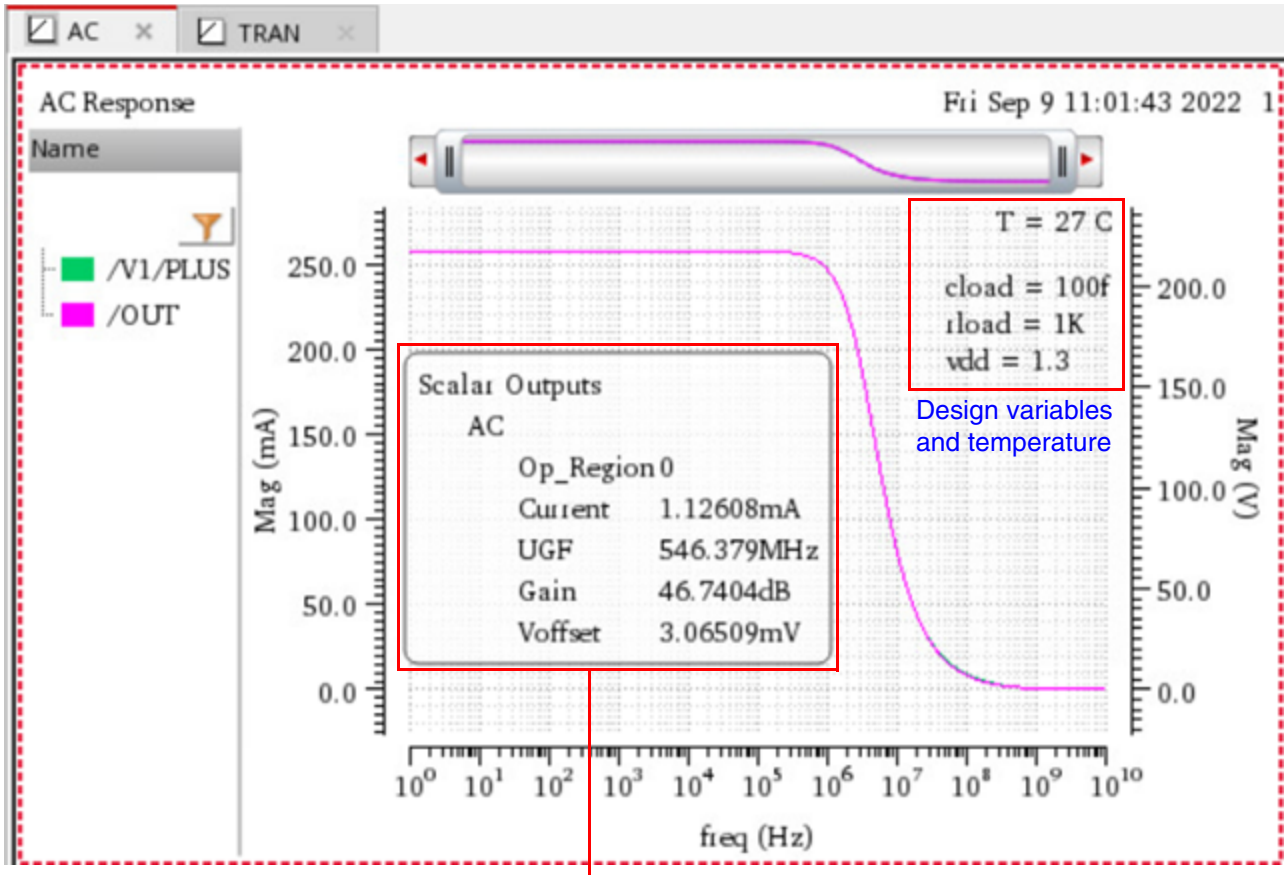
© 2023 Cadence Design Systems, Inc.

Simulation results are displayed in the *Results* tab. The following figure shows that `AC` has five scalar outputs and `TRAN` has four scalar outputs.



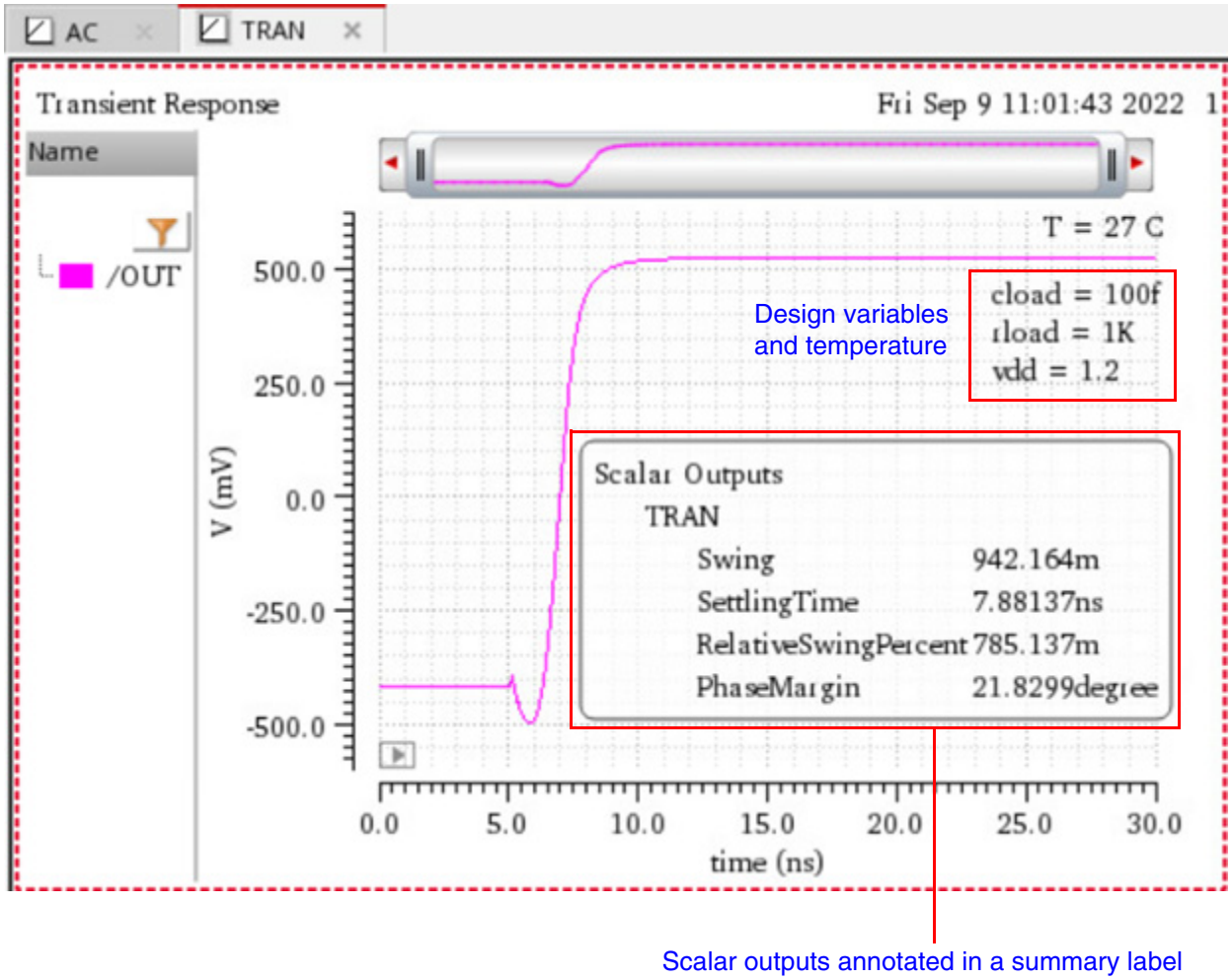
© 2023 Cadence Design Systems, Inc.

The following figure shows scalar outputs of AC annotated in a summary label when you click *Plot All* to plot simulation results. The summary label displays the test name and scalar outputs with their corresponding units.



© 2023 Cadence Design Systems, Inc.

The following figure shows scalar outputs of `TRAN` annotated in a summary label when you click *Plot All* to plot simulation results. The summary label displays the test name and scalar outputs with their corresponding units.



Related Topics

Hiding the Summary Label for Scalar Outputs

Viewing Summary Information of Scalar Outputs

Editing Properties of the Summary Label

scalarOutputs

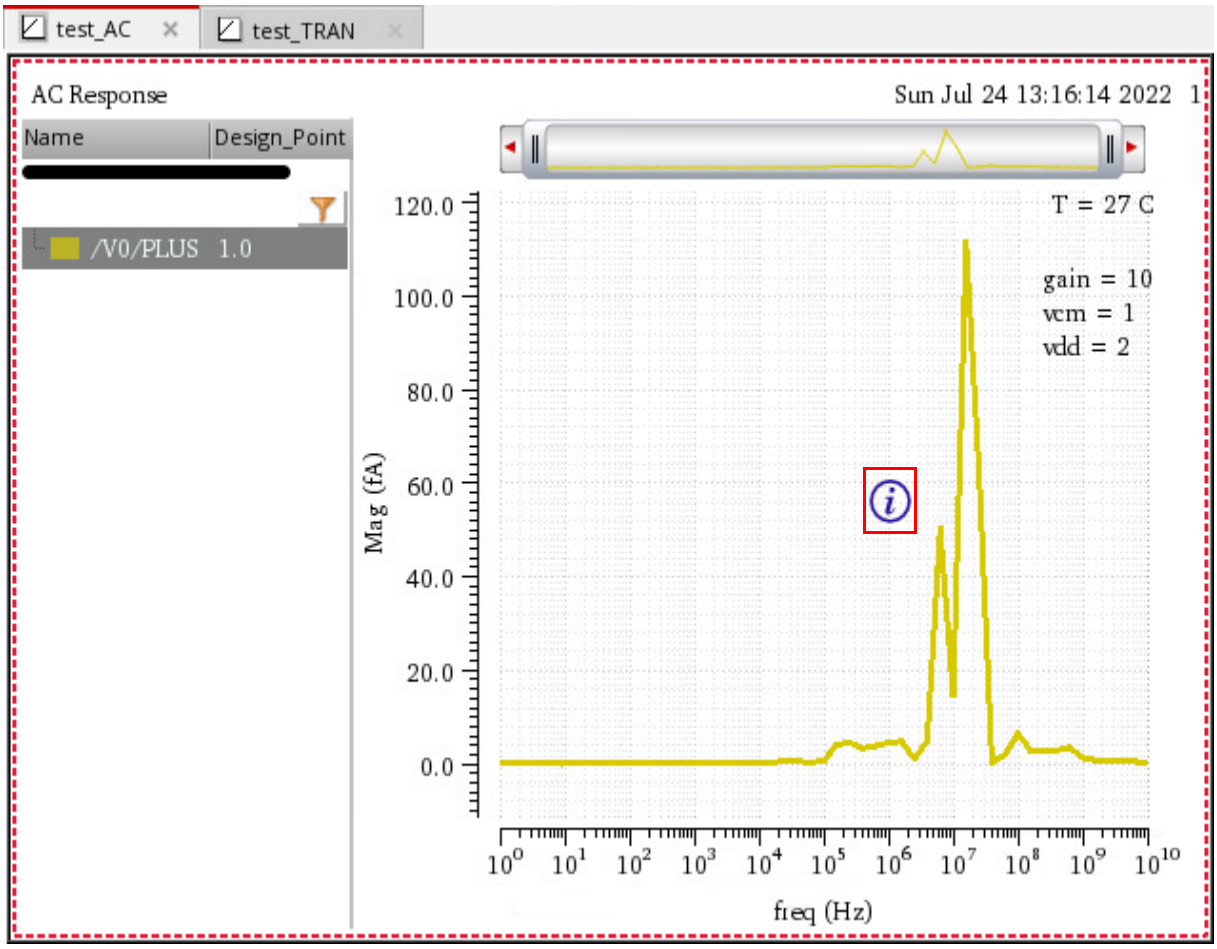
© 2023 Cadence Design Systems, Inc.

Hiding the Summary Label for Scalar Outputs

You can hide the summary label for scalar outputs if you want to focus only on the graph. To hide the summary label for scalar outputs:

1. Right-click anywhere on the summary label and choose *Expand/Collapse*.

   The summary label is collapsed and an icon ⓘ is displayed.



2. Right-click the ⓘ icon and choose *Expand/Collapse* to display the summary label again.

Related Topics

Annotating Scalar Outputs for Single-Point Simulation

Viewing Summary Information of Scalar Outputs

© 2023 Cadence Design Systems, Inc.

Editing Properties of the Summary Label

Viewing Summary Information of Scalar Outputs

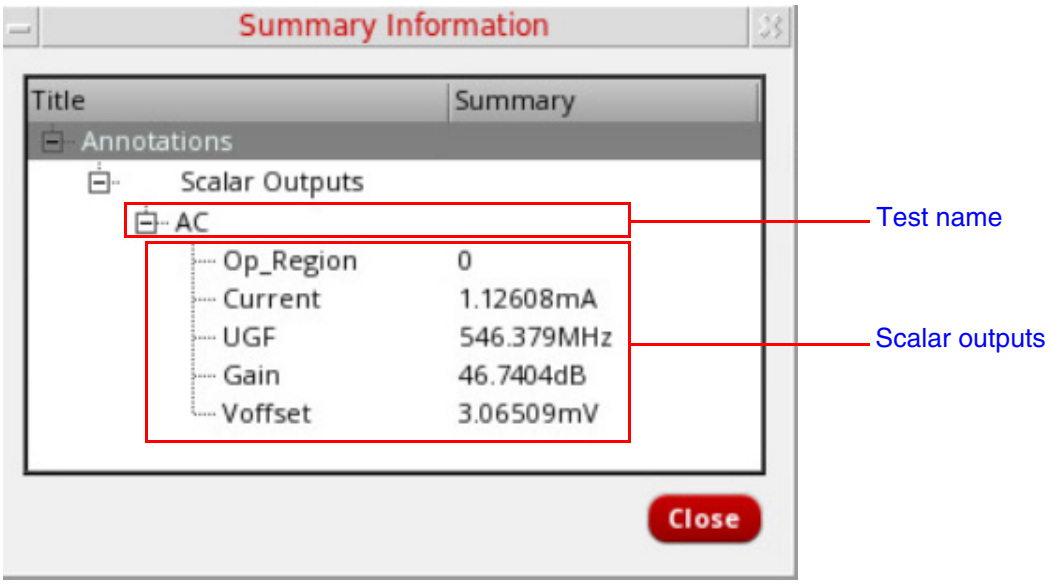
You can view summarized information of scalar outputs in the Summary Information window, which can be placed side-by-side with the graph window.

To view summary information of scalar outputs out of the graph window:

1. Right-click anywhere in the summary label and choose *Summary Information*.

   If the summary label is hidden, right-click the ⓘ icon and choose *Summary Information*.

   The Summary Information window appears, displaying scalar outputs and their values.



2. Click *Close* to close the window.

You can also copy the summary information to the clipboard. To do so, right-click the summary label and choose *Copy*.

Related Topics

Annotating Scalar Outputs for Single-Point Simulation

Hiding the Summary Label for Scalar Outputs

© 2023 Cadence Design Systems, Inc.

Editing Properties of the Summary Label

© 2023 Cadence Design Systems, Inc.

Editing Properties of the Summary Label

To edit properties of the summary label:

1. Right-click anywhere in the summary label and choose *Summary Properties*.

   If the summary label is hidden, right-click the icon and choose *Summary Properties*.

   The Summary Data Properties for Graph form opens.

2. In the *Font/Color* field, click *Default* to open the Define Font form where you can select the font, font style, and font size. You can also choose to apply *Strikeout* or *Underline* effects on the selected font.
3. Click *OK* to apply the changes and return to the Summary Data Properties for Graph form.
4. Click *Foreground* to choose the font color for the summary label.
5. Click *OK* to apply the settings.

Related Topics

Annotating Scalar Outputs for Single-Point Simulation

Hiding the Summary Label for Scalar Outputs

Viewing Summary Information of Scalar Outputs

© 2023 Cadence Design Systems, Inc.

Graph Properties Form (Circular Graphs)

The Graph Properties Form for a circular graph lets you specify various graphs and grid properties. You can also use this form to annotate VSWR (Voltage Standing Wave Ratio) circles and Q contours.

The form contains the following tabs.

Tab	Description
General Tab	Lets you specify graph properties of a circular graph.
Grid Tab	Lets you specify grid properties of a circular graph.
Annotations Tab	Lets you display VSWR circles and Q contours in a circular graph.
Markers Tab	Lets you display the tracking information for the points that you click or point to on a circular graph.

General Tab

The following table describes the fields that are available on the *General* tab of the Graph Properties form.

Field	Description
Title	This section provides options to specify the title and font of a circular graph.
Graph Title	Displays the title of the circular graph at the top-left corner of the graph window or subwindow. The default graph title includes the name of the analysis and the frequency range. You must deselect the *Default* check box to edit the graph title.
Font	Specifies the font properties, such as style, size, and effects, for the graph title.
Graph	This section provides options to specify the properties of a circular graph.

© 2023 Cadence Design Systems, Inc.

Field	Description
Notation	Specifies the notation to be used in the graph labels. Possible values are: ■ *Engineering*: Uses engineering notation. For example, `29E6 Hz`. ■ *Scientific*: Uses scientific notation. For example, `2.9E7 Hz`. ■ *Suffix*: (Default) Uses suffix notation. For example, `29M Hz`.
Legend Position	Specifies the position of the trace legend in the circular graph. Possible values are *left*, *inside*, and *above*. Default value is *left*.
Legend Font	Specifies the font properties, such as style, size, and effects, of the trace legend.
Reload Using Current Context	Specifies whether to reload signals in the graph according to the data from the current in-context results directory. This check box is selected by default. If this check box is not selected, signals are reloaded based on their individual databases.
Frequency Filter	This section provides options to filter graph data by frequency.
Enable	Specifies whether to enable the frequency filter. The frequency filter is disabled by default. If you select this check box, a frequency filter is displayed as a slider bar on top of the circular graph.
Min	Specifies the minimum frequency. This field becomes available only when you select the *Enable* check box. You can click the *Reset* command to reset the minimum frequency to its default value.
Max	Specifies the maximum frequency. This field becomes available only when you select the *Enable* check box. You can click the *Reset* command to reset the maximum frequency to its default value.

© 2023 Cadence Design Systems, Inc.

Grid Tab

The following table describes the fields that are available on the *Grid* tab of the Graph Properties form.

Field	Description
Grid Type	Lets you select the type of grid for a circular graph: ■ *Polar*: Displays the polar grids. ■ *Impedance*: Displays the impedance Smith grid (Z Smith grid). ■ *Admittance*: Displays the admittance Smith grid (Y Smith grid). ■ *Immittance*: Displays the immittance Smith chart (ZY Smith grid).
Circles	This section provides options to specify how you want to draw circles on the polar grid. This section is available only for polar grids.
Number of Circles	Specifies the number of circles you want to draw on the polar grid. The default value is `0`, which indicates that the number of the circles displayed depends upon the plotted data.
Display Circle Grids	Specifies whether to display circular grids. This check box is selected by default.
Display Circle Grid Labels	Specifies whether to display labels for the circular grids. This check box is selected by default.
Display Unit Circle	Specifies whether to display a unit circle, which means a circle with radius `1`. This check box is selected by default.
Radials	This section provides options to specify how you want to draw radials on the polar grid. This section is available only for polar grids.
Number of Radials	Specifies the number of radials you want to draw in each quarter of the polar grid. The default value is `2`.
Display Radials	Specifies whether to display radials. This check box is selected by default.

© 2023 Cadence Design Systems, Inc.

Field	Description
Display Radial Labels	Specifies whether to display labels for the radials. This check box is selected by default.
Display	This section provides options to specify how you want to display circular grids. This section is not available for polar grids.
Coefficient Ring	Specifies whether to display the angles of reflection coefficient in degrees.
Compressed Smith	Specifies whether to display extra horizontal grid lines (arcs) outside the Smith chart boundary.
Detailed Grid	Specifies when to display detailed grid lines. Possible values: ■ *off*: Detailed grid lines are always hidden. ■ *on*: Detailed grid lines are always displayed. ■ *auto*: Detailed grid lines are automatically displayed when zooming in on the Smith chart. This option is not available if the *Compressed Smith* check box is selected.
Grid Labels	Specifies whether to display the grid labels. This check box is selected by default.
Labels	This section provides options to specify properties of grid labels. This section is not available for polar grids.
Imag. Circles	Specifies whether to display labels on the imaginary circles.
Font	Specifies the font properties, such as size, style, and effects, of the grid labels.
Text Size	Specifies the text size for the grid labels. Possible values: ■ *Auto*: (Default) Automatically adjusts the text size for the grid labels. ■ *Fixed*: Uses a fixed text size for the grid labels.

© 2023 Cadence Design Systems, Inc.

Field	Description
Grid Lines	This section provides options to specify properties for the standard and detailed grid lines in the impedance, admittance, and immittance charts. This section is not available for polar grids.
Standard	Lets you select the color, width, and style of the standard grid lines for the impedance, admittance, and immittance charts.
Detailed	Lets you select the color, width, and style of the detailed grid lines for the impedance, admittance, and immittance charts.

Annotations Tab

The following table describes the fields that are available on the *Annotations* tab of the Graph Properties form.

This tab is not applicable for polar grids.

Field	Description
Smith Chart	This section provides options to display VSWR (Voltage Standing Wave Ratio) circles and Q contours on the Impedance, Admittance, and Immittance Smith charts.
VSWR Circles	This section lets you display VSWR circles for the specified VSWR values. You can also specify options to annotate VSWR circles.
VSWR	Specifies the VSWR value for which you want to draw the VSWR circle. Click the '*+*' icon to specify the VSWR value. You can specify multiple VSWR values.
Color	Specifies the color of the VSWR circles. The default color is `#c0c000` (La Rioja).
Width	Specifies the thickness of the VSWR circles. Possible values are: *Fine*, *Medium*, *Thick*, *ExtraThick*, and *ExtraFine*. The default value is *Fine*.
Style	Specifies the style of the VSWR circles. Possible values are: *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. The default value is *Solid*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Q Contours	This section lets you display Q contours for the specified Q values. You can also specify options to annotate Q contours.
Q	Specifies the Q value for which you want to draw the Q contours. Click the '*+*' icon to specify the Q value. You can specify multiple Q values.
Color	Specifies the color of the Q contours. The default color is `##c000c0`(Deep Magenta).
Width	Specifies the thickness of the Q contours. Possible values are: *Fine*, *Medium*, *Thick*, *ExtraThick*, and *ExtraFine*. The default value is *Fine*.
Style	Specifies the style of the Q contours. Possible values are: *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. The default value is *Solid*.

Markers Tab

The following table describes the fields that are available on the *Markers* tab of the Graph Properties form.

Field	Description
Labels	This section provides an option to specify the text format of the tracking marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Tracking Format (Trace)	Specifies the text format to be displayed in the label of a tracking marker. If you leave this field blank, the following default values are used depending upon the selected grid type: ■ Polar grid: `%F%n%C` Frequency and real and imaginary Cartesian values are displayed in the label text of a tracking marker. `%n` indicates that values are displayed in a new line. ■ Impedance (Z Smith) grid: `%F%n%Z` Frequency and impedance values, such as resistance and reactance are displayed in the label text of a tracking marker. ■ Admittance (Y Smith) grid: `%F%n%A` Frequency and admittance values, such as conductance and susceptance are displayed in the label text of a tracking marker. ■ Immittance (ZY Smith) grid: `%F%n%Z%n%A` Frequency, impedance, and admittance values are displayed in the label text of a tracking marker.
Impedance/ Admittance Values	This section provides options to specify graph properties of a circular graph.
Normalize	Specifies whether to normalize the Smith reference values. When you select the *Normalize* check box, the Smith reference values are multiplied with the impedance value specified in the *Characteristic Impedance* field
Characteristic Impedance	Specifies the characteristic impedance value. The default value is `50` ohms.

© 2023 Cadence Design Systems, Inc.

5

Working with the Calculator

The Virtuoso® Visualization and Analysis XL Calculator is a tool that you use to perform calculations on the data generated by a simulator.

This chapter includes the following topics:

- About the Calculator on page 602
- Using the Calculator Graphical User Interface (GUI) on page 607
- Working with Expressions on page 635
- Working with the Calculator in ADE on page 665

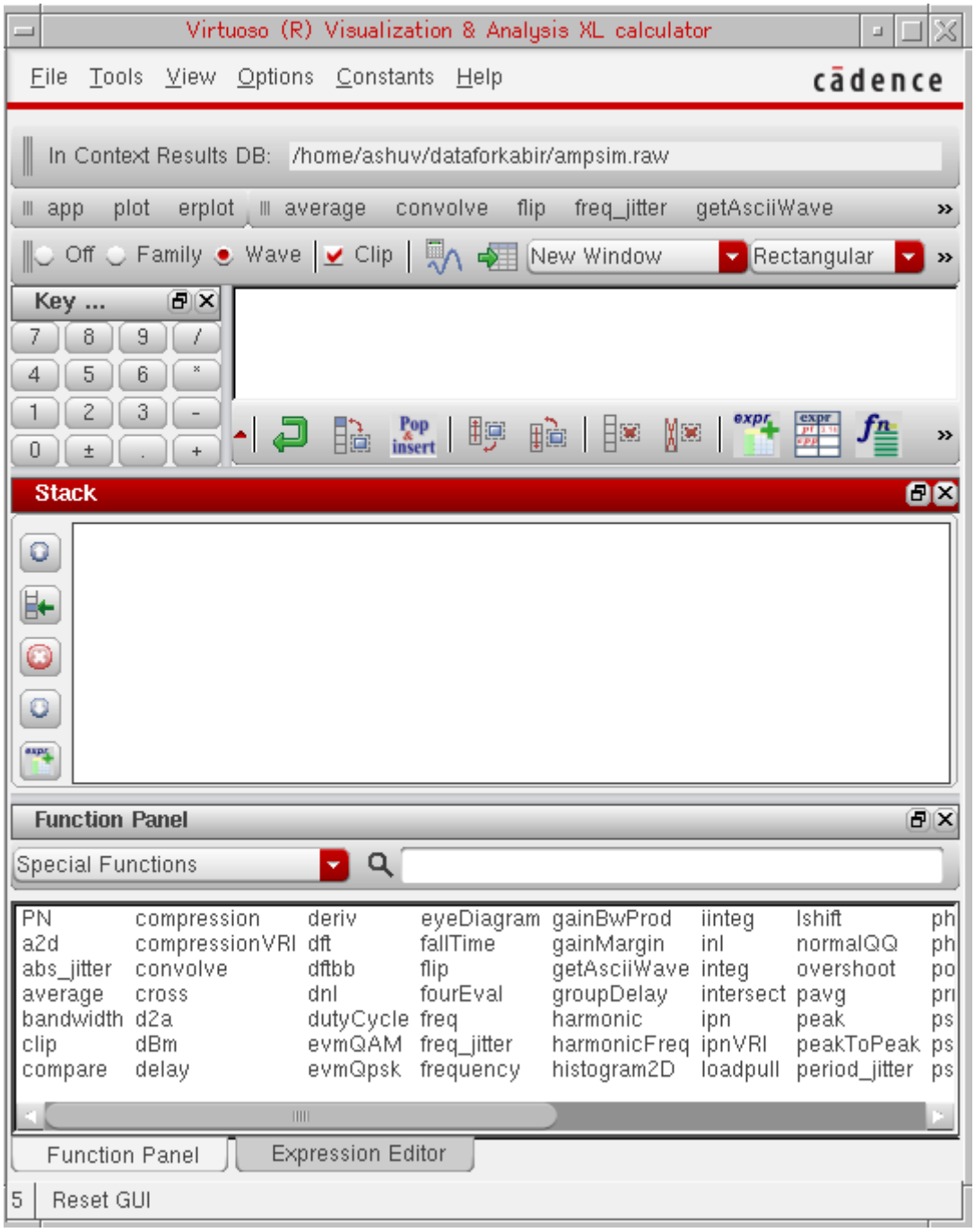
© 2023 Cadence Design Systems, Inc.

About the Calculator

Virtuoso Visualization and Analysis XL Calculator is a tool that you use to perform calculations on signals and datasets generated by a simulator. You can also use the

© 2023 Cadence Design Systems, Inc.

Calculator to build expressions on simulation data and can save these expressions for future use.



You can run the Calculator from Virtuoso Visualization and Analysis XL tool or from the Analog Design Environment (ADE). When you run the Calculator from Virtuoso Visualization and Analysis XL tool, you can open the Calculator in the SKILL mode. In both the modes, you

© 2023 Cadence Design Systems, Inc.

use the Results Browser to access simulation results. The simulation results are the signals generated after the simulation is run in a simulation environment, such as ADE and Spectre. To analyze these signals for a particular condition set, you can build expressions in the Calculator and evaluate them.

For example, if you want to calculate the difference between the output signal (`Vout`) and the input signal (`Vin`) for a simulation, you can build an expression `Vout-Vin` by using the Calculator. To build this expression, you can get the expressions for `Vin` and `Vout` using the Results Browser.

However, if you are working in ADE and want to build expressions for the latest simulation run to analyze design specifications, you can open the Calculator directly from ADE. You then build the required expressions in the Calculator and import the expressions to ADE before you run the simulation. The expressions are evaluated immediately after you run the simulation. The generated output helps you to modify the simulation circuit.

In ADE, you can use the Calculator to build expressions, which are then evaluated in ADE after the simulation is run. When you run the Calculator from Virtuoso Visualization and Analysis XL tool, you can use the Calculator to both build and evaluate expressions.

When you run the Calculator from Virtuoso Visualization and Analysis XL tool, you work on signals, whereas when you run the Calculator from ADE, you typically work on schematic objects, such as nets and terminals. In addition, when you run the Calculator from ADE, you can open the Results Browser from the Calculator and work on the saved simulation results, if required.

Calculator works in two modes—RPN Mode and Algebraic Mode. The two Calculator modes provide different sets of operations for building and evaluating expressions. The default Calculator type is set to RPN Mode.

Calculator contains various dockable assistants, a menu bar, a Buffer, and toolbars that you can use to build and evaluate expressions. For more details about these GUI elements, see Using the Calculator Graphical User Interface (GUI) on page 607.

Opening the Calculator Window

This section describes the different methods that you can use to open the Calculator.

- Opening the Calculator from Virtuoso Visualization and Analysis XL Tool on page 605
  - Opening the Calculator from Virtuoso Visualization and Analysis XL in SKILL Mode on page 605
- Opening the Calculator from CIW in Stand-Alone Mode on page 607

© 2023 Cadence Design Systems, Inc.

■ Opening the Calculator from ADE on page 607

Opening the Calculator from Virtuoso Visualization and Analysis XL Tool

Depending upon the simulation environment you are working in, you can open the Calculator in the SKILL mode from the Virtuoso Visualization and Analysis XL tool. When you run the Virtuoso Visualization and Analysis XL tool, the Results Browser appears. In the Results Browser, you select the saved simulation results on which you want to build and evaluate expressions.

Opening the Calculator from Virtuoso Visualization and Analysis XL in SKILL Mode

Perform the following steps to open the Calculator from Virtuoso Visualization and Analysis XL in the SKILL mode:

1. Start the Virtuoso Visualization and Analysis XL tool by typing the following command in a terminal window:

   ```
   viva -expr skill &
   ```

   The Virtuoso Visualization and Analysis XL window appears. This window consists of Results Browser as an assistant.

   Note: If you type only `viva &` in the terminal window, the default mode is SKILL.

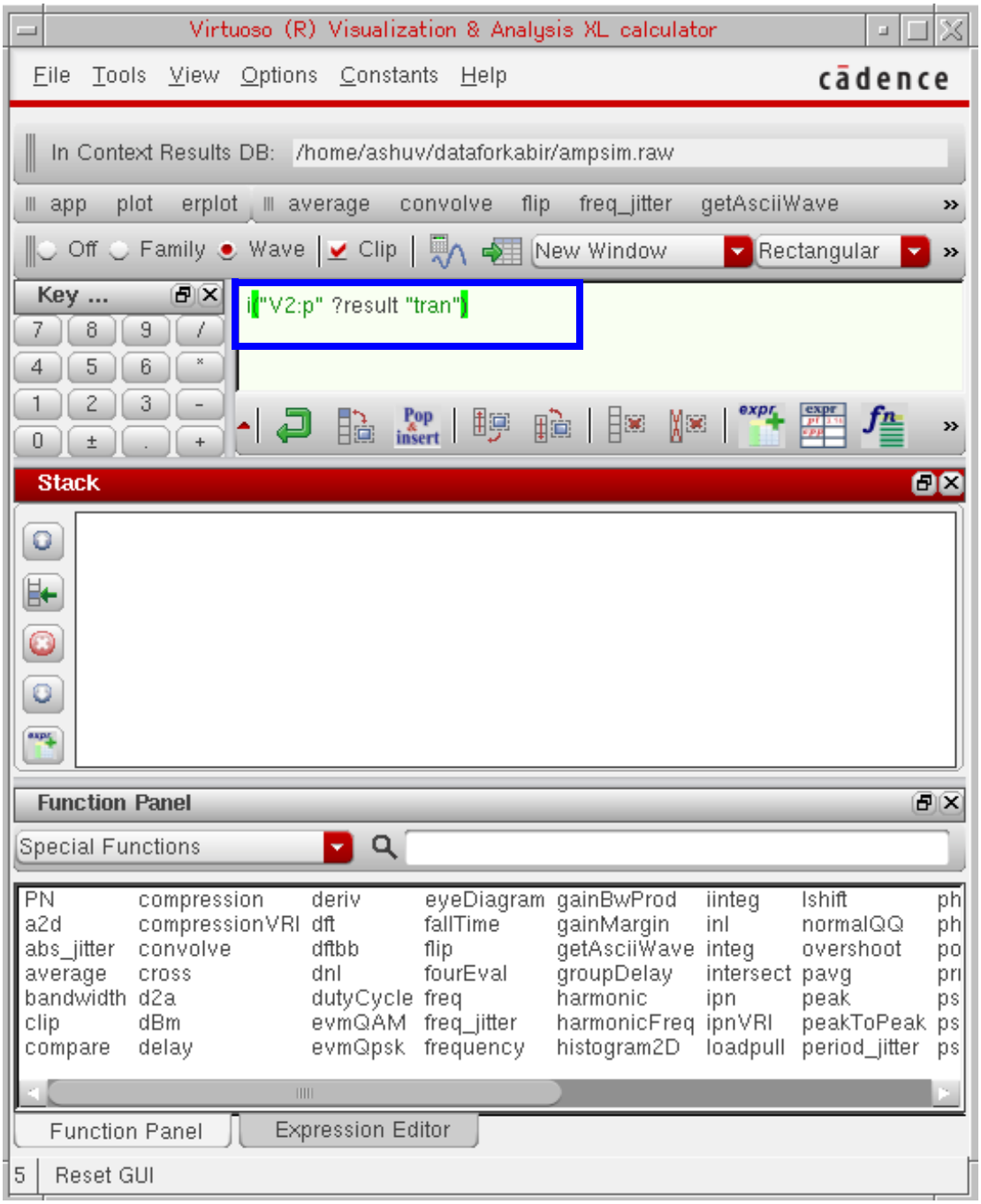
2. In the Results Browser, select the required signal and do one of the following:

   - Right-click the signal and choose *Calculator*.
   - Choose *Tools – Calculator*.

   The Calculator window appears. The signal you have selected in the Results Browser appears in the Buffer.

© 2023 Cadence Design Systems, Inc.

See Selecting Signals and Waveforms to Build Expressions on page 636 to know how to select a signal in the Results Browser.



Note: To verify which result data directory is currently in use, view the *In Context Results DB* field in the Calculator window.

© 2023 Cadence Design Systems, Inc.

Opening the Calculator from CIW in Stand-Alone Mode

To open the Calculator in the stand-alone mode, perform the following:

➡ In the CIW, choose *Tools – Calculator*.

The Calculator window appears. Click the button to open the Results Browser and select the signal for which you want to build expressions.

Opening the Calculator from ADE

The Calculator can be opened in ADE Explorer and ADE Assembler to build expressions to analyze simulation output. For detailed information about how to open and work with the Calculator in ADE, see Working with the Calculator in ADE on page 665.

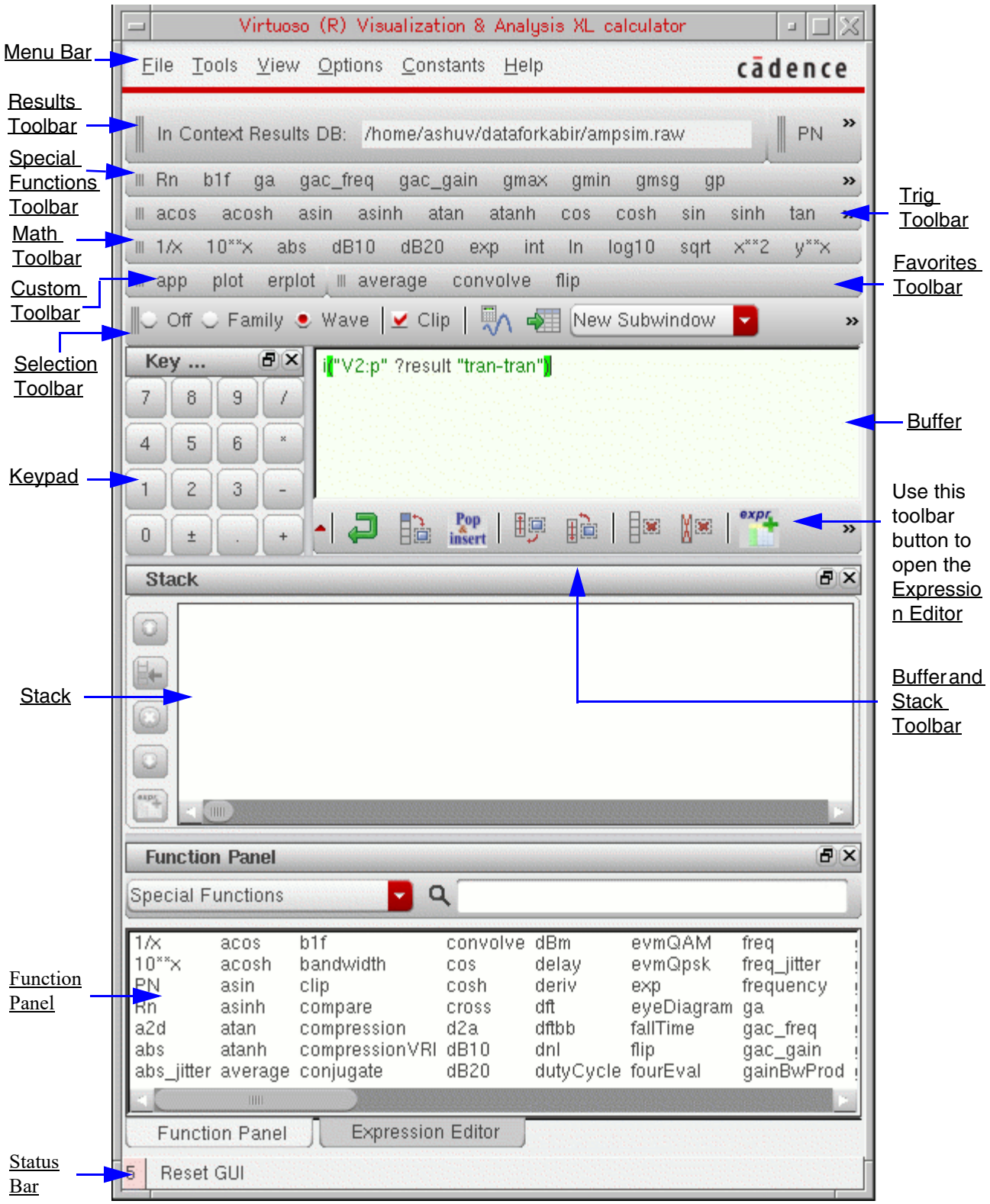
Using the Calculator Graphical User Interface (GUI)

The Calculator graphical user interface (GUI) consists of various toolbars, a menu bar, a Buffer, a status bar, toolbars, and dockable assistants—Expression Editor, Function Panel, Stack, and Keypad— that you use to build and evaluate expressions. You can show, hide, move, and place the dockable assistants and toolbars anywhere inside or outside the main Calculator window. For example, by default, the Keypad is placed in the lower-right corner of the Calculator window. However, if required, you can drag the Keypad to place it next to the Buffer. When you perform drag-and-drop operations on assistants and toolbars, they are automatically positioned according to the available space.

Choose *File – Reset GUI* anytime during the session to restore the default placement of toolbars and assistants.

Note: In the IC 6.1.5 release, the improved version of the Calculator helps you open more than one assistant simultaneously.

© 2023 Cadence Design Systems, Inc.



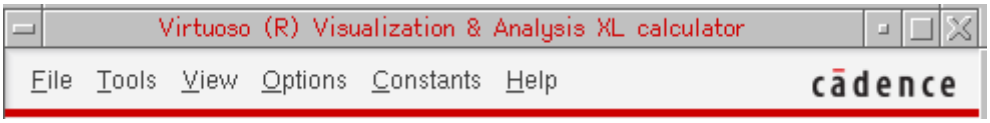
© 2023 Cadence Design Systems, Inc.

This topic describes the following components of the Virtuoso Visualization and Analysis XL Calculator.

- Menu Bar on page 609
- Toolbars on page 612
- Buffer on page 621
- Assistants on page 625

Menu Bar

The menu bar displays the various menus that contain commands for working with the Calculator.



The table below lists a description of the various Calculator menus.

Menu Name	Commands	Description	For More Information, See
File			
	Reset GUI	Resets the Calculator GUI to default settings	
	Close	Closes the Calculator window	
Tools			
	Plot	Plots the signal in the selected graph window	Chapter 4, "Working with Graphs."
	Table	Opens the Virtuoso Visualization and Analysis XL Table.	Working with the Calculator in ADE on page 665
	Browser	Opens the Results Browser window if you want to work on the saved simulation data in ADE	

© 2023 Cadence Design Systems, Inc.

Menu Name	Commands	Description	For More Information, See
	Send to ADE Outputs	Displays the expression created in the Buffer on the ADE Explorer/Assembler *Output Setup* section	
View			
	Display Results Dir	Displays the *In Context Results DB* field on the Result Directory toolbar	Results Toolbar on page 613
	Show Keypad	Displays the Keypad in the lower-right corner of the Calculator window	
	Show Stack	Displays the Stack assistant, which lists expressions that you pushed into the Stack assistant from the Expression Editor and Buffer assistants	
	Math Toolbar	Displays the Math toolbar that includes mathematical functions such as ln and exp	
	Trig Toolbar	Displays the Trigonometry toolbar that includes trigonometric functions such as sin and cos	
	Custom Toolbar	Displays the Custom toolbar that includes buttons to select the mode in which you want to display the output	Custom Toolbar on page 616
	Schematic Selection Toolbar	Displays the Schematic Selection toolbar	Schematic Selection Toolbar on page 617
	Configure Schematic Selection Toolbar	Displays the Configure Schematic Selection toolbar form, in which you select the types of simulation analyses, such as tran, ac, dc, and so on. The function buttons for the selected simulation analyses are displayed on the Schematic Selection toolbar.	
Options			

© 2023 Cadence Design Systems, Inc.

Menu Name	Commands	Description	For More Information, See
	Mode	Sets the Calculator mode *RPN Mode*—Sets the Calculator mode to RPN. Turning this option off changes the Calculator mode to Algebraic *Algebraic Mode*—Sets the Calculator mode to algebraic. Turning this option off changes the Calculator mode to RPN	Building Expressions in the Buffer on page 638
	Notation	Specifies the notation type to be used for displaying data. You can specify one of the following three notation types: *Engineering* - Displays data by using the engineering notation *Suffix* - Displays data by using the suffix notation *Scientific* - Displays data by using the scientific notation	
	Significant Digits	Sets the number of significant digits for the results displayed in the table	
	Edit color	Sets the colors for the various types of Buffer contents, including Results Dir, DataSet name, Calc Function, Node names, Data Access Functions, Number, Brackets, Operators, and Strings	
Constants		Displays a list of constants, such as boltzmann, charge, degPerRad, epp0, pi, twopi, and sqrt2. These constants are used while building expressions.	Adding Constants to Expressions on page 645
Help			
	Contents	Displays the *Virtuoso Visualization and Analysis Tool User Guide*	
	Cadence Online Support	Displays the Cadence customer support website in your default Web browser	

© 2023 Cadence Design Systems, Inc.

Menu Name	Commands	Description	For More Information, See
	Online User Forum (cdnusers.org)	Displays the online users' forum website in your default Web browser	
	Known Problems and Solutions	Displays *Virtuoso Visualization and Analysis Tool Known Problems and Solutions*	
	What's new	Displays *Virtuoso Visualization and Analysis Tool What's New*	
	About Visualization and Analysis	Displays the version number of the Virtuoso Visualization and Analysis XL tool	

Toolbars

Do one of the following to display the Calculator toolbars that are currently hidden:

- From the *View* menu, choose the toolbar you want to display.
- Right-click any toolbar and choose the toolbar you want to display.

Calculator has the following toolbars:

- Results Toolbar on page 613
- Test Toolbar on page 617
- Special Functions Toolbar on page 615
- Buffer on page 621
- Trig Toolbar on page 615
- Math Toolbar on page 616
- Custom Toolbar on page 616
- Favorites Toolbar on page 617
- Schematic Selection Toolbar on page 617

© 2023 Cadence Design Systems, Inc.

- Selection Toolbar on page 618
- Buffer and Stack Toolbar on page 620

Results Toolbar

The Results toolbar displays the *In Context Results DB* field.

In Context Results DB

You build expressions using the data files stored in the results directory. A data file contains output signals that are generated after a simulation is run.

The *In Context Results DB* drop-down list box contains the results directories that are currently open in the Results Browser. By default, the results directory that is set as in-context in the Results Browser is displayed on the top of the list.

You may need to build expressions by using data from one results directory and evaluate the same expression against other results directories. In this case, you can directly open a new results directory from the Calculator by clicking the (Open Results) button displayed on the right of the Results toolbar. When you click this button, the *Select Waveform Database* form appears. In this form, browse and select the results directory that you want to open. For more information about this form, see Opening a Results Directory. The results directory that you open from the Calculator becomes the in-context result directory in the Results Browser and Calculator. This new in-context results directory is now used to build or evaluate the expressions.

Tip

Alternatively, you can also type the complete results directory path name in this field and press the `Enter` or the `Tab` key.

When you change the in-context directory in the Results Browser, the *In Context Results DB* field on the Result toolbar displays the path of the changed in-context directory.

It is recommended that you set the database context to the directory you want to use to build expressions because expressions created using any other result directory may result in an error when evaluated.

© 2023 Cadence Design Systems, Inc.

Important

It is recommended that you set the database context to the directory you want to use to build expressions because expressions created using any other result directory may result in an error when evaluated.

If you do not specify a database context directory, the results directory from which you select the first signal becomes the context results directory, and the directory path is displayed in the *In Context Results DB* field on the Result toolbar.

You can open multiple results directories in the Results Browser, but the *In Context Results DB* field displays the path of the database context directory. For example, if the database context directory is set to the `psf` results directory, this field displays the following value:

```
/home/ashuv/psf
```

Now, if you select a signal from the `psf` results directory, the corresponding expression in the Buffer does not include the results directory name.

```
v(I7.net7" ?result "tran)
```

However, if you select a signal from a results directory that is not the database context directory, the corresponding expression in the Buffer includes the results directory name.

```
v(\\7.net7" ?result "tran" ?resultsDir "./viva_psf)
```

Here, `?resultsDir "./viva_psf"` is the name of the results directory from which the signal is selected.

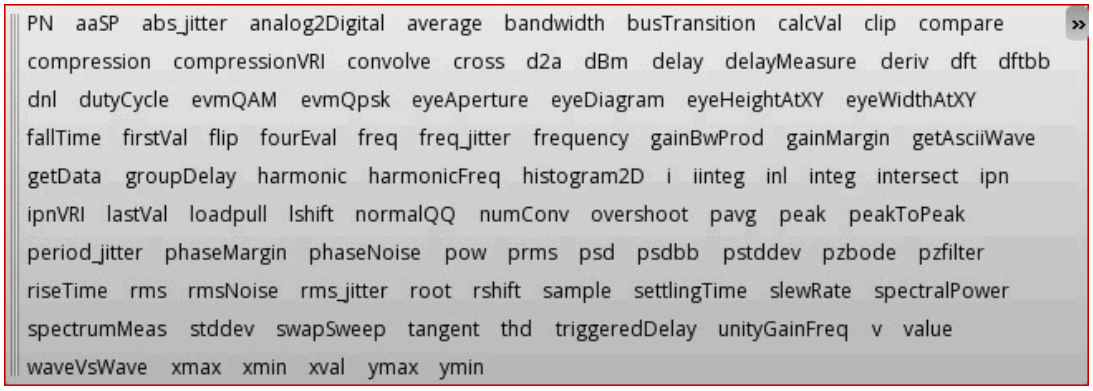
Note: If the Calculator is opened from ADE, the *In Context Results DB* `value` is determined by the results directory currently selected in ADE.

Important

When you create expressions, the expressions include alias name for the analysis to display the results directory name. For example, the expression `v(I7.net7" ?result "tran)` is now displayed as `v(I7.net7" ?result "tran)`.

© 2023 Cadence Design Systems, Inc.

Special Functions Toolbar



The Special Functions toolbar displays all special functions, such as `analog2digital`, `average` and `bandwidth`. By default, this toolbar is hidden and when you select to show the toolbar, it is displayed next to the Result toolbar. To view the complete list of functions, click the arrow button on the toolbar.

The Function Panel displays the special functions by default.

RF Functions Toolbar

This toolbar displays the set of RF functions, such as `Rn`, that can be used while building expressions in the Buffer. This toolbar is hidden by default.

Trig Toolbar

This toolbar displays the set of trigonometric functions, such as acos, that can be used while building expressions in the Buffer.

The Trig toolbar is hidden by default. You can set the `trigToolBar` `.cdsenv` variable to hide or show the toolbar.

© 2023 Cadence Design Systems, Inc.

Math Toolbar

This toolbar displays the set of mathematical functions, such as log, that can be used while building expressions in the Buffer.

The Math toolbar is hidden by default. You can set the `mathToolBar` `.cdsenv` variable to hide or show the toolbar.

Custom Toolbar

This toolbar displays the modes in which the results generated by evaluating expressions can be plotted. With the Custom toolbar, you can also work on the expressions that are saved in Stack or Expression Editor. This toolbar is hidden by default.

app

When you click this button, the Buffer contents are appended to the expression table in the Expression Editor. The Buffer is populated with the latest entry in the Stack. To plot or calculate an expression listed in the expression table, select the expression and click the *Eval* button.

plot

When you click this button, the expression in the Buffer is plotted in the append mode in the selected graph window.

The *plot* command is also available in the *Tools* menu.

Note: This command overrides the option selected in the *plot destination* list box in the signal selection toolbar.

erplot

When you click this button, the Buffer contents are plotted in the selected graph in the replace mode.

Note: This button overrides the *plot destination* selected on the Selection toolbar.

© 2023 Cadence Design Systems, Inc.

Favorites Toolbar

The Favorites toolbar displays the functions that are set as favorite functions in the Function Panel. By default, this toolbar is displayed next to the Custom toolbar and does not contain any function. You can add frequently used functions to this toolbar for quick access.

To add a function to the Favorites toolbar:

➤ Right-click the function in the Function Panel and choose *Add to Favorites*.

To delete a function from the Favorites toolbar:

➤ Right-click the function in the Function Panel and choose *Delete From Favorites*.

When you add a function as a favorite, it is simultaneously added to the Favorites toolbar and to the *Favorites category* in the Function Panel.

Note: The functions you specify as favorite are saved in the `wsFavorite` file in the `cdssearch` path or in the `.cadence` directory available under your home directory. When you open a favorite function, the tool first checks this file in the `cdssearch` path, and if this it is not found, it checks the `.cadence` directory.

Test Toolbar

The Test toolbar is available only when the Calculator is opened from ADE. For more information, see Test Toolbar on page 677.

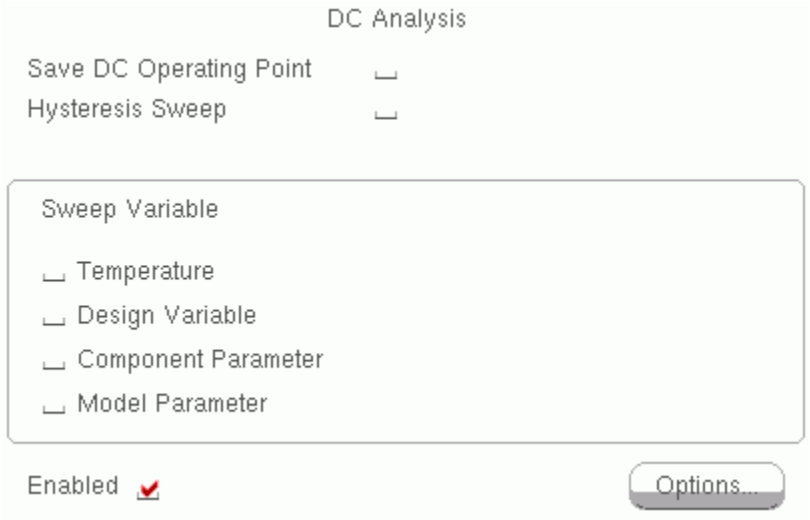
Schematic Selection Toolbar

Note: The Schematic Selection toolbar is available only when the Calculator is opened from ADE.

This toolbar displays a set of function buttons, such as *vt* (voltage transient), that you can apply to schematic objects to build expressions. For more information about the Schematic Selection toolbar, see Schematic Selection Toolbar on page 671.

© 2023 Cadence Design Systems, Inc.

Following are the four cases that describe how *vdc* and *vs* work based on the fields selected in the DC Analysis form:



- Case 1—When the *Save DC Operating Point* check box is selected and a Sweep Variable is set, `vdc("node_name")` returns the voltage of the DC operating point, which is a scalar value, and `vs("node_name")` returns a waveform of the DC sweep.
- Case 2 and 3—When the *Save DC Operating Point* check box is enabled or disabled and no variable is swept, `vs("node_name")` returns the voltage of the DC operating point, which is a scalar value, and `vdc("node_name")` also returns a scalar value.
- Case 4—When *Save DC Operating Point* check box is disabled and a Sweep Variable is set, `vdc("node_name")` returns a waveform of the DC sweep and `vs("node_name")` also returns a waveform.

Selection Toolbar

You can use the Selection toolbar to select a signal from the Results Browser or the graph window. The selected signal is displayed in the Buffer. You then use the tools available on the Selection toolbar to evaluate the selected signal and to define the simulation output format.

The Selection toolbar includes the following options:

© 2023 Cadence Design Systems, Inc.

Off

If selected, the Calculator does not import the following to the Buffer:

- Signals selected in the Results Browser
- Nets selected in the schematic design
- Traces selected in the waveform graph window

Family

If selected, it prompts you to select a signal from a parametric dataset in the Results Browser or a trace from a set of parametric leaf waveforms. This option is used to build expressions for the entire signal family.

Wave

If selected, it prompts you to select a signal in the Results Browser or a trace in the graph window. This option is used to build expressions for the selected trace.

Clip

If selected, the Calculator works on the visible x-axis range of the trace. When you plot the trace after selecting *Clip*, the graph window displays only the clipped part of the trace. If you want the Calculator to work on the complete trace, clear the *clip* check box before importing the signal to the Buffer.

The *Clip* check box is selected by default. However, you can use the `clipSelectionMode .cdsenv` variable to change the default behavior of the check box.

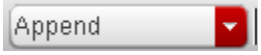
Note: All these options are selected by default. However, you can set the `signalSelection .cdsenv` variable to disable the options.

Evaluate & Plot Expression

This button is used to evaluate the expression displayed in the Buffer. If the result is scalar, it is displayed in the Buffer. If the result is a waveform, it is plotted in the graph window.

Send to Table

This button is used to display the results of the evaluated expressions in a table.



Plot Destination

© 2023 Cadence Design Systems, Inc.

You can use this drop-down list box to specify the window used to plot the result. The list has the following destination options:

- *Append* – Adds the result to the selected graph window
- *Replace* – Replaces the graph in the selected graph window (or subwindow) with the result
- *New Subwindow* – Plots the result in a new subwindow in the selected graph window
- *New Window* – Plots the result in a new graph window

Graph Type

You can use this drop-down list box to specify the type of graph to be used to plot the result. The list has the following graph types available:

- *Rectangular*
- *Polar*
- *Impedance*
- *Admittance*

Open Browser

You can use this button to reopen the Results Browser.

Buffer and Stack Toolbar

This toolbar contains the buttons that help to perform operations on the Buffer and Stack contents. By default this toolbar is displayed below the Buffer and next to the keypad.

The table below describes all the toolbar buttons.

Button	Description
	Displays the Stack assistant
	Pushes the Buffer contents into the Stack
	Swaps the Buffer contents and the first Stack item

© 2023 Cadence Design Systems, Inc.

Button	Description
	Pops the selected Stack item and appends it to the Buffer contents
	Moves the Buffer contents to the bottom of the Stack and moves the first Stack item to the Buffer
	Moves the Buffer contents to the top of the Stack and moves the last Stack item to the Buffer
	Clears the Buffer
	Clears the Buffer and the Stack
	Adds the expression displayed in the Buffer to the Expression Editor
	Opens the Expression Editor
	Displays the Function Panel
	Undo the last command. The maximum number of commands that can be undone is 8, which is set by the `undoStackSize .cdsenv` variable.
	Redo the last undo command.

Buffer

The Buffer provides an area where you can create or edit expressions that are used to analyze the output data or signals generated after a simulation is run. Buffer is a fixed assistant and is displayed next to the Keypad. When you select a signal from the Results Browser, the signal appears in the Buffer. You can now use this signal to build an expression.

While creating or editing expressions in the Buffer, you can use the Keypad, Stack, Function Panel, and Expression Editor assistants or use the keyboard.

After you have created or evaluated the expressions, you can save the expressions by moving them from the Buffer to the Stack or Expression Editor.

- To push the Buffer contents to the Stack, press *Enter* or select on the Buffer and Stack toolbar.

© 2023 Cadence Design Systems, Inc.

- To move the Buffer contents to the Expression Editor, click the button on the Buffer and Stack toolbar.

After you move the Buffer contents to the Stack or Expression Editor, the Buffer still displays those contents. When you provide a new input to the Buffer to build another expression, the Buffer displays the new entry.

To delete a single character in the Buffer, press the *Backspace* or *Delete* key.

If you want to edit an expression saved in the Stack, move the expressions to the Buffer. When you double-click an expression in the Stack, the expression is moved to the Buffer replacing the previous Buffer contents. You can also drag expressions from Stack or Expression Editor to the Buffer.

Expressions saved in the Expression Editor are edited in the expression table.

The Buffer supports the following features:

- Function Names Auto-Completion on page 622
- Multiline Expressions on page 622
- Color Coding on page 623
- Dependent Expressions on page 624
- Incomplete Parentheses, Quotation Marks, and Expressions on page 624

Function Names Auto-Completion

The Buffer supports auto-completion of function names. When you type the first three letters of a function name in the Buffer, all function names starting with these letters are displayed. You select the required function to complete the function name.

To display a list of functions starting with a particular letter, type the letter in the Buffer and press *Ctrl+E* from the keyboard.

Multiline Expressions

Long expressions that cannot fit in one line in the Buffer can be created as multiline expressions. To create multiline expressions, place the insertion point before the character where you want to start a new line and press *Alt+Enter* from the keyboard.

© 2023 Cadence Design Systems, Inc.

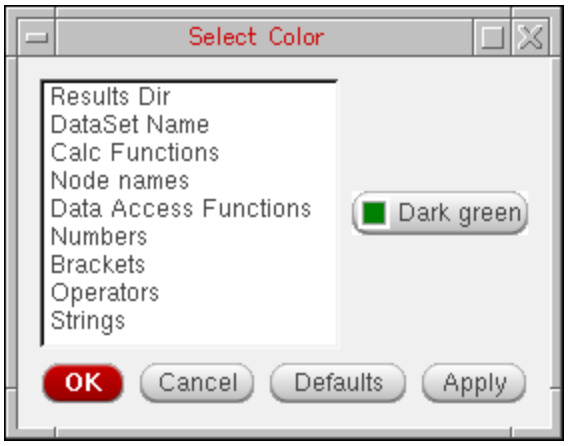
Multiline expressions are moved to the Stack or Expression Editor as a single expression. They are saved and maintained in multiple lines but are treated as a single expression when run.

Color Coding

To differentiate between the various parts of an expression in the Buffer, you can specify different display colors for different parts of the expression, such as results directory, dataset names, functions, node names, data access functions, numbers, brackets, operators, and strings.

1. To set the Buffer color coding, click *Options – Edit Buffer Color*.

   The *Select Color* form appears.



2. Select the category for which you want to set the color.
3. Click the color button on the right to display the available color set.
4. Click the ellipses (...) button to expand the color set.
5. Select the required color.
6. Click *Apply.*

If you do not specify the color settings, the expression is displayed in the default color set.

© 2023 Cadence Design Systems, Inc.

Dependent Expressions

Dependent expressions are the expressions that contain more than one expression. For example, E2=4+E1, where E1=2+3. In this case E2 is a dependent expression because to evaluate the value of E2, you first need to calculate expression E1.

You can create dependent expressions using the Buffer and can store them in the expression table in the Expression Editor.

Note: To avoid errors, ensure that all expressions required to calculate a dependent expression exist in the expression.

For more information about how to work with dependent expressions, see Building Dependent Expressions on page 647.

Incomplete Parentheses, Quotation Marks, and Expressions

When you build an expression in the Buffer, the Calculator checks for missing parentheses, incomplete quotation marks, and incomplete expressions. For example, when you type the opening (left) parenthesis, it is highlighted in red to indicate that it needs to be closed. After you type the closing (right) parenthesis, the parenthesis pair is highlighted in green or in a color that you have configured for parentheses. If you do not correctly close a pair of parenthesis, the Buffer considers the expression as incomplete.

Similarly, expressions that do not contain the closing quotation mark are also considered incomplete, and the entire string after the opening quotation mark, including the quotation mark, is displayed in red. In addition, expressions ending with an operator are also considered incomplete.

You cannot push incomplete expressions to the Stack or Expression Editor. If you press *Enter* to move the incomplete expression in the Buffer to the Stack, the insertion point is placed in the next line in the Buffer assuming that you are building a multiline expression.

To check if all parenthesis pairs in an expression are complete, place the insertion point after each opening parenthesis and observe the color of the highlight—if the pair is complete, it is highlighted in green.

Note: While an expression is being built in the Buffer, the Buffer background color is set to light gray. This indicates that the expression in the Buffer is not yet complete and cannot be moved to the Stack. After the expression is complete, the background color changes to the default color.

© 2023 Cadence Design Systems, Inc.

Video

The Using Calculator Buffer - New Features video demonstrates the new features of the Calculator Buffer.

Assistants

Calculator contains the following assistants, which help in building and evaluating expressions:

- Stack on page 625
- Function Panel on page 627
- Expression Editor on page 631
- Keypad on page 633
- Status Bar on page 634

Stack

Stack is an area where you can store the expressions created in the Buffer.

The table below describes the buttons available to the left of the Stack window.

Button	Description
	Moves the selected item(s) up in the Stack.
	Moves the selected item to the Buffer. Alternatively, you can double-click the selected Stack item to move it to the Buffer.

© 2023 Cadence Design Systems, Inc.

Button	Description
	Deletes the selected item(s) from the Stack. Alternatively, you can also use the *Delete* key to delete the Stack items.
	Moves the selected item(s) down in the Stack.

Pushing Expressions onto the Stack

After you make an entry into the Buffer, you can move the contents to the Stack. To push the current expression onto the Stack click the Enter key or the button.

Displaying the Stack

Click on the Buffer and Stack toolbar to show or hide the Stack.

Clearing Stack and Buffer

There are several ways to clear the Buffer and Stack:

- To remove an item from the Stack without deleting the item from the Buffer, press the *Delete* Key.
- To clear the Buffer without deleting the item from the Stack, click the button.
- To delete an item from both the Buffer and the Stack, click the button.

For more information about the buttons that help you perform Stack and Buffer operations, see the Buffer and Stack Toolbar on page 620.

The `showStack .cdsenv` variable determines whether the Stack is displayed by default.

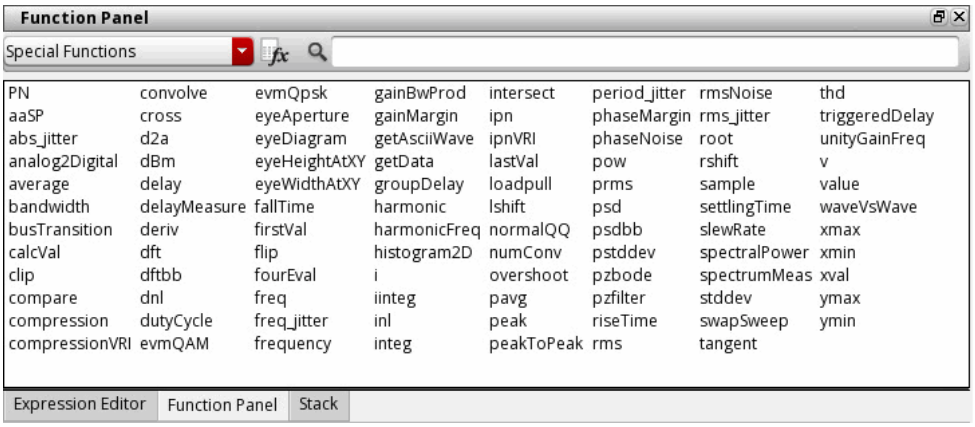
The default number of items stored in the Stack is 8. However, you can set the `StackSize .cdsenv` variable to define the Stack size.

Note: To differentiate between items in the Stack, alternate Stack rows are coded in different colors.

© 2023 Cadence Design Systems, Inc.

Function Panel

This assistant displays a list of functions. The functions displayed are determined by the function category selected in the Function Panel drop-down list box.



Function Categories in the SKILL Mode

The following table lists the function categories available in the SKILL mode.

Category	Functions Available
All	Displays all the available functions, except for the user-defined functions
Favorites	Displays the functions that you specify as favorites. To add a function to the *Favorites* list, right-click the required function in the Function Panel and choose *Add to Favorites.*
Math	1/x, 10**x, abs, dB10, dB20, exp, int, ln, log10, sqrt, x**2, and y**x
Modifier	db10, db20, imag, mag, phase, real, phaseDeg, phaseDegUnwrapped, phaseRad, phaseRadUnwrapped, and conjugate
RF Functions	Rn, b1f, ga, gac_freq, gac_gain, gmax, gmin, gmsg, gp, gpc_freq, gpc_gain, gt, gumx, kf, loadStability, nc_freq, nc_gain, nf, nfmin, rn, s11, s12, s21, s22, and sourceStability

© 2023 Cadence Design Systems, Inc.

Category	Functions Available
Special Functions	average, bandwitdth, clip, compare, compression, compressionVRI, convolve, cross, dBm, delay, deriv, dft, dftbb, dnl, dutyCyle, evmQAM, evmQpsk, eyeDiagram, falltime, flip, fourEval, freq, freq_jitter, frequency, gainBwProd, gainMargin, getAsciiWave, getData, groupDelay, harmonic, harmonicFreq, histogram2D, iinteg, integ, intersect, ipn, ipnVRI, lshift, overshoot, peak, peakToPeak, period_jitter, phaseMargin, phaseNoise,pow, psd, psdbb, pzbode, pzfilter, riseTime, rms, rmsNoise, root, rshift, sample, settlingTime, slewRate, spectralPower, stddev, tangent, thd, unityGainFreq, value, xmax, xmin, xval, ymax, and ymin
Spectre RF Functions	ifreq, ih, it, pir, pmNoise, pn, pvi, pvr, spm, totalNoise, vf, vh, vt, ypm, and zpm
Trigonometric	acos, acosh, asin, asinh, atan, atanh, cos, cosh, sin, sinh, tan, and tanh
AWD Programmed Keys	f1, f2, f3, f4, rf1, rf2, rf3, rf4, rf5, rf6, rf7, and rf8
SKILL User Defined Functions	Includes the functions you define. For more information, see Appendix E, “Defining New SKILL Functions,”.
Expressions	Displays the Expression Editor where you can save and build expressions. For more information, seeExpression Editor on page 631.

For more information about the listed above, see Appendix B, “Calculator Functions.”

The category displayed by default in the Function Panel is *Special Functions.* You can change the default category by using the `defaultCategory .cdsenv` variable.

Using Functions in the Function Panel

The Function Panel has two types of functions:

1. Single parameter Function
2. Multi-parameter Function

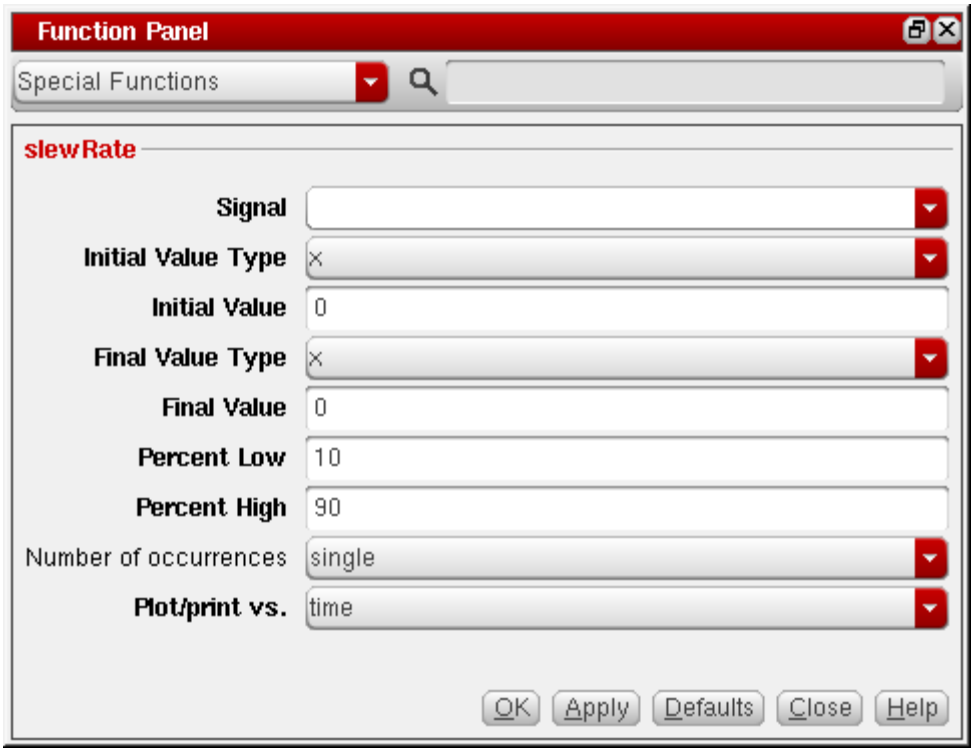
© 2023 Cadence Design Systems, Inc.

If you select a single-parameter function, it can be applied directly to the Buffer contents. For example, *average* is a single parameter function. When you select this function, it is displayed in the Buffer and the Buffer contents become the function arguments.

```
average(i("V2:p" ?result "tran-tran"))
```

If you select a multi-parameter function, the parameter panel for the function appears.

For example, if you select the multi-parameter function, `slewRate`, the various parameters that the function requires appear in the parameter panel, grouped under the function name as shown in the figure below. You can now specify the various parameter values. By default, the parameters contain the values that are set as defaults for that function.



The *Signal* drop-down list maintains a record of all previously selected signals in the current session. It also includes *Buffer* as one of the options. If you select *Buffer*, the Buffer contents are displayed as the *Signal* parameter value.

The default number of signals stored in the drop-down list is *8*. However, you can set the `signalHistorySize .cdsenv` variable to change the number of signals stored.

© 2023 Cadence Design Systems, Inc.

After you specify the parameter values for the selected function, you can do one of the following:

Action	Description
Click *OK*	Creates the expression, adds the expression to the Buffer, and again displays the list of functions in the Function Panel
Click *Apply*	Creates the expression, adds the expression to the Buffer, and keeps the parameter list for the same function open
Click *Defaults*	Discards the changes that you have made and resets the parameters to the default values
Click Close	Closes the parameter panel and displays the list of functions in the Function Panel
Click *Help*	Displays more information about the selected function

Searching for a Function

You can use the *Search* field in the Function Panel to search for any function.

The function search supports category-based search. For example, if you select *Math* in the function category drop-down list and search for a specific function, the search is run only on the *Math* category. To search for a function all the categories, select *All*.

The function search also supports regular-expressions-based search, for example, you can specify *a* to search for the function names that contain the letter *a* or *A* and specify *^a* to search for function names that begin with the letter *a.*

To view the online help for a function, right-click the required function in the Function Panel and choose *Help.*

© 2023 Cadence Design Systems, Inc.

Expression Editor

You can move expressions from the Buffer or Stack assistants to the Expression Editor and can store them for future use. The Expression Editor stores expressions in an expression table.

To open the Expression Editor, do the following:

➡ Click the button on the Buffer and Stack toolbar.

The following table describes the buttons available in the Expression Editor.

Button	Description
	Adds the expression created in the Buffer to the Expression Editor
	Deletes the selected expressions from the expression table
`EP`	Copies the expression selected in the expression table to the Buffer
	Copies the name of the expression selected in the expression table to the Buffer
	Saves the expressions selected in the Expression Editor to a file
	Restores expressions from a file
`Eval`	Evaluates the expression selected in the expression table. **Note:** To cancel the evaluation, press *Ctrl+C*

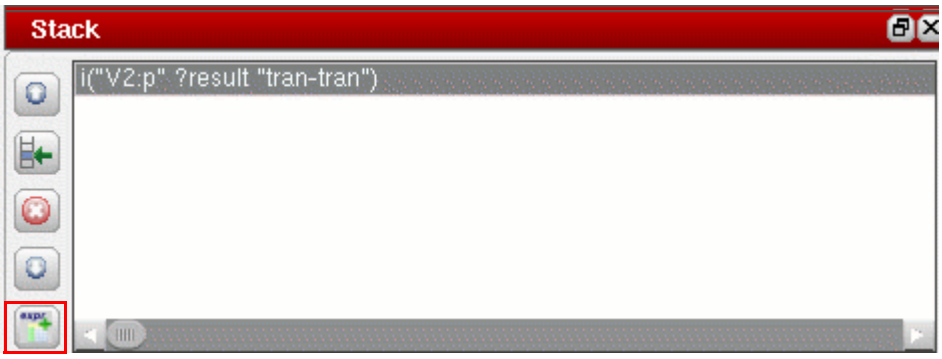
© 2023 Cadence Design Systems, Inc.

Button	Description
	Evaluates the expressions selected in the expression table and displays the results in the Virtuoso Visualization and Analysis XL Table.
	Sends the expressions selected in the expression table to the Outputs table in ADE Assembler or ADE Explorer.

Adding Expressions to the Expression Editor

To add an expression to the Expression Editor, do one of the following:

- From the Buffer. click the command on the Buffer and Stack toolbar.
- From the Stack, select the required expression and click the *expr+* command on the left to move the selected expression to the Expression Editor.



Note: You can also drag expressions from the Stack to the Expression Editor.

When you add an expression to the expression table, the Expression Editor associates a name (signal alias) with the expression—the default expression names are E1, E2, and E3, and so on. If required, you can modify the default expression names. Expression names support alphanumeric values and the first letter of an expression name must be an alphabet.

In Expression Editor, you can also plot circular graphs by selecting the required circular graph type from the *Graph Type* drop-down list on the Selection toolbar.

Editing Expressions in the Expression Editor

To edit an expression in the Expression Editor, double-click the row you want to edit.

© 2023 Cadence Design Systems, Inc.

Deleting Expressions from the Expression Editor

To delete an expression from the Expression Editor, select the expression you want to delete by selecting the check box associated with the expression and click the E- button or press the *Delete* key.

For more information about how to work with expressions in the Expression Editor, see Working with Expressions on page 635.

Sorting Expressions in the Expression Editor

You can sort the expressions listed in Expression Editor based on the expression names or text. To sort the expressions in increasing or decreasing order, click the arrow symbol displayed with each column heading in the expression table.

Keypad

The Keypad contains buttons for numbers and simple arithmetic functions. If you define any function buttons, they too are displayed on the Keypad.

© 2023 Cadence Design Systems, Inc.

To define a function button and associate a function with it, type the following command in the CIW:

```
userButton=envSetVal("viva.calculator" "userButtonX" 'string
"button_name;function_name")
```

Here,

userButtonX defines button number that you want to map to a function. You can define a maximum of 12 function buttons. Therefore, the value of X can be a number from 1 through 12

button_name is the name of the function button that you want to display on the Keypad.

function_name is the name of the function that you want to associate with the button.

For example, if you want to display the *average* function on the Keypad with the button name *MyAvg*, in the CIW type:

```
userButton=envSetVal("viva.calculator" "userButton1" 'string
"MyAvg;average)
```

The defined function button is displayed on the Keypad.

The Keypad is displayed by default. You can change the default behavior by using the `showKeypad .cdsenv` variable.

Status Bar

The status bar displays the status of the action performed in the Calculator window. It also displays warning and error messages. An example is shown in the figure below.

Setting Calculator GUI Using .cdsenv Variables

The variables in the `.cdsenv` file determine the default behavior of the Calculator. Thereafter, the local defaults override the `.cdsenv` variables if `usePreviousGuiSettings .cdsenv` is set to true:

Following are the environment variables that you use to define the settings for Calculator assistants:

- `showKeyPad`

© 2023 Cadence Design Systems, Inc.

- showStack
- rpnMode
- clipSelectionMode
- signalSelection
- plotStyle
- defaultCategory
- mathToolBar
- TrigToolBar
- schematicToolBar
- schematicAnalyses
- userButton
- xLocation
- yLocation
- width
- height

Note: If `usePreviousGuiSettings` is set to false, the `.cshrc` settings override the local defaults.

Working with Expressions

You can build expressions in the Calculator in the RPN and algebraic modes. After the expressions are created, you can edit, save, load, or delete them using the Expression Editor, Stack, and Buffer.

This section contains the following topics which explain how you can work with expressions:

- Selecting Signals and Waveforms to Build Expressions on page 636
- Building Expressions in the Buffer on page 638
- Editing an Expression in Buffer on page 646
- Building Dependent Expressions on page 647

© 2023 Cadence Design Systems, Inc.

- Saving Expressions on page 649
- Dragging Expressions across Buffer and Assistants on page 651
- Loading Expressions from an Expression File on page 651
- Evaluating and Plotting Expressions on page 652
- Creating Custom Calculator Functions on page 655
- Working with the Calculator in ADE on page 665

Selecting Signals and Waveforms to Build Expressions

To build and evaluate an expression, you need to select a signal in the Results Browser or in the graph window. This section describes the following two ways that you can use to select signals:

- Selecting Signals from the Results Browser on page 636
- Selecting Traces from the Graph Window on page 637

Selecting Signals from the Results Browser

1. Ensure that the *Family*, which is a set of parametric leaf waveforms, or *Wave* option on the Selection toolbar is selected.
2. Choose *Tools – Browser*.

   The Results Browser window appears.
3. In the left panel of the Results Browser, double-click the required results directory.

   The associated datasets appear in the list view to the right.

   Note: You can double-click the directory to expand (display the subdirectories) or collapse (hide the subdirectories) it.
4. Double-click the required dataset.

   The associated signals are displayed.
5. Right-click the required signal and choose *Calculator*.

   The signal is displayed in the Buffer along with the dataset name. For example:

   ```
   v("vin" ?result "tran")
   ```

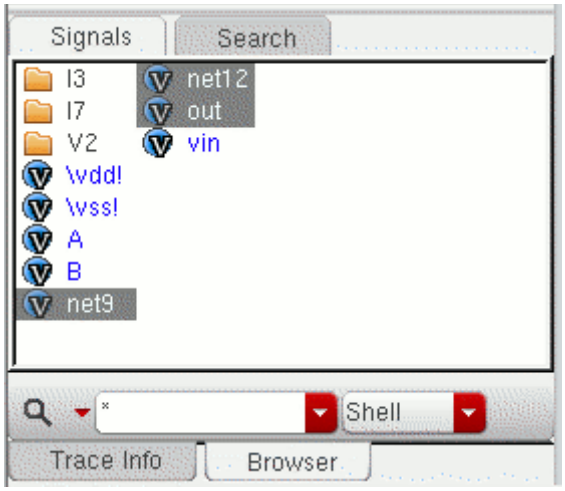
   Here, `vin` is the name of the signal.

© 2023 Cadence Design Systems, Inc.

`?result "tran` is the dataset from which the signal is selected. This `tran` dataset contains the signals generated during the transient analysis of a simulation.

When you send a signal from the Results Browser to the Calculator, the selected signal is displayed in the Buffer. Now, when you send another signal from the Results Browser to the Calculator, the previous signal is pushed onto the stack and the new signal is displayed in the Buffer.

When you select multiple signals in the Result Browser and send them to the Calculator window, the recently selected signal is displayed in the Buffer and all other signals are pushed to the Stack.



For example, if you select three signals in the order—`net9`, `net12`, and `out`— in the Results Browser (as shown in the figure), the signal `out` is displayed in the buffer and the signals `net9` and `net12` are pushed to the stack.

Selecting Traces from the Graph Window

Perform the following steps to select a trace:

1. Ensure that the *Wave* option on the Selection toolbar is selected.
2. In the graph window, select the trace for the signal you want to add to the Calculator. By default, the Calculator works on the visible X-axis range of a zoomed-in trace. If you want the Calculator to work on the complete trace, deselect the *clip* check box before importing the signal to the Buffer.

   An expression for the selected trace appears in the Buffer.

© 2023 Cadence Design Systems, Inc.

- If the selected trace is a leaf parametric waveform, an expression for the leaf is added to the Buffer.
- In the ADE, if the selected trace is the result of an evaluated expression, the SKILL function name for that trace is displayed in the Buffer.

To select a family of traces:

1. Ensure that the *Family* option on the Selection toolbar is selected.
2. In the graph window, select a parametric trace.

   An expression for the parametric family appears in the Buffer.

Building Expressions in the Buffer

You can build expression in the Calculator in the following two modes:

- RPN Mode
- Algebraic Mode

To set the Calculator to work in RPN or Algebraic mode, choose *Options – Mode – RPN Mode/Algebraic Mode*.

This section describes how to build expressions in the RPN and Algebraic modes, how to use functions from the Function Panel while building expressions, and how to add constants or design variables to the expressions.

- Building Expressions in the RPN Mode on page 638
- Building Expressions in the Algebraic Mode on page 639
- Examples of Building Expressions on page 640
- Adding Constants to Expressions on page 645
- Adding Design Variables to the Buffer on page 645

Building Expressions in the RPN Mode

RPN is the default Calculator mode, which can also be set by using the `rpnMode` `.cdsenv` variable.

In the RPN mode, you write expressions without using parenthesis to define priorities for evaluating operators.

© 2023 Cadence Design Systems, Inc.

To understand this better, consider the following expression:

```
(3 + 5) * (7 - 2)
```

In this expression, the parentheses tell you to first add 3 to 5, then subtract 2 from 7, and finally multiply the two results. In the RPN mode, you list the numbers and operators one after the other to form a stack, instead of using parentheses. For example, the above expression is written as follows in the RPN mode:

```
3 5 + 7 2 - *
```

To build this expression, you perform the following Buffer and Stack operations:

1. Add 3 to the Buffer and press *Enter* to push it to the Stack.
2. Add 5 to the Buffer.
3. Apply the + operation. This pops the topmost item, 3, from the Stack and moves it to the Buffer. The Buffer now has the following expression: 3+5. Now when you press any key from the keypad, this expression is pushed to the Stack.
4. Add 7 to the Buffer and push it to the Stack. The Stack contains: 7, 3+5
5. Add 2 to the Buffer.
6. Apply the - operation. This pops the topmost item, 7, from the Stack and moves it to the Buffer. The Buffer now contains the expression: 7-2.
7. Apply the * operation. This pops the topmost item, 3+5, from the Stack and moves it to the Buffer and the following expression is created in the Buffer: (3+5)*(7-2).

Important

In the RPN mode, if the Buffer contains a complete expression and you click a number button on the Keypad, the expression is pushed to the Stack. However, if you press any key from the keyboard, the character associated with the key is appended to the expression in the Buffer.

Note: It is must to add parenthesis to the expression that you want to fetch from the Buffer.

Building Expressions in the Algebraic Mode

To change the Calculator mode to algebraic, choose *Options – Algebraic Mode*.

In the Algebraic mode you build expressions from left to right. When you click an operator or a function, the operator or function is added to the Buffer to the right of the cursor.

To build an expression in the Algebraic mode:

© 2023 Cadence Design Systems, Inc.

1. Ensure that the *Wave* button on the Selection toolbar is selected.
2. Select a signal. For more information, see Selecting Signals and Waveforms to Build Expressions on page 636
3. Click the function you want to use.
   - If you select a single-parameter function, it appears in the Buffer.
   - If you select a multi-parameter function, the parameter panel for the function appears. Enter the required information and click *OK*.

     The expression is displayed in the Buffer. For more information, see Using Functions in the Function Panel on page 628.
4. On the Selection toolbar, click the button to evaluate the expression.

   If the expression evaluates to a scalar, the result appears in the Buffer. If the expression evaluates to a signal, the graph window appears with a trace for the evaluated expression.

Examples of Building Expressions

This section describes how you can build expressions in the RPN and Algebraic mode.

Example of Building Expressions in the RPN Mode

Suppose you want to build an expression in the RPN mode that multiplies two signals:

1. Select the first signal, `signal1`, from the Results Browser. The signal is displayed in the Buffer.
2. Select another signal, `signal2`, from the Results Browser. This pushes `signal1` to the Stack and the Buffer displays `signal2`.
3. Click `*` on the Keypad to multiply `signal1` and `signal2`. This pops `signal1` from the Stack and the following expression is created in the Buffer:

   ```
   signal1*signal2
   ```
4. Click on the Selection toolbar to evaluate the expression. If the result is a scalar, it is displayed in the Buffer. If the result is a waveform, it is plotted in the graph window.

Example of Building Expressions in the Algebraic Mode

To build the same expression in the Algebraic mode, you perform the following steps:

© 2023 Cadence Design Systems, Inc.

1. Select the first signal, `signal1`, from the Results Browser. The signal is displayed in the Buffer.
2. Press *Enter* to move `signal1` to the Stack.
3. Select another signal, `signal2`, from the Results Browser. The signal is displayed in the Buffer.
4. Click `*` on the Keypad to multiply `signal1` and `signal2`. The Buffer now contains the following expression: `Signal2*`
5. Pop `Signal1` from the Stack. This creates the following expression in the Buffer: `signal2*signal1`
6. Click on the Selection toolbar to evaluate the expression. If the result is a scalar, it is displayed in the Buffer. If the result is a waveform, it is plotted in the graph window.

Example of Building Expressions Using Functions in Function Panel

Suppose you want to build an expression for measuring the delay between an input and output signal by using the `delay` function. This example assumes you are working with the SKILL Calculator in the RPN mode:

1. Ensure that the *Wave* option on the selection toolbar is selected.
2. In the Results Browser window, right-click the required input signal and choose *Calculator*. The following signal is displayed in the Buffer.

   ```
   v("vin" ?result "tran")
   ```

3. In the Function panel, select the `delay` function from the *Special Functions* category. The parameter panel for `delay` appears.

   In the parameter panel, `v("vin" ?result "tran")`, which is the last selected signal, appears in the *Signal1* field.

4. In the Results Browser, right-click the `out` signal and choose *Calculator*. The following signal appears in the Buffer:

   ```
   v("out" ?result "tran")
   ```

5. In the Function Panel, do the following:

   a. In the *Signal 2* drop-down list, select *Buffer*. The Buffer content, which is the `out` signal that you selected, appears in *Signal2*.

   b. In the *Edge Type1* field, choose *rising*

© 2023 Cadence Design Systems, Inc.

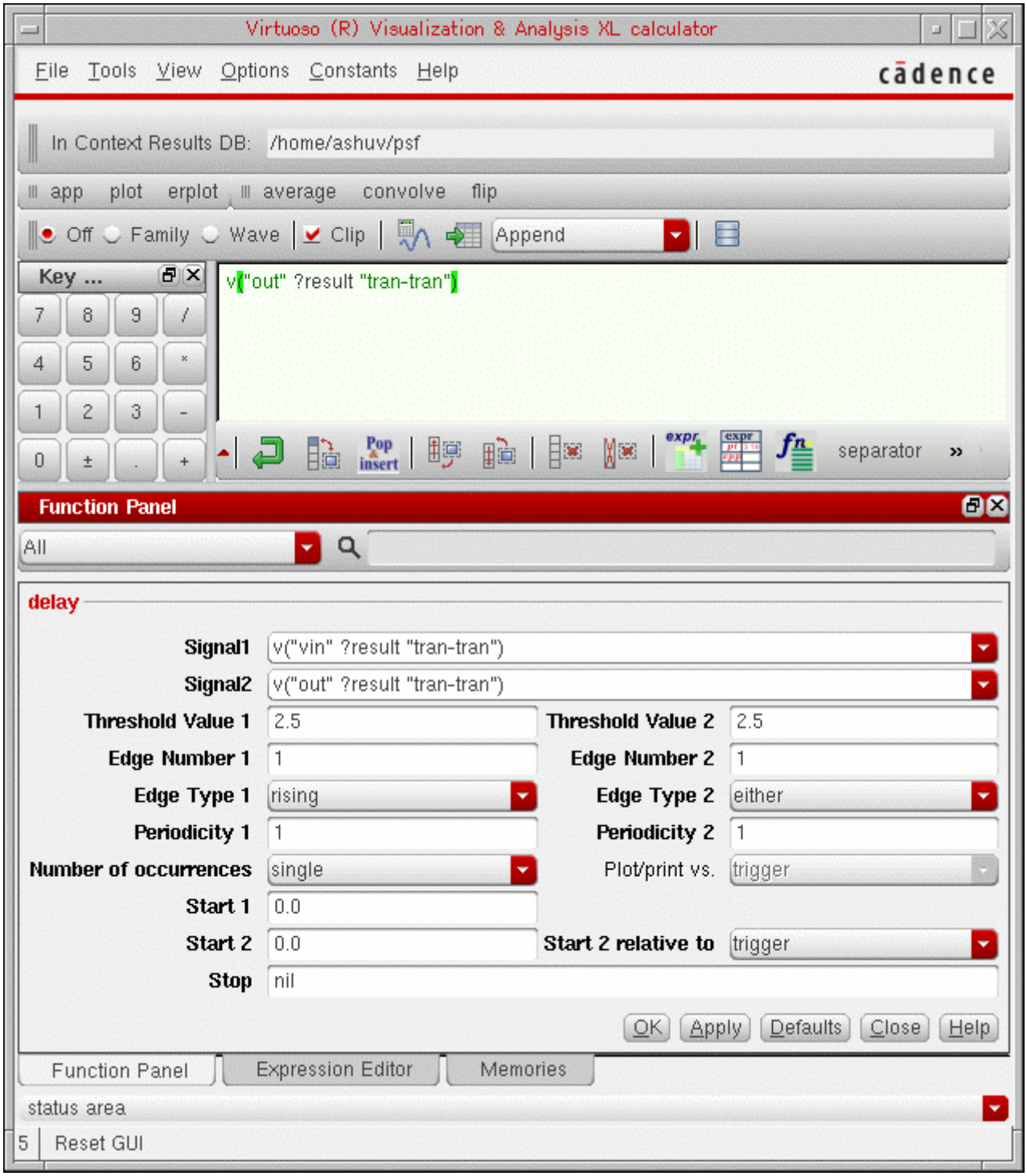
c. In the *Edge Type 2* field, choose *rising*.

© 2023 Cadence Design Systems, Inc.

Ensure that the following values appear in the parameter list:

Field Name	Value	Field Name	Value
Signal1	`vin`	*Edge Type 1*	*rising*
Signal2	`out`	*Threshold Value 2*	`2.5`
Threshold Value 1	`2.5`	*Edge Number 2*	`1`
Edge Number 1	`1`	*Edge Type 2*	*rising*

© 2023 Cadence Design Systems, Inc.



6. Click *Apply*. The following expression appears in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
delay(v("vin" ?result "tran") 2.5 1 "rising" v("out" ?result
"tran") 2.5 1 "rising" 0 0 nil nil)
```

7. Click on the Selection toolbar to evaluate the expression. If the result is a scalar, it is displayed in the Buffer. If the result is a waveform, it is plotted in the graph window.

Adding Constants to Expressions

To add a constant to an expression in the Buffer:

➤ From the *Constants* menu, select a constant. The constant appears in the Buffer.

For example, if you want to add the *Boltzmann* constant to a signal displayed in the Buffer, select *Bolzmann* from the *Constants* menu. *Boltzmann* appears in the Buffer and the signal is pushed to the top of the Stack. Apply the + operation. This pops the signal from the top of the Stack and builds the following expression in the Buffer:

```
v("\\vss! " ?result "tran")+boltzmann
```

For more information about constants, see Appendix D, “Constants.”

Adding Design Variables to the Buffer

You can use design variables in the expressions when the Calculator is opened in the SKILL mode. When you open a simulation result in the Results Browser, the Results Browser displays a directory named *variables* that contains design variables for the simulation. To select design variables, do the following:

1. Select the *Wave* option on the Selection toolbar.
2. In the Results Browser, select the required design variable from the *variables* directory.

The selected design variable appears in the Buffer as shown in the figure below.

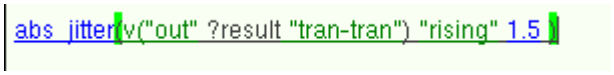
© 2023 Cadence Design Systems, Inc.

Editing an Expression in Buffer

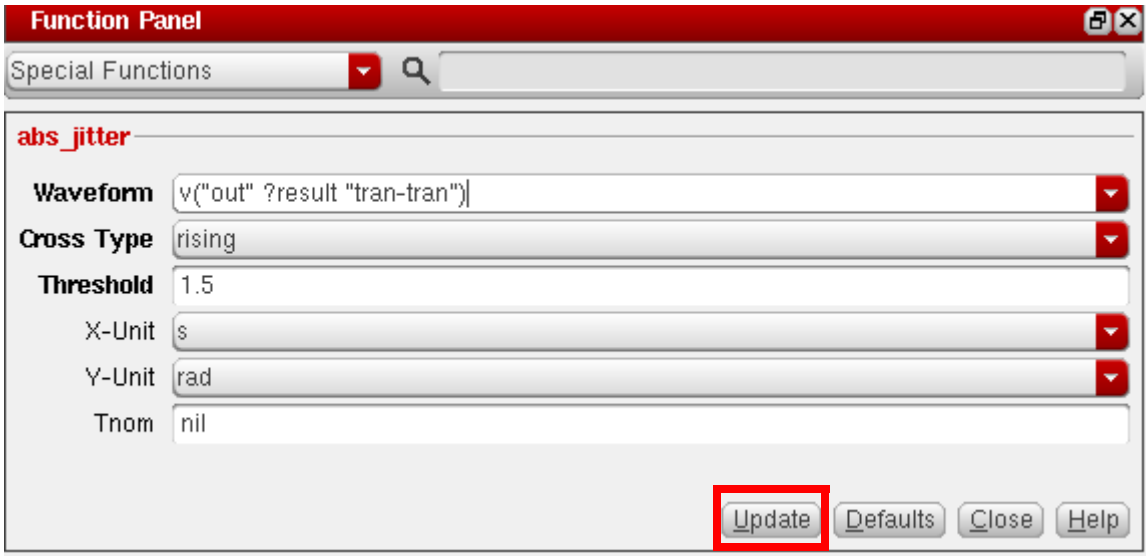
After an expression is created in the Buffer, you can do one of the following to modify the argument values of function applied in the expression:

- Type the new values for the function arguments in the Buffer. To do this, you must know the function syntax and the valid argument values.
- Press the `Ctrl` key and click the function name in the Buffer. The function is opened for editing in the Function Panel, where you can modify the argument values. After the changes are done, click the *Update* button to update the expression in the Buffer.

  Consider the example below in which the `abs_jitter` function is applied on the `out` signal in the Buffer.

  When you place the pointer on the function name, the expression becomes an active link. Now, if you click the function name, the function is opened in edit mode in the Function Panel, as shown in the figure below:

  Note: To edit the function manually in the Buffer, click the function name. The cursor appears at the position where you clicked in the Buffer. You can then edit the function manually.

  You can next modify the values of the function arguments. When done, click the *Update* button to update the expression in the Buffer.

  The method explained above is useful in the following cases:

© 2023 Cadence Design Systems, Inc.

- When you do not know the syntax and the valid values of the function arguments you want to edit.
- When you work on long expressions that include nested functions. For example, the expression below includes four functions: `loadpull`, `compression`, `dB20`, and `harmonic`. If you want to modify the argument values of a particular function in this expression, you need to click only the function name and the function appears in the Function Panel in edit mode.

```
awvRfLoadPull(compression(dB20(harmonic(v("out" ?result "tran-tran") 0)) ?x 0 ?compress 1)
?maxValue nil ?minValue nil ?numCont 9 ?closeCont nil)
```

  If you want to calculate harmonic on some other signal, you just need to click the `harmonic` function in the Buffer and it is opened for editing in the Function Panel, where you can specify a different signal.

Important Points to Note:

- The function template cannot be opened upon clicking the function in the Buffer if a function is applied incorrectly in the Buffer or if the function arguments do not match the original arguments in the template. An error message is displayed in this case.
- You can edit only predefined functions in the Buffer using this functionality. If you have defined your own function, you need to edit the function manually in the Buffer.
- You cannot edit a function expression if it is invalid or blank.

Building Dependent Expressions

Dependent expressions are the expressions that contain more than one expression.

Suppose you want to compare the clip results of input and output signals that are generated during transient analysis. To create a single expression for this task is a tedious process because the task is complex and requires you to perform multiple operations. Therefore, you can divide the task into subtasks and create separate expressions for each subtask. You can then use the names associated with the expressions for the various subtasks to create the final expression.

1. Select the input and output signals from the Results Browser. The expressions for the signals are displayed in the Buffer. Move the expressions for the signals from the Buffer to the Expression Editor and save them with names *sig1* and *sig2* in the expression table.

   sig1=v("in_p" ?result "tran")

   sig2=v("out" ?result "tran")

© 2023 Cadence Design Systems, Inc.

2. Next, in the Buffer, create expressions to apply the *clip* function on both the signals, *sig1* and *sig2*. Store the expressions created for the *clip* function in the Expression Editor with names *cl_s1* and *cl_s2*.

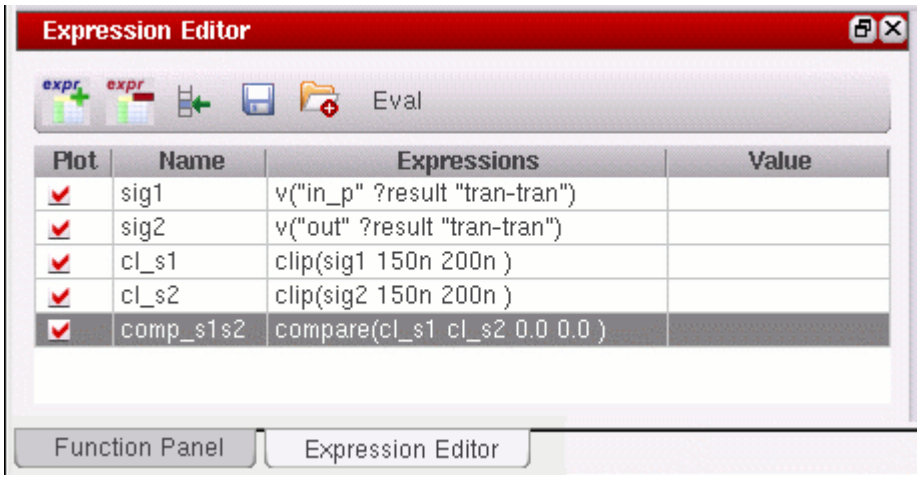
   cl_s1=clip(sig1 150n 200n)

   cl_s2="clip(sig2 150n 200n)

3. Finally, in the Buffer, create an expression to compare the clip results for *sig1* and *sig2* by using the expression names *cl_s1* and *cl_s2*. Move this expression from the Buffer to the Expression Editor and save it with the name *comp_s1s2.*

   comp_s1s2=compare(cl_s1 cl_s2 0.0 0.0)

The expressions created in the expression table for each step are shown in the figure below.



From this example, you can see how dependent expressions simplify the task of creating long and complex expressions.

Note the following when you create dependent expressions:

- Expressions can be created in any order, irrespective of the dependencies. In the above example, it is not necessary to define expression *E1* before expression *E2*.
- An expression can be dependent on multiple expressions. For example, *E2=E1+E3*. Similarly, multiple expressions can be dependent on the same expression. For example, *E2=5*E1* and *E3=2+E1*. Here, both *E2* and *E3* are dependent on E1.
- While creating dependent expressions, ensure that there is no cyclic dependency between the expressions. The following are expressions with cyclic dependency because the expressions are dependent on each other:

  E1=2+E2,

© 2023 Cadence Design Systems, Inc.

*E2=5*E1*

- While evaluating or plotting a dependent expression, it is not required to evaluate or plot all expressions on which the dependent expression is based.
- Error messages and warnings, if any, are displayed in the CIW.

By default, the *plot* check box in the expression table is selected for all expressions, indicating that the expressions have already been evaluated and plotted. If the expression is scalar, the result is displayed in the expression table and if the expression is a waveform, the result is plotted in the graph window.

To plot an expression again or to edit an expression, move the expression from the Expression Editor to the Buffer. To do this, deselect the plot check box and click the *Copy Expression to Calculator Buffer* button in the Expression Editor. This moves the expression name (signal alias) to the Buffer. To move the entire expression to the Buffer, drag the expression from the Expression Editor to the Buffer.

To evaluate expressions in the expression table, click the *Eval* button. The corresponding result column displays the results generated. If you do not want to evaluate any expression, clear the corresponding *plot* check box.

Important

Ensure that all expressions that are used in calculating a dependent expression exist in the expression table else an error occurs.

Video

The Building Dependent Expressions Using the Expression Editor video demonstrates how to create and use dependent expressions using the Expression Editor.

Saving Expressions

This section describes how you can save expression created in the Buffer to a file and to the Expression Editor. This section covers the following topics:

- Saving Expressions to a File on page 650
- Saving Expressions to the Expression Editor on page 651

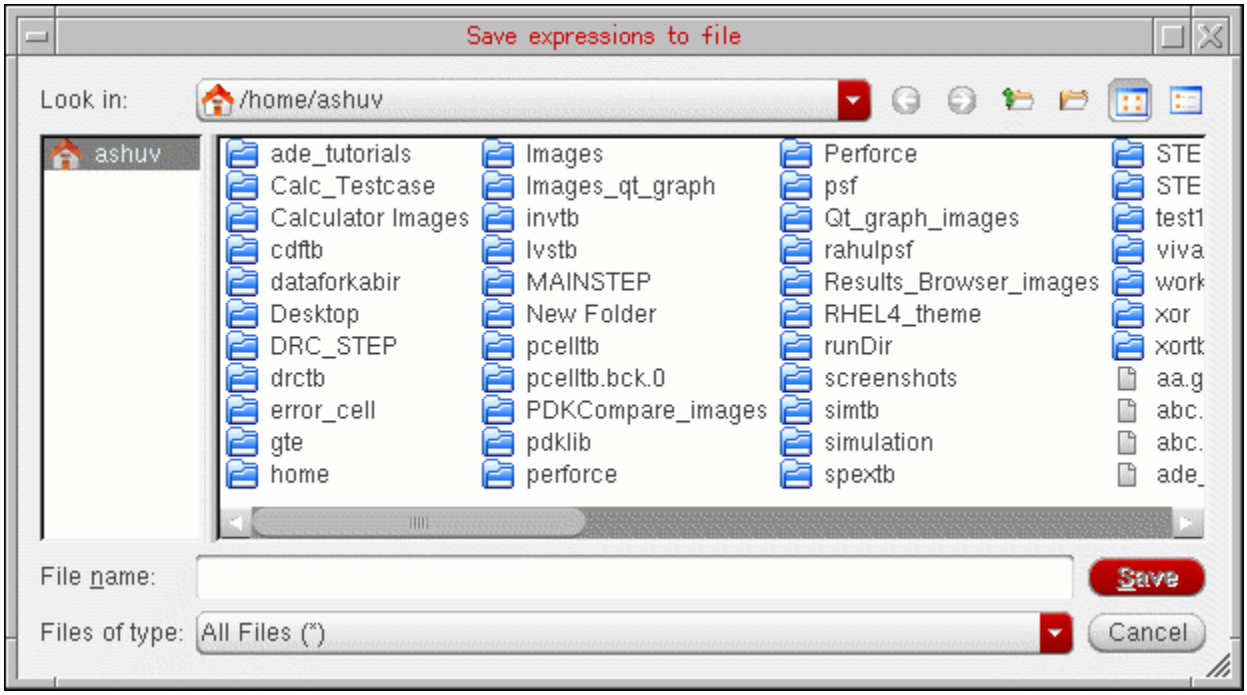
© 2023 Cadence Design Systems, Inc.

Saving Expressions to a File

To save an expression in the Expression Editor to a file, perform the following steps:

1. In the Expression Editor, click the button.

   The *Save expressions to file* form appears.



2. In the *Look in* field, select the directory where you want to save the expression file.
3. Do one of the following:
   - To overwrite an existing expression file, select that file from the list box below the *Look in* field.
   - To create a new file, in the *File name* field, type a name for the expression file that you want to create.
4. Click *Save*.

   The Virtuoso Visualization and Analysis XL tool saves the expression list to the file.

To automatically save the expressions in the current session to the `defaultVarFileName` file when you close the Calculator or exit the Virtuoso Visualization and Analysis XL tool, set the `writeDefaultVarFileOnExit.cdsenv` variable to `true`.

© 2023 Cadence Design Systems, Inc.

Saving Expressions to the Expression Editor

You can save expressions created in the Buffer to the Expression Editor. To add an expression from the Buffer to the Expression Editor, click the command on the Buffer and Stack toolbar. For more information, see Expression Editor on page 631.

Dragging Expressions across Buffer and Assistants

Calculator supports the following drag-and-drop operations:

- You can drag expressions from the Expression Editor and place them into any text area. If you drag expressions outside the Calculator window without specifying any destination text file, you are prompted to provide a filename and its location. You can create a new text file or you can append expressions to an existing text file.
- You can drag expressions from the Expression Editor to the Buffer.
- You can drag expressions from the Stack to the Expression Editor to add these expressions to the expression table in the Expression Editor. However, you cannot drag expressions from the Expression Editor to add them to the Stack. To drag expressions from the Expression Editor to the Stack, you need to first drag the expressions from the Expression Editor to the Buffer. You can then press *Enter* to push the expressions to the Stack.
- If you drag multiple expressions at a time from the Expression Editor to the Buffer, the first expression is added to the Buffer and the remaining expressions are moved to the Stack.
- When you drag expressions from the Expression Editor or Stack to the Buffer, the expressions are appended to the current expression in the Buffer.

You cannot perform drag-and-drop operation on expressions in the Buffer. To move expressions from the Buffer to the Expression Editor or Stack, use Buffer and Stack toolbar buttons.

Loading Expressions from an Expression File

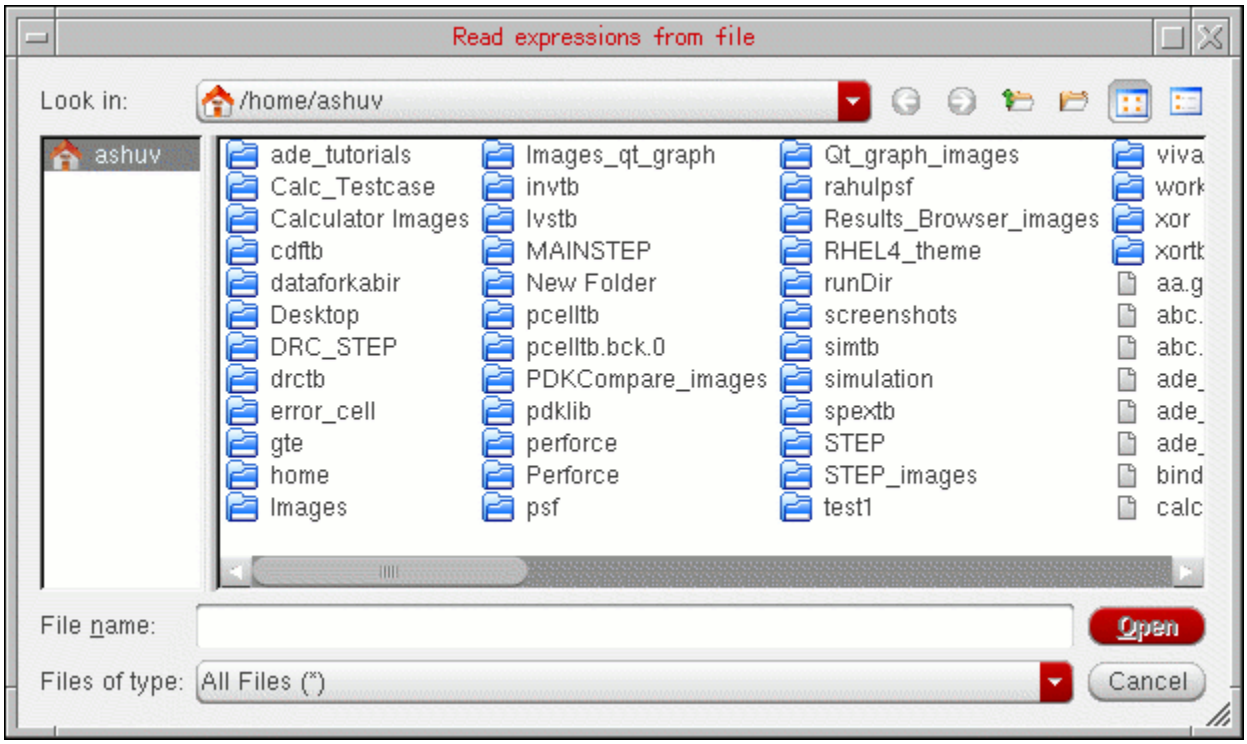
You can load an expression file to the expression table in the Expression Editor. The file that you load replaces the current set of expressions in a session, and you lose any unsaved information.

To load an expression file to the Expression Editor, perform the following steps:

1. In the Expression Editor, click .

© 2023 Cadence Design Systems, Inc.

The *Read expressions from file* form appears.



2. In the *Look in* field, select the directory which contains the required expression file.
3. Do one of the following:
   - From the list box below the *Look in* field, select the expression file you want to open.
   - In the *File name* field, type the name of the file you want to open.
4. Click *Open*.

   The Virtuoso Visualization and Analysis XL tool loads the specified expression file.

If the `readDefaultVarFileOnStartup.cdsenv` variable is set to `true`, The Virtuoso Visualization and Analysis tool automatically loads the `defaultVarFileName` file.

Evaluating and Plotting Expressions

You can evaluate or plot the expressions in the Buffer and the Expression Editor. When you evaluate or plot expressions in the Buffer and if the result is a scalar, it is displayed in the Buffer. If the result is a waveform, it is displayed in the graph window.

© 2023 Cadence Design Systems, Inc.

When you evaluate or plot expressions in the Expression Editor and if the result is a scalar, it is displayed in the *Value* column in the expression table. If the result is a waveform, it is displayed in the graph window.

To evaluate and plot an expression in the Expression Editor,

1. In the Expression Editor, select the expressions you want to evaluate and plot by selecting the *plot* check box corresponding to the expression.
2. Click the Eval button.

   The Calculator evaluates and plots the selected expressions.

Displaying Results in a Table

You can display the results in a table after evaluating expressions.

On the Selection toolbar, you can use the following options to select the destination table where the results are displayed:

- *Append*—Adds the result to an existing table
- *Replace*—Replaces the existing table with the result
- *New Window*—Displays the result in a new table
- *New Subwindow*—Displays the result in a new table

You can display the expressions or results evaluated by the Buffer in a table.

To display results in a table, do one of the following

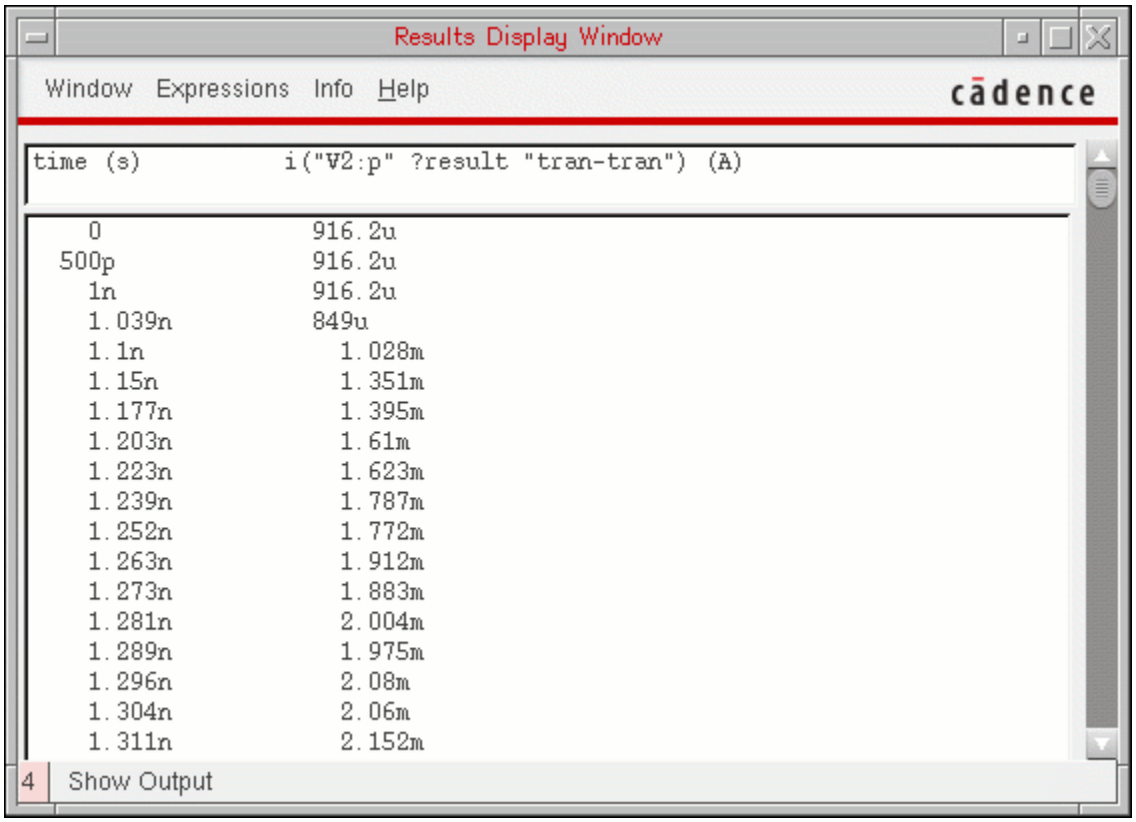
➥ Click the button on the Selection toolbar.

The *Results Display Window* appears.

➥ Choose *Tools – Table.*

© 2023 Cadence Design Systems, Inc.

The *Results Display Window* appears.



© 2023 Cadence Design Systems, Inc.

Creating Custom Calculator Functions

The Function Panel in Calculator provides a set of standard functions that you can use to evaluate and analyze waveform results. You can also create your own functions (custom functions) and add them in the Function Panel. Before you start, you must have a SKILL file in which the SKILL definition of the function you want to create is defined.

The creation of custom Calculator function involves two parts:

- Specify a SKILL file, which includes the SKILL definitions of the functions you want to create. This SKILL file is mandatory, which means if a SKILL file is not available, you cannot create the custom function.
- Create the user-interface template for the custom function, which includes information about function arguments and their valid values. You can either use the same SKILL file, or specify a separate `.ocn` file to add the UI template.

This section includes the following topics to describe the custom Calculator functions in detail:

- Adding Custom Calculator Functions
- Using the Calculator Function Template UI Builder Form
- Viewing Custom Functions
- Sharing Custom Functions

Video

The Custom Calculator Functions video is available on Cadence Online Support that explains this feature in detail.

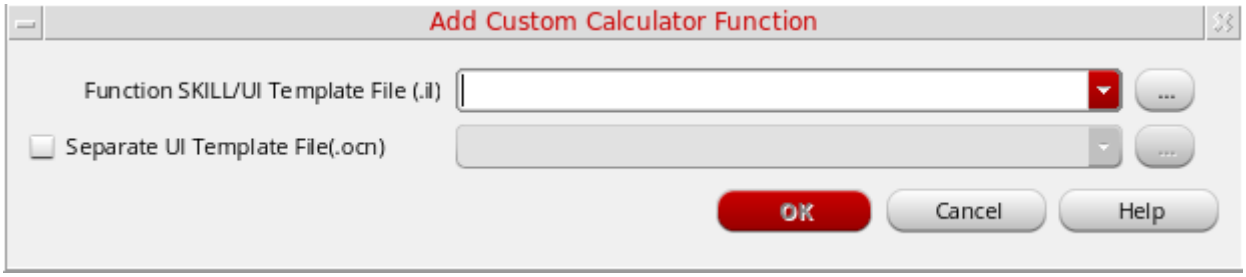
Also see: Sharing Custom SKILL Calculator Functions

Adding Custom Calculator Functions

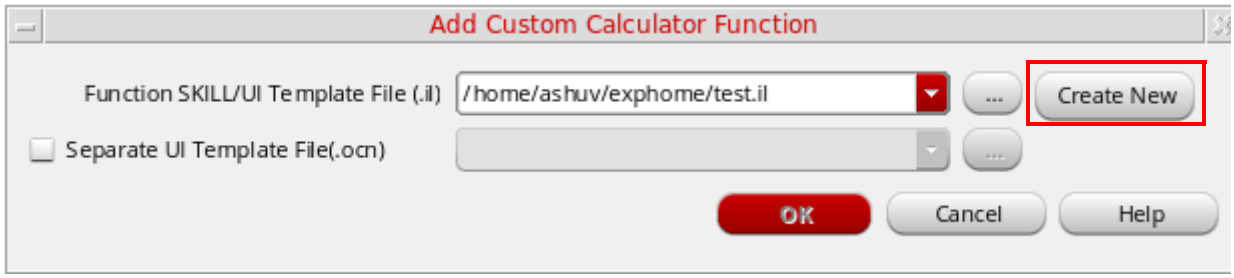
To add a custom function, perform the following steps:

© 2023 Cadence Design Systems, Inc.

1. In the Function Panel, click the button. The *Add Custom Calculator Function* form is displayed.



2. In this form, in the *Function SKILL File/ UI Template File (.il)* field, specify the name of the SKILL (`.il`) or SKILL++ (`.ils`) file in which the SKILL code of the function to be created is defined.
   You can also click the ellipses button to select the SKILL file from the Select SKILL File form. All the files that you specified in the previous session are listed in this drop-down. As an example, consider that a `test.il` SKILL file is specified in this field that includes the definition of the `psd` (power spectral density) function. After you specify the SKILL file, the *Create New* button appears next to this field.



3. To define or use the template UI file, which includes the information about the template (user-interface) of the function to be created, you have the following two options:

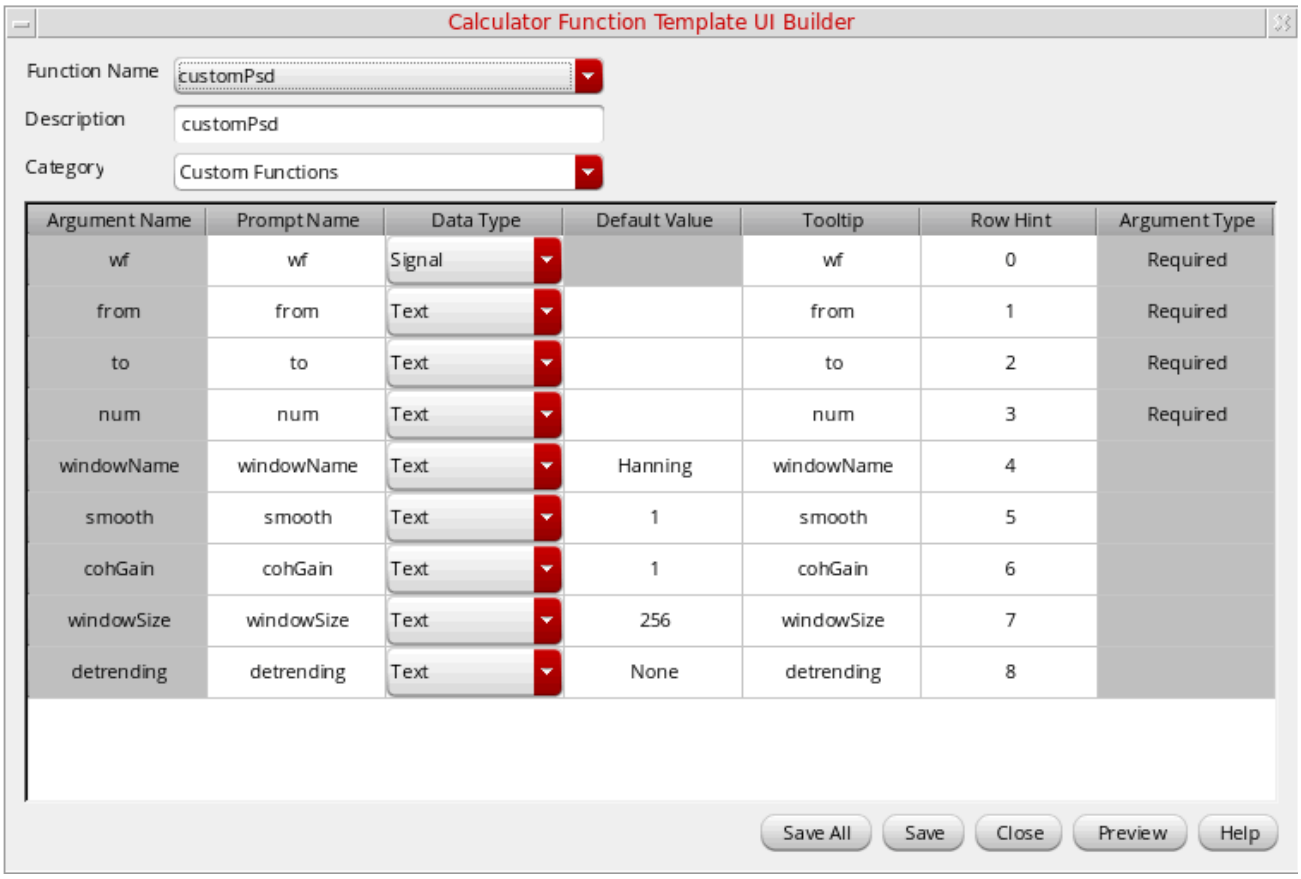
   - Use the SKILL file selected in the *Function SKILL/UI Template File* field.

     This option is useful when you have multiple functions defined in the SKILL file. It helps you create or use the same UI template for all the SKILL functions, without the need to create or use individual `.ocn` files for each function.

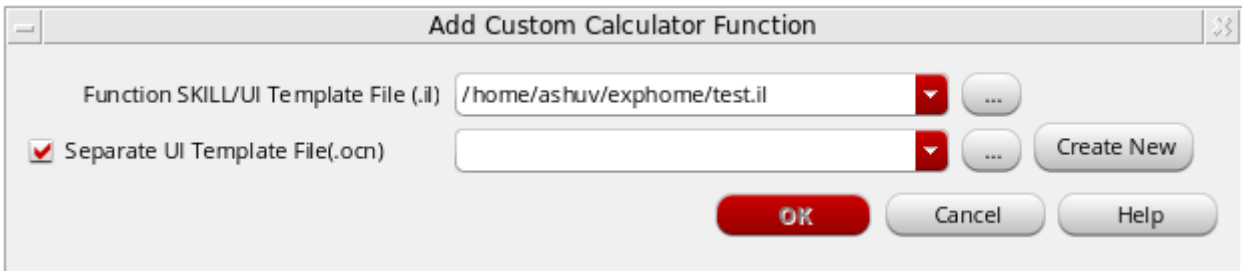
     When you use this option, if the UI template does not exist in the specified SKILL file, click the *Create New* button displayed next to this field to define the UI template. When you click this button, the Calculator Function Template UI Builder

© 2023 Cadence Design Systems, Inc.

form appears, as shown in the figure below. For more information about how to use this form, see Using the Calculator Function Template UI Builder Form on page 658.



- Specify a new UI template `.ocn` file in the *Separate UI Template File (.ocn)* field, which includes information about the template (user-interface) of the function to be created. You can also click the ellipses button to select the SKILL file from the Select Template File form.

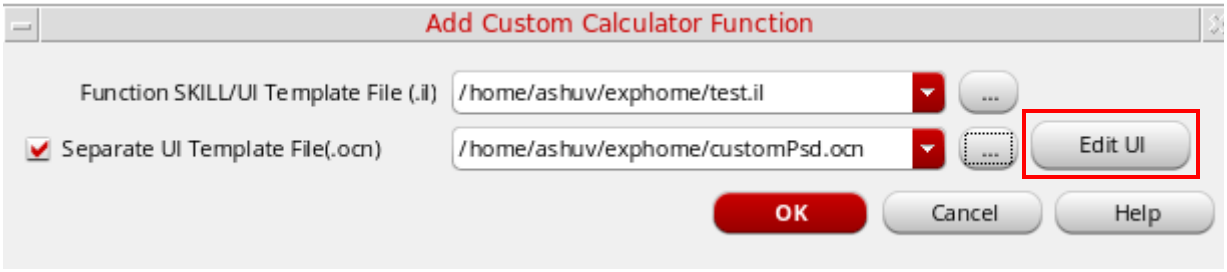


If you do not have an existing `.ocn` file corresponding to the function that you want to create, click the *Create New* button, which is displayed next to this field, to create

© 2023 Cadence Design Systems, Inc.

a new `.ocn` file based on the definitions in the specified SKILL file. This button becomes active after you specify the SKILL file.

If you already have an `.ocn` file, you can specify the path of the file. The *Create New* button is then changed to *Edit UI*, as shown in the figure below. You can click the *Edit UI* button to edit the UI defined in the specified SKILL file.



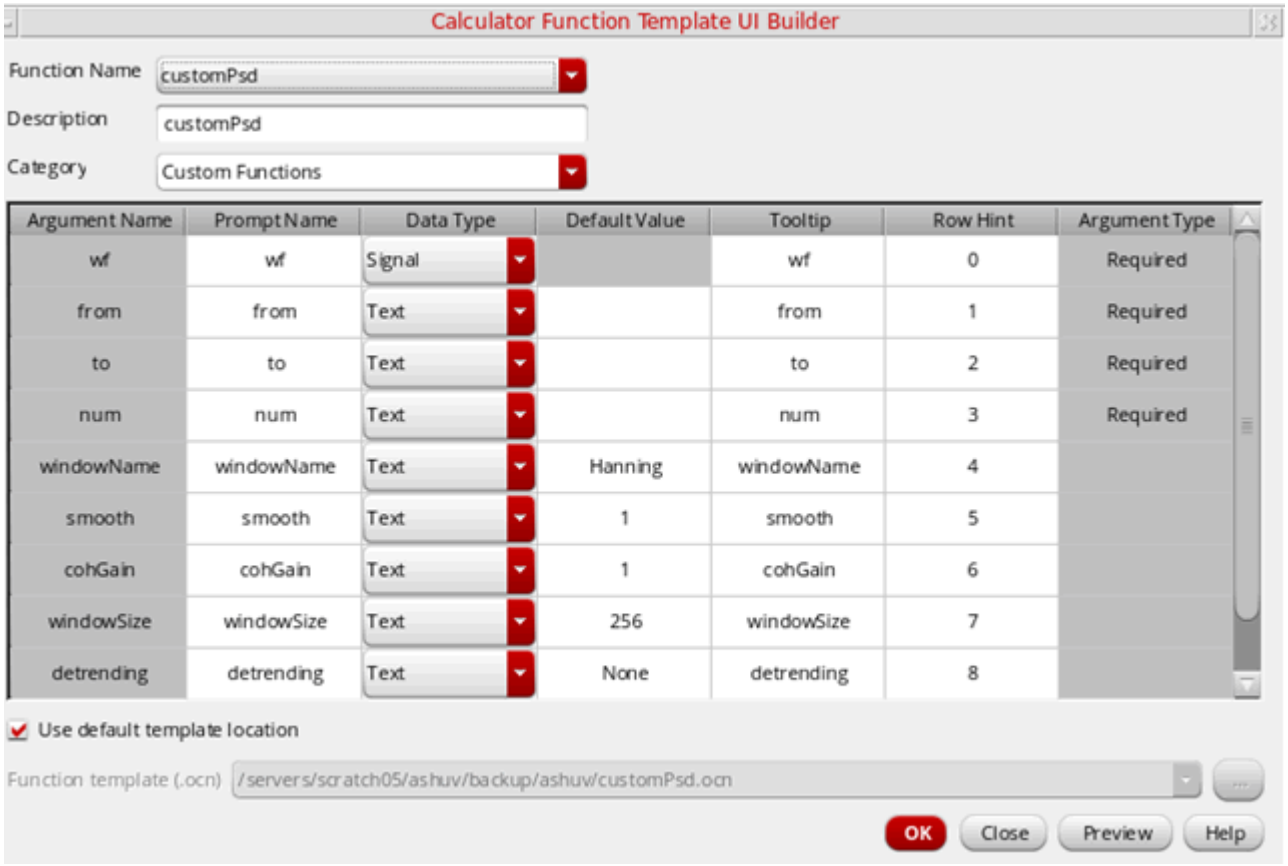
When you click *Create New* or *Edit UI* button, the *Calculator Function Template UI Builder* form appears. For more information about how to use this form, see the next section.

Using the Calculator Function Template UI Builder Form

You can use this form to define the UI template for the custom Calculator function. This form includes a list of argument values as specified in the given SKILL file. For example, for the

© 2023 Cadence Design Systems, Inc.

`customPsd` function defined in the `test.il` SKILL file selected in step1, the following form fields appear:



In this form, specify the following field values:

- *Function Name*—From this drop-down list, select the name of the function for which you want to create the UI template. It includes the names of all the functions defined in the SKILL file you selected in the *Add SKILL Functions* form.
- *Description*—Provide a brief description about the function you selected in the previous step.
- *Category*—In this drop-down list, select the category in which you want to list the function in the Function Panel. When you are defining a user-defined custom function, select `Custom Functions`. You can also add a new category in the Function Panel by typing the name of the category in this field. The function you create is displayed under this category.

  An argument table is displayed below the category fields that by default displays the argument and their attributes, such as name, type, value, and so on, which are defined in the SKILL file. You can modify these arguments according to your requirements.

© 2023 Cadence Design Systems, Inc.

- *Argument Name*—Lists the argument names specified in the SKILL file for the selected function. You cannot change the argument name values. In most of the functions, the first argument is always a waveform.
- *Prompt Name*—Lists the UI names corresponding to the argument names that you want to display in the function template. By default, prompt names are the same as the argument names; however, you can modify them and make them more informative.
- *Data Type*—Includes a drop-down list box to select the data types you want to use for the given argument. The data types can be as follows:
  - `Signal`—Represents the primary waveform or a signal.
  - `Text`—Respresents a text box.
  - `Cyclic`—Represents a drop-down list box. When you specify the data type as cyclic, you must provide a value in the *Default Value* field.
  - `Integer`—Represents an integer value.
  - `float`—Represents a floating point value.
  - `Net`—Represents a waveform or a signal.
  - `Terminal`—Represents a terminal value.
  - `Port`—Represents a port.
- *Default Value*—Lists the default value that you want to specify for each field. The value you specify in this field is displayed in the function UI. To specify the options for the cyclic data type fields, you can specify a comma-separated list of options in this field. For example, the Window Type is a cyclic field and you can specify the values in the following format: `"Hanning", "Kaiser", "Rectangular"`. The first value you specify becomes the default value.

  Note: The *Default Value* field is not available when the *Data Type* field is set to `Signal`.
- *Tooltip*—Specifies the tooltip that you want to display when you hover the pointer on the given field.
- *Row Hint*—Specifies the row number in which you want to place a given field. If you want to display two or more fields side-by-side in the same row, you can specify the same row number in this field.
- *Argument Type*—Lists the type of the argument fields that are defined in the SKILL file. Argument types can be `Required`, `Optional`, and `Keyed`. You cannot change these argument types.

© 2023 Cadence Design Systems, Inc.

Note: The argument types of the fields are the same as specified in the function definition in given SKILL file.

- *Use Default Template Location*—Select this check box if you want to save the `.ocn` file at the default location, which is the same directory in which the SKILL file is saved. If you want to specify another location to save the `.ocn` file, browse and specify the new location in the *Function template (.ocn)* field. By default, this check box is selected, which means that the UI template is saved with the SKILL file.

  Note: This check box is available only when you select the Separate UI Template File (.ocn) option.

This form includes the following buttons:

- *OK*—Apply the changes and close the form.
- *Save All*—Save the UI template settings for all the SKILL functions defined in the specified SKILL file. For example, if the SKILL file contains four functions, the UI template is created for all the four functions.

  Note: This button is displayed only when you open this form to define the UI template in the specified SKILL file.

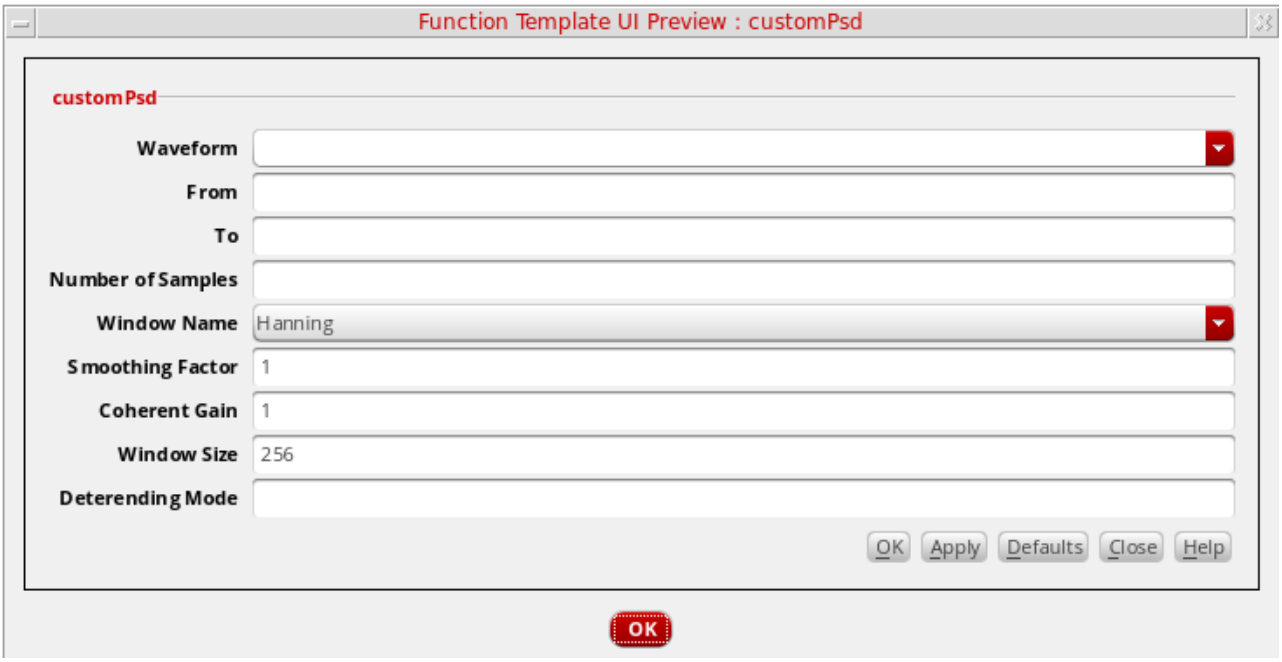
- *Save*—Save the UI template for only the function you selected in the *Function Name* field.

  Note: This button is displayed only when you open this form to define the UI template in the specified SKILL file.

- *Close*—Close the form without saving.

© 2023 Cadence Design Systems, Inc.

- *Preview*—View the structure of the new UI that you have specified in the above fields. For example, for the `customPsd` function UI defined in the above fields, the following preview appears. To close the preview, click *OK.*



- *Help*—Opens the documentation.

The specified `customPsd` function and its template is added to the Function Panel of the Calculator. To view this function, open Function Panel and select the `Custom Functions` category.

Viewing Custom Functions

When you add the custom functions, they appear in blue in the Function Panel as shown in the figure above. For example, when you click the `customPsd` function, the function UI is

© 2023 Cadence Design Systems, Inc.

opened in Function Panel with the arguments you specified in the `.ocn` file.

Now, you can use this function to create and evaluate expressions. By default, the information about custom functions is saved at the following location: `~/.cadence/dfII/viva/customCalcFunctions.ini`.

In addition, you can also use the following SKILL function to add a function template to the Function Panel of Calculator:

```
awvLoadCustomCalcFunction(?fileName filename ?funcName funcName
?templateFileName templateFileName)
```

For more information about this SKILL command, see `awvLoadCustomCalcFunction` in *Virtuoso Visualization and Analysis XL SKILL Reference*.

Sharing Custom Functions

You can also create custom functions and share them between multiple users. To do this, perform the following steps:

1. Create a central file, `central.ini`.
2. In this file, add the `awvLoadCustomCalcFunction` function to create the custom functions.
3. In the `.cdsinit` file, add the `awvLoadSharedCustomFunctionsFile` function to share and load the custom function created in the previous step.

© 2023 Cadence Design Systems, Inc.

```
awvLoadSharedCustomFunctionsFile("~/central.ini")
```

The shared custom function appears in red on Function Panel.

© 2023 Cadence Design Systems, Inc.

Working with the Calculator in ADE

While working in ADE, if you want to analyze simulation results, you can directly build expressions by opening the Virtuoso Visualization and Analysis XL Calculator from the ADE window and build expressions by using the simulation data generated during the latest run.

This section covers the following topics:

- Opening the Calculator using ADE on page 665
- Additional Features in ADE on page 671
- Working with Expressions in ADE on page 678

Opening the Calculator using ADE

When you open the Calculator from ADE, it displays some additional GUI components, such as Schematic Selection toolbar, Test toolbar. The Schematic Selection toolbar contains a set of standard functions, such as `vt` for voltage transient, that can be applied on the schematic objects, such as nets, terminals, and flip-flops, to build expressions. The Test toolbar contains the path of the simulation directories available for the current simulation run.

When you select a function on the Schematic Selection toolbar, the schematic design view opens. In the schematic design view, select a schematic object on which you want to perform this function. The complete expression for the selected function and the selected schematic object is created in the Buffer. For more information about how to select a schematic object, see Working with Expressions in ADE on page 678.

While working in the ADE environment, you can open the Calculator by using any of the following two methods:

- Opening the Calculator from ADE Explorer on page 665
- Opening the Calculator from ADE Assembler on page 668

Opening the Calculator from ADE Explorer

To open the Calculator from the ADE Explorer, open ADE Explorer and do one of the following:

➥ Choose *Tools – Calculator*

➥ In the ADE Explorer Output Setup pane:

© 2023 Cadence Design Systems, Inc.

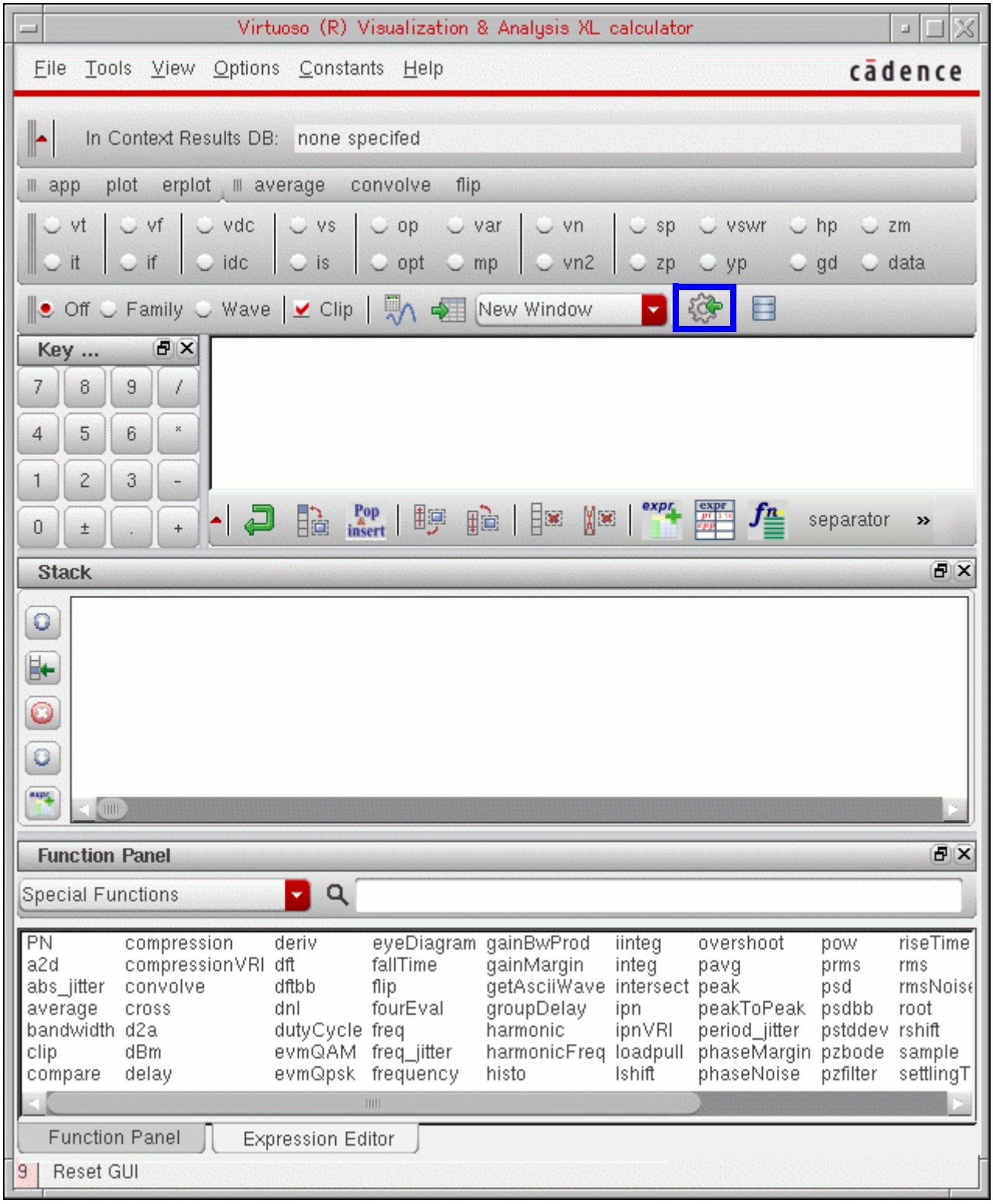
a. Click *Expression* from the drop-down list. A new row of type `expr` is added in the Outputs Setup tab.

b. In the *Details* column, either directly type an output expression or click the ellipses button to open the Calculator to add an output expression.

c. To move the expression created in the Buffer to ADE Explorer, click click the button.

The Calculator window appears. Notice the Schematic Selection toolbar and the Send to ADE button on the Selection toolbar in the figure below, which appear when the Calculator is opened from ADE Explorer.

© 2023 Cadence Design Systems, Inc.



© 2023 Cadence Design Systems, Inc.

For more information about the Schematic Selection toolbar, see "Schematic Selection Toolbar" on page 617.

For more information about Send to ADE toolbar button, see Send to ADE Button on page 678.

For information about the Calculator menu and toolbar options, see "Using the Calculator Graphical User Interface (GUI)" on page 607.

Opening the Calculator from ADE Assembler

To open the Calculator in ADE Assembler, open the ADE Assembler window and do one of the following:

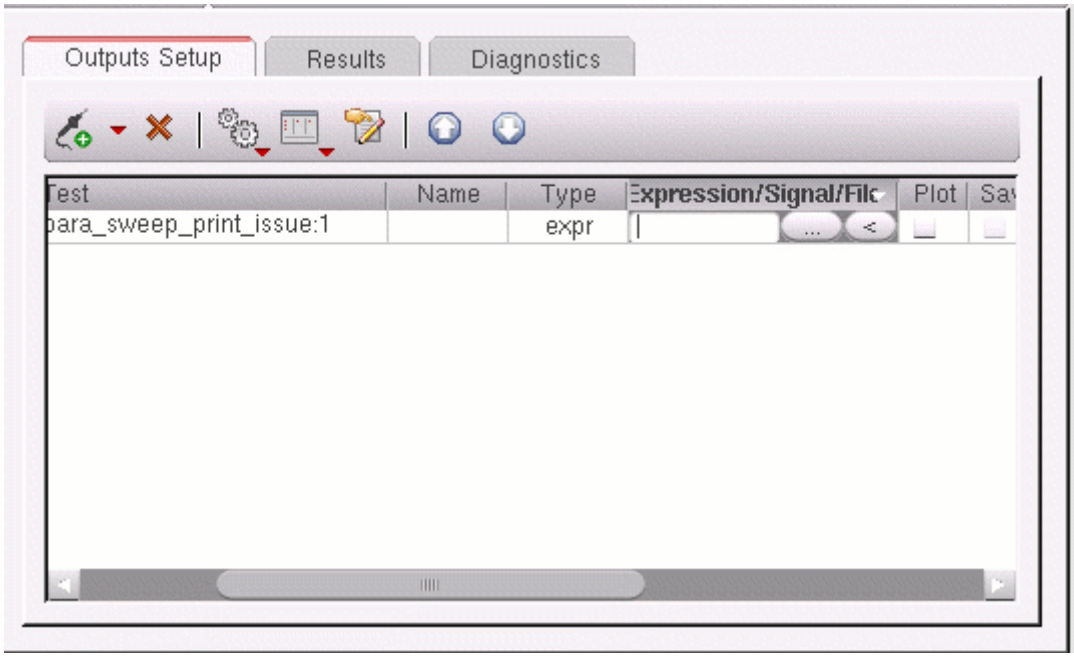
- Choose *Tools – Calculator*

  Alternatively, you can open the Calculator window by clicking the button in the ADE Assembler window.

- On the *Outputs Setup* tab:

  a. Select *Type* as *expr.*

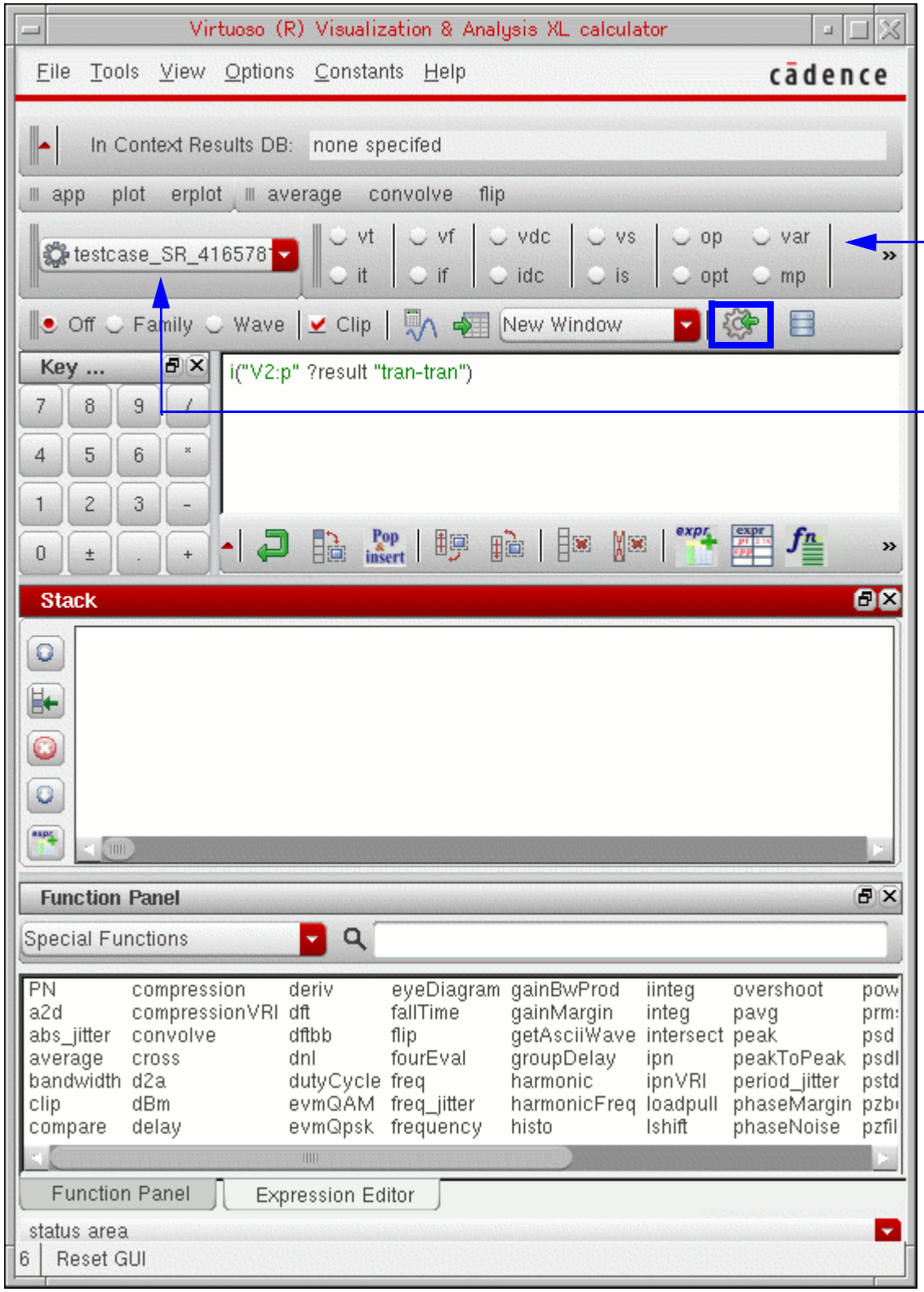
  b. Double-click anywhere in the *Expression/Signal/File* column and then click the ellipses (...) button.



© 2023 Cadence Design Systems, Inc.

The Calculator window appears. Notice the Schematic Selection toolbar and the Test directory displayed on the Result toolbar in the figure below. Also, note an additional toolbar button on the Selection toolbar that can be used to send expressions displayed in the Buffer to the ADE Assembler Outputs Setup. These appear when the Calculator is opened from ADE Assembler. For more information, see Additional Features in ADE on page 671.

© 2023 Cadence Design Systems, Inc.



© 2023 Cadence Design Systems, Inc.

After you have created the required expression in the Buffer, you can move the expression to ADE Assembler for evaluation. To move the expression to ADE XL, in the Calculator window, choose *Tools – Send Buffer to ADE Outputs*. The expression is displayed in the *Output Setup* window of ADE Assembler and is evaluated after you run the simulation.

In ADE Assembler, you can simultaneously work on multiple test directories in a session. To select a test directory, select it from the *Test* drop-down list in the Calculator window.

For more information about the Schematic Selection toolbar, see Schematic Selection Toolbar on page 671.

For more information about the Calculator menus and toolbar options, see "Using the Calculator Graphical User Interface (GUI)" on page 607.

Additional Features in ADE

When you open the Calculator in ADE, the Calculator GUI displays the following additional features:

- Schematic Selection Toolbar on page 671
- Test Toolbar on page 677
- Send to ADE Button on page 678

Schematic Selection Toolbar

This toolbar displays a set of standard functions that can be applied on the schematic objects to build expressions. When you click a function button on the toolbar, the schematic design view appears. In the schematic design view, select a schematic object on which you want to apply the selected function. When you select the schematic object, an expression is created and displayed in the Buffer.

By default, the toolbar displays the function buttons for all simulation analyses types. However, you can configure the Schematic Selection toolbar to display function buttons for the selected analyses types.

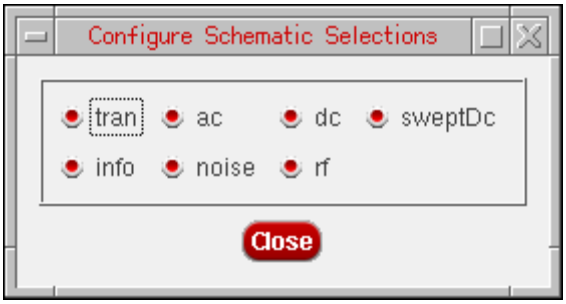
To display the function button for a particular analysis type, do one of the following:

➡ Right-click in the toolbar area and choose *Configure selections*.

© 2023 Cadence Design Systems, Inc.

➡ Choose *View – Configure Schematic Selection Toolbar*.

The *Configure Schematic Selections* form appears. Select the simulation analyses types for which you want to display function buttons on the Schematic Selection toolbar.



Note: You can click the button shown in the Result toolbar to hide the Schematic Selection toolbar.

The following table lists the available function buttons:

Button	Description
vt	Transient voltage
it	Transient current
vf	Frequency voltage
if	Frequency current
vdc	DC voltage
idc	DC terminal current
vs	Source sweep voltage
is	Source sweep current (I compared to V graphs)
op	DC operating point
os	DC operating point
ot	Transient operating point
opt	Transient operating point

© 2023 Cadence Design Systems, Inc.

var	Design variables (ADE design variables). For information about using design variables with Calculator functions, see Adding Design Variables to Expressions on page 678
mp	Model parameter
vn	Noise voltage
vn2	Noise voltage square
sp	Scattering parameters
zp	Impedance parameters
vswr	Voltage standing wave ratio
yp	Admittance parameters
hp	H-parameters
gd	Group delay
zm	Input impedance if all other ports are matched
data	Plots a previous analysis

The Schematic Selection toolbar can also be displayed using the `schematicToolbar .cdsenv` variable.

The function buttons for all analyses types are displayed by default which can be changed by setting the `schematicAnalyses .cdsenv` variable.

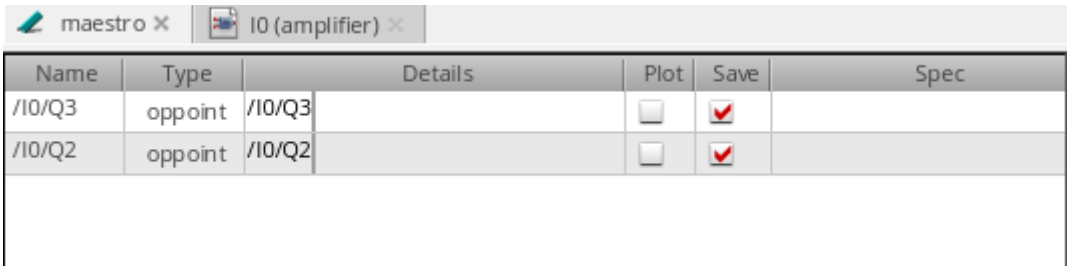
© 2023 Cadence Design Systems, Inc.

Creating Expressions For DC Operating Point Functions

You can use Calculator to create expressions for DC operating point functions, such as OS and OT, that you want to save during a simulation run in ADE Explorer. After the expressions are created, you can send them back to the Outputs Setup pane of the ADE Explorer window. These expressions are then evaluated, saved, or plotted after the simulation run is complete.

As a prerequisite step, do one of the following:

- Specify the operating points that you want to save in the *Save Operating Points* pane of the ADE Explorer window. The operating points are listed as shown in the figure below.



  For more information about how to add operating points and specify parameters to be saved for each operating point, see Saving Operating Points in *Virtuoso ADE Explorer User Guide*.

- Run a simulation in ADE Explorer without specifying the operating points.
- In the *In Context Results DB* field in Calculator, specify the results directory path that contains the simulation results.

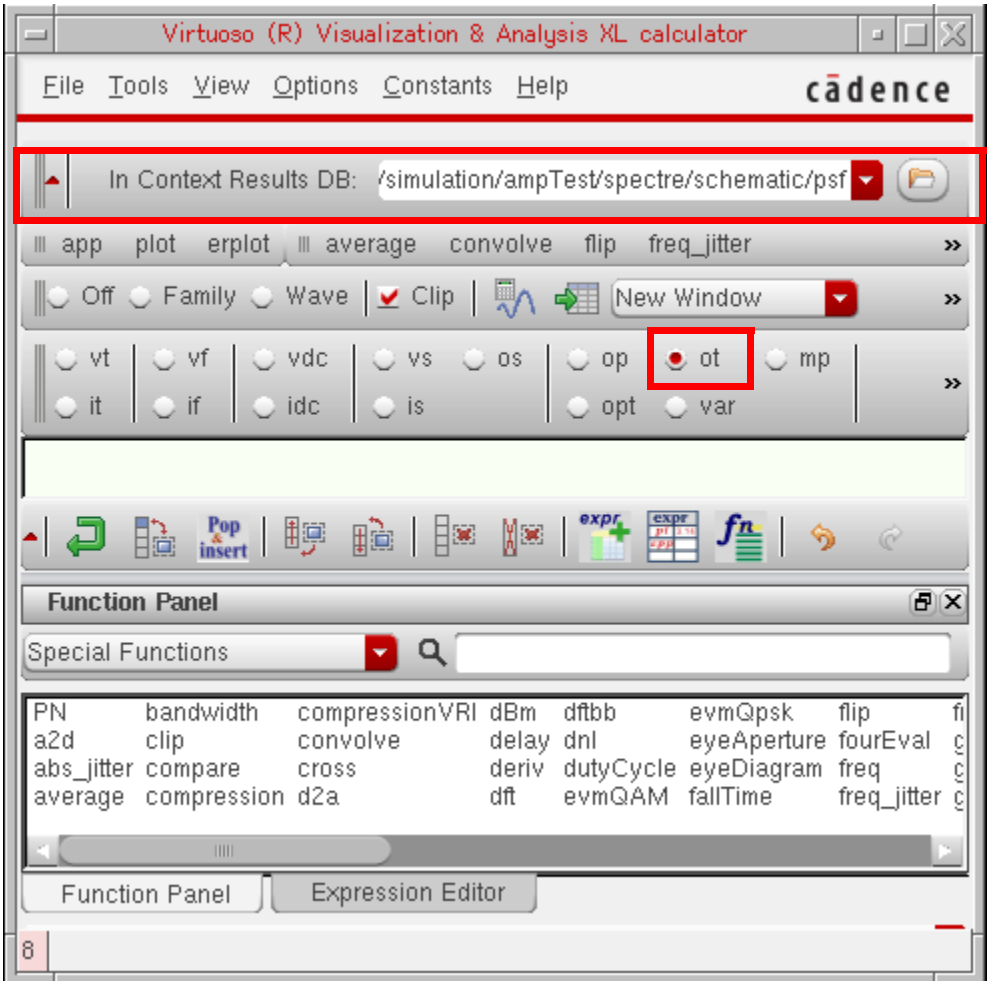
Now, perform the following steps to create expressions for DC operating point functions:

1. In the Outputs Setup tab of ADE Explorer, click *Expression* from the drop-down list. A new row of type `expr` is added in the Outputs Setup tab.

2. In the *Name* column, specify the name for the expression output.
3. In the *Details* column, either directly type an output expression or click the ellipses button to open the Calculator to add an output expression. Alternatively, choose *Tools – Calculator* to open the Calculator.

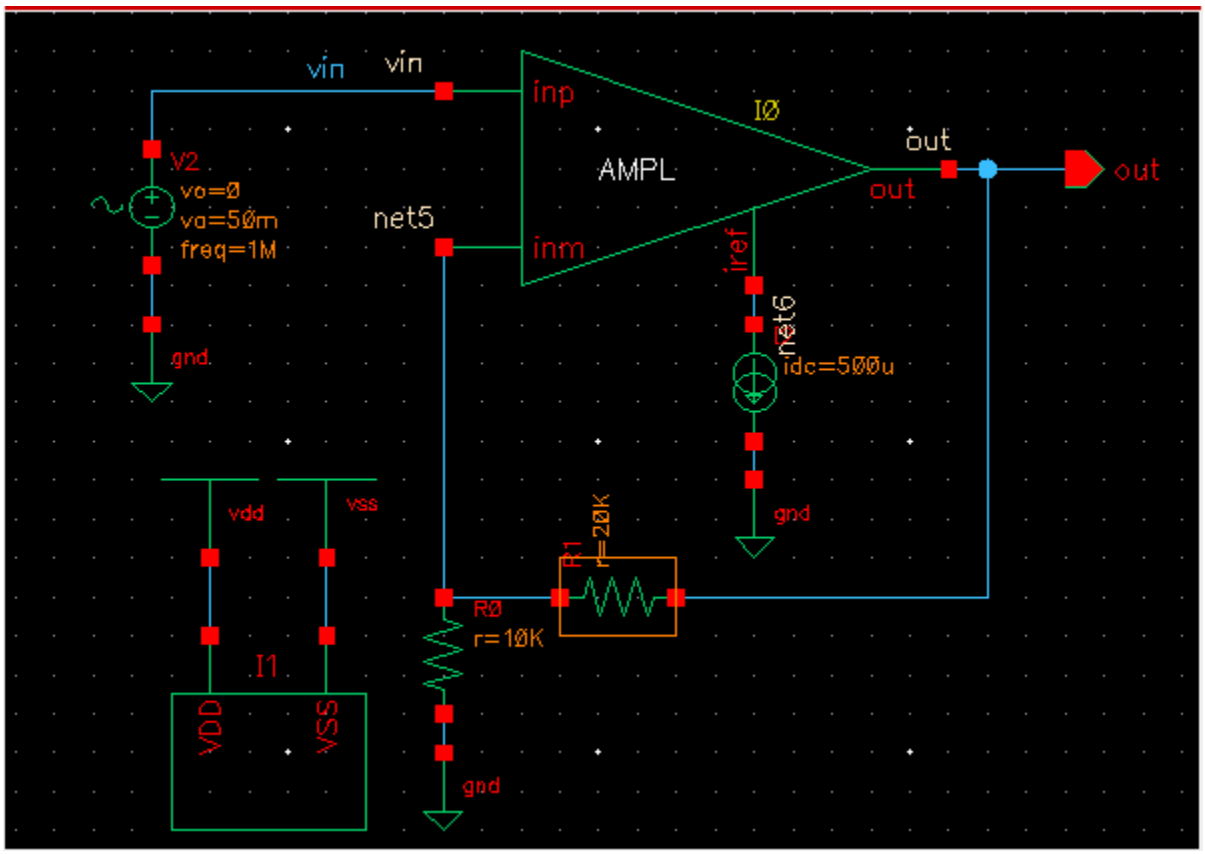
© 2023 Cadence Design Systems, Inc.

The Calculator window appears with the *In Context Results DB* field set to the results directory that has been saved in the current ADE Explorer setup.

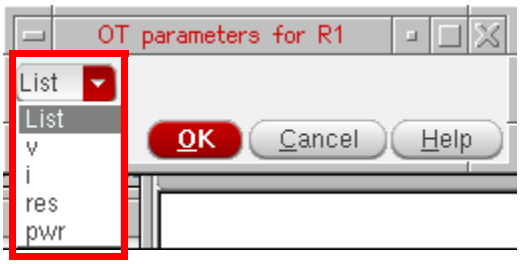


© 2023 Cadence Design Systems, Inc.

4. To create an expression for the `OT` function, select *ot* on the Schematic Selection toolbar. The design is opened in the schematic window as shown in the figure below.

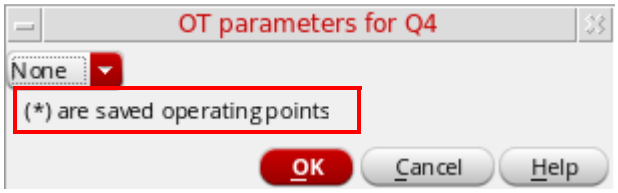


5. Select an instance. The *OT parameters for <instance_name>* form appears. This form lists all the operating point parameters in a drop-down list. If you select an instance that is not added to the Save Operating Points tab of the ADE Explorer window, the drop-down list in this form contains `None`.



© 2023 Cadence Design Systems, Inc.

If you already have some operating points saved for the OS and OT instances, they are displayed with an asterisk (*) prefix in the drop-down list (see the figure below).

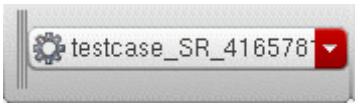


6. Select the parameter for which you want to build expressions. For example, if you select current (`i`), the following expression is created in the Buffer: `OT("/R1","i")`
7. After building an expression, do one of the following:
   - In the Calculator window, choose *Tools – Send Buffer Exploression to ADE Outputs*.
   - Click the *Send Buffer Exploression to ADE Outputs* button available on the Schematic Selection toolbar.
   - In the ADE Explorer Outputs Setup pane, click the to paste the expression from the Buffer.

   The expression appears in the *Details* field.
8. Change the *Plot* and *Save* check boxes for the expression, if required.
9. Run the simulation. The specified expressions are plotted in the graph window if the corresponding *Plot* check box is selected.

Test Toolbar

This field is displayed on the left of the Schematic Selection toolbar if you open the Calculator from ADE Asembler.



The *Test* field lists the different set of results directories available in the current session, which you can use to build expressions. The results directory in which you are working currently is displayed on the *In Context Results DB* field. If you select another test directory from the *Test* list, the *In Context results DB* field displays continues to display the results directory for which the context in ADE XL is set.

The `displayContext.cdsenv` variable controls the display of the *Test* field.

© 2023 Cadence Design Systems, Inc.

Send to ADE Button

This toolbar button is displayed on the Selection toolbar.

You can use this toolbar button to move expressions that you create in the Buffer by using the simulation data for the current run to the ADE Outputs section for further analysis.

Working with Expressions in ADE

This section describes how to select a schematic object to build an expression and how to evaluate expressions in ADE.

Selecting Schematic Objects

Before you build an expression in the Calculator, you first need to select a schematic object in the schematic design view. After you select the schematic object, the corresponding expression is created in the Buffer.

Perform the following steps to select a schematic object:

1. On the Schematic Selection toolbar, click the required schematic function button.

   The schematic design view appears.

2. Select an object in the schematic design view. The function selected in the Calculator is applied to the selected schematic object and an expression is created in the Buffer.

   For some function, such as *var*, a form with additional options appears before the object is sent to the Buffer.

Adding Design Variables to Expressions

In ADE, you can add design variables from the schematic to the expression that you build in the Calculator.

To add a design variable to the expression, perform the following steps:

1. On the Schematic Selection toolbar, select the *var* option.

   The Virtuoso Schematic Editing window opens and a form named *Select an Instance* appears.

© 2023 Cadence Design Systems, Inc.

2. Select an instance on the schematic.

   The design variables for the instance are displayed in the *Select an Instance* form.

3. Select the design variable that you want to add to your expression and click *OK*.

   The selected design variable appears in the Buffer as follows:

   ```
   VAR("CAP")
   ```

Evaluating Expressions

You can load the expressions created using the Calculator into the ADE *Outputs Setup* window. The purpose of Virtuoso Visualization and Analysis XL Calculator in ADE is to build expressions for the current simulation run.

When the Calculator is opened from ADE, the expressions are based on the schematic objects; therefore the expressions cannot be evaluated in the Buffer. Instead the expressions are evaluated in ADE immediately after the simulation is run and the results are displayed in the ADE *Outputs Setup* window. You can use the results to analyze the circuit design. For more information about how to import results to the ADE, see Opening the Calculator using ADE on page 665.

Reloading Expressions using ADE

You can reload a saved expression in the Calculator while running simulation in ADE Explorer and Assembler.

For example, if you run a simulation in ADE and create expressions in the Expression Editor, such as `E0 = VT("/vin")` and `E1 = E0 + 1`, where expression `E1` is dependent on `E0` and evaluate these expressions by clicking the *Eval* button in the Expression Editor. The output signals are plotted in a graph window.

Now, if you run the simulation again with different dataset values, the new values for `E0` and `E1` are generated in the ADE Outputs section. To update these expression values in the Virtuoso Visualization and Analysis XL graph window, choose *File – Reload*. The updated values for `E1` and `E0` are displayed in the graph window.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

6

Distributed Plot

Distributed Plot is a standalone process that accepts plotting commands. This process is attached to the main `virtuoso` process.

This standalone process is useful in resource-intensive operations such as plotting large simulation datasets, and prolonged plotting operations. The Distributed Plot process can be launched either on a separate remote machine or on the same machine running the main virtuoso process.

Except for Direct Plot, Wave Compare Specs, Create Datasheet, and Checks/Asserts, Distributed Plot mode supports all plotting operations, such as Plot All, Quick Plot All, Plot Across Corners, Plot Across Design Points, Plot Outputs – Transient, Plotting Template, and Plot Outputs – Expressions.

Features that are not supported in the Distributed Plot mode use the local `viva` process of the main `virtuoso` process.

Advantages of Distributed Plot

Distributed Plot offers the following advantages:

- Increases productivity: You can use an additional `viva` process in tandem with the default `virtuoso` process.
- Offers scalability: You can remotely launch the plotting window in a separate machine from the one that is running the main `virtuoso` process.

Video

The Distributed Plot video demonstrates how to use Distributed Plot.

Related Topics

Enabling or Disabling Distributed Plot

© 2023 Cadence Design Systems, Inc.

Opening the Distributed Plot Assistant

Opening the Distributed Plot Remotely

distributedPlotEnabled

distributedPlotEnableCIW

distributedPlotLoadCdsInit

distributedPlotJobPolicy

distributedPlotEnableNonSupportedPlotWarning

distributedPlotPolicyNameFilter

distributedPlotLingerTime

© 2023 Cadence Design Systems, Inc.

Enabling or Disabling Distributed Plot

To enable Distributed Plot:

1. Depending on the application, do one of the following:

   - From the menu bar of ADE Assembler, choose *Options — Plotting/Printing*. In the *Distributed Plot* section of the ADE Assembler Plotting/Printing Options form, select the *Enable Distributed Plot* check box.

     The icon in the toolbar of the *Results* tab in ADE Assembler indicates that the distributed plot is enabled. The icon indicates that the distributed plot is disabled.

   - From the menu bar of ADE Explorer, choose *Results — Plotting/Printing Options*. In the *Distributed Plot* section of the ADE Explorer Plotting/Printing Options form, select the *Enable Distributed Plot* check box.

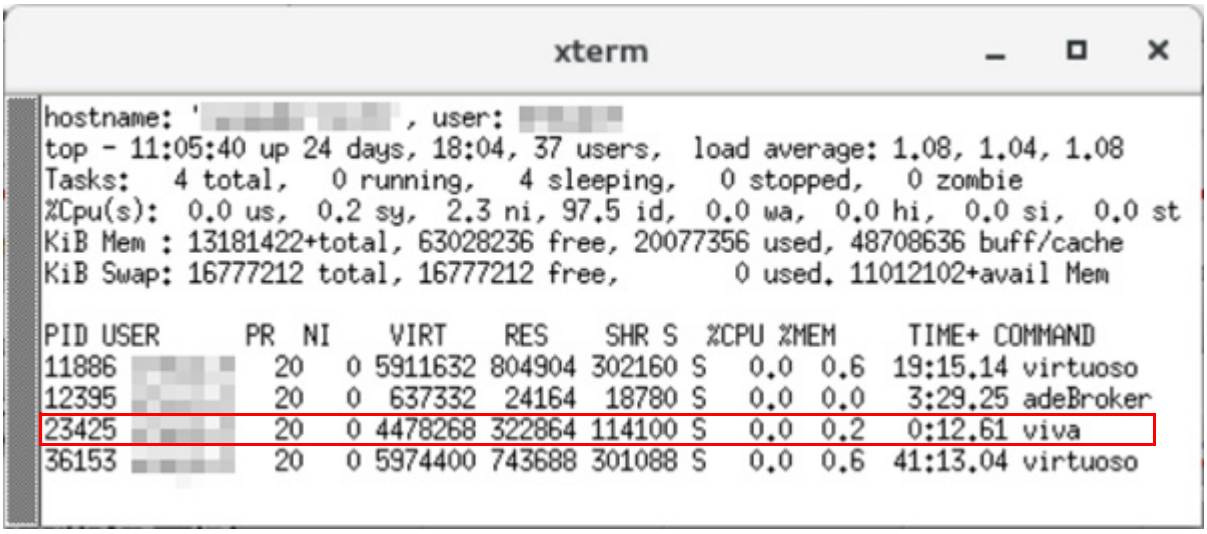
     The icon in the *Run* toolbar of ADE Explorer indicates that the distributed plot is enabled. The icon indicates that the distributed plot is disabled.

   If you plot a signal after enabling Distributed Plot, the plot operation is performed in a separate process that is independent of the main `virtuoso` process.

2. To view the list of available process:

   In the *Results* tab of ADE Assembler or in the *Run* toolbar of ADE Explorer, click the down arrow next to the icon to display additional options, and then choose *Show Process List.*

   The following figure shows that the distributed plot operation is performed in the process named `viva`, which is independent of the main `virtuoso` process.



© 2023 Cadence Design Systems, Inc.

Note: `adeBroker` is a bookkeeper process that keeps track of all processes running in the current Virtuoso session.

Related Topics

Distributed Plot

Opening the Distributed Plot Assistant

Opening the Distributed Plot Remotely

Specifying Linger Time

distributedPlotEnabled

distributedPlotEnableCIW

distributedPlotLoadCdsInit

distributedPlotJobPolicy

distributedPlotEnableNonSupportedPlotWarning

distributedPlotPolicyNameFilter

distributedPlotLingerTime

© 2023 Cadence Design Systems, Inc.

Specifying Linger Time

Linger time is the period in seconds for which the distributed plot process waits before exiting when there is no active plotting window is visible.

To specify the linger time for distributed plot process:

1. From the menu bar of ADE Assembler, choose *Options — Plotting/Printing*.

   From the menu bar of ADE Explorer, choose *Results — Plotting/Printing Options*.

2. In the *Distributed Plot* section of the ADE Assembler Plotting/Printing Options form or ADE Explorer Plotting/Printing Options form, specify the time in second in the *Linger Time* field. The default value is `60` second.

   It is recommended not to set the linger time to less than 30 second.

Related Topics

distributedPlotLingerTime

© 2023 Cadence Design Systems, Inc.

Opening the Distributed Plot Assistant

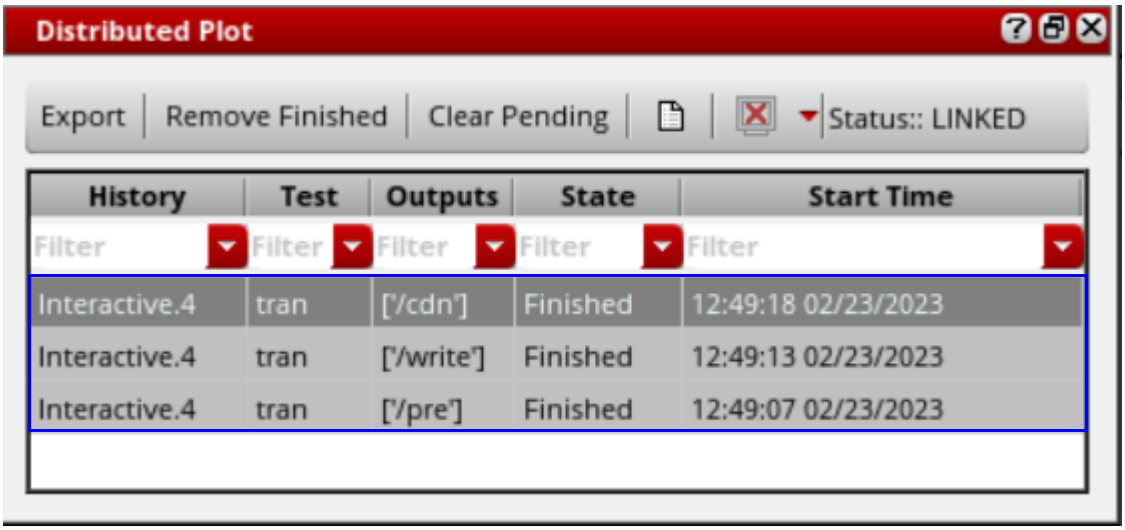
The Distributed Plot assistant lets you view and manage various plotting jobs.

To open the Distributed Plot assistant:

1. In the *Results* tab of ADE Assembler or in the *Run* toolbar of ADE Explorer, click the down arrow next to the icon to display additional options, and then choose *Show Assistant.*

   The Distributed Plot assistant opens.

   When you plot signals after enabling distributed plot, the plotting jobs are displayed in the Distributed Plot assistant, as shown in the following figure.



2. In the Distributed Plot assistant, do the following:

   - *Export*: Exports the details of jobs to a CSV file.
   - *Remove Finished*: Clears the jobs that are either in the finished or in canceled state.
   - *Clear Pending*: Clears the jobs that are in the pending state.
   - : Opens the log file for the distributed plot.
   - : Lists following options:
     - *Exit Distributed Plot Service*: Exits the distributed plot service.
     - *Kill Distributed Plot Process*: Kills the distributed plot process.

© 2023 Cadence Design Systems, Inc.

Related Topics

Distributed Plot

Enabling or Disabling Distributed Plot

Specifying Linger Time

Opening the Distributed Plot Remotely

distributedPlotEnabled

distributedPlotEnableCIW

distributedPlotLoadCdsInit

distributedPlotJobPolicy

distributedPlotEnableNonSupportedPlotWarning

distributedPlotPolicyNameFilter

distributedPlotLingerTime

© 2023 Cadence Design Systems, Inc.

Opening the Distributed Plot Remotely

To remotely open the Distributed Plot window in a separate machine:

1. Open the Job Policy Setup form:
   - From the menu bar of ADE Assembler, choose *Options — Job Setup*.
   - From the menu bar of ADE Explorer, choose *Setup — Job Setup*.

   The Job Policy Setup form opens.
2. Click the *Simulation* tab.
3. In the *Job Policy Name* field, specify the name of the job policy. For example, `distributedPlotJob`.
4. Click *Save*.

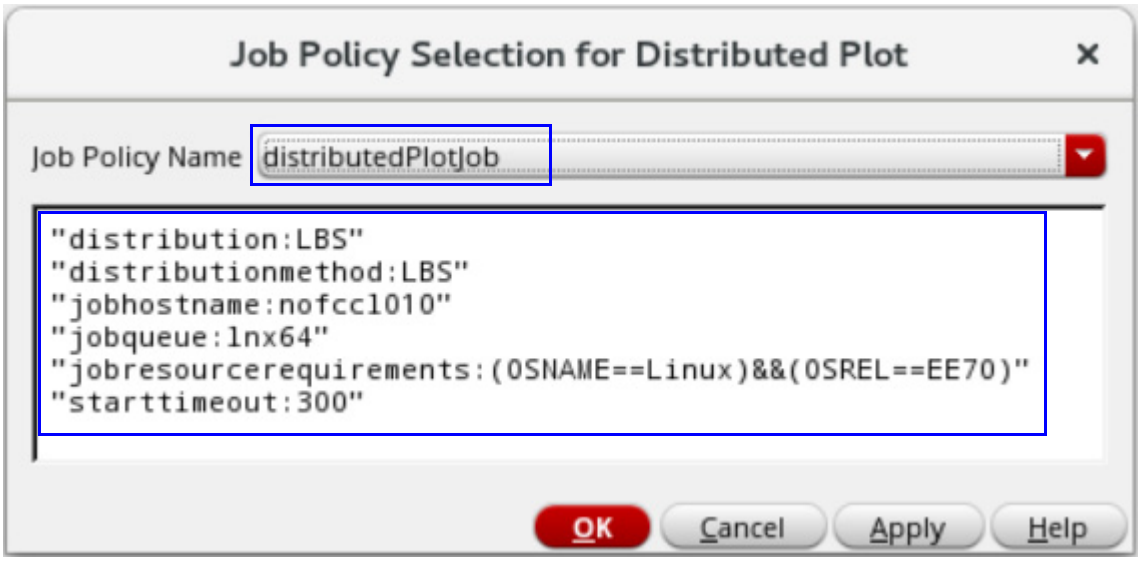
   The Save Job Policy form opens.
5. Select the directory where you want to save the job policy in the *Select Path* list and click *OK* to close the Save Job Policy form.
6. In the *Setup* section, from the *Distribution* Method drop-down list, select a job distribution method. For example, `LBS`.
7. If you select *LBS* as the job distribution method, specify the following options:
   a. Select the *Queue* check box and select an available queue from the drop-down list.
   b. Select the *Host* check box and select an available host from the list area.
   c. In the *Resource Requirement* field, specify any additional resource requirements to submit the job. For example:
      ```
      (OSNAME=Linux)&&(OSREL=EE70)
      ```
   d. Leave other options in the *Setup* section blank.
8. In the *Start Timeout* field in *Timeout* section, specify the time (in seconds) for which the tool should wait to remotely launch the distributed plot before quitting.
9. Click *OK* to close the Job Policy Setup form.
10. In the *Results* tab of ADE Assembler or in the *Run* toolbar of ADE Explorer, click the down arrow next to the icon to display additional options.
11. From the list of options, choose *Job Distribution*.

    The Job Policy Selection for Distributed Plot form opens.

© 2023 Cadence Design Systems, Inc.

12. From the *Job Policy Name* drop-down list, select the job policy that you have set up earlier. For example, `distributedPlotJob`.

    The information about the selected job policy is displayed in the box, as shown in the following figure.



13. Click *OK*.

14. Right-click a result and choose *Plot*.

    The selected result is plotted in a separate machine specified as the host machine in the Job Policy Setup form.

Related Topics

Enabling or Disabling Distributed Plot

Opening the Distributed Plot Assistant

Specifying Linger Time

Specifying an LBS Job Policy

distributedPlotEnabled

distributedPlotEnableCIW

distributedPlotLoadCdsInit

© 2023 Cadence Design Systems, Inc.

distributedPlotJobPolicy

distributedPlotEnableNonSupportedPlotWarning

distributedPlotPolicyNameFilter

distributedPlotLingerTime

© 2023 Cadence Design Systems, Inc.

7

Virtuoso Visualization and Analysis XL Forms

This section describes the forms that can be invoked from the Results Browser, Virtuoso Visualization and Analysis XL Table, and Virtuoso Visualization and Analysis XL Calculator.

- Apply Range Form
- Create Bus Form
- Create Graph Marker Form
- Digital Bus Trace Properties Form
- The Edit Marker Label form
- The Export Waveforms Form
- The Format Attributes Form
- The Graph Properties Form (Circular Graphs)
- The Horizontal Marker Properties Form
- The Point Marker Properties Form
- The Point Marker Properties for Circular Graph Form
- The Polar Reference Point Marker Properties Form
- The Preferences Form
- The Print Form
- The Printer Properties Form
- The Reference Point Marker Properties Form
- The Reveal Columns Form
- The Sample Values Form

© 2023 Cadence Design Systems, Inc.

- The Save As CSV Form
- The Save Image Form
- The Select Color Form
- The Select Waveform Database Form
- The Set Sweep Ranges Form
- The Smith Reference Point Marker Properties Form
- The Swap Sweep Var Form
- The Value At Form
- The Vertical Marker Properties Form
- The Waveforms to be Exported Form
- The Window Properties Form
- The Y vs Y Form
- The Zoom To Form

Related Topics

Virtuoso Visualization and Analysis XL Table

Results Browser

© 2023 Cadence Design Systems, Inc.

Apply Range Form

The Apply Range form lets you specify the range of x-axis values corresponding to which you want to view y-axis values of the waveform.

Field	Description
<= value between <=	Specifies the range of x-axis values corresponding to which you want to view y-axis values. The default start and end values of the range depends on waveform data.

Related Topics

Filtering Table Data

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

Create Bus Form

The Create Bus form lets you create a bus from the selected digital signals.

Field	Description
Signal/Expr Names	Displays a list of signals and expressions from which a bus is to be created.
Radix	Specify one of the following radix types for the bus: ■ *Binary* ■ *Octal* ■ *Ascii* ■ *Hex* ■ *Signed Decimal* ■ *Unsigned Decimal*
Bus Name	Specifies the name of the bus.
Plot Mode	Specifies one of the following plotting modes for the bus: ■ *Append*: Adds the bus to the current subwindow. ■ *Replace*: Replaces the signals in the current subwindow with the bus. ■ *New Subwindow*: Plots the bus in a new subwindow within the same graph window. ■ *New Window*: Plots the bus in a new graph window.

© 2023 Cadence Design Systems, Inc.

Create Graph Marker Form

The Create Graph Marker form lets you create point, fixed-y, horizontal, and vertical markers on a rectangular graph.

The form contains the following tabs.

Tab	Description
Point Tab	Lets you specify the fields you need to specify when creating a point marker.
FixedY Tab	Lets you specify the fields you need to specify when creating a fixed-y marker.
Horizontal Tab	Lets you specify the fields you need to specify when creating a horizontal marker.
Vertical Tab	Lets you specify the fields you need to specify when creating a vertical marker.
Edge Tab	Lets you specify the fields you need to specify when creating an edge marker.

Point Tab

The following table describes the fields that are available on the *Point* tab of the Create Graph Marker form. You can specify these fields to place a point marker on the selected trace.

Field	Description
Label	Specifies the format specifiers whose corresponding information you want to display in the label of the point marker. The default value is `%M: %X %Y`, which indicates that *`marker name`*`: x-axis value y-axis value` is displayed in the label of the point marker. You can also click *Edit* to open the Editing Marker Label form to specify label format for the point marker.
Expression	Specifies the expression you want to display in the marker label. For example: `average(v("out" ?result "tran-tran") )`

© 2023 Cadence Design Systems, Inc.

Field	Description
Trace	Lets you select the *Attach to Trace* check box to attach the marker to the closest interpolated point on the trace. If there are multiple traces in the graph, select the trace to which you want to attach the marker from the drop-down list.
Position	Specifies the x- and y-axes values where you want to place the point marker on the trace.

FixedY Tab

The following table describes the fields that are available on the *FixedY* tab of the Create Graph Marker form.

Field	Description
Label	Specifies the format specifiers whose corresponding information you want to display in the label of the fixed-y marker. The default value is `%M: %X %Y`, which indicates that *`marker name`*`: `*`x-axis value y-axis value`* is displayed in the label of the point marker. You can also click *Edit* to open the Editing Marker Label form to specify label format for the point marker.
Expression	Specifies the expression whose evaluated value you want to display in the marker label. For example: `average(v("out" ?result "tran-tran") )` If there is an error in evaluating the expression, `Expr = <error>` is displayed in the marker label.
Trace	Lets you select the trace to which you want to attach the marker.
Options	Lets you select the *Show Marker Level when Selected* check box to display a y-level indicator when the fixed-y marker is selected. You can drag this y-level indicator to drag the marker vertically, which makes it easy to change the y position of the marker.
Y Position	Specifies the y-axis value where you want to place the fixed-y marker.

© 2023 Cadence Design Systems, Inc.

Horizontal Tab

The following table describes the fields that are available on the *Horizontal* tab of the Create Graph Marker form.

Field	Description
Y Positions	Specifies the y-axis value where you want to create the horizontal marker. The following options in the drop-down list displayed nest to the Y Positions field let you specify whether you want to display the intercept labels. ■ *off*: The intercepts by the horizontal marker are not displayed on the trace. ■ *on when hover*: The intercepts by the horizontal marker are displayed on the trace only when hovering over the horizontal marker. This is the default value. ■ *on*: The intercepts by the horizontal marker are displayed on the trace.

© 2023 Cadence Design Systems, Inc.

Vertical Tab

The following table describes the fields that are available on the *Vertical* tab of the Create Graph Marker form.

Field	Description
X Positions	Specifies the x-axis value where you want to create the vertical marker. The following options in the drop-down list displayed nest to the *X Positions* field let you specify whether you want to display the intercept labels. ■ *off*: The intercepts by the vertical marker are not displayed on the trace. ■ *on when hover*: The intercepts by the vertical marker are displayed on the trace only when hovering over the horizontal marker. This is the default value. ■ *on*: The intercepts by the vertical marker are displayed on the trace.

Edge Tab

The following table describes the fields that are available on the *Edge* tab of the Create Graph Marker form.

Field	Description
Edge	Specifies the edge where you want to create the edge marker.
Trace	Lets you select the trace to which you want to attach the edge marker.

Related Topics

Adding a Fixed-Y Marker

© 2023 Cadence Design Systems, Inc.

Digital Bus Trace Properties Form

The Digital Bus Trace Properties for `busName` form lets you specify the properties of the bus.

Field	Description
Name	Displays a list of signals and expressions from which a bus is to be created.
Style	Specifies the line style and line thickness of the bus. Available line styles are as follows: *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. Available line thickness are as follows: *Fine*, *Medium*, *Thick*, *ExtraThick*, and *ExtraFine*.
Color	Specifies the foreground color of the bus.
Radix	Specifies one of the following plotting modes for the bus: ■ *binary* ■ *octal* ■ *hex* ■ *sdecimal* ■ *udecimal* ■ *ascii*
Value Font	Click *Default* to open the Define Font form where you can specify the font, font style, and font size for bus values. You can also choose to apply *Strikeout* or *Underline* effects on the selected font.

© 2023 Cadence Design Systems, Inc.

The Edit Marker Label form

The Edit Marker Label form lets you edit the information that you want to display in marker labels.

Field	Description
Label Format	Specifies the format specifiers whose corresponding information you want to display in the marker label. For example, if you want to display the marker name in the marker label, type `%M`. Alternatively, you can select or deselect the check boxes next to the format specifiers that you want to add to or remove from the marker label.
Specifier	Lets you select or deselect the format specifiers from the list. Format specifiers displayed in the list may vary depending on the type of graph and marker.
Reset	Resets the Label Format field to the last saved value.
Include Corner info	Specifies whether to print the corner information in the header of the CSV file. The *Include Corner info* check box appears dimmed and becomes available only if CSV is selected from the Files of type drop-down list.
Only Save Subset of Data	Provides options to export only the required part of the waveform or signal.
X Range Start	Start point of the x range for which you want to export the waveform
End	End point of the x range for which you want to export the waveform
Resample with Step Size	Specifies the step size to resample the exported data based on linear interpolation. The greater the step size, the fewer data points in the exported data.
Use Log Interpolation	Specifies whether to use the logarithmic scale for interpolation.
When the range boundary does not match a data point	Provides options to specify how waveform data is exported when the specified start and end points of x range do not match an actual data point.

© 2023 Cadence Design Systems, Inc.

Field	Description
Include first data point outside the range	Includes the first data point that immediately falls before the specified X Range Start value and the last data point that falls immediately after the specified X Range End value when there are no actual data points corresponding to the specified X Range Start and End values.
Only include data points inside the range	Includes only actual data points inside the specified x range.
Add new point on the range boundary by interpolation	Includes the interpolated values for the specified range boundary (X Range Start and End values) when there are no actual data points corresponding to the specified X Range Start and End values.Probability values closer to 100% will require more simulations before the yield estimate can be determined to be lower or higher than the target. Smaller probability values require fewer simulations before autostop is triggered.

© 2023 Cadence Design Systems, Inc.

The Export Waveforms Form

The Export Waveforms form lets you specify options to export waveforms.

Field	Description
File name	Name of the file to which you want to export the selected waveforms.
Files of type	Specifies one of the following formats in which you want to export the waveform: ■ *VCSV* ■ *Matlab* ■ *PSF* ■ *SPECTRE* ■ *SST2* ■ *CSV* ■ *PSFXL*
Use names from graph	Specifies whether to use the trace names that are displayed on the graph. This check box is disabled when you export a signal from the Results Browser.
Include Corner info	Specifies whether to print the corner information in the header of the CSV file. The *Include Corner info* check box appears dimmed and becomes available only if CSV is selected from the Files of type drop-down list.
Only Save Subset of Data	Provides options to export only the required part of the waveform or signal.
X Range Start	Start point of the x range for which you want to export the waveform
End	End point of the x range for which you want to export the waveform

© 2023 Cadence Design Systems, Inc.

Field	Description
Resample with Step Size	Specifies the step size to resample the exported data based on linear interpolation. The greater the step size, the fewer data points in the exported data.
Use Log Interpolation	Specifies whether to use the logarithmic scale for interpolation.
When the range boundary does not match a data point	Provides options to specify how waveform data is exported when the specified start and end points of x range do not match an actual data point.
Include first data point outside the range	Includes the first data point that immediately falls before the specified X Range Start value and the last data point that falls immediately after the specified X Range End value when there are no actual data points corresponding to the specified X Range Start and End values.
Only include data points inside the range	Includes only actual data points inside the specified x range.
Add new point on the range boundary by interpolation	Includes the interpolated values for the specified range boundary (X Range Start and End values) when there are no actual data points corresponding to the specified X Range Start and End values.Probability values closer to 100% will require more simulations before the yield estimate can be determined to be lower or higher than the target. Smaller probability values require fewer simulations before autostop is triggered.

Related Topics

Exporting Waveforms from a Single Results Directory

Exporting Contents of the Table

© 2023 Cadence Design Systems, Inc.

The Format Attributes Form

The Format Attributes form lets you change the format attributes of cells and headers of the Virtuoso Visualization and Analysis XL Table.

Field	Description
Active Format Location	Specifies whether to change the format attributes of cells or headers of the table. Possible values are: ■ *Cell*: Changes the format attributes only for the cells of the Table. This is the default value. ■ *Header*: Changes the format attributes for both cells and headers of the Table.
Scale Format	Specifies the notation for the values in the cells and headers of the Table. Possible values are: ■ *Scientific*: Changes the notation of values to scientific. For example, `1.234E-8`. This option is available only when the *Active Format Location* is set to *Cell*. ■ *Engineering*: Changes the notation of values to engineering. For example, `12.34E-9`. ■ *Suffix*: Changes the notation of values to suffix. For example: `12.34n`.
Scale Factor	Specifies the units to which you want to convert values in the Table. The units displayed in the *Scale Factor* drop-down list depends on the selected scale format. For example, if the *Scale Format* is set to: ■ *Suffix*, units are: *T*, *G*, *M*, *k*, *m*, *u*, *n*, *p*, *f*, *a*. ■ *Engineering*, unit are: *10E12*, *10E9*, *10E6*, *10E3*, *10E-3*, *10E-6*, *10E-9*, *10E-12*, *10E-15*, *10E-18*. This option is available only when the *Active Format Location* is set to *Header*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Significant Digits	Specifies the number of significant digits. The value `1.234e-3` contains `4` significant digits. The value `1.2e-3` contains `2` significant digits.

Related Topics

Formatting Attributes of the Table

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

The Graph Properties Form (Circular Graphs)

The Graph Properties Form for a circular graph lets you specify various graphs and grid properties. You can also use this form to annotate VSWR (Voltage Standing Wave Ratio) circles and Q contours.

The form contains the following tabs.

Tab	Description
The General Tab	Lets you specify graph properties of a circular graph.
The Grid Tab	Lets you specify grid properties of a circular graph.
The Annotations Tab	Lets you display VSWR circles and Q contours in a circular graph.
The Markers Tab	Lets you display the tracking information for the points that you click or point to on a circular graph.

The General Tab

The following table describes the fields that are available on the *General* tab of the Graph Properties form.

Field	Description
Title	This section provides options to specify the title and font of a circular graph.
Graph Title	Displays the title of the circular graph at the top-left corner of the graph window or subwindow. The default graph title includes the name of the analysis and the frequency range. You must deselect the *Default* check box to edit the graph title.
Font	Specifies the font properties, such as style, size, and effects, for the graph title.
Graph	This section provides options to specify the properties of a circular graph.

© 2023 Cadence Design Systems, Inc.

Field	Description
Notation	Specifies the notation to be used in the graph labels. Possible values are: ■ *Engineering*: Uses engineering notation. For example, `29E6 Hz`. ■ *Scientific*: Uses scientific notation. For example, `2.9E7 Hz`. ■ *Suffix*: (Default) Uses suffix notation. For example, `29M Hz`.
Legend Position	Specifies the position of the trace legend in the circular graph. Possible values are *left*, *inside*, and *above*. Default value is *left*.
Legend Font	Specifies the font properties, such as style, size, and effects, of the trace legend.
Reload Using Current Context	Specifies whether to reload signals in the graph according to the data from the current in-context results directory. This check box is selected by default. If this check box is not selected, signals are reloaded based on their individual databases.
Frequency Filter	This section provides options to filter graph data by frequency.
Enable	Specifies whether to enable the frequency filter. The frequency filter is disabled by default. If you select this check box, a frequency filter is displayed as a slider bar on top of the circular graph.
Min	Specifies the minimum frequency. This field becomes available only when you select the *Enable* check box. You can click the *Reset* command to reset the minimum frequency to its default value.
Max	Specifies the maximum frequency. This field becomes available only when you select the *Enable* check box. You can click the *Reset* command to reset the maximum frequency to its default value.

© 2023 Cadence Design Systems, Inc.

The Grid Tab

The following table describes the fields that are available on the *Grid* tab of the Graph Properties form.

Field	Description
Grid Type	Lets you select the type of grid for a circular graph: ■ *Polar*: Displays the polar grids. ■ *Impedance*: Displays the impedance Smith grid (Z Smith grid). ■ *Admittance*: Displays the admittance Smith grid (Y Smith grid). ■ *Immittance*: Displays the immittance Smith chart (ZY Smith grid).
Circles	This section provides options to specify how you want to draw circles on the polar grid. This section is available only for polar grids.
Number of Circles	Specifies the number of circles you want to draw on the polar grid. The default value is `0`, which indicates that the number of the circles displayed depends upon the plotted data.
Display Circle Grids	Specifies whether to display circular grids. This check box is selected by default.
Display Circle Grid Labels	Specifies whether to display labels for the circular grids. This check box is selected by default.
Display Unit Circle	Specifies whether to display a unit circle, which means a circle with radius `1`. This check box is selected by default.
Radials	This section provides options to specify how you want to draw radials on the polar grid. This section is available only for polar grids.
Number of Radials	Specifies the number of radials you want to draw in each quarter of the polar grid. The default value is `2`.
Display Radials	Specifies whether to display radials. This check box is selected by default.

© 2023 Cadence Design Systems, Inc.

Field	Description
Display Radial Labels	Specifies whether to display labels for the radials. This check box is selected by default.
Display	This section provides options to specify how you want to display circular grids. This section is not available for polar grids.
Coefficient Ring	Specifies whether to display the angles of reflection coefficient in degrees.
Compressed Smith	Specifies whether to display extra horizontal grid lines (arcs) outside the Smith chart boundary.
Detailed Grid	Specifies when to display detailed grid lines. Possible values: ■ *off*: Detailed grid lines are always hidden. ■ *on*: Detailed grid lines are always displayed. ■ *auto*: Detailed grid lines are automatically displayed when zooming in on the Smith chart. This option is not available if the *Compressed Smith* check box is selected.
Grid Labels	Specifies whether to display the grid labels. This check box is selected by default.
Labels	This section provides options to specify properties of grid labels. This section is not available for polar grids.
Imag. Circles	Specifies whether to display labels on the imaginary circles.
Font	Specifies the font properties, such as size, style, and effects, of the grid labels.
Text Size	Specifies the text size for the grid labels. Possible values: ■ *Auto*: (Default) Automatically adjusts the text size for the grid labels. ■ *Fixed*: Uses a fixed text size for the grid labels.

© 2023 Cadence Design Systems, Inc.

Field	Description
Grid Lines	This section provides options to specify properties for the standard and detailed grid lines in the impedance, admittance, and immittance charts. This section is not available for polar grids.
Standard	Lets you select the color, width, and style of the standard grid lines for the impedance, admittance, and immittance charts.
Detailed	Lets you select the color, width, and style of the detailed grid lines for the impedance, admittance, and immittance charts.

The Annotations Tab

The following table describes the fields that are available on the *Annotations* tab of the Graph Properties form.

This tab is not applicable for polar grids.

Field	Description
Smith Chart	This section provides options to display VSWR (Voltage Standing Wave Ratio) circles and Q contours on the Impedance, Admittance, and Immittance Smith charts.
VSWR Circles	This section lets you display VSWR circles for the specified VSWR values. You can also specify options to annotate VSWR circles.
VSWR	Specifies the VSWR value for which you want to draw the VSWR circle. Click the '*+*' icon to specify the VSWR value. You can specify multiple VSWR values.
Color	Specifies the color of the VSWR circles. The default color is `#c0c000` (La Rioja).
Width	Specifies the thickness of the VSWR circles. Possible values are: *Fine*, *Medium*, *Thick*, *ExtraThick*, and *ExtraFine*. The default value is *Fine*.
Style	Specifies the style of the VSWR circles. Possible values are: *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. The default value is *Solid*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Q Contours	This section lets you display Q contours for the specified Q values. You can also specify options to annotate Q contours.
Q	Specifies the Q value for which you want to draw the Q contours. Click the '*+*' icon to specify the Q value. You can specify multiple Q values.
Color	Specifies the color of the Q contours. The default color is `##c000c0`(Deep Magenta).
Width	Specifies the thickness of the Q contours. Possible values are: *Fine*, *Medium*, *Thick*, *ExtraThick*, and *ExtraFine*. The default value is *Fine*.
Style	Specifies the style of the Q contours. Possible values are: *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. The default value is *Solid*.

The Markers Tab

The following table describes the fields that are available on the *Markers* tab of the Graph Properties form.

Field	Description
Labels	This section provides an option to specify the text format of the tracking marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Tracking Format (Trace)	Specifies the text format to be displayed in the label of a tracking marker. If you leave this field blank, the following default values are used depending upon the selected grid type: ■ Polar grid: `%F%n%C` Frequency and real and imaginary Cartesian values are displayed in the label text of a tracking marker. `%n` indicates that values are displayed in a new line. ■ Impedance (Z Smith) grid: `%F%n%Z` Frequency and impedance values, such as resistance and reactance are displayed in the label text of a tracking marker. ■ Admittance (Y Smith) grid: `%F%n%A` Frequency and admittance values, such as conductance and susceptance are displayed in the label text of a tracking marker. ■ Immittance (ZY Smith) grid: `%F%n%Z%n%A` Frequency, impedance, and admittance values are displayed in the label text of a tracking marker.
Impedance/ Admittance Values	This section provides options to specify graph properties of a circular graph.
Normalize	Specifies whether to normalize the Smith reference values. When you select the *Normalize* check box, the Smith reference values are multiplied with the impedance value specified in the *Characteristic Impedance* field
Characteristic Impedance	Specifies the characteristic impedance value. The default value is `50` ohms.

© 2023 Cadence Design Systems, Inc.

The Horizontal Marker Properties Form

The Horizontal Marker Properties form lets you edit the properties of a horizontal marker.

Field	Description
Name	Specifies the name of the horizontal marker.
Y Position	Specify the y-axis coordinate where you want to place the horizontal marker.
Label	Lets you specify whether you want to display a label where the horizontal marker intercepts the y axis. ■ *Follow Intercepts*: Label is displayed based on the following options specified in the *Intercepts* field: ❑ *off*: Intercept label on the y axis is not displayed ❑ *on when hover*: Intercept label on the y axis is displayed only on hovering over the horizontal marker. ❑ *on*: Intercept label on the y axis is displayed ■ *Off*: Intercept label on the y axis is not displayed, irrespective of the options specified in the *Intercepts* field. ■ *On*: Intercept label on the y axis is displayed, irrespective of the options specified in the *Intercepts* field. The default value is *Follow Intercepts*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Intercepts	Lets you specify the following options: *Maximum*: Specifies the maximum number of intercepts displayed by the horizontal marker on the trace. For example, if you specify `2` in this field, only the first two intercepts by the horizontal marker are displayed on the trace. The default value is `20`. You can specify any value between `0`–`200`. The following options lets you specify when you want to display the intercept labels. ■ *off*: The intercepts by the horizontal marker are not displayed on the trace. ■ *on when hover*: The intercepts by the horizontal marker are displayed on the trace only when hovering over the horizontal marker. This is the default value. ■ *on*: The intercepts by the horizontal marker are displayed on the trace. The following options lets you specify the notation format for horizontal label on the y axis and intercepts labels on the trace: *Scientific*: To change the notation of values to scientific. For example: `1.234E-8`. *Engineering*: To change the notation of values to engineering. For example:`12.34E-9`. *Suffix*: To change the notation of values to suffix. For example: `12.34n`. This is the default value.
Font/Color	Specifies the style, size, and effect of the selected font for the horizontal label, intercepts labels, and marker name. Click *Foreground* to select the color for the horizontal marker and its name.
LineStyle	Specifies one of the following line styles for the horizontal marker: *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. The default value is *Dash*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Next/Prev SnapPoint	Specifies one of the following snapping criteria based on which the selected horizontal marker is snapped when you click the *Next POI* or *Previous POI* command on the *Snap* toolbar: ■ *Local Maxima*: Snaps the marker to a local maxima. Local maxima is the transition point when there is a change in the slope from the rising to falling edge starting from the marker's current position. The local maxima is calculated as the change in slope from rising to falling edge starting from the current marker position and the transition point is known as the local maxima. ■ *Local Minima*: Snaps the marker to a local minima. Local minima is the transition point when there is a change in the slope from the falling to rising edge starting from the marker's current position. ■ *Local Max or Min*: Snaps the marker to either a local maxima or a local minima. ■ *Specific Y Value*: Snaps the marker to the y-axis value specified either in *Y Position* field or in the text field on the *Snap* toolbar. ■ *Data Point*: Snaps the marker to the data points available on the trace. ■ *Global Maxima*: Snaps the marker to the maximum y-axis value on the trace. ■ *Global Minima*: Snaps the marker to the minimum y-axis value on the trace. You can also use the `N` or `P` bindkeys to snap the marker.
Significant Digits	Specifies the number of significant digits to be displayed in the intercepts by the marker on the trace. The default value is `4`. To change this value, you need to first select *Manual* from the drop-down list. For example: ■ The value `4.1442e-8` contains `5` significant digits. ■ The value `4.1e-8` contains `2` significant digits.

© 2023 Cadence Design Systems, Inc.

The Point Marker Properties Form

The Point Marker Properties form lets you edit the properties of a point marker.

Field	Description
Name	Specifies the name of the point marker.
Label	Specifies the format specifiers whose corresponding information you want to display in the marker label. For example, if you specify `%N` is this field, the name of the trace is displayed in the label of the point marker. Alternatively, you can click the *Edit* button to open the Edit Marker Label form wherein you can edit the information you want to display in the marker label.
Expression	Specify the expression whose evaluated value you want to display in the label of the point marker. For example, if you specify the expression `ln(exp(1+4)**2)` in this field, the evaluated value `10` of this expression is displayed in the marker label. $\log_e\left((e^{(1+4)})^2\right) = \log_e\left((e^5)^2\right) = \log_e(e^{10}) = 10 \times \log_e e = 10 \times 1 = 10$ If you select *Buffer Contents* from the *Expression* drop-down list, the expression in the Calculator Buffer is set in the *Expression* field.
Trace	Lets you select the trace to which you want to attach the point marker. If there are multiple traces in a graph window, select the *Attach to Trace* check box, and then select a trace from the drop-down list. Now, when you click either the *Next POI* or the *Previous POI* button on the *Snap* toolbar, the point marker is attached to the nearest interpolated data point on the selected trace. Alternatively, you can also use the `N` and `P` bindkeys to move the point marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Position	Lets you select the x- and/or y-axis coordinates of the point on the trace where you want to place the point marker. ■ *byXMode*: Places the point marker to the specified x-axis value. ■ *byYMode*: Places the point marker to the specified y-axis value. ■ *byXYMode*: Places the point marker to the specified x- and y-axis values. ■ *byFixedYMode*: Places the point marker at the specified fixed-y value. This field is specific to fixed-y markers.
Marker Level	Lets you display a y-level indicator when the fixed-y marker is selected. You can drag this y-level indicator to drag the marker vertically. This field is specific to fixed-y markers, and is available only when *byFixedYMode* is selected in the *Position* drop-down list.
Font/Color	Specifies the style, size, and effect of the selected font for the label of the point marker. Click *Foreground* to select the color of the point marker.
Notation	Specifies the notation format for the label of the point marker: ■ *Scientific*: Uses the scientific notation in the marker label. For example: `1.234E-8.` ■ *Engineering*: Uses the engineering notation in the marker label. For example:`12.34E-9.` ■ *Suffix*: Uses the suffix notation in the marker label. For example: `12.34n`. This is the default value.

© 2023 Cadence Design Systems, Inc.

Field	Description
Significant Digits	Specifies the number of significant digits to be displayed in the intercepts by the marker on the trace. The default value is `4`. To change this value, you need to first select *Manual* from the drop-down list. For example: ■ The value `4.1442e-8` contains `5` significant digits. ■ The value `4.1e-8` contains `2` significant digits.
Cross Hairs	Specifies whether you want to display cross hairs for the point marker. The cross hairs display the x- and y-axis intercepts by the point marker. ■ *Off*: Cross hairs are not displayed. ■ *On*: Cross hairs are displayed. ■ *Dynamic*: Cross hairs are displayed only when you click the point marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Next/Prev SnapPoint	Specifies one of the following snapping criteria based on which the selected point marker is snapped when you click the *Next POI* or *Previous POI* command on the *Snap* toolbar: ■ *Local Maxima*: Snaps the marker to a local maxima. Local maxima is the transition point when there is a change in the slope from the rising to falling edge starting from the marker's current position. The local maxima is calculated as the change in slope from rising to falling edge starting from the current marker position and the transition point is known as the local maxima. ■ *Local Minima*: Snaps the marker to a local minima. Local minima is the transition point when there is a change in the slope from the falling to rising edge starting from the marker's current position. ■ *Local Max or Min*: Snaps the marker to either a local maxima or a local minima. ■ *Specific Y Value*: Snaps the marker to the y-axis value specified either in *Y Position* field or in the text field on the *Snap* toolbar. ■ *Data Point*: Snaps the marker to the data points available on the trace. ■ *Global Maxima*: Snaps the marker to the maximum y-axis value on the trace. ■ *Global Minima*: Snaps the marker to the minimum y-axis value on the trace. You can also use the `N` or `P` bindkeys to snap the marker.

© 2023 Cadence Design Systems, Inc.

The Point Marker Properties for Circular Graph Form

The Point Marker Properties for Circular Graph form lets you edit the properties of a point marker for a circular graph.

Field	Description
Name	Specifies the name of the point marker.
Label	Specifies the format specifiers whose corresponding information you want to display in the marker label. For example, if you specify `%F` is this field, the frequency value displayed in the label of the point marker. Alternatively, you can click the *Edit* button to open the Edit Marker Label form wherein you can edit the information you want to display in the marker label.
Expression	Specify the expression whose evaluated value you want to display in the label of the point marker. For example, if you specify the expression `ln(exp(1+4)**2)` in this field, the evaluated value `10` of this expression is displayed in the marker label. $\log_e\left((e^{(1+4)})^2\right) = \log_e\left((e^5)^2\right) = \log_e(e^{10}) = 10 \times \log_e e = 10 \times 1 = 10$ If you select *Buffer Contents* from the *Expression* drop-down list, the expression in the Calculator Buffer is set in the *Expression* field.
Trace	Lets you select the trace to which you want to attach the point marker. If there are multiple traces in a graph window, select the *Attach to Trace* check box, and then select a trace from the drop-down list. Now, when you click either the *Next POI* or the *Previous POI* button on the *Snap* toolbar, the point marker is attached to the nearest interpolated data point on the selected trace. Alternatively, you can also use the `N` and `P` bindkeys to move the point marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Frequency	Specifies the frequency value of data point on the circular graph where you want to place the point marker. ■ If the data point for the specified frequency exists on the graph, the marker is moved to that data point. ■ If the data point for the specified frequency does not exist on the graph, and the frequency value is within the range of frequencies for the waveform, the marker is moved to a point that is obtained using linear interpolation of the following values: ❑ The greatest frequency value from the dataset which is lower than the specified frequency value. ❑ The smallest frequency value from the dataset which is greater than the specified frequency value. ■ If the specified frequency is out of range of frequencies for the waveform, the marker is not moved and an error message is displayed on the status bar of Virtuoso Visualization and Analysis XL.
Real/Imag Position	Displays the real and imaginary values of the data point on the trace where the point marker is currently placed. This is a read-only field.
Gamma	Displays the magnitude and the angle of the data point on the trace where the point marker is currently placed. This is a read-only field.
Impedance (Z)	Displays the impedance of the data point on the trace where marker is currently placed. This is a read-only field.
Admittance (Y)	Displays the admittance of the data point on the trace where marker is currently placed. This is a read-only field.
Font/Color	Specifies the style, size, and effect of the selected font for the label of the point marker. Click *Foreground* to select the color of the point marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Notation	Specifies the notation format for the label of the point marker: ■ *Scientific*: Uses the scientific notation in the marker label. For example: `3.631E7 Hz`. ■ *Engineering*: Uses the engineering notation in the marker label. For example:`36.31E6 Hz`. ■ *Suffix*: Uses the suffix notation in the marker label. For example:`36.31M Hz`. This is the default value.
Significant Digits	Specifies the number of significant digits to be displayed in the label of the point marker. The default value is `4`. To change this value, you need to first select *Manual* from the drop-down list. For example: ■ The value `36.308M Hz` contains `5` significant digits. ■ The value `36.3M Hz` contains `3` significant digits.

© 2023 Cadence Design Systems, Inc.

The Polar Reference Point Marker Properties Form

The Polar Reference Point Marker Properties form lets you edit the properties of a reference point marker on a polar graph.

Field	Description
Frequency	Specifies the frequency value of data point on the polar graph where you want to place the reference point marker. ■ If the data point for the specified frequency exists on the graph, the marker is moved to that data point. ■ If the data point for the specified frequency does not exist on the graph, and the frequency value is within the range of frequencies for the waveform, the marker is moved to a point that is obtained using linear interpolation of the following values: ❑ The greatest frequency value from the dataset which is lower than the specified frequency value. ❑ The smallest frequency value from the dataset which is greater than the specified frequency value. ■ If the specified frequency is out of range of frequencies for the waveform, the marker is not moved and an error message is displayed on the status bar of Virtuoso Visualization and Analysis XL.
Real/Imag Position	Displays the real and imaginary values of the data point on the trace where the point marker is currently placed. This is a read-only field.
Gamma	Displays the magnitude and the angle of the data point on the trace where the point marker is currently placed. This is a read-only field.
Impedance (Z)	Displays the impedance of the data point on the trace where marker is currently placed. This is a read-only field.
Admittance (Y)	Displays the admittance of the data point on the trace where marker is currently placed. This is a read-only field.

© 2023 Cadence Design Systems, Inc.

Field	Description
Notation	Specifies the notation to be used. Possible values are: ■ *Engineering*: Uses engineering notation. For example, `29E6 Hz`. ■ *Scientific*: Uses scientific notation. For example, `2.9E7 Hz`. ■ *Suffix*: (Default) Uses suffix notation. For example, `29M Hz`.

© 2023 Cadence Design Systems, Inc.

The Preferences Form

The Preferences form lets you specify the number of actions that can be undone or redone in the Virtuoso Visualization and Analysis XL Table.

Field	Description
Limit Undo Actions	Specifies the number of actions that can be undone or redone. You can specify any integer value between `1`–`10`. The default value is `2`, which indicates that only last two performed actions can be undone or redone.

Related Topics

Limiting the Number of Actions to Undo or Redo

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

The Print Form

The Print form lets you print the contents of the Virtuoso Visualization and Analysis XL Table.

Field	Description
Printer	Lets you select a printer and configure its properties.
Name	Lets you select the name of the printer to be used.
Location	Displays the location of the selected printer.
Type	Displays the type of the selected printer.
Properties	Opens the Printer Properties form where you can configure page properties and job options.
Options	Expands the Print form to display additional options contained in the *Copies* and *Options* sections.
Copies	Lets you specify the output settings and range of pages to be printed.
Print Range	Lets you specify the following options: ■ *Print all*: Prints all the pages. ■ *Pages from*: Specifies the range of pages to be printed. ■ *Page Set*: Specifies whether to print all pages, odd pages, or even pages.
Output Settings	Lets you specify the following options: ■ *Copies*: Number of copies to be printed. ■ *Collate*: Prints all the pages of a single copy together when printing multiple copies. ■ *Reverse*: Starts printing from the last page.
Print	Prints the contents of the Table.

Related Topics

The Printer Properties Form

Printing Contents of the Table

Virtuoso Visualization and Analysis XL Forms

© 2023 Cadence Design Systems, Inc.

The Printer Properties Form

The Printer Properties form lets you configure the page settings and job options. The form contains the following tabs:

Tab	Description
The Page Tab	Lets you configure the page settings, such as size, orientation, margin, and layout.
The Job Options Tab	Lets you configure job options, such as print schedule, job priority, and level of information sensitivity.

The Page Tab

The following table describes the fields available on the *Page* tab of the Printer Properties form.

Field	Description
Unit	Lets you select the unit in which width, height, and margins of the page are displayed in the form. The default value is *millimeters* (*mm*).
Paper	Lets you specify options for print schedule, billing information, and job priority.
Page size	Lets you select the size of the paper. The default value is *A4*.
Width	Displays the width of the selected paper size.
Height	Displays the height of the selected paper size.
Orientation	Specifies the orientation of the paper. Possible values are *Portrait* and *Landscape*. The default value is *Portrait*.
Margins	Specifies the left, right, top, and bottom margins of the page. The default value for all margins are `3.56mm`.
Page Layout	Lets you specify options for page per sheet and page order.

© 2023 Cadence Design Systems, Inc.

Field	Description
Pages per sheet	Specifies the number of pages you want to print per sheet. The default value is `1`(`1`x`1`).
Page order	Specifies the order in which pages are printed. By default, pages are printed from left to right and then from top to bottom

The Job Options Tab

The following table describes the fields available on the *Job Options* tab of the Printer Properties form.

Field	Description
Job Control	Lets you specify options for print schedule, billing information, and job priority.
Scheduled Printing	Lets you select the print schedule. The default value is *Print Immediately*.
Billing Information	Specifies the billing information.
Job priority	Specifies the job priority. The default value is *50*.
Banner Pages	Lets you specify the level of information sensitivity.
Start	Lets you select the level of information sensitivity to be printed at the start of every page. The default value is *None*.
End	Lets you select the level of information sensitivity to be printed at the end of every page. The default value is *None*.

Related Topics

The Print Form

Printing Contents of the Table

© 2023 Cadence Design Systems, Inc.

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

The Reference Point Marker Properties Form

The Reference Point Marker Properties form lets you edit the properties of a reference point marker.

Field	Description
Label	Specifies the format specifiers whose corresponding information you want to display in the marker label. Alternatively, you can click the *Edit* button to open the Edit Marker Label form wherein you can edit the information you want to display in the marker label.
Position	Lets you select the x- and/or y-axis coordinates of the point on the trace where you want to place the point marker. ■ byXMode: Places the point marker to the specified x-axis value. ■ byYMode: Places the point marker to the specified y-axis value. ■ byXYMode: Places the point marker to the specified x- and y-axis values.
Font/Color	Specifies the style, size, and effect of the selected font for the label of the point marker. Click *Foreground* to select the color of the point marker.
Notation	Specifies the notation format for the label of the point marker: ■ *Scientific*: Uses the scientific notation in the marker label. For example: `1.234E-8`. ■ *Engineering*: Uses the engineering notation in the marker label. For example:`12.34E-9`. ■ *Suffix*: Uses the suffix notation in the marker label. For example: `12.34n`. This is the default value.

© 2023 Cadence Design Systems, Inc.

Field	Description
Significant Digits	Specifies the number of significant digits to be displayed in the intercepts by the marker on the trace. The default value is `4`. To change this value, you need to first select *Manual* from the drop-down list. For example: ■ The value `4.1442e-8` contains `5` significant digits. ■ The value `4.1e-8` contains `2` significant digits.
Cross Hairs	Specifies whether you want to display cross hairs for the point marker. The cross hairs display the x- and y-axis intercepts by the point marker. ■ *Off*: Cross hairs are not displayed. ■ *On*: Cross hairs are displayed. ■ *Dynamic*: Cross hairs are displayed only when you click the point marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Next/Prev SnapPoint	Specifies one of the following snapping criteria based on which the selected reference marker is snapped when you click the *Next POI* or *Previous POI* command on the *Snap* toolbar: ■ *Local Maxima*: Snaps the marker to a local maxima. Local maxima is the transition point when there is a change in the slope from the rising to falling edge starting from the marker's current position. The local maxima is calculated as the change in slope from rising to falling edge starting from the current marker position and the transition point is known as the local maxima. ■ *Local Minima*: Snaps the marker to a local minima. Local minima is the transition point when there is a change in the slope from the falling to rising edge starting from the marker's current position. ■ *Local Max or Min*: Snaps the marker to either a local maxima or a local minima. ■ *Specific Y Value*: Snaps the marker to the y-axis value specified either in *Y Position* field or in the text field on the *Snap* toolbar. ■ *Data Point*: Snaps the marker to the data points available on the trace. ■ *Global Maxima*: Snaps the marker to the maximum y-axis value on the trace. ■ *Global Minima*: Snaps the marker to the minimum y-axis value on the trace. You can also use the `N` or `P` bindkeys to snap the marker.

© 2023 Cadence Design Systems, Inc.

The Reveal Columns Form

The Reveal Columns form lets you select the hidden columns that you want to reveal again in the Virtuoso Visualization and Analysis XL Table.

Field	Description
Col	Displays the Index number of the hidden column. The index number of the first column is `0`.
Header Value	Displays the header name of the hidden column.
Select All	Lets you select all the hidden columns.

Related Topics

Revealing Columns in the Table

Hiding Columns in the Table

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

The Sample Values Form

The Sample Values form lets you filter data in the selected column of the Virtuoso Visualization and Analysis XL Table based on the specified range of x-axis values.

Field	Description
Start	Start value of the range of x-axis values. The default value depends on x-axis data of the waveform.
End	Last value of the range of x-axis values. The default value depends on x-axis data of the waveform.
Step	The step value. **Note:** Do not specify the *Step* value if the Log check box is selected.
Log	Specifies whether to use the logarithmic scale. By default, this check box is not selected.

Related Topics

Filtering Table Data

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

The Save Image Form

The Save Image form lets you save the image of graphs plotted in subwindows in the specified file formats.

Field	Description
Look in	Specifies the results directory where you want to save the graph image. By default, this field displays the path to the directory from where the you opened the results in the Results Browser. You can use the following commands to quickly navigate between the directories: ■ : Navigates to the previous directory. ■ : Navigates to the next directory. ■ : Navigates to the parent directory of the currently selected directory. ■ : Creates a new subdirectory in the currently selected directory. ■ : Switches to the list view. ■ : Switched to the detail view.
File name	Specifies the name of the file with which you want to save the graph image. You can also specify macros, `$TRACE` and `$SUBWIN`, in the filename. These are used to save image files with the trace names and subwindows names.
Files of type	Specifies the file format in which you want to save the graph image.
Save	Lets you specify the subwindows you want to save.
Selected subwindow only	Saves image of the graphs plotted in current subwindow.

© 2023 Cadence Design Systems, Inc.

Field	Description
All subwindows, using	Saves image of the graphs plotted in all subwindows of the current graph window. Select one of the following options: ■ *Single file*: The graph images of all subwindows of the current graph window are saved in a single file. ■ *Multiple files*: The graph images of all subwindows of the current graph windows are saved separately in multiple files. If you select this option, ensure that you specify the macros, `$TRACE` or `$SUBWIN` in the *File names* field to save the graph images either by trace names or by subwindow numbers.
Render exactly as screen	Saves the exact copy of graph as it is visible on the screen. This also saves all attributes of the graph, including zooming and panning properties. When you select this option, the *Resize Image(s)* and the *Graph Display* sections are disabled.
Resize Images	Lets you resize the graph images being saved. This section becomes unavailable when you select the *Render exactly as screen* check box.
Width	Specify the width of the graph image. The image width can be specified in the following units: *pixels*, *inch*, *cm*, *mm*, *picas*, and *points*. When you change the unit, the image width is automatically changed as per the selected unit.
Height	Specifies the height of the graph image. The image height can be specified in the following units: *pixels*, *inch*, *cm*, *mm*, *picas*, and *points*. When you change the unit, the image height is automatically changed as per the selected unit.

© 2023 Cadence Design Systems, Inc.

Field	Description
Resolution	Specifies the resolution of the graph image. The image resolution can be specified in the following units: *pixels/in*, and *pixels/cm*. When you change the resolution unit, the image height and widths also change accordingly. This field becomes unavailable when you save the image in the following formats: ■ *Encapsulated PostScript* (`.eps`) ■ *Adobe PDF* (`.pdf`) ■ *Scalable Vector Graphics* (`.svg`)
Maintain aspect ratio	Maintains the aspect ratio of the graph image while modifying the width or height of the image. The aspect ratio of the image is ratio of the image width to image height.
Import dimensions	Imports the height and width of the waveform window and sets them as the height and width of the image.
Graph Display	Lets you specify the graph attributes and properties you want to display in the graph image.
Title	Displays or hides the graph title in the graph image.
Axes	Displays or hides graph axes in the graph image.
Legend	Displays or hides the trace legend in the graph image.
Grids	Displays or hides grids in the graph image.
Replace background color with	Lets you save the graph image with a different background color. Click the rectangular box next to this field to open the Select Color form wherein you can select the color of your choice.
Copy to Clipboard	Copies the graph image of the current subwindow to the clipboard. This option becomes unavailable when you select the *All subwindows, using* check box in the *Save* section.

© 2023 Cadence Design Systems, Inc.

Field	Description
Format Options	Opens the Format Option form wherein you can change the quality of the graph image being saved. On the quality bar, which ranges from `20`–`100`, you can move the slider to the right or to the left to increase or decrease the quality of the saved image. For example, image quality of the file saved with quality `20%` is `1/5`th of the file that is saved with quality `100%`. This option works only when the graph image is being saved in the JPEG format (`.jpg`).

© 2023 Cadence Design Systems, Inc.

The Save As CSV Form

The Save As CSV form lets you save the contents of the Virtuoso Visualization and Analysis XL Table in a CSV file.

Field	Description
Look in	Specifies the directory in which you want to save the CSV file. By default, this field displays the path of the directory from where the Virtuoso or Virtuoso or Virtuoso Visualization and Analysis XL is opened. You can use the following commands to quickly navigate between the directories: ■ : Navigates to the previous directory. ■ : Navigates to the next directory. ■ : Navigates to the parent directory of the currently selected directory. ■ : Creates a new subdirectory in the currently selected directory. ■ : Switches to the list view. ■ : Switched to the detail view.
File name	Specifies the name of the CSV file to which you want to save the contents of the Table.
Files of type	Displays all files with the CSV format in the currently selected directory.

Related Topics

Saving Table Contents to a CSV File

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

The Save Image Form

The Save Image form lets you save the image of graphs plotted in subwindows in the specified file formats.

Field	Description
Look in	Specifies the results directory where you want to save the graph image. By default, this field displays the path to the directory from where the you opened the results in the Results Browser. You can use the following commands to quickly navigate between the directories: ■ : Navigates to the previous directory. ■ : Navigates to the next directory. ■ : Navigates to the parent directory of the currently selected directory. ■ : Creates a new folder inside the currently selected directory. ■ : Switches to the list view. ■ : Switched to the detail view.
File name	Specifies the name of the file with which you want to save the graph image. You can also specify macros, `$TRACE` and `$SUBWIN`, in the filename. These are used to save image files with the trace names and subwindows names.
Files of type	Specifies the file format in which you want to save the graph image.
Save	Lets you specify the subwindows you want to save.
Selected subwindow only	Saves image of the graphs plotted in current subwindow.

© 2023 Cadence Design Systems, Inc.

Field	Description
All subwindows, using	Saves image of the graphs plotted in all subwindows of the current graph window. Select one of the following options: ■ *Single file*: The graph images of all subwindows of the current graph window are saved in a single file. ■ *Multiple files*: The graph images of all subwindows of the current graph windows are saved separately in multiple files. If you select this option, ensure that you specify the macros, `$TRACE` or `$SUBWIN` in the *File names* field to save the graph images either by trace names or by subwindow numbers.
Render exactly as screen	Saves the exact copy of graph as it is visible on the screen. This also saves all attributes of the graph, including zooming and panning properties. When you select this option, the *Resize Image(s)* and the *Graph Display* sections are disabled.
Resize Images	Lets you resize the graph images being saved. This section becomes unavailable when you select the *Render exactly as screen* check box.
Width	Specify the width of the graph image. The image width can be specified in the following units: *pixels*, *inch*, *cm*, *mm*, *picas*, and *points*. When you change the unit, the image width is automatically changed as per the selected unit.
Height	Specifies the height of the graph image. The image height can be specified in the following units: *pixels*, *inch*, *cm*, *mm*, *picas*, and *points*. When you change the unit, the image height is automatically changed as per the selected unit.

© 2023 Cadence Design Systems, Inc.

Field	Description
Resolution	Specifies the resolution of the graph image. The image resolution can be specified in the following units: *pixels/in*, and *pixels/cm*. When you change the resolution unit, the image height and widths also change accordingly. This field becomes unavailable when you save the image in the following formats: ■ *Encapsulated PostScript* (`.eps`) ■ *Adobe PDF* (`.pdf`) ■ *Scalable Vector Graphics* (`.svg`)
Maintain aspect ratio	Maintains the aspect ratio of the graph image while modifying the width or height of the image. The aspect ratio of the image is ratio of the image width to image height.
Import dimensions	Imports the height and width of the waveform window and sets them as the height and width of the image.
Graph Display	Lets you specify the graph attributes and properties you want to display in the graph image.
Title	Displays or hides the graph title in the graph image.
Axes	Displays or hides graph axes in the graph image.
Legend	Displays or hides the trace legend in the graph image.
Grids	Displays or hides grids in the graph image.
Replace background color with	Lets you save the graph image with a different background color. Click the rectangular box next to this field to open the Select Color form wherein you can select the color of your choice.
Copy to Clipboard	Copies the graph image of the current subwindow to the clipboard. This option becomes unavailable when you select the *All subwindows, using* check box in the *Save* section.

© 2023 Cadence Design Systems, Inc.

Field	Description
Format Options	Opens the Format Option form wherein you can change the quality of the graph image being saved. On the quality bar, which ranges from `20`–`100`, you can move the slider to the right or to the left to increase or decrease the quality of the saved image. For example, image quality of the file saved with quality `20%` is `1/5`th of the file that is saved with quality `100%`. This option works only when the graph image is being saved in the JPEG format (`.jpg`).

© 2023 Cadence Design Systems, Inc.

The Select Color Form

The Select Color form lets you specify the default colors for various signal types in the Results Browser.

Field	Description
Signal Type	Lets you select the signal type for which you want to change the default color. Valid values are: ■ *Voltage (V)* ■ *Current (I)* ■ *Power (P)* ■ *Logic (L)* ■ *Logic Bus (B)* ■ *Expression (E)* ■ *S Param (S)* ■ *Y Param (Y)* ■ *Z Param (Z)* ■ *Scalars*
Color	Lets you select the color that you want to set to the selected signal type.

Related Topics

Results Browser

© 2023 Cadence Design Systems, Inc.

The Select Waveform Database Form

The Select Waveform Database form lets you specify options to select waveform database files.

Field	Description
Look in	Specifies the results directory you want to open. If required, you can open multiple results directories. By default, this field displays the path of the directory from where the you opened the results recently in the Results Browser. You can use the following commands to quickly navigate between the directories: ■ : Navigates to the previousdirectory. ■ : Navigates to the next directory. ■ : Navigates to the parent directory of the currently selected directory. ■ : Descends one level down to the currently selected directory. ■ : Refreshes the contents of the currently selected directory. ■ : Switches to the list view. ■ : Switched to the detail view.
File name	Specifies the name of the results directory you want to open.
Files of type	Specifies the type of files you want to be displayed. ■ *All DB*: Displays only the database files. ■ *All Files*: Displays all types of files.

Related Topics

Opening a Results Directory

Plotting a Signal from the Results Browser

Results Browser

© 2023 Cadence Design Systems, Inc.

The Set Sweep Ranges Form

The Set Sweep Ranges form lets you specify the time or frequency range for plotting transient or ac signals. You can also specify values for sweep variables for plotting data from a parametric sweep.

Field	Description
Time Range	Provides options to specify time range for plotting transient signals.
Start	Time at which you want the plot to begin. If you want the graph to be plotted from the first data point in the signal, select *Default* from the Start drop-down list.
End	Time at which you want the plot to end. If you want the graph to be plotted till the last data point in the signal, select *Default* from the Start drop-down list.
Freq Range	Provides options to specify frequency range for plotting ac signals.
Start	Frequency at which you want the plot to begin. If you want the graph to be plotted from the first data point in the signal, select *Default* from the Start drop-down list.
End	Frequency at which you want the plot to end. If you want the graph to be plotted till the last data point in the signal, select *Default* from the Start drop-down list.

Related Topics

Plotting a Transient Signal Over a Time Range

Plotting an AC Signal over a Frequency Range

Results Browser

© 2023 Cadence Design Systems, Inc.

The Smith Reference Point Marker Properties Form

The Smith Reference Point Marker Properties form lets you edit the properties of a reference point marker on a Impedance, Admittance, and Immittance Smith graph.

Field	Description
Frequency	Specifies the frequency value of data point on the graph where you want to place the reference point marker. ■ If the data point for the specified frequency exists on the graph, the marker is moved to that data point. ■ If the data point for the specified frequency does not exist on the graph, and the frequency value is within the range of frequencies for the waveform, the marker is moved to a point that is obtained using linear interpolation of the following values: ❑ The greatest frequency value from the dataset which is lower than the specified frequency value. ❑ The smallest frequency value from the dataset which is greater than the specified frequency value. ■ If the specified frequency is out of range of frequencies for the waveform, the marker is not moved and an error message is displayed on the status bar of Virtuoso Visualization and Analysis XL.
Real/Imag Position	Displays the real and imaginary values of the data point on the trace where the reference point marker is currently placed. This is a read-only field.
Gamma	Displays the magnitude and the angle of the data point on the trace where the reference point marker is currently placed. This is a read-only field.
Impedance (Z)	Displays the impedance of the data point on the trace where the reference point marker is currently placed. This is a read-only field.
Admittance (Y)	Displays the admittance of the data point on the trace where the reference point marker is currently placed. This is a read-only field.

© 2023 Cadence Design Systems, Inc.

Field	Description
Notation	Specifies the notation to be used. Possible values are: ■ *Engineering*: Uses engineering notation. For example, `29E6 Hz`. ■ *Scientific*: Uses scientific notation. For example, `2.9E7 Hz`. ■ *Suffix*: (Default) Uses suffix notation. For example, `29M Hz`.

© 2023 Cadence Design Systems, Inc.

The Swap Sweep Var Form

The Swap Sweep Var form lets you plot an output waveform whose x-axis variable is swapped with the specified sweep variable of the selected input waveform. The y-axis variable of the input and output waveforms remains same.

The Swap Sweep Var option is enabled only for the simulation results that involve more than one sweep variable.

Field	Description
Variable to Sweep	Lets you select the sweep variable that you want to swap with the x axis of the selected waveform.
Plot to	Lets you select the following options: ■ *Replace*: The output waveform is plotted in the current graph window, replacing the input waveform. ■ *New subwindow*: The output waveform is plotted to a new subwindow of the graph window in which the input waveform is plotted. ■ *New Window*: The output waveform is plotted in a new graph window. ■
Graph Title	Specifies the title of the output graph. The default title is Signals vs. `variable` at value `X Value`. If the selection in the *Variable to Sweep* drop-down list is `temp` and the *X Value* field is set to `50ns`, the default title of the output graph will be: *Signals vs. temp at value 50ns.*
X Value	Specifies the x-axis value at which the input waveform is swapped with the sweep variable. This is a mandatory field.

© 2023 Cadence Design Systems, Inc.

The Value At Form

The Value At form lets you specify the x-axis value corresponding to which you want to view the y-axis value of waveform.

Field	Description
Value At	X-axis value corresponding to which you want to view the y-axis value of the waveform.

Related Topics

Filtering Table Data

Virtuoso Visualization and Analysis XL Table

© 2023 Cadence Design Systems, Inc.

The Vertical Marker Properties Form

The Vertical Marker Properties form lets you edit the properties of a vertical marker.

Field	Description
Name	Specifies the name of the vertical marker.
X Position	Specify the x-axis coordinate where you want to place the vertical marker.
Label	Lets you specify whether you want to display a label where the vertical marker intercepts the x axis. ■ *Follow Intercepts*: Label is displayed based on the following options specified in the *Intercepts* field: ❑ *off*: Intercept label on the x axis is not displayed ❑ *on when hover*: Intercept label on the x axis is displayed only on hovering over the vertical marker. ❑ *on*: Intercept label on the x axis is displayed ■ *Off*: Intercept label on the x axis is not displayed, irrespective of the options specified in the *Intercepts* field. ■ *On*: Intercept label on the x axis is displayed, irrespective of the options specified in the *Intercepts* field. The default value is *Follow Intercepts*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Intercepts	Lets you specify the following options: *Maximum*: Specifies the maximum number of intercepts displayed by the vertical marker on the trace. For example, if you specify `2` in this field, only the first two intercepts by the vertical marker are displayed on the trace. The default value is `20`. You can specify any value between `0`–`200`. The following options lets you specify when you want to display the intercept labels. ■ *off*: The intercepts by the vertical marker are not displayed on the trace. ■ *on when hover*: The intercepts by the vertical marker are displayed on the trace only when hovering over the vertical marker. This is the default value. ■ *on*: The intercepts by the vertical marker are displayed on the trace. The following options lets you specify the notation format for vertical label on the x axis and intercepts labels on the trace: ■ *Scientific*: Sets the notation format to scientific. For example: `-4.386e-1V`. ■ *Engineering*: Sets the notation format to engineering. For example:`-438.6E-3V`. ■ *Suffix*: To change the notation format to suffix. For example: `-438.6mV`. This is the default value.
Font/Color	Specifies the style, size, and effect of the selected font for the x-axis label, intercepts labels, and marker name. Click *Foreground* to select the color for the vertical marker and its name.
LineStyle	Specifies one of the following line styles for the vertical marker: *Solid*, *Dot*, *Dash*, *DashDot*, and *DashDotDot*. The default value is *Dash*.

© 2023 Cadence Design Systems, Inc.

Field	Description
Next/Prev SnapPoint	Specifies one of the following snapping criteria based on which the selected horizontal marker is snapped when you click the *Next POI* or *Previous POI* command on the *Snap* toolbar: ■ *Local Maxima*: Snaps the marker to a local maxima. Local maxima is the transition point when there is a change in the slope from the rising to falling edge starting from the marker's current position. The local maxima is calculated as the change in slope from rising to falling edge starting from the current marker position and the transition point is known as the local maxima. ■ *Local Minima*: Snaps the marker to a local minima. Local minima is the transition point when there is a change in the slope from the falling to rising edge starting from the marker's current position. ■ *Local Max or Min*: Snaps the marker to either a local maxima or a local minima. ■ *Specific Y Value*: Snaps the marker to the y-axis value specified in the text field on the *Snap* toolbar. ■ *Specific X Value*: Snaps the marker to the x-axis value specified either in *X Position* field or in the text field on the *Snap* toolbar. ■ *Data Point*: Snaps the marker to the data points available on the trace. ■ *Global Maxima*: Snaps the marker to the maximum y-axis value on the trace. ■ *Global Minima*: Snaps the marker to the minimum y-axis value on the trace. You can also use the `N` or `P` bindkeys to snap the marker.

© 2023 Cadence Design Systems, Inc.

Field	Description
Significant Digits	Specifies the number of significant digits to be displayed in the intercepts by the marker on the trace. The default value is `4`. To change this value, you need to first select *Manual* from the drop-down list. For example: ■ The value `4.1442e-8` contains `5` significant digits. ■ The value `4.1e-8` contains `2` significant digits.

© 2023 Cadence Design Systems, Inc.

The Waveforms to be Exported Form

The Waveforms to be Exported form lets you specify options to export waveforms from multiple results directories to an export list.

Field	Description
Look in	Specifies the directory in which you want to save the exported waveform data.
File name	Name of the file to which you want to export the selected waveforms.
Files of type	Specifies one of the following formats in which you want to export the waveform: ■ *VCSV* ■ *Matlab* ■ *PSF* ■ *SPECTRE* ■ *SST2* ■ *CSV* ■ *PSFXL*
Export List	Displays a list of signals you selected in the Results Browser to export.

Related Topics

Exporting Waveforms from Multiple Results Directories

Results Browser

© 2023 Cadence Design Systems, Inc.

The Window Properties Form

The Window Properties form lets you specify the background color and gradient for a graph window or subwindow.

Field	Description
Color	Provides options to specify the background color and gradient for a graph window or subwindow.
Background	Specifies the background color of a graph window or subwindow.
Use Gradient	Adds a gradient to the background, which means the background color fades at the bottom of the graph.
Subwindow Minimum Size	Specifies the minimum width and minimum height of subwindows in a new graph window.

Related Topics

graphMinWidth

graphMinHeight

© 2023 Cadence Design Systems, Inc.

The Y vs Y Form

The Y vs Y form lets you change the independent axis to plot several signals at once against a new x-axis variable, which is the y-axis variable (dependent axis) of the selected trace.

Field	Description
Select a Trace	Lets you select the trace whose dependent axis (y-axis variable) is used as the independent axis (x-axis variable) of the output waveform.
Plot Destination	Lets you select the following options: ■ *New subwindow*: The output waveform is plotted to a new subwindow of the graph window in which the input waveform is plotted. ■ *New Window*: The output waveform is plotted in a new graph window.
Graph Title	Specifies the title of the output graph. The default title is *Unit1* vs. *Unit2*, where *Unit2* is the y-axis variable of the selected trace and *Unit1* is the y-axis variable of the existing trace. For example: *V vs. I*

© 2023 Cadence Design Systems, Inc.

The Zoom To Form

The Zoom To form lets you specify the options to zoom out the graph window to the specified size of the pan bar.

Field	Description
Minimum	Specifies the minimum value for the pan bar.
Maximum	Specifies the maximum value for the pan bar.

© 2023 Cadence Design Systems, Inc.

A

Virtuoso Visualization and Analysis XL Tool Environment Variables

The variables and values that specify the basic behavior of the components of the Virtuoso Visualization and Analysis XL are part of the `.cdsenv` file. For information about the order in which the tool reads the `.cdsenv` file, see Creating the .cdsenv File in Chapter 10 of the *Cadence Design FrameworkII User Guide*.

This appendix describes the Virtuoso Visualization and Analysis XL tool variables in the `.cdsenv` file. In each entry, the first column is the tool, the second column is the variable, the third column is the data type, and the fourth column contains the value to be used.

In the SKILL mode, you can use the `envGetVal` and `envSetVal` functions to retrieve and set the `.cdsenv` variables in the CIW. For more information on the CIW or the `envGetVal` and `envSetVal` functions, see Chapter 2 in the *Cadence User Interface SKILL Functions Reference*.

Note: The `envSetVal` settings work for the expressions that are plotted from ADE or from Virtuoso Visualization and Analysis XL when opened in the stand-alone mode. These settings also work for the signals that are plotted when Virtuoso Visualization and Analysis XL is opened in the stand-alone mode. On the other hand, the signals plotted from ADE follow the `display.drf` settings.

The environment variables for Results Browser and Calculator are included in the .cdsenv file at the following location:

```
./tools.lnx86/dfII/samples/wavescan/.cdsenv
```

The environment variables for graph are included in the .cdsenv file at the following location:

```
./tools.lnx86/dfII/samples/viva/.cdsenv
```

This chapter includes the following sections that decsribes the environement variable for Virtuoso Visualization and Analysis XL:

■ Graph Variables on page 761

© 2023 Cadence Design Systems, Inc.

- Results Browser Variables on page 967
- Calculator Variables on page 971
- Table Environment Variables on page 998

© 2023 Cadence Design Systems, Inc.

Graph Variables

This section describes the environment variables that are used to set the graph properties:

- Font String on page 762
- Graph Frame Variables on page 765
- Graph Environment Variables on page 778
- Rectangular Graphs Environment Variables on page 783
- Strip Environment Variables on page 784
- Digital Strip Environment Variables on page 785
- Circular Graph Environment Variables on page 786
- Axis Environment Variables on page 792
- Dependent Axis Environment Variables on page 793
- Independent Axis Environment Variables on page 797
- String Independent Axis Environment Variables on page 802
- Trace Environment Variables on page 803
- Trace Legend Environment Variables on page 804
- Digital Trace Environment Variables on page 807
- Digital Bus Trace Environment Variables on page 808
- Histogram Environment Variables on page 812
- Horizontal Marker Environment Variables on page 813
- Reference Line Marker Environment Variables on page 819
- Vertical Marker Environment Variables on page 823
- Point Marker Environment Variables on page 829
- Reference Point Marker Environment Variables on page 834
- Specification Marker Environment Variables on page 838
- Intercept Marker Environment Variables on page 845
- Circle Marker Environment Variables on page 847

© 2023 Cadence Design Systems, Inc.

- Delta Marker Environment Variables on page 850
- Graph Label Environment Variables on page 866
- Probe Environment Variables on page 867
- Polar Grid Environment Variables on page 868
- Smith Grid Environment Variables on page 869
- Application Environment Variables on page 924
- XRange Bookmark Variables on page 931
- YRange Bookmark Variables on page 943
- Region Bookmark Variables on page 955

Font String

The font values for Virtuoso Visualization and Analysis XL is displayed in the following format:

```
Default,10,-1,5,50,0,0,0,0,0
```

The various fields in the font string are as follows:

<Font family>, <Point size>, <Pixel size>, <Style hint>, <Weight>, <Style>, <Underline>, <Strikeout>, <Fixed pitch>, <Raw mode>

Point size—Specifies point size of the font. If the font size is specified using pixels, it should be -1.

Pixel size—Specifies pixel size of the font. The font size value is -1 if it is specified using pixels. The pixel size makes the font device dependent. If you want to set the size of the font in a device independent You can use Point size to set the size of the font in a device independent manner.

Style hint—Specifies the style hint that is used by the font matching algorithm to find an appropriate default family if a selected font family is not available in the system. Style hints are not supported on X11 since this information is not provided by the window system. The value of '5' represents the enumeration:

QFont—Any Style. Any value other than `5` is ignored on X11 based systems, which means on(i.e., essentially on most unix systems).

Weight—Weighting scale for font rendering, which ranges from 0 to 99. This is similar to the scales used in Windows or CSS. A weight of `0` is ultralight and `99` is black.

© 2023 Cadence Design Systems, Inc.

Following values can be used for this aspect:

Enumeration	Value
Light	25
Normal	50
DemiBold	63
Bold	75
Black	87

Style—This aspect describes the different styles of glyphs that are used to display text. Following values are supported for this aspect:

Enumeration	Value	Description
StyleNormal	0	Normal glyph used in text with no style.
StyleItalic	1	Italic glyph designed for representing the italicized text.
StyleOblique	2	Italic glyph that are typically based on the graphs with no style. However, these glyphs are not fine-tuned for representing italicized text.

Underline—Set the value to `1` if you want to underline the text. Set the value to `0` if you do not want to underline the text.

Strikeout—Set the value to `1` if you want to strikeout the text. Set the value to 0 if you do not want to strikeout the text.

Fixed pitch—Set the value to `1` if you want to set the for the fixed pitch preference. Set the value to `0` if you do not want to set the fixed pitch.

Raw mode—Set the value to 1 if you want to use raw mode for font name matching. Otherwise, set the value to 0. This option is applicable only for the X11 systems. If the raw mode is enabled, the X font with a complete font name that matches with the family name is searched and all other values set for QFont are ignored. If the font name matches multiple fonts, the first font returned by X window system is used.

© 2023 Cadence Design Systems, Inc.

Important

When you set the font-related environment variables in the .cdsinit using the `envSetVal` command, you first need to load the `viva` context before setting the variable. For this, you can add the following command in your .cdsinit file:

```
loadContext(strcat(prependInstallPath("etc/context/") "viva.cxt"))
callInitProc("viva")
```

Now, you can use the `envSetVal` command to set the font variable in the .cdsinit file.

© 2023 Cadence Design Systems, Inc.

Graph Frame Variables

- `viva.graphFrame width string "1000"`
- `viva.graphFrame height string "800"`
- `viva.graphFrame autoTraceSelect string "true"`
- `viva.graphFrame rightMouseZoom string "false"`
- `viva.graphFrame selectBySweep string "false"`
- `viva.graphFrame useSplitter string "true"`
- `viva.graphFrame title string "Window"`
- `viva.graphFrame graphMinWidth string "200"`
- `viva.graphFrame graphMinHeight string "140"`
- `viva.graphFrame graphLayoutType string "Auto"`
- `viva.graphFrame useSpacer string "true"`
- `viva.graphFrame background string "black"`
- `viva.graphFrame useGradient string "false"`

width

Controls the width of the Graph Window.

Syntax

`viva.graphFrame width string "`*`width_pixels`*`"`

Values

`width_pixels`	Width of the graph window Default: Valid values: A positive integer

© 2023 Cadence Design Systems, Inc.

Example

```
viva.graphFrame width string "1000"
```

height

Controls the height of the Graph Window.

Syntax

```
viva.graphFrame height string "height_pixels"
```

Values

height_pixels	Height of the graph window Default: Valid values: A positive integer

Example

```
viva.graphFrame height string "800"
```

autoTraceSelect

Specifies whether the tool selects the trace closest to the system cursor.

Syntax

```
viva.graphFrame autoTraceSelect string "true" | "false"
```

Values

true	The tool automatically selects the trace closest to the system cursor. This is the default value.

© 2023 Cadence Design Systems, Inc.

`false`	The tool does not select a trace automatically.

Example

```
viva.graphFrame autoTraceSelect string "true"
```

rightMouseZoom

Specifies whether you can use the right mouse button to zoom your graph.

Syntax

```
viva.graphFrame rightMouseZoom string "true" | "false"
```

Values

`true`	Use the right mouse button to zoom your graph.This is the default value.
`false`	Use the middle mouse button to zoom your graph. This enables the right mouse pop-up menu in the Graph Window.

Example

```
viva.graphFrame rightMouseZoom string "false"
```

selectBySweep

Specifies whether traces from parametric sweeps are selected by family or by individual leaf.

Syntax

```
viva.graphFrame selectBySweep string "true" | "false"
```


© 2023 Cadence Design Systems, Inc.

Values

`true`	The *Trace – Select* by Family command is selected.
`false`	The *Trace – Select* by Family command is not selected. This is the default value.

Example

```
viva.graphFrame selectBySweep string "false"
```

useSplitter

Syntax

```
viva.graphFrame useSplitter string "true" | "false"
```

Values

`true`	This is the default value.
`false`	

Example

```
viva.graphFrame useSplitter string "true"
```

title

Specifies the title of the graph window.

```
viva.graphFrame title string "window_title"
```


© 2023 Cadence Design Systems, Inc.

Values

`window_title`	The title of the graph window.

Example

```
viva.graphFrame title string "Window"
```

graphLayoutType

Specifies the subwindow display layout type of the graph window.

```
viva.graphFrame graphLayoutType string "Auto"
```

Default Value

`Auto`

Valid Values

- `Card`
- `Horizontal`
- `Vertical`

useSpacer

Adds an empty spacer widget to make the number of open subwindows even.

`viva.graphFrame useSpacer 'string "`*`true`*`"`

Default Value

`true`

Valid Values

- `true`—Adds an empty spacer widget.

© 2023 Cadence Design Systems, Inc.

- `false`—Restores the legacy behavior of not adding an empty spacer widget.

background

Controls the background color of the window/subwindow.

```
viva.graphFrame background 'string "color_value"
```

Default Value

`color_value`	The title of the graph window. Default: `black` Valid values: A valid color name, Hex color code, or HTML color code.

Example

```
viva.graphFrame background string "black"
```

viva.graphFrame background string "#FFFFFF"

useGradient

Specifies whether the subwindow uses gradient color in the background.

```
viva.graphFrame useGradient 'string "true" | "false"
```

Default Value

`false`

Valid Values

- `true`—Enables gradient in the subwindow background.
- `false`—Disables gradient in the subwindow background.

© 2023 Cadence Design Systems, Inc.

Example

```
viva.graphFrame useGradient string "false"
```


© 2023 Cadence Design Systems, Inc.

linkDepAxes

```
viva.graphFrame linkDepAxes string {"true" | "false"}
```

Description

Specifies whether to link dependent axes (y axes) of graphs in the linked subwindows. The default value is `true`, which indicates that dependent axes (y axes) that have the same base unit are linked together.

GUI Equivalent

The *Link Dependent (Y) Axes* check box in the Link Options form.

Examples

```
envGetVal("viva.graphFrame" "linkDepAxes")
envSetVal("viva.graphFrame" "linkDepAxes" 'string "false")
```

Related Topics

linkIndepAxes

useLinkAxesUnitFilter

linkAxesUnitFilterSet

Linking Graphs in Subwindows

Linking Axes in Subwindows

Unlinking Graphs in Subwindows

© 2023 Cadence Design Systems, Inc.

linkIndepAxes

```
viva.graphFrame linkIndepAxes string {"true" | "false"}
```

Description

Specifies whether to link independent (x) axes of graphs in the linked subwindows. The default value is `true`, which indicates that independent axes (x axes) that have the same base unit are linked together.

GUI Equivalent

The *Link Independent (X) Axes* check box in the Link Options form.

Examples

```
envGetVal("viva.graphFrame" "linkIndepAxes")
envSetVal("viva.graphFrame" "linkIndepAxes" 'string "false")
```

Related Topics

linkDepAxes

useLinkAxesUnitFilter

linkAxesUnitFilterSet

Linking Graphs in Subwindows

Linking Axes in Subwindows

Unlinking Graphs in Subwindows

© 2023 Cadence Design Systems, Inc.

useLinkAxesUnitFilter

```
viva.graphFrame useLinkAxesUnitFilter string {"true" | "false"}
```

Description

Specifies whether to link only those axes that have the base units specified by the environment variable `linkAxesUnitFilterSet`. The default value is `false`, which indicates that the selective linking of dependent and independent axes is disabled.

GUI Equivalent

The *Only Link Axes with These Units* check box in the Link Options form.

Examples

```
envGetVal("viva.graphFrame" "useLinkAxesUnitFilter")
envSetVal("viva.graphFrame" "useLinkAxesUnitFilter" 'string "true")
```

Related Topics

linkDepAxes

linkIndepAxes

linkAxesUnitFilterSet

Linking Graphs in Subwindows

Linking Axes in Subwindows

Unlinking Graphs in Subwindows

© 2023 Cadence Design Systems, Inc.

linkAxesUnitFilterSet

```
viva.graphFrame linkAxesUnitFilterSet string "listOfBaseUnits"
```

Description

Specifies base units of the axes that you want to link together. You can specify multiple base units as a list of comma-separated values. The default value is "".

This environment variable is considered only when the environment variable `useLinkAxesUnitFilter` is set to `true`.

GUI Equivalent

The text field below the *Only Link Axes with These Units* check box in the Link Options form.

Examples

```
envGetVal("viva.graphFrame" "linkAxesUnitFilterSet")
envSetVal("viva.graphFrame" "linkAxesUnitFilterSet" 'string "V,A,Hz,s")
```

Related Topics

linkDepAxes

linkIndepAxes

useLinkAxesUnitFilter

Linking Graphs in Subwindows

Linking Axes in Subwindows

Unlinking Graphs in Subwindows

© 2023 Cadence Design Systems, Inc.

graphMinWidth

```
viva.graphFrame graphMinWidth string "minimumWidthOfSubwindow"
```

Description

Specifies the minimum width of subwindows in a new graph window. The default value is `250`.

GUI Equivalent

The *Subwindow Minimum Size Width* field in the Window Properties form.

Examples

```
envGetVal("viva.graphFrame" "graphMinWidth")
envSetVal("viva.graphFrame" "graphMinWidth" 'string "250")
```

Related Topics

The Window Properties Form

© 2023 Cadence Design Systems, Inc.

graphMinHeight

```
viva.graphFrame graphMinHeight string "minimumHeightOfSubwindow"
```

Description

Specifies the minimum height of subwindows in a new graph window. The default value is `250`.

GUI Equivalent

The *Subwindow Minimum Size Height* field in the Window Properties form.

Examples

```
envGetVal("viva.graphFrame" "graphMinHeight")
envSetVal("viva.graphFrame" "graphMinHeight" 'string "300")
```

Related Topics

The Window Properties Form

© 2023 Cadence Design Systems, Inc.

Graph Environment Variables

- `viva.graph titleFont string "Default,10,-1,5,50,0,0,0,0,0"`
- `viva.graph subTitle string ""`
- `viva.graph displayDate string "true"`
- `viva.graph useDefaultTitle string "true"`
- `viva.graph defaultSubtitle string "true"`
- `viva.graph traceMarkerOn string "true"`
- `viva.graph traceMarkerSignificantDigits string "4"`
- `viva.graph traceMarkerSigDigitsMode string "Auto"`
- `viva.graph snapOn string "snapOff"`

  The other valid value for `SnapOn` environment variable is: `snapToData`.
- `viva.graph selectByFamily string "false"`
- `viva.graph useCurrentContext string "true"`
- `viva.graph colorByParameter string "trace"`
- `viva.graph symbolByParameter string "family"`
- `viva.graph stripByParameter string "VDD"`
- `viva.graph colorBank string "default"`

  You can use this variable to customize the set of trace colors in a Virtuoso Visualization and Analysis XL session. The default value of this environment variable is the default color bank that is specified in the `display.drf` file.

  Note: This variable does not change the colors defined in the `display.drf` file. Also, when you plot simulation results from ADE, the color bank from the `display.drf` file is honored.

  The color names should be separated by a comma, a semicolon, or a space. The color name can be a list of colors defined in the SVG color keyword, such as steelblue, azzure, or the RGB hexadecimal value of the color in the `#RRGGBB` format.

  ```
  viva.graph colorBank string "red; blue; green; yellow"

  viva.graph colorBank string "#ff0000, #0000ff, #008000, #ffff00"
  ```


© 2023 Cadence Design Systems, Inc.

You can specify any number of colors names using this variable. However, it is recommended that you should not specify more than 18 colors because lot of colors might lead to confusion.

The legal color name and corresponding RGB value is listed at the following link:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

- `titleDisplayMode`

  Determines how the graph title is to be wrapped around if it exceeds the space available on the graph.

  ```
  viva.graph titleDisplayMode string "elideMiddle"
  ```

  Default value: `elideMiddle`

  Valid Values:

  - `elideLeft`—Graph title is elided from the left
  - `elideRight`—Graph title is elided from the right
  - `elideMiddle`—Graph title is elided in the middle
  - `multiLine`—Graph title can take multiple lines until it exceeds space available

© 2023 Cadence Design Systems, Inc.

- `reloadSingleStyle`:

  Determines whether to apply a single style to all the traces that were added to a trace group after the graph is reloaded.

  The `reloadSingleStyle` setting applies only when reloading different waveforms of a parametric signal.

  Each waveform of the parametric signal is assigned a unique color if the environment variable `reloadSingleStyle` is set to `false`.

  If this environment variable is set to `true`, each waveform of the signal is assigned the same color and style of the already plotted signal.

  When a graph window is created, it stores the value of this environment value and uses the same value when reloading data again.

  Therefore, if the value of this environment variable is `true` at the time of creation of the graph window, the same value will be used when reloading the graph window. Resetting the value of this environment variable does not have any effect on reloading the graph windows that were created earlier. The new value of the environment variable is used only for reloading the graph windows that are created after resetting the value of the variable.

  In `.cdsenv`:

  ```
  viva.graph reloadSingleStyle string "false"
  ```

  In `.cdsinit` or the CIW:

  ```
  envSetVal("viva.graph" "reloadSingleStyle" 'string
  "false")
  ```

  Default Value

  `false`

  Valid Values

  `true`: The trace style applied to all the traces added to the trace group after reloading the graph is same. The style to be applied is randomly selected from one of the existing traces in the trace group.

  `false`: The trace style applied to the traces added to the trace group after reloading the graph is different from that of the existing traces in the group. The trace styles of the existing traces are preserved.

© 2023 Cadence Design Systems, Inc.

- `undoStackSize`

  Specifies the maximum number of commands that can be undone in the graph window.

  In `.cdsenv`:

  ```
  viva.graph undoStackSize string "30"
  ```

  In `.cdsinit` or the CIW:

  ```
  envSetVal("viva.graph" "undoStackSize" 'string
  "30")
  ```

  Default Value `30`

- `enableDirectMeasurementAssistant`

  Specifies whether to enable or disable the *Direct Measurements* assistant.

  In `.cdsenv`:

  ```
  viva.graph enableDirectMeasurementAssistant string
  "true"
  ```

  In `.cdsinit`:

  ```
  envSetVal("viva.graph"
  "enableDirectMeasurementAssistant" 'string "true")
  ```

  Possible Values: `true`: Enables the *Direct Measurements* assistant.

  `false`: Disables the *Direct Measurements* assistant.

  Default Value `true`

- `dimOnSelection`

  Specifies whether to dim the unselected traces in a graph window or subwindow.

  In `.cdsenv`:

  ```
  viva.graph dimOnSelection string "true"
  ```

  In `.cdsinit`:

© 2023 Cadence Design Systems, Inc.

```
envSetVal("viva.graph" "dimOnSelection" 'string
"true")
```

Possible Values: `true`: Unselected traces appear dim.

`false`: Unselected traces do not appear dim.

Default Value `true`

© 2023 Cadence Design Systems, Inc.

Rectangular Graphs Environment Variables

- `viva.rectGraph useGradient string "false"`
- `viva.rectGraph traceMarkerDisplay string "(%X, %Y)"`
- `viva.rectGraph referenceLinesOn string "true"`
- `viva.rectGraph enableEdgeMeasurement string "false"`
- `viva.rectGraph legendPosition string "left"`

  Note: When you specify the `legendPosition` option as `left`, the trace legend is displayed on the left of the graph or strip. To move the trace legend inside the graph, specify `inside`. To move the trace legend to be displayed at the top of each strip, specify `above`.

- `viva.rectGraph eyeDiagramLegendPosition string "above"`

  The defalt trace legend position for an eye diagram is at the top of the plot. When you specify the `legendPosition` option as `left`, the trace legend is displayed on the left of the graph. To move the trace legend inside the graph, specify `inside`.

- `viva.rectGraph showDeltaChildLabels string "false"`

  When you set this variable to `false`, the labels are not displayed for the point markers that are part of a delta marker.

- `viva.rectGraph showZoomBar string "false"`

  When this variable is set to `false`, the zoom bar is not displayed on graph. To display the zoom bar, set this variable to `true`.

- `envSetVal("viva.rectGraph" "showEdgeBrowser" 'string "true")`

  When this variable is set to `false`, the Edge Browser is not displayed on the graph. This works only if the `enableEdgeMeasurement` environment variable is set to `true`.

© 2023 Cadence Design Systems, Inc.

Strip Environment Variables

- `viva.rectGraph stripChartOn 'string "true"`

  When you set this variable, all the new signals are plotted in a new strip.

- `viva.rectGraph stripChartOn string "false"`
- `viva.rectGraph stripByFamily string "false"`
- `viva.rectGraph stripHeight string "50"`
- `viva.rectGraph minStripHeight string "32"`
- `viva.rectGraph activeStripCue string "Vertical Bar"`
- `viva.rectGraph activeStripCueColor string "yellow"`

© 2023 Cadence Design Systems, Inc.

Digital Strip Environment Variables

- `viva.rectGraph digitalStripHeight string "32"`
- `viva.rectGraph maxDigitalStripHeight string "40"`
- `viva.rectGraph minDigitalStripHeight string "32"`

© 2023 Cadence Design Systems, Inc.

Circular Graph Environment Variables

- `viva.circGraph useGradient string "false"`
- `viva.circGraph gridTypestring "Polar"`
- `viva.circGraph characteristicImpedance string "50"`
- `viva.circGraph normalizeSmithValues string "true"`
- `viva.circGraph traceMarkerAlwaysVisible string "false"`
- `viva.circGraph flexibleAxis string "false"`

  Determines whether the circular graphs with different dependent axes can be plotted in the same window. When set to `false` (default value), graphs with matching axis units can only be plotted in the same window. In case you use the drag operation, a new window is created. When this variable is set to `true`, graphs with different axis units can be plotted together in the same window.

© 2023 Cadence Design Systems, Inc.

legendPosition

```
viva.circGraph legendPosition string {"above" | "left" | "inside"}
```

Description

Specifies the position of the trace legend in a circular graph. The valid values are as follows:

- `left`: The trace legend is displayed to the left of a circular graph. This is the default value.
- `above`: The trace legend is displayed at the top of a circular graph.
- `inside`: The trace legend is displayed inside a circular graph.

GUI Equivalent

Command	The *Graph Options* tab of the Circular Graph Properties form.
Field	*Legend Position*

Examples

```
envGetVal("viva.circGraph" "legendPosition")
envSetVal("viva.circGraph" "legendPosition" 'string "above")
envSetVal("viva.circGraph" "legendPosition" 'string "inside")
```

Related Topics

Editing Circular Graph Properties

© 2023 Cadence Design Systems, Inc.

plotToSmithView

```
viva.circGraph plotToSmithView string {"true" | "false"}
```

Description

Specifies how a Smith chart is plotted initially. Valid values are as follows:

- `true`: The Smith chart fits to the extent of the standard Smith grid. This is the default value.
- `false`: The Smith chart fits to data when opened for the first time.

If a compressed Smith grid is required to show data, the initial plot shows the full extent of the compressed Smith regardless of the value of this environment variable.

GUI Equivalent

None

Examples

```
envGetVal("viva.circGraph" "plotToSmithView")
envSetVal("viva.circGraph" "plotToSmithView" 'string "false")
```


© 2023 Cadence Design Systems, Inc.

showFreqFilterBar

```
viva.circGraph showFreqFilterBar string {"false" | "true"}
```

Description

Specifies whether to show the frequency filter bar in a circular graph window or subwindow. The valid values are as follows:

- `false`: The frequency filter bar is not displayed in a circular graph. This is the default value.
- `true`: The frequency filter bar is displayed in a circular graph.

GUI Equivalent

Command	The *Graph Options* tab of the Circular Graph Properties form.
Field	*Show Frequency Filter*

Examples

```
envGetVal("viva.circGraph" "showFreqFilterBar")
envSetVal("viva.circGraph" "showFreqFilterBar" 'string "true")
```

Related Topics

Editing Circular Graph Properties

Filtering Data Using the Frequency Slider Bar

© 2023 Cadence Design Systems, Inc.

traceMarkerDisplay

`viva.circGraph traceMarkerDisplay string` *`formatOfTraceMarker`*

Description

Specifies the format in which trace marker values are displayed. The possible values are as follows:

- `%C`: Displays the real and imaginary Cartesian values.
- `%P`: Displays the polar values, such as magnitude and angle.
- `%Z`: Displays the impedance values for resistance and reactance.
- `%A`: Displays the admittance values for as conductance and susceptance.
- `%R`: Displays the reflection coefficients, such as Gamma (magnitude) and angle (degree). The range of angle is `0`–`±180` degree. For the upper-half of the circle, the values of angle vary from `0`–`180` degree. For the lower-half of the circle, values of angle vary from `-180`–`0` degree.
- `%F`: Displays the frequency value, which includes the independent axis data.

The default value is an empty string, which indicates that the information displayed in the label of a tracking marker depends upon the type of circular grid:

- **Polar grid**: `%F%n%C`

  Frequency and real and imaginary Cartesian values are displayed in the label text of a tracking marker. %n indicates that values are displayed in a new line.

- I**mpedance (Z Smith) grid**: `%F%n%Z`

  Frequency and impedance values, such as resistance and reactance are displayed in the label text of a tracking marker.

- **Admittance (Y Smith) grid**: `%F%n%A`

  Frequency and admittance values, such as conductance and susceptance are displayed in the label text of a tracking marker.

- **Immittance (ZY Smith) grid**: `%F%n%Z%n%A`

  Frequency, impedance, and admittance values are displayed in the label text of a tracking marker.

© 2023 Cadence Design Systems, Inc.

GUI Equivalent

Command	The *Markers* tab of the Graph Properties form for circular graph.
Field	*Tracking Format (Trace)*

Examples

```
envGetVal("viva.circGraph" "traceMarkerDisplay")
envSetVal("viva.circGraph" "traceMarkerDisplay" 'string "%Z, %P")
envSetVal("viva.circGraph" "traceMarkerDisplay" 'string "%Z %P")
envSetVal("viva.circGraph" "traceMarkerDisplay" 'string "(%Z) %P")
```

Related Topics

Editing Circular Graph Properties

© 2023 Cadence Design Systems, Inc.

Axis Environment Variables

- `viva.axis majorGridsOn string "true"`
- `viva.axis minorGridsOn string "true"`
- `viva.axis majorGridForeground string "gray"`
- `viva.axis minorGridForeground string "lightGray"`
- `viva.axis autoAxisColor string "true"`
- `viva.axis foreground string "white"`
- `viva.axis background string "black"`
- `viva.axis font string "Default,10,-1,5,50,0,0,0,0,0"`
- `viva.axis majorTicsOn string "true"`
- `viva.axis minorTicsOn string "true"`

© 2023 Cadence Design Systems, Inc.

Dependent Axis Environment Variables

- `viva.depAxis baseAndToplineReferenceHint string "0.0,5.0"`
- `viva.depAxis threshold string "20,80"`
- `viva.depAxis logScale string "false"`
- `viva.depAxis forceOrigin string "false"`
- `viva.depAxis notation string "suffix"`
- `viva.depAxis showUnits string "true"`

  The `showUnits` variable is used to show or hide the units on the Y-axis

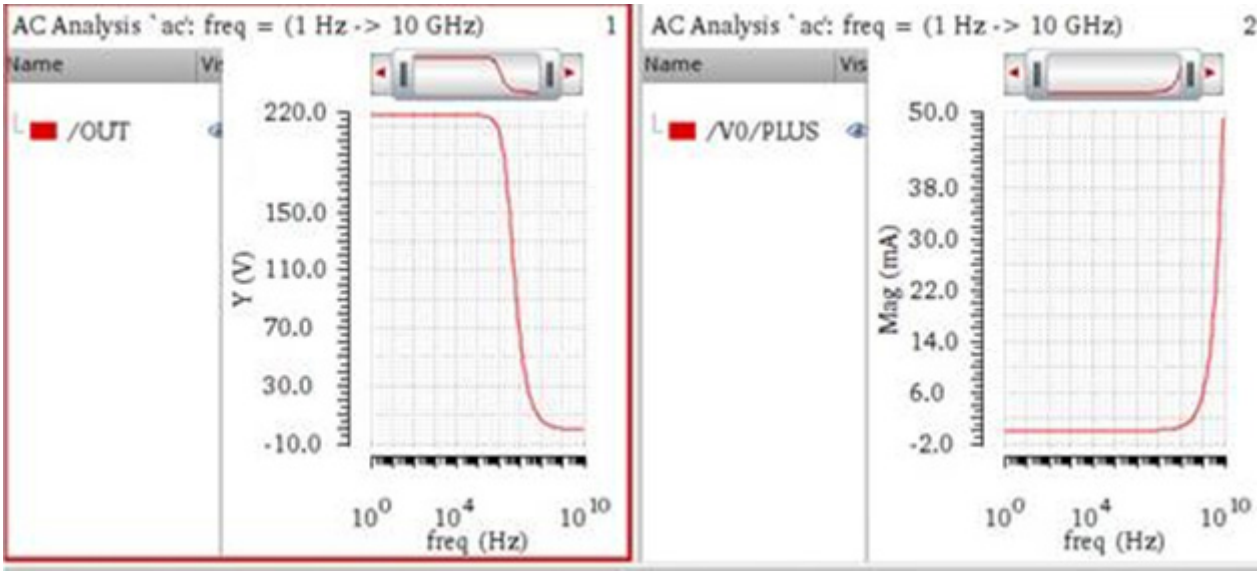
- `viva.depAxis flexible string "false"`

  Specifies whether a new trace is to be plotted in an existing window or subwindow that share the same Y-axis units with the trace being appended.

  When this variable is set to `false`, which is the default value, and if the Y-axis units of the existing trace do not match with the corresponding X-axis units of the trace being appended, a new Y-axis is added to display the .

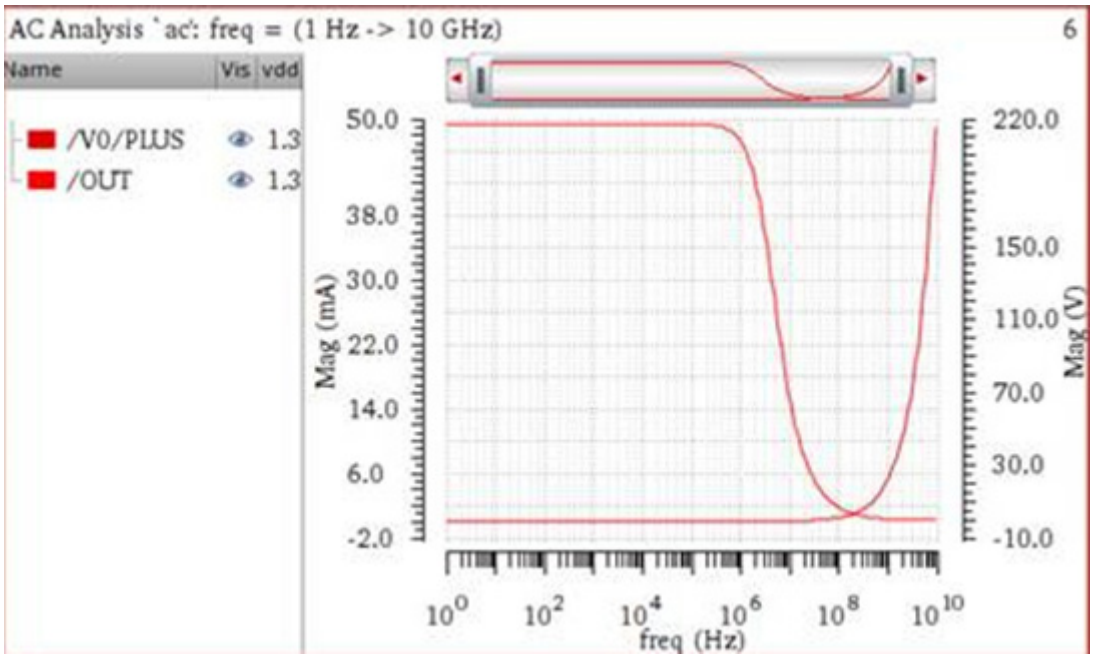
  When this variable is set to `true`, the new trace is appended to the existing graph irrespective of the Y-axis units.

  For example, consider the waveforms for the two signals shown in the figure below, /OUT and /V0/PLUS. These waveforms have same X-axis, but Y-axes are different.



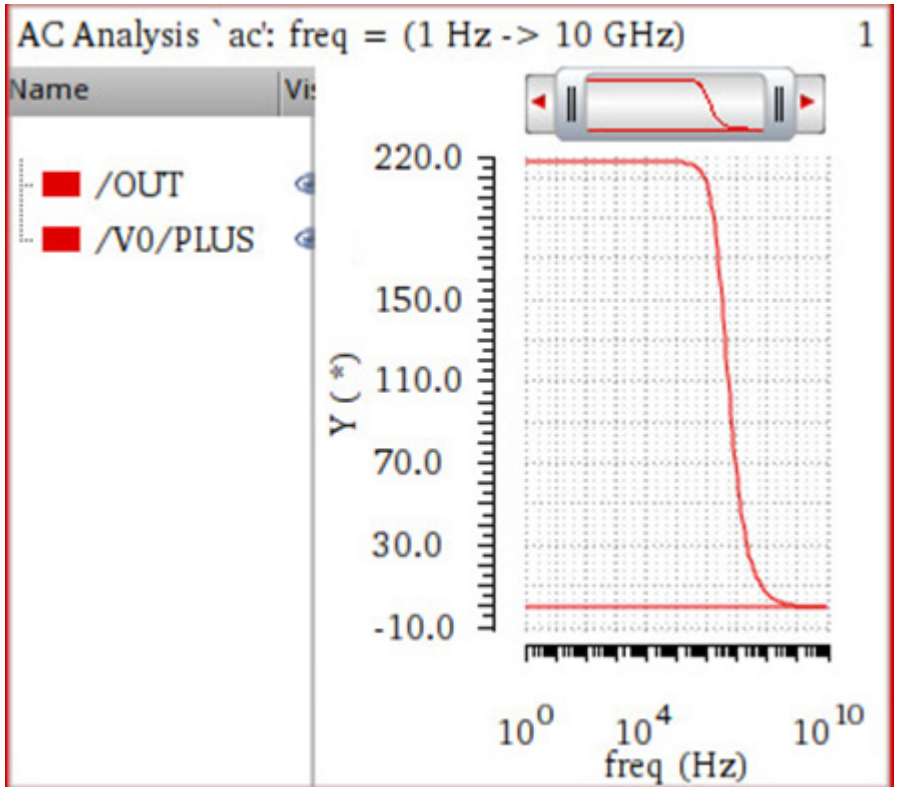
© 2023 Cadence Design Systems, Inc.

If the variable is set to false and you combine the two waveforms, a new Y-axis is added to the right. In the figure below, two Y-axes are shown for Mag (V) and Mag (mA).



© 2023 Cadence Design Systems, Inc.

If the variable is set to true, and you combine the two waveforms, one Y-axis `Y(*)` is used.



- `viva.depAxis labelAngle string "0"`

labelAngle

Specifies the angle through which labels of the dependent axis are rotated.

In `.cdsenv`:

```
viva.depAxis labelAngle string "0"
```

In `.cdsinit` or the CIW:

```
envSetVal("viva.depAxis" "labelAngle" 'string "0")
```

Valid Values: `0–360`

A string value specifying the angle

Default Value: `"0"`

GUI Equivalent: *Label Angle* field on the *Dependent Axis Properties* form.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

Independent Axis Environment Variables

- `viva.indepAxis logScale string "false"`
- `viva.indepAxis forceOrigin string "false"`
- `viva.indepAxis notation string "suffix"`
- `viva.indepAxis showUnits string "true"`

  The `showUnits` variable is used to show or hide the units on the X-axis.

- `viva.indepAxis flexible string "false"`

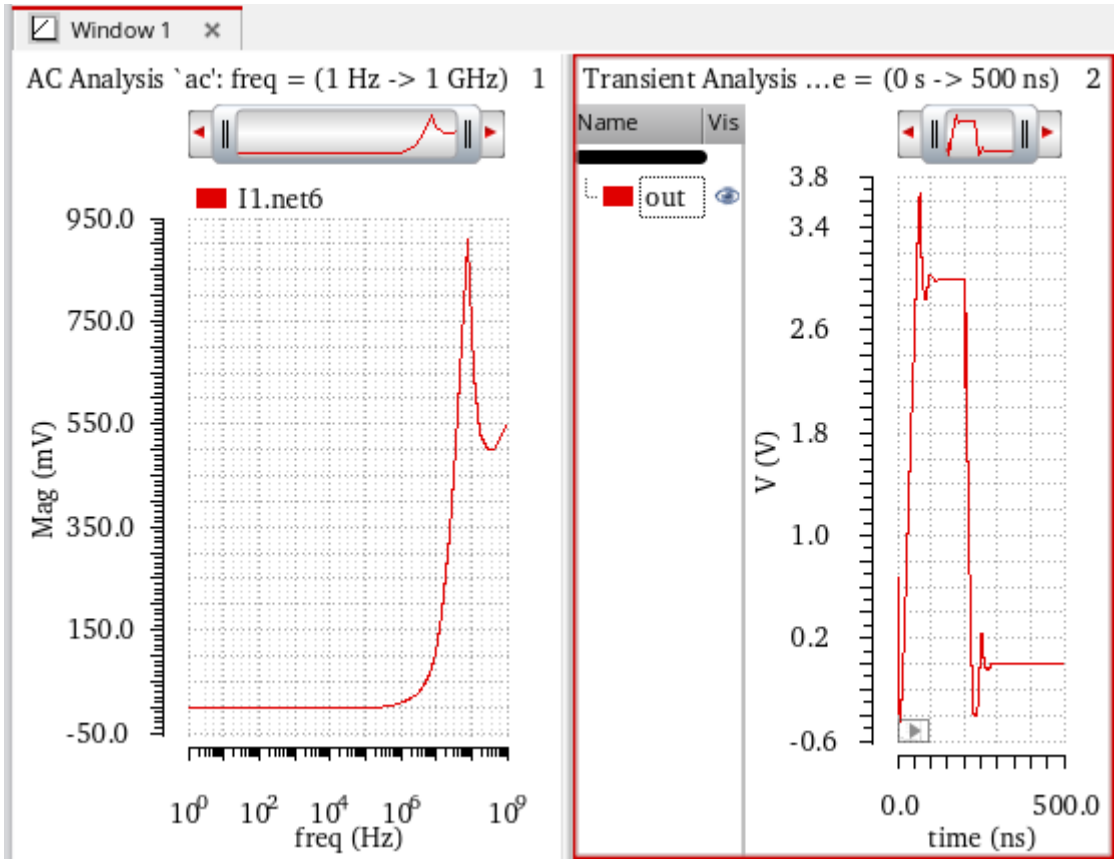
  Specifies whether a new trace is to be plotted in a new window or subwindow or in an existing window or subwindow based on the X-axes units.

  When this variable is set to `false`, which is the default value, and if the X-axis units of the existing trace do not match with the corresponding X-axis units of the trace being appended, the new trace is plotted in a new window. In addition, if you drag trace from one graph to another, an icon appears that prevents the traces to be appended.

  When this variable is set to `true`, the new trace is appended to the existing graph irrespective of the X-axis units.

© 2023 Cadence Design Systems, Inc.

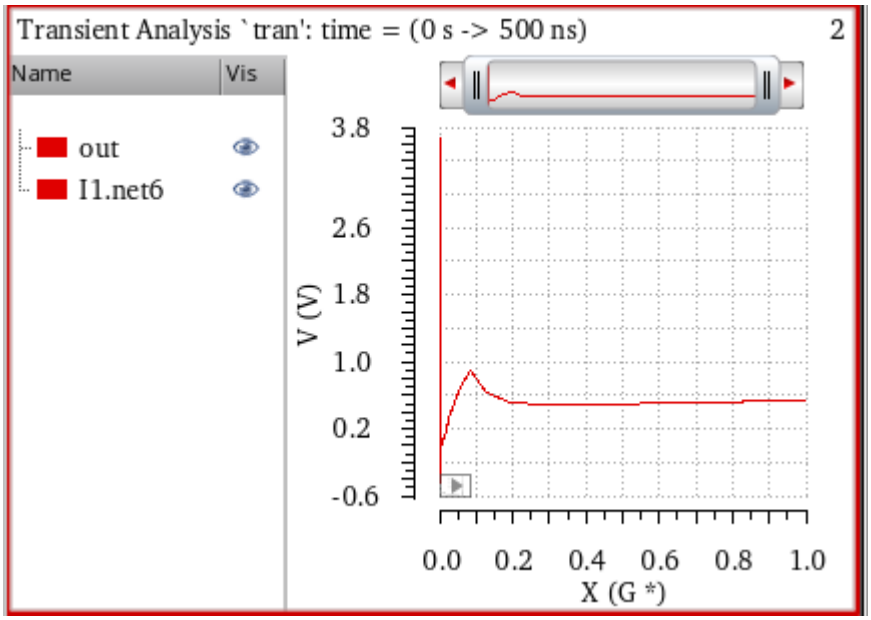
For example, consider the following traces with different X-axis units, `freq(Hz)` and `time(ns)`:



Now, if this variable is set to `false`, these traces will be plotted in different subwindows even if the plotting mode is set to append, and you cannot move them together because their X-axes are not the same.

© 2023 Cadence Design Systems, Inc.

On the other hand, if you set the variable to `true`, these traces will be plotted in one subwindow irrespective of the different X-axes values, given that the plotting mode is append.



Notice that X-axis unit label changes to `X(G*)` indicating that the two X-axes are combined.

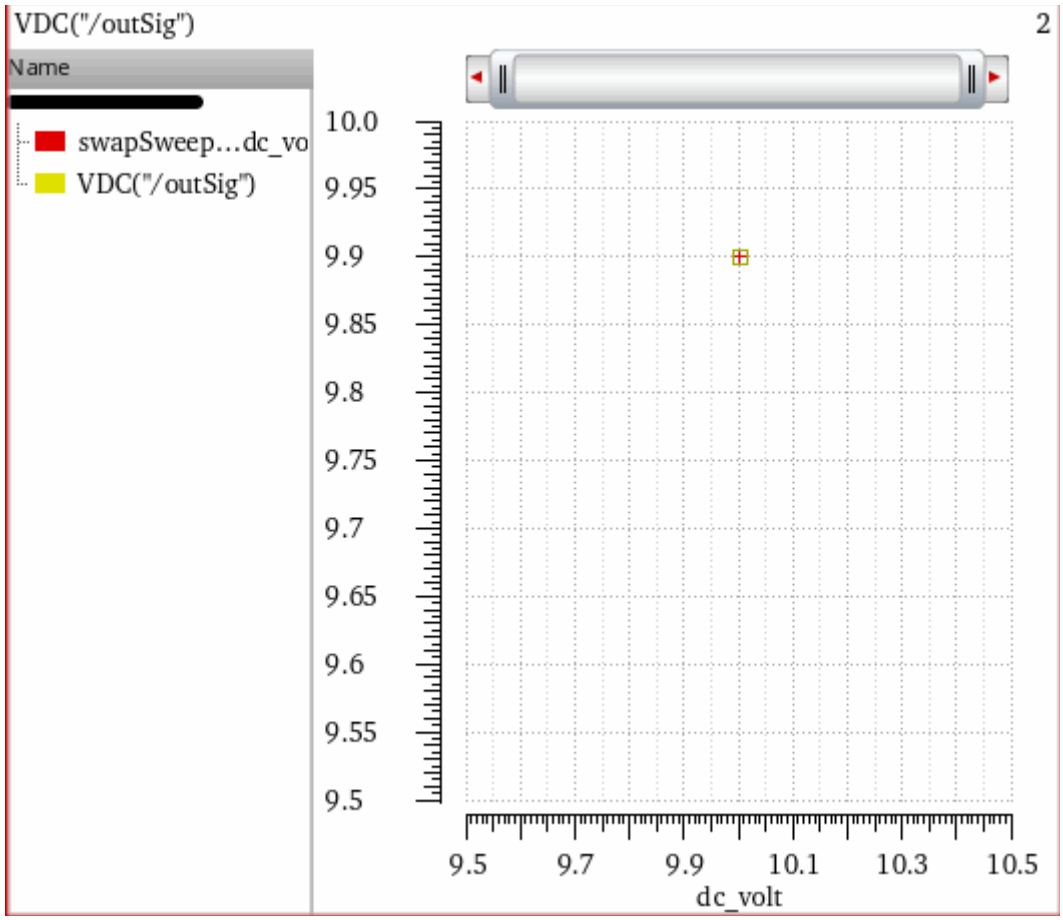
- `viva.indepAxis sameName string "false"`

  When this variable is set to `true`, a new trace is plotted in a new window or subwindow if the X-axis units and name of the active graph do not match with the corresponding X-axis units and name of the graph being appended. When set to `false`, the new trace is appended to the active graph irrespective of the unit names. By default, this variable is set to `false`.

  For example, the figures below show how this environment variable works for signals with different X-axis names, `X_Axis` and `X_Axis_1`, respectively.

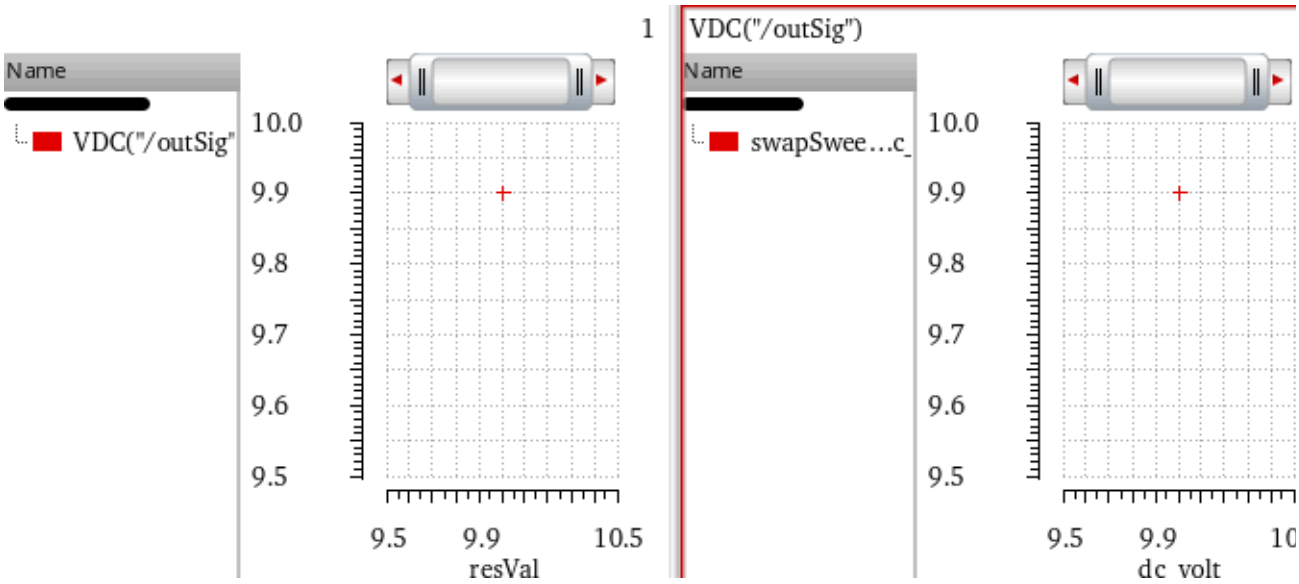
© 2023 Cadence Design Systems, Inc.

When set to `false`, both the signals are plotted in the active window irrespective of the X-axis units and names.



© 2023 Cadence Design Systems, Inc.

When set to `true`, the signals are plotted in different subwindows because the X-axis names are different.



- `viva.indepAxis labelAngle string "0"`

labelAngle

Specifies the angle through which labels of the independent axis are rotated.

In `.cdsenv`:

```
viva.indepAxis labelAngle string "0"
```

In `.cdsinit` or the CIW:

```
envSetVal("viva.indepAxis" "labelAngle" 'string "0")
```

Valid Values: `0–360`

A string value specifying the angle

Default Value: `"0"`

GUI Equivalent: *Label Angle* field on the *Independent Axis Properties* form.

© 2023 Cadence Design Systems, Inc.

String Independent Axis Environment Variables

- `viva.stringIndepAxis traceStyle string "points"`

  This variable controls the default line style for X-axis with strings or corners.

© 2023 Cadence Design Systems, Inc.

Trace Environment Variables

- `viva.trace hiliteColor string "lime"`
- `viva.trace useGlow string "false"`
- `viva.trace lineThickness string "fine"`
- `viva.trace lineStyle string "solid"`
- `viva.trace depModifier string "Magnitude"`
- `viva.trace indepModifier string "Magnitude"`
- `viva.trace symbolsOn string "false"`
- `viva.trace symbolStyle string "plus"`
- `viva.trace toplineOn string "true"`
- `viva.trace midlineOn string "true"`
- `viva.trace baselineOn string "true"`
- `viva.trace dToAHiVoltage string "5.0"`
- `viva.trace dToALoVoltage string "0.0"`
- `viva.trace dToAXVoltage string "(vhi + vlo)/2"`
- `viva.trace dToAUnit string "V"`
- `viva.trace autoReferenceLines string "true"`
- `viva.trace baseAndtoplineReferenceHint string "5.0"`
- `viva.trace threshold string "20_80"`

© 2023 Cadence Design Systems, Inc.

Trace Legend Environment Variables

- showVisColumn
- font
- printSaveImageFont

showVisColumn

Controls the display of the column containing the visibility buttons for each signal plotted on the graph.

Syntax

```
viva.traceLegend showVisColumn string "true"
```

Default Value

`true`

Valid Values

- `true`—Shows the visibility buttons by default.
- `false`—Hides the visibility buttons by default.

font

Sets the font for the trace legend.

Syntax

```
viva.traceLegend font string ""
```

Default Value

The font currently in use.

© 2023 Cadence Design Systems, Inc.

Valid values

```
viva.traceLegend font string "Default,18,-1,5,55,0,0,0,0,0"
```

For more details, see Font String on page 762.

printSaveImageFont

Sets the font for saving and printing the graph image.

Note: This environment variable does not work if you choose the *Render exactly as screen* option in the *Save Image* form.

Syntax

```
viva.traceLegend printSaveImageFont string ""
```

Default Value

The font currently in use.

Valid Values

```
Default,18,-1,5,55,0,0,0,0,0
```

For more details, see Font String on page 762.

showTraceTooltips

Determines whether to display tooltips when you move the mouse pointer over trace names in the trace legend.

Syntax

```
viva.traceLegend showTraceTooltips string "true"
```

Default Value

```
true
```


© 2023 Cadence Design Systems, Inc.

Valid Values

- `true`—Tooltips are displayed when you move mouse pointer over trace names in the trace legend.
- `false`—No tooltips are displayed when you move mouse pointer over trace names in the trace legend.

© 2023 Cadence Design Systems, Inc.

Digital Trace Environment Variables

- `viva.digitalTrace foreground string "green"`

© 2023 Cadence Design Systems, Inc.

Digital Bus Trace Environment Variables

- `viva.digitalBusTrace radix string "hex"`
- `viva.digitalBusTrace foreground string "green"`
- `viva.digitalBusTrace font string "Default,8,-1,5,50,0,0,0,0,0"`

© 2023 Cadence Design Systems, Inc.

font

`viva.digitalBusTrace font string "`*`fontProperties`*`"`

Description

Specifies the font properties of the bus values. The font string lists the following properties:

`Font family`, *`Point size`*, *`Pixel size`*, *`Style hint`*, *`Weight`*, *`Style`*, *`Underline`*, *`Strikeout`*, *`Fixed pitch`*, *`Raw mode`*

The default value is `"Default,8,-1,5,50,0,0,0,0,0"`

The description of these font properties are as follows:

- **Font family**: Specifies the family of fonts. The default value is `Default`, which indicates that the default font family of the system is used.
- **Point size**: Specifies the point size of the font. If the font size is specified in pixels, it must be set to `-1`. Default value is `8`.
- **Pixel size**: Specifies the pixel size of the font. Setting pixel size makes the font device dependent. You can use point size to set the size of the font in a device-independent manner. The default value is `-1`.
- **Style hint**: Specifies the style hints that are used by the font-matching algorithm to find an appropriate default font family if the selected font family is not available in the system.

  Style hints are not supported on X11 since this information is not provided by the windows system.

  The default value is `5`.
- **Weight**: Uses a weighting scale from `0`–`99` similar to the scales used in Windows or CSS. A weight of `0` is ultralight, while `99` is black. The default value is `50`.

  This enumeration contains the following predefined font weights.

Enumeration	Value
`Light`	25
`Normal`	50
`DemiBold`	63
`Bold`	75

© 2023 Cadence Design Systems, Inc.

Enumeration	Value
`Black`	87

- **Style**: Specifies the different styles of glyphs that are used to display text.

  This enumeration contains the following predefined styles.

Enumeration	Value	
`StyleNormal`	`0`	Normal glyphs used in unstyled text. This is the default value.
`StyleItalic`	`1`	Italic glyphs that are designed for representing the italicized text.
`StyleOblique`	`2`	Glyphs with an italic appearance that are typically based on the unstyled glyphs, but are not fine-tuned for representing the italicized text.

- **Underline**: Specifies whether to underline the text in the Smith grid labels. The default value is `0`, which indicates that the text is not underlined. To make the text underlined, set this value to `1`.
- **Strikeout**: Specifies whether to strike through the text in the point marker label. The default value is `0`, which indicates that the text is not strike-through. To make the text strike-through, set this value to `1`.
- **Fixed pitch**: Specifies whether to set a fixed pitch. The default value is `0`, which indicates that the fixed pitch is not set. To set a fixed pitch, set this value to `1`.
- **Raw mode**: This options is applicable only for X11 systems. It specifies whether to enable or disable raw mode. The default value is `0`, which indicates that raw mode is disabled.

  To enable raw mode, set this value to `1`. If raw mode is enabled, the X font with a complete font name that matches with the family name is searched and all other values set for QFont are ignored. If the font name matches multiple fonts, the first font returned by X window system is used.

© 2023 Cadence Design Systems, Inc.

Important

When you set the font-related environment variables in the `.cdsinit` using the `envSetVal` command, you first need to load the viva context before setting the variable. For this, you can add the following command in your `.cdsinit` file:

```
loadContext(strcat(prependInstallPath("etc/context/") "viva.cxt"))
callInitProc("viva")
```

Now, you can use the `envSetVal` command to set the font variable in the `.cdsinit` file.

GUI Equivalent

The *Value Font* field in the Digital Bus Trace Properties for *`nameOfBus`* form.

Examples

```
envGetVal("viva.digitalBusTrace" "font")
envSetVal("viva.digitalBusTrace" "font" 'string "Default,10,-1,5,50,0,1,1,0,0")
```


© 2023 Cadence Design Systems, Inc.

Histogram Environment Variables

- `viva.histogramTrace densityEstimator string "true"`

  Plots a curve that estimates the distribution concentration. By default, this variable is set to `true`.

- `viva.histogramTrace deviationLines string "true"`

  Shows the standard deviation lines in the graph indicating the mean, (mean - standard deviation), and (mean + standard deviation) values. The standard deviation is a sample standard deviation. By default, this variable is set to `true`.

- `viva.histogramTrace yieldLines string "false"`

  Shows the markers associated with the histogram. By default, this variable is set to `false`.

- `viva.histogramTrace maxPoints string "10000"`

  Controls the maximum number of Histogram points that are to be displayed in the plot. By default, this variable is set to `1000`.

- `viva.histogramTrace combinedLegendPosition string "left"`

  Controls the position of the trace legend. The default position when plotting combined histograms is at the left. To move the trace legend inside the graph, specify `inside`. To move the trace legend to be displayed at the top of each strip, specify `above`.

© 2023 Cadence Design Systems, Inc.

Horizontal Marker Environment Variables

This section describes the environment variables for horizontal marker:

- font on page 813
- notation on page 813
- foreground on page 814
- interceptStyle on page 815
- lineStyle on page 815
- maxIntercepts on page 816
- significantDigits on page 816
- sigDigitsMode on page 817
- snapPoint on page 817

font

Specifies the font of the horizontal marker.

Syntax

```
viva.horizMarker font string "Default,10,-1,5,50,0,0,0,0,0"
```

Valid Values

See Font String on page 762.

notation

Specifies the notation of the horizontal marker.

Syntax

```
viva.horizMarker notation string "suffix"
```

Default Value

`Suffix`

© 2023 Cadence Design Systems, Inc.

Valid Values

- `Scientific`
- `Enginnering`

defaultLabel

Specifies the default label of the horizontal marker.

Syntax

```
viva.horizMarker defaultLabel string "%M"
```

Default Value

- `%M`—Name of the horizontal marker

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

foreground

Specifies the foreground color of the horizontal marker.

Syntax

```
viva.vertMarker foreground string "black"
```

Default Value

`black`

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

© 2023 Cadence Design Systems, Inc.

interceptStyle

Specifies the intercept style of the horizontal marker. This variable also controls the visibility of axes intercepts.

Syntax

```
viva.horizMarker interceptStyle string "OnWhenHover"
```

Default Value

`OnWhenHover`	Displays the intercepts when you hover mouse pointer on the horizontal marker.

Valid Values

- `On`: Always displays intercepts on the horizontal marker.
- `Off`: Does not display intercepts on the horizontal marker.

lineStyle

Specifies the line style of the horizontal marker.

Syntax

```
viva.horizMarker lineStyle string "dash"
```

Default Value

`dash`

Valid Values

- `dash`
- `solid`
- `dot`

© 2023 Cadence Design Systems, Inc.

- dashdot
- dashdotdot
- none

maxIntercepts

```
viva.horizMarker maxIntercepts string numberOfMaxIntercepts
```

Description

Specifies the maximum number of intercepts displayed by a horizontal marker. The default value is 20. You can specify any integer value between 0–200.

GUI Equivalent

The *Max Intercepts* field in the Horizontal Marker Properties for *markerName* form.

Examples

```
envGetVal("viva.horizMarker" "maxIntercepts")
envSetVal("viva.horizMarker" "maxIntercepts" 'string "4")
```

significantDigits

Specifies the significant digits for the horizontal marker.

```
viva.horizMarker significantDigits string "4"
```

Default value

4

Valid Values

Any integer value.

© 2023 Cadence Design Systems, Inc.

sigDigitsMode

Sets the mode to be used to specify significant digits. When this variable is set to `Manual`, the settings defined in the `significantDigits` variable is applied. When it is set to `Auto`, the default value for `significantDigits` is used.

```
viva.horizMarker sigDigitsMode string "Auto"
```

Default Value

`Auto`

Valid Values

`Manual`

snapPoint

Specifies the snap data point of the horizontal marker.

Syntax

```
viva.horizMarker snapPoint string "Data Point"
```

Default Value

`Data Point`: Shifts the marker to a specific data point on the curve.

Valid Values

- `Local Maxima:` Shifts the marker to the maxima value (peak) local to the curve.
- `Local Minima:` Shifts the marker to the minima value local to the curve.
- `Local Max or Min:` Shifts the marker to either maxima or minima value local to the curve.
- `Specific Y value:` Shifts the marker to a specific Y-axis value.
- `Specific X value:` Shifts the marker to a specific X-axis value.

  Note: The `Specific X value` option is not available for horizontal markers.

© 2023 Cadence Design Systems, Inc.

- `Global Maxima: S`hifts the marker to the maxima value (peak) global to the curve.
- `Global Minima`: Shifts the marker to the minima value global to the curve.

© 2023 Cadence Design Systems, Inc.

Reference Line Marker Environment Variables

This section describes the environment variables for reference line marker:

- font on page 819
- interceptStyle on page 819
- lineStyle on page 820
- foreground on page 820
- defaultLabel on page 821
- significantDigits on page 821
- sigDigitsMode on page 821

font

Specifies the font of the Reference Line marker.

Syntax

```
viva.referenceLineMarker font string "Default,10,-1,5,50,0,0,0,0,0"
```

Valid Values

See Font String on page 762.

interceptStyle

Specifies the intercept style of the reference line marker.

Syntax

```
viva.referenceLineMarker interceptStyle string "off"
```

Valid Values

- `off`: Turns off the intercepts on the reference line marker. This is the default value.
- `on`: Turns on the intercepts on the reference line marker.

© 2023 Cadence Design Systems, Inc.

lineStyle

Specifies the line style of the reference line marker.

Syntax

```
viva.referenceLineMarker lineStyle string "Dot"
```

Default Value

`Dot`

Valid Values

- `dash`
- `solid`
- `dot`
- `dashdot`
- `dashdotdot`
- `none`

foreground

Specifies the foreground color of the reference line marker.

Syntax

```
viva.referenceLineMarker foreground string "aquamarine"
```

Default Value

`aquamarine`

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

© 2023 Cadence Design Systems, Inc.

defaultLabel

Specifies the default label of the vertical marker.

Syntax

```
viva.referenceLineMarker defaultLabel string "%Y"
```

Default Value

- `%Y`—Y-axis coordinates

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

significantDigits

Specifies the significant digits for the reference line marker.

```
viva.referenceLineMarker significantDigits string "4"
```

Default value

`4`

Valid Values

Any integer value

sigDigitsMode

Sets the mode to be used to specify significant digits. When this variable is set to `Manual`, the settings defined in the `significantDigits` variable is applied. When it is set to `Auto`, the default value for `significantDigits` is used.

```
viva.referenceLineMarker sigDigitsMode string "Auto"
```


© 2023 Cadence Design Systems, Inc.

Default Value

`Auto`

Valid Values

`Manual`

© 2023 Cadence Design Systems, Inc.

Vertical Marker Environment Variables

This section describes the environment variables for vertical marker:

- font on page 823
- notation on page 823
- defaultLabel on page 824
- foreground on page 824
- interceptStyle on page 825
- lineStyle on page 825
- significantDigits on page 826
- sigDigitsMode on page 827
- snapPoint on page 827

font

Specifies the font of the Vertical marker.

Syntax

```
viva.vertMarker font string "Default,10,-1,5,50,0,0,0,0,0"
```

Valid Values

See Font String on page 762.

notation

Specifies the notation of the vertical marker.

Syntax

```
viva.vertMarker notation string "suffix"
```

Default Value

Suffix

© 2023 Cadence Design Systems, Inc.

Valid Values

- `Scientific`
- `Enginnering`

defaultLabel

Specifies the default label of the vertical marker.

Syntax

```
viva.vertMarker defaultLabel string "%M"
```

Default Value

- `%M`—Name of the vertical marker

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

foreground

Specifies the foreground color of the vertical marker.

Syntax

```
viva.vertMarker foreground string "black"
```

Default Value

`black`

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

© 2023 Cadence Design Systems, Inc.

interceptStyle

Specifies the intercept style of the vertical marker. This variable also controls the visibility of axes intercepts.

Syntax

```
viva.vertMarker interceptStyle string "OnWhenHover"
```

Default Value

`OnWhenHover`: Displays the intercepts when you hover mouse pointer on the vertical marker.

Valid Values

- `On` Always displays intercepts on the vertical marker.
- `Off` Does not display intercepts on the vertical marker.

lineStyle

Specifies the line style of the vertical marker.

Syntax

```
viva.vertMarker lineStyle string "dash"
```

Default Value

`dash`

Valid Values

- `dash`
- `solid`
- `dot`
- `dashdot`
- `dashdotdot`

© 2023 Cadence Design Systems, Inc.

- none

maxIntercepts

```
viva.vertMarker maxIntercepts string numberOfMaxIntercepts
```

Description

Specifies the maximum number of intercepts displayed by a vertical marker. The default value is `20`. You can specify any integer value between `0`–`200`.

GUI Equivalent

The *Max Intercepts* field in the Vertical Marker Properties for *`markerName`* form.

Examples

```
envGetVal("viva.vertMarker" "maxIntercepts")
envSetVal("viva.vertMarker" "maxIntercepts" 'string "4")
```

significantDigits

Specifies the significant digits for the vertical marker.

```
viva.vertMarker significantDigits string "4"
```

Default value

`4`

Valid Values

Any integer value

© 2023 Cadence Design Systems, Inc.

sigDigitsMode

Sets the mode to be used to specify significant digits. When this variable is set to `Manual`, the settings defined in the `significantDigits` variable is applied. When it is set to `Auto`, the default value for `significantDigits` is used.

```
viva.vertMarker sigDigitsMode string "Auto"
```

Default Value

`Auto`

Valid Values

`Manual`

snapPoint

Specifies the snap data point of the vertical marker.

Syntax

```
viva.vertMarker snapPoint string "Data Point"
```

Default Value

`Data Point:` Shifts the marker to a specific data point on the curve.

Valid Values

- `Local Maxima:` Shifts the marker to the maxima value (peak) local to the curve.
- `Local Minima:` Shifts the marker to the minima value (peak) local to the curve.
- `Local Max or Min:` Shifts the marker to either maxima or minima value local to the curve.
- `Specific Y value:` Shifts the marker to a specified Y value.
- `Specific X value` Shifts the marker to a specific X-axis value.

  Note: The `Specific X value` option is not available for horizontal markers.

© 2023 Cadence Design Systems, Inc.

- `Global Maxima:` Shifts the marker to the maxima value (peak) global to the curve.
- `Global Minima:` Shifts the marker to the minima value global to the curve.

© 2023 Cadence Design Systems, Inc.

Point Marker Environment Variables

This section describes the environment variables for point marker:

- font on page 829
- notation on page 829
- defaultLabel on page 830
- foreground on page 830
- significantDigits on page 831
- sigDigitsMode on page 831
- snapPoint on page 831
- drawCrossHairs on page 832
- circDefaultLabel on page 833

font

Specifies the font of the point marker.

Syntax

```
viva.pointMarker font string "Default,10,-1,5,50,0,0,0,0,0"
```

Valid Values

See Font String on page 762.

notation

Specifies the notation of the point marker.

Syntax

```
viva.pointMarker notation string "suffix"
```

Default Value

`suffix`

© 2023 Cadence Design Systems, Inc.

Valid Values

- `Engineering`
- `Scientific`

defaultLabel

Specifies the default label of the point marker.

Syntax

```
viva.pointMarker defaultLabel string "%M: %X %Y %D"
```

Default Value

- `%M`—Name of the point marker
- `%X`—X-axis coordinates
- `%Y`—Y-axis coordinates
- `%D`—Absolute path to the results directory from where the signal is plotted

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

foreground

Specifies the foreground color of the point marker.

Syntax

```
viva.pointMarker foreground string "black"
```

Default Value

`black`

© 2023 Cadence Design Systems, Inc.

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

significantDigits

Specifies the significant digits for the point marker.

```
viva.pointMarker significantDigits string "4"
```

Default value

`4`

Valid Values

Any integer value

sigDigitsMode

Sets the mode to be used to specify significant digits. When this variable is set to `Manual`, the settings defined in the `significantDigits` variable is applied. When it is set to `Auto`, the default value for `significantDigits` is used.

```
viva.pointMarker sigDigitsMode string "Auto"
```

Default Value

`Auto`

Valid Values

`Manual`

snapPoint

Specifies the snap point of the point marker.

© 2023 Cadence Design Systems, Inc.

Syntax

```
viva.pointMarker snapPoint string "Data Point"
```

Default Value

`Data Point`Shifts the marker to a specific data point on the curve.

Valid Values

- `Local Maxima:` Shifts the marker to the maxima value (peak) local to the curve.
- `Local Minima`: Shifts the marker to the minima value (peak) local to the curve.
- `Local Max or Min`: Shifts the marker to either maxima or minima value local to the curve.
- `Specific Y value`: Shifts the marker to a specified Y value.
- `Specific X value`: Shifts the marker to a specific X-axis value.

  Note: The `Specific X value` option is not available for horizontal markers.
- `Global Maxima:` Shifts the marker to the maxima value (peak) global to the curve.
- `Global Minima:`Shifts the marker to the minima value global to the curve.

drawCrossHairs

Specifies whether to display the X and Y intercepts for point markers in the graph.

Syntax

```
viva.pointMarker drawCrossHairs string "Dynamic"
```

Default Value

- `Dynamic`—Displays the X and Y intercepts when the point marker is clicked.

Valid Values

- `On`—Shows the X and Y intercepts
- `Off`—Hides the X and Y intercepts

© 2023 Cadence Design Systems, Inc.

circDefaultLabel

Specifies the default label of the point marker on a circular graph.

Syntax

```
viva.pointMarker circDefaultLabel string "%M: %F (%Z)"
```

Default Value

An empty string, which indicates that the information displayed in the label of a point marker depends upon the type of circular grid:

- **Polar grid**: `%M: %F%n%C`

  Name of the point marker, frequency, and real and imaginary Cartesian values are displayed in the label text of a point marker on a circular graph. %n indicates that values are displayed in a new line.

- **Impedance (Z Smith) grid**: `%M: %F%n%Z`

  Name of the point marker, frequency, and impedance values, such as resistance and reactance are displayed in the label text of a point marker on a circular graph.

- **Admittance (Y Smith) grid**: `%M: %F%n%A`

  Name of the point marker, frequency, and admittance values, such as conductance and susceptance are displayed in the label text of a point marker on a circular graph.

- **Immittance (ZY Smith) grid**: `%M: %F%n%Z%n%A`

  Name of the point marker, frequency, impedance, and admittance values are displayed in the label text of a point marker on a circular graph.

Valid Values

For information about all the valid values, see Adding a Point Marker on page 501.

© 2023 Cadence Design Systems, Inc.

Reference Point Marker Environment Variables

This section describes the environment variables for reference point marker:

- font on page 834
- notation on page 834
- foreground on page 835
- defaultLabel on page 835
- circDefaultLabel on page 836
- drawCrossHairs on page 836
- significantDigits on page 836
- sigDigitsMode on page 837

font

Specifies the font of the reference point marker.

Syntax

```
viva.refPointMarker font string "Default,10,-1,5,50,0,0,0,0,0"
```

Valid Values

See Font String on page 762.

notation

Specifies the notation of the reference point marker.

Syntax

```
viva.refPointMarker notation string "suffix"
```

Default Value

```
suffix
```


© 2023 Cadence Design Systems, Inc.

Valid Values

- `Engineering`
- `Scientific`

foreground

Specifies the foreground color of the reference point marker.

Syntax

```
viva.refPointMarker foreground string "black"
```

Default Value

`black`

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

defaultLabel

Specifies the default label of the reference point marker.

Syntax

```
viva.refPointMarker defaultLabel string "%X %Y"
```

Default Value

- `%X`—X-axis coordinates
- `%Y`—Y-axis coordinates

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

© 2023 Cadence Design Systems, Inc.

circDefaultLabel

Specifies the circular default label of the reference point marker.

Syntax

```
viva.refPointMarker circDefaultLabel string "%F (%Z)"
```

Default Value

`%F (%Z)`

- `%F`—Frequency
- `%Z`— Impedance values for resistance and reactance.

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

drawCrossHairs

Specifies whether to display the X and Y intercepts for reference point markers in the graph.

Syntax

```
viva.refPointMarker drawCrossHairs string "Dynamic"
```

Default Value

- `Dynamic`—Displays the X and Y intercepts when the reference point marker is clicked.

Valid Values

- `On`—Shows the X and Y intercepts
- `Off`—Hides the X and Y intercepts

significantDigits

Specifies the significant digits for the reference point marker.

© 2023 Cadence Design Systems, Inc.

viva.refPointMarker significantDigits string "4"

Default value

4

Valid Values

Any integer value

sigDigitsMode

Sets the mode to be used to specify significant digits. When this variable is set to `Manual`, the settings defined in the `significantDigits` variable is applied. When it is set to `Auto`, the default value for `significantDigits` is used.

viva.refPointMarker sigDigitsMode string "Auto"

Default Value

`Auto`

Valid Values

`Manual`

© 2023 Cadence Design Systems, Inc.

Specification Marker Environment Variables

This section describes the environment variables for specification marker:

- font on page 838
- passcolor on page 838
- failcolor on page 839
- lineStyle on page 839
- lineThickness on page 840
- lineColor on page 840
- displayMode on page 841
- significantDigits on page 841
- showLabel on page 842
- rule on page 842
- type on page 843
- visibility Mode on page 843

font

Specifies the font of the spec marker.

Syntax

```
viva.specMarker font string "Default,10,-1,5,50,0,0,0,0,0"
```

Valid Values

See Font String on page 762.

passcolor

Specifies the color of the pass region of the spec marker.

Syntax

```
viva.specMarker passcolor string "#00CC00"
```


© 2023 Cadence Design Systems, Inc.

Default Value

#00CC00

Valid Values

All values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

Note: Color values are specified in the #RRGGBB format. For example, the cyan color in RGB format is (0,255, 255) and in hexadecimal format, it is written as #00FFFF.

failcolor

Specifies the color of the fail region of the spec marker.

Syntax

```
viva.specMarker failcolor string "#CC0000"
```

Default Value

#CC0000

Valid Values

All values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

Color values are specified in the #RRGGBB format. For example, the cyan color in RGB format is (0,255, 255) and in hexadecimal format, it is written as #00FFFF.

lineStyle

Specifies the line style of the spec marker.

Syntax

```
viva.specMarker lineStyle string "solid"
```


© 2023 Cadence Design Systems, Inc.

Default Value

`solid`

Valid Values

- `dot`
- `dash`
- `dashDot`
- `dashDotDot`
- `none`

lineThickness

Specifies the line thickness of the spec marker.

Syntax

```
viva.specMarker lineThickness string "medium"
```

Default Value

`medium`

Valid Values

- `fine`
- `thick`
- `extraThick`

lineColor

Specifies the line color of the spec marker.

Syntax

```
viva.specMarker lineColor string "white"
```


© 2023 Cadence Design Systems, Inc.

Default Value

`white`

Valid Values

- `black`
- `red`
- `blue`
- `green`
- `yellow`

displayMode

Specifies the spec marker region display mode of the spec marker.

Syntax

`viva.specMarker displayMode string "both"`

Default Value

`both`

Valid Values

- none—Displays no spec marker on the graph
- `pass`—Displays only pass spec marker region on the graph
- `fail`—Displays only fail spec marker region on the graph
- `thresholdOnly`—Displays threshold region on the graph

significantDigits

Specifies the significant digits for the specification marker.

`viva.specMarker significantDigits string "4"`

© 2023 Cadence Design Systems, Inc.

Default value

`4`

Valid Values

Any integer value

showLabel

Displays the spec marker labels on the graph.

Syntax

```
viva.specMarker showLabel string "true"
```

Default Value

`true`

Valid Values

- `true`—Shows the spec marker labels.
- `false`—Hides the spec marker labels.

rule

Sets the specifictaion type to be used for the spec marker.

Syntax

```
viva.specMarker rule string "range"
```

Default Value

`range`

Valid Values

- `none`
- `minimize`

© 2023 Cadence Design Systems, Inc.

- `maximize`
- `<`
- `>`
- `tol`
- `info`

type

Specifies whether the specification type of spec waveform is drawn normally or in a sampleHold style.

Syntax

```
viva.specMarker type string "line"
```

Default Value

`line`

Valid Values

- `line`
- `stairStep`

visibility Mode

Specified whether to display specification markers on the graph.

Syntax

```
viva.specMarker visibilityMode string "OnWhenSelected"
```

Default Value

`OnWhenSelected`

© 2023 Cadence Design Systems, Inc.

Valid Values

- `On`
- `Off`
- `OnWhenSelected`

© 2023 Cadence Design Systems, Inc.

Intercept Marker Environment Variables

This section describes the environment variables for Intercept marker:

- foreground on page 845
- lineStyle on page 845

foreground

Specifies the foreground color of the intercept marker.

Syntax

```
viva.interceptMarker foreground string "black"
```

Default Value

```
black
```

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

lineStyle

Specifies the line style of the intercept edge marker.

Syntax

```
viva.interceptMarker lineStyle string "solid"
```

Default Value

```
solid
```


© 2023 Cadence Design Systems, Inc.

Valid Values

- `dot`
- `dash`
- `dashDot`
- `dashDotDot`
- `none`

© 2023 Cadence Design Systems, Inc.

Circle Marker Environment Variables

This section describes the environment variables for circular marker:

- font on page 847
- notation on page 847
- defaultLabel on page 848
- foreground on page 848
- background on page 849

font

Specifies the font of the circular marker.

Syntax

```
viva.circleMarker font string "Default,10,-1,5,50,0,0,0,0,0"
```

Valid Values

See Font String on page 762.

notation

Specifies the notation of the circular marker.

Syntax

```
viva.circleMarker notation string "suffix"
```

Default Value

```
suffix
```

Valid Values

```
Engineering
```


© 2023 Cadence Design Systems, Inc.

defaultLabel

Specifies the mnemonic label for the circular markers.

Syntax

```
viva.circleMarker defaultLabel string "%M: %X %Y"
```

Default Value

- `%M`—Name of the marker
- `%X`—X-axis coordinates
- `%Y`—Y-axis coordinates

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

foreground

Specifies the foreground color of the circular marker.

Syntax

```
viva.cicleMarker foreground string "red"
```

Default Value

`red`

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

© 2023 Cadence Design Systems, Inc.

background

Specifies the background color of the circular marker.

Syntax

```
viva.circleMarker background string "#8e8a8e"
```

Default Values

`#8e8a8e`

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

© 2023 Cadence Design Systems, Inc.

Delta Marker Environment Variables

This section describes the environment variables for delta markers:

- foreground on page 850
- notation on page 850
- showChildLabels on page 851
- defaultLabel on page 851
- significantDigits on page 853
- sigDigitsMode on page 853
- deltaMeasure on page 853

foreground

Specifies the foreground color of the delta marker.

Syntax

```
viva.multiDeltaMarker foreground string "black"
```

Default Value

```
black
```

Valid Values

All the values defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

notation

Specifies the notation of the delta marker.

Syntax

```
viva.multiDeltaMarker notation string "suffix"
```


© 2023 Cadence Design Systems, Inc.

Default Value

`suffix`

Valid Values

- `Engineering`
- `Scientific`

showChildLabels

Shows or hides marker labels for the delta markers.

Syntax

```
viva.multiDeltaMarker showChildLabels string "true"
```

Values

- `true`—Shows the labels for all markers that combine to form the delta marker.
- `false`—Hides the labels for all markers that combine to form the delta marker.

defaultLabel

Specifies the mnemonic label for the delta markers.

Syntax

```
viva.multiDeltaMarker defaultLabel string "[dx:%W] [dy:%H] [s:%S]"
```

Default Value

`[dx:%W] [dy:%H] [s:%S]`

where,

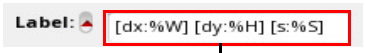
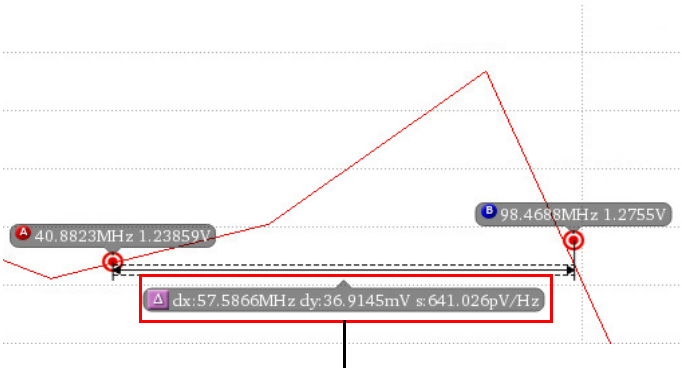
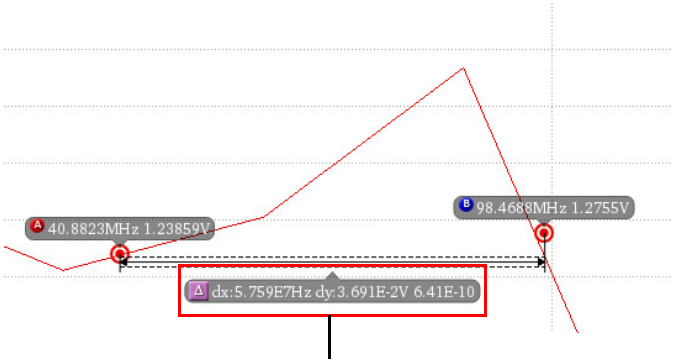
- `dx:%W`—Delta value on X-axis
- `dy:%H`—Delta value on Y-axis

© 2023 Cadence Design Systems, Inc.

- `s:%S`—Slope (dy/dx)

  Note: If you do not enclose these labels in parenthesis `[ ]`, the corresponding information is displayed without suffix notations in the marker label.

  The following table shows how values are displayed in the marker labels of a delta marker when format strings are specified with and without parenthesis, `[]`.

Format String	Values in the Marker Label
 Format strings with `[]`	 Values in the marker label are displayed with suffix notation.
 Format strings without `[]`	 Values in the marker label are displayed without suffix notation.

For more information about the values that you can use for marker labels, see Marker Labels.

© 2023 Cadence Design Systems, Inc.

Valid Values

For information about valid values, see Adding a Point Marker on page 501.

significantDigits

Specifies the significant digits for the delta marker.

```
viva.multiDeltaMarker significantDigits string "4"
```

Default value

`4`

Valid Values

Any integer value

sigDigitsMode

Sets the mode to be used to specify significant digits. When this variable is set to `Manual`, the settings defined in the `significantDigits` variable is applied. When it is set to `Auto`, the default value for `significantDigits` is used.

```
viva.multiDeltaMarker sigDigitsMode string "Auto"
```

Default Value

`Auto`

Valid Values

`Manual`

deltaMeasure

Determines the axis for which the delta measurement values are to be displayed in the delta marker label. Valid values of this variable include the change in X-axis value, Y-axis value, or both X- and Y-axes values. You can also use the delta marker properties form to control the display of these measurement values.

© 2023 Cadence Design Systems, Inc.

In addition, Virtuoso Visualization and Analysis XL controls the display of these measurement values depending on the markers between which the delta marker is created. For example, if a delta marker is created between two or more vertical markers, then only the change in X-axis value is displayed.

Syntax

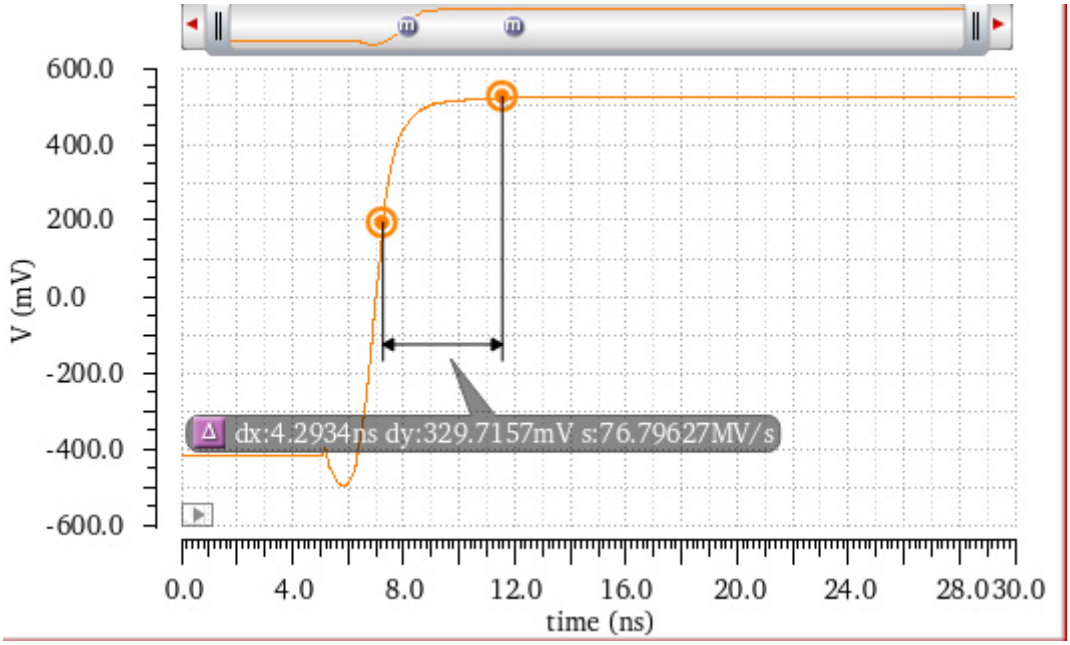
```
viva.multiDeltaMarker deltaMeasure string "X Only"
```

Default Value

`X Only`

Valid Values

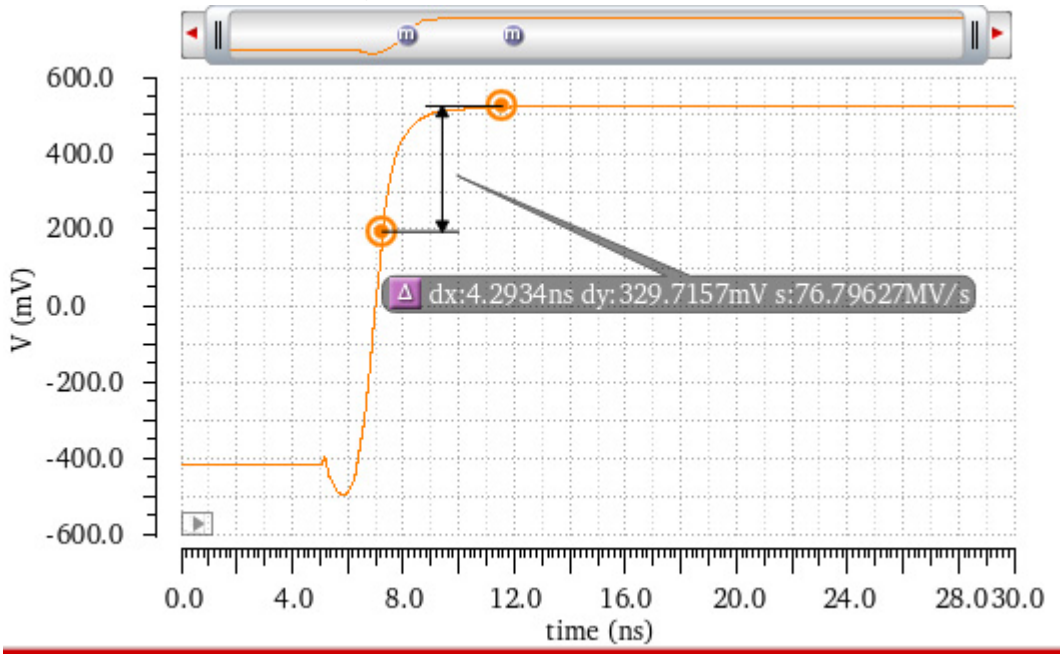
- `X Only`—Shows the change in X-axis only.



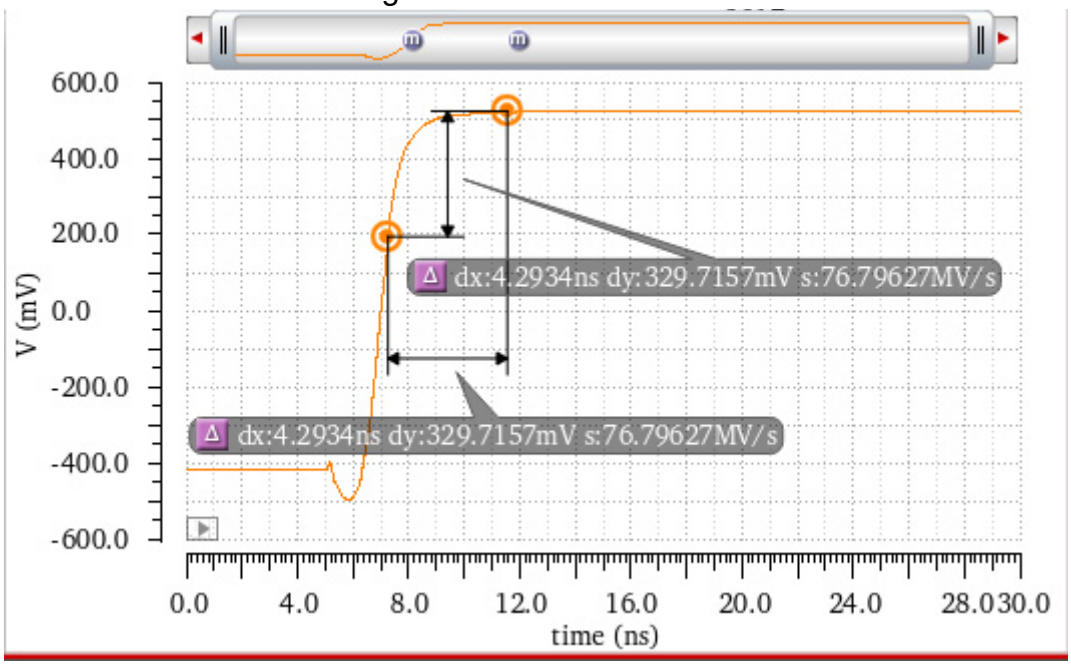
Note: Any invalid value, if specified, is ignored and the default value, `X Only`, is used.

© 2023 Cadence Design Systems, Inc.

- `Y Only`—Shows the change in Y-axis only.



- `X and Y`—Shows the change in both X- and Y-axes.



© 2023 Cadence Design Systems, Inc.

Transient Edge Markers Environment Variables

This section describes the environment variables for transient edge markers:

- defaultLabel
- font
- foreground
- lineStyle
- notation
- showMeasSummary
- significantDigits
- snapPoint
- threshold
-

© 2023 Cadence Design Systems, Inc.

defaultLabel

`viva.transEdgeMarker defaultLabel string` *`defaultLabel`*

Description

Specifies the default label for the transient edge marker. The default value is `"%M"`, which indicates that the default label for the transient edge marker is the shorthand name of the edge marker.

The shorthand name of an edge marker is `E:` *`edgeNumber`*. For example, if you apply a transient marker on the first edge of a trace, the default label is `E:1`. If you apply a transient marker on the fourth edge of a trace, the default label is `E:4`.

GUI Equivalent

None

Examples

```
envGetVal("viva.transEdgeMarker" "defaultLabel")
envSetVal("viva.transEdgeMarker" "defaultLabel" 'string "%M")
```


© 2023 Cadence Design Systems, Inc.

significantDigits

```
viva.transEdgeMarker significantDigits string numberOfSignificantDigits
```

Description

Specifies the number of significant digits for the transient edge marker. The default value is `"4"`.

GUI Equivalent

The *Significant Digits* field on the Transient Edge Marker Properties for Edge *edgeNumber* form.

Examples

```
envGetVal("viva.transEdgeMarker" "significantDigits")
envSetVal("viva.transEdgeMarker" "significantDigits" 'string "3")
```


© 2023 Cadence Design Systems, Inc.

font

```
viva.transEdgeMarker font string fontProperties
```

Description

Specifies the font properties of the transient edge marker. The default value is `"Default,10,-1,5,50,0,0,0,0,0"`.

For more information about the format of the font properties, see Font String.

GUI Equivalent

The *Font/Color* field on the Transient Edge Marker Properties for Edge *edgeNumber* form.

Examples

```
envGetVal("viva.transEdgeMarker" "font")
envSetVal("viva.transEdgeMarker" "font" 'string "Default,20,-1,5,50,0,0,0,0,0")
```


© 2023 Cadence Design Systems, Inc.

notation

```
viva.transEdgeMarker notation string {"suffix" | "engineering" | "scientific"}
```

Description

Specifies the notation to be used in the transient edge marker. The default value is `"suffix"`.

Possible values are: `suffix`, `engineering`, and `scientific`.

GUI Equivalent

The *Notation* field on the Transient Edge Marker Properties for Edge *`edgeNumber`* form.

Examples

```
envGetVal("viva.transEdgeMarker" "notation")
envSetVal("viva.transEdgeMarker" "notation" 'string "scientific")
```


© 2023 Cadence Design Systems, Inc.

threshold

```
viva.transEdgeMarker threshold string {"10_90" | "20_80" | "30_70"}
```

Description

Specifies the threshold levels of signal where the transient edge marker is applied on the rising or the falling edge of the signal. The default value is `"20_80"`.

GUI Equivalent

- The *Threshold* field on the Transient Edge Marker Properties for Edge *edgeNumber* form.
- The *Threshold* command when you right-click a transient edge marker.

Examples

```
envGetVal("viva.transEdgeMarker" "threshold")
envSetVal("viva.transEdgeMarker" "threshold" 'string "10_90")
envSetVal("viva.transEdgeMarker" "threshold" 'string "30_70")
```


© 2023 Cadence Design Systems, Inc.

foreground

`viva.transEdgeMarker foreground string` *`foregroundColor`*

Description

Specifies the foreground color of the transient edge marker. The default value is `"black"`.

GUI Equivalent

The *Foreground* field on the Transient Edge Marker Properties for Edge *`edgeNumber`* form.

Examples

```
envGetVal("viva.transEdgeMarker" "foreground")
envSetVal("viva.transEdgeMarker" "foreground" 'string "red")
```


© 2023 Cadence Design Systems, Inc.

lineStyle

```
viva.transEdgeMarker lineStyle string {"dash" | "solid" | "dot" | "dashdot" |
     "dashdotdot"}
```

Description

Specifies the line style of the transient edge marker. The default value is `"solid"`.

Possible values are: `dash`, `solid`, `dot`, `dashdot`, and `dashdotdot`.

GUI Equivalent

None

Examples

```
envGetVal("viva.transEdgeMarker" "lineStyle")
envSetVal("viva.transEdgeMarker" "lineStyle" 'string "dashdotdot")
```


© 2023 Cadence Design Systems, Inc.

showMeasSummary

```
viva.transEdgeMarker showMeasSummary string {"on when hover" | "on" | "off"}
```

Description

Specifies when to show the measurement labels on the transient edge marker.

Possible values are:

- `on when hover`: (Default value) Measurement labels are shown only when you hover over the transient edge marker.
- `on`: Measurement labels are always shown.
- `off`: Measurement labels are always hidden.

GUI Equivalent

- The *Annotation* field on the Transient Edge Marker Properties for Edge *`edgeNumber`* form.
- The *Show Measurement* command when you right-click a transient edge marker.

Examples

```
envGetVal("viva.transEdgeMarker" "showMeasSummary")
envSetVal("viva.transEdgeMarker" "showMeasSummary" 'string "on")
```


© 2023 Cadence Design Systems, Inc.

snapPoint

`viva.transEdgeMarker snapPoint string` *`snapPoint`*

Description

Specifies the snap point of the transient edge marker. The default value is `"Data Point"`.

GUI Equivalent

None

Examples

```
envGetVal("viva.transEdgeMarker" "snapPoint")
envSetVal("viva.transEdgeMarker" "snapPoint" 'string "Data Point")
```


© 2023 Cadence Design Systems, Inc.

Graph Label Environment Variables

- `viva.graphLabel font string "Default,10,-1,5,50,0,0,0,0,0"`
- `viva.graphLabel foreground string "white"`
- `viva.graphLabel background string "lightGray"`

© 2023 Cadence Design Systems, Inc.

Probe Environment Variables

- `viva.probe font string Default,10,-1,5,50,0,0,0,0,0"`
- `viva.probe foreground string "black"`
- `viva.probe background string "white"`
- `viva.probe autoTopBaseline string "true"`
- `viva.probe topLine string "0.0"`
- `viva.probe baseLine string "0.0"`
- `viva.probe autoMinMax string "true"`
- `viva.probe minValue string "0.0"`
- `viva.probe maxValue string "0.0"`

© 2023 Cadence Design Systems, Inc.

Polar Grid Environment Variables

- `viva.polarGrid font string "Default,10,-1,5,50,0,0,0,0,0"`
- `viva.polarGrid numCircles string "4"`
- `viva.polarGrid circlesForeground string "gray"`
- `viva.polarGrid circlesOn string "true"`
- `viva.polarGrid circlesLabelsOn string "true"`
- `viva.polarGrid numRadials string "2"`
- `viva.polarGrid radialsOn string "true"`
- `viva.polarGrid radialsLabelsOn string "true"`

© 2023 Cadence Design Systems, Inc.

Smith Grid Environment Variables

The Smith grid environment variables are used to define properties of the Smith grid. The Smith grid environment variables are identified by the partition name `viva.smithGrid`.

© 2023 Cadence Design Systems, Inc.

font

```
viva.smithGrid font string "fontProperties"
```

Description

Specifies the font properties for the labels on the Smith grid in a string format. The font string lists the following properties:

Font family, *Point size*, *Pixel size*, *Style hint*, *Weight*, *Style*, *Underline*, *Strikeout*, *Fixed pitch*, *Raw mode*

The description of these font properties are as follows:

- **Font family**: Specifies the family of fonts. The default value is `Default`, which indicates that the default font family of the system is used.
- **Point size**: Specifies the point size of the font. If the font size is specified in pixels, it must be set to `-1`.
- **Pixel size**: Specifies the pixel size of the font. Setting pixel size makes the font device dependent. You can use point size to set the size of the font in a device-independent manner.
- **Style hint**: Specifies the style hints that are used by the font-matching algorithm to find an appropriate default font family if the selected font family is not available in the system.

  Style hints are not supported on X11 since this information is not provided by the windows system.
- **Weight**: Uses a weighting scale from `0`–`99` similar to the scales used in Windows or CSS. A weight of `0` is ultralight, while `99` is black.

  This enumeration contains the following predefined font weights.

Enumeration	Value
`Light`	25
`Normal`	50
`DemiBold`	63
`Bold`	75
`Black`	87

- **Style**: Specifies the different styles of glyphs that are used to display text.

© 2023 Cadence Design Systems, Inc.

This enumeration contains the following predefined styles.

Enumeration	Value	
`StyleNormal`	`0`	Normal glyphs used in unstyled text.
`StyleItalic`	`1`	Italic glyphs that are designed for representing the italicized text.
`StyleOblique`	`2`	Glyphs with an italic appearance that are typically based on the unstyled glyphs, but are not fine-tuned for representing the italicized text.

- **Underline**: Specifies whether to underline the text in the Smith grid labels. The default value is `0`, which indicates that the text is not underlined. To make the text underlined, set this value to `1`.
- **Strikeout**: Specifies whether to strike through the text in the point marker label. The default value is `0`, which indicates that the text is not strike-through. To make the text strike-through, set this value to `1`.
- **Fixed pitch**: Specifies whether to set a fixed pitch. The default value is `0`, which indicates that the fixed pitch is not set. To set a fixed pitch, set this value to `1`.
- **Raw mode**: This options is applicable only for X11 systems. It specifies whether to enable or disable raw mode. The default value is `0`, which indicates that raw mode is disabled.

  To enable raw mode, set this value to `1`. If raw mode is enabled, the X font with a complete font name that matches with the family name is searched and all other values set for QFont are ignored. If the font name matches multiple fonts, the first font returned by X window system is used.

Important

When you set the font-related environment variables in the `.cdsinit` using the `envSetVal` command, you first need to load the viva context before setting the variable. For this, you can add the following command in your `.cdsinit` file:

```
loadContext(strcat(prependInstallPath("etc/context/") "viva.cxt"))
callInitProc("viva")
```

Now, you can use the `envSetVal` command to set the font variable in the `.cdsinit` file.

© 2023 Cadence Design Systems, Inc.

GUI Equivalent

The *Font* field in the *Labels* section on the *Grid* tab of the Graph Properties form for the circular graphs. This field is not available for Polar grids.

Examples

```
envGetVal("viva.smithGrid" "font")
envSetVal("viva.smithGrid" "font" 'string "Default,10,-1,5,50,0,1,1,0,0")
```


© 2023 Cadence Design Systems, Inc.

autoSizeLabels

```
viva.smithGrid autoSizeLabels string {"true" | "false"}
```

Description

Specifies whether to automatically adjust the font size for the Smith grid labels.

Valid values are:

- `true`: (Default) Automatically adjusts the font size for the Smith grid labels.
- `false`: Uses the font size specified the font environment variable.

GUI Equivalent

The *Text Size* field in the *Labels* section on the *Grid* tab of the Graph Properties form for the circular graphs. This field is not available for Polar grids.

Examples

```
envGetVal("viva.smithGrid" "autoSizeLabels")
envSetVal("viva.smithGrid" "autoSizeLabels" 'string "false")
```


© 2023 Cadence Design Systems, Inc.

showDetailed

```
viva.smithGrid showDetailed string {"auto" | "off" | "on"}
```

Description

Specifies when to display detailed grid lines on the Smith chart. Possible values are:

- `auto`: (Default) Detailed grid lines are automatically displayed when zooming in on the Smith chart.
- `off`: Detailed grid lines are always hidden.
- `on`: Detailed grid lines are always displayed.

GUI Equivalent

The *Detailed Grid* field in the *Display* section on the *Grid* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "showDetailed")
envSetVal("viva.smithGrid" "showDetailed" 'string "off")
```


© 2023 Cadence Design Systems, Inc.

showLabels

```
viva.smithGrid showLabels string {"true" | "false"}
```

Description

Specifies whether to display labels on the Smith grid. The default value is `true`, which indicates that labels are displayed on the Smith grid.

GUI Equivalent

The *Grid Labels* field in the *Display* section on the *Grid* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "showLabels")
envSetVal("viva.smithGrid" "showLabels" 'string "false")
```


© 2023 Cadence Design Systems, Inc.

showCompressed

```
viva.smithGrid showCompressed string {"true" | "false"}
```

Description

Specifies whether to display extra horizontal grid lines (arcs) outside the Smith chart boundary. The default value is `false`, which indicates that grid lines outside the Smith unit circle are not displayed.

GUI Equivalent

The *Compressed Smith* field in the *Display* section on the *Grid* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "showCompressed")
envSetVal("viva.smithGrid" "showCompressed" 'string "true")
```


© 2023 Cadence Design Systems, Inc.

showCoefficientRing

```
viva.smithGrid showCoefficientRing string {"true" | "false"}
```

Description

Specifies whether to display the angles of reflection coefficient in the outer ring. The default value is `false`, which indicates that reflection coefficient and outer ring are not displayed on the Smith grid.

GUI Equivalent

The *Coefficient Ring* field in the *Display* section on the *Grid* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "showCoefficientRing")
envSetVal("viva.smithGrid" "showCoefficientRing" 'string "true")
```


© 2023 Cadence Design Systems, Inc.

showVSWRCircles

```
viva.smithGrid showVSWRCircles string {"true" | "false"}
```

Description

Specifies whether to display VSWR circles for the specified VSWR values. The default value is `false`, which indicates that VSWR circles are not displayed on the Smith grid.

GUI Equivalent

The *VSWR Circles* field in the *Smith Chart* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "showVSWRCircles")
envSetVal("viva.smithGrid" "showVSWRCircles" 'string "true")
```


© 2023 Cadence Design Systems, Inc.

VSWRCircleValues

```
viva.smithGrid VSWRCircleValues string VSWRvalues
```

Description

Specifies the VSWR values for which you want to draw the VSWR circles on the Smith grid.

The default value is `""`, which indicates that no VSWR circles are drawn on the Smith grid.

VSWR circles for the specified VSWR values are drawn only when the `showVSWRCircles` environment variable is set to `true`.

GUI Equivalent

The *VSWR* field in the *Smith Chart* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "VSWRCircleValues")
envSetVal("viva.smithGrid" "VSWRCircleValues" 'string "6, 8")
```


© 2023 Cadence Design Systems, Inc.

VSWRCirclePenColor

`viva.smithGrid VSWRCirclePenColor string` *`hexColorCode`*

Description

Specifies the color of the VSWR circles.

The default value is `#C0C000`.

VSWR circles for the specified VSWR values are drawn only when the `showVSWRCircles` environment variable is set to `true`.

GUI Equivalent

The *Color* field in the *VSWR Circles* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "VSWRCirclePenColor")
envSetVal("viva.smithGrid" "VSWRCirclePenColor" 'string "#0000FF")
```


© 2023 Cadence Design Systems, Inc.

VSWRCirclePenWidth

`viva.smithGrid VSWRCirclePenWidth string` *`widthOfVSWRCircles`*

Description

Specifies the width of the VSWR circles.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `fine`.

VSWR circles for the specified VSWR values are drawn only when the `showVSWRCircles` environment variable is set to `true`.

GUI Equivalent

The *Width* field in the *VSWR Circles* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "VSWRCirclePenWidth")
envSetVal("viva.smithGrid" "VSWRCirclePenWidth" 'string "extraThick")
```


© 2023 Cadence Design Systems, Inc.

VSWRCirclePenStyle

`viva.smithGrid VSWRCirclePenStyle string` *`widthOfVSWRCircles`*

Description

Specifies the style of the VSWR circles.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `solid`.

VSWR circles for the specified VSWR values are drawn only when the `showVSWRCircles` environment variable is set to `true`.

GUI Equivalent

The *Style* field in the *VSWR Circles* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "VSWRCirclePenStyle")
envSetVal("viva.smithGrid" "VSWRCirclePenStyle" 'string "dashDotDot")
```


© 2023 Cadence Design Systems, Inc.

showQContours

```
viva.smithGrid showQContours string {"true" | "false"}
```

Description

Specifies whether to display Q contours for the specified Q values. The default value is `false`, which indicates that Q contours are not displayed on the Smith grid.

GUI Equivalent

The *Q Contours* field in the *Smith Chart* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "showQContours")
envSetVal("viva.smithGrid" "showQContours" 'string "true")
```


© 2023 Cadence Design Systems, Inc.

QContourValues

```
viva.smithGrid QContourValues string VSWRvalues
```

Description

Specifies the Q values for which you want to draw the Q contours on the Smith grid.

The default value is `""`, which indicates that no Q contours are drawn on the Smith grid.

Q contours for the specified Q values are drawn only when `showQContours` environment variable is set to `true`.

GUI Equivalent

The *Q* field in the *Smith Chart* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "QContourValues")
envSetVal("viva.smithGrid" "QContourValues" 'string "3, 5")
```


© 2023 Cadence Design Systems, Inc.

QContourPenColor

```
viva.smithGrid QContourPenColor string hexColorCode
```

Description

Specifies the color of the Q contours.

The default value is `#C000C0`.

Q contours for the specified Q values are drawn only when the `showQContours` environment variable is set to `true`.

GUI Equivalent

The *Color* field in the *Q Contours* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "QContourPenColor")
envSetVal("viva.smithGrid" "QContourPenColor" 'string "#0000FF")
```


© 2023 Cadence Design Systems, Inc.

QContourPenWidth

`viva.smithGrid QContourPenWidth string` *`widthOfVSWRCircles`*

Description

Specifies the width of the Q contours.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `fine`.

Q contours for the specified Q values are drawn only when the `showQContours` environment variable is set to `true`.

GUI Equivalent

The *Width* field in the *Q Contours* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "QContourPenWidth")
envSetVal("viva.smithGrid" "QContourPenWidth" 'string "extraThick")
```


© 2023 Cadence Design Systems, Inc.

QContourPenStyle

`viva.smithGrid QContourPenStyle string` *`widthOfVSWRCircles`*

Description

Specifies the style of the Q contours.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `solid`.

Q contours for the specified Q values are drawn only when the `showQContours` environment variable is set to `true`.

GUI Equivalent

The *Style* field in the *Q Contours* section on the *Annotations* tab of the Graph Properties form for a Smith chart. This field is not available for polar grids.

Examples

```
envGetVal("viva.smithGrid" "QContourPenStyle")
envSetVal("viva.smithGrid" "QContourPenStyle" 'string "dashDotDot")
```


© 2023 Cadence Design Systems, Inc.

impedanceStandardGridPenColor

```
viva.smithGrid impedanceStandardGridPenColor string
     colorOfImpedanceStandardGrid
```

Description

Specifies the color of standard grid lines in an Impedance (Z) Smith chart.

The default value is `#000000`.

GUI Equivalent

The *Color* field in the *Standard Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceStandardGridPenColor")
envSetVal("viva.smithGrid" "impedanceStandardGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

impedanceStandardGridPenWidth

```
viva.smithGrid impedanceStandardGridPenWidth string
     widthOfImpedanceStandardGrid
```

Description

Specifies the width of standard grid lines in an Impedance (Z) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Standard Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceStandardGridPenWidth")
envSetVal("viva.smithGrid" "impedanceStandardGridPenWidth" 'string "extraThick")
```


© 2023 Cadence Design Systems, Inc.

impedanceStandardGridPenStyle

```
viva.smithGrid impedanceStandardGridPenStyle string
     styleOfImpedanceStandardGrid
```

Description

Specifies the style of standard grid lines in an Impedance (Z) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `solid`.

GUI Equivalent

The *Style* field in the *Standard Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceStandardGridPenStyle")
envSetVal("viva.smithGrid" "impedanceStandardGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

impedanceDetailedGridPenColor

```
viva.smithGrid impedanceDetailedGridPenColor string
     colorOfImpedanceDetailedGrid
```

Description

Specifies the color of detailed grid lines in an Impedance (Z) Smith chart.

The default value is `#000000`.

GUI Equivalent

The *Color* field in the *Detailed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceDetailedGridPenColor")
envSetVal("viva.smithGrid" "impedanceDetailedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

impedanceDetailedGridPenWidth

```
viva.smithGrid impedanceDetailedGridPenWidth string
     widthOfImpedanceDetailedGrid
```

Description

Specifies the width of detailed grid lines in an Impedance (Z) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Detailed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceDetailedGridPenWidth")
envSetVal("viva.smithGrid" "impedanceDetailedGridPenWidth" 'string "extraThick")
```


© 2023 Cadence Design Systems, Inc.

impedanceDetailedGridPenStyle

```
viva.smithGrid impedanceDetailedGridPenStyle string
     styleOfImpedanceDetailedGrid
```

Description

Specifies the style of detailed grid lines in an Impedance (Z) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

GUI Equivalent

The *Style* field in the *Detailed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceDetailedGridPenStyle")
envSetVal("viva.smithGrid" "impedanceDetailedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

impedanceCompressedGridPenColor

```
viva.smithGrid impedanceCompressedGridPenColor string
     colorOfImpedanceCompressedGrid
```

Description

Specifies the color of the compressed grid lines in an Impedance (Z) Smith chart.

The default value is `#000000`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Color* field in the *Compressed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceCompressedGridPenColor")
envSetVal("viva.smithGrid" "impedanceCompressedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

impedanceCompressedGridPenWidth

```
viva.smithGrid impedanceCompressedGridPenWidth string
     widthOfImpedanceCompressedGrid
```

Description

Specifies the width of compressed grid lines in an Impedance (Z) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Width* field in the *Compressed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceCompressedGridPenWidth")
envSetVal("viva.smithGrid" "impedanceCompressedGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

impedanceCompressedGridPenStyle

```
viva.smithGrid impedanceCompressedGridPenStyle string
     styleOfImpedanceCompressedGrid
```

Description

Specifies the style of compressed grid lines in an Impedance (Z) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Style* field in the *Detailed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Impedance (Z) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Impedance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceCompressedGridPenStyle")
envSetVal("viva.smithGrid" "impedanceCompressedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

admittanceStandardGridPenColor

```
viva.smithGrid admittanceStandardGridPenColor string
     colorOfAdmittanceStandardGrid
```

Description

Specifies the color of standard grid lines in an Admittance (Y) Smith chart.

The default value is `#000000`.

GUI Equivalent

The *Color* field in the *Standard Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceStandardGridPenColor")
envSetVal("viva.smithGrid" "admittanceStandardGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

admittanceStandardGridPenWidth

```
viva.smithGrid admittanceStandardGridPenWidth string
     widthOfAdmittanceStandardGrid
```

Description

Specifies the width of standard grid lines in an Admittance (Y) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Standard Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceStandardGridPenWidth")
envSetVal("viva.smithGrid" "admittanceStandardGridPenWidth" 'string "extraThick")
```


© 2023 Cadence Design Systems, Inc.

admittanceStandardGridPenStyle

```
viva.smithGrid admittanceStandardGridPenStyle string
     styleOfAdmittanceStandardGrid
```

Description

Specifies the style of standard grid lines in an Admittance (Y) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `solid`.

GUI Equivalent

The *Style* field in the *Standard Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "impedanceStandardGridPenStyle")
envSetVal("viva.smithGrid" "impedanceStandardGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

admittanceDetailedGridPenColor

```
viva.smithGrid admittanceDetailedGridPenColor string
     colorOfAdmittanceDetailedGrid
```

Description

Specifies the color of detailed grid lines in an Admittance (Y) Smith chart.

The default value is `#000000`.

GUI Equivalent

The *Color* field in the *Detailed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceDetailedGridPenColor")
envSetVal("viva.smithGrid" "admittanceDetailedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

admittanceDetailedGridPenWidth

```
viva.smithGrid admittanceDetailedGridPenWidth string
     widthOfAdmittanceDetailedGrid
```

Description

Specifies the width of detailed grid lines in an Admittance (Y) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Detailed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceDetailedGridPenWidth")
envSetVal("viva.smithGrid" "admittanceDetailedGridPenWidth" 'string "extraThick")
```


© 2023 Cadence Design Systems, Inc.

admittanceDetailedGridPenStyle

```
viva.smithGrid admittanceDetailedGridPenStyle string
     styleOfAdmittanceDetailedGrid
```

Description

Specifies the style of detailed grid lines in an Admittance (Y) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

GUI Equivalent

The *Style* field in the *Detailed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceDetailedGridPenStyle")
envSetVal("viva.smithGrid" "admittanceDetailedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

admittanceCompressedGridPenColor

```
viva.smithGrid admittanceCompressedGridPenColor string
     colorOfAdmittanceCompressedGrid
```

Description

Specifies the color of the compressed grid lines in an Admittance (Y) Smith chart.

The default value is `#000000`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Color* field in the *Compressed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceCompressedGridPenColor")
envSetVal("viva.smithGrid" "admittanceCompressedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

admittanceCompressedGridPenWidth

```
viva.smithGrid admittanceCompressedGridPenWidth string
     widthOfAdmittanceCompressedGrid
```

Description

Specifies the width of compressed grid lines in an Admittance (Y) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Width* field in the *Compressed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceCompressedGridPenWidth")
envSetVal("viva.smithGrid" "admittanceCompressedGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

admittanceCompressedGridPenStyle

```
viva.smithGrid admittanceCompressedGridPenStyle string
     styleOfAdmittanceCompressedGrid
```

Description

Specifies the style of compressed grid lines in an Admittance (Y) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Style* field in the *Compressed Grid Lines* section on the *Grid* tab of the Graph Properties form for an Admittance (Y) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Admittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "admittanceCompressedGridPenStyle")
envSetVal("viva.smithGrid" "admittanceCompressedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

immittanceZStandardGridPenColor

```
viva.smithGrid immittanceZStandardGridPenColor string
     colorOfImmittanceZStandardGrid
```

Description

Specifies the color of standard Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

The default value is `#C00000`.

GUI Equivalent

The *Color* field in the *Standard Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZStandardGridPenColor")
envSetVal("viva.smithGrid" "immittanceZStandardGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

immittanceZStandardGridPenWidth

```
viva.smithGrid immittanceZStandardGridPenWidth string
     widthOfImmittanceZStandardGrid
```

Description

Specifies the width of standard Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Standard Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZStandardGridPenWidth")
envSetVal("viva.smithGrid" "immittanceZStandardGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

immittanceZStandardGridPenStyle

```
viva.smithGrid immittanceZStandardGridPenStyle string
     styleOfImmittanceZStandardGrid
```

Description

Specifies the style of standard Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

GUI Equivalent

The *Style* field in the *Standard Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZStandardGridPenStyle")
envSetVal("viva.smithGrid" "immittanceZStandardGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

immittanceZDetailedGridPenColor

```
viva.smithGrid immittanceZDetailedGridPenColor string
     colorOfImmittanceZDetailedGrid
```

Description

Specifies the color of detailed Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

The default value is `#C00000`.

GUI Equivalent

The *Color* field in the *Detailed Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZDetailedGridPenColor")
envSetVal("viva.smithGrid" "immittanceZDetailedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

immittanceZDetailedGridPenWidth

```
viva.smithGrid immittanceZDetailedGridPenWidth string
     widthOfImmittanceZDetailedGrid
```

Description

Specifies the width of detailed Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Detailed Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZDetailedGridPenWidth")
envSetVal("viva.smithGrid" "immittanceZDetailedGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

immittanceZDetailedGridPenStyle

```
viva.smithGrid immittanceZDetailedGridPenStyle string
     styleOfImmittanceZDetailedGrid
```

Description

Specifies the style of detailed Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

GUI Equivalent

The *Style* field in the *Detailed Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZDetailedGridPenStyle")
envSetVal("viva.smithGrid" "immittanceZDetailedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

immittanceZCompressedGridPenColor

```
viva.smithGrid immittanceZCompressedGridPenColor string
     colorOfImmittanceZCompressedGrid
```

Description

Specifies the color of the compressed Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

The default value is `#C00000`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Color* field in the *Compressed Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZCompressedGridPenColor")
envSetVal("viva.smithGrid" "immittanceZCompressedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

immittanceZCompressedGridPenWidth

```
viva.smithGrid immittanceZCompressedGridPenWidth string
     widthOfImmittanceZCompressedGrid
```

Description

Specifies the width of compressed Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Width* field in the *Compressed Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZCompressedGridPenWidth")
envSetVal("viva.smithGrid" "immittanceZCompressedGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

immittanceZCompressedGridPenStyle

```
viva.smithGrid immittanceZCompressedGridPenStyle string
     styleOfImmittanceZCompressedGrid
```

Description

Specifies the style of compressed Impedance (Z Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Style* field in the *Compressed Impedance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceZCompressedGridPenStyle")
envSetVal("viva.smithGrid" "immittanceZCompressedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

immittanceYStandardGridPenColor

```
viva.smithGrid immittanceYStandardGridPenColor string
     colorOfImmittanceYStandardGrid
```

Description

Specifies the color of standard Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

The default value is `#00C000`.

GUI Equivalent

The *Color* field in the *Standard Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYStandardGridPenColor")
envSetVal("viva.smithGrid" "immittanceYStandardGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

immittanceYStandardGridPenWidth

```
viva.smithGrid immittanceYStandardGridPenWidth string
     widthOfImmittanceYStandardGrid
```

Description

Specifies the width of standard Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Standard Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYStandardGridPenWidth")
envSetVal("viva.smithGrid" "immittanceYStandardGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

immittanceYStandardGridPenStyle

```
viva.smithGrid immittanceYStandardGridPenStyle string
     styleOfImmittanceYStandardGrid
```

Description

Specifies the style of standard Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

GUI Equivalent

The *Style* field in the *Standard Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYStandardGridPenStyle")
envSetVal("viva.smithGrid" "immittanceYStandardGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

immittanceYDetailedGridPenColor

```
viva.smithGrid immittanceYDetailedGridPenColor string
     colorOfImmittanceYDetailedGrid
```

Description

Specifies the color of detailed Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

The default value is `#00C000`.

GUI Equivalent

The *Color* field in the *Detailed Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYDetailedGridPenColor")
envSetVal("viva.smithGrid" "immittanceYDetailedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

immittanceYDetailedGridPenWidth

```
viva.smithGrid immittanceYDetailedGridPenWidth string
     widthOfImmittanceYDetailedGrid
```

Description

Specifies the width of detailed Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

GUI Equivalent

The *Width* field in the *Detailed Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYDetailedGridPenWidth")
envSetVal("viva.smithGrid" "immittanceYDetailedGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

immittanceYDetailedGridPenStyle

```
viva.smithGrid immittanceYDetailedGridPenStyle string
     styleOfImmittanceYDetailedGrid
```

Description

Specifies the style of detailed Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

GUI Equivalent

The *Style* field in the *Detailed Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYDetailedGridPenStyle")
envSetVal("viva.smithGrid" "immittanceYDetailedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

immittanceYCompressedGridPenColor

```
viva.smithGrid immittanceYCompressedGridPenColor string
     colorOfImmittanceYCompressedGrid
```

Description

Specifies the color of the compressed Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

The default value is `#00C000`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Color* field in the *Compressed Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYCompressedGridPenColor")
envSetVal("viva.smithGrid" "immittanceYCompressedGridPenColor" 'string "#FF0000")
```


© 2023 Cadence Design Systems, Inc.

immittanceYCompressedGridPenWidth

```
viva.smithGrid immittanceYCompressedGridPenWidth string
     widthOfImmittanceYCompressedGrid
```

Description

Specifies the width of compressed Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `fine`, `medium`, `thick`, `extraThick`, and `extraFine`. The default value is `extraFine`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Width* field in the *Compressed Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYCompressedGridPenWidth")
envSetVal("viva.smithGrid" "immittanceYCompressedGridPenWidth" 'string
"extraThick")
```


© 2023 Cadence Design Systems, Inc.

immittanceYCompressedGridPenStyle

```
viva.smithGrid immittanceYCompressedGridPenStyle string
     styleOfImmittanceYCompressedGrid
```

Description

Specifies the style of compressed Admittance (Y Smith) grid in an Immittance (ZY) Smith chart.

Valid values are `solid`, `dot`, `dash`, `dashDot`, and `dashDotDot`. The default value is `dot`.

This environment variable works only when the environment variable `showCompressed` is set to `true`.

GUI Equivalent

The *Style* field in the *Compressed Admittance Grid Lines* section on the *Grid* tab of the Graph Properties form for an Immittance (ZY) Smith chart. This field is available when the *Compressed Smith* check box is selected and *Immittance* is selected as the *Grid Type*.

Examples

```
envGetVal("viva.smithGrid" "immittanceYCompressedGridPenStyle")
envSetVal("viva.smithGrid" "immittanceYCompressedGridPenStyle" 'string "dashDot")
```


© 2023 Cadence Design Systems, Inc.

Application Environment Variables

enableFlexibleBind

Determines whether the graph or trace settings, such as graph title, trace zoom in, trace visibility, delta and AB markers are to be saved while loading a graph file. The default value of this variable is `true`, which means these settings will be saved.

Syntax

viva.application enableFlexibleBind string "*true***"**

Valid Values

- `true`
- `false`

VIVALicenseCheckoutOrder

Specifies the license checkout order preference for Virtuoso Visualization and Analysis XL. By default, this variable is set to `ViVA, ADE`, which results in the following license check out tasks being performed:

- Checks out the Virtuoso Visualization and Analysis XL license, if available.
- If the checkout operation in the previous step fails, checks out the ADE license according to the checkout order specified in the `maestroCheckoutOrder` .cdsenv variable.

If this variable is set to `ADE, ViVA`, the license check out tasks are performed in the following order:

- Checks out the ADE license according to the checkout order specified in the `maestroCheckoutOrder` .cdsenv variable.
- If the checkout operation in the previous step fails, checks out the Virtuoso Visualization and Analysis XL license.

Note: The license is released when all the Virtuoso Visualization and Analysis XL windows are closed.

© 2023 Cadence Design Systems, Inc.

Syntax

Important

The `VIVALicenseCheckoutOrder` environment variable can be set using `viva.application` and `license` tool partitions. However, the `viva.application` tool partition will be removed in a future release. It is recommended that you use the `license` tool partition to set this variable.

Important

You must set this environment variable in the `.cdsinit` file or the `.cdsenv` file. Do not set this environment variable in the CIW.

```
license VIVALicenseCheckoutOrder string "VIVA,ADE,Verifier"
```

For more information, see `VIVALicenseCheckoutOrder` in the `CheckoutOrder Variables` section of *Virtuoso Software Licensing and Configuration User Guide.*

```
viva.application VIVALicenseCheckoutOrder string "VIVA,ADE"
```

Valid Values

- `VIVA,ADE`
- `ADE,VIVA`

Default Values

`VIVA,ADE`

VIVA_UseNextLicense

Controls whether the prompt is to be displayed to confirm the checkout of next specified license in the `VIVALicenseCheckoutOrder` variable when the first license checkout fails.

When this variable is set to `prompt`, which is the default value, if the first license specified in `VIVALicenseCheckoutOrder` is not available, a prompt message appears to confirm if you want to check out the next specified license.

If this variable is set to `always`, no prompt message is displayed when the first license checkout fails and the next specified license is automatically checked out if available.

© 2023 Cadence Design Systems, Inc.

Syntax

Important

The `VIVA_UseNextLicense` environment variable can be set using `viva.application` and `license` tool partitions. However, the `viva.application` tool partition will be removed in a future release. It is recommended that you use the `license` tool partition to set this variable.

```
license VIVA_UseNextLicense string "prompt"
```

```
viva.application VIVA_UseNextLicense string "prompt"
```

Valid Values

- `prompt`
- `always`

Default Values

`prompt`

maxTraceAge

Controls the maximum number of waveforms that are to be displayed in an info-balloon during the RTT simulation run in ADE Explorer. By default, a maximum of 4 waveforms is displayed in the info-balloon. The most recent waveform is displayed at the top of the legend.

Syntax

```
viva.application maxTraceAge 'string "4"
```

Valid Values

`Any integer value`

Default Value

`4`

© 2023 Cadence Design Systems, Inc.

confirmCloseWindow

```
viva.application confirmCloseWindow string {"true" | "false"}
```

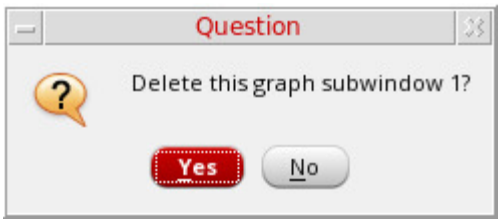
Description

Specifies whether to display a confirmation dialog box when deleting a graph window or a subwindow by:

- Selecting the *Edit – Delete* command.
- Right-clicking a graph window or a subwindow, and then selecting *Delete* from the context-sensitive menu.
- Pressing the bindkey `Delete`.

Valid values are as follows:

- `true`: (Default value) A confirmation dialog box appears before deleting the active graph window or subwindow:



- `false`: The active graph window or the subwindow is deleted without displaying any confirmation dialog box.

GUI Equivalent

Edit – Delete (`Del`)

Examples

```
envGetVal("viva.application" "confirmCloseWindow")
envSetVal("viva.application" "confirmCloseWindow" 'string "false")
```


© 2023 Cadence Design Systems, Inc.

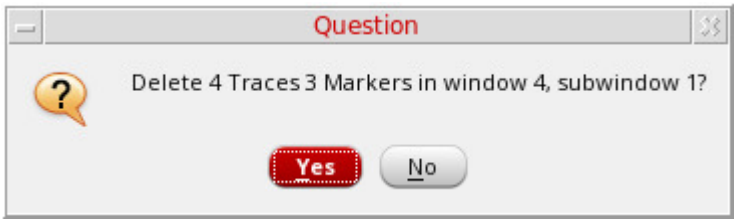
confirmDeleteAll

```
viva.application confirmDeleteAll string {"true" | "false"}
```

Description

Specifies whether to display a confirmation dialog box when the *Edit – Delete All* command is selected or the bindkey `E` is pressed. Valid values are as follows:

- `true`: (Default value) A confirmation dialog box appears, displaying the count of objects (traces and markers) to be deleted in the active graph window or subwindow:



- `false`: The objects are deleted without displaying any confirmation dialog box.

GUI Equivalent

Edit – Delete All (`E`)

Examples

```
envGetVal("viva.application" "confirmDeleteAll")
envSetVal("viva.application" "confirmDeleteAll" 'string "false")
```


© 2023 Cadence Design Systems, Inc.

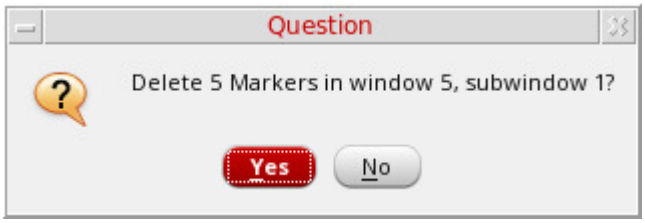
confirmDeleteAllMarkers

```
viva.application confirmDeleteAllMarkers string {"true" | "false"}
```

Description

Specifies whether to display a confirmation dialog box when the *Marker – Delete All* command is selected or the bindkey `Ctrl+E` is pressed. Valid values are as follows:

- `true`: A confirmation dialog box appears, displaying the count of markers to be deleted in the active graph window or subwnidow:



- `false`: (Default value) The markers are deleted without displaying any confirmation dialog box.

GUI Equivalent

Marker – Delete All (`Ctrl+E`)

Examples

```
envGetVal("viva.application" "confirmDeleteAllMarkers")
envSetVal("viva.application" "confirmDeleteAllMarkers" 'string "true")
```


© 2023 Cadence Design Systems, Inc.

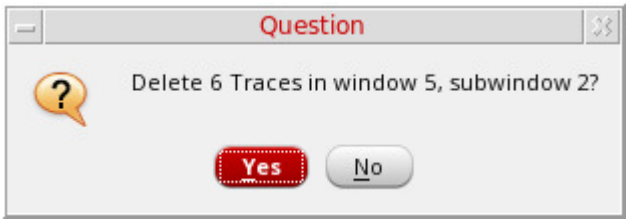
confirmDeleteAllTraces

```
viva.application confirmDeleteAllTraces string {"true" | "false"}
```

Description

Specifies whether to display a confirmation dialog box when the *Trace – Delete All* command is selected or the bindkey `Shift+E` is pressed. Valid values are as follows:

- `true`: A confirmation dialog box appears, displaying the count of traces to be deleted in the active graph window or subwnidow:



- `false`: (Default value) The traces are deleted without displaying any confirmation dialog box.

GUI Equivalent

Trace – Delete All (`Shift+E`)

Examples

```
envGetVal("viva.application" "confirmDeleteAllTraces")
envSetVal("viva.application" "confirmDeleteAllTraces" 'string "true")
```


© 2023 Cadence Design Systems, Inc.

XRange Bookmark Variables

- defaultLabel
- font
- significantDigits
- sigDigitsMode
- foreground
- background
- fillColor
- notation
- lineStyle
- showLabel
- zoomScaleFactor

© 2023 Cadence Design Systems, Inc.

defaultLabel

`viva.xrangeBookmark defaultLabel string` *`labelText`*

Description

Specifies the text to be displayed on the x-range bookmark label. The default value is `""`, which indicates that no text is displayed on the x-range bookmark label by default.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Label Text* field on the XRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.xrangeBookmark" "defaultLabel")
envSetVal("viva.xrangeBookmark" "defaultLabel" 'string "myBookmark")
```


© 2023 Cadence Design Systems, Inc.

font

`viva.xrangeBookmark font string` *`fontProperties`*

Description

Specifies the font properties of the x-range bookmark. The default value is `"Default,10,-1,5,50,0,0,0,0,0"`.

For more information about the format of the font properties, see Font String.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Font/Color* field on the XRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.xrangeBookmark" "font")
envSetVal("viva.xrangeBookmark" "font" 'string "Default,20,-1,5,50,0,0,0,0,0"")
```


© 2023 Cadence Design Systems, Inc.

significantDigits

```
viva.xrangeBookmark significantDigits string numberOfSignificantDigits
```

Description

Specifies the number of significant digits for the x-range bookmark. The default value is `"4"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Significant Digits* field on the XRange Bookmark Properties for *bookmark* form.

Examples

```
envGetVal("viva.xrangeBookmark" "significantDigits")
envSetVal("viva.xrangeBookmark" "significantDigits" 'string "3")
```


© 2023 Cadence Design Systems, Inc.

sigDigitsMode

```
viva.xrangeBookmark sigDigitsMode string {"Auto" | "Manual"}
```

Description

Sets the mode to be used to specify the significant digits for the x-range bookmark.

Possible values are:

- `Auto`: (Default value) When it is set to `Auto`, the default value for `significantDigits` is used.
- `Manual`: When it is set to `Manual`, the settings defined in the `significantDigits` variable is applied.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Significant Digits* field on the XRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.xrangeBookmark" "sigDigitsMode")
envSetVal("viva.xrangeBookmark" "sigDigitsMode" 'string "Manual")
```


© 2023 Cadence Design Systems, Inc.

foreground

`viva.xrangeBookmark foreground string` *`foregroundColor`*

Description

Specifies the foreground color of the x-range bookmark. The default value is `"lightblue"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Foreground* field on the XRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.xrangeBookmark" "foreground")
envSetVal("viva.xrangeBookmark" "foreground" 'string "red")
```


© 2023 Cadence Design Systems, Inc.

background

`viva.xrangeBookmark background string` *`backgroundColor`*

Description

Specifies the background color of the x-range bookmark. The default value is `"black"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.xrangeBookmark" "background")
envSetVal("viva.xrangeBookmark" "background" 'string "yellow")
```


© 2023 Cadence Design Systems, Inc.

fillColor

`viva.xrangeBookmark fillColor string` *`fillColor`*

Description

Specifies the foreground color of the x-range bookmark. The default value is `"blue"`.

All possible values are defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Foreground* field on the XRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.xrangeBookmark" "fillColor")
envSetVal("viva.xrangeBookmark" "fillColor" 'string "red")
```


© 2023 Cadence Design Systems, Inc.

notation

```
viva.xrangeBookmark notation string {"suffix" | "engineering" | "scientific"}
```

Description

Specifies the notation to be used in the x-range bookmark. The default value is `"suffix"`.

Possible values are: `suffix`, `engineering`, and `scientific`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Notation* field on the XRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.xrangeBookmark" "notation")
envSetVal("viva.xrangeBookmark" "notation" 'string "scientific")
```


© 2023 Cadence Design Systems, Inc.

lineStyle

```
viva.xrangeBookmark lineStyle string {"dash" | "solid" | "dot" | "dashdot" |
     "dashdotdot"}
```

Description

Specifies the line style of the x-range bookmark. The default value is `"dash"`.

Possible values are: `dash`, `solid`, `dot`, `dashdot`, and `dashdotdot`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.xrangeBookmark" "lineStyle")
envSetVal("viva.xrangeBookmark" "lineStyle" 'string "dashdotdot")
```


© 2023 Cadence Design Systems, Inc.

showLabel

```
viva.xrangeBookmark showLabel string {"on when hover" | "on" | "off"}
```

Description

Specifies when to show bookmark label on the x-range bookmark.

Possible values are:

- `on when hover`: (Default value) Bookmark labels are shown only when you hover over the bookmark.
- `on`: Bookmark labels are always shown.
- `off`: Bookmark labels are always hidden.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Annotation* field on the XRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.xrangeBookmark" "showLabel")
envSetVal("viva.xrangeBookmark" "showLabel" 'string "on")
```


© 2023 Cadence Design Systems, Inc.

zoomScaleFactor

`viva.xrangeBookmark zoomScaleFactor string` *`zoomScaleFactor`*

Description

Specifies the scale factor to be used when zooming the x-range bookmark. The default value is `"1.5"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.xrangeBookmark" "zoomScaleFactor")
envSetVal("viva.xrangeBookmark" "zoomScaleFactor" 'string "2.0")
```


© 2023 Cadence Design Systems, Inc.

YRange Bookmark Variables

- defaultLabel
- font
- significantDigits
- sigDigitsMode
- foreground
- background
- fillColor
- notation
- lineStyle
- showLabel
- zoomScaleFactor

© 2023 Cadence Design Systems, Inc.

defaultLabel

`viva.yrangeBookmark defaultLabel string` *`labelText`*

Description

Specifies the text to be displayed on the y-range bookmark label. The default value is `""`, which indicates that no text is displayed on the y-range bookmark label by default.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Label Text* field on the YRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.yrangeBookmark" "defaultLabel")
envSetVal("viva.yrangeBookmark" "defaultLabel" 'string "myBookmark")
```


© 2023 Cadence Design Systems, Inc.

font

`viva.yrangeBookmark font string` *`fontProperties`*

Description

Specifies the font properties of the y-range bookmark. The default value is `"Default,10,-1,5,50,0,0,0,0,0"`.

For more information about the format of the font properties, see Font String.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Font/Color* field on the YRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.yrangeBookmark" "font")
envSetVal("viva.yrangeBookmark" "font" 'string "Default,20,-1,5,50,0,0,0,0,0"")
```


© 2023 Cadence Design Systems, Inc.

significantDigits

```
viva.yrangeBookmark significantDigits string numberOfSignificantDigits
```

Description

Specifies the number of significant digits for the y-range bookmark. The default value is `"4"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Significant Digits* field on the YRange Bookmark Properties for *bookmark* form.

Examples

```
envGetVal("viva.yrangeBookmark" "significantDigits")
envSetVal("viva.yrangeBookmark" "significantDigits" 'string "3")
```


© 2023 Cadence Design Systems, Inc.

sigDigitsMode

```
viva.yrangeBookmark sigDigitsMode string {"Auto" | "Manual"}
```

Description

Sets the mode to be used to specify the significant digits for the y-range bookmark.

Possible values are:

- `Auto`: (Default value) When it is set to `Auto`, the default value for `significantDigits` is used.
- `Manual`: When it is set to `Manual`, the settings defined in the `significantDigits` variable is applied.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Significant Digits* field on the YRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.yrangeBookmark" "sigDigitsMode")
envSetVal("viva.yrangeBookmark" "sigDigitsMode" 'string "Manual")
```


© 2023 Cadence Design Systems, Inc.

foreground

`viva.yrangeBookmark foreground string` *`foregroundColor`*

Description

Specifies the foreground color of the y-range bookmark. The default value is `"lightblue"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Foreground* field on the YRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.yrangeBookmark" "foreground")
envSetVal("viva.yrangeBookmark" "foreground" 'string "red")
```


© 2023 Cadence Design Systems, Inc.

background

`viva.yrangeBookmark background string` *`backgroundColor`*

Description

Specifies the background color of the y-range bookmark. The default value is `"black"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.yrangeBookmark" "background")
envSetVal("viva.yrangeBookmark" "background" 'string "yellow")
```


© 2023 Cadence Design Systems, Inc.

fillColor

`viva.yrangeBookmark fillColor string` *`fillColor`*

Description

Specifies the foreground color of the y-range bookmark. The default value is `"blue"`.

All possible values are defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Foreground* field on the YRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.yrangeBookmark" "fillColor")
envSetVal("viva.yrangeBookmark" "fillColor" 'string "red")
```


© 2023 Cadence Design Systems, Inc.

notation

```
viva.yrangeBookmark notation string {"suffix" | "engineering" | "scientific"}
```

Description

Specifies the notation to be used in the y-range bookmark. The default value is `"suffix"`.

Possible values are: `suffix`, `engineering`, and `scientific`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Notation* field on the YRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.yrangeBookmark" "notation")
envSetVal("viva.yrangeBookmark" "notation" 'string "scientific")
```


© 2023 Cadence Design Systems, Inc.

lineStyle

```
viva.yrangeBookmark lineStyle string {"dash" | "solid" | "dot" | "dashdot" |
     "dashdotdot"}
```

Description

Specifies the line style of the y-range bookmark. The default value is `"dash"`.

Possible values are: `dash`, `solid`, `dot`, `dashdot`, and `dashdotdot`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.yrangeBookmark" "lineStyle")
envSetVal("viva.yrangeBookmark" "lineStyle" 'string "dashdotdot")
```


© 2023 Cadence Design Systems, Inc.

showLabel

```
viva.yrangeBookmark showLabel string {"on when hover" | "on" | "off"}
```

Description

Specifies when to show bookmark label on the y-range bookmark.

Possible values are:

- `on when hover`: (Default value) Bookmark labels are shown only when you hover over the bookmark.
- `on`: Bookmark labels are always shown.
- `off`: Bookmark labels are always hidden.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Annotation* field on the YRange Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.yrangeBookmark" "showLabel")
envSetVal("viva.yrangeBookmark" "showLabel" 'string "on")
```


© 2023 Cadence Design Systems, Inc.

zoomScaleFactor

`viva.yrangeBookmark zoomScaleFactor string` *`zoomScaleFactor`*

Description

Specifies the scale factor to be used when zooming the y-range bookmark. The default value is `"1.5"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.yrangeBookmark" "zoomScaleFactor")
envSetVal("viva.yrangeBookmark" "zoomScaleFactor" 'string "2.0")
```


© 2023 Cadence Design Systems, Inc.

Region Bookmark Variables

- defaultLabel
- font
- significantDigits
- sigDigitsMode
- foreground
- background
- fillColor
- notation
- lineStyle
- showLabel
- zoomScaleFactor

© 2023 Cadence Design Systems, Inc.

defaultLabel

`viva.regionBookmark defaultLabel string` *`labelText`*

Description

Specifies the text to be displayed on the region bookmark label. The default value is `""`, which indicates that no text is displayed on the region bookmark label by default.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Label Text* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "defaultLabel")
envSetVal("viva.regionBookmark" "defaultLabel" 'string "myBookmark")
```


© 2023 Cadence Design Systems, Inc.

font

`viva.regionBookmark font string` *`fontProperties`*

Description

Specifies the font properties of the region bookmark. The default value is `"Default,10,-1,5,50,0,0,0,0,0"`.

For more information about the format of the font properties, see Font String.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Font/Color* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "font")
envSetVal("viva.regionBookmark" "font" 'string "Default,20,-1,5,50,0,0,0,0,0"")
```


© 2023 Cadence Design Systems, Inc.

significantDigits

`viva.regionBookmark significantDigits string` *`numberOfSignificantDigits`*

Description

Specifies the number of significant digits for the region bookmark. The default value is `"4"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Significant Digits* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "significantDigits")
envSetVal("viva.regionBookmark" "significantDigits" 'string "3")
```


© 2023 Cadence Design Systems, Inc.

sigDigitsMode

```
viva.regionBookmark sigDigitsMode string {"Auto" | "Manual"}
```

Description

Sets the mode to be used to specify the significant digits for the region bookmark.

Possible values are:

- `Auto`: (Default value) When it is set to `Auto`, the default value for `significantDigits` is used.
- `Manual`: When it is set to `Manual`, the settings defined in the `significantDigits` variable is applied.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Significant Digits* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "sigDigitsMode")
envSetVal("viva.regionBookmark" "sigDigitsMode" 'string "Manual")
```


© 2023 Cadence Design Systems, Inc.

foreground

`viva.regionBookmark foreground string` *`foregroundColor`*

Description

Specifies the foreground color of the region bookmark. The default value is `"lightblue"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Foreground* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "foreground")
envSetVal("viva.regionBookmark" "foreground" 'string "red")
```


© 2023 Cadence Design Systems, Inc.

background

`viva.regionBookmark background string` *`backgroundColor`*

Description

Specifies the background color of the region bookmark. The default value is `"black"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.regionBookmark" "background")
envSetVal("viva.regionBookmark" "background" 'string "yellow")
```


© 2023 Cadence Design Systems, Inc.

fillColor

`viva.regionBookmark fillColor string` *`fillColor`*

Description

Specifies the foreground color of the region bookmark. The default value is `"blue"`.

All possible values are defined at the following location:

http://www.w3.org/TR/SVG/types.html#ColorKeywords

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Foreground* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "fillColor")
envSetVal("viva.regionBookmark" "fillColor" 'string "red")
```


© 2023 Cadence Design Systems, Inc.

notation

```
viva.regionBookmark notation string {"suffix" | "engineering" | "scientific"}
```

Description

Specifies the notation to be used in the region bookmark. The default value is `"suffix"`.

Possible values are: `suffix`, `engineering`, and `scientific`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Notation* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "notation")
envSetVal("viva.regionBookmark" "notation" 'string "scientific")
```


© 2023 Cadence Design Systems, Inc.

lineStyle

```
viva.regionBookmark lineStyle string {"dash" | "solid" | "dot" | "dashdot" |
     "dashdotdot"}
```

Description

Specifies the line style of the region bookmark. The default value is `"dash"`.

Possible values are: `dash`, `solid`, `dot`, `dashdot`, and `dashdotdot`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.regionBookmark" "lineStyle")
envSetVal("viva.regionBookmark" "lineStyle" 'string "dashdotdot")
```


© 2023 Cadence Design Systems, Inc.

showLabel

```
viva.regionBookmark showLabel string {"on when hover" | "on" | "off"}
```

Description

Specifies when to show bookmark label on the region bookmark.

Possible values are:

- `on when hover`: (Default value) Bookmark labels are only shown when you hover over the bookmark.
- `on`: Bookmark labels are always shown.
- `off`: Bookmark labels are always hidden.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

The *Annotation* field on the Region Bookmark Properties for *`bookmark`* form.

Examples

```
envGetVal("viva.regionBookmark" "showLabel")
envSetVal("viva.regionBookmark" "showLabel" 'string "on")
```


© 2023 Cadence Design Systems, Inc.

zoomScaleFactor

`viva.regionBookmark zoomScaleFactor string` *`zoomScaleFactor`*

Description

Specifies the scale factor to be used when zooming the region bookmark. The default value is `"1.5"`.

This environment variable works only when it is set in the `.cdsinit` file.

GUI Equivalent

None

Examples

```
envGetVal("viva.regionBookmark" "zoomScaleFactor")
envSetVal("viva.regionBookmark" "zoomScaleFactor" 'string "2.0")
```


© 2023 Cadence Design Systems, Inc.

Results Browser Variables

This section describes the following Results Browser environment variables:

- historyLength on page 967
- plotstyle on page 968
- dataDirHome on page 969
- ignorePathsForDirValidityCheck on page 969
- ignoreDirLevelForDirValidityCheck on page 969

historyLength

Specifies the maximum number of dataset paths saved in the *Location* pull-down in the Results Browser.

Syntax

viva.browser historyLength string "*maxDirectories***"**

Values

maxDirectories	Maximum number of data directories that fit in the *Location* field. Default: 10 Valid values: 0–20

© 2023 Cadence Design Systems, Inc.

plotstyle

Specifies the default plotting style for a new graph. This sets the default value for the plot style pull-down in the top right corner of the Results Browser.

Syntax

```
viva.browser plotStyle string "Append" | "Replace" | "New SubWin" | "New Win"
```

Values

`append`	Appends the new graph to the current graph.
`replace`	Replaces the current graph with the new graph. This is the default value.
`newsub`	Plots the graph in a new subwindow.
`newwin`	Plots the graph in a new window.

© 2023 Cadence Design Systems, Inc.

dataDirHome

Specifies the default directory for the Choose Data Directory dialog box.

Syntax

viva.browser dataDirHome string "*directory***"**

Values

directory	Directory which the Choose Data Directory dialog box defaults to. Default: `./`

ignorePathsForDirValidityCheck

Specifies the results directory paths that are to be ignored by the Select Waveform database while checking the validity of results directories. This variable can help increase the performance of the Select Waveform Database form by not checking if the specified paths are valid simulation results directories or files.

Syntax

`viva.browser ignorePathsForDirValidityCheck string "`*directoryPath*"

Values

directoryPath	Directory paths that are to be ignored. Default value: `/, /opt, /mnt, /home`

ignoreDirLevelForDirValidityCheck

Specifies the hierarchy level below which the results directory or files are to be ignored by the Select Waveform Database form while checking the validity of results directories or files. This variable can help increase the performance of the Select Waveform Database.

For example, if you specify this variable as `3`, then all the paths below `/././.` hierarchy level will be ignored for checking. The default value is `2`.

© 2023 Cadence Design Systems, Inc.

Syntax

```
viva.browser ignoreDirLevelForDirValidityCheck string "hierarchy"
```

Values

hierarchy	Hierarchy level below which paths are to be ignored Default value: 2

© 2023 Cadence Design Systems, Inc.

Calculator Variables

This section describes the following Calculator environment variables:

- usePreviousGuiSettings on page 972
- rpnMode on page 973
- clipSelectionMode on page 974
- displayContext on page 975
- plotStyle on page 976
- signalselection on page 977
- familyMode on page 978
- defaultCategory on page 979
- sizeKeyPad on page 980
- stackSize on page 981
- undoStackSize on page 982
- signalHistorySize on page 983
- xLocation on page 984
- yLocation on page 985
- width on page 986
- height on page 987
- showKeypad on page 988
- showStack on page 989
- mathToolBar on page 990
- trigToolBar on page 991
- schematicToolBar on page 992
- schematicAnalyses on page 993
- userButton on page 994
- restoreCustomCalcFunctions on page 995

© 2023 Cadence Design Systems, Inc.

usePreviousGuiSettings

Specifies whether the local defaults are to override the .cdsenv settings.

Syntax

```
viva.calculator usePreviousGuiSettings string "true" | "false"
```

Values

`true`	Local defaults for the following variables to override the `.cdsenv` settings: .`showKeyPad`, `showStack`, `rpnMode`, `clipSelectionMode`, `signalSelection`, `plotStyle`, `defaultCategory`, `mathToolBar`,`TrigToolBar`, `schematicToolBar`, `schematicAnalyses`, `userButton`, `xLocation`, `yLocation`, `width`, and `height`. This is the default value.
`false`	`.cdsenv` variable settings to override the local defaults.

© 2023 Cadence Design Systems, Inc.

rpnMode

Specifies whether the Calculator is in the rpn mode by default. You can change this setting through the Options menu in the Calculator window (*Options–Set RPN*).

Syntax

```
viva.calculator rpnMode string "true" | "false"
```

Values

`true`	Calculator is in rpn mode. This is the default value.
`false`	Calculator is in algebraic mode.

© 2023 Cadence Design Systems, Inc.

clipSelectionMode

Controls whether the *Clip* check box is selected by default in the Calculator window.

Syntax

```
viva.calculator clipSelectionMode string "true" |"false"
```

Values

`true`	The *Clip* check box is selected, hence the calculator works on the visible X-axis of a zoomed in trace. This is the default value.
`false`	The *Clip* check box is not selected, hence the calculator works on the complete trace.

© 2023 Cadence Design Systems, Inc.

displayContext

Specifies whether the test or results directory is displayed.

Syntax

```
viva.calculator displayContext string "true" | "false"
```

Values

`true`	*Test* and *Results Dir* are displayed. If there is no current test, only the results directory is displayed.
`false`	*Test* and *Results Dir* are not displayed.

© 2023 Cadence Design Systems, Inc.

plotStyle

Controls the plot destination for graphs from the Calculator window.

Syntax

```
viva.calculator signalselection string "append" |"replace" | "new Subwindow"
     |"New Window"
```

Values

`append`	The trace is appended to an existing graph. This is the default value.
`replace`	The trace replaces the existing trace.
`new Subwindow`	The trace is plotted to a new subwindow.
`new Window`	The trace is plotted to a new window.

© 2023 Cadence Design Systems, Inc.

signalselection

Controls the *Selection choices* in the Calculator window.

Syntax

```
viva.calculator signalselection string "off" |"wave" | "family"
```

Values

`off`	Selection choice in the Calculator window is set to off. This is the default value.
`wave`	Selection choice in the Calculator window is set to *wave*.
`family`	Selection choice in the Calculator window is set to *family*.

© 2023 Cadence Design Systems, Inc.

familyMode

Controls whether the *Family* button is selected by default in the Calculator window.

Syntax

```
viva.calculator familyMode string "true" | "false"
```

Values

true	The *Family* button is selected.
false	The *Family* button is not selected. This is the default value.

© 2023 Cadence Design Systems, Inc.

defaultCategory

Specifies the function category to be displayed.

Syntax

```
viva.calculator defaultCategory string "All" | "Favorites" | "Math" |
     "Modifier" | "Programmed Keys" | "RF Functions" | "Special Functions" |
     "Trigonometric" | "AWD Programmed Keys" | "SKILL Defined User Functions"
     | "Memories"
```

Values

`All`	Displays all the functions.
`Favorites`	Displays your favorite functions.
`Math`	Displays the math functions.
`Modifier`	Displays the modifier functions.
`Programmed Keys`	Displays the programmed keys.
`RF Functions`	Displays the RF functions.
`Special Functions`	Displays the special functions. This is the default value.
`Trigonometric`	Displays the trigonometric functions.
`AWD Programmed Keys`	Displays the AWD programmed keys.
`SKILL Defined User Functions`	Displays the SKILL defined functions.
`Memories`	Displays the memories you created.

© 2023 Cadence Design Systems, Inc.

sizeKeyPad

Controls the size of buttons of numeric keypad. It can be defined in small, medium or large sizes.

Syntax

viva.caculator sizeKeyPad string "small" "medium" "large"

Values

small	Button size will be small. This is default value
medium	Button size will be 1.5 times of small buttons.
large	Button size will be 2 times of small buttons.

© 2023 Cadence Design Systems, Inc.

stackSize

Controls the maximum number of expressions displayed in the Calculator stack. A scrollable list is displayed after this number is exceeded.

Syntax

viva.calculator stackSize string "*stack_number***"**

Values

stack_number	Number of expressions displayed in the Calculator. Default: 8 Valid values: Any positive integer

© 2023 Cadence Design Systems, Inc.

undoStackSize

Controls the maximum number of commands that can be undone.

Syntax

viva.calculator undoStackSize string "*undo_stack_number***"**

Values

undo_stack_number	Number of commands that can be undone. Default: 8 Valid values: 0–20

© 2023 Cadence Design Systems, Inc.

signalHistorySize

Controls the maximum number of items stored in the *Signal* field drop-down.

Syntax

viva.calculator signalHistorySize string "*signal_history_size***"**

Values

signal_history_size	Number of items stored in the *Signal* field drop-down in the function panel of the Calculator. Default: 8 Valid values: 0–20

© 2023 Cadence Design Systems, Inc.

xLocation

Controls the position where the Calculator window appears.

Syntax

viva.calculator xLocation string "*x_position***"**

Values

x_position	Horizontal distance of the Calculator window from the left of the screen. Default: 600 Valid values: A positive integer.

© 2023 Cadence Design Systems, Inc.

yLocation

Controls the position of the Calculator window.

Syntax

viva.calculator yLocation string "*y_position***"**

Values

y_position	Vertical distance of the Calculator window from the top of the screen. Default: 50 Valid values: A positive integer.

© 2023 Cadence Design Systems, Inc.

width

Controls the width of the calculator window.

Syntax

viva.calculator width string "*width_pixels***"**

Values

width_pixels	Width of the Calculator window. Default: 640 Valid values: A positive integer greater than 640.

© 2023 Cadence Design Systems, Inc.

height

Controls the height of the Calculator window.

Syntax

viva.calculator height string "*height_pixels***"**

Values

height_pixels	Height of the Calculator window. Default: 330 Valid values: A positive integer greater than 330.

© 2023 Cadence Design Systems, Inc.

showKeypad

Specifies whether the keypad is displayed.

Syntax

```
viva.calculator showKeypad string "true" | "false"
```

Values

`true`	Keypad is displayed. This is the default value.
`false`	Keypad is not displayed.

© 2023 Cadence Design Systems, Inc.

showStack

Specifies whether the stack (in RPN mode) or history (in Algebraic mode) is displayed.

Syntax

```
viva.calculator showStack string "true" | "false"
```

Values

`true`	Stack is displayed.
`false`	Stack is not displayed. This is the default value.

© 2023 Cadence Design Systems, Inc.

mathToolBar

Specifies whether the Math tool bar is displayed by default.

Syntax

```
viva.calculator schematicToolBar string "true" | "false"
```

Values

`true`	Math tool bar is displayed.
`false`	Math tool bar is not displayed. This is the default value.

© 2023 Cadence Design Systems, Inc.

trigToolBar

Specifies whether the trigonometric tool bar is displayed by default.

Syntax

```
viva.calculator schematicToolBar string "true" | "false"
```

Values

`true`	Trigonometric tool bar is displayed.
`false`	Trigonometric tool bar is not displayed. This is the default value.

© 2023 Cadence Design Systems, Inc.

schematicToolBar

Specifies whether the schematic access buttons are displayed.

Syntax

```
viva.calculator schematicToolBar string "true" | "false"
```

Values

`true`	Schematic access buttons are displayed. This is the default value
`false`	Schematic access buttons are not displayed.

© 2023 Cadence Design Systems, Inc.

schematicAnalyses

Controls the analyses for which the schematic access buttons are displayed. This variable is active only when the schematicToolBar variable is set to True.

Syntax

```
viva.calculator schematicAnalyses string "tran" | "ac" | "dc" | "sweptDc" |
     "info" | "noise" | "rf"
```

Values

`tran`	Displays the *vt* and *i* buttons.
`ac`	Displays *vf* and *if* buttons.
`dc`	Displays the *vdc* and *idc* buttons.
`sweptDc`	Displays the *vs* and *is* buttons.
`info`	Displays the *op*, *var*, *opt*, and *mp* buttons.
`noise`	Displays the *vn* and *vn2* buttons.
`rf`	Displays the *sp*, *zp*, *vswr*, *yp*, *hp*, *gd*, *zm*, and *data* buttons.

© 2023 Cadence Design Systems, Inc.

userButton

Associates a function with the *user* button.

Syntax

```
viva.calculator userButton1 string "abbreviation;function_name"
viva.calculator userButton2 string "abbreviation;function_name"
viva.calculator userButton3 string "abbreviation;function_name"
viva.calculator userButton4 string "abbreviation;function_name"
viva.calculator userButton5 string "abbreviation;function_name"
viva.calculator userButton6 string "abbreviation;function_name"
viva.calculator userButton7 string "abbreviation;function_name"
viva.calculator userButton8 string "abbreviation;function_name"
viva.calculator userButton9 string "abbreviation;function_name"
viva.calculator userButton10 string "abbreviation;function_name"
viva.calculator userButton11 string "abbreviation;function_name"
viva.calculator userButton12 string "abbreviation;function_name"
```

Values

abbreviation	Name to be displayed on the user button. You can enter up to 6 characters for the abbreviation.
function_name	Function to be associated with the *user* button.

Example

```
viva.calculator userButton1 string "bw;bandwidth"
```


© 2023 Cadence Design Systems, Inc.

restoreCustomCalcFunctions

```
viva.calculator restoreCustomCalcFunctions string {"true" | "false"}
```

Description

Specifies whether to disable loading of the custom functions from the `customFunctions.ini` file. Valid values are as follows:

- `true`: The custom functions are loaded from the `customFunctions.ini` file. This is the default value.
- `false`: The custom functions are not loaded from the `customFunctions.ini` file.

Yo must set this environment variable in `.cdsenv` file for the settings to take effect.

GUI Equivalent

None

Examples

```
envGetVal("viva.calculator" "restoreCustomCalcFunctions")
envSetVal("viva.calculator" "restoreCustomCalcFunctions" 'string "false")
```

useD2A

Controls how the calculation results are represented for digital buses. When this variable is set to `t`, the digital to analog (D2A) method is used, in which the results are displayed as a number scaled between `0-5V`. When this variable is set to `nil` (default value), the results are displayed as a string representing the bus value in binary format, for example `0b1z01`.

Syntax

```
calculator.bus useD2A boolean nil
```


© 2023 Cadence Design Systems, Inc.

Values

`t`	Results are displayed as a number scaled between `0-5V`.
`nil`	(Default value) Results are displayed as a string representing the bus value in binary format.

autoCorrectNoSpaceAfterMinus

Adds a space character after minus operator while evaluating the expressions of type `a -b`. By default, such expressions are automatically corrected and a warning message is displayed.

Syntax

```
viva.calculator autoCorrectNoSpaceAfterMinus string "true"
```

Values

`true`	(Default value) Space character is added automatically after the minus operator to autocorrect the expression.
`nil`	Space character is not added and evaluation of only first expression takes place. For example, only `a` is evaluated and plotted in the `a -b` expression.

Note: This variable works only for single expressions containing the unary minus operator, such as `a -b`. If you have nested expressions, auto-correction using this variable does not work. For example, this variable will not work for the following expression:
`func1(func2(func1((VT("out1 -VT("out"))arg1)arg1 arg2)arg1 ar2 arg3)`

maxEyePeriods

Specifies the maximum number of eye periods for the selected eye diagram. When the specified limit is reached, the eye diagram cannot be plotted and an error message is displayed.

Syntax

```
calculator.eyeDiagram maxEyePeriods string "1e6"
```


© 2023 Cadence Design Systems, Inc.

Default Value

`1e6`

Valid Values

Any string value

Example

```
calculator.eyeDiagram maxEyePeriods string "10e6"
```


© 2023 Cadence Design Systems, Inc.

Table Environment Variables

This section describes the following Table environment variables:

- notation
- significantDigits

notation

Specifies the notation to be used for the table data. The *Format Attributes* form in the table still displays the default value for notation.

Syntax

```
viva.table notation string "suffix"
```

Default Value

`suffix`

Valid Values

- `Engineering`
- `Scientific`

significantDigits

Specifies the significant digits to be used for the table data. The *Format Attributes* form in the table still displays the default value for significant digits.

```
viva.table significantDigits string "4"
```

Default value

`4`

Valid Values

Any integer value

© 2023 Cadence Design Systems, Inc.

B

Calculator Functions

This chapter describes the functions in the function panel for the SKILL mode. The following function categories are available in the SKILL mode:

- Special Functions
- Modifier Functions
- Trigonometric Functions
- Math Functions
- RF Functions
- Spectre RF Functions

Basic Steps For Running Calculator Functions

You need to perform the following steps in Virtuoso Visualization and Analysis XL to run the Calculator functions:

1. Open the results database in the Results Browser window. For information about how to open and access result databases in Results Browser, see Opening a Results Directory.
2. Right-click the signal you want to use as an input to the function and choose *Calculator*. Alternatively, you can plot the signal in the graph window and then send the plotted signal from graph to Calculator.
3. The expression for the selected signal is displayed in the Buffer. This signal expression can then be used as an input signal while running the functions. For some functions, you need to input more than one signals. In this case, you can use Stack to store the signals. For information about how to use Stack, see Stack on page 625.
4. Now, open the Function Panel and select the signal you want to run. For more information about how to use the Function Panel, see Function Panel on page 627.
5. If the function includes argument fields, a form appears in Function Panel where you can provide the argument values. By default, the default values are populated in the argument

© 2023 Cadence Design Systems, Inc.

fields. If signal does not require arguments, it is directly applied to the signal expression in the Buffer.

6. Click *OK*. The expression for the function is displayed in the Buffer.
7. Set the output plotting mode from the *Plotting Mode* drop-down on the *Selection* toolbar.
8. Then, click the Evaluate Buffer icon .

If the result is a scalar value, it will be displayed in the Buffer. If the result is a waveform, it is plotted in the graph window in the specified plotted mode.

Important

If you provide a digital logic signal as input in some Calculator functions, then upon evaluation, these functions return a value `5` to represent logic `1` signal and a value `0` to represent logic `0` signal. These values are hardcoded because it is not feasible to find out an analog value corresponding to a digtial logic value.

The affected functions include:
`clip`, `clipX`, `convolve`, `deriv`, `dft`, `dftbb`, `flip`, `frequency`, `gainMargin`, `iinteg`, `imag`, `integ`, `lshift`, `overshoot`, `phaseMargin`, `psd`, `psdbb`, `riseTime`, `root`, `settlingTime`, `slewRate`, `stddev`, `xmax`, `xmin`, `xval`, `ymax`, and `ymin`.

Note: When a Calculator function includes two input signal arguments, the first signal field in the function UI displays the content of the Buffer and the second signal field displays the first item on the Stack. Some examples of such functions are—`compare`, `delay`, `intersect`, `dftbb`, and so on.

© 2023 Cadence Design Systems, Inc.

Performing Floating-Point Comparison

A floating-point comparison function is used in many Calculator functions to find whether two floating-point values are equal within a relative tolerance limit. These functions include `clip`, `value`, `leafValue` and Set Sweep Range form in Result Browser.

When performing floating-point comparison, the equality between two doubles is determined by the following rules:

- If both the doubles are non-zero, the relative comparison is used to determine the equality.

  `abs(a-b)/min(a,b) < maxRelDiff`

  here,

  `maxRelDiff` is set to `10 * DBL_EPSILON` (approx. `2.22e-15`)

  `DBL_EPSILON` = `2^-52`, which is a predefined constant value.

- If one of the doubles is `0`, the absolute comparison is used. In this case, the doubles are considered equal if the other double is less than `maxRelDiff*maxRelDiff` (approx. `4.93e-30`).

Using Unary Minus Operator

A unary minus operator takes precedence over the minus operator. The following expressions show how these operators are used in the SKILL code:

- `a - b` is read as ((difference a b))
- `a -b` is read as (a (minus b))
- `a - -b` is read as ((difference a (minus b)))

For example, if you write two expressions, `VT(net1)` and `-VT(net2)`, they are treated as two different elements in SKILL.

© 2023 Cadence Design Systems, Inc.

Using Division Operator in the Expression for a Calculator Function

If you use the division operator in the expression for a calculator function, the expression returns `nil`, if the following conditions are met:

- The denominator contains an engineering suffix and the value of the denominator is less than 2G.

  Note: Numbers up to 2G are of type fixnum (integer).

- The value of numerator is less than the denominator.

For example, the expression `1/100M` returns `0` because the denominator (100M) contains the engineering suffix 'M' and the value of the denominator is less than 2G. Also, the value of numerator (1) is less than the value of the denominator (100M).

It is because of the integer division, that is, when you divide an integer by another integer, you get an integer as a result, not a floating-point number.

To get the desired result, you must change the expression to either `1.0/100M` or `1/100.0M`.

© 2023 Cadence Design Systems, Inc.

Special Functions

This section describes the following special functions available in the Virtuoso Visualization and Analysis XL Calculator:

- aaSP
- analog2Digital
- abs_jitter
- average
- bandwidth
- busTransition
- calcVal
- clip
- compare
- compression
- compressionVRI
- convolve
- cross
- d2a
- dBm
- delay
- delayMeasure
- deriv
- dft
- dftbb
- dnl
- dutyCycle
- evmQAM
- evmQpsk

© 2023 Cadence Design Systems, Inc.

- eyeDiagram
- eyeHeightAtXY
- eyePeakToPeakJitter
- eyeWidthAtXY
- eyeAperture
- fallTime
- firstVal
- flip
- fourEval
- freq
- freq_jitter
- frequency
- gainBwProd
- gainMargin
- getAsciiWave
- getData
- groupDelay
- harmonic
- harmonicFreq
- histogram2D
- i
- iinteg
- inl
- integ
- intersect
- ipn
- ipnVRI

© 2023 Cadence Design Systems, Inc.

- kurtosis
- lastVal
- loadpull
- lshift
- normalQQ
- normalQQPValue
- numConv
- overshoot
- pavg
- peak
- peakToPeak
- period_jitter
- phaseMargin
- phaseNoise
- PN
- pow
- prms
- psd
- psdbb
- pstddev
- pzbode
- pzfilter
- riseTime
- rms
- rmsNoise
- root
- rshift

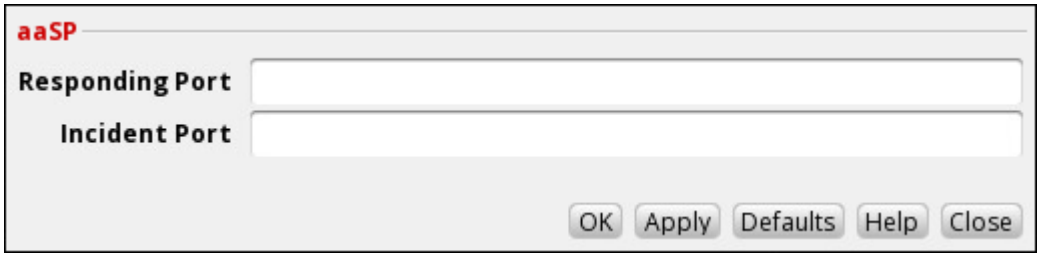
© 2023 Cadence Design Systems, Inc.

- sample
- settlingTime
- skewness
- slewrate
- spectralPower
- spectrumMeasurement
- stddev
- tangent
- thd
- triggeredDelay
- unityGainFreq
- v
- value
- waveVsWave
- xmax
- xmin
- xval
- ymax
- ymin

© 2023 Cadence Design Systems, Inc.

aaSP

Plots the S-Parameter waveform for a multi-port network. S-Parameters describe the input-output relationship between two ports in a multi-port network. Generally, S-parameters vary with frequency.



This function includes the following arguments:

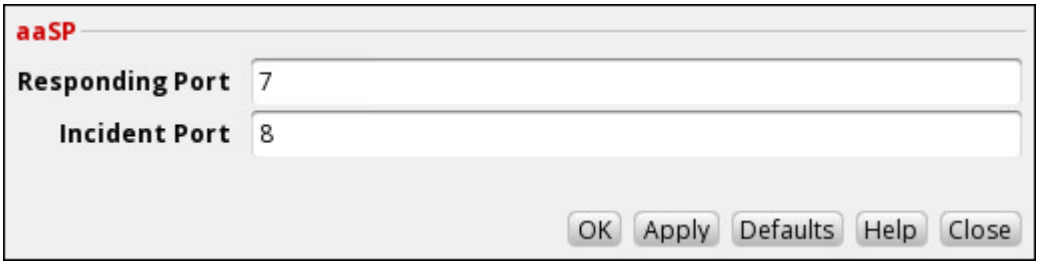
- *Responding Port*—The port number of the responding port.
- *Incident Port*—The port number of the incident port.

Example

The following example plots the `s78` parameter waveform, which is the response at port `7` due to a signal at port `8`.

Specify the following arguments:

- *Responding Port*—`7`
- *Incident Port*—`8`



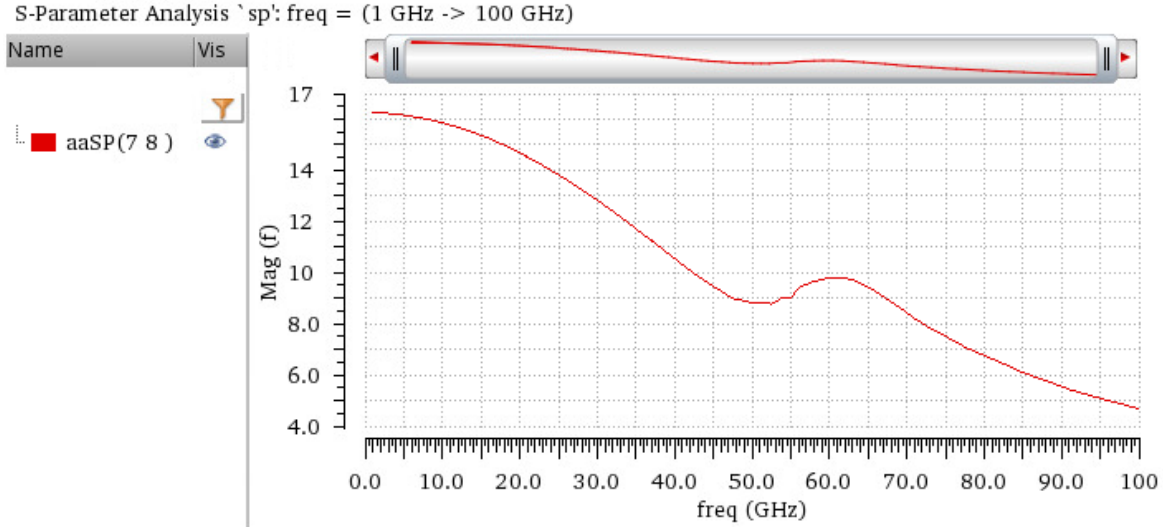
Note: Before using this function, you need to set the results database context to the results database that contains the S-parameter data file. For more information about how to change the in-context results directory, see Setting the In-Context Results Directory.

The following expression is created in the Buffer:

```
aaSP(7 8 )
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the following S-parameter waveform, which displays the variation of amplitude versus frequency.



© 2023 Cadence Design Systems, Inc.

analog2Digital

Returns the digital form of an analog input waveform, which may be a scalar, list or family of waveforms or a string representation of expressions. The corresponding SKILL command name for this function is `awvAnalog2Digital`.

This function includes the following arguments:

- *Waveform*—Name of the analog signal.
- *Logic Threshold*—Select the logic voltage threshold as `hilo` or `centre`. The default value is `hilo`.
- *Voltage High/Low*—When the *Logic Threshold* is set to `hilo`, provide the high and low or voltage values. Any value higher than the Voltage High is the high state and any value lower than the Voltage Low is the low state. The default value of these fields are `nil`. If you do not specify these values, they are calculated internally using the formula below:

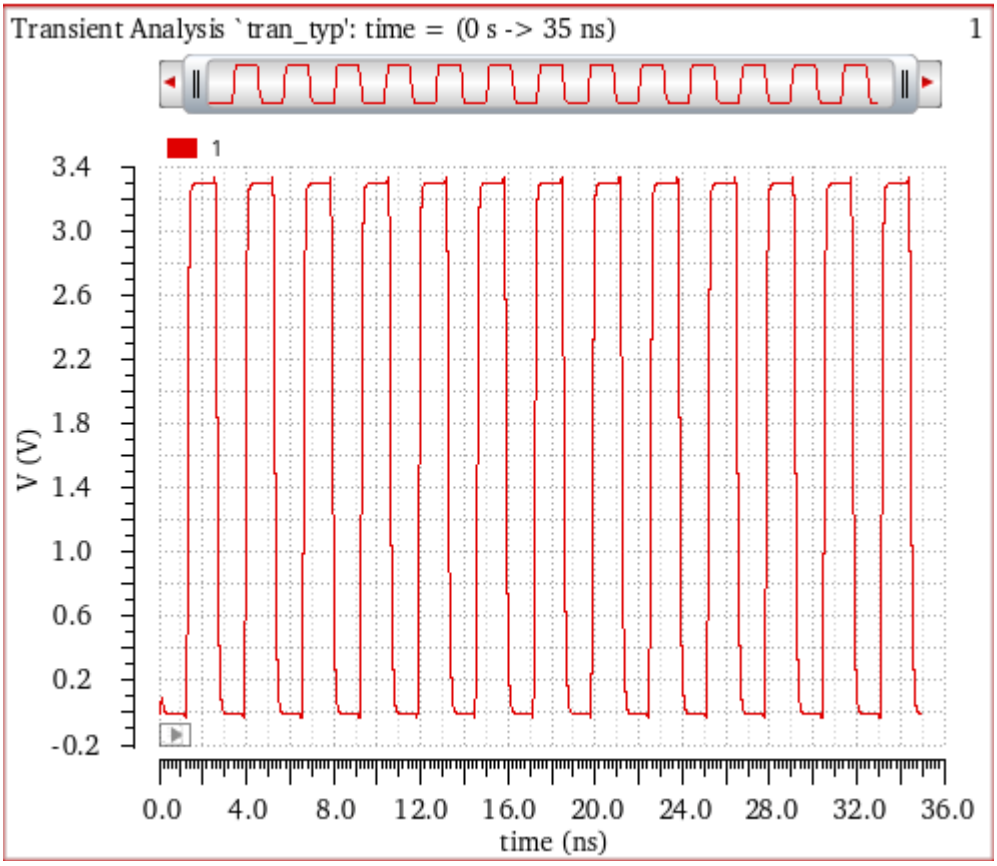
  ```
  vHigh = (topLine - bottomLine)*0.6 + bottomline

  vLow = (topLine - bottomLine)*0.4 + bottomline
  ```

- *Center Voltage*—When the *Logic Threshold* is set to `center`, provide the centre voltage value. Any value higher than the centre voltage is the high state and any value lower than the centre voltage is the low state. If you do not specify this value, it is calculated internally using `vCenter=average(wave)`.
- *Time to X*—The value that determines logic X. The timeX value is used to decide the state X. This field is available only when the threshold type is `hilo`. When threshold is `hilo`, the analog signal will be converted to logic X if:
  - analog signal value lies between vHigh and vLow
  - lapse time between vHigh and vLow is larger than timeX

© 2023 Cadence Design Systems, Inc.

Example 1

Consider the following input signal:



The `analog2Digital` function is applied on this input signal with the following arguments:

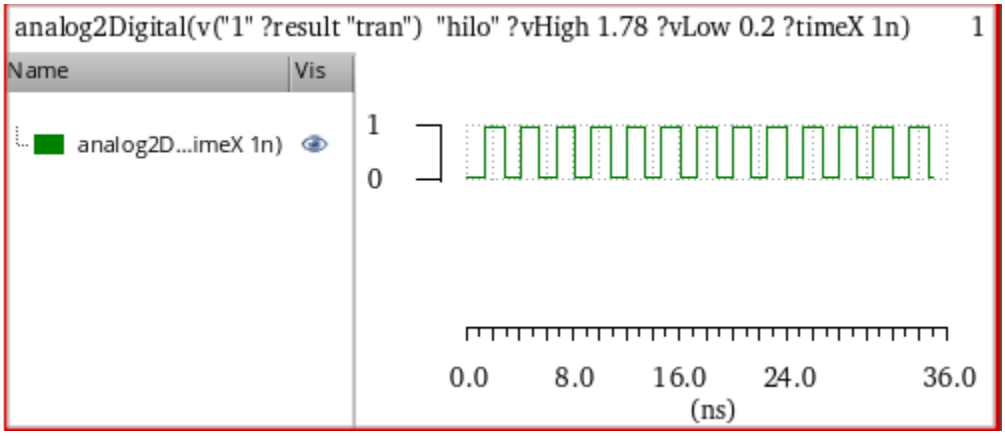
- *Signal*—`v("clk" ?result "tran" ?resultsDir "./mixed/test/adc_8bit.raw")`
- *Logic Threshold*—`hilo`
- *Voltage High/Low*—`1.78` and `0.2`
- *Centre Voltage*—`nil`
- *Time to X*—`1n`

The corresponding expression created in the Buffer is as follows:

```
analog2Digital(v("clk" ?result "tran" ?resultsDir "./mixed/test/
adc_8bit.raw")  "hilo" ?vHigh 1.78 ?vLow 0.2 ?timeX 1n)
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression and plot the results in the append mode, the given analog signal is converted into a digital signal as shown in the figure below.

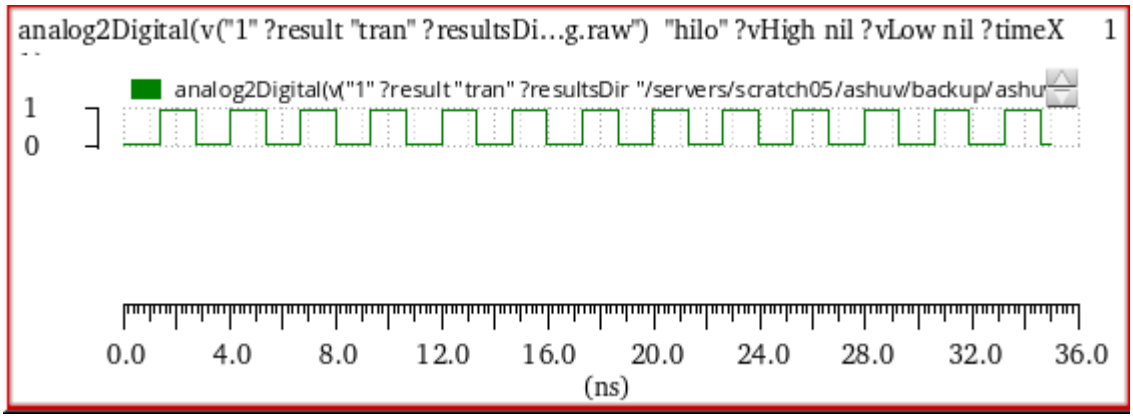


Example 2

This example shows the output waveform generated when the Logic Threshold is hilo, but no voltage high and voltage low values are specified. The following expression is created in the Buffer:

```
analog2Digital(v("1" ?result "tran")  "hilo" ?vHigh nil ?vLow nil
?timeX 1)
```

When you evaluate this expression, the following output waveform is generated.



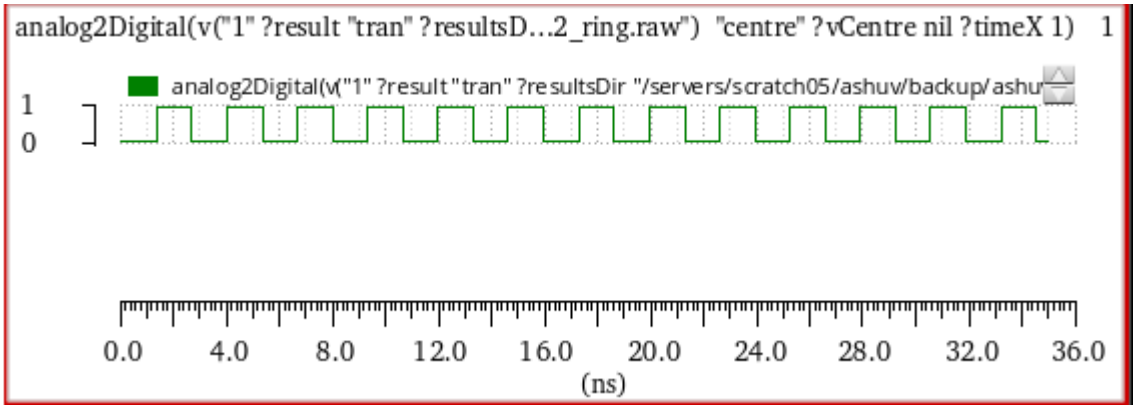
Example 3

This example shows the output waveform generated when the Logic Threshold is `center`, but no center voltage value is specified. The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
analog2Digital(v("1" ?result "tran")  "centre" ?vCentre nil ?timeX 1)
```

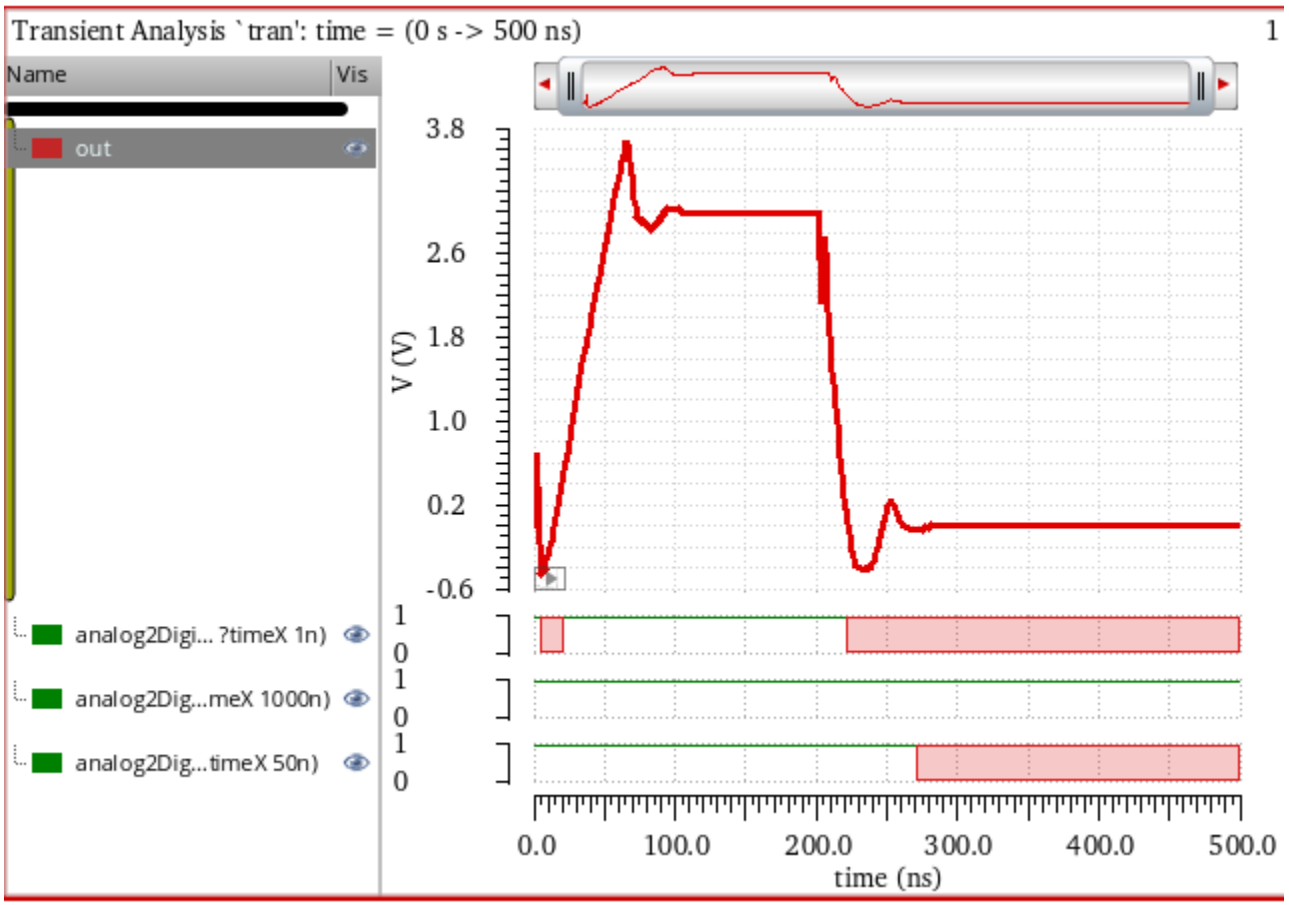
When you evaluate this expression, the following output waveform is generated.



© 2023 Cadence Design Systems, Inc.

Example 4

This example shows how the logix X of the generated digital waveform changes when you change the Time to X value. Consider the following waveform from a transient analysis. Run the `analog2Digital` function with *Voltage High/Low*=`250mV` and `1.0308`, and *Time to X*=`1ns`,`1000ns` and `50ns`, respectively.



Digital waveforms that are generated when you run the `analog2Digital` function for the given values of Time to X are displayed in the figure above.

- When *Time to X*=`1000ns`, the signal is in high state only because this value is large.
- When *Time to X*=`1ns`, the logix X of signal changes after `1ns` when the signal crosses the given voltage high or voltage low values.
- When *Time to X*=`50ns`, the logix X of signal changes after `50ns`

© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN command for `analog2Digital` is:

```
analog2Digital(
     o_wave
     t_thresholdType
     [ ?vhi n_vhi ]
     [ ?vlo n_vlo ]
     [ ?vc n_vc ]
     [ ?timeX n_timex ]
     )
     => o_digWave / n_digval / nil
```

For more information, see `analog2Digital` in OCEAN *Reference*.

© 2023 Cadence Design Systems, Inc.

abs_jitter

Returns a waveform that contains the absolute jitter values in the input waveform for the given threshold. The output waveform can be expressed in degrees, radians, or unit intervals (UI). The absolute jitter can be plotted as a function of cycle number, crossing time, or reference clock time.

The function is available only in the SKILL mode and includes the following fields:

- *Waveform*—Name of the waveform, expression, or family of waveforms. The expression displayed in the Buffer can be added to this field by selecting *buffer*.
- *Cross Type*—Specifies whether the jitter value can be calculated on rising (`rising`) or falling (`falling`) curves. Points at which the curves of the waveform intersect with the threshold.
- *Threshold*—Value at which the input waveform intersects to calculate the absolute jitter.

  Ensure that the specified threshold value ranges between ('ymin,'ymax) Y-axis coordinates and input waveform crosses the threshold value more than once. Otherwise, the funcion will display an error message.

  Note: With the tolerance value, the signal must cross the threshold and reach `threshold ± tolerance` to consider the event as a crossing event. The tolerance value is calculated as `0.1*stddev(signal)`, where the standard deviation is calculated using the y data points of the input signal.
- *X-Unit*—Unit defined for X-axis of the output waveform. Specify whether you want to output the absolute jitter against *time* or *cycle*. Cycle numbers refer to the nth occurrence where the waveform crosses the given threshold. When you select time (s), the absolute jitter is calculated against the reference clock time corresponding to the specified *Tnom* value.

© 2023 Cadence Design Systems, Inc.

- *Y-Unit*—Unit defined for Y-axis of the output waveform. Specify whether you want to calculate the phase in seconds (*s*), radians (*rad)*, or unit intervals (*UI*).
- *Tnom*—Nominal time period of the input waveform. The waveform is expected to be a periodic waveform that contains noise. If you do not enter the *Tnom* value, the `abs_jitter` function autocalculates the approximate time period of the input waveform by using the following equation:

  ```
  (lastCrossing-firstCrossing)/(numCrossings-1)
  ```

  where,

  - Crossing times are determined by the time at which the specified threshold is crossed.
  - `numCrossings` determines the number of crossings.
- *Zero Ref*—Defines the timing of the first edge of the ideal reference clock used for absolute jitter calculation. The reference clock period is equal to `Tnom`, the k-th edge is at time `ZeroRef + k * Tnom`. When this argument is `nil` or not specified, its default value is equal to the time of the first crossing of the input waveform so that the first value of the jitter is zero.

  For the driven circuit, *Tnom* and *Zero Ref* should be consistent with periodic input source. For autonomous circuit, average period and average jitter can be used as *Tnom* and *Zero Ref*.

Additional Information

The absolute jitter can be defined as follows:

For a given waveform $v(t), t_{start} \le t \le t_{end}$, with the following properties:

- Oscillating with expected nominal period, `T`,
- Between minimum and maximum values, $v_{min} \le v(t) \le v_{max}$,
- Rising and falling through a given threshold, v_{th}
- In time intervals, $t_k; 0 \le k \le N$

The absolute jitter for the waveform can be defined as:

$$J_a(k) = t_k - k \cdot T$$

© 2023 Cadence Design Systems, Inc.

The period jitter of the waveform can be defined as:

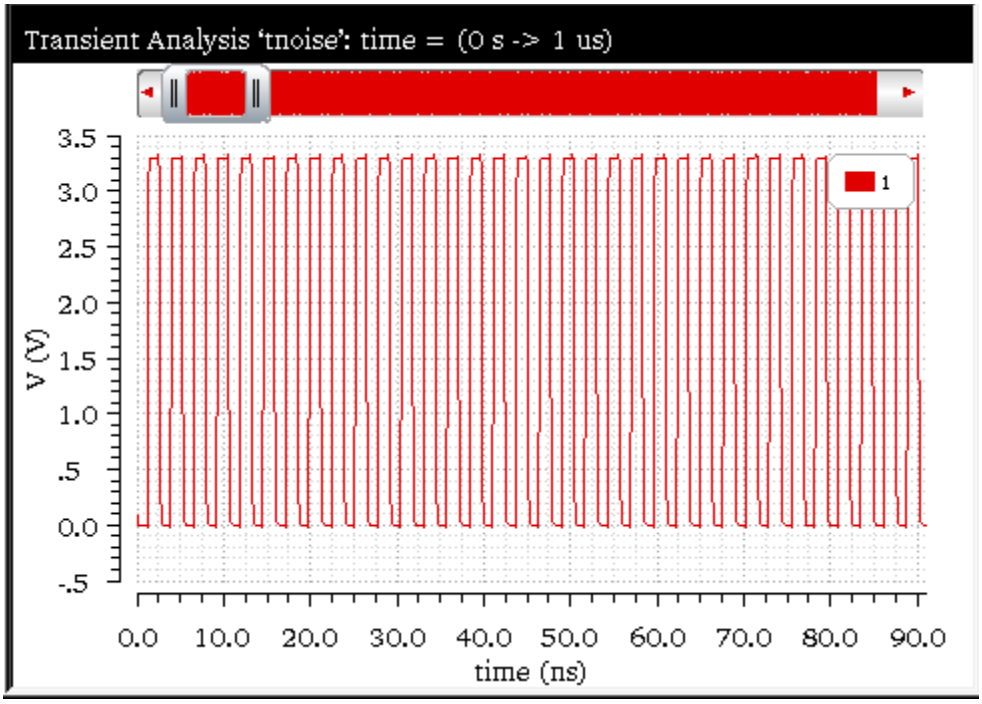
$$J_p(k) = t_k - t_{k-1} - T = J_a(k) - J_a(k-1) = \frac{d}{dk} J_a(k)$$

The jitter can be expressed in units of time (seconds) or in units of phase (radians or unit intervals). These values can be converted by using the following formula:

$$J_a[s] = \frac{J}{T}[UI] = \frac{J}{T} \cdot 2\pi[rad]$$

Example

Consider the following input waveform:



The `abs_jitter` function is applied to this waveform with the following arguments:

- *Waveform*—`v("1" ?result "tran")`
- *Cross Type*—`rising`
- *Threshold*—`1.5`

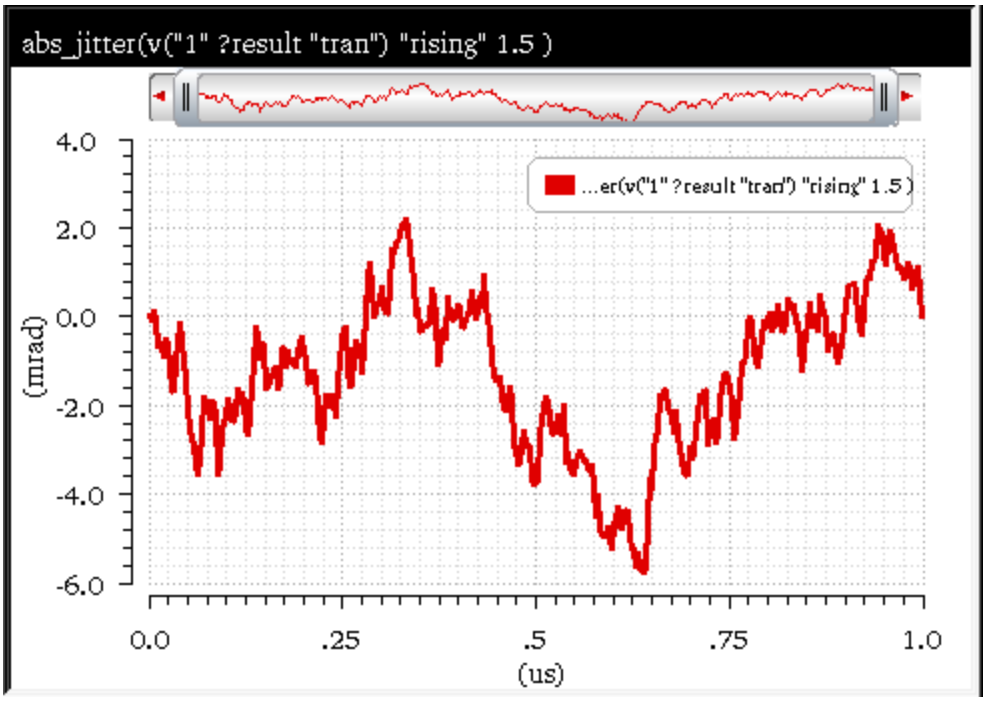
© 2023 Cadence Design Systems, Inc.

- *X-unit*—`s`
- *Y-unit*—`rad`
- *Tnom*—`nil`
- *Zero Ref*—`nil`

The expression created in the Buffer is as follows:

```
abs_jitter(v("1" ?result "tran" ?resultsDir "nand2_ring_tnoise.raw")
"rising" 1.5)
```

When you evaluate this expression, the following output waveform showing the absolute jitter values is displayed in the new graph window:



Related OCEAN Function

The equivalent OCEAN command for `abs_jitter` is:

```
abs_jitter(o_waveform t_crossType n_threshold
     ?xUnit t_xUnit ?yUnit t_yUnit ?Tnom n_Tnom)
     => o_waveform/nil
```

For more information, see abs_jitter in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

average

Computes the average of a waveform over its entire range.

This function can be applied to both scalar and vector inputs.

If `y=f(x)`:

For a continuous signal, the average is calculated as the integral of the expression `f(x)` over the range of `x`, divided by the range of `x`.

$$Average(y) = \frac{\int_{from}^{to} f(x)dx}{to - from}$$

For a discrete signal, the average is calculated by dividing the sum of all `y` values by the total number of data points.

$$Average(y) = \frac{\sum_{from}^{to} f(x)}{N}$$

where, *`from`* is the initial value for `x` and *`to`* is the final value of `x`, and *`N`* is the total number of data points.

You need not explicitly specify the `from` and `to` values when calculating the average amplitude of a waveform using the `average` function because the function takes the length of the entire waveform.

If you want to calculate the average only for a part of a waveform, you can first use the `clip` function to specify the `from` and `to` values, and then apply the `average` function on the clipped part of the waveform.

```
clip(v("out" ?result "tran-tran") 100n 400n )
average(clip(v("out" ?result "tran-tran" ) 100n 400n ))
```


© 2023 Cadence Design Systems, Inc.

Where, `v("out" ?result "tran-tran")` is the waveform of the signal, and `100n` and `400n` are the `from` and `to` values, respectively.

This function includes the following arguments:

- *Signal*—Displays the name of the signal.
- *continuous*—Specifies whether to consider the signal as a continuous signal. For a continuous signal, the average is calculated using the integral. If you set it to *no*, the signal is considered as a discrete signal. For a discrete signal, the average is calculated by dividing the sum of all `y` values with the total number of data points.

Additional Information

When you use the `average` function to create expressions that are to be measured across corners, the function includes an additional argument, `overall`. The expression created is as follows:

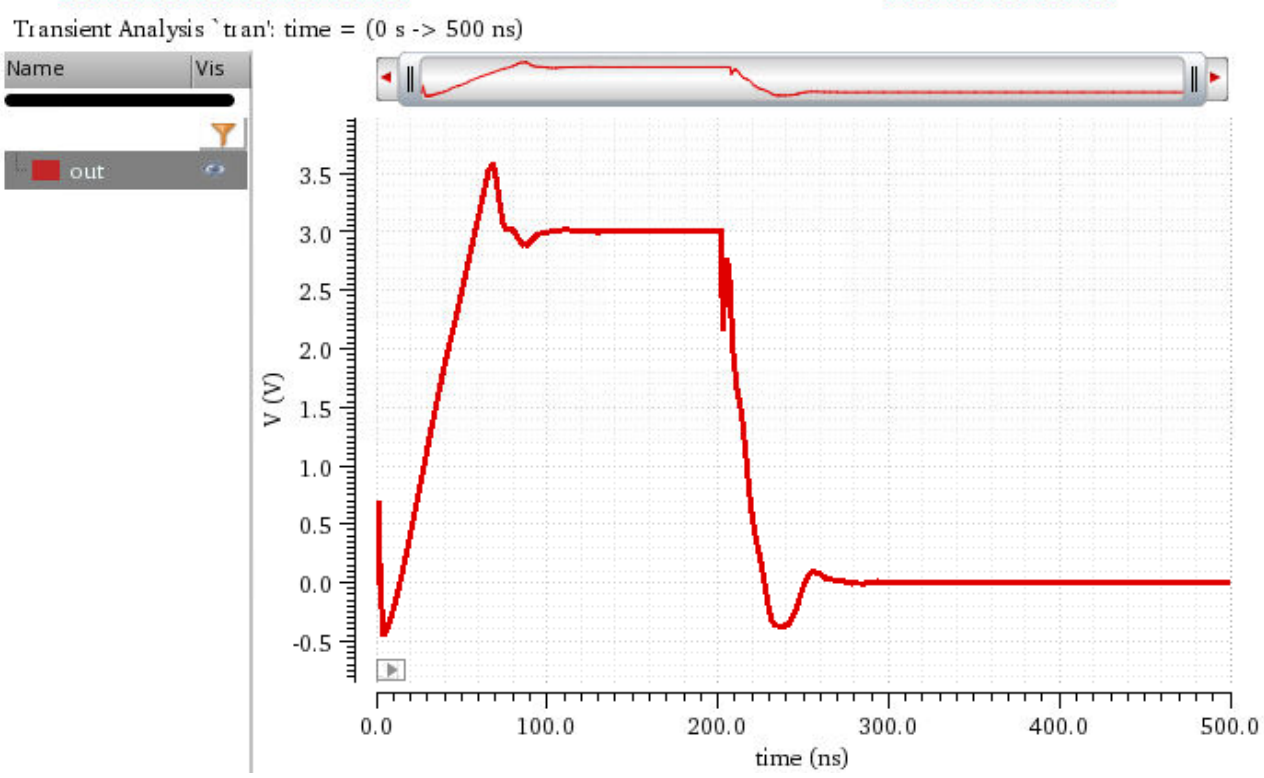
```
average(v("out" ?results "tran") ?overall t)
```

When the `overall` argument is set to `t`, it performs the calculation on the results of corner simulations for each design point that are treated as discrete values for evaluation. When set to `nil`, it creates a waveform from the data points and then calculates the average value of the waveform. If the input is a scalar and the `overall` argument is set to `t`, the `ymax` function returns the input scalar value.

© 2023 Cadence Design Systems, Inc.

Example (continuous = yes)

This example shows how the average amplitude is calculated when you apply the `average` function on the following input signal:



The following arguments are specified in this example:

- *Signal*—`v("out" ?result "tran")`
- *continuous*—`yes`

When you apply the `average` function on this signal, the following expression is created in the Buffer:

```
average(v("out" ?result "tran-tran"))
```

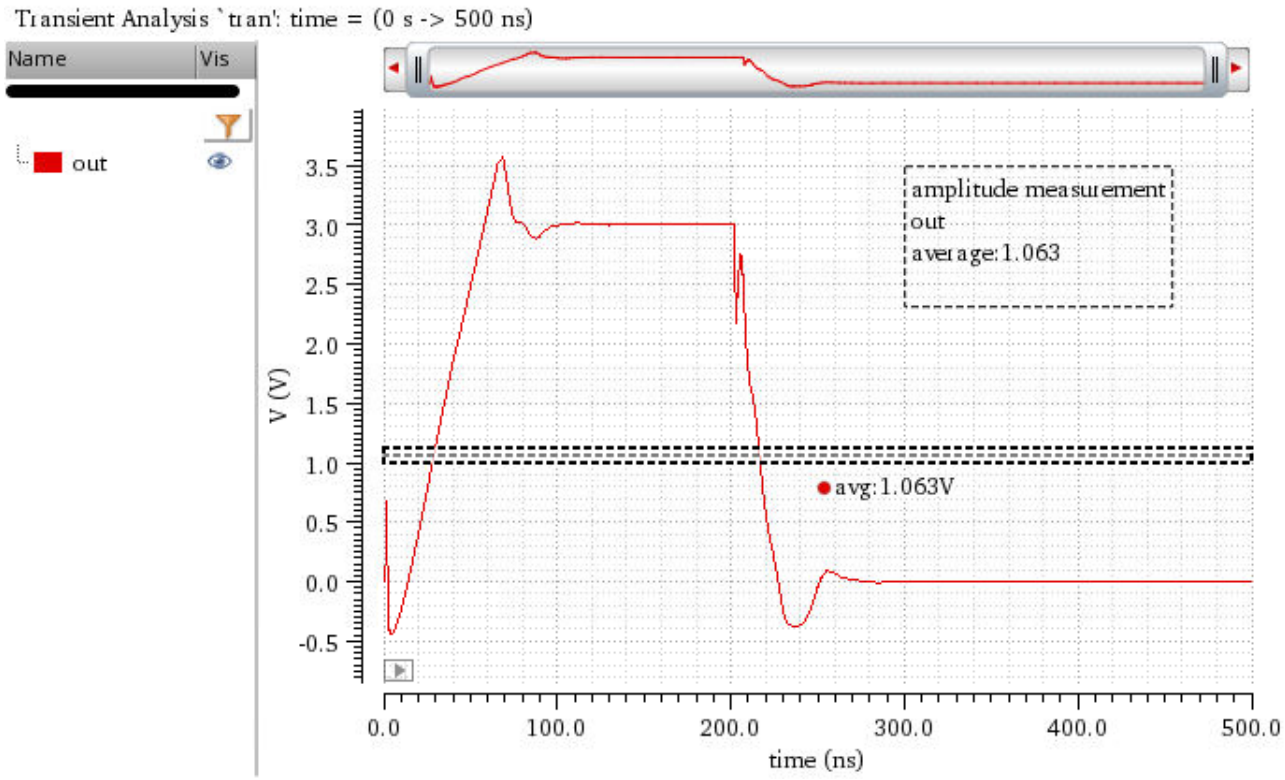
Now, when you evaluate this expression, the following output is displayed in the Buffer:

```
1.063
```

Note that the average amplitude of the signal is calculated over its entire range (entire length of the waveform), that is from `0-500ns`.

© 2023 Cadence Design Systems, Inc.

You can also use the *Amplitude* tab of the Direct Measurements assistant to calculate and annotate the average level of the signal.

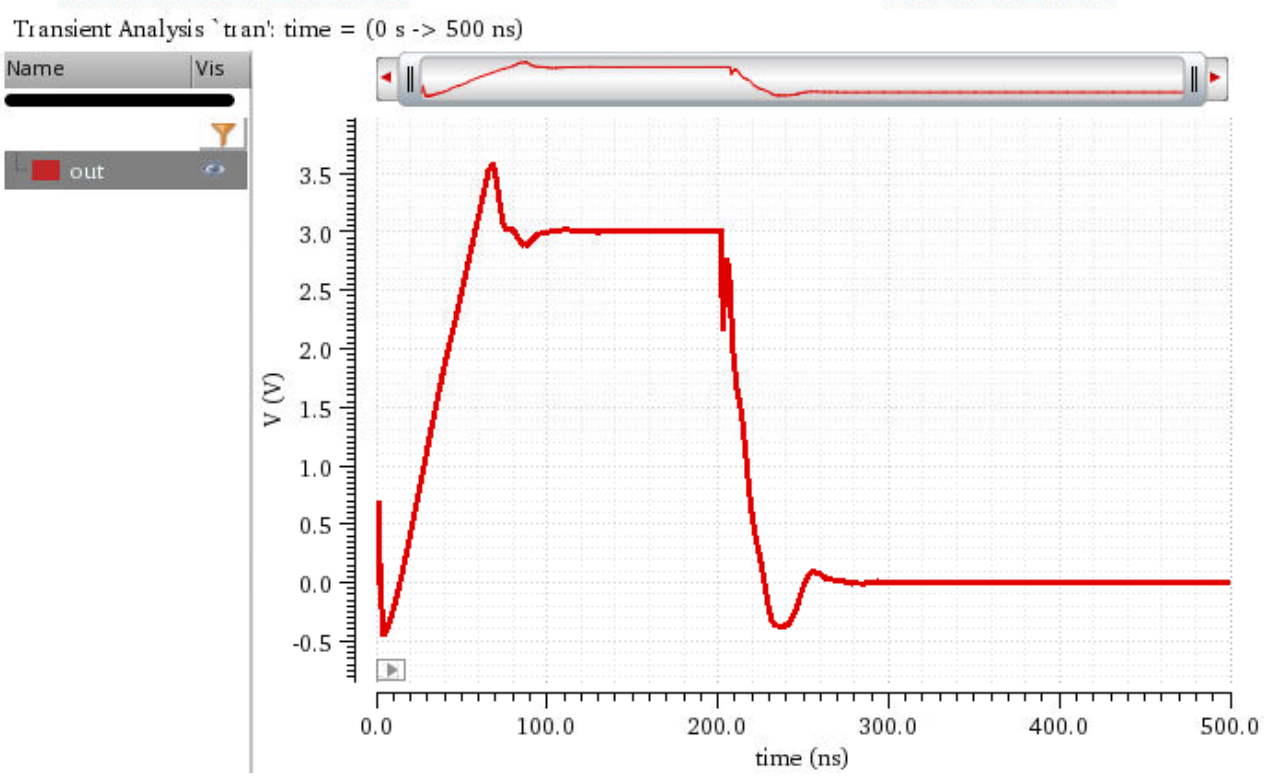


For more information, refer to Specifying the Measurement Region and Measuring the Amplitude of a Signal.

© 2023 Cadence Design Systems, Inc.

Example (continuous = no)

This example shows how the average amplitude is calculated when you apply the `average` function on the following input signal:



The following arguments are specified in this example:

- *Signal*—`v("out" ?result "tran")`
- *continuous*—`no`

When you apply the `average` function on this signal, the following expression is created in the Buffer:

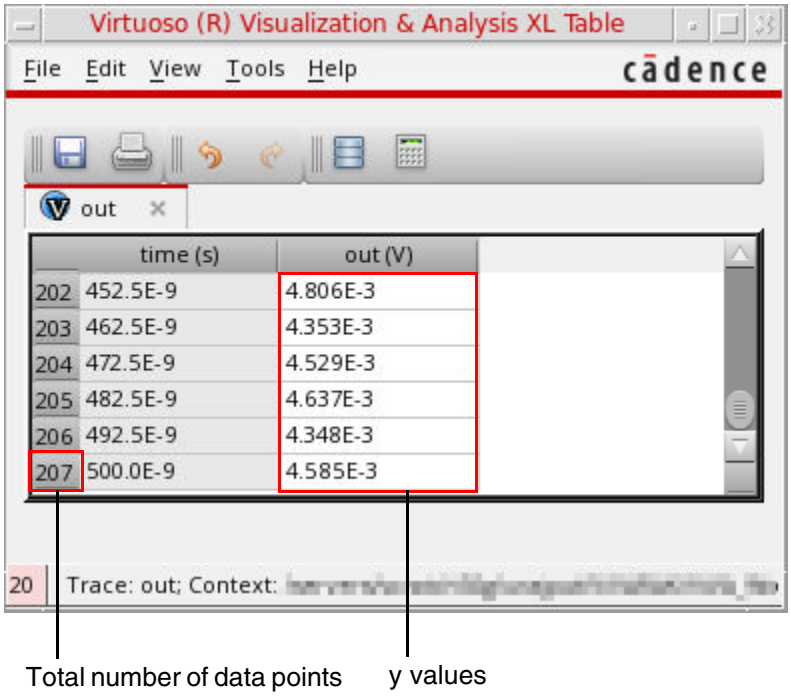
```
average(v("out" ?result "tran-tran")?continuous nil)
```

Now, when you evaluate this expression, the following output is displayed in the Buffer:

If you send this signal to Virtuoso Visualization and Analysis XL Table, you will observe that it has `207` data points and the sum of all `y` values is `231`. If you manually calculate the

© 2023 Cadence Design Systems, Inc.

average by dividing the sum of all `y` values with the total number of data points (`231/207`), it is equal to `1.11`.



Related OCEAN Function

The equivalent OCEAN function for `average` is:

```
average( o_waveform )
     => n_average/o_waveformAverage/nil
```

For more information, see average in *OCEAN Reference*.

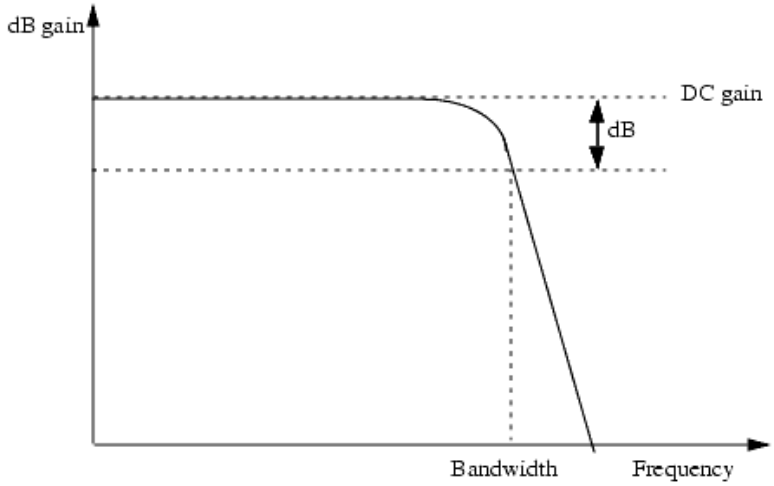
© 2023 Cadence Design Systems, Inc.

bandwidth

Measures the bandwidth of a low/high/band-pass freuqnecy-domain signal. Bandwidth is the frequency where the gain drops by 3dB (by default) as measured by the dB20 function.

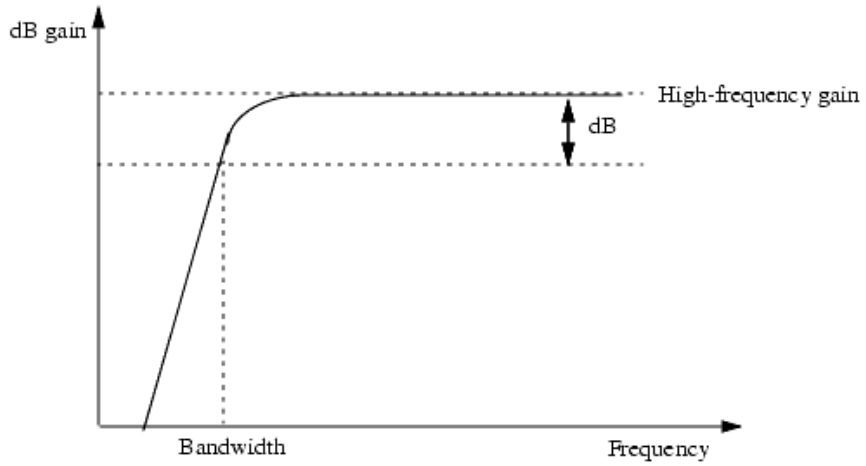
This function includes the following fields:

- *Signal* —Name of the signal. In SKILL mode, Virtuoso Visualization and Analysis XL wraps the signal with the `mag`, `dB`, or `log` function.
- *Db*—Decibels down from the peak, which means how far below the peak value you want to see data. In SKILL mode, *Db* is a number equal to or greater than zero.
- *Type*—Response type.
  - When `low`, computes the low-pass bandwidth by determining the smallest frequency at which the magnitude of the input waveform drops *`Db`* decibels below the DC gain. DC gain is obtained by zero-order extrapolation from the lowest or highest computed frequency, if necessary. An error occurs if the magnitude of the input waveform does not drop *`Db`* decibels below the DC gain.



© 2023 Cadence Design Systems, Inc.

- When `high`, computes the high-pass bandwidth by determining the largest frequency at which the magnitude of the input waveform drops *db* decibels below the gain at the highest frequency in the response waveform. An error occurs if the magnitude of the input waveform does not drop n decibels below the gain at high frequency.



- When `band`, computes the band-pass bandwidth by:

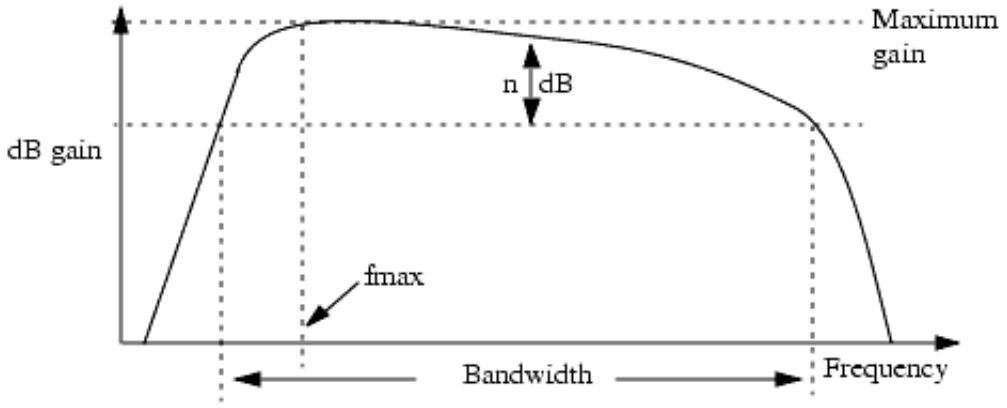
  a. Determining the lowest frequency (f_{max}) at which the magnitude of the input waveform is maximized

  b. Determining the highest frequency less than f_{max} at which the input waveform magnitude drops *Db* decibels below the maximum

  c. Determining the lowest frequency greater than f_{max} at which the input waveform magnitude drops *Db* decibels below the maximum

© 2023 Cadence Design Systems, Inc.

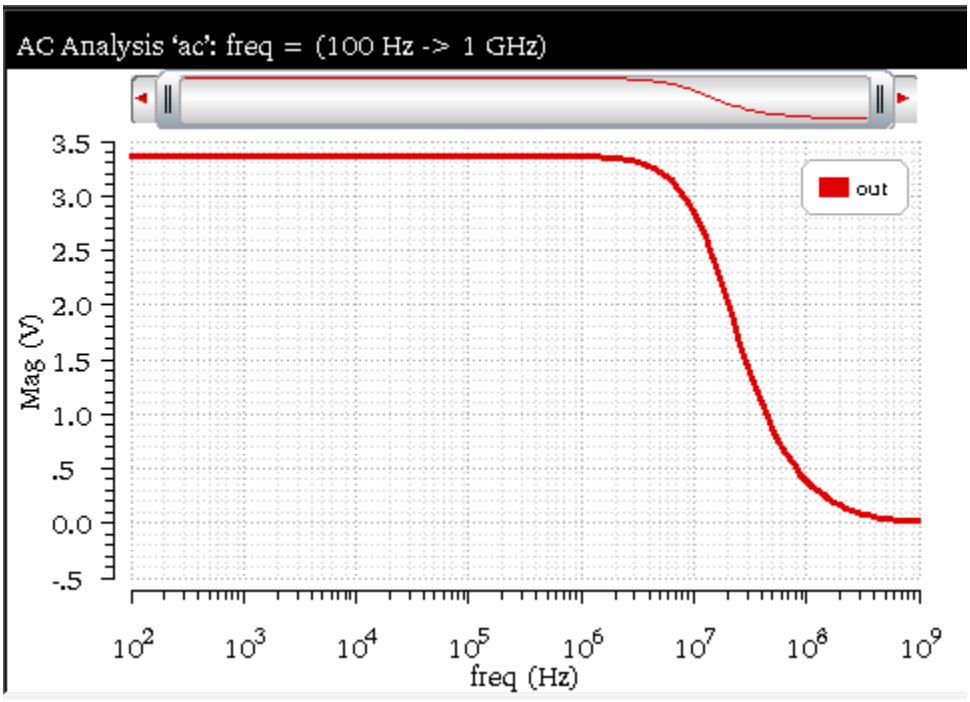
d. Subtracting the value returned by - *step b* from the value returned by step c. The value returned by step b or step c must exist.



Note: The bandwidth function includes the *Db* option; however, the signal is magnitude. This function modifies the magnitude signal to db scale internally.

Example

This example calculates the bandwidth of the following AC voltage signal, `v("out" ?result "ac")`



© 2023 Cadence Design Systems, Inc.

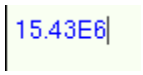
The following arguments are specified in this example:

- *Signal*—`v("out" ?result "ac")`
- *Db*—`3`
- *Type*—`low`

The following expression is created in the Buffer:

```
bandwidth(mag(v("out" ?result "ac" ?resultsDir "./ampTest.raw")) 3
"low" )
```

Now, when you evaluate this expression, the following output value is displayed in the Buffer as result:



Related OCEAN Function

The equivalent OCEAN function for `bandwidth` is:

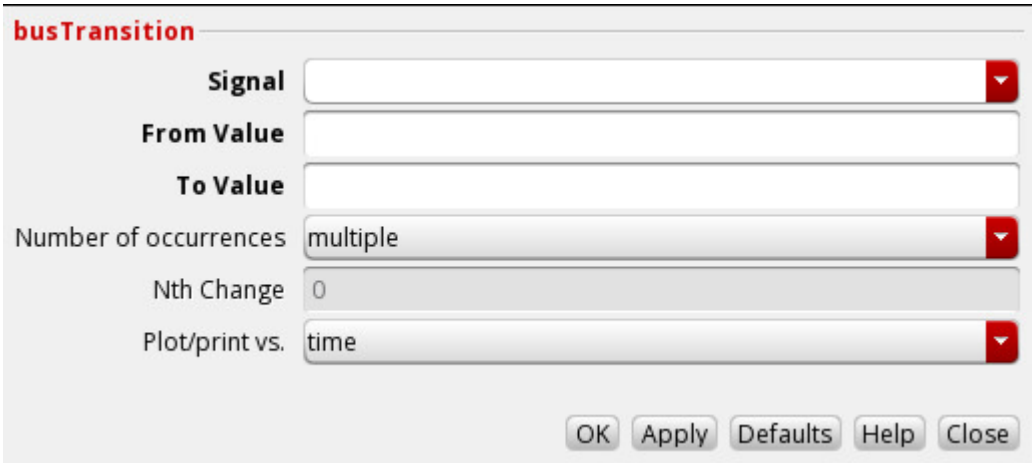
```
bandwidth( o_waveform n_db t_type )
     => n_value/o_waveform/nil
```

For more information about this OCEAN function, see bandwidth in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

busTransition

Returns the time when a bus value is changed from one specified value to another specified value.



This function includes the following arguments:

- *Signal*—Name of the bus signal.
- *From Value*—The starting value from which the bus change is to be calculated. The prefixes are used to define the formats, such as 0b for binary, 0 for octal, 0x for hexadecimal. No prefix indicates decimal format.
- *To Value*—The target value to which the bus is changed. The prefixes are used to define the formats, such as `0b` for binary, `0` for octal, `0x` for hexadecimal. No prefix indicates decimal format.
- *Number of occurrence*s—Select `single` to calculate one point or select `multiple` to calculate until the end of X range. Specify the *Nth Change* argument if you select `single`. Select the independent variable against which you want to plot the bus changes from the *Plot/print vs* drop-down list if you select `multiple`.
- *Nth Change*—When n > 0, function returns the nth X value when the bus is changed, when n<0, returns the last nth value. When n=0 (default value), returns the change of bus for all X values.

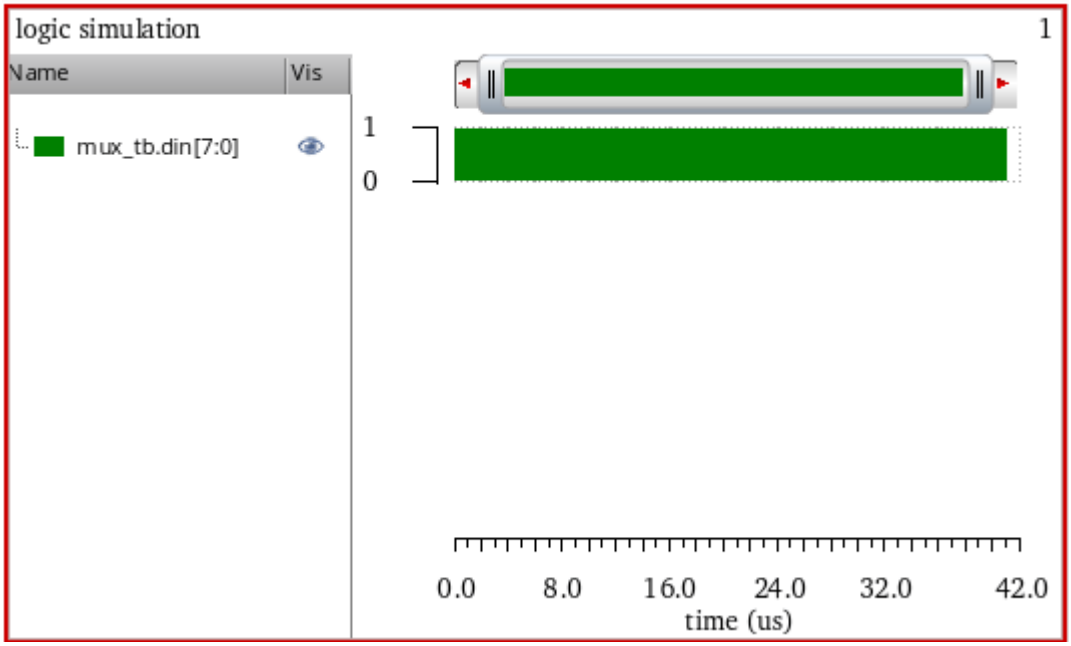
  Note: This argument is not available if *Number of occurrences* is set to `multiple`.
- *Plot/print vs*—Determines the X units of the generated waveform. Valid values: `time` and `cycle`. Default value: `time`

  Note: This argument is not available if *Number of occurrences* is set to `single`.

© 2023 Cadence Design Systems, Inc.

Example

The example below shows the output values and waveforms generated when you apply `busTransition` function on the following input bus waveform, `mux_tb.din[7:0]`:



Now, specify the following argument values for this function:

- *Signal*—`getData("mux_tb.din[7:0]" ?result "sst2LargeBus")`
- *From Value*—`0x10`
- *To Value*—`0x11`
- *Nth Change*—`1`
- *Number of occurrences*—`single`
- *Plot/Print vs.*—`time`

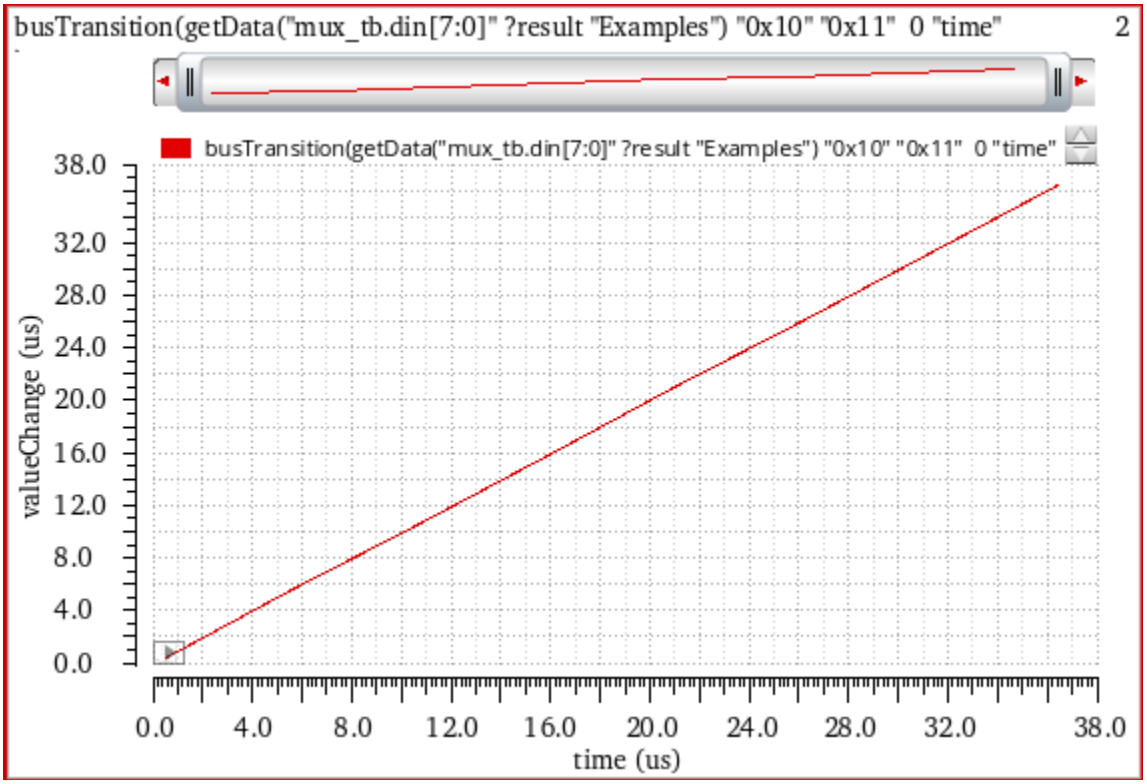
After you click *OK* to apply the `busTransition` function, the following expression is created in the Buffer:

```
busTransition(getData("mux_tb.din[7:0]" ?result "sst2LargeBus")
"0x10" "0x11"  1))
```


© 2023 Cadence Design Systems, Inc.

Now, when you click the evaluate button to run this function, a scalar value is displayed in the buffer, as shown in the figure below:

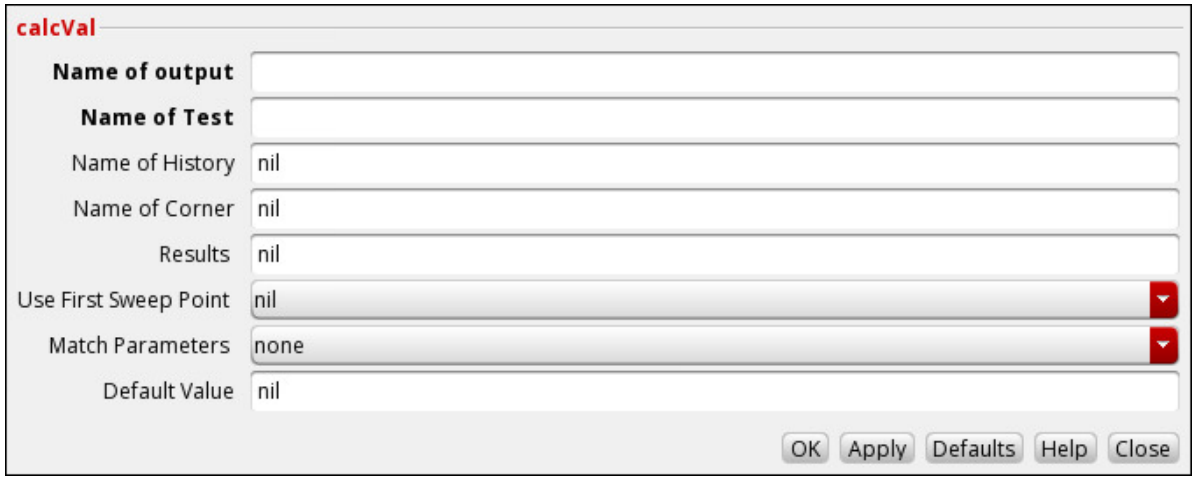
If you specify the argument value of *Number of occurrence* as `multiple` and run the `busTransition` function, an output waveform is displayed in the graph window, as shown in the figure below.



© 2023 Cadence Design Systems, Inc.

calcVal

Stores the output value from one test and passes it on to another test.



This function includes the following arguments:

- *Name of output*—The name of the output to be used.
- *Name of Test*—The name of the test that contains the output.
- *Name of History*—The name of the history from which the results are to be retrieved.

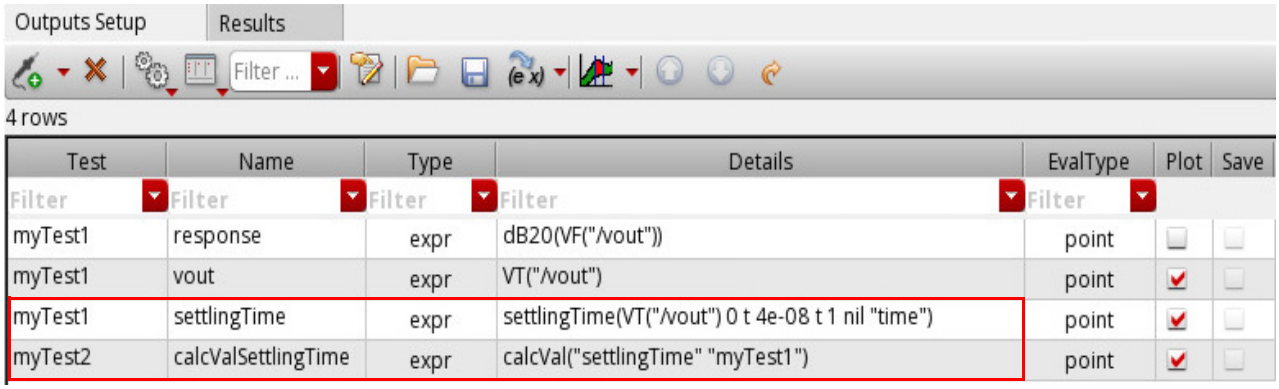
  Note: If you do not specify the history, the value of the output from the current simulation run is used.
- *Name of Corner*—The name of the corner.
- *Results*—The name of the analysis from which the `signal` type output is to be retrieved.

  Note: Specify this argument only for the outputs of type `signal`.
- *Use First Sweep Point*—Specifies that when sweep values are used, the result of only the first swept point is to be used for calculation.
- *Match Parameters*—Specifies whether to look for points that have matched variables and parameters in the current run and the source run. The source run can be of any other test in the current history, or the same or another test in another history.
- *Default Value*—The value to be returned when `calcVal` evaluation fails for any reason.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following example:

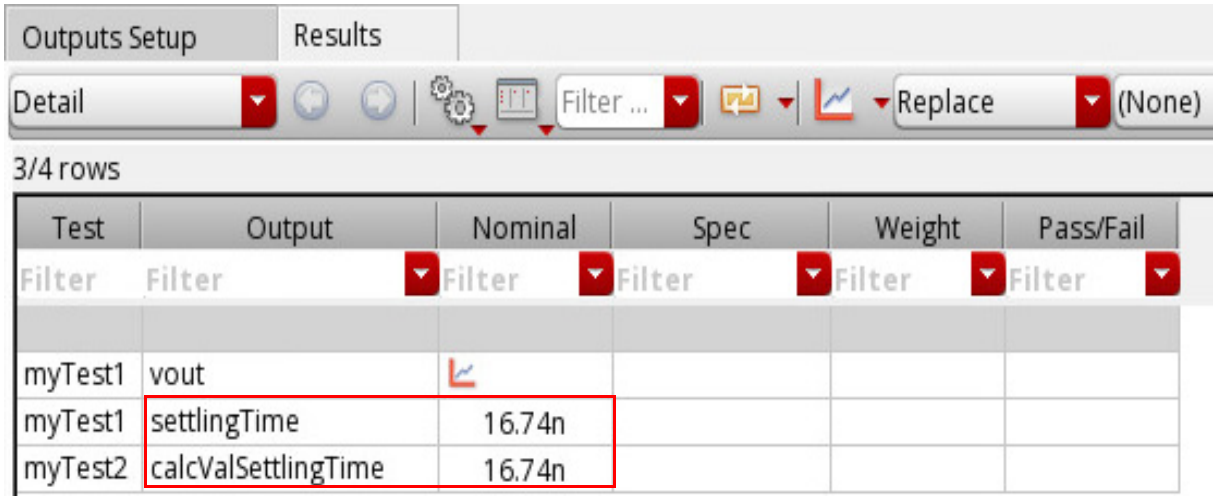


Test	Name	Type	Details	EvalType	Plot	Save
Filter	Filter	Filter	Filter	Filter		
myTest1	response	expr	dB20(VF("/vout"))	point	☐	☐
myTest1	vout	expr	VT("/vout")	point	☑	☐
myTest1	settlingTime	expr	settlingTime(VT("/vout") 0 t 4e-08 t 1 nil "time")	point	☑	☐
myTest2	calcValSettlingTime	expr	calcVal("settlingTime" "myTest1")	point	☑	☐

There are two tests, `myTest1` and `myTest2` in the *Output Setup*. `myTest1` has an output, `settlingTime` of type expression.

`myTest2` has an output expression `calcValSettlingTime` defined using the `calcVal` expression, which stores the value of the `settlingTime` output calculated from the `myTest1` test.

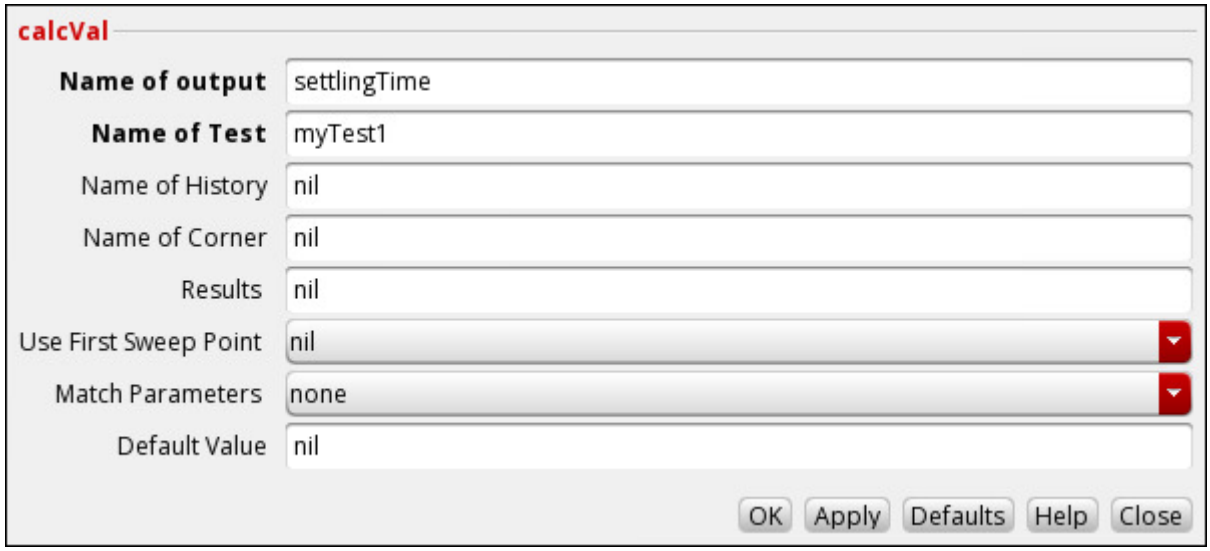
When you run the simulation, the `calcValSettlingTime` expression returns the same value that `settlingTime` returns, which is `16.74n`, as shown in the following figure.



Test	Output	Nominal	Spec	Weight	Pass/Fail
Filter	Filter	Filter	Filter	Filter	Filter
myTest1	vout				
myTest1	settlingTime	16.74n			
myTest2	calcValSettlingTime	16.74n			

© 2023 Cadence Design Systems, Inc.

Now, calculate value of the `settlingTime` output from the `myTest1` test using the `calcVal` Calculator function. Specify the arguments, as shown in the following figure.



The following expression is created in the Buffer:

```
calcVal("settlingTime" "myTest1" )
```

When you evaluate this expression, the function returns the same value, `16.74E-9`.

Note: For more information about the `calcVal` function, see calcVal in *Virtuoso ADE SKILL Reference*.

© 2023 Cadence Design Systems, Inc.

clip

Returns the portion of a signal between two points along the x axis.You can use the `clip` function to restrict the range of action of other special functions of the calculator such as `integ`, `rms`, and `frequency`.

Note: The `clip` function works only for sorted data values. If you run this function on unsorted data values, an error message is displayed. The `clip` function is not supported for DC hysteresis sweep analysis.

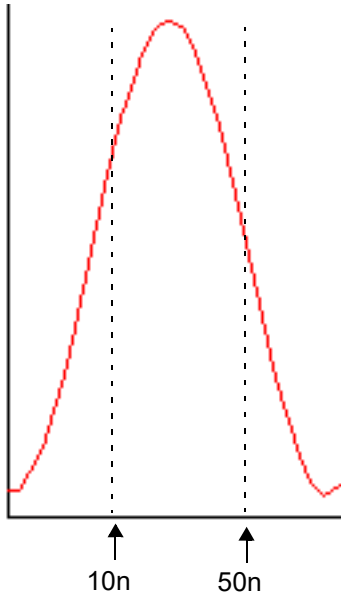
This function includes the following arguments:

- *Signal*—Displays the name of the signal.
- *From*—Starting point on the x axis from where the clipping is to be started.
- *To*—Ending point on the x axis where the clipping is to be ended.
- *interpolate*—Specifies whether to use the interpolated values for the *From* and *To* fields in the clipped result when their specified values do not correspond to any actual data points. The default is *yes*. If you set it to *no*, the result will include only the actual data points between *From* and *To* values.

© 2023 Cadence Design Systems, Inc.

Example 1

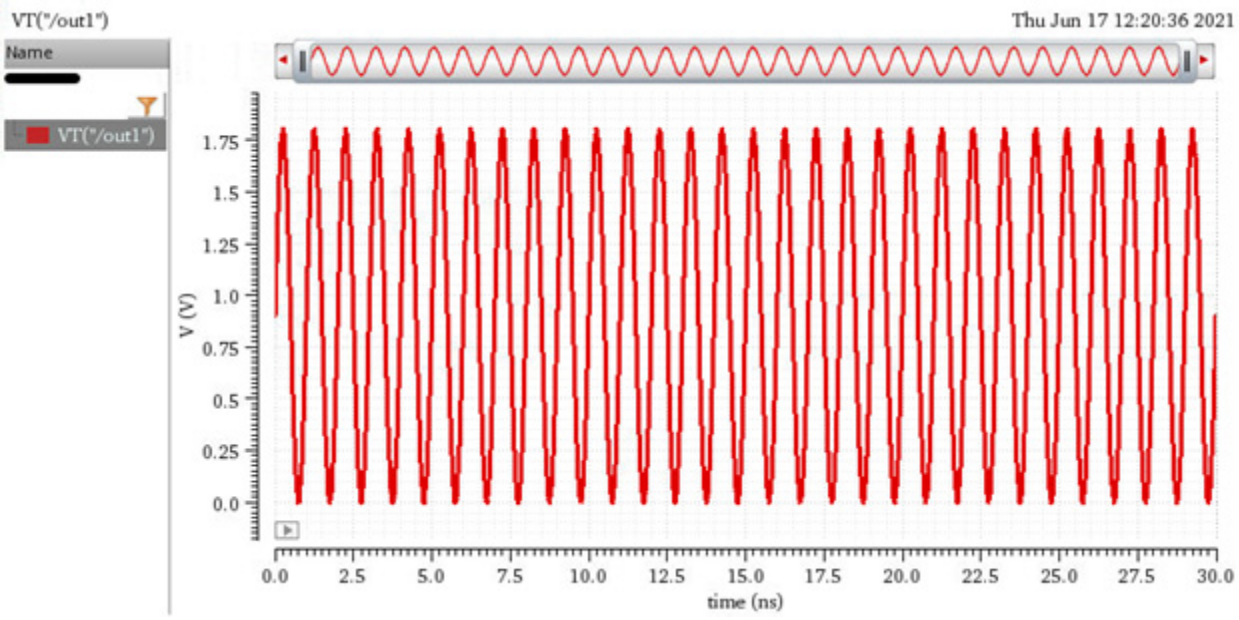
The following diagram illustrates how the result with the values clip `(sig=V(sinewave), from=10n, to=50n )` is determined.



© 2023 Cadence Design Systems, Inc.

Example 2 (interpolate = no)

This example shows the clipped waveform generated when you apply the `clip` function on the following input signal:



The following arguments are specified in this example:

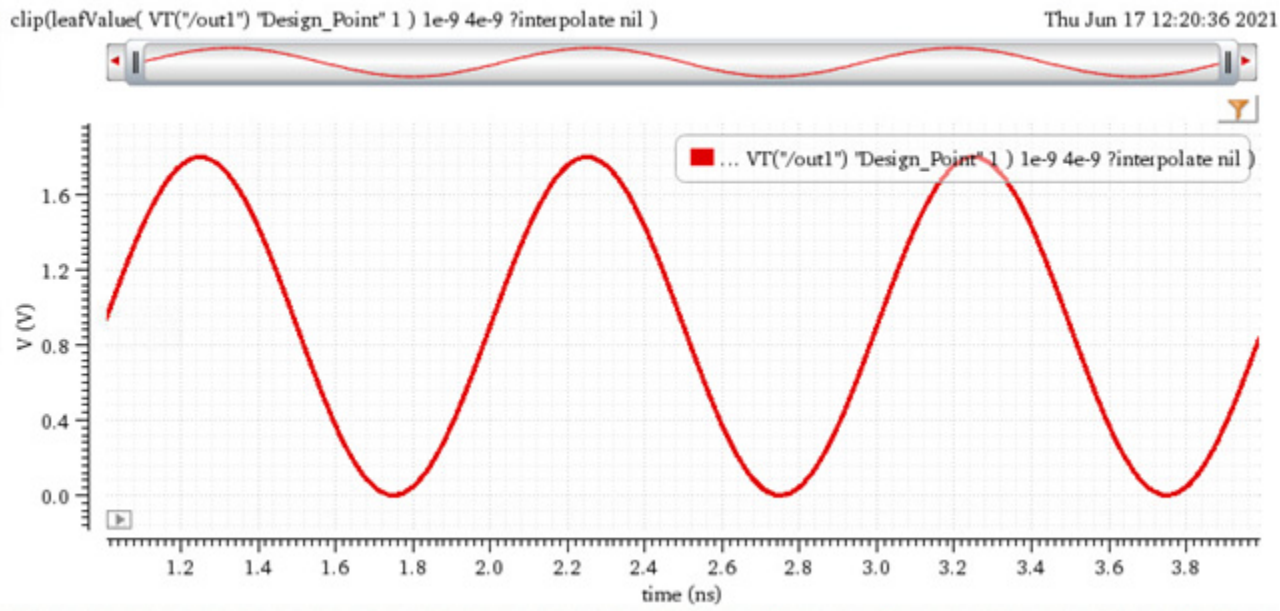
- *Signal*—`leafValue( VT("/out1") "Design_Point" 1 )`
- *From*—`1e-9`
- *To*—`4e-9`
- *interpolate*—`no`

The following expression is created in the Buffer:

```
clip(leafValue( VT("/out1") "Design_Point" 1 ) 1e-9 4e-9 ?interpolate
nil )
```

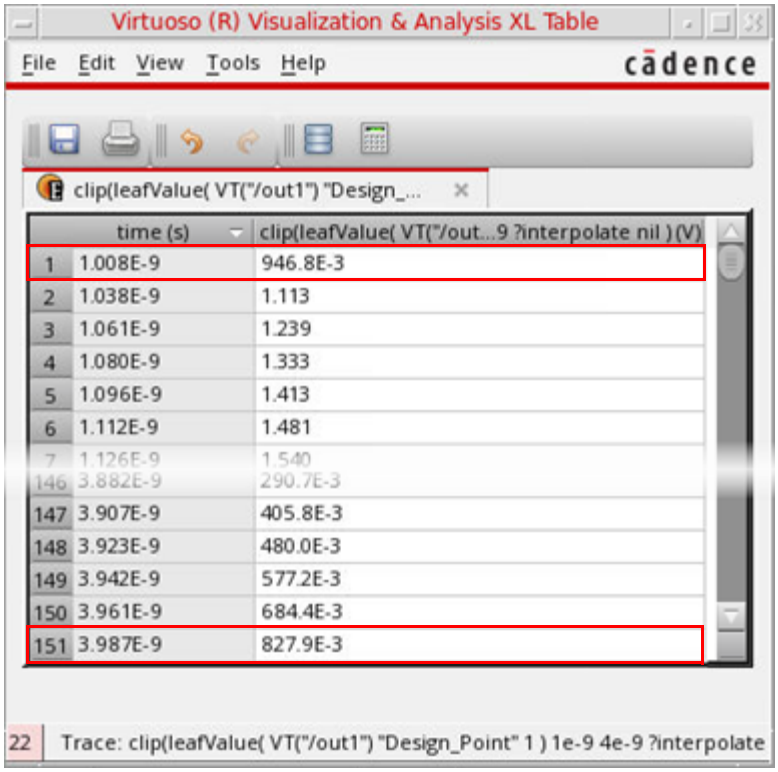

© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output is displayed in the graph window.



© 2023 Cadence Design Systems, Inc.

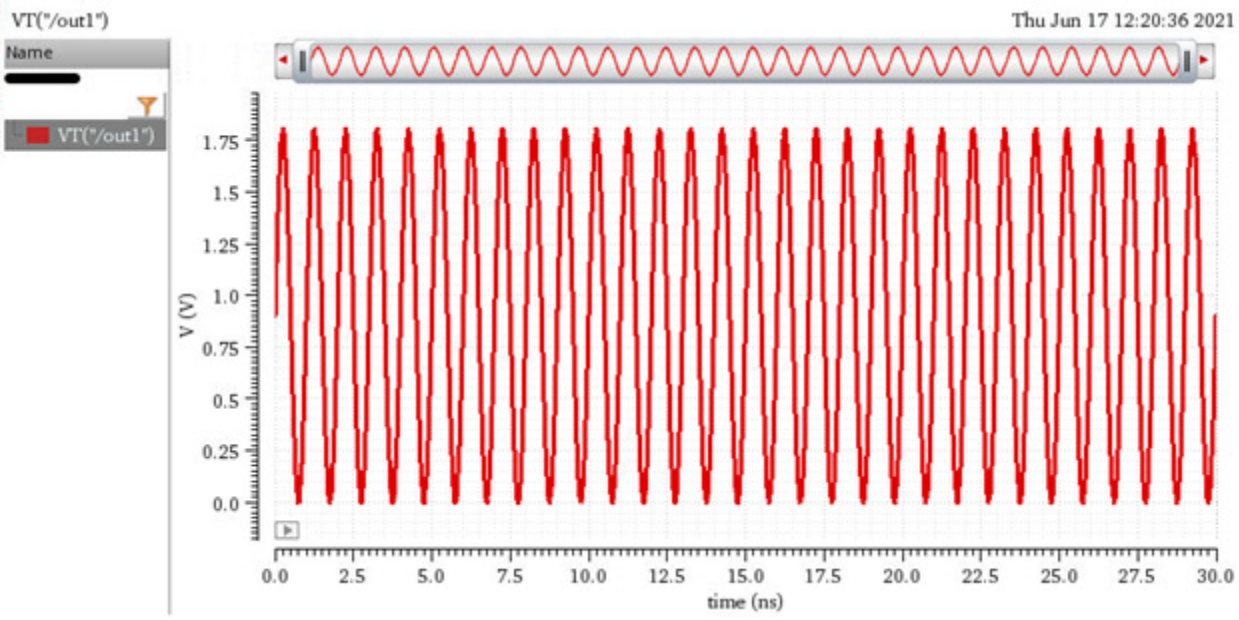
If you send this output waveform to the Virtuoso Visualization and Analysis XL Table, you will observe that the result includes only actual data points that exist between the specified *From* and *To* values because you set *interpolate* to *no*.



© 2023 Cadence Design Systems, Inc.

Example 3 (interpolate = yes)

This example shows the clipped waveform generated when you apply `clip` function on the following input signal:



The following arguments are specified in this example:

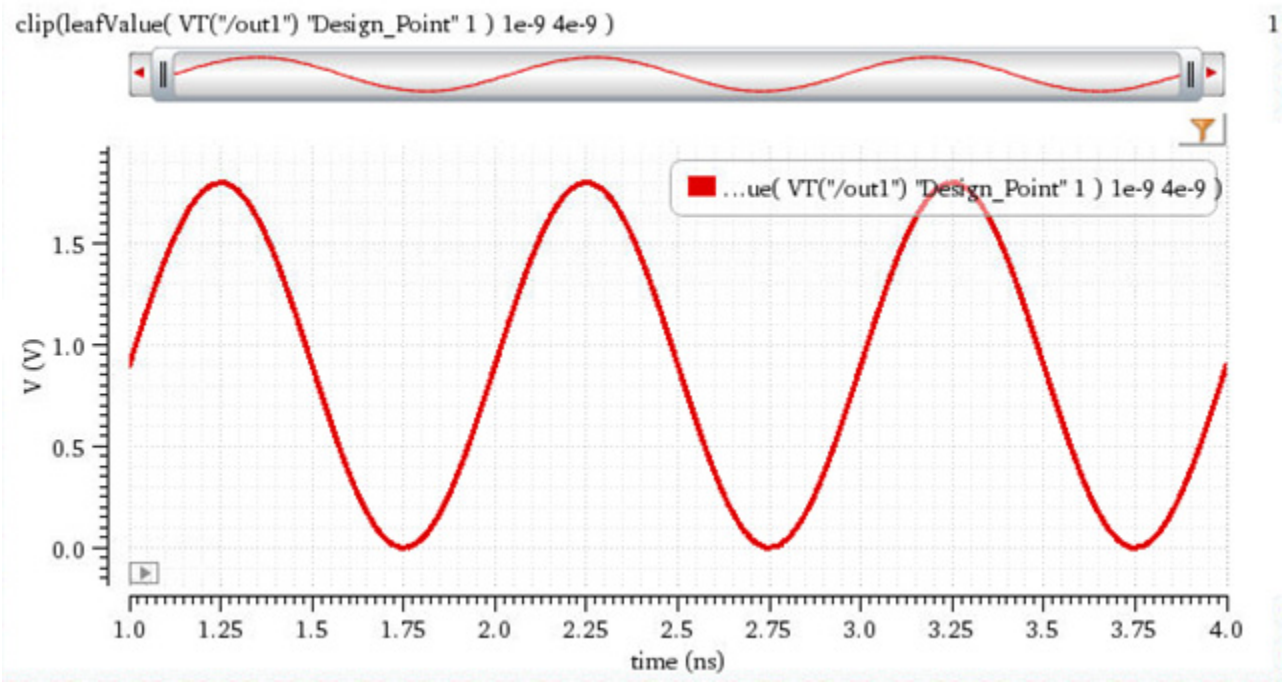
- *Signal*—`leafValue( VT("/out1") "Design_Point" 1 )`
- *From*—`1e-9`
- *To*—`4e-9`
- *interpolate*—`yes`

The following expression is created in the Buffer:

```
clip(leafValue( VT("/out1") "Design_Point" 1 ) 1e-9 4e-9 )
```

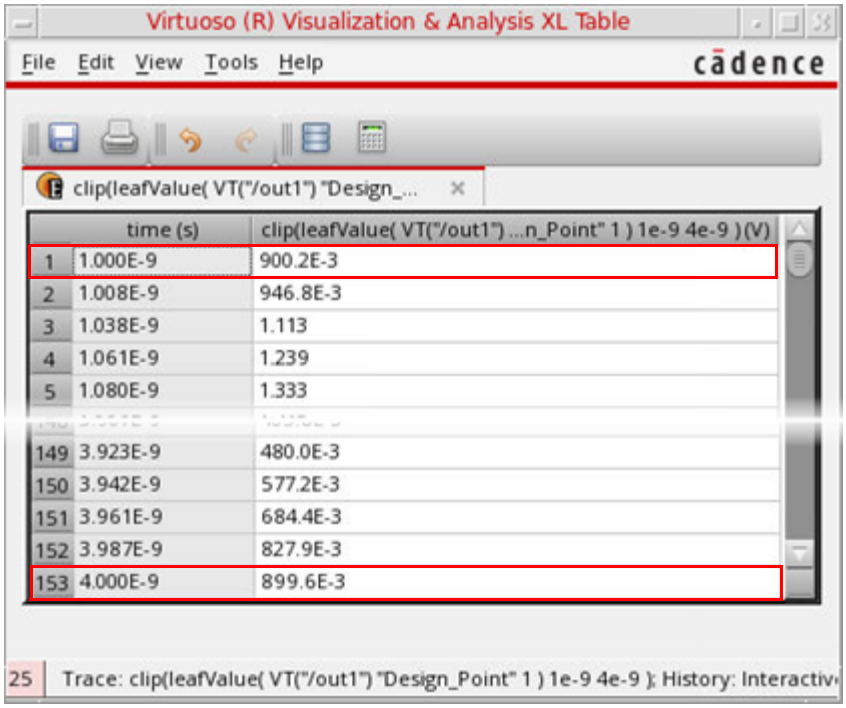

© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output is displayed in the graph window.



© 2023 Cadence Design Systems, Inc.

If you send this output waveform to the Virtuoso Visualization and Analysis XL Table, you will observe that the result includes interpolated values for the *From* and *To* fields because you set *interpolate* to *yes*.



Related OCEAN Function

The equivalent OCEAN function for `clip` is:

```
clip( o_waveform n_from n_to )
     => o_waveform/nil
```

For more information about this OCEAN function, see clip in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

compare

Compares the two given waveforms based on the specified values for absolute and relative tolerances. This function compares only the sections of the two waveforms where the X or independent axes overlap.

- *Signal1*—Name of the first signal.
- *Signal2*—Name of the second signal.
- *Absolute Tolerance*—Specifies the absolute tolerance.
- *Relative Tolerance*—Specifies the relative tolerance.

  The following situations are valid:

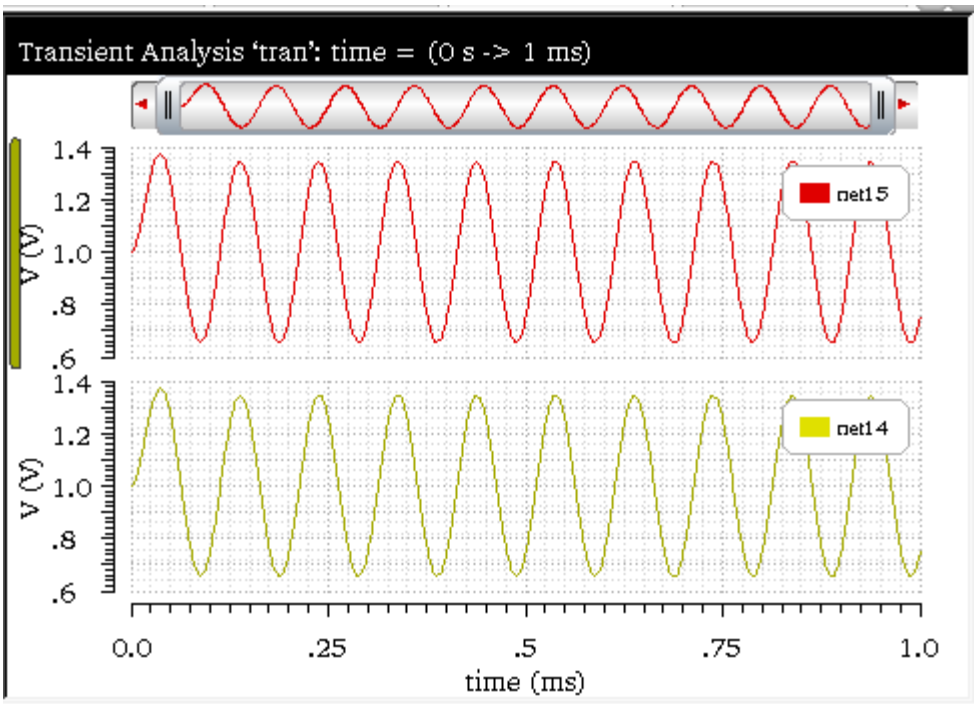
  - If neither relative nor absolute tolerance is specified, the function returns the difference of the two waveforms (Signal1 - Signal2).
  - If only the absolute tolerance is specified, the function returns the difference of the two waveforms only when the absolute value of the difference is greater than the absolute tolerance (|Signal1 - Signal2| > f_abstol); otherwise it returns a zero waveform.
  - If only the relative tolerance is specified, the function returns the difference of the two waveforms only when the absolute value of the difference is greater than the product of the relative tolerance and the larger of the absolute values of the two waveforms (|Signal1 - Signal2| > f_reltol * max(|Signal1|, |Signal2|)); otherwise it returns a zero waveform.
  - If both relative and absolute tolerances are specified, the function returns the difference of the two waveforms only when the absolute value of the difference is greater than the sum of the separately calculated tolerance components (|Signal1 - Signal2| > f_abstol + f_reltol * max(|Signal1|, |Signal2|)); otherwise it returns a zero waveform.

© 2023 Cadence Design Systems, Inc.

Note: The function also compares parametric waveforms. However, for a successful comparison of parametric waveforms, the family tree structures of the two input waveforms should be the same. For both the input waveforms, the number of child waveforms at each level should also be the same, except at the leaf level where the elements are simple scalars.

Example

This example shows the output waveform generated when you apply the `compare` function on the following input signals:



The following arguments are specified in this example:

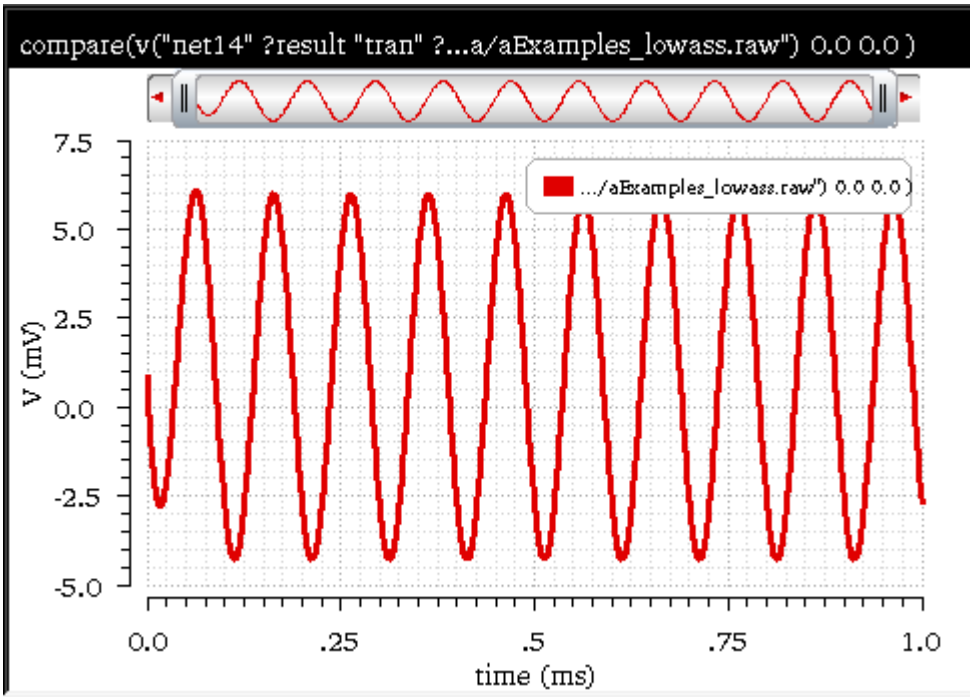
- *Signal1*—`v("net14" ?result "tran")`
- *Signal2*—`v("net15" ?result "tran")`
- *Absolute Tolerance*—`0.0`
- *Relative Tolerance*—`0.0`

The following expression is created in the Buffer:

```
compare(v("net14" ?result "tran" ?resultsDir "./aExamples.raw")
v("net15" ?result "tran" ?resultsDir "./aExamples.raw") 0.0 0.0 )
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN function for `compare` is:

```
compare( o_waveform1 o_waveform1 [f_abstol [f_reltol]] )
     => o_comparisonWaveform/nil
```

For more information about this OCEAN function, see compare in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

compression

Returns the *N*th compression point value of a waveform at the specified extrapolation point. This function is available only in the SKILL mode.

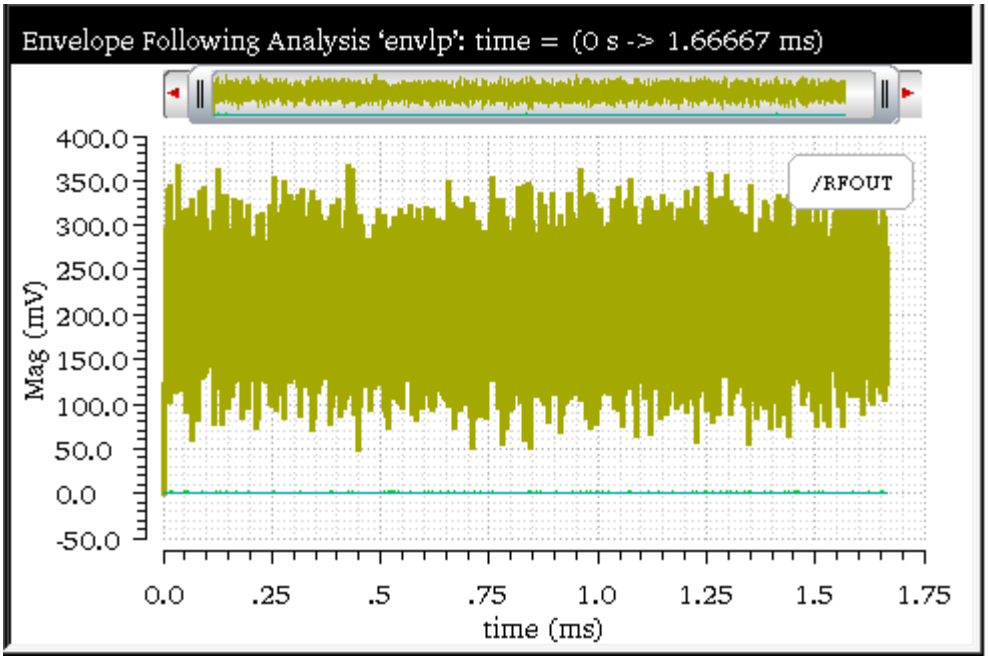
The compression function uses the power waveform to extrapolate a line of constant slope (dB/dB) according to a specified input or output power level. This line represents constant small-signal power gain (ideal gain). The function finds the point where the power waveform drops n dB from the constant slope line and returns either the X coordinate (input referred) value or the Y coordinate (output referred) value.

- *Signal*—Name of the signal.
- *Harm Num*—Harmonic number.
- *Ext. Point (X)*—Extrapolation point of the waveform. The extrapolation point is the X-axis value.
- *Compression dB*—Compression coefficient (N).

© 2023 Cadence Design Systems, Inc.

Example

This example shows the compression value generated when you apply the `compression` function on the following input waveform:



The following arguments are specified in this example:

- *Signal*—`v("/RFout")`
- *Harm Num.*—`1`
- *Ext. Point (X)*— `-30`
- *Compression dB*—`1`

The following expression is created in the Buffer:

```
compression(dB20(harmonic(mag(v("/RFOUT" ?result "envlp_fd"
?resultsDir "./viva/rfworkshop/simulation/EF_example_envlp/spectre/
schematic/psf"")), 1)), ?x -30, ?compress 1)
```

Now, when you evaluate this expression, the following output value is displayed in the Buffer:

```
21.96E-9
```


© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN function for `compression` is:

```
compression( o_waveform [ ?x f_x ] [ ?y f_y ]
     [ ?compression f_compression ] [ ?io s_measure ] )
     => f_compPoint/nil
```

For more information about this OCEAN function, see compression in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

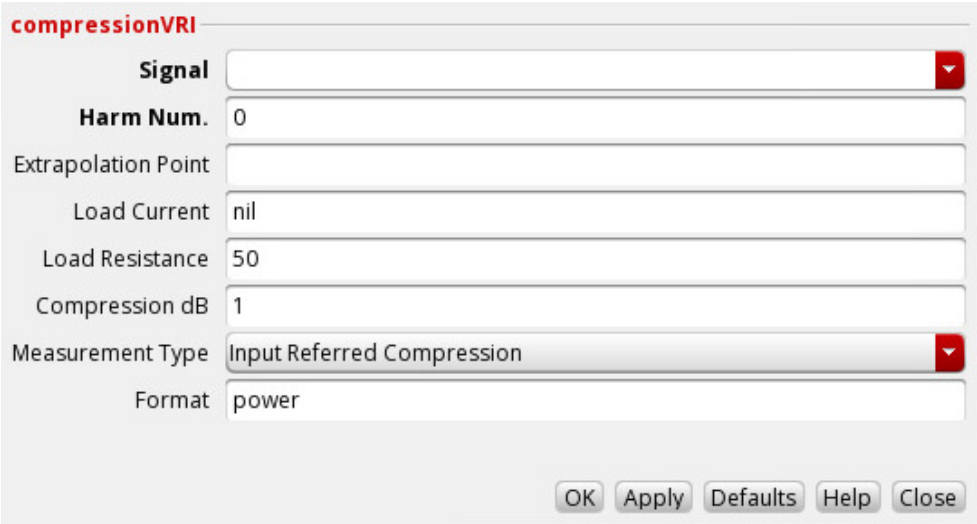
compressionVRI

Performs an `Nth` compression point measurement on a power waveform. This function is available only in the SKILL mode.

You can use this function to simplify the declaration of a compression measurement.

This function extracts the specified harmonic from the input waveform(s), and uses dBm(spectralPower((i or v/r),v)) to calculate a power waveform. The function passes this power curve and the remaining arguments to the compression function to complete the measurement.

The compression function uses the power waveform to extrapolate a line of constant slope (dB/dB) according to a specified input or output power level. This line represents constant small-signal power gain (ideal gain). The function finds the point where the power waveform drops n dB from the constant slope line and returns either the X coordinate (input referred) value or the Y coordinate (output referred) value.



- *Signal*—Name of the signal.
- *Harm Num*—Harmonic index of the waveform.
- *Extrapolation Point*—Extrapolation point for the waveform. The default value is the minimum x value of the input voltage waveform.
  The extrapolation point is the coordinate value in dBm that indicates the point on the output power waveform where the constant-slope power line begins. This point should be in the linear region of operation.
- *Load Current*—Current into the output port.

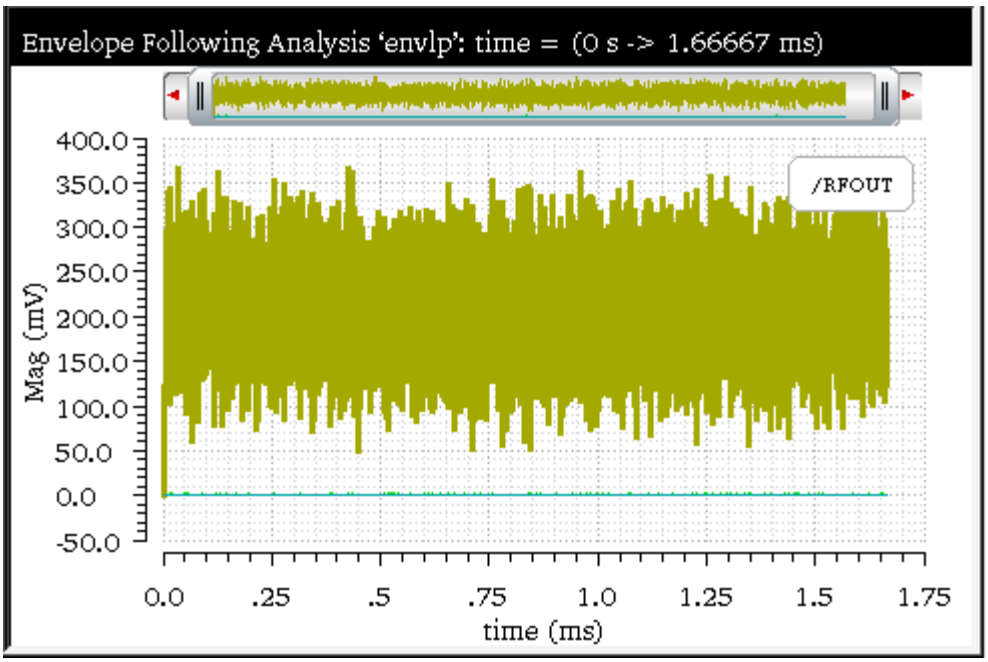
© 2023 Cadence Design Systems, Inc.

Note: This argument must be a family of spectrum waveforms created by parametrically sweeping the input power (in dBm) of the circuit. When specified, the output power is calculated using voltage and current.

- *Load Resistance*—Resistance value. The default value is 50.
- *Compression dB*—Specifies the delta (in dB) between the power waveform and the ideal gain line that marks the compression point. The default value is 1.
- *Measurement Type*—Select the *Input Referred Compression* option to get the X-coordinate of the intercept. Select *Output Referred Compression* to get the Y-coordinate of the intercept.
- *Format*—Default value: `power`.

Example

This example shows the compression value generated when you apply the `compressionVRI` function on the following input waveform:



The following arguments are specified in this example:

- *Signal*—`v("/RFout")`
- *Harm Num.*—`1`
- *Extrapolation Point*— `-30`

© 2023 Cadence Design Systems, Inc.

- Load Resistance—`50`
- *Compression dB*—`1`
- *Measurement Type*—`Input Referred Comparison`

The following expression is created in the Buffer:

```
compressionVRI(mag(v("/RFOUT" ?result "envlp_fd" ?resultsDir "./
viva/rfworkshop/simulation/EF_example_envlp/spectre/schematic/
psf")) 1 ?gcomp 1 ?epoint -30 ?rport 50 )
```

Now, when you evaluate this expression, the following output value is displayed in the Buffer:

Related OCEAN Function

The equivalent OCEAN function for `compressionVRI` is:

```
compressionVRI( o_vport x_harm [?iport o_iport] [?rport f_rport]
     [?epoint f_epoint] [?gcomp f_gcomp] [?measure s_measure] )
     => o_waveform/n_number/nil
```

For more information about this OCEAN function, see compressionVRI in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

convolve

Returns a waveform consisting of the time domain convolution of two signals.

This function includes the following arguments in the SKILL mode:

- *Signal1*—Name of the first signal.
- *Signal2*—Name of the second signal.
- *From*—*S*tarting point (X-axis value) of the integration range.
- *To*—End point (X-axis value) of the integration range.
- *type*—*S*pecifies whether is the interpolation is linear or log.
- *By*—Specifies the step size.

Additional Information

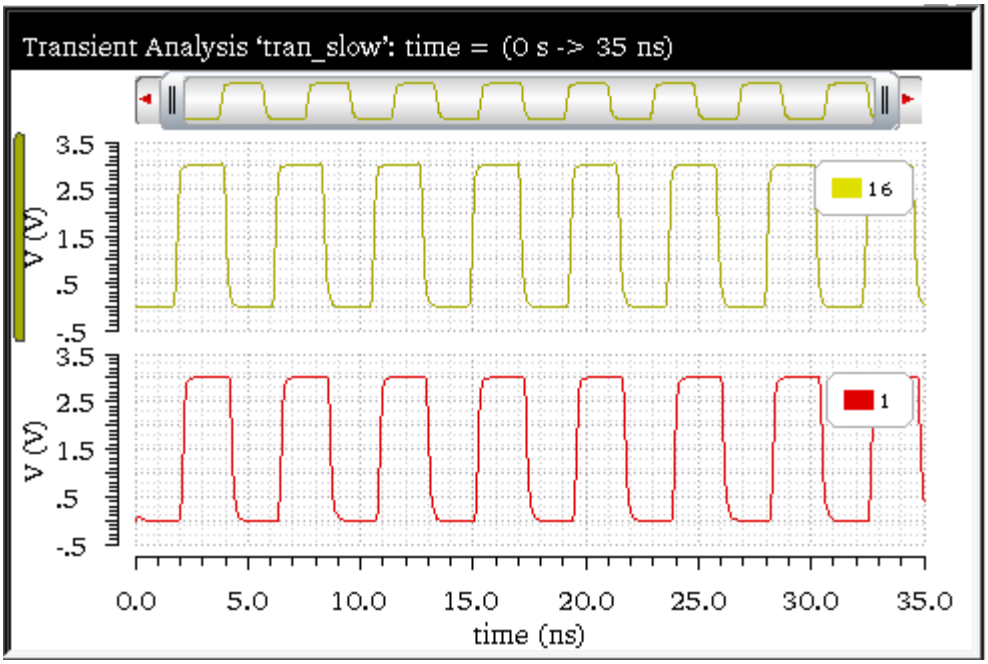
Covolution is defined by the following equation:

$$\int_{from}^{to} f1(s) f2(t-s)\,ds$$

© 2023 Cadence Design Systems, Inc.

Example

This example calculates the time domain convolution of the following two transient voltages, `v("/1")` and `v("/16")` and displays the output waveform generated:



The following arguments are specified in this example:

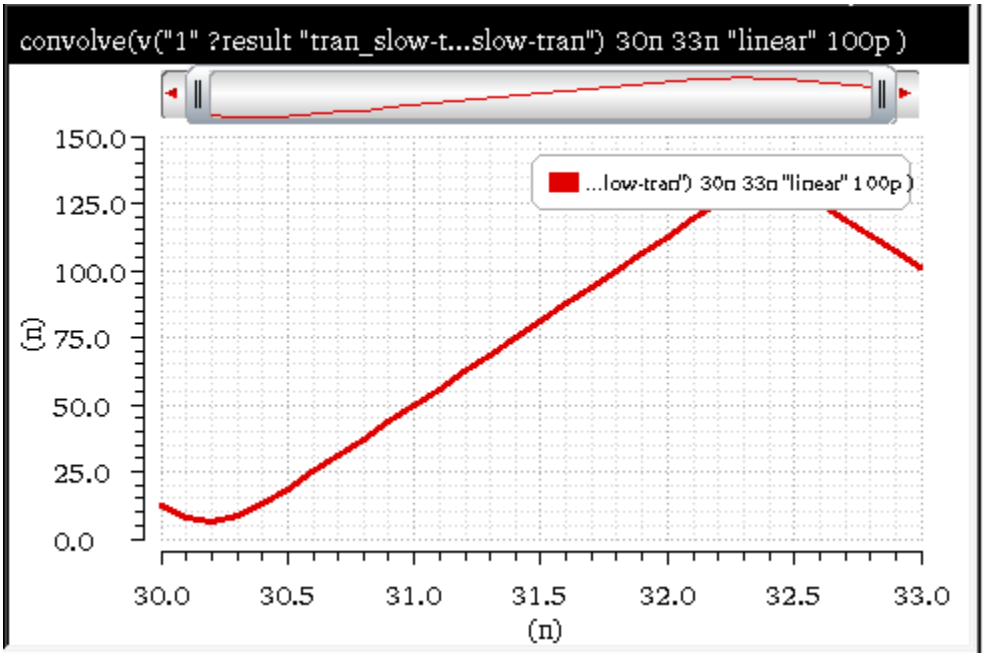
- *Signal1*—`v("1" ?result "tran")`
- *Signal2*—`v("16" ?result "tran")`
- *From*—`30n`
- *To*— `33n`
- *Type*—`linear`
- *By*—`100p`

The following expression is created in the Buffer:

```
convolve(v("1" ?result "tran") v("16" ?result "tran") 30n 33n
"linear" 100p )
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform showing the time domain convolution is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN function for `convolve` is:

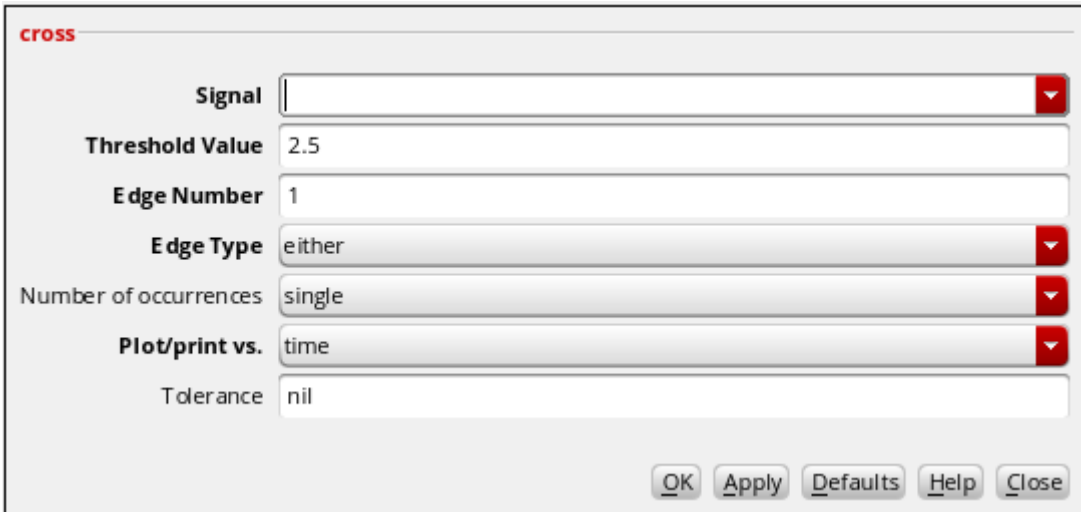
```
convolve( o_waveform1 o_waveform2 n_from n_to t_type n_by )
     => o_waveform/n_number/nil
```

For more information about this OCEAN function, see convolve in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

cross

Returns the X value where a signal crosses the threshold Y value.



This function includes the following arguments:

- *Signal*—Name of the signal.
- *Threshold Value*—Threshold to be crossed.
- *Edge Number*—Occurrence of the crossing. The first crossing is *Edge Number*=`1`, the second crossing is *Edge Number*=`2`, and so on. The value of *Edge Number* can be negative numbers: *Edge Number*=`-1` for the previous occurrence, *Edge Number*=`-2` for the occurrence before the previous occurrence, and so on.
- *Edge Type*—Direction of the crossing event. `rising` directs the function to look for crossings where the Y value is increasing, `falling` for crossings where the Y value is decreasing, and `either` for crossings in either direction.

  When the threshold is maximum or minimum, both the rising and falling edge types can be used to calulate cross.

  Note: When you are using a parametric sweep, ensure that the sweep values are specified in increasing order to calculate the cross for falling edge type. Otherwise, the function evaluates to nil and following warning message appears:

  ```
  WARNING: The cross function cannot be applied on this expression due to one of the following reasons:
  - Threshold value is invalid
  - Edge Type field is set to 'falling', but the X-axis values of input waveform are specified in decreasing order
  ```

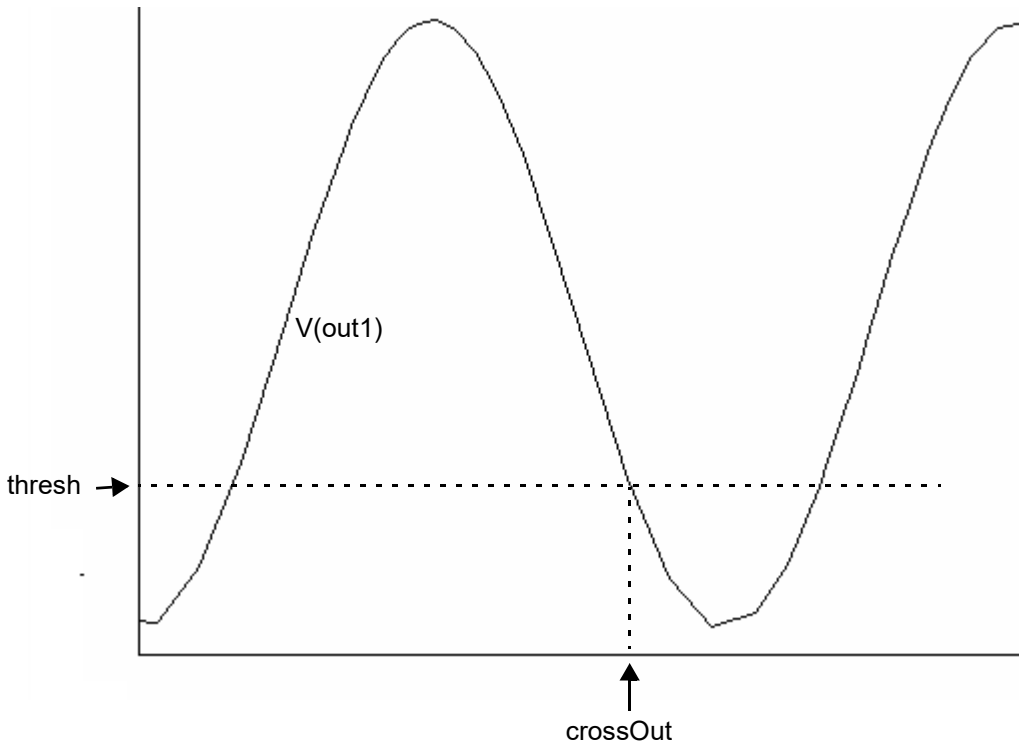

© 2023 Cadence Design Systems, Inc.

```
Specify the edge type as 'rising' or 'either', and ensure that the threshold
value is valid.
```

- *Number of occurrences*—Select `single` to calculate one point or select `multiple` to calculate until the end of X range.
- *Plot/print vs.*—Specifies whether X-axis of the output waveform is time (or another X-axis parameter for non-transient data) or cycle.
- *Tolerance*—Sets the tolerance value to be used to detect the threshold crossings. You can specify a tolerance value in cases where the noise is large. In such cases, the waveform can cross a given threshold multiple times within one transition. These double crossings are counted as two crossings and generate huge spikes in the time domain jitter waveform.

Example 1

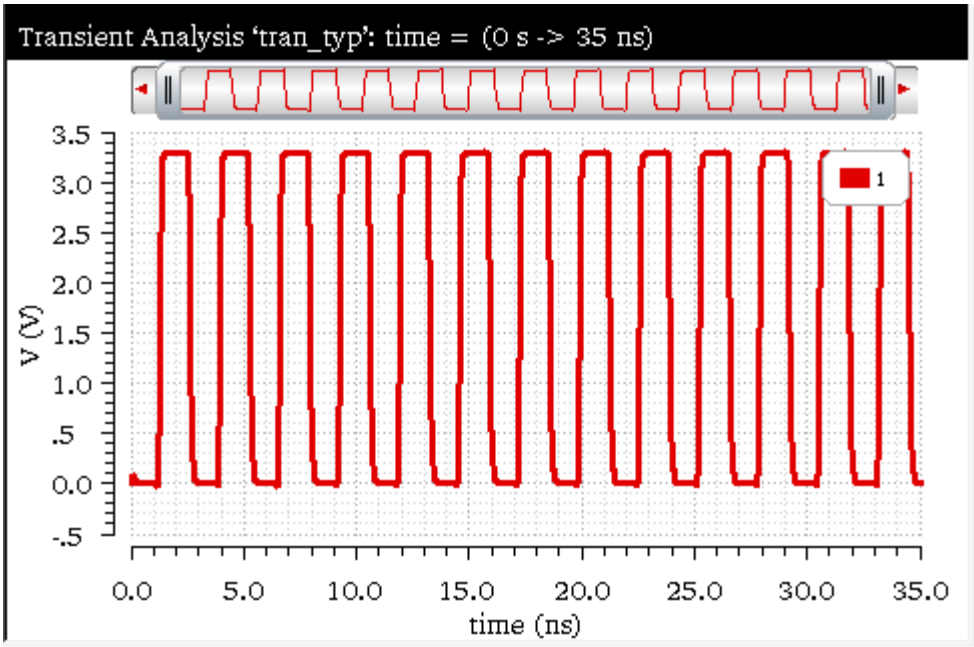
The following diagram illustrates how the result is determined for the values *signal*=`V(out1)`, *Threshold Value*=`1`, *Edge Number*=`1`, and *Edge Type*=`falling`



© 2023 Cadence Design Systems, Inc.

Example 2

This example shows the output waveform generated when you apply the `cross` function on the following input waveform:



The following arguments are specified in this example:

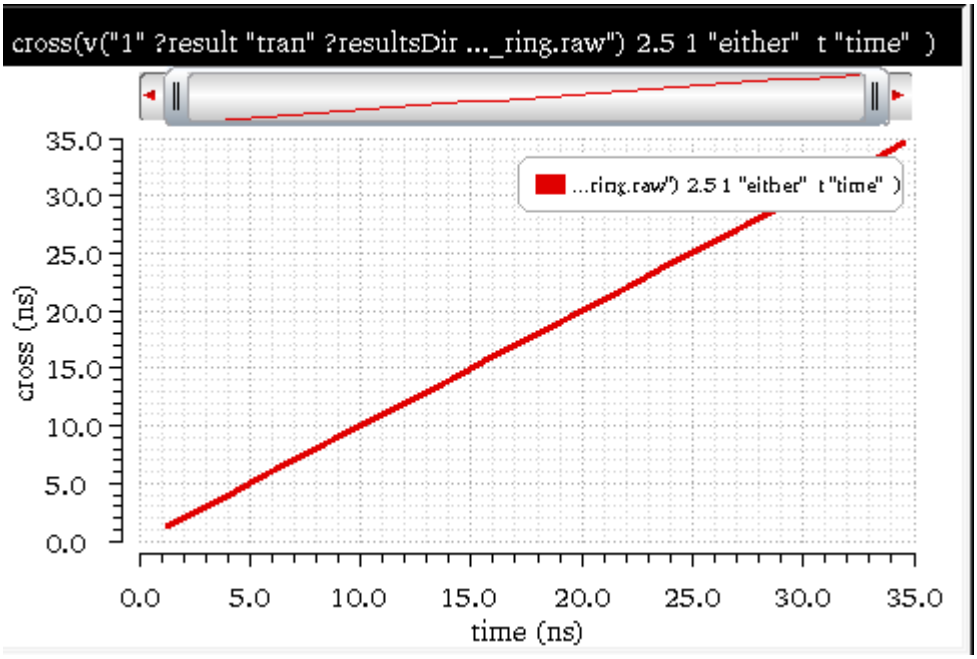
- *Signal*—`v("1" ?result "tran")`
- *Threshold*—`2.5`
- *Edge Number*—`1`
- *Edge Type*—`either`
- *No of occurrences*—`multiple`
- *Plot/print vs.*—`time`
- *Tolerance*—`nil`

The following expression is created in the Buffer:

```
cross(v("1" ?result "tran") 2.5 1 "either"  t "time" )
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `cross` is:

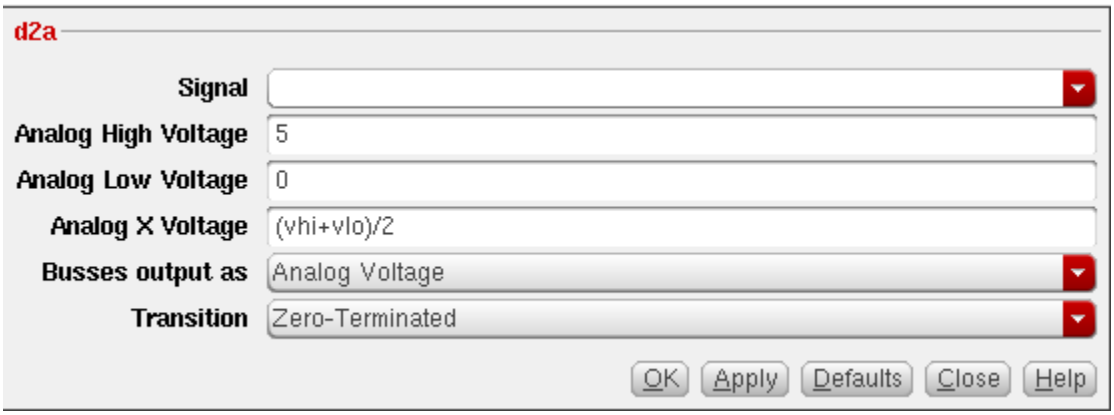
```
cross( o_waveform n_crossVal x_n s_crossType [g_multiple [s_Xname]] )
     => o_waveform/g_value/nil
```

For more information about the OCEAN function, see cross in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

d2a

Returns the analog output from a given digital waveform.



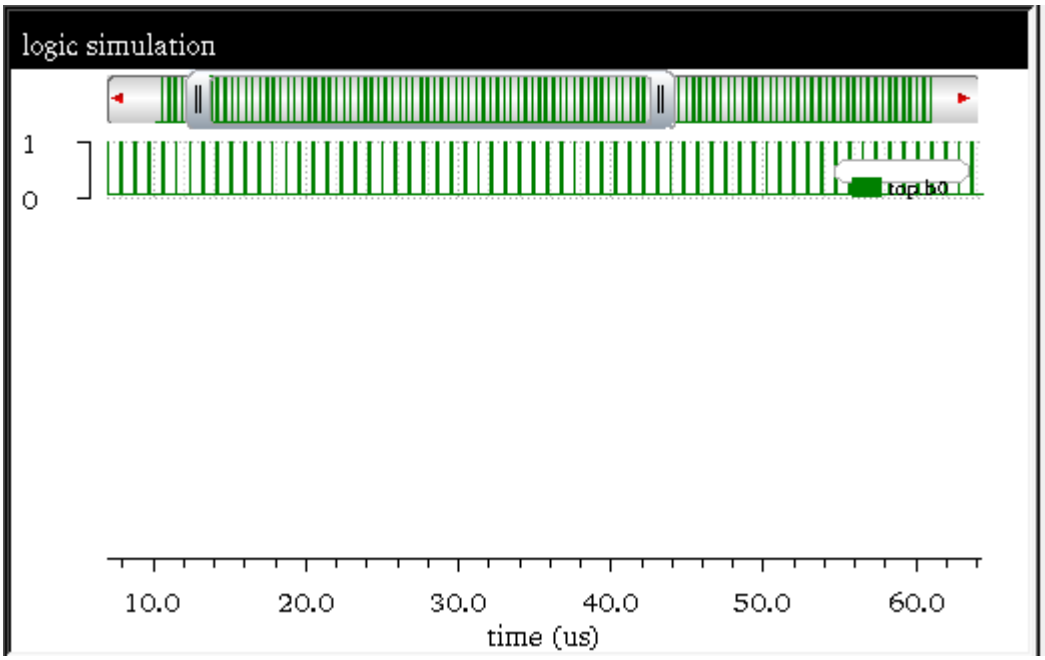
This function includes the following arguments:

- *Signal*—Name of the waveform, expression, or family of waveforms. The expression displayed in the Buffer can be added to this field by selecting *buffer*.
- *Analog High Voltage*—Highest voltage value when the digital signal is in the high-state (1)
- *Analog Low Voltage*—Lowest analog value when the digital signal is in the low-state (0)
- *Analog X Voltage*—Specify the corresponding analog value to which you want to convert the state X of the digital waveform. This value can be a number or a string expression of `vhi` and `vlo`, such as `(vhi+vlo)/2`.
- *Busses output as*—Set this field to `Analog Voltage` or `Analog bits` if the given digital waveform is a bus.
- *Transition*—Set the transition to `Piecewise Linear` or `Zero-Terminated` if the given digital waveform is a bus.

© 2023 Cadence Design Systems, Inc.

Example

This example shows the conversion of digital waveform (logic signal) to an analog waveform. The input signal is as shown in the figure below:



The following arguments are specified in this example:

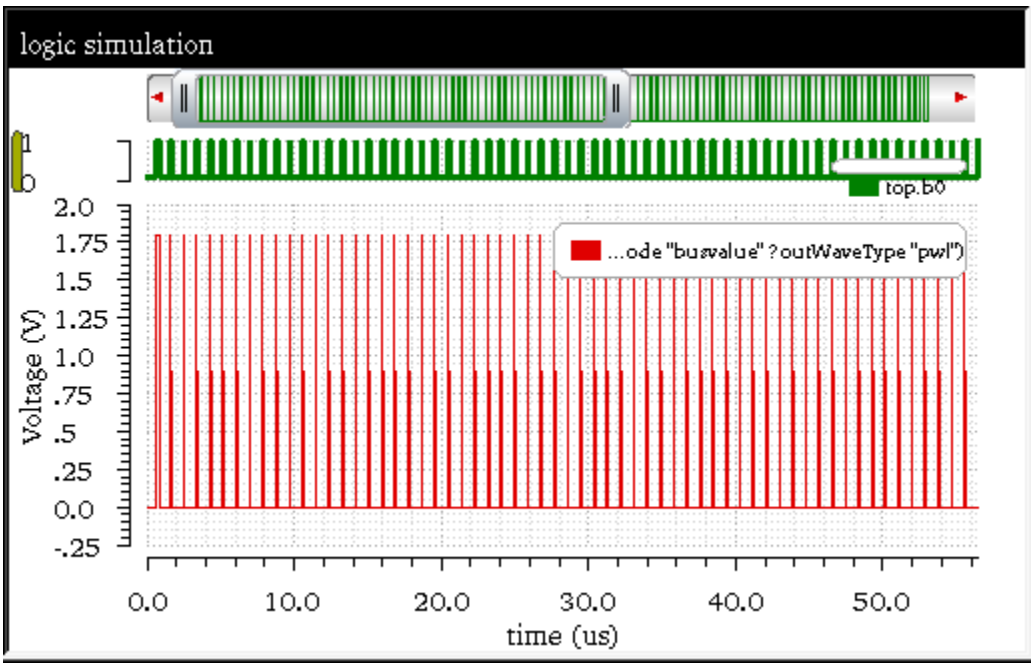
- *Signal*—`getData(top.b0" ?result "waves.shm")`
- *Analog High Voltage*—`1.8`
- *Analog Low Voltage*—`0`
- *Analog X Voltage*—`(vhi+vlo)/2`
- *Busses output as*—`Analog Voltage`
- *Transition*—`Piecewise-Linear`

The following expression is created in the Buffer:

```
awvDigital2Analog(getData("top.b0" ?result "waves.shm") 1.8 0
"(vhi+vlo)/2" ?mode "busvalue" ?outWaveType "pwl")
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform, which is an analog signal, is displayed in the graph window.



Related SKILL Function

The equivalent SKILL command for `d2a` is:

```
awvDigital2Analog( o_waveform n_vhi n_vlo s_VX
     @key s_mode s_outWaveType s_vprevSTART )
     => o_waveform | nil
```

For more information about this SKILL function, see awvDigital2Analog in *Virtuoso Visualization and Analysis XL SKILL Reference*.

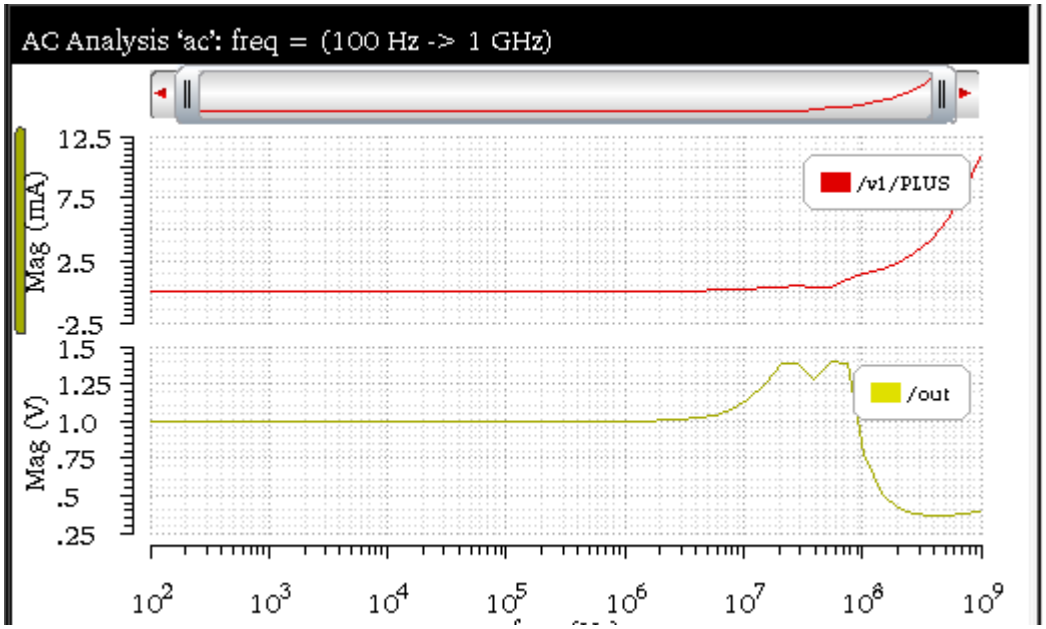
© 2023 Cadence Design Systems, Inc.

dBm

Returns 10 times the log10 of the specified waveform object plus 30. This function converts a signal, in watts, to dbm, where `dbm=10*log(x)+30`.

Example

This example shows the output waveform generated when you apply the `dBm` function on the following input waveforms:



The above figure shows the current waveform, `/v1/PLUS` and voltage waveform, `/out`. If you want to calculate the `dBm` power of these signals, firstly you will have to multiply the terminal voltage and current signals.

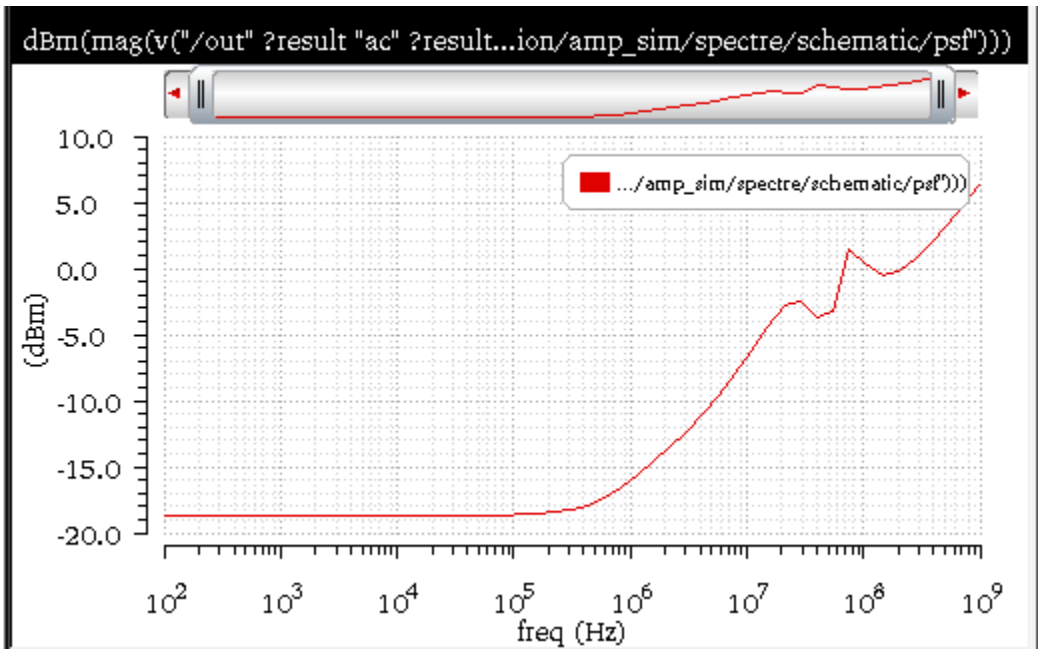
```
mag(v("/out" ?result "ac"))*mag(i("/v1/PLUS" ?result "ac"
?resultsDir ./viva/ADE_ViVA/ADEviva/simulation/amp_sim/spectre/
schematic/psf))
```

When you apply the `dBm` function, the following expression is created in the Buffer:

```
dBm(mag(v("/out" ?result "ac"))*mag(i("/v1/PLUS" ?result "ac"
?resultsDir ./viva/ADE_ViVA/ADEviva/simulation/amp_sim/spectre/
schematic/psf)))
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN function for `dBm` is:

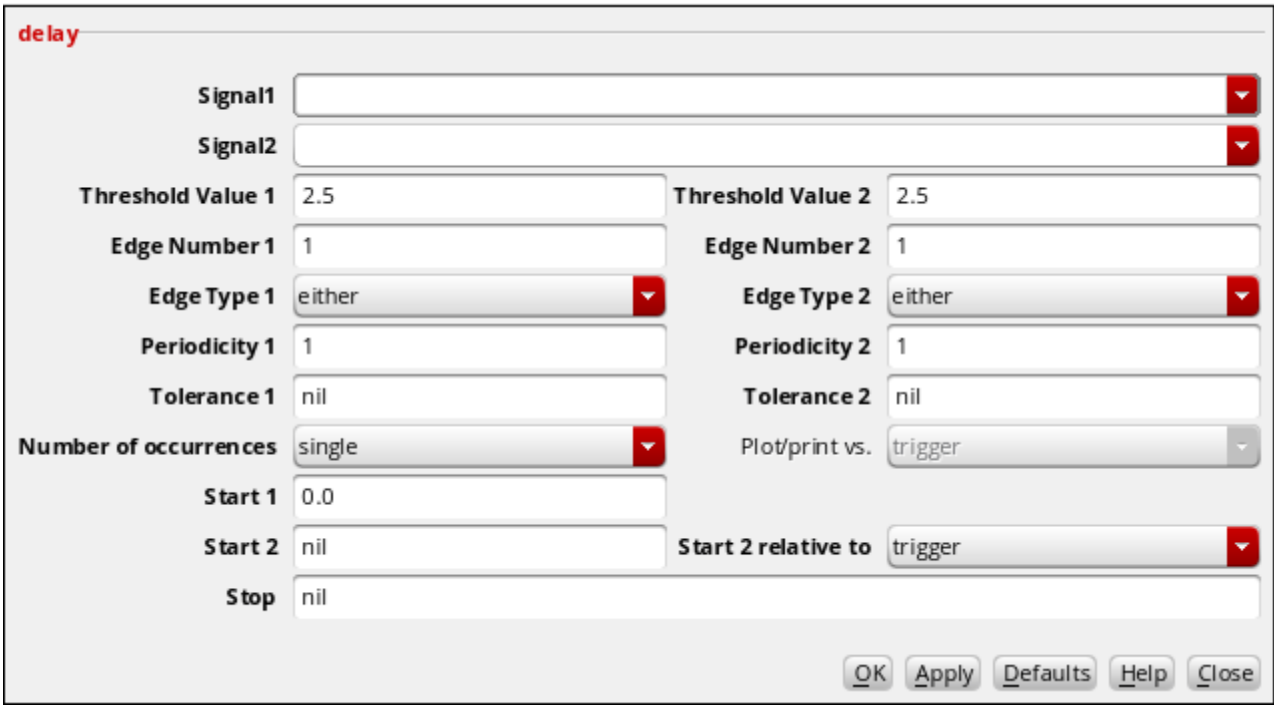
```
dbm( {o_waveform | n_number} )
     => o_waveform/n_number/nil
```

For more information about the OCEAN function, see dBm in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

delay

Computes the delay between two points or multiple sets of points in a waveform using the cross function. This function is available only in the SKILL mode.



This function includes the following arguments:

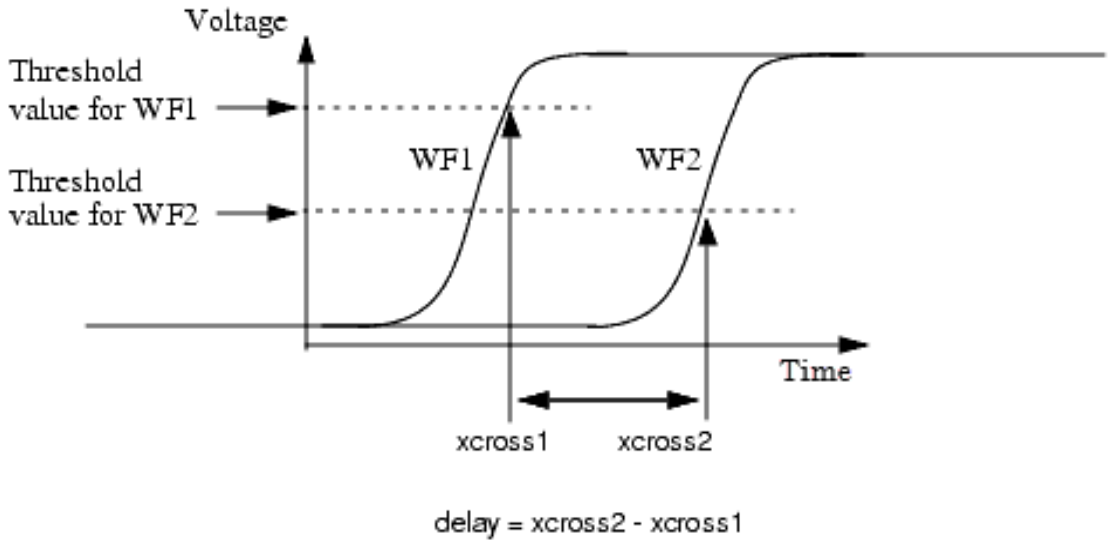
- *Signal1*—Name of the first signal.
- *Signal2*—Name of the second signal.
- *Threshold Value 1*—First threshold to be crossed.
- *Edge Number 1*—Number that specifies which crossing is to be the trigger event. For example, if *Edge Number1=2*, the trigger event is the second edge of the first waveform with the specified type that crosses *Threshold Value 1*.
- *Edge Type 1*—Direction of the first crossing event. `rising` directs the function to look for crossings where the Y value is increasing, `falling` for crossings where the Y value is decreasing, and `either` for crossings in either direction.
- *Periodicity 1*—Periodic interval for the first waveform. See Example 2 for details.
- *Tolerance1*—Tolerance value to be added to the threshold value for the first signal so that it is counted as a crossing point. The value must be specified as an absolute value.

© 2023 Cadence Design Systems, Inc.

- *Threshold Value 2*—Second threshold to be crossed.
- *Edge Number* 2—Number that specifies which crossing is to be the trigger event. For example, if *Edge Number2=2*, the trigger event is the second edge of the second waveform with the specified type that crosses *Threshold Value 2*.
- *Edge Type 2*—Direction of the second crossing event. `rising` directs the function to look for crossings where the Y value is increasing, `falling` for crossings where the Y value is decreasing, and `either` for crossings in either direction.
- *Periodicity 2*—Periodic interval for the second waveform. See Example 2 for details.
- *Tolerance2*—Tolerance value to be added to the threshold value for the second signal so that it is counted as a crossing point. The delay function calculates the delay between crossing points.
- *Number of occurrences*—Specifies whether you want to retrieve only one occurrence of a delay event for the given waveform (*single*), or all occurrences of overshoot for the given waveform which you can later plot or print (*multiple*).
  `Single`–If Start2 is specified, find the cross point of Signal1 at edge number1. Use this as the trigger point for target Signal2 and ignore Signal2 before Signal1 trigger event to find its cross point. Calculate the delay between the two cross points.
  If Start2 is not specified, find the cross point at edge number1 of Signal1 and cross point of edge number2 of Signal2 with target at time t=0 and calculate the delay between two cross points.
  `Multiple`–Find all the cross points of Signal1. If Start2 is specified, find cross points of target(Signal2) staring at trigger event (first cross point of Signal1) and ignore Signal2 before Signal1 trigger event. If Start2 is not specified, find cross points of target(Start2) at time at t=0. Calculate the delay between each of the cross points falling at interval of period1 and period2 of Signal1 and Signal2 respectively.
- *Plot/print vs.*—Specifies whether you want to retrieve delay data against *trigger* time, *target* time (or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform.
  The value in this field is ignored when you specify *Number of Occurences* as *single*.
- *Start 1*—Time that specifies when the delay measurement is to be started.
- *Start 2*—Time to start observing the target event.
- *Start 2 relative to*—Specifies whether the Start 2 is relative to trigger or time.

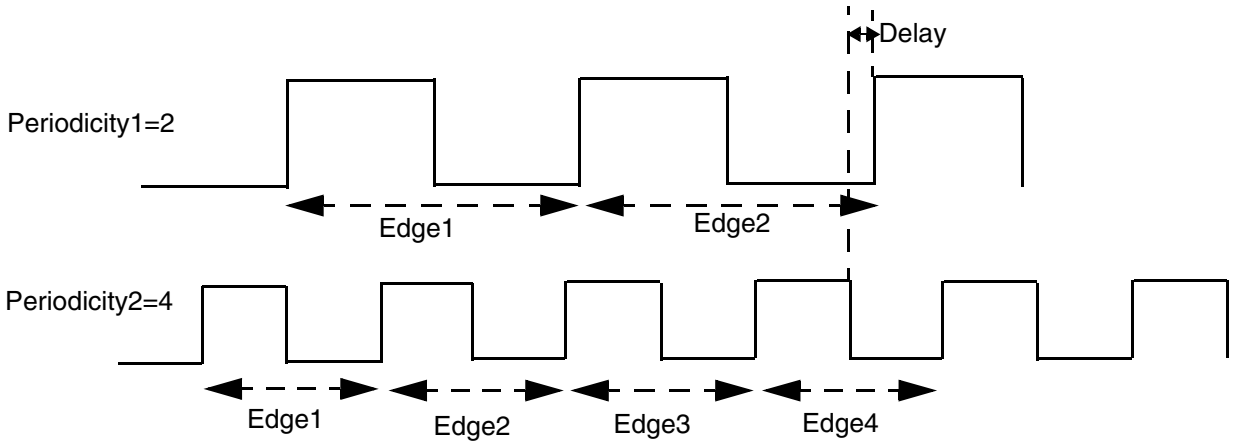
© 2023 Cadence Design Systems, Inc.

- *Stop*—Time to stop observing the target event.



Example 1

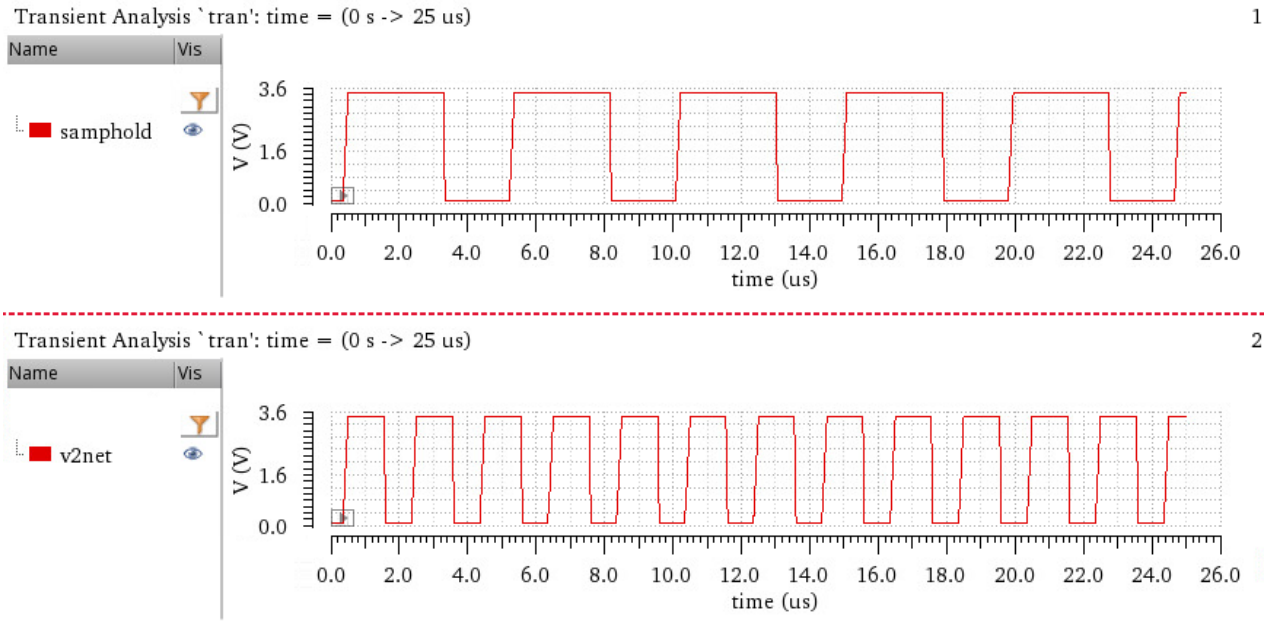
The delay algorithm for multiple occurrences returns the difference between the X points for the specified edges on the respective waveforms (with the specified periodicity). For example: if the edge number specified for both the waveforms is 1 and periodicity is 2, then it will compute the difference between the 1,3,5,7 ... edges for both the waveforms.



© 2023 Cadence Design Systems, Inc.

Example 2

Consider the following example, where `samphold` is the first signal and `v2net` is the second signal.



To calculate the delay between the crossing points of the signals, `samphold` and `v2net`, specify the following arguments:

- *Signal1*—`v("samphold" ?result "tran")`
- *Signal2*—`v("v2net" ?result "tran")`
- *Threshold Value 1*—`2.5`
- Threshold Value 2—`2.5`
- *Edge Number 1*—`1`
- *Edge Number 2*—`1`
- *Edge Type 1*—`rising`
- *Edge Type 2*—`rising`
- *Periodicity* 1—`6e-6`
- *Periodicity 2*—`1`
- *Tolerance1*—`0.2e-6`
- *Tolerance2*—`0.2e-6`

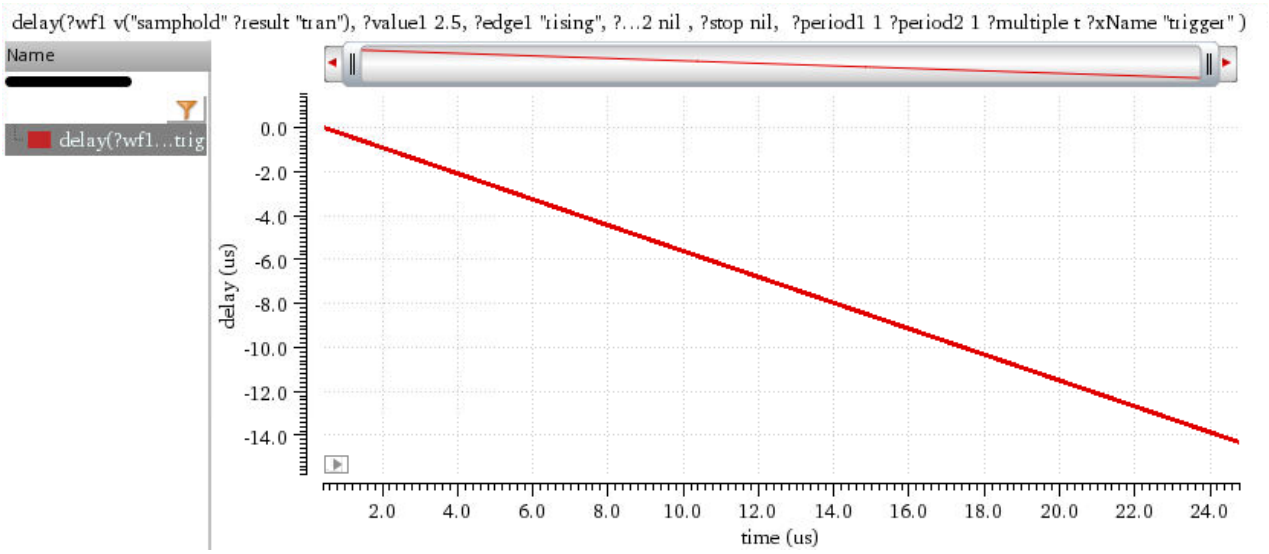
© 2023 Cadence Design Systems, Inc.

- *Number of occurrences*—`multiple`
- *Plot/print vs.*—`trigger`
- *Start 1*—`0.0`
- *Start 2*—`nil`
- *Start 2 relative to*—`trigger`
- *Stop*—`nil`

The following expression is created in the Buffer:

```
delay(?wf1 v("samphold" ?result "tran"), ?value1 2.5, ?edge1
"rising", ?nth1 1, ?td1 0.0, ?tol1 0.2e-6, ?wf2 v("v2net" ?result
"tran"), ?value2 2.5, ?edge2 "rising", ?nth2 1, ?tol2 0.2e-6,  ?td2
nil , ?stop nil,  ?period1 1 ?period2 1 ?multiple t ?xName "trigger" )
```

Now, when you evaluate this expression, the following output waveform showing the delay between the two signals is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `delay` is:

```
delay(
     [ ?wf1 o_waveform1 ]
     [ ?value1 n_value1 ]
     [ ?edge1 s_edge1 ]
```


© 2023 Cadence Design Systems, Inc.

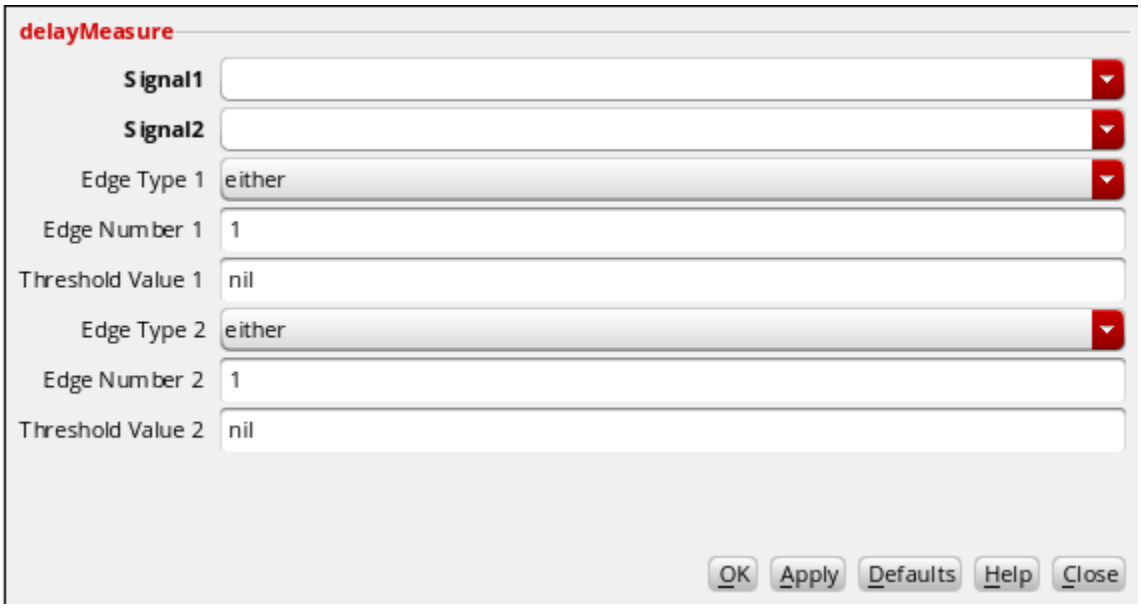
```
[ ?nth1 x_nth1 ]
[ ?td1 n_td1 ]
[ ?wf2 o_waveform2 ]
[ ?value2 n_value2 ]
[ ?edge2 s_edge2 ]
[ ?nth2 x_nth2
[ ?td2 n_td2 ]
[ ?td2r0 n_td2r0 ]}
[ ?stop n_stop ]
[ ?histoDisplay g_histoDisplay ]
[ ?noOfHistoBins x_noOfHistoBins ]
[ ?period1 period1 ]
[ ?period2 period2 ]
[ ?multiple multiple ]
[ ?xName xName ]
@rest args
)
=> o_waveform / n_value / nil
```

For more information about this OCEAN function, see delay in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

delayMeasure

Calculates the delay between a trigger event and a target event.



This function includes the following arguments:

- *Signal1*—Name of the first signal.
- *Signal2*—Name of the second signal.
- *Edge Type 1*—Direction of the first crossing event. `rising` directs the function to look for crossings where the Y value is increasing, `falling` for crossings where the Y value is decreasing, and `either` for crossings in either direction. It is an optional argument.
- *Edge Number 1*—Number that specifies which crossing is to be the trigger event. For example, if *Edge Number1=2*, the trigger event is the second edge of the first waveform with the specified type that crosses *Threshold Value 1*. It is an optional argument.
- *Threshold Value 1*—First threshold to be crossed. It is an optional argument. If the threshold value is nil or blank, threshold is calculated internally using `average(wave1)`.
- *Edge Type 2*—Direction of the second crossing event. `rising` directs the function to look for crossings where the Y value is increasing, `falling` for crossings where the Y value is decreasing, and `either` for crossings in either direction. It is an optional argument.
- *Edge Number 2*—Number that specifies which crossing is to be the trigger event. For example, if *Edge Number2=2*, the trigger event is the second edge of the second

© 2023 Cadence Design Systems, Inc.

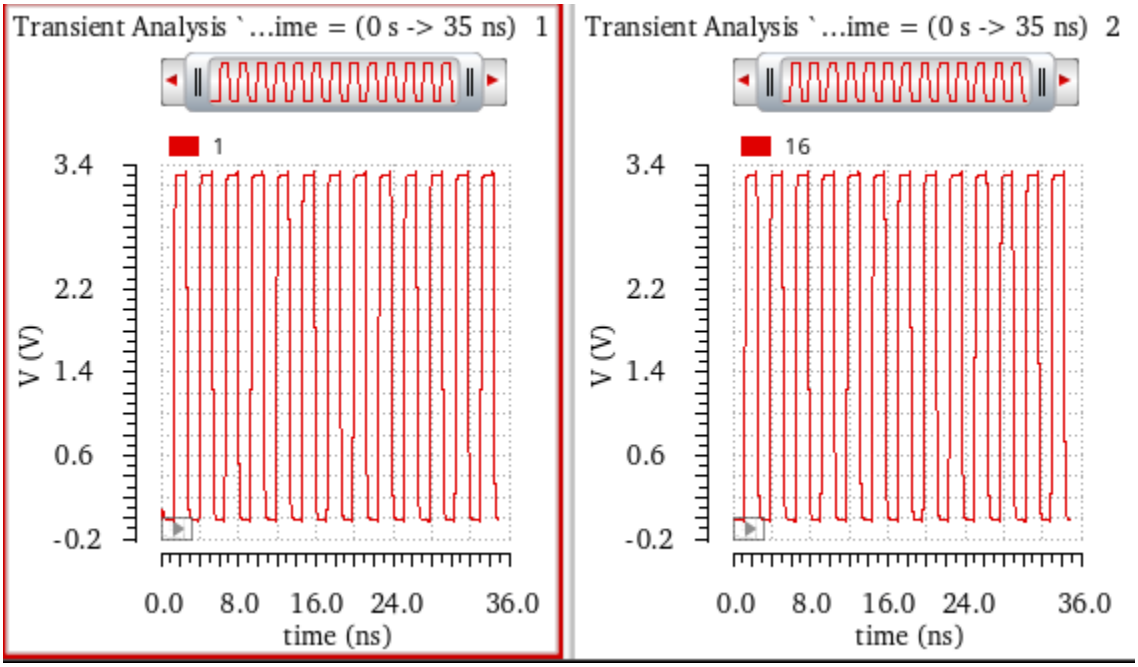
waveform with the specified type that crosses *Threshold Value 2*. It is an optional argument.

- *Threshold Value 2*—Second threshold to be crossed. It is an optional argument. If the threshold value is nil or blank, threshold is calculated internally using `average(wave2)`.

Note: The `delayMeasure` function outputs the same scalar value as that of the `delay` function when applied for a single occurrence.

Example

Consider the following example in which the `delayMeasure` function is applied on two transient waveforms shown in the figure below:



Specify only the required arguments:

- *Signal1*—`v("1" ?result "tran" ?resultsDir "nand2_ring.raw")`
- *Signal2*—`v("16" ?result "tran" ?resultsDir "nand2_ring.raw")`

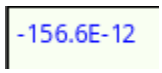
Note: The default values are used for the remaining optional arguments.

The following expression is created in the Buffer:

```
delayMeasure(v("1" ?result "tran") v("16" ?result "tran") )
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output measuring the delay in two signals is displayed in the Buffer:



Example 2

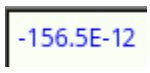
In this example, specify the following argument values:

- *Signal1*—`v("1" ?result "tran" ?resultsDir "nand2_ring.raw")`
- *Signal2*—`v("16" ?result "tran" ?resultsDir "nand2_ring.raw")`
- *Edge Type 1*—`rising`
- *Edge Number 1*—`2`
- *Threshold Value 1*—`2.5`
- *Edge Type 2*—`rising`
- *Edge Number 2*—`2`
- *Threshold Value 2*—`2.5`

The following expression is created in the Buffer:

```
delayMeasure(v("1" ?result "tran") v("16" ?result "tran") ?nth1 2
?value1 2.5 ?nth2 2 ?value2 2.5 )
```

Now, when you evaluate this expression, the following output measuring the delay in two signals is displayed in the Buffer:



Related OCEAN Function

The equivalent OCEAN command for `delayMeasure` is:

```
delayMeasure(
     o_waveform1
     o_waveform2
     [ ?edge1 s_edge1 ]
     [ ?nth1 x_nth1 ]
     [ ?value1 n_value1 ]
```


© 2023 Cadence Design Systems, Inc.

```
[ ?edge2 s_edge2 ]
[ ?nth2 x_nth2
[ ?value2 n_value2 ]
=> n_value / nil
```

For detailed information, see delayMeasure in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

deriv

Returns the derivative of a signal with respect to the X-axis.

Note:

- After the second derivative, the results become inaccurate because the derivative is obtained numerically.
- Use the magnitude value instead of dB in frequency domain.

Additional Information

The following three methods can be used to calculate the finite difference:

- Backward differencing, `[f(x)-f(x-h)]/h at X`
- Forward differencing, `[f(x+h)-f(x)]/h at X`
- Centered differencing, `[f(x+h)-f(x-h)]/2h at X`

The `deriv` function uses the backward differencing method. For better understanding, consider the below example in which you have the following points:

```
(28C, 1.002V)

(30C, 1.012V)

(32C, 1.03V)
```

When you sweep from `0` to `100`, the derivative is calculated as:

```
(30C, [1.012-1.002]/[30-28]) = (30C, 0.005V)
```

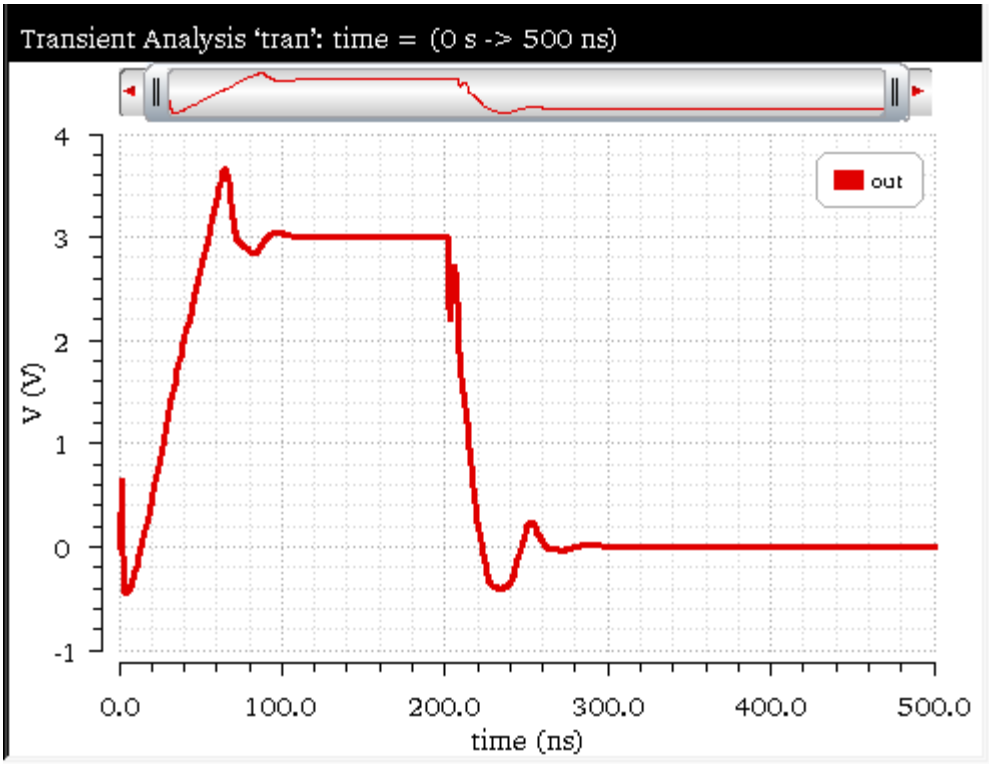
When you sweep from `100` to `0`, the derivative is calculated as:

```
(30C, [1.012-1.03]/[30-32]) = (30C, 0.009V)
```


© 2023 Cadence Design Systems, Inc.

Example

This example shows the output waveform generated when you apply the `deriv` function on the following input waveform, `(v("out" ?result "tran-tran")`:

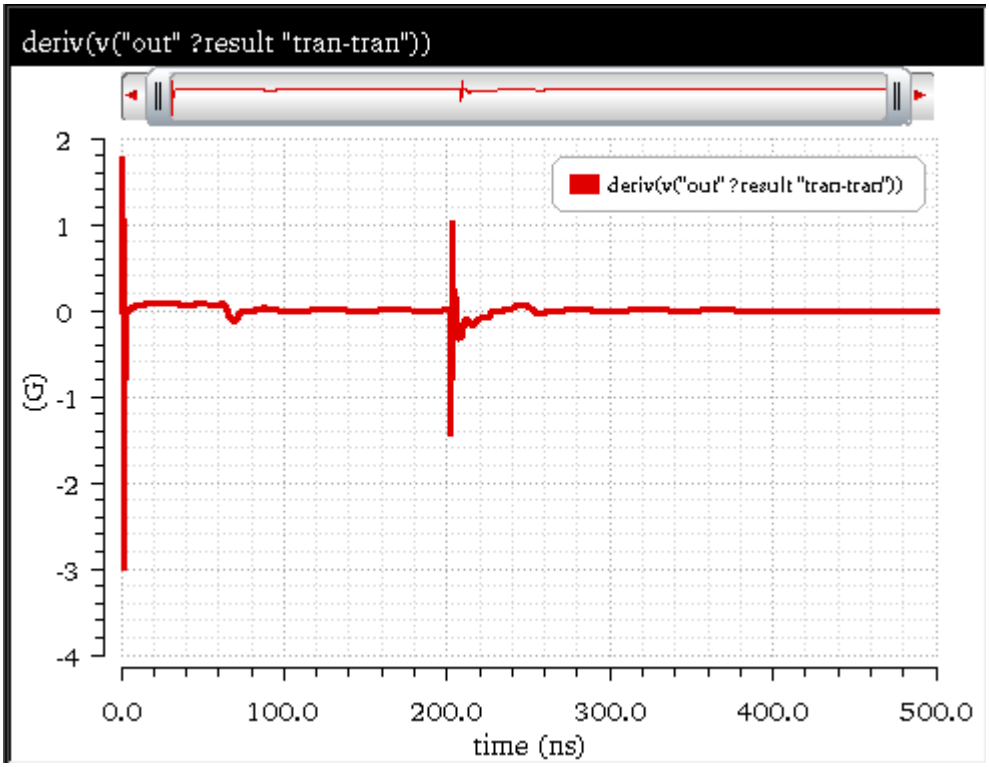


When you apply the `deriv` function on this waveform, the following expression is created in the Buffer:

```
deriv(v("out" ?result "tran-tran")
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform showing the derivative is displayed in the graph window.



Related OCEAN Function

The equivalent OCEAN command for `deriv` is:

```
deriv( o_waveform )
     => o_waveform/nil
```

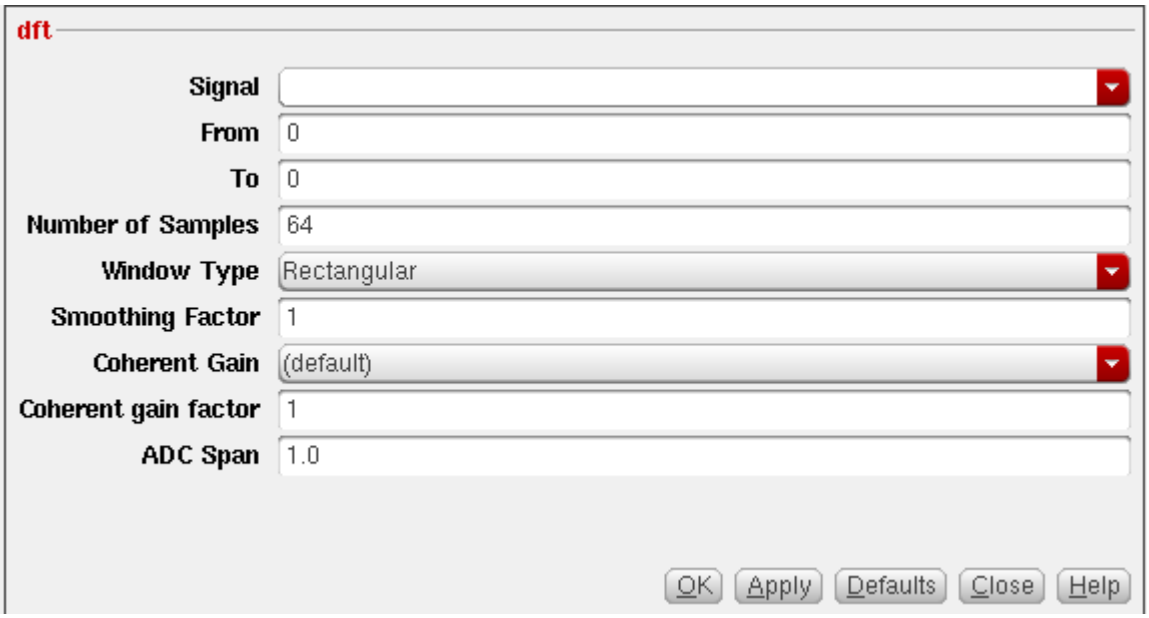
For more information about this OCEAN function, see deriv in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

dft

(Discrete Fourier Transform)

Computes the discrete Fourier transform of a signal. This function is available only in the SKILL mode.



The tool which converts a temporal (time domain) description of a signal (real or complex) into one in terms of its frequency components is called the Fourier Transform. dft (Discrete Fourier Transform) is the discrete formulation of the Fourier Transform, which takes such regularly spaced data values (samples in time domain), and returns the value of the Fourier Transform for a set of values in frequency domain which are equally spaced. Most of the time, however, we work on real-valued signals only.

Consider a complex series (signal) `w(k)` with `N` samples of the form:

```
w(0), w(1), w(2), ..., w(k), ..., w(N-1)
```

Further, assume that the series outside the range `0, N-1` is extended `N`-periodic, that is, `w(k)=w(k+N)` for all `k`. The dft of this series will be denoted `W(n)`, will also have `N` samples and will be defined as:

$$W(n) = \frac{1}{N}\sum_{k=0}^{N-1} w(k)\left(e^{-2\pi i k\frac{n}{N}}\right)$$ where *n=0, ..., N-1*

© 2023 Cadence Design Systems, Inc.

- The first sample `W(0)` of the transformed series is the DC component, more commonly known as the average of the input series.
- The dft of a real series results in a symmetric series about the Nyquist frequency (described below).
- The highest positive (or negative) frequency sample is called the Nyquist frequency. This is the highest frequency component that should exist in the input series for the DFT to receive 'unpredictable' results. More specifically, if there are no frequencies above Nyquist frequency, the original signal can be reconstructed from the samples. The Nyquist Theorem (or Shannon's Sampling Theorem) specifies this – that for a band limited signal, you must sample at a frequency over twice the maximum frequency of the signal to reconstruct it from the samples.

While the dft transform above can be applied to any complex valued series, in practice for large series it can take considerable time to compute, the time taken being proportional to the square of the number of points (samples) in the series. A much faster algorithm has been developed by Cooley and Tukey called the FFT (Fast Fourier Transform). The only requirement of the most popular implementation of this algorithm (Radix-2 Cooley-Tukey) is that the number of points in the series be a power of `2` i.e. $N=2^n$.

Given *N* input points, the fft returns *N* frequency components, of which the first `(N/2 +1)` are valid. (The other components are mirror images and are considered invalid since the frequencies they represent do not satisfy the Nyquist Theorem above.) They start with the DC component, and are spaced apart by a frequency of `(1/(n deltaT))`. The magnitude of the complex number returned is the frequency's relative strength.

The dft function computes the discrete Fourier Transform of the buffer by fft algorithm where `deltaT=(t2-t1)/N`. The waveform is sampled at the following *N* timepoints:

*t1, t1 + deltaT, t1 + 2 * deltaT, º, t1 + (N - 1) * deltaT*

The output of `dft()` is a frequency waveform, `W(f)`, which has `(N/2+1)` complex values: the dc term, the fundamental, and `(N/2-1)` harmonics.

Note: The last time point, `(t1+(N - 1)*deltaT)`, is `(t2-deltaT)` rather than `t2`. The `dft` function assumes that `w(t1)` equals `w(t2)`.

This function includes the following arguments:

- *Signal*—Name of the signal.
- *From*—Starting point of the range over which you want to compute the transform.
- *To*—Ending point of the range over which you want to compute the transform. Be sure to cover at least one complete period of your slowest frequency.

© 2023 Cadence Design Systems, Inc.

- *Number of Samples*—Number of samples you want to take in expanding the Fourier transform. Sample at a rate that is at least twice your highest frequency component (the Nyquist rate). Pick a sampling rate high enough that closely spaced frequency components can be resolved.
- *Window Type*—Window you want to use.

  Valid values: `Blackman`, `BlackmanHarris`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hanning`, `Hamming`, `Kaiser`, `Nuttall`, `Parzen`, `Rectangular`, and `Triangular`.
  Default value: `Rectangular`.
- *Smoothing Factor*—Smoothing factor applicable to the Kaiser window only.
  The Smoothing Factor field accepts values from 0 to 15. The value 0 implies no smoothing and is equivalent to a rectangular window.
- *Coherent Gain*—Scaling parameter. A non-zero value scales the power spectral density by 1/(f_cohGain). Valid values: 0 <= f_cohGain <= 1. You can use `1` if you do not want the scaling parameter to be used. Default value: `1`.

  The default coherent gain is the gain of the windowing function. If you want to specify your own coherent gain value, you can choose one of the following option in the *Coherent gain* drop-down list and then specify the value in the *Coherent gain factor* field:

  - `none`—The window function is applied to the input and the DFT is calculated. No scaling is performed on the DFT output.
  - `default`—The DFT output is divided by the gain of the windowing function.
  - `magnitude`—The scaling factor is the value provided in the *Coherent gain factor* field.
  - `dB20`—The DFT output is divided by the value provided in the *Coherent gain factor* field. The Coherent gain factor is specified in the dB20 units.
  - `dB10`—The DFT output is divided by the value provided in the *Coherent gain factor* field. The Coherent gain factor is specified in the dB10 units.
- *Coherent gain factor*—When you do a dft, the applied *Window Type* (aside from the rectangular type) changes the signal's amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.
- *ADC Span*—Specifies the peak saturation level of the FFT waveform. When specified the magnitude of the input waveform is divided by adc span value before computing FFT. This is full-scale span ignoring any DC offsets.
  Valid values : Any floating point number
  Default Value : 1.0

© 2023 Cadence Design Systems, Inc.

When you run the transient analysis, keep the maximum time step small enough to represent the highest frequency component accurately. The maximum time step should be smaller than the sampling period that you use for the dft of the time domain waveform. The samples in the dft will either find a data point (calculated by the simulator) or an interpolated point between two data points.

Choosing a maximum timestep during transient simulation that is smaller than the dft sampling period ensures that sampling maintains a resolution at least equal to that of the transient time-domain waveform.

The start and stop times should not coincide with the boundaries of the time-domain waveform. The boundary solutions might be imprecise and generate incorrect results if used in other calculations.

One of the uses of fast Fourier Transform windowing is to reduce discontinuities at window edges caused by having a non integral number of periods of a signal in a window. This removes the abrupt edges, making them fall off smoothly to zero, and can improve the validity of the fft components obtained. You can also use fft windowing to 'dig out' the details of signal components that are close Gin frequency or that consist of both large and small amplitudes.

The DFT function interpolates the From and To points in the following three cases:

- The specified number of samples in the DFT function is higher than the number of raw data points.
- The raw data points are non-uniform.
- The From and To points are not aligned with the raw data points.

  A warning message also appears to inform about the interpolation.

  Note: The interpolation warning message is not displayed when the time distance between a sample point and its closest raw data point is less than `0.01*step`, where `step` is the average step size of the data points in the range [`from,to`]

  ```
  step = (to-from) / number of raw data points in [from,to]
  ```

  The DFT function assumes that `w(from)` equals to `w(to)`. A warning message appears when `w(from)` is not equal to `w(to)` in the following situation:

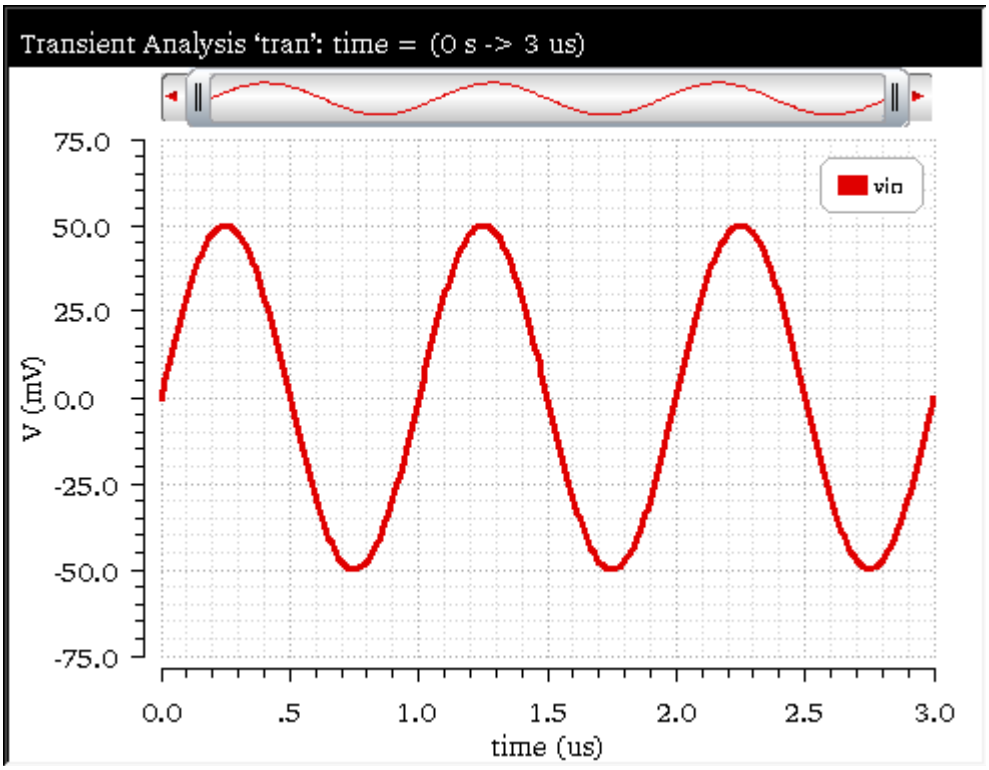
  window function is Rectangular or not specified, and

  `|w(from)-w(to)|>1e-3*range(w)`, where `range(w)` is `max(w)- min(w)` in `[from,to]`

© 2023 Cadence Design Systems, Inc.

Example

This example shows the DFT output generated when you apply the `dft` function on the following input signal:



The following arguments are specified in this example:

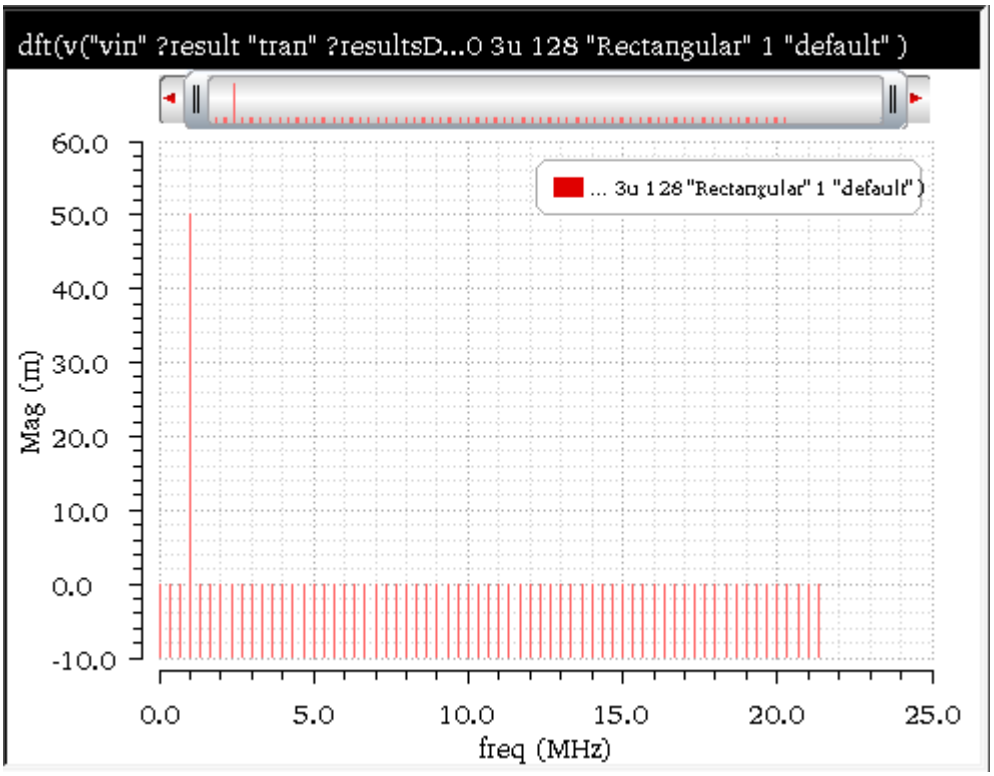
- *Signal*—`v("vin" ?result "tran")`
- *From*—`0`
- To—`3u`
- *Number of Samples*—`128`
- *Window Type*—`Rectangular`
- *Smoothing Factor*—`1`
- *Coherent Gain*—`(default)`
- *Coherent gain factor*—`1`

The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
dft(v("vin" ?result "tran") 0 30u 128 "Rectangular" 1 "default" )
```

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `dft` is:

```
dft( o_waveform n_from n_to x_num [t_windowName [n_param1]] )
     => o_waveform/nil
```

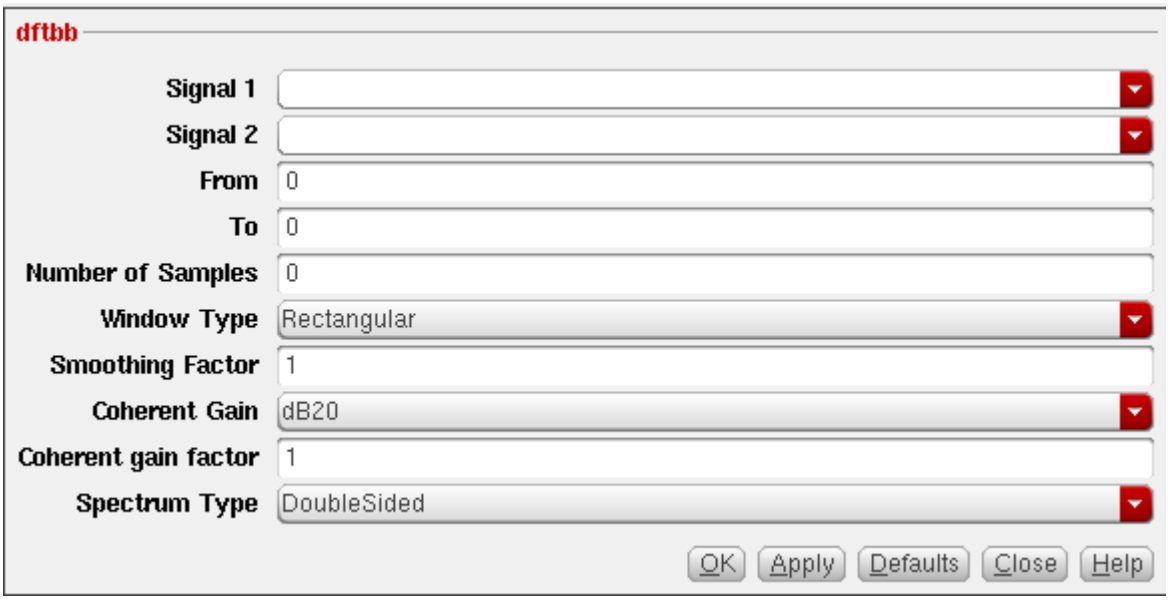
For more information about this OCEAN function, see dft in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

dftbb

(Discrete Fourier Transform Baseband)

Computes the Discrete Fourier Transform (fast Fourier transform) of a complex signal. This function is available only in the SKILL mode.



This function includes the following arguments:

- *Signal1*—First waveform.
- *Signal2*—Second waveform.
- *From*—Starting point of the range over which you want to compute the transform.
- *To*—Ending point of the range over which you want to compute the transform. Be sure to cover at least one complete period of your slowest frequency.
- *Number of Samples*—Number of samples you want to take in expanding the Fourier transform. Sample at a rate that is at least twice your highest frequency component (the Nyquist rate). Pick a sampling rate high enough that closely spaced frequency components can be resolved. The default value is `64`.
- *Window Type*—Window you want to use.
  Valid values: `Blackman`, `BlackmanHarris`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hanning`, `Hamming`, `Kaiser`, `Nuttall`, `Parzen`, `Rectangular`, and `Triangular`.
  Default value: `Rectangular`.

© 2023 Cadence Design Systems, Inc.

- *Smoothing Factor*—Smoothing factor applicable to the Kaiser window only. The Smoothing Factor field accepts values from 0 to 15. The value 0 implies no smoothing and is equivalent to a rectangular window.
- *Coherent Gain*—Scaling parameter. A non-zero value scales the power spectral density by 1/(f_cohGain). Valid values: 0 <= f_cohGain <= 1. You can use `1` if you do not want the scaling parameter to be used. Default value: `1`.
- *Coherent gain factor*
  When you do a dftbb, the applied *Window Type* (aside from the rectangular type) changes the signal's amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.
- *Spectrum Type*—Specifies the type of spectrum that can be either singleSided or doubleSided. When *Spectrum Type* is single-sided, the resultant waveform is only on one side of the Y-axis starting from 0 to `N-1`. When it is double- sided, the resultant waveform is symmetric to the Y-axis from `-N/2` to `N/2`.

Additional Information

Complex signal `z(t)= x(t) + j*y(t)`:

```
N-1
Z(n)= ReZ(n) +j*ImZ(n)=SUM[ z(k)*exp(-j*theta*n*k)],
k=0
```

where,

```
theta=2*Pi/N; n=0, 1, ..., N-1.
```

Both waveforms are sampled at the following `N` timepoints:

```
t1, t1 + deltaT, t1 + 2 * deltaT, ..., t1 + (N - 1) * deltaT
```

The output of dftbb `(waveform1, waveform2)` are `N` complex values.

The dftbb function is required because the dft function gives out the amplitudes `(sqrt(Re**2+Im**2))` of dfts of real signals only – not Re and Im. Therefore, you cannot replace one dft of the complex signal `z(t)= i(t) + j*q(t)` with two dfts of two real signals `i(t)` and `q(t)`:

```
N-1
I(n)= ReI(n) +j*ImI(n)= SUM[ i(k)*exp(-j*theta*n*k)],
k=0
N-1
Q(n)= ReQ(n) +j*ImQ(n)= SUM[ q(k)*exp(-j*theta*n*k)],
k=0
```


© 2023 Cadence Design Systems, Inc.

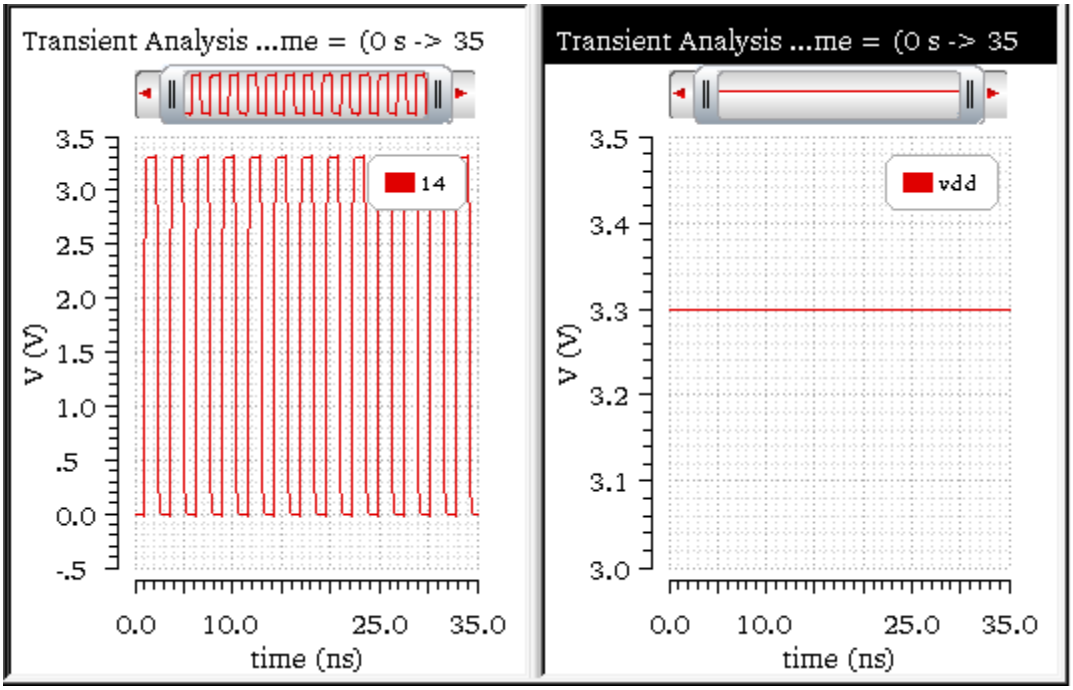
and then compute:

```
ReZ(n)= ReI(n) - ImQ(n);
ImZ(n)= ImI(n) + ReQ(n); for n=0, 1, ..., N-1.
```

The above definition is for single-sided output waveforms. This holds true for double-sided output waveforms except that the previous output waveform is translated so that `n` varies from `-N/2 to (N/2)-1`.

Example

This example displays the output DFT waveform when you apply the `dftbb` function on the following input waveforms:



The following arguments are specified in this example:

- *Signal1*—`v("14" ?result "tran")`
- *Signal2*—`v("vdd" ?result "tran")`
- *From*—`0`
- *To*—`30u`
- *Number of Samples*—`512`

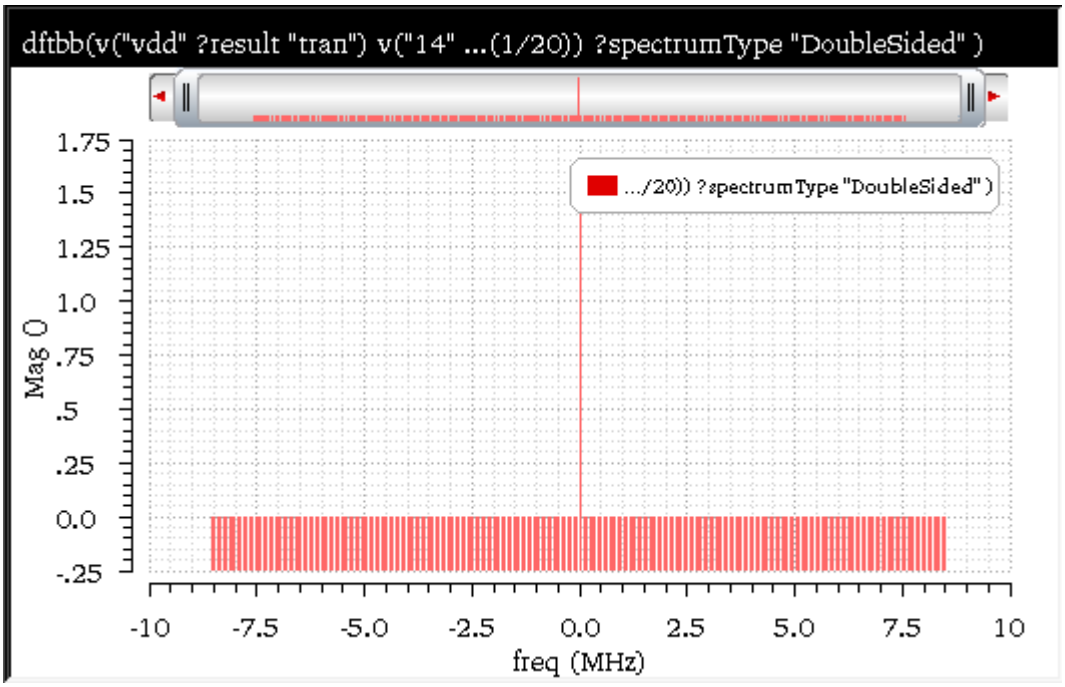
© 2023 Cadence Design Systems, Inc.

- *Window Type*—`Rectangular`
- *Smoothing Factor*—`1`
- *Coherent Gain*—`dB20`
- *Coherent gain factor*—`1`
- *Spectrum Type*—`DoubleSided`

The following expression is created in the Buffer:

```
dftbb(v("14" ?result "tran" ?resultsDir "nand2_ring.raw") v("vdd"
?result "tran" ?resultsDir "nand2_ring.raw") 0 30u 512 ?windowName
"Rectangular" ?smooth 1 ?cohGain (10**(1/20)) ?spectrumType
"DoubleSided" )
```

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `dftbb` is:

© 2023 Cadence Design Systems, Inc.

```
dftbb( o_waveform1 o_waveform2 f_timeStart f_timeEnd x_num
     ?windowName t_windowName ?smooth x_smooth ?cohGain f_cohGain
     ?spectrumType s_spectrumType)
     => o_waveformComplex/nil
```

For more information about this OCEAN function, see dftbb in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

dnl

(Differential Non-Linearity)

Computes the differential non-linearity of a transient simple or parametric waveform.

DNL can be calculated as:

- For `lsb`:

$$DNL(i+1) = \frac{V_{out}(i+1) - V_{out}(i)}{\text{Ideal LSB size}} - 1$$

- For `abs`:

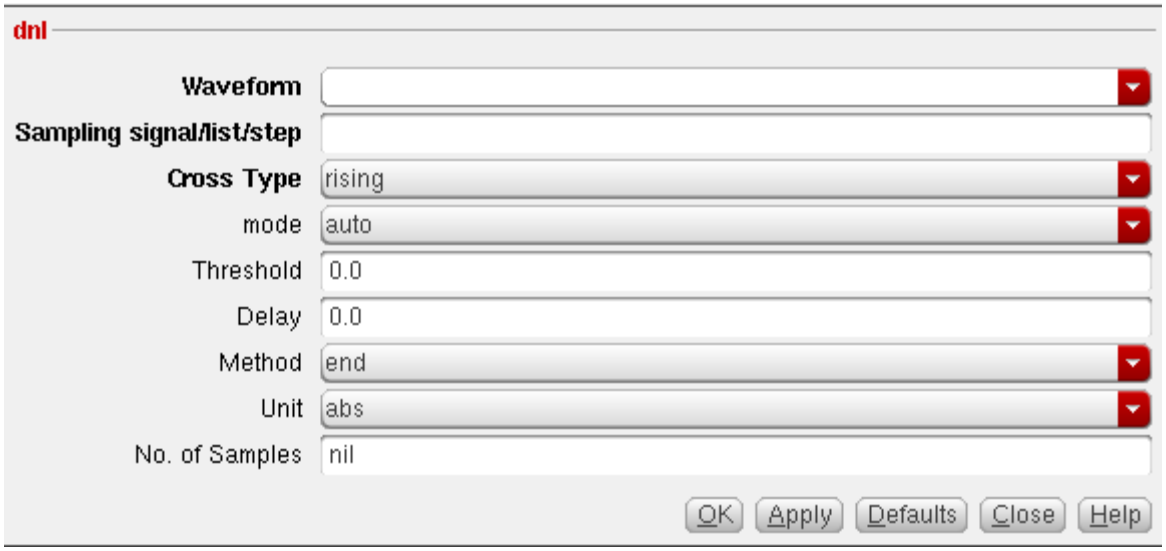
$$DNL(i+1) = [V_{out}(i+1) - V_{out}(i)] - \text{Ideal LSB size}$$

To calculate the Ideal LSB size, a straight line is drawn on the input signal. If the *method* is `end`, the line is drawn by connecting first and last samples. If the *method* is `fit`, the line is a best-fit line based on all samples.

The Ideal LSB size is calculated as follows:

$$\text{Ideal LSB size} = \frac{(\text{the last y value on the line} - \text{the first y value on the line})}{\text{number of steps}}$$

© 2023 Cadence Design Systems, Inc.



This function includes the following arguments:

- *Waveform*—Name of the signal.
- *Sampling signal/list/step*—Signal used to obtain the points for sampling the *Waveform* (points at which the waveform crosses the threshold while either `rising` or `falling` as specified in the *cross Type* field with the *delay* added to them), list of domain values at which the sample points are obtained from the Waveform, or the sampling interval.
- *Cross Type*—Specifies the points at which the curves of the waveform intersect with the threshold. While intersecting, the curve may be either rising (`rising`) or falling (`falling`). Possible values are: `rising`, `falling`, and `either`.
- *mode*—Specifies the mode used to calculate the threshold value. If you want to specify the threshold value, select `user`. If you want that the Virtuoso Visualization and Analysis XL calculates the threshold value, select `auto`. The auto threshold is calculated as:

  Auto Threshold Value = integral of the waveform divided by the X range.
- *Threshold*—Threshold value against which the frequency is to be calculated. You need to specify the threshold value only if the *mode* is `user`.
- *Delay*—Delay time after which the sampling begins.

  Note: When sampling interval is specified as a signal or list, delay time is applied to each crossing point. When sampling interval is specified as a number, delay time is applied to waveform start time.

© 2023 Cadence Design Systems, Inc.

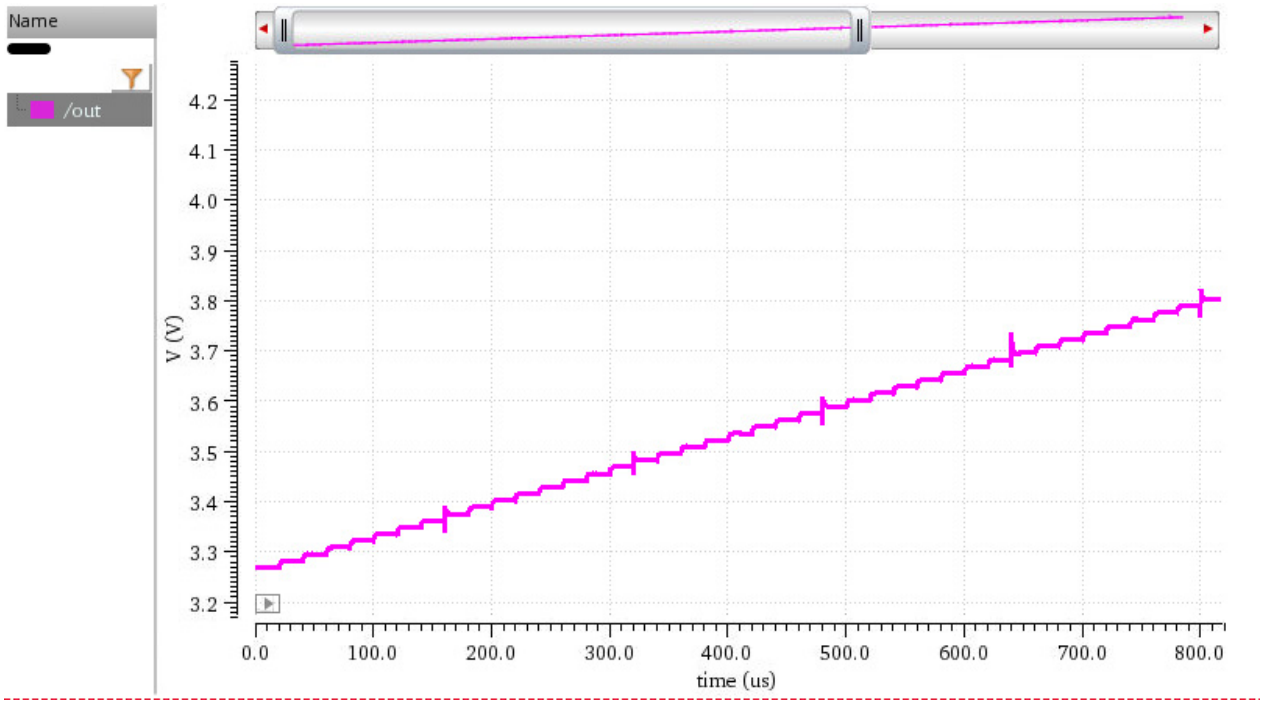
- *Method*—Specifies whether the end-point (`end`) or straight line (`fit`) method is used
- *Unit*—Specifies whether the output waveform is to be output as an absolute value (`abs`) or multiples of least significant bit (`lsb`).
- *No. of Samples*—Specifies the samples used for calculating the non-linearity. If not specified, the samples are taken against the entire data window.

Note: For each of the three ways in which the sample points can be specified, only a few of the other optional arguments are significant,

- For *Sampling signal*, the fields *Cross Type*, *mode*, *Threshold*, *Delay*, *Method*, and *Units* are significant.
- For *list*, the fields *Method* and *Units* are significant.
- For *step*, the fields *Method*, *Units*, and *No. of samples* are significant.

Example

This example shows the output waveform generated when you apply the `dnl` function on the following input signal:



The following arguments are specified in this example:

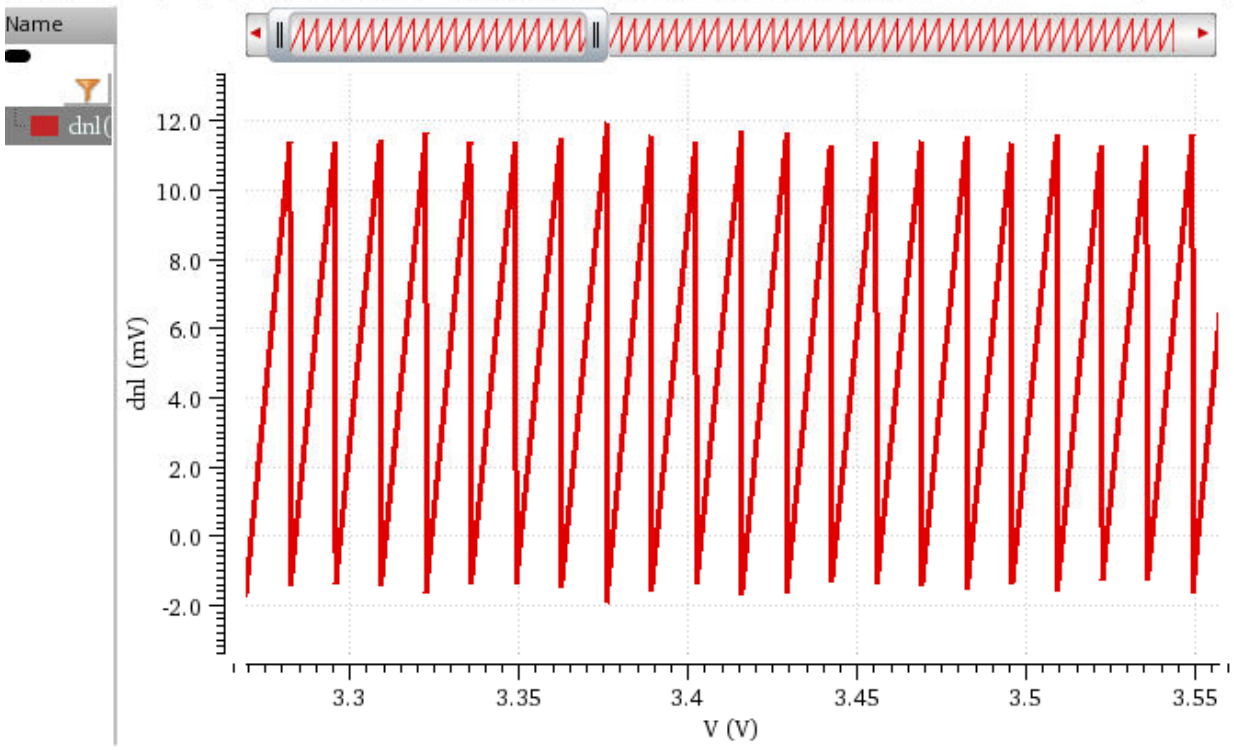
© 2023 Cadence Design Systems, Inc.

- *Waveform*—`v("/out" ?result "tran")`
- *Sampling signal/list/step*—`1e-05`
- *Cross Type*—`rising`
- *mode*—`auto`
- *Threshold*—`0.0`
- *Delay*—`4.5e-06`
- *Method*—`end`
- *Unit*—`abs`
- *No. of Samples*—`512`

The following expression is created in the Buffer with the specified arguments:

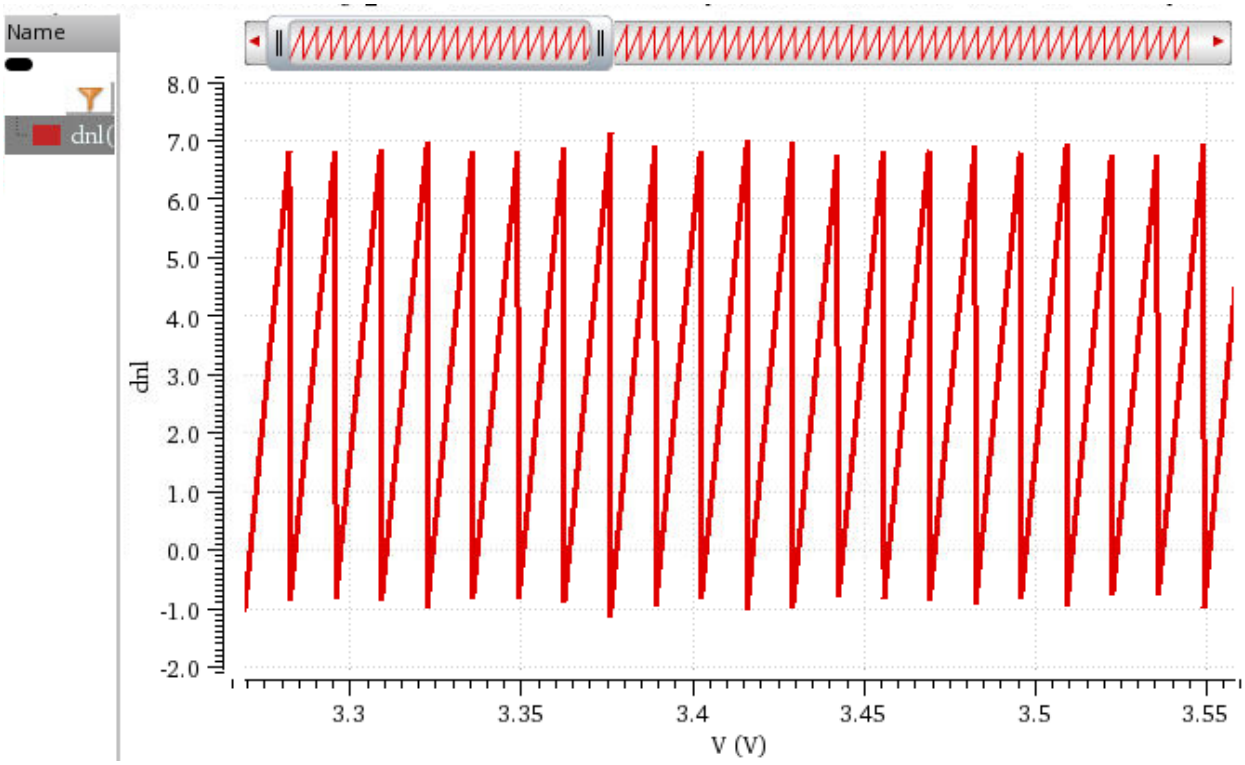
```
dnl(v("/out" ?result "tran") 1e-05  ?mode "auto"  ?crossType "rising" ?delay 4.5e-06 ?method "end" ?units "abs" ?nbsamples 512)
```

Now, when you evaluate this expression, the following output waveform is displayed in the graph window. Notice that the X-axis label of this waveform is same as that of the Y-axis label of the input waveform.



© 2023 Cadence Design Systems, Inc.

Now, if you change the unit to `lsb` and then evaluate the expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `dnl` is:

```
dnl( o_dacSignal o_sample|o_pointList|n_interval
     [?mode t_mode] [?threshold n_threshold] [?crossType t_crossType]
     [?delay f_delay] [?method t_method][?units x_units]
     [?nbsamples n_nbsamples] )
     => n_dnl/nil
```

For more information about this OCEAN function, see dnl in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

dutyCycle

Calculates the ratio of the time for which the signal remains high to the period of the signal. This function should be used for the periodic signals only.

The calculation of duty cycle depends on whether the first complete interval starts with a low or a high interval. Therefore, results may vary depending on start time that determines whether cycles are low-to-high or high-to-low.

If the first complete interval is `high1` and `low1` follows, the period is `high1+low1` and the duty cycle is `high1`/(`high1+low1`). If `low1` is the first complete interval and `high2` follows, the period is `low1+high2` and the duty cycle `high2`/(`low1+high2`).

This function includes the following arguments:

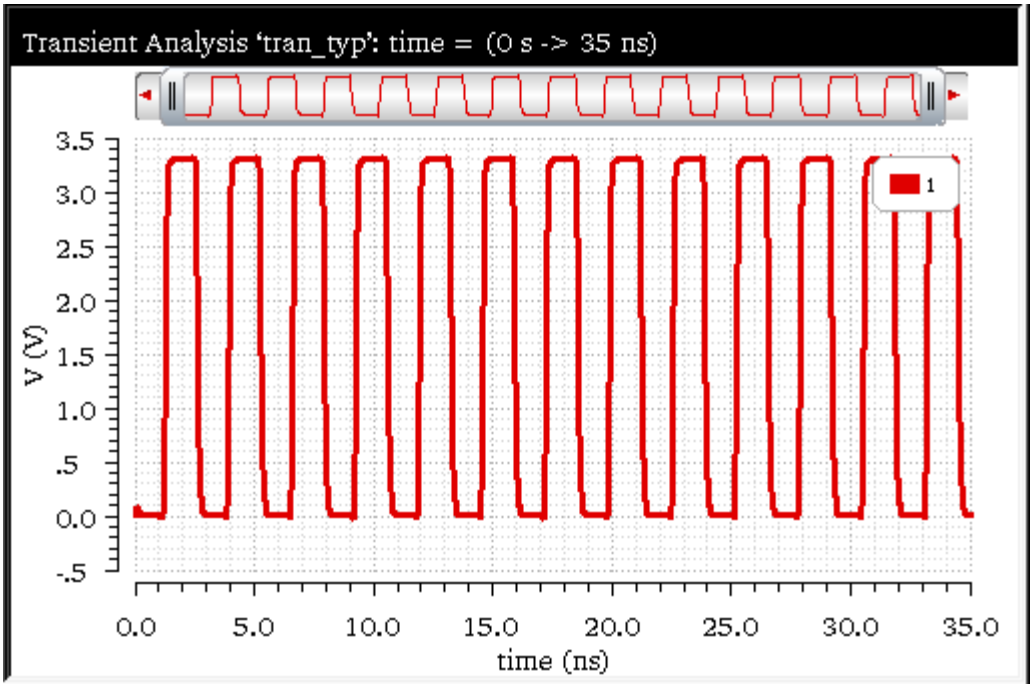
- *Waveform*—Name of the signal, expression, or family of waveforms.
- *mode*—Specifies the mode used to calculate the threshold value. If you want to specify the threshold value, select `user`. If you want that the Virtuoso Visualization and Analysis XL calculates the threshold value, select `auto`. The Auto Threshold Value is calculated as the average which is integral of the waveform divided by the X range.

  Note: For a periodic waveform, if X- axis range is not a multiple of period, then threshold computation using `auto` mode does not represent the threshold. Therefore, in this case, it is recommended to use the mode as `user` and provide a threshold value.
- *threshold*—Threshold value. You need to specify the threshold value only if the *mode* is `user`.
- *Plot/print vs.*— Specifies whether X axis of the output waveform is *time* (or another X-axis parameter for non-transient data) or *cycle*.
- *Output Type*—Type of output. If set to `plot`, the output is a waveform; if set to `average`, the output is an average value. The default value is `plot`.

© 2023 Cadence Design Systems, Inc.

Example

This example shows the output waveform generated when you apply the `dutyCycle` function on the following input waveform:



The following arguments are specified in this example:

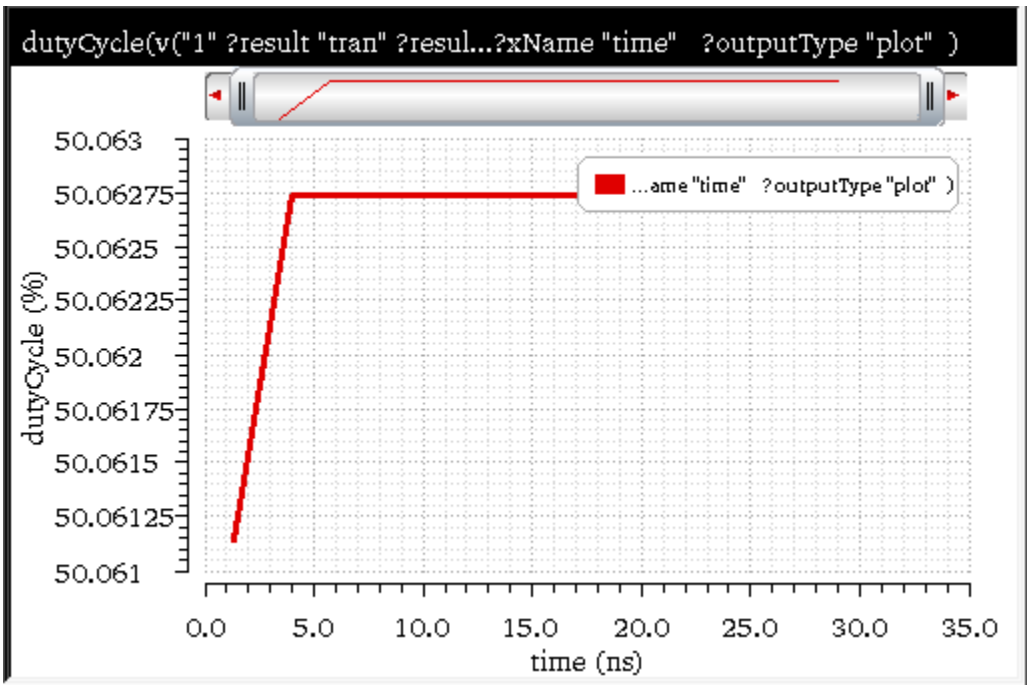
- *Waveform*—`v("1" ?result "tran")`
- *mode*—`auto`
- *threshold*—`0.0`
- *Plot/print vs.*—`time`
- *Output Type*—`plot`

The following expression is created in the Buffer:

```
dutyCycle(v("1" ?result "tran" ?resultsDir "nand2_ring.raw") ?mode
"auto" ?xName "time"   ?outputType "plot"  )
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `dutyCycle` is:

```
dutyCycle( o_waveform
     [?mode t_mode] [?threshold n_threshold]
     [?xName t_xName] [?outputType t_outputType] )
     => o_waveform/f_average/nil
```

For more information about this OCEAN function, see dutyCycle in *OCEAN Reference*.

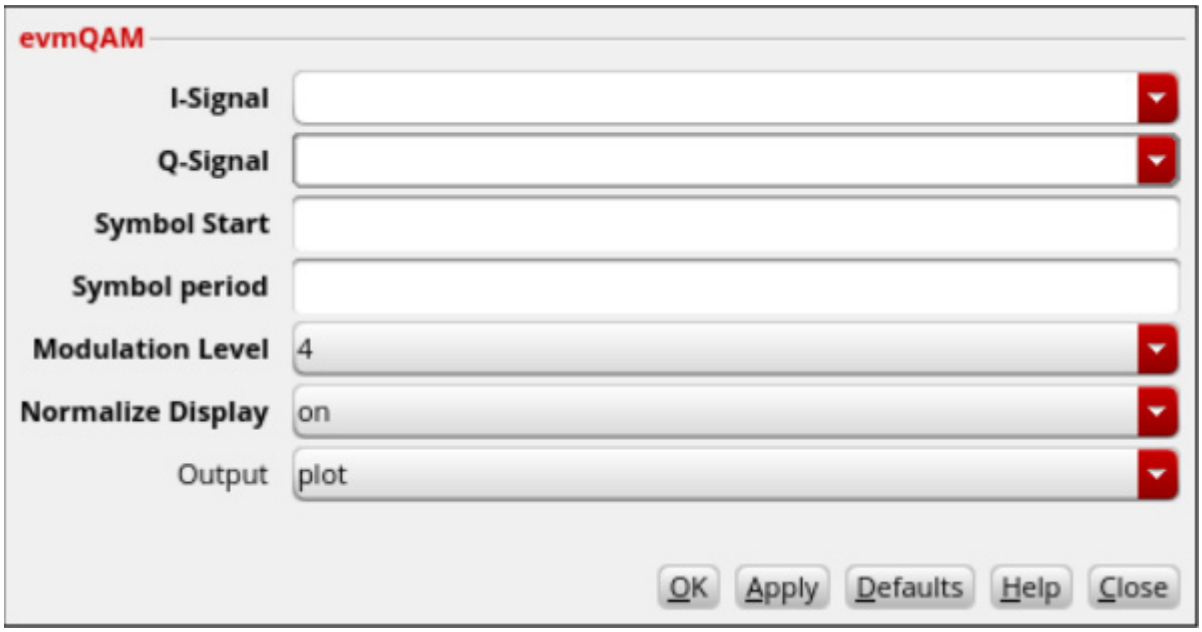
© 2023 Cadence Design Systems, Inc.

evmQAM

(Error Vector Magnitude Quadrature Amplitude Modulation)

Processes the I and Q waveform outputs from the transient simulation run to calculate the Error Vector Magnitude (EVM) for multi-mode modulations. The function plots the I versus Q scatterplot. EVM is a useful measurement to describe the overall signal amplitude and phase modulated signal quality. It is based on a statistical error distribution normalized from an ideal digital modulation. Quadrature Amplitude Modulation (QAM) is a typical modulation scheme where EVM is useful. The EVM is calculated by detecting the I and Q signal levels corresponding to the four possible I and Q symbol combinations and calculating the difference between the original signal level and the ideal signal level.

Note: This function is not supported for families of waveforms. The evmQAM function is available only in the SKILL mode.



This function includes the following arguments:

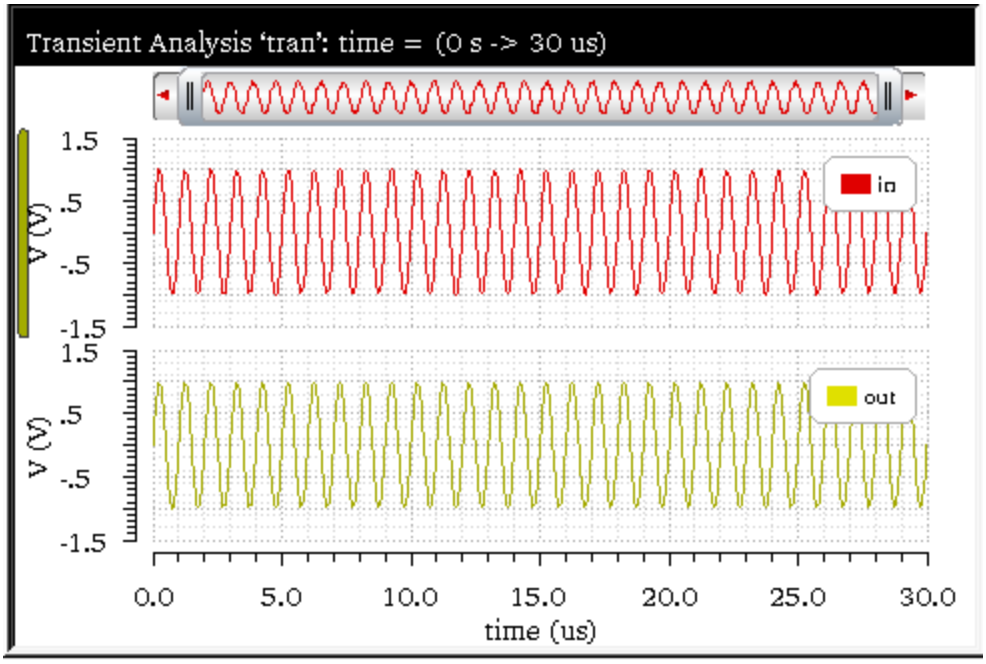
- *I-Signal*—Waveform for the I signal.
- *Q-Signal*—Waveform for the Q signal.
- *Symbol Start*—Start time for the first valid symbol.
- *Symbol period*—Period for the symbol. Each period is represented by a data rate. The data rate at the output is determined by the particular modulation scheme being used. For example, if the data rate selected is 5.5 Mbps, it corresponds to a period of 181.8 ns.

© 2023 Cadence Design Systems, Inc.

- *Modulation Level*—Modulation level. Valid values are 4, 16, 64, and 256.
- *Normalize Display*—Normalizes the scatter plot to the ideal values +1 and -1 (for example, when superimposing scatter plots from different stages in the signal flow, where the levels may be different but the you want to see relative degradation or improvement in the scatter).
- *Output*— Specifies whether to calculate rms or peak EVM. Possible values are: *plot*, *rms*, and *max*. The default value is *plot*, which lets you plot the constellation waveforms.

Example

This example shows the scatter plot generated when you apply `evmQpsk` function on the following input waveforms:



The following arguments are specified in this example:

- *I-Signal*—`v("in" result "tran")`
- *Q-Signal*—`v("out" result "tran")`
- *Symbol Start*—`0`
- *Symbol period*—`1e-8`
- *Modulation Level*—`16`

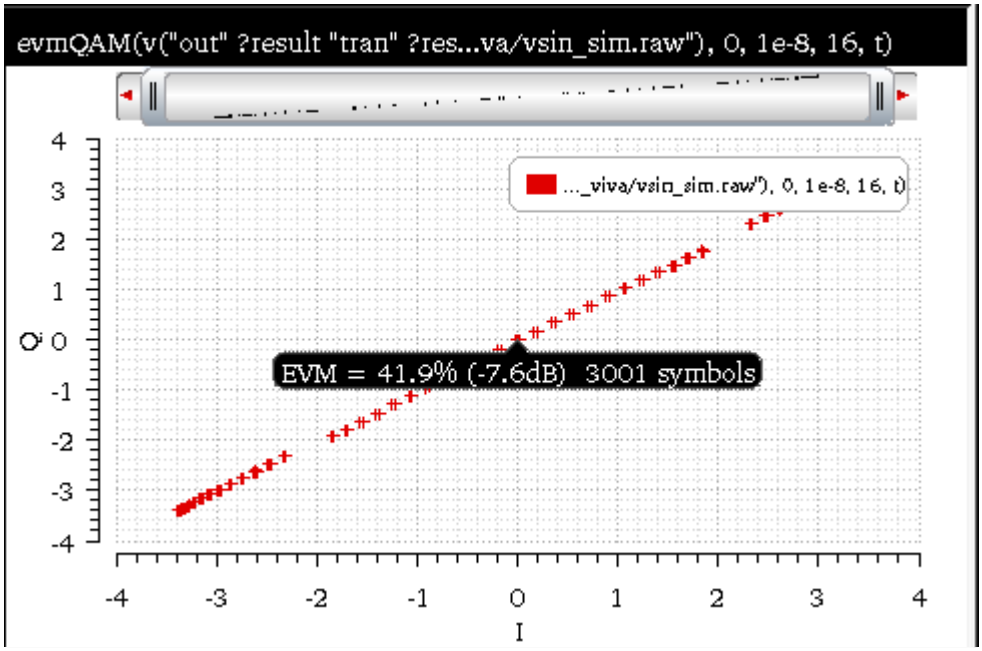
© 2023 Cadence Design Systems, Inc.

- *Normalize Display*—`on`
- Output—`rms`

The following expression is created in the Buffer:

```
evmQAM(v("in" ?result "tran" ?resultsDir "./vsin_sim.raw"), v("out"
?result "tran" ?resultsDir "./vsin_sim.raw"), 0, 1e-8, 16, t ?output
"rms")
```

Now, when you evaluate this expression, the following scatter plot is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN function for `evmQAM` is:

```
evmQAM( o_waveformI o_waveformQ n_tDelay n_samplingT x_levels g_normalize )
     => o_waveform/nil
```

For more information about this OCEAN function, see evmQAM in *OCEAN Reference*.

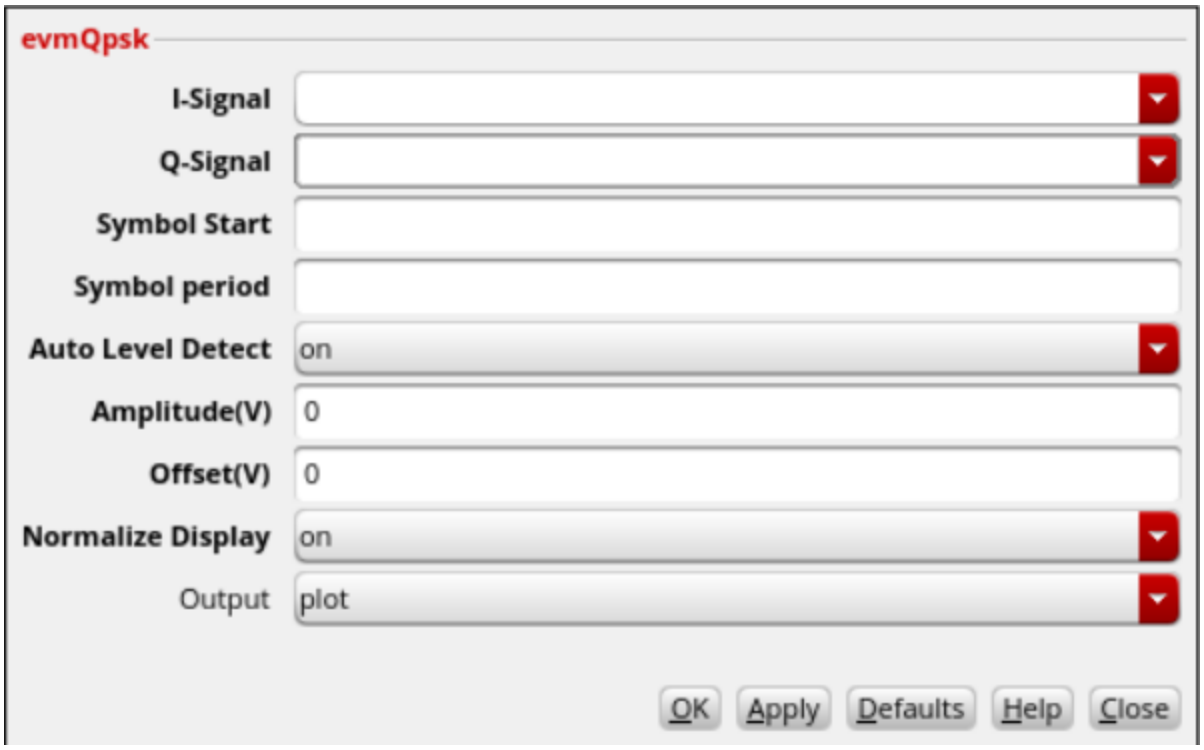
© 2023 Cadence Design Systems, Inc.

evmQpsk

(Error Vector Magnitude Quadrature Phase Shift Keying)

Processes the I and Q waveform outputs from the transient simulation run to calculate the Error Vector Magnitude (EVM) and plot the I versus Q scatter plot. EVM is a useful measurement to describe the overall signal amplitude and phase modulated signal quality. It is based on a statistical error distribution normalized from an ideal digital modulation. Quadrature Phase Shift Keying (QPSK) is a typical modulation scheme where EVM is useful. The EVM is calculated by detecting the I and Q signal levels corresponding to the four possible I and Q symbol combinations and calculating the difference between the original signal level and the ideal signal level.

Note: This function is not supported for families of waveforms. This function is available only in the SKILL mode.



This function includes the following arguments:

- *I-Signal*—Waveform for the I signal.
- *Q-Signal*—Waveform for the Q signal.
- *Symbol Start*—Start time for the first valid symbol.

© 2023 Cadence Design Systems, Inc.

- *Symbol period*—Period for the symbol. Each period is represented by a symbol rate. The symbol rate at the output is determined by the particular modulation scheme being used.
  Symbol Period = 1/(Symbol Rate)
  Symbol Rate = (Data Rate)/(Number of bits per symbol)

  For example, if the data rate selected is 5.5 Mbps and for QPSK (2 bits are conveyed per symbol)

  Symbol rate = 5.5e6/2 = 2.75e6

  Symbol period = 1/2.75M = 3.636364e-07s

- *Auto Level Detect* `on` indicates that you want the amplitude (*Amplitude*) and DC offset (*Offset*) to be calculated automatically. *Amplitude* is calculated by averaging the rectified voltage level of the signal streams and DC *Offset* by averaging the sum of an equal number of positive and negative symbols in each signal stream. These values are used to determine the EVM value. If *Auto Level Detect* is set to `off`, you must specify values for the *Amplitude* and *Offset* fields.

- *Amplitude(V)* is the amplitude of the signal. You need to specify a value in this field only if *Auto Level Detect* is set to `off`.

- *Offset(V)* is the DC offset value. You need to specify a value in this field only if *Auto Level Detect* is set to `off`.

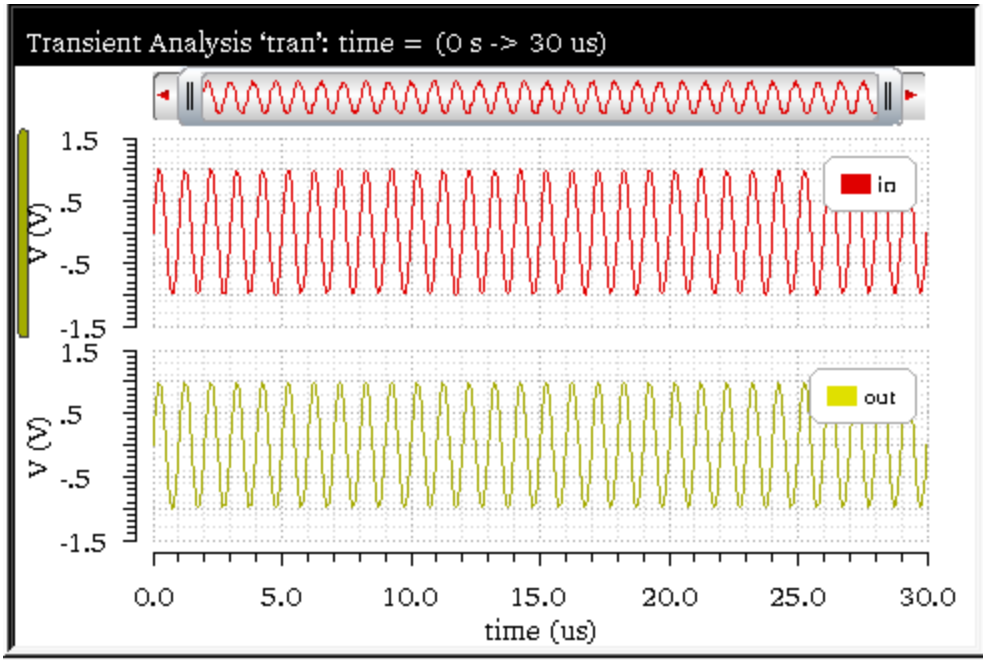
- *Normalize Display* normalizes the scatter plot to the ideal values +1 and -1 (for example, when superimposing scatter plots from different stages in the signal flow, where the levels may be different but the you want to see relative degradation or improvement in the scatter).

- *Output*— Specifies whether to calculate rms or peak EVM. Possible values are: *plot*, *rms*, and *max*. The default value is *plot*, which lets you plot the constellation waveforms.

© 2023 Cadence Design Systems, Inc.

Example

This example shows the scatter plot generated when you apply `evmQpsk` function on the following input waveforms:



The following arguments are specified in this example:

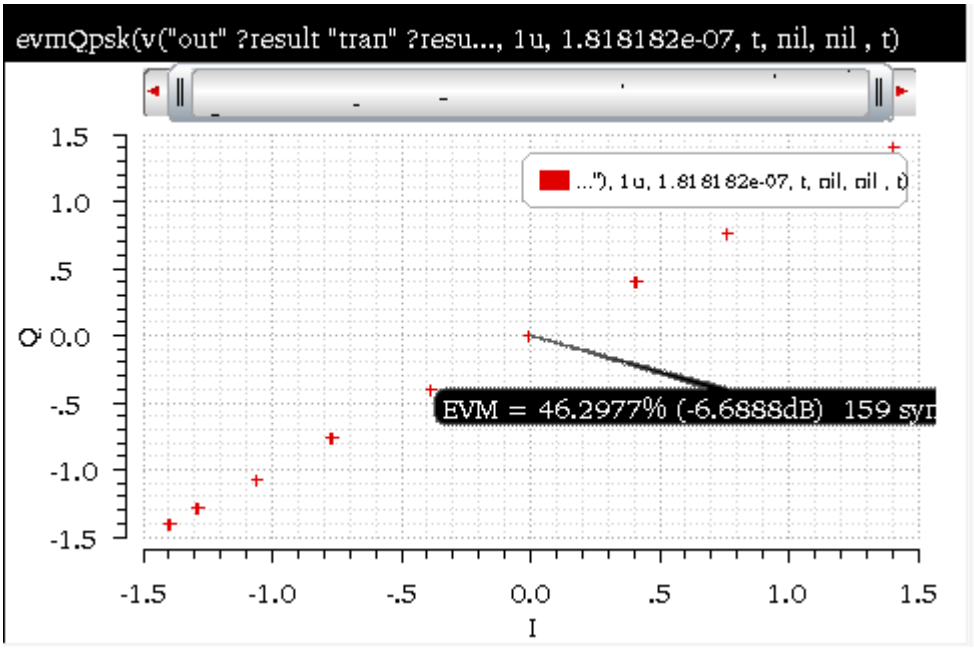
- *I-Signal*—`v("in" result "tran")`
- *Q-Signal*—`v("out" result "tran")`
- *Symbol Start*—`1u`
- *Symbol period*—`1.818182e-07`
- *Auto Level Detect*—`on`
- *Amplitude(V)*—`0`
- *Offset(V)*—`0`
- *Normalize Display*—`on`
- *Output*—`rms`

The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
evmQpsk(v("in" ?result "tran" ?resultsDir "./vsin_sim.raw"), v("out"
?result "tran" ?resultsDir "./vsin_sim.raw"), 1u, 1.818182e-07, t,
nil, nil , t ?output "rms")
```

Now, when you evaluate this expression, the following scatter plot is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN function for `evmQpsk` is:

```
evmQpsk( o_waveform1 o_waveform2 n_tDelay n_sampling
     g_autoLevelDetect n_voltage n_offset g_normalize )
     => o_waveform/nil
```

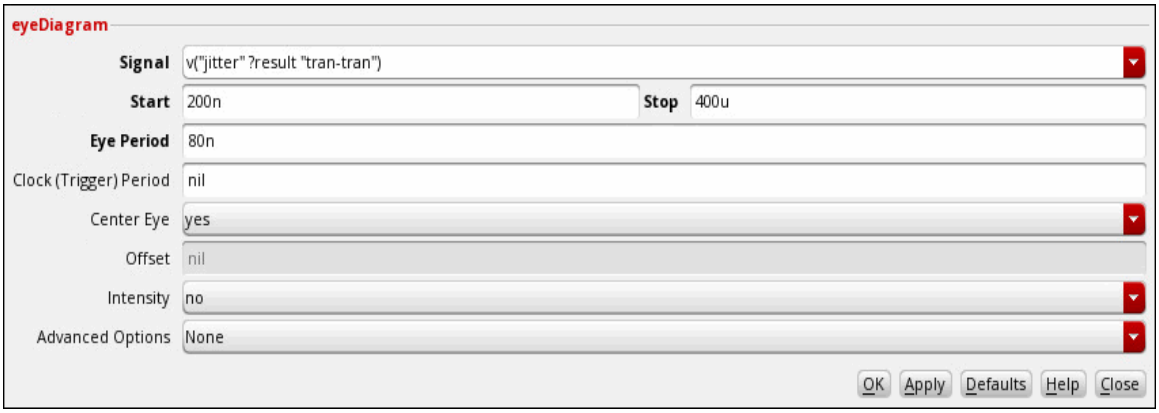
For more information about this OCEAN function, see evmQpsk in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

eyeDiagram

Gives an eye-diagram plot in which the waveform signal is divided into fixed time periods, which are then superimposed on each other. The result is a plot that has many overlapping lines enclosing an empty space known as the eye. The quality of the receiver circuit is characterized by the dimension of the eye. An open eye means that the detector can distinguish between `1's` and `0's` in its input, while a closed eye means that a detector placed on `Vout` is likely to give errors for certain input bit sequences.

This function is available only in the SKILL mode.



This function includes the following arguments:

- *Signal*—Name of the signal to measure.
- *Start*—X-axis value from where the eye-diagram plot is to begin.
- *Stop*—X-axis value where the eye-diagram plot is to end.
- *Eye Period*—Time period for the eye diagram.
- *Clock (Trigger) Period*—Time interval between triggering events.

  Note: If the trigger period is not specified, the eye period is used as the trigger period.
- *Center Eye*—Specifies whether to place the eye diagram in the center of the graph, which means the center of the eye opening is placed at the half of the eye period. Valid values: `yes` or `no`. If you select `yes`, the eye diagram is centered based on the crossing times across the average signal value between the start and the stop time.
- *Offset*—Specifies the offset value that is used to shift the eye. If you specify a positive value, the eye-right is shifted to the right according to the specified value. If you specify a negative value, the eye-left is shifted to left. This field becomes inactive if you select `no` from the *Center Eye* option.

© 2023 Cadence Design Systems, Inc.

- *Intensity*—Specifies whether to highlight the intersection points in the eye diagram Valid values: `yes` or `no`. If you select `no`, the eye digram is plotted with the default color scheme.
- *Advanced Option*s—Specifies whether the vertical (`Max Vertical Opening`) or horizontal opening (`Max Horizontal Opening`) of the eye is to be calculated.

Additional Information

Calculating Horizontal and Vertical Eye Width:

The waveform is folded on the X-axis between the start time (`n_start`) and stop time (`n_stop`) by the length `n_period`.

The function performs the following steps to calculate the horizontal eye opening:

- Calculates all the X points are where the folded waveform intersects the line `y = yMid`

  where, `yMid = ymax(o_waveform)+ min(o_waveform))/2`
- From these calculated X points, returns the two consecutive X points, which have the maximum distance between them, as the horizontal eye opening.

  `X[k]` and `X[k-1]` for which the (`X[k]-X[k-1]`) value is maximum.

The function performs the following steps to calculate vertical eye width:

- Calculates the horizontal eye width to find the consecutive X points, `X[k]` and `X[k-1]` having maximum distance between them.
- Calculates all Y points where the folded waveform intersects the following line:

  ```
  x = (X[k]-X[k-1])/2
  ```
- From these calculated Y points, returns two consecutive Y points, which have the maximum distance between them, as the vertical eye opening.

  `Y[k]` and `Y[k-1]` for which (`Y[k]-Y[k-1]`) is maximum.

Assumptions

The following assumptions have been made while calculating the advance option values:

- The opening of an eye approximately lies on following:

  ```
  (ymax(o_waveform)+ymin(o_waveform))/2
  ```

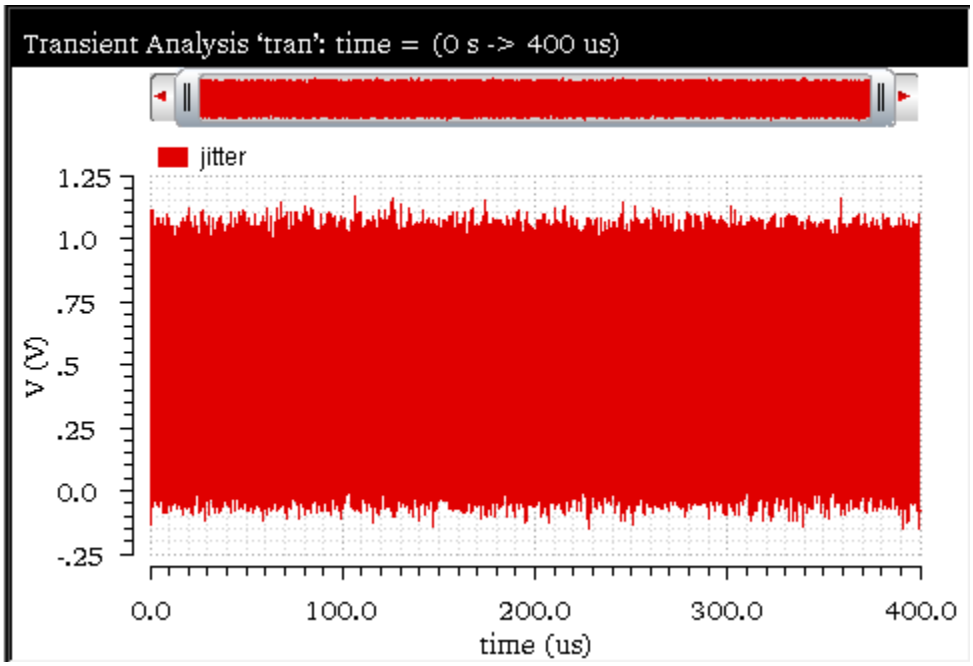

© 2023 Cadence Design Systems, Inc.

- Only one eye opening exists in the area mentioned above in which the waveform is folded.

For more information about eye diagram, see the Eye Diagram assistant.

Example

This example shows the eye diagram plot generated when you apply the `eyeDiagram` function on the following input signal:



The following arguments are specified in this example:

- *Signal*—`v("jitter" ?result "tran-tran")`
- *Start*—`200n`
- *Stop*—`400u`
- *Eye Period*—`80n`
- *Clock (Trigger) Period*—`nil`
- *Center Eye*—`yes`
- *Offset*—`nil`
- *Intensity*—`no`

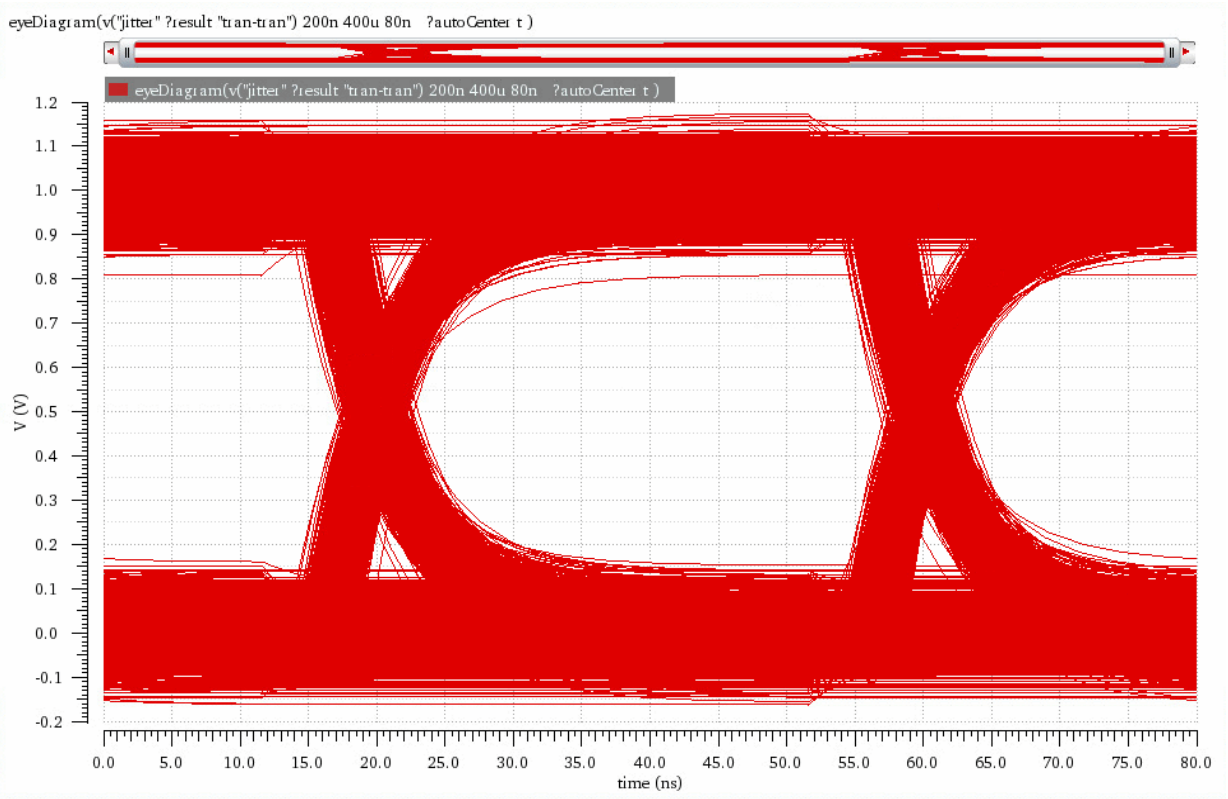
© 2023 Cadence Design Systems, Inc.

- *Advanced Options*—`None`

The following expression is created in the Buffer:

```
eyeDiagram(v("jitter" ?result "tran-tran") 200n 400u 80n   )
```

Now, when you evaluate this expression, the following eye diagram plot is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `eyeDiagram` is:

```
eyeDiagram ( o_waveform n_start n_stop n_period ?advOptions t_advOptions )
     => o_waveform/nil
```

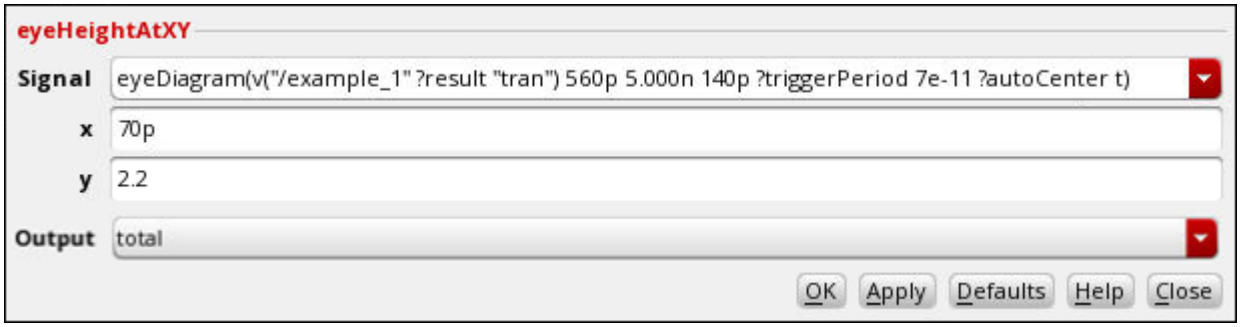
For more information about this OCEAN function, see eyeDiagram in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

eyeHeightAtXY

Calculates the height of an eye at the specified point (x,y) inside the eye diagram.

This function is available only in the SKILL mode.



This function includes the following arguments:

- *Signal*—The eye diagram waveform that is used to calculate the eye height.
- *x*—The x-axis value that is used to calculate the eye height.
- *y*—The y-axis value that is used to calculate the eye height.

  Note: The specified coordinates (x,y) must lie within the open eye whose height you want to calculate.
- *Output*—Specifies whether you want to calculate the absolute eye height or the eye height relative to the specified y-axis value. Possible values are:
  - *total*: (Default) Calculates the total eye height.
  - *above*: Calculates the eye height that is above the specified y-axis value.
  - *below*: Calculates the eye height that is below the specified y-axis value.

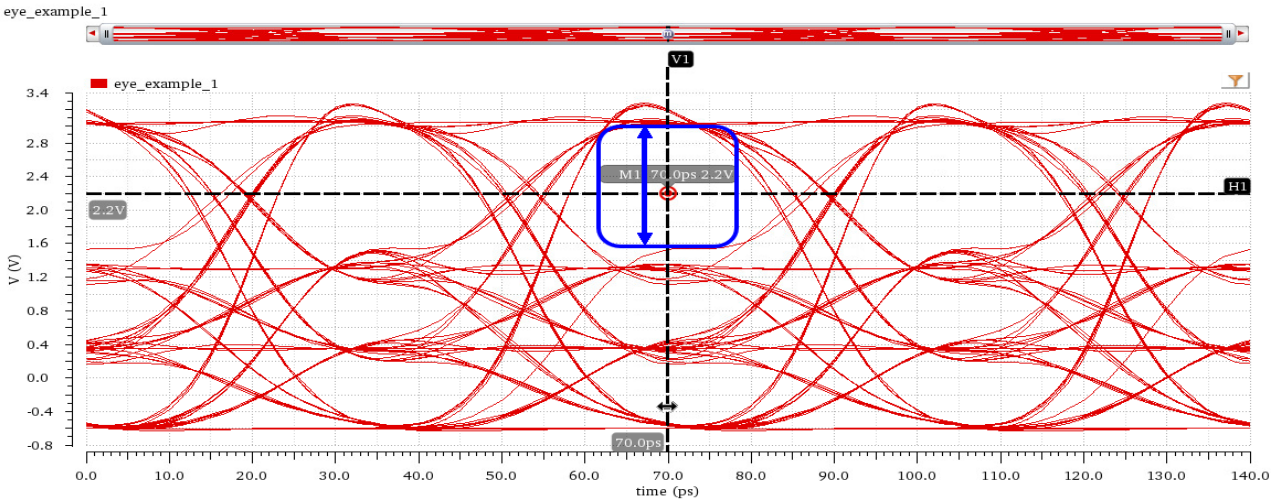
Video

You can watch a video demonstration on how to use this function at Enhancements in eyeHeightAtXY and eyeWidthAtXY Functions. Also, read the related blog at Virtuoso Video Diary: Do More With eyeHeightAtXY and eyeWidthAtXY Calculator Functions in Virtuoso Visualization and Analysis XL.

© 2023 Cadence Design Systems, Inc.

Example 1

This example shows the height calculation of the highlighted eye at point `M1` (`70ps`, `2.2v`). Note that the point, `M1`, lies within the eye whose height is to be calculated.

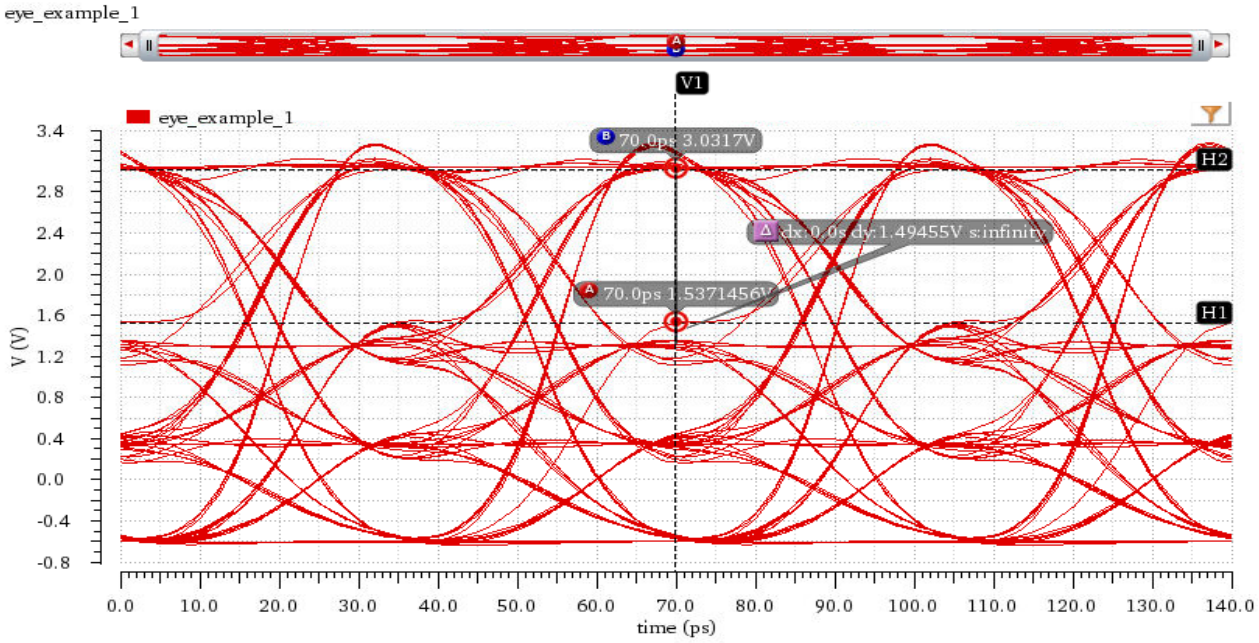


Eye height is the difference of two intercepts made with the innermost traces of the eye in the Y-axis direction. You can calculate the eye height at `70ps` and `2.2v` with the help of an AB marker.

Create a vertical marker at `70ps` and then insert the points, `A` and `B`, where the vertical marker intercepts the innermost traces of the eye. The delta of the AB marker (`dy`) indicates

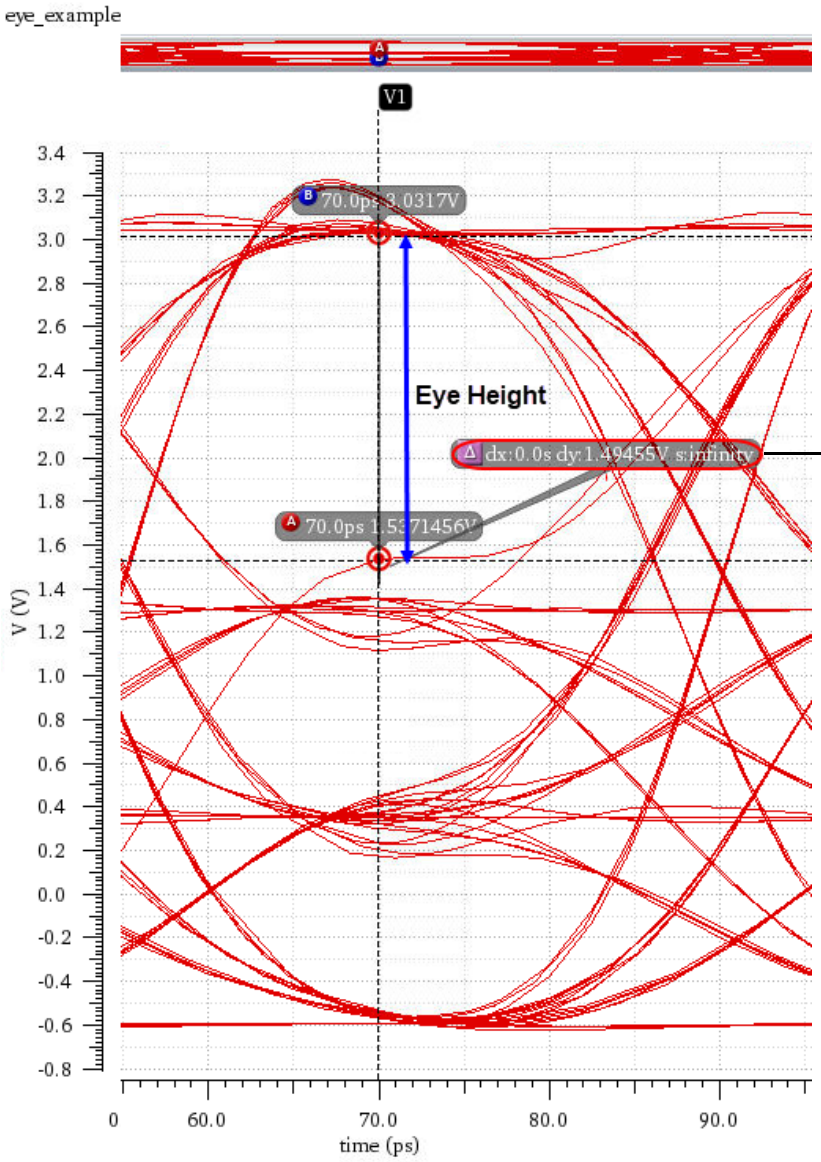
© 2023 Cadence Design Systems, Inc.

the difference of the Y-axis values of points `B` (`3.0317v`) and `A` (`1.537145v`), which is equal to `1.49455v`. This is the eye height, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

The following snippet shows a magnified version of the information displayed in the previous figure.



The Delta (`dy`) value shows the eye height.
Eye height= `1.4945`

Now, calculate the total eye height using the `eyeHeightAtXY` function. Specify the following arguments:

- *Signal*—`eyeDiagram(v("/example_1" ?result "tran") 560p 5.000n 140p ?triggerPeriod 7e-11 ?autoCenter t)`
- *x*—`70p`
- *y*—`2.2`

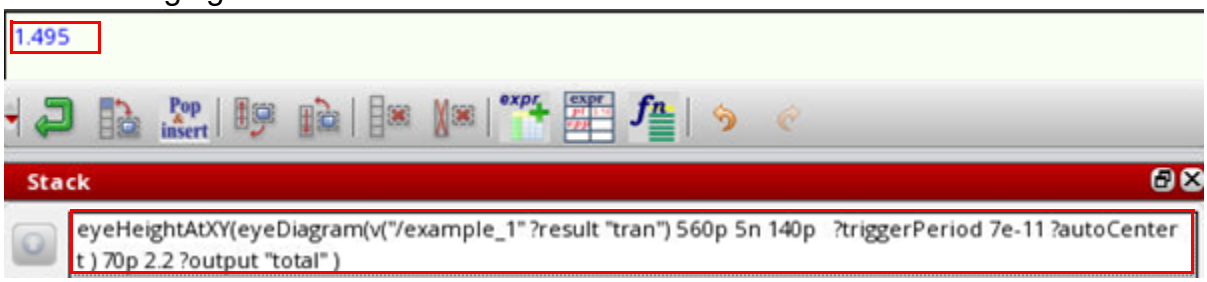
© 2023 Cadence Design Systems, Inc.

- *Output*—`total`

The following expression is created in the Buffer:

```
eyeHeightAtXY(eyeDiagram(v("/example_1" ?result "tran") 560p 5.000n
140p ?triggerPeriod 7e-11 ?autoCenter t) 70p 2.2 ?output "total" )
```

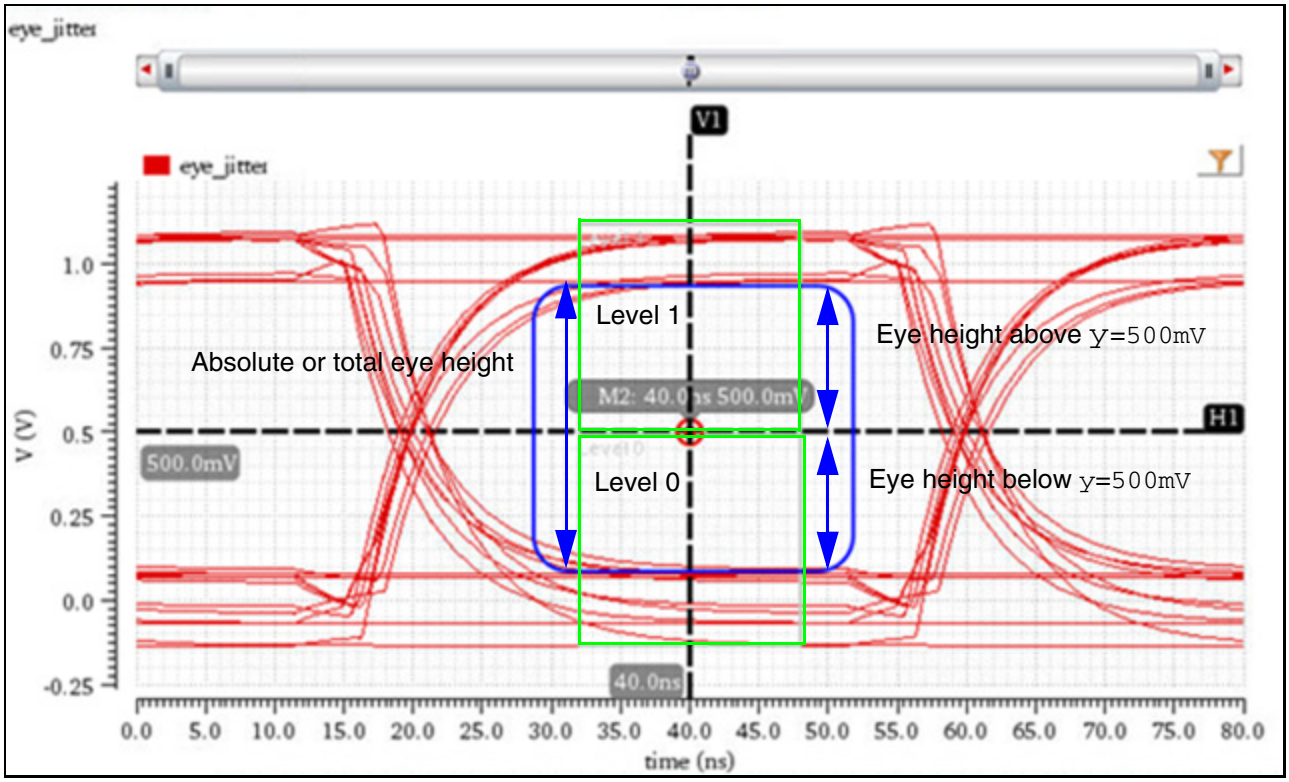
When you evaluate this expression, the function returns the value of eye height, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

Example 2

This example shows the height calculation of the highlighted eye at the point `M2` (`40ns`, `500mV`). Note that the point `M2` lies within the eye whose height is to be calculated.



$$eyeHeightAtXY_{total} = eyeHeightAtXY_{above} + eyeHeightAtXY_{below}$$

$eyeHeightAtXY_{below}$ = Eye height in the level 0 region

$eyeHeightAtXY_{above}$ = Eye height in the level 1 region

Now, we will calculate the total eye height at `M2`, eye height above `y=500mV`, and eye height below `y=500mV` by using the `eyeHeightAtXY` function.

To calculate the total eye height at `M2` (`x=40ns`, `y=500mV`), specify the following arguments:

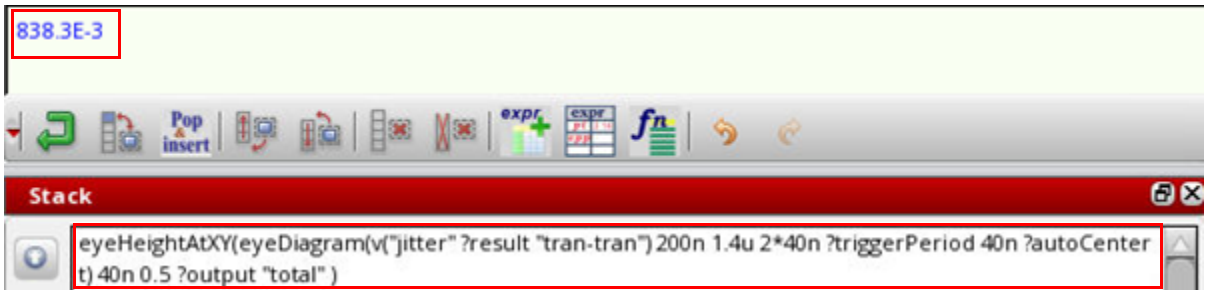
© 2023 Cadence Design Systems, Inc.

- *Signal*—`eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t)`
- *x*—`40n`
- *y*—`0.5`
- *Output*—`total`

The following expression is created in the Buffer:

```
eyeHeightAtXY(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
     ?triggerPeriod 40n ?autoCenter t) 40n 0.5 ?output "total" )
```

When you evaluate this expression, the function returns the value of total eye height, as shown in the following figure.



To calculate the eye height at `M2` (`x=40ns`, `y=500mV`) below `y=500mV` (eye height in the region Level 0), specify the following arguments:

- *Signal*—`eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t)`
- *x*—`40n`
- *y*—`0.5`
- *Output*—`below`

The following expression is created in the Buffer:

```
eyeHeightAtXY(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
     ?triggerPeriod 40n ?autoCenter t) 40n 0.5 ?output "below" )
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the value of eye height that is below `y=0.5V` (`500mV`), as shown in the following figure.

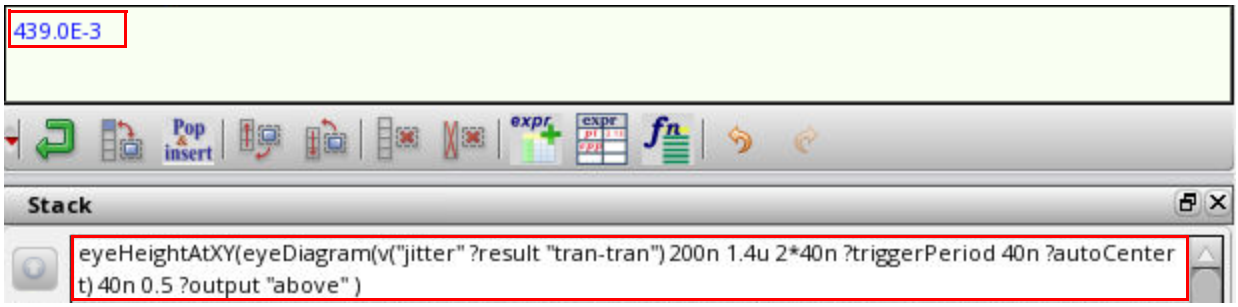
To calculate the eye height at `M2` (`x=40ns`, `y=500mV`) above `y=500mV` (eye height in the region Level 1), specify the following arguments:

- *Signal*—`eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t)`
- *x*—`40n`
- *y*—`0.5`
- *Output*—`above`

The following expression is created in the Buffer:

```
eyeHeightAtXY(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
     ?triggerPeriod 40n ?autoCenter t) 40n 0.5 ?output "above" )
```

When you evaluate this expression, the function returns the value of eye height that is above `y=0.5V` (`500mV`), as shown in the following figure.



Observe that the total eye height at `M2` is the sum of eye heights in the level 0 and level 1 regions.

© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN command for `eyeHeightAtXY` is:

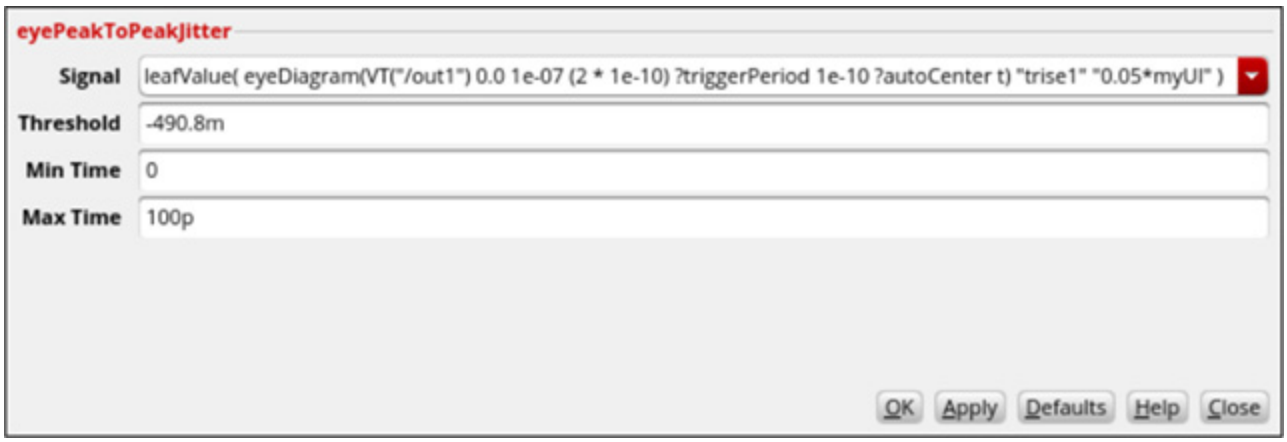
```
eyeHeightAtXY ( o_eyeDiagram f_x f_y ?output t_output)
     => f_eyeHeight/nil
```

For more information about this OCEAN function, see eyeHeightAtXY in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

eyePeakToPeakJitter

Calculates the peak-to-peak jitter at the specified threshold within the region of an eye diagram. The peak-to-peak jitter is the time between the first and the last crossing.



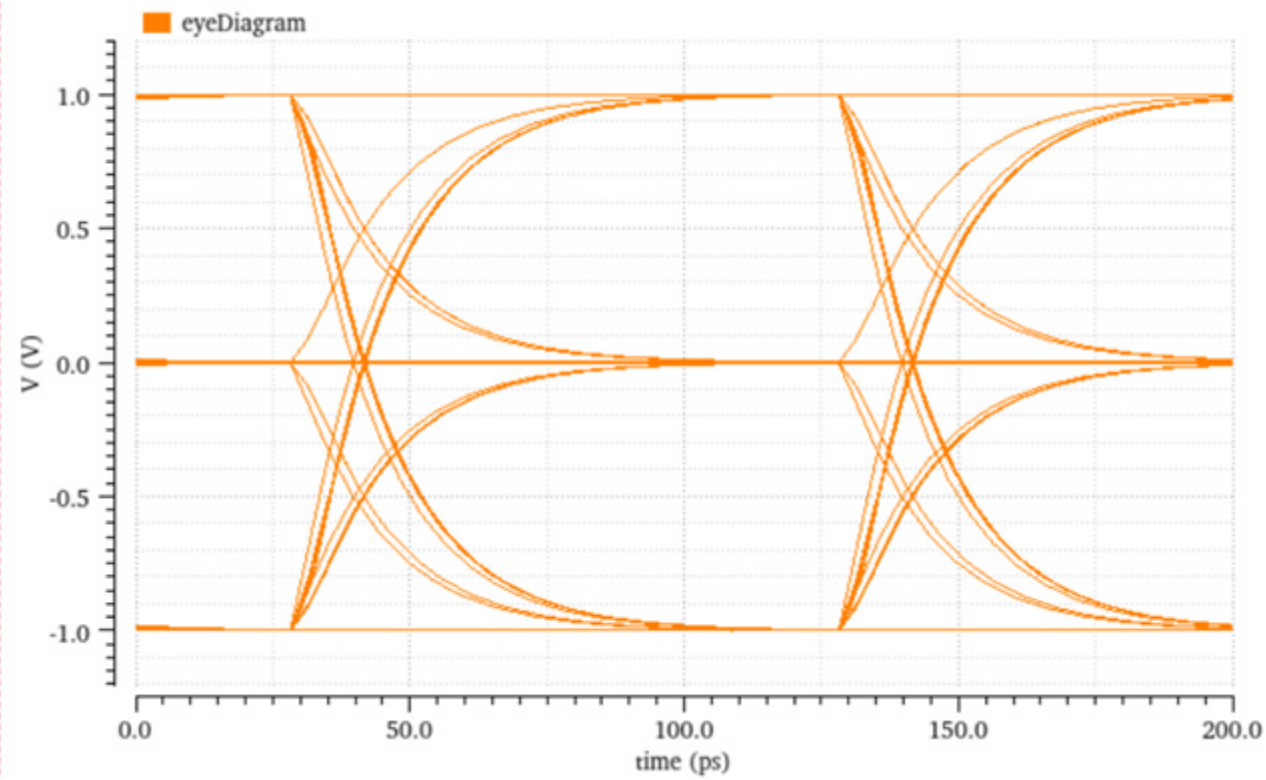
This function includes the following arguments:

- *Signal*—The eye diagram waveform for which the peak-to-peak jitter is to be calculated.
- *Threshold*—Threshold value at which the peak-to-peak jitter is to be calculated.
- *Min Time*—Start time of the crossing.
- *Max Time*—End time of the crossing.

© 2023 Cadence Design Systems, Inc.

Examples

The following figure shows the input waveform of the eye diagram for which peak-to-peak jitter is to be calculated.



To calculate the peak-to-peak jitter the threshold value `-490.8mV` of this eye diagram using the calculator function `eyePeakToPeakJitter`, specify the following arguments:

- *Signal*: `leafValue( eyeDiagram(VT("/out1") 0.0 1e-07 (2 * 1e-10) ?triggerPeriod 1e-10 ?autoCenter t) "trise1" "0.05*myUI" )`
- *Threshold*: `-490.8m`
- *Min Time*: `0`
- *Max Time*: `100p`

The following expression is created in the Buffer:

```
eyePeakToPeakJitter(leafValue( eyeDiagram(VT("/out1") 0.0 1e-07 (2 * 1e-10)
?triggerPeriod 1e-10 ?autoCenter t) "trise1" "0.05*myUI" ) -490.8m 0 100p )
```

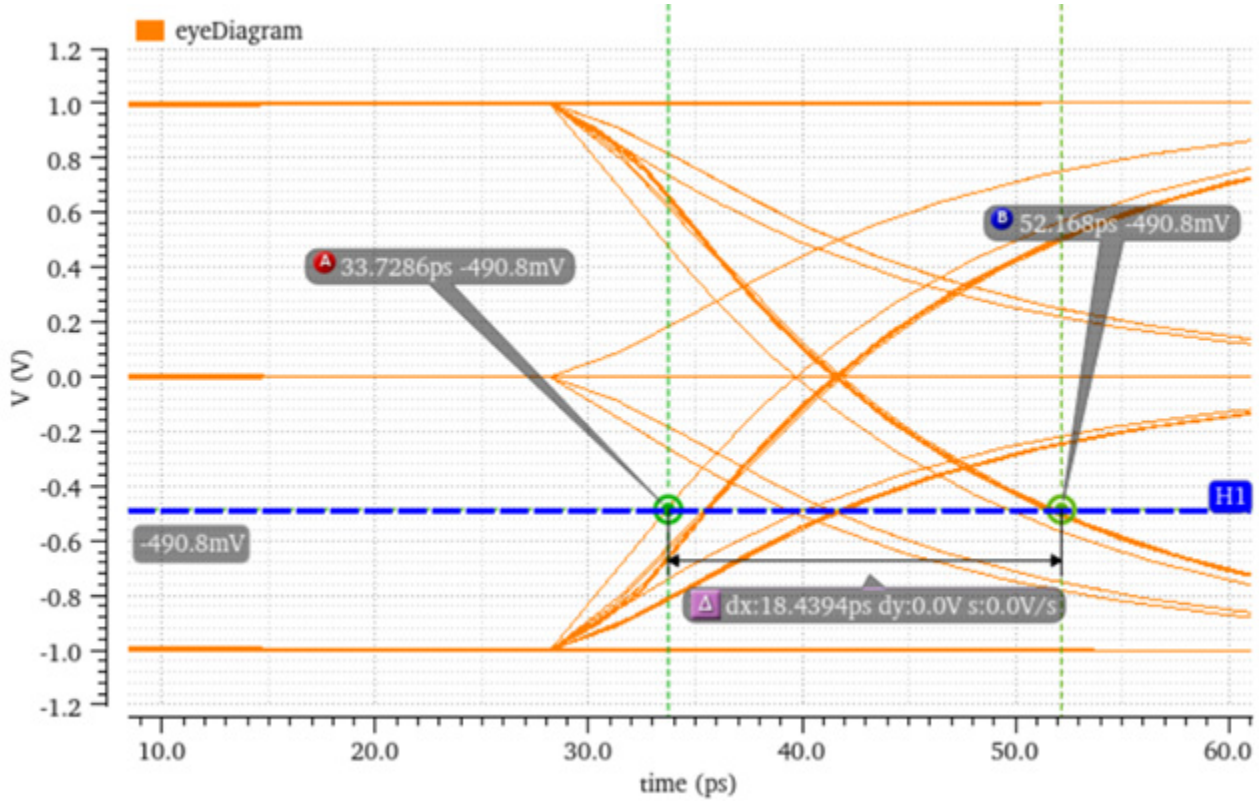
When you evaluate this expression, the function returns the value of the peak-to-peak jitter at the threshold value `-490.8mV` of the eye diagram. The peak-to-peak jitter is the distance

© 2023 Cadence Design Systems, Inc.

between the first (`0p`) and last crossing (`100p`)at the threshold value `-490.8mV` of the eye diagram.

```
=> 18.44E-12
```

The calculated value of peak-to-peak jitter can be verified by placing the `A` and `B` markers on the first and last crossing at the threshold value of `-490.8mV`. The distance between these two markers is `18.4ps`.



Related Topics

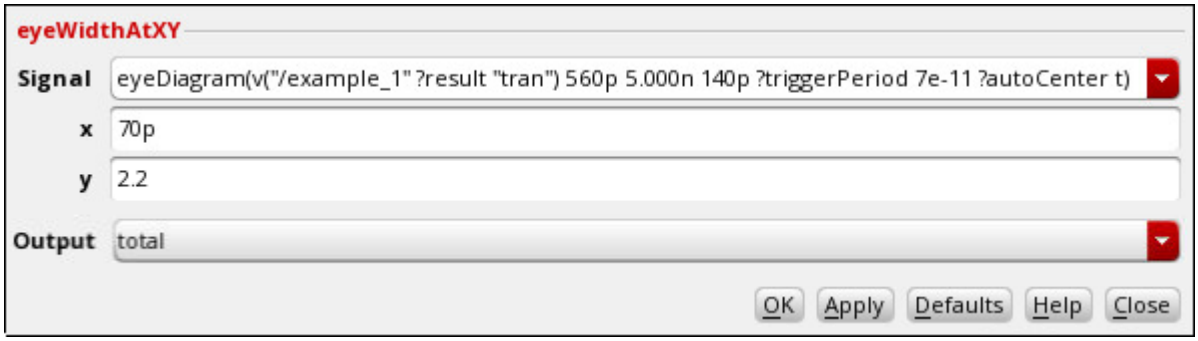
eyePeakToPeakJitter

© 2023 Cadence Design Systems, Inc.

eyeWidthAtXY

Calculates the width of an eye at the specified point (x,y) inside the eye diagram.

This function is available only in the SKILL mode.



This function includes the following arguments:

- *Signal*—The eye diagram waveform that is used to calculate the eye width.
- *x*—The x-axis value that is used to calculate the eye width.
- *y*—The y-axis value that is used to calculate the eye width.

  Note: The specified coordinates (x,y) must lie within the open eye whose width you want to calculate.
- *Output*—Specifies whether you want to calculate the total eye width or the eye width relative to the specified x-axis value. Possible values are:
  - *total*: (Default) Calculates the total eye width.
  - *left*: Calculates the eye width that is left to the specified x-axis value.
  - *right*: Calculates the eye width that is right to the specified x-axis value.

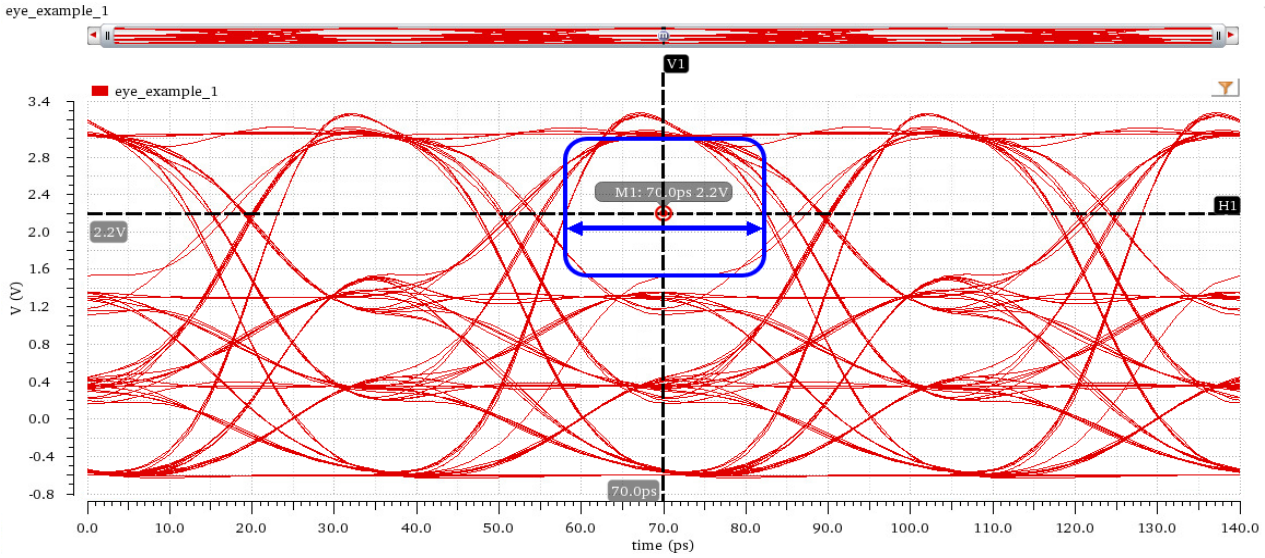
Video

You can watch a video demonstration on how to use this function at Enhancements in eyeHeightAtXY and eyeWidthAtXY Functions. Also, read the related blog at Virtuoso Video Diary: Do More With eyeHeightAtXY and eyeWidthAtXY Calculator Functions in Virtuoso Visualization and Analysis XL.

© 2023 Cadence Design Systems, Inc.

Example 1

This example shows the width calculation of the highlighted eye at point `M1(70ps, 2.2v)`. Note that the point `M1` lies within the eye whose width is to be calculated.

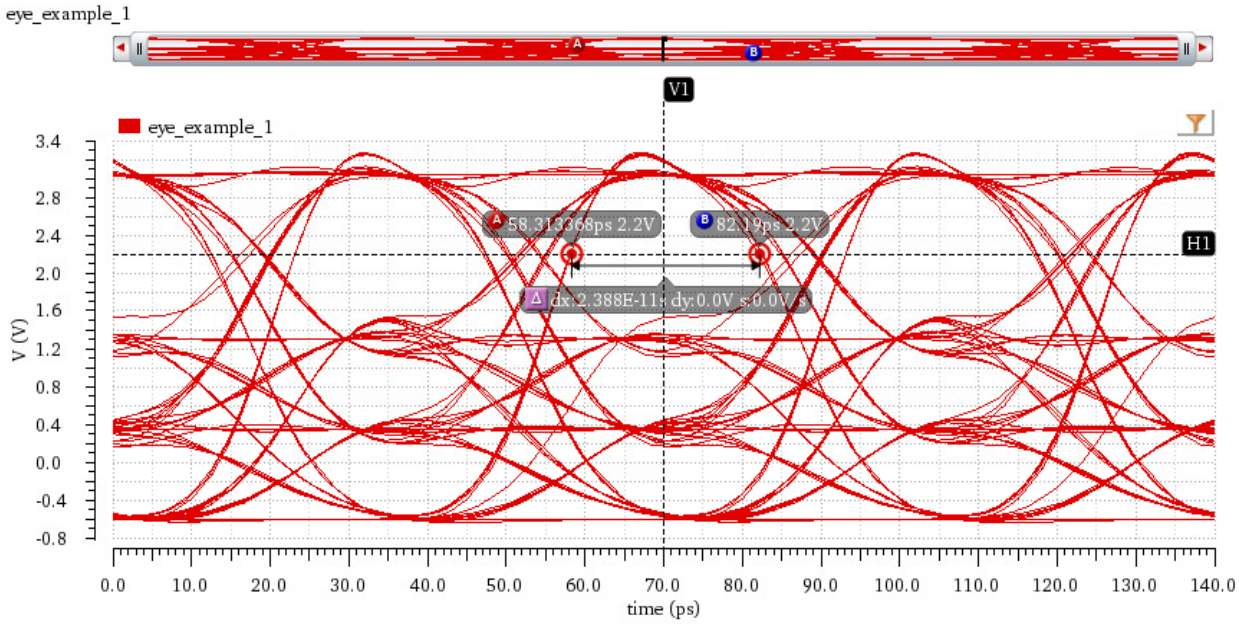


Eye width is the difference of two intercepts made with the innermost traces of the eye in the X-axis direction. You can calculate the eye width at `70ps` and `2.2V` with the help of an AB marker.

Create a horizontal marker at `2.2V` and then insert the points, `A` and `B`, where the horizontal marker intercepts the innermost traces of the eye. The delta of the AB marker (`dx`) indicates

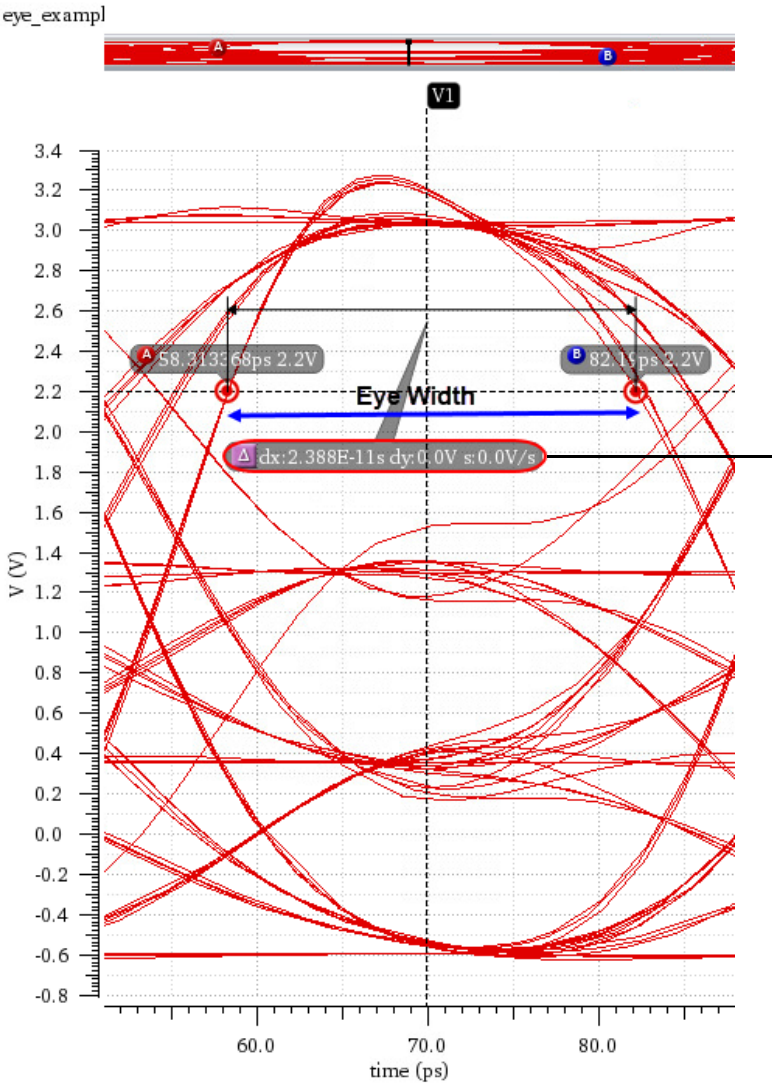
© 2023 Cadence Design Systems, Inc.

the difference of the X-axis values of points `B` (`82.19ps`) and `A` (`58.31ps`), which is equal to `23.88ps` or `2.388e-11` second. This is the eye width, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

The following snippet shows a magnified version of information displayed in the previous figure.



The Delta (`dx`) value shows the eye width.
Eye width= `2.388e-11`

Now, calculate the total eye width using the `eyeWidthAtXY` function. Specify the following arguments:

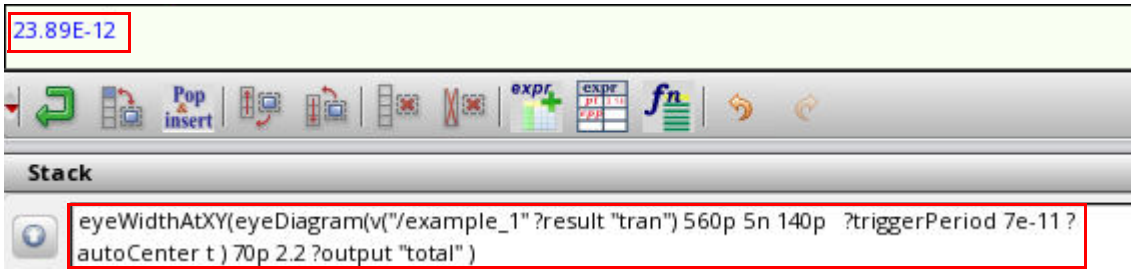
- *Signal*—`eyeDiagram(v("/example_1" ?result "tran") 560p 5.000n 140p ?triggerPeriod 7e-11 ?autoCenter t)`
- *x*—`70p`
- *y*—`2.2`
- *Output*—`total`

© 2023 Cadence Design Systems, Inc.

The following expression is created in the Buffer:

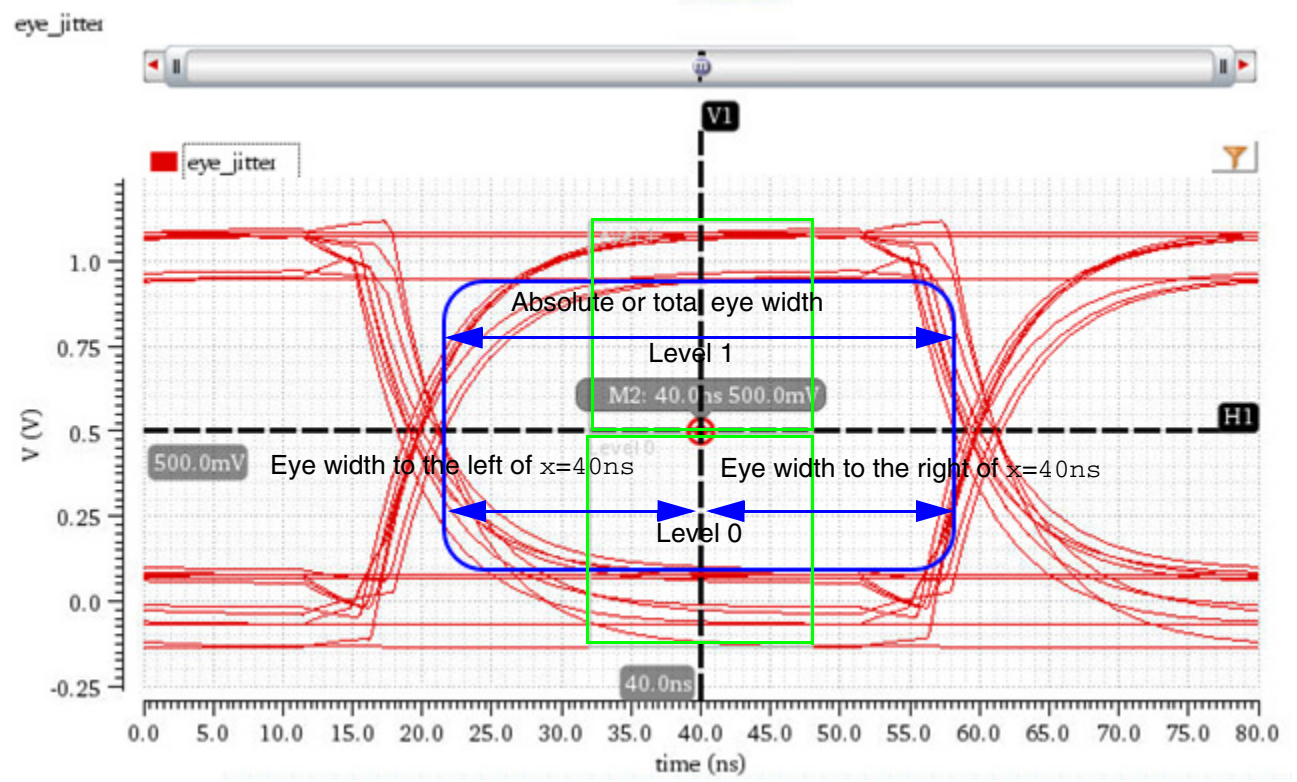
```
eyeWidthAtXY(eyeDiagram(v("/example_1" ?result "tran") 560p 5.000n
140p ?triggerPeriod 7e-11 ?autoCenter t) 70p 2.2 ?output "total")
```

When you evaluate this expression, the function returns the value of eye width, as shown in the following figure.



Example 2

This example shows the width calculation of the highlighted eye at point `M2` (`40ns`, `500mV`). Note that the point, `M2`, lies within the eye whose width is to be calculated.



© 2023 Cadence Design Systems, Inc.

$$eyeWidthAtXY_{total} = eyeWidthAtXY_{left} + eyeWidthAtXY_{right}$$

Now, calculate the total eye width at `M2`, eye width to the left of `x=40ns`, and eye width to the right of `x=40ns` by using the `eyeWidthAtXY` function.

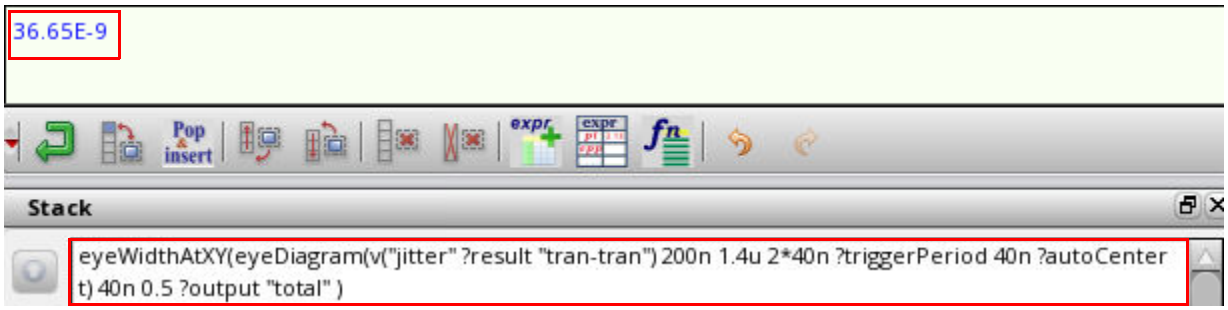
To calculate the total eye width at `M2` (`x=40ns`, `y=500mV`), specify the following arguments:

- *Signal*—`eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t)`
- *x*—`40n`
- *y*—`0.5`
- *Output*—`total`

The following expression is created in the Buffer:

```
eyeWidthAtXY(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
     ?triggerPeriod 40n ?autoCenter t) 40n 0.5 ?output "total" )
```

When you evaluate this expression, the function returns the value of total eye width, as shown in the following figure.



To calculate the eye width at `M2` (`x=40ns`, `y=500mV`) to the left of `x=40ns`, specify the following arguments:

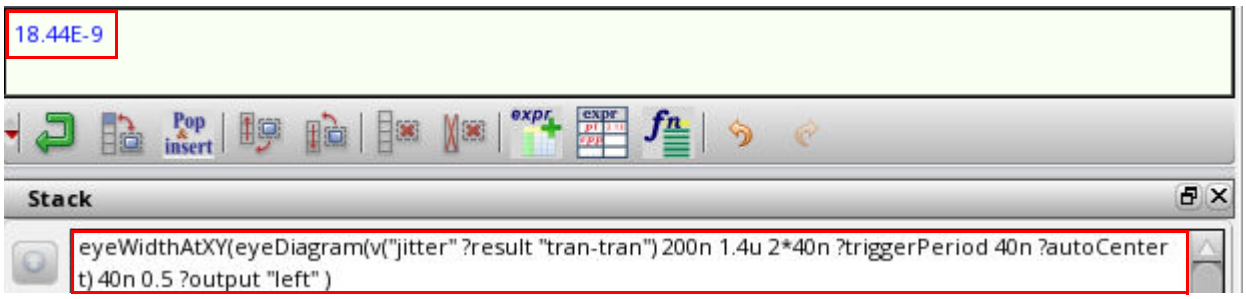
- *Signal*—`eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t)`
- *x*—`40n`
- *y*—`0.5`
- *Output*—`left`

© 2023 Cadence Design Systems, Inc.

The following expression is created in the Buffer:

```
eyeWidthAtXY(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
     ?triggerPeriod 40n ?autoCenter t) 40n 0.5 ?output "left" )
```

When you evaluate this expression, the function returns the value of eye width that is left to `x=40ns`, as shown in the following figure.



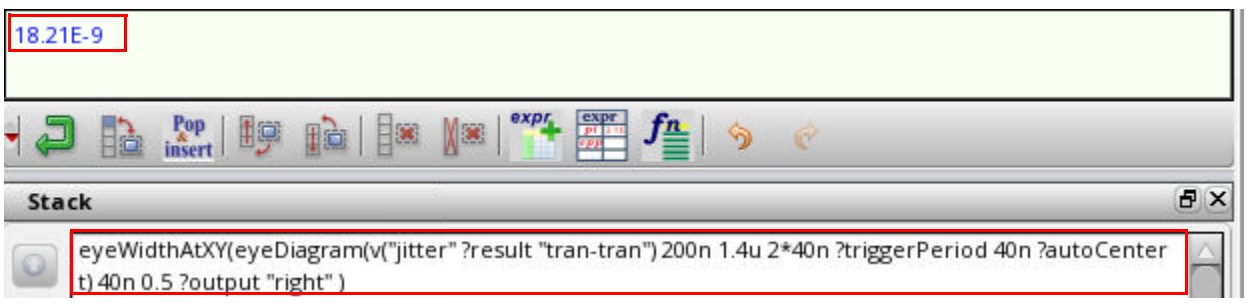
To calculate the eye width at `M2` (`x=40ns`, `y=500mV`) to the right of `x=40ns`, specify the following arguments:

- *Signal*—`eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t)`
- *x*—`40n`
- *y*—`0.5`
- *Output*—`right`

The following expression is created in the Buffer:

```
eyeWidthAtXY(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
     ?triggerPeriod 40n ?autoCenter t) 40n 0.5 ?output "right" )
```

When you evaluate this expression, the function returns the value of eye width that is right to `x=40ns`, as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

Observe that the total eye width at `M2` is the sum of eye widths to the left and to the right of the specified x value.

Related OCEAN Function

The equivalent OCEAN command for `eyeWidthAtXY` is:

```
eyeWidthAtXY ( o_eyeDiagram f_x f_y ?output t_output)
     => f_eyeWidth/nil
```

For more information about this OCEAN function, see eyeWidthAtXY in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

eyeAperture

Returns the aperture of the input eye diagram signal.

This function includes the following fields:

- *Signal*—Input signal, which is an eye diagram waveform for which the eye aperture is to be calculated.
- *Vref*—Reference voltage value.
- *AC Height*—AC height, which specifies the height of the left side of the aperture window.
- *DC Height*—DC height, which specifies the height of the right side of the aperture window.
- *Display Aperture*—Specifies whether to display the aperture in the eye diagram or calculate the eye width.
  Valid values: `yes` or `no`. When this argument is set to `yes`, the eye aperture is displayed in the output plot. When set to `no`, the eye aperture width is returned.
  Default value: `no`.
- *Optimize*—Specifies whether to calculate the reference voltage that can be used to achieve the maximum eye aperture width.
  Valid values: `yes` or `no`.
  Default value: `no`.

Note the following points:

The `eyeAperture` function results in an evaluation error in the following cases:

- For open eye diagram
- Eye period is not set as `2UI`

© 2023 Cadence Design Systems, Inc.

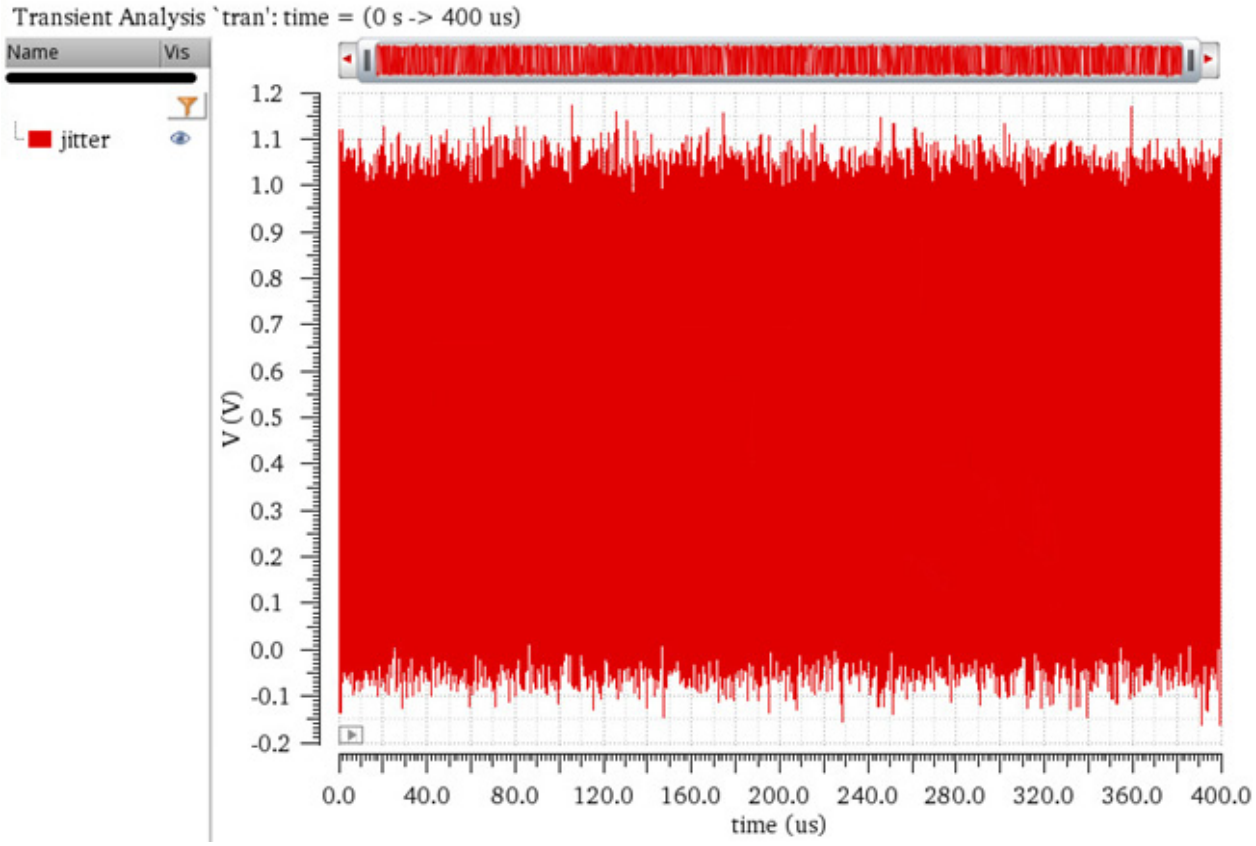
- Value of the AC or DC height exceeds the height of the eye
- Value specified for `Vref` is incorrect

Examples

Part1: Calculating eye aperture on a single eye diagram waveform

Consider the following input signal from the transient analysis:

```
v("jitter" ?result "tran" ?resultsDir "./ViVA_Workshop_IC618/design/prbs.raw")
```



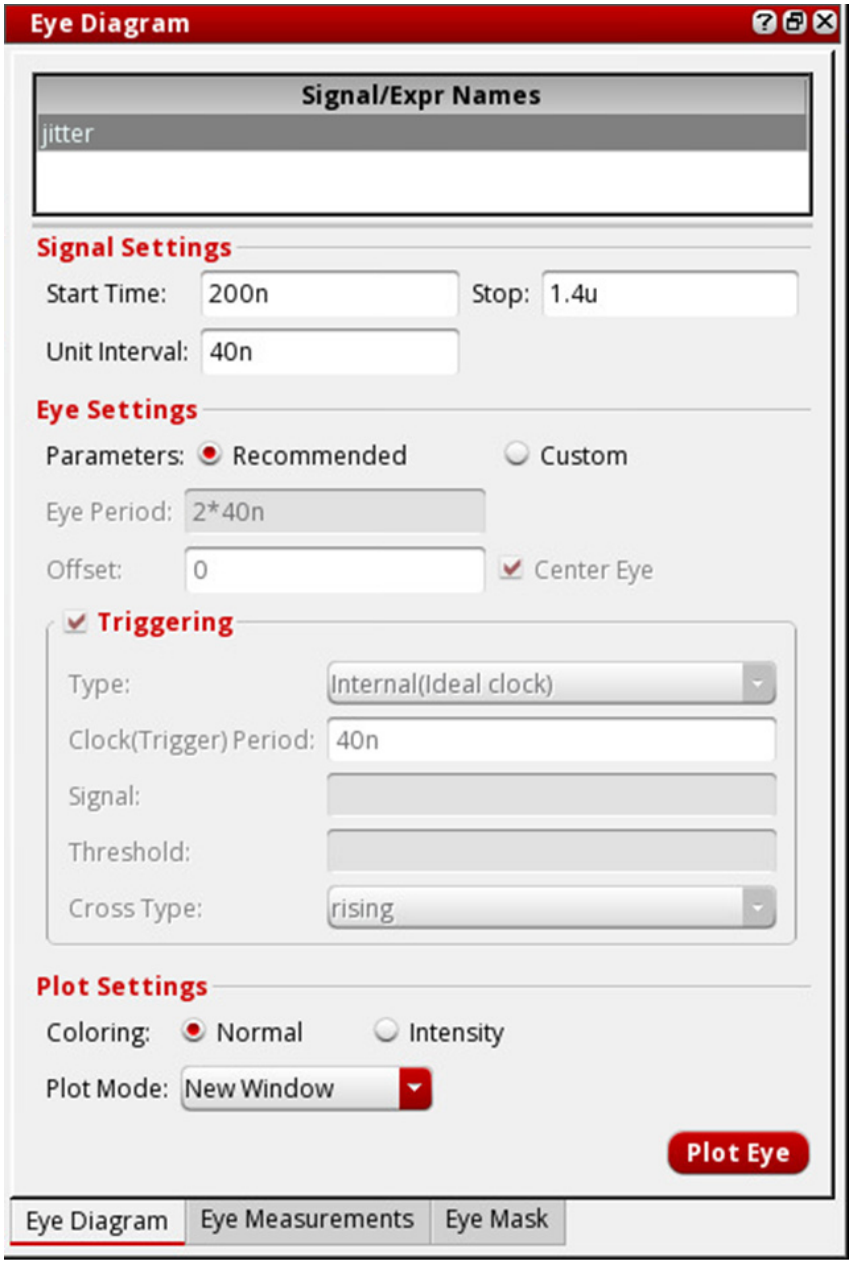
Note: The `eyeAperture` function requires an eye diagram waveform as an input; therefore, firstly, you need to plot an `eyeDiagram` from this input signal. To plot the eye diagram, specify the following field values in the *Eye Diagram* assistant:

- *Start Time*—`200n`
- *Stop*—`1.4u`
- *Unit Interval*—`40n`

© 2023 Cadence Design Systems, Inc.

- *Eye Period*—`2*40n` (2UI)

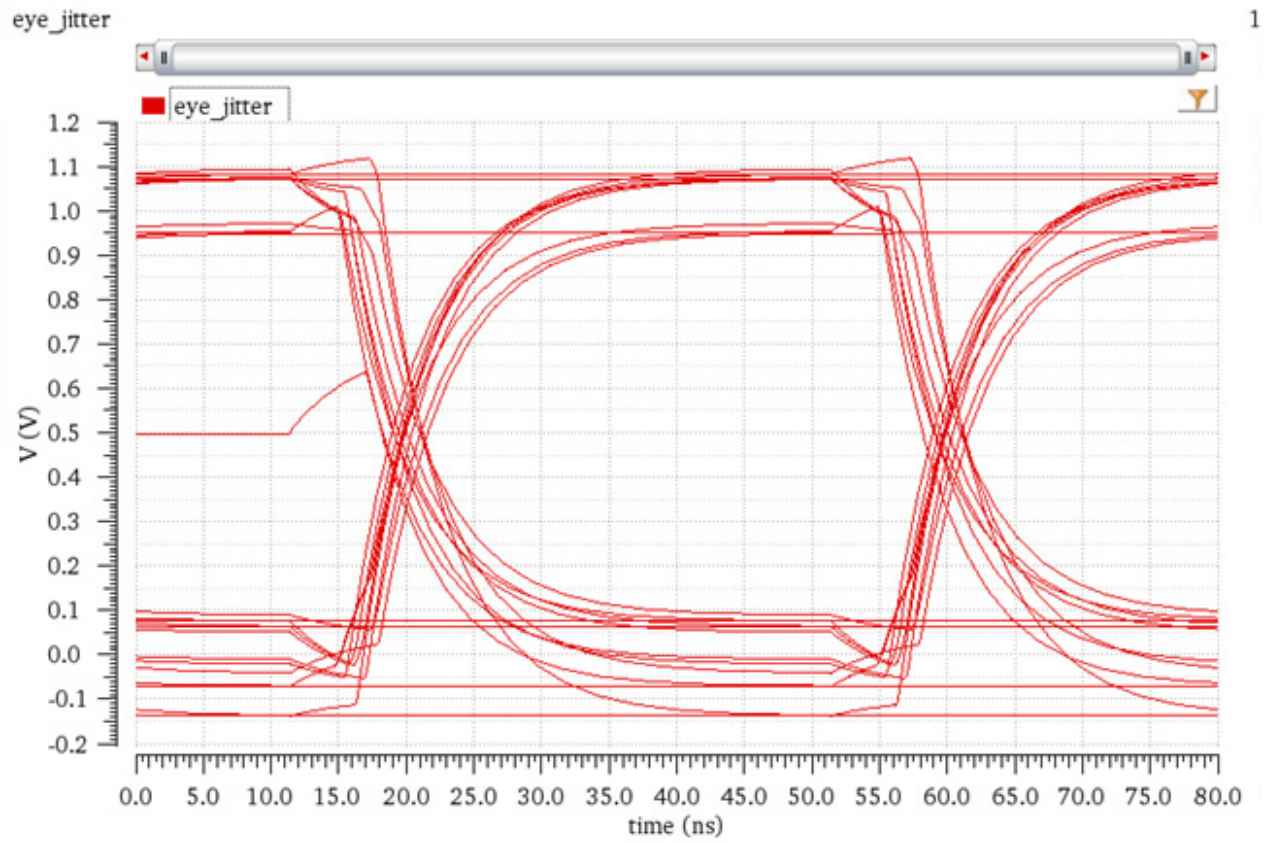
The following figure shows these settings configured in the *Eye Diagram* assistant.



Now, click the *Plot Eye* button. For more information about how to use the *Eye Diagram* assistant, see Eye Diagram Assistant.

© 2023 Cadence Design Systems, Inc.

The following eye diagram is plotted in a new window.



Now, when you send this eye diagram to Calculator, the following expression is displayed in the Buffer:

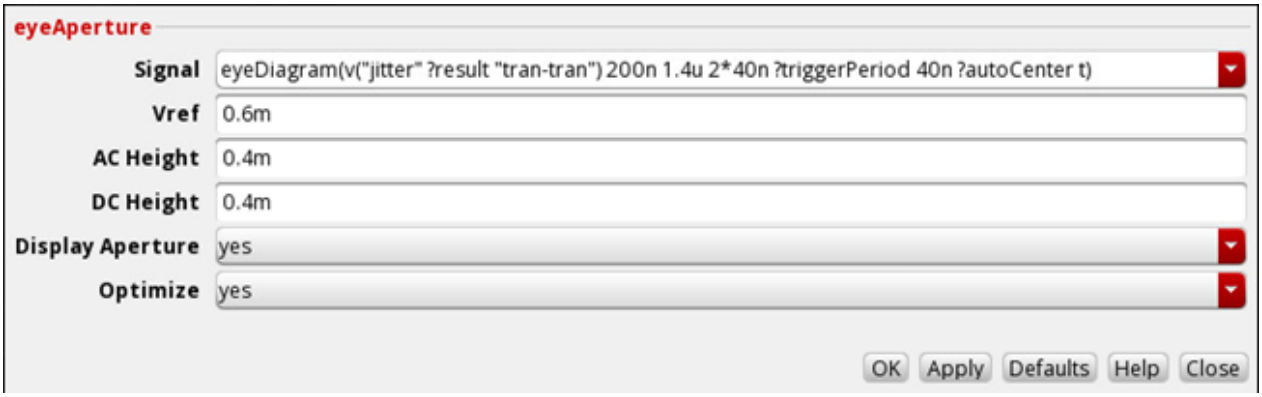
```
eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
?triggerPeriod 40n ?autoCenter t)
```

Now, apply the `eyeAperture` function on this Buffer expression with the following arguments:

- *Signal*—`eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t)`
- *Vref*—`0.6m`
- *AC Height*—`0.4m`
- *DC Height*—`0.4m`
- *Display Aperture*—`yes`

© 2023 Cadence Design Systems, Inc.

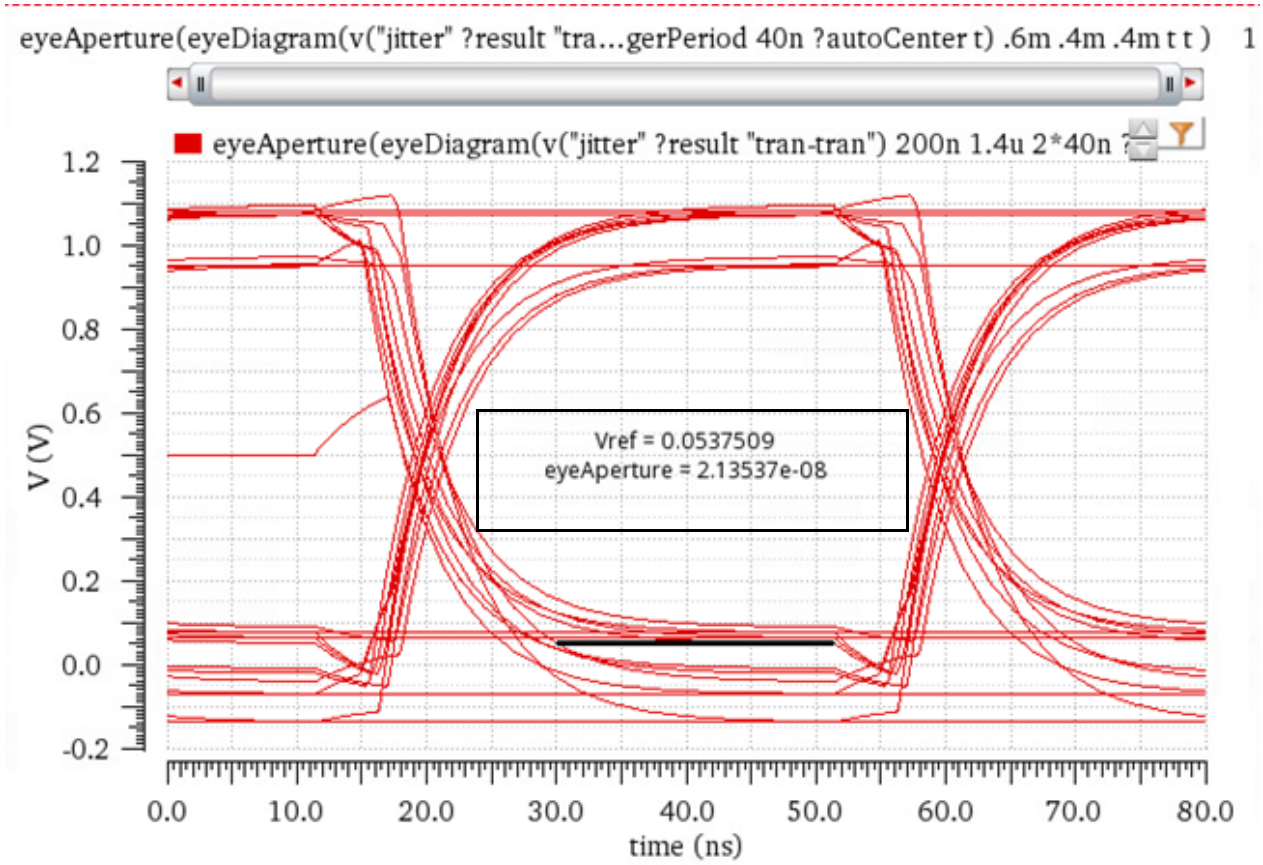
- *Optimize*—`yes`



After you apply the function, the following new expression is created in the Buffer:

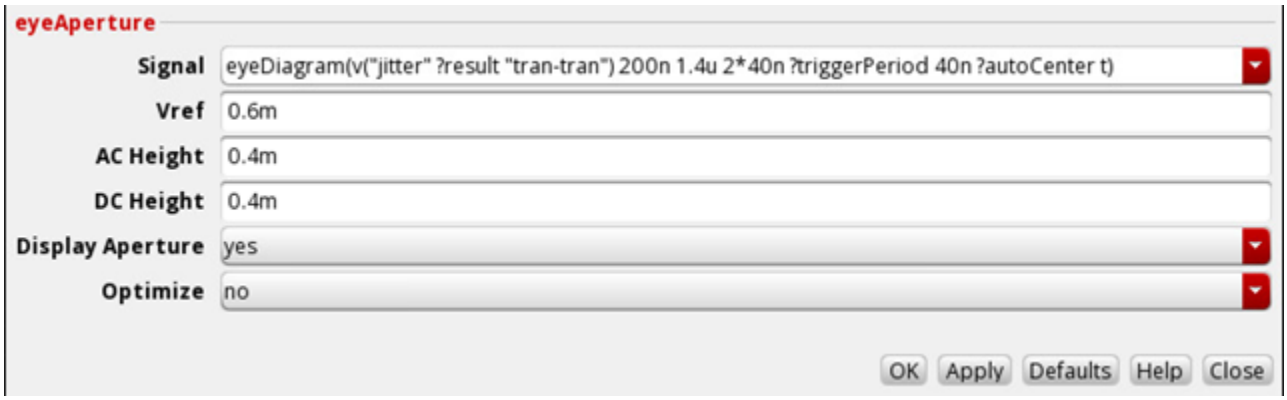
```
eyeAperture(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u
2*40n ?triggerPeriod 40n ?autoCenter t) 0.6m 0.4m 0.4m t t )
```

When you evaluate this expression, the following output is displayed in a new window:



© 2023 Cadence Design Systems, Inc.

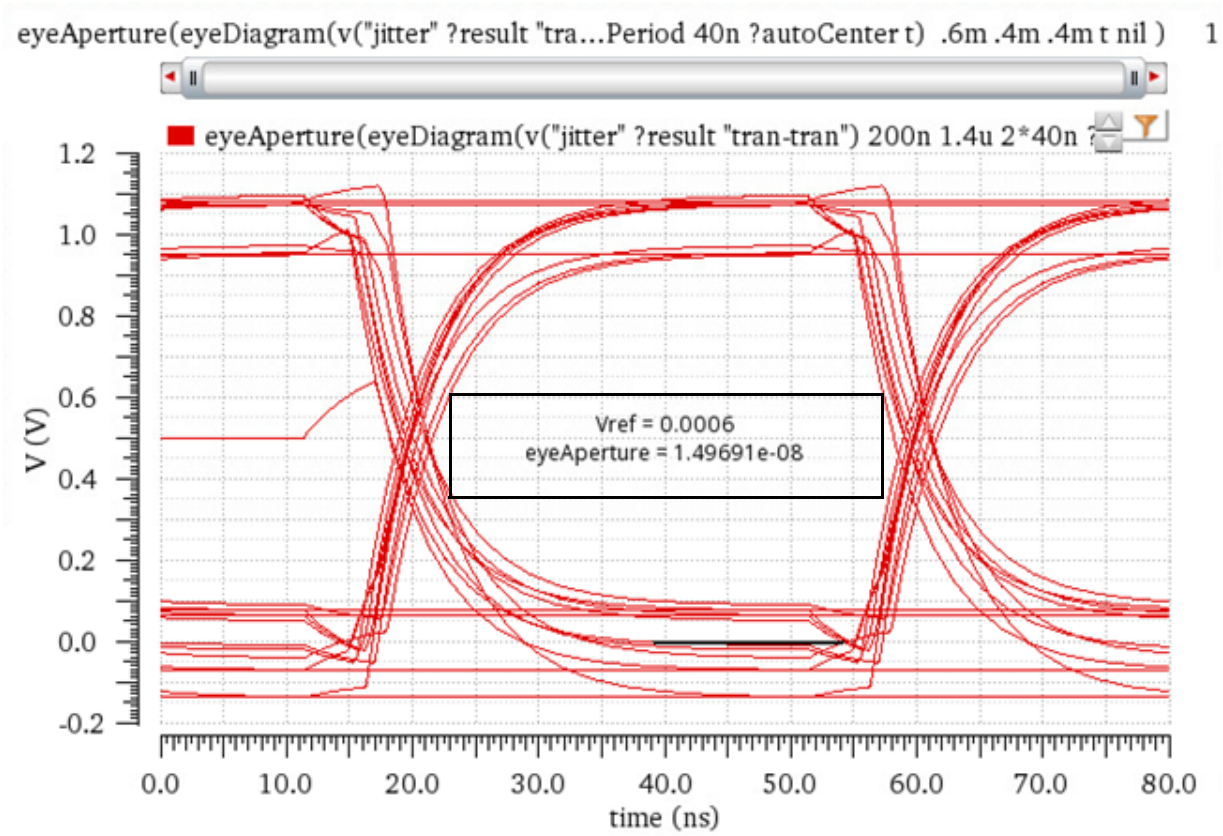
Now, apply the `eyeAperture` function again with the *Optimize* value as `no`.



The following expression is created in the Buffer:

```
eyeAperture(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u
2*40n ?triggerPeriod 40n ?autoCenter t)  0.6m 0.4m 0.4m t nil )
```

When you evaluate this expression, the following output appears in a new window:



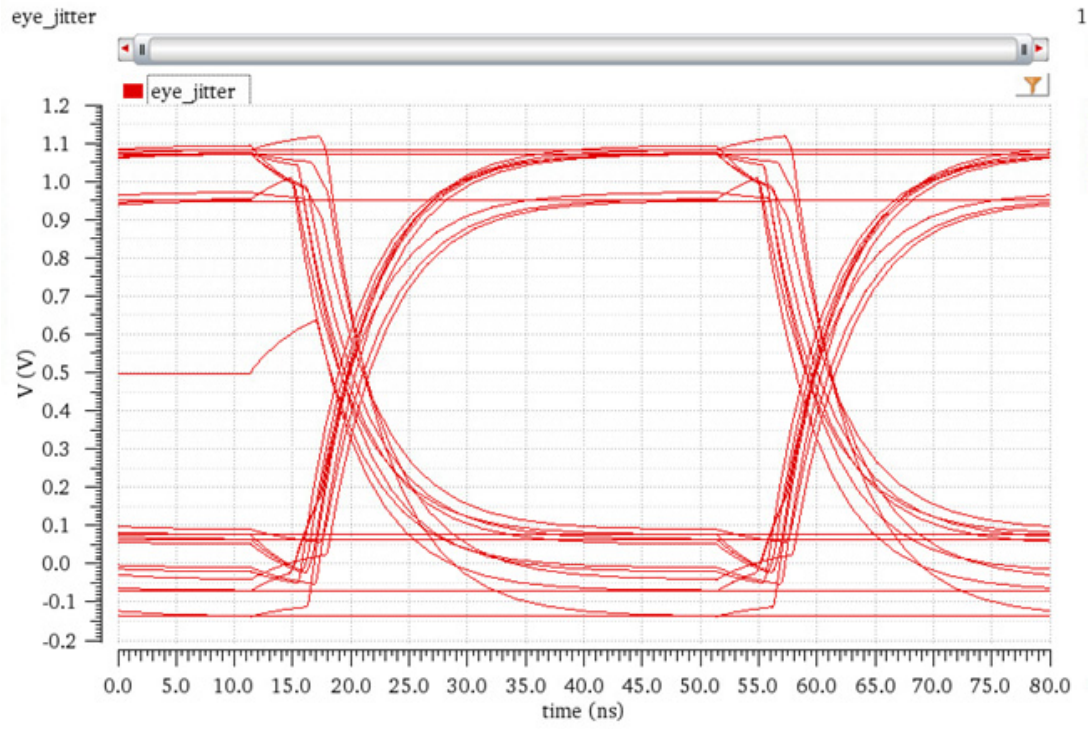
© 2023 Cadence Design Systems, Inc.

Part2: Calculating eye aperture on two eye diagram waveforms

Eye Diagram1: Create an eye diagram waveform for the `jitter` signal by using the *Eye Diagram* assistant with the following field values.

- *Start Time*—`200n`
- *Stop*—`1.4u`
- *Unit Interval*—`40n`
- *Eye Period*—`2*40n` (2UI)

The following eye diagram plot appears:



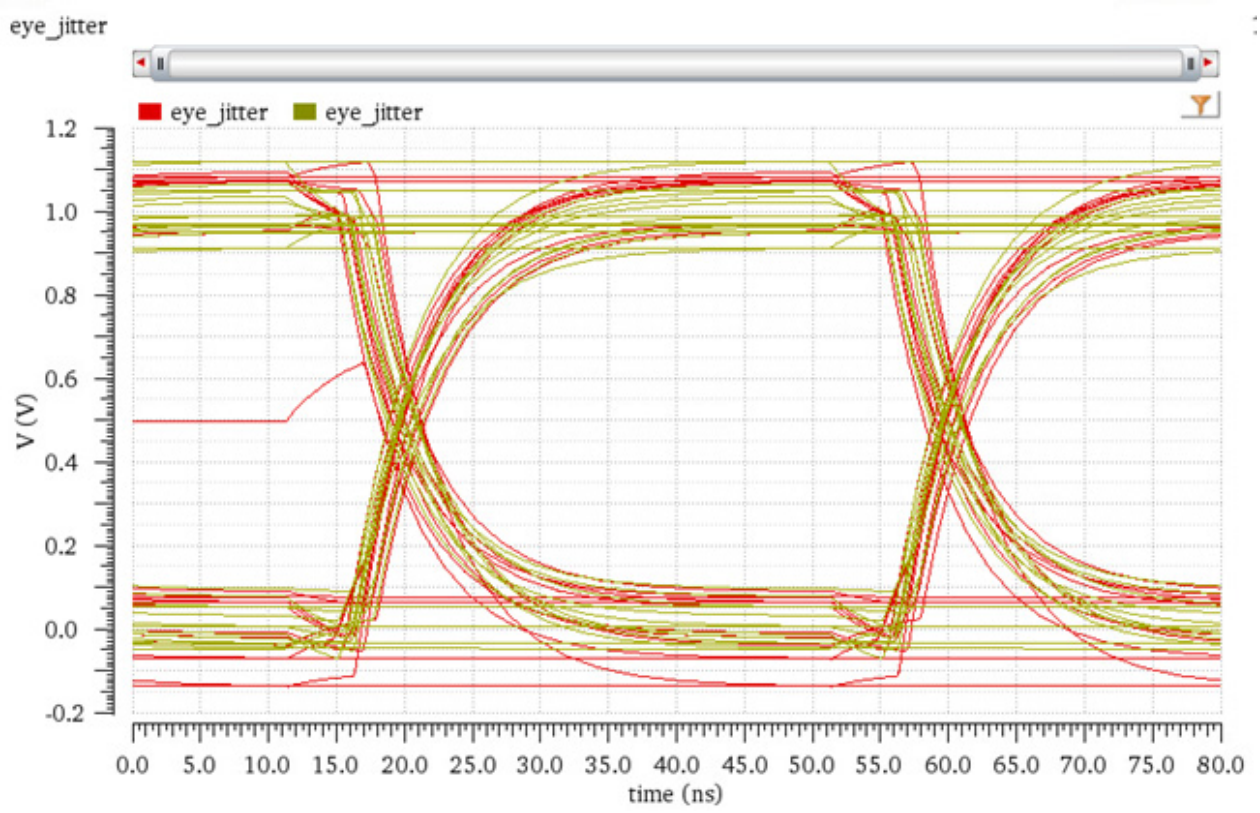
Eye Diagram2: Create the second eye diagram for the `jitter` signal by using the *Eye Diagram* assistant with the following field values:

- *Start Time*—`1.4u`
- *Stop*—`2.8u`
- *Unit Interval*—`40n`
- *Eye Period*—`2*40n` (2UI)

© 2023 Cadence Design Systems, Inc.

Plot the eye diagram in `Append` mode. This new eye diagram is plotted in the same window in which the first eye diagram (created in part1) was plotted.

The following eye diagrams are displayed in the append mode.



Now, select both the eye diagram waveforms by holding down the `Ctrl` key, right-click and choose *Send To – Calculator as a list*.

The following expression is created in the Buffer:

```
list(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n
?triggerPeriod 40n ?autoCenter t) eyeDiagram(v("jitter" ?result
"tran-tran") 1.4u 2.8u 2*40n ?triggerPeriod 40n ?autoCenter t) )
```

Now, apply the `eyeAperture` function on this expression with the following argument values:

- *Signal*—`list(eyeDiagram(v("jitter" ?result "tran-tran") 200n 1.4u 2*40n ?triggerPeriod 40n ?autoCenter t) eyeDiagram(v("jitter" ?result "tran-tran") 1.4u 2.8u 2*40n ?triggerPeriod 40n ?autoCenter t) )`

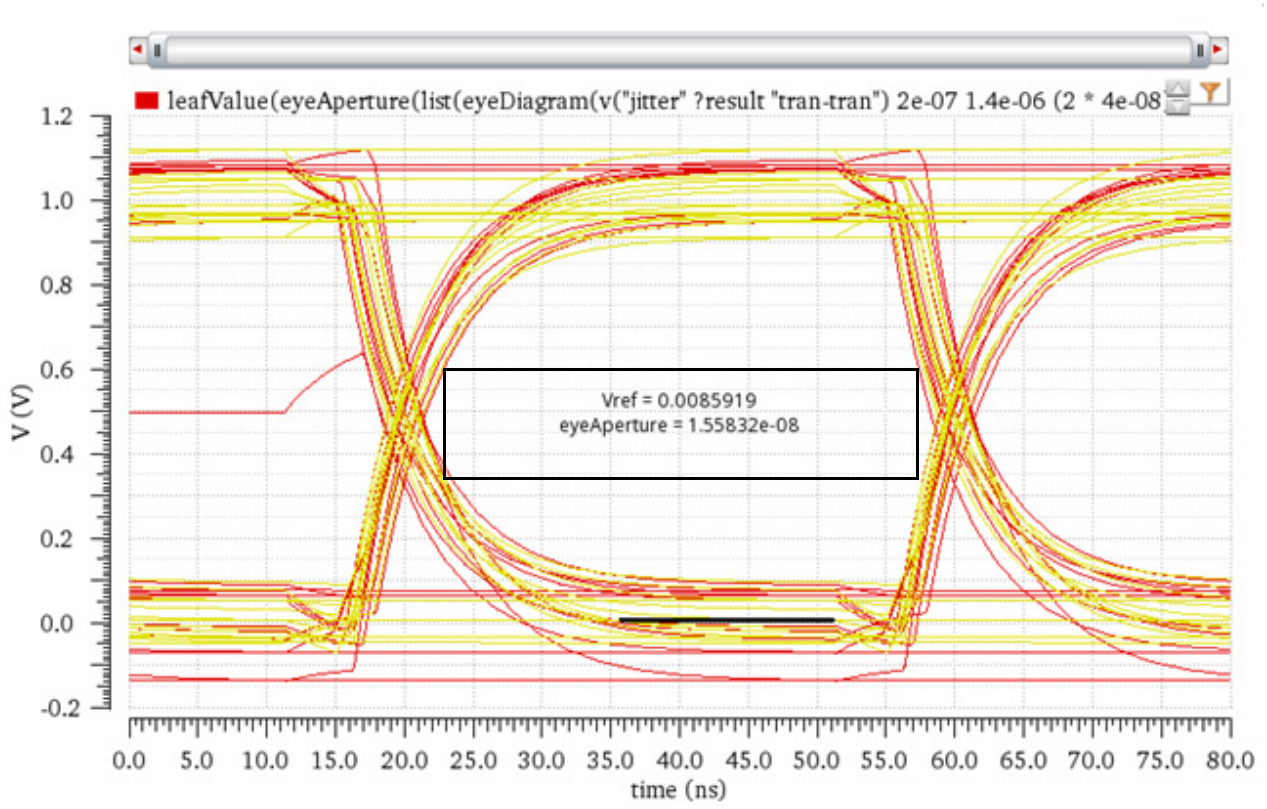
© 2023 Cadence Design Systems, Inc.

- *Vref*—`0.6m`
- *AC Height*—`0.4m`
- *DC Height*—`0.4m`
- *Display Aperture*—`yes`
- *Optimize*—`yes`

When you apply the function, the following expression is created in the Buffer:

```
eyeAperture(list(eyeDiagram(v("jitter" ?result "tran-tran") 200n
1.4u 2*40n ?triggerPeriod 40n ?autoCenter t) eyeDiagram(v("jitter"
?result "tran-tran") 1.4u 2.8u 2*40n ?triggerPeriod 40n ?autoCenter
t) ) 0.6m 0.4m 0.4m t t )
```

When you evaluate this expression, the following output is displayed in a new window:



© 2023 Cadence Design Systems, Inc.

Additional Information

If you specify an output expression using `eyeAperture` and `eyeDiagram` functions in the ADE Outputs Setup pane, the output waveform for eyeAperture is generated after the simulation is complete. In addition, the scalar values for VREF and eyeAperture are displayed in CIW.

For example, create the following expression in ADE Outputs Setup:

```
eyeAperture(eyeDiagram(VT("/net3")0 VAR("a") 1e-10)0 (VAR("a")/100)
```

Now, when you run a simulation, the following output waveform is generated that displays the `eyeAperture` and `Vref` values.

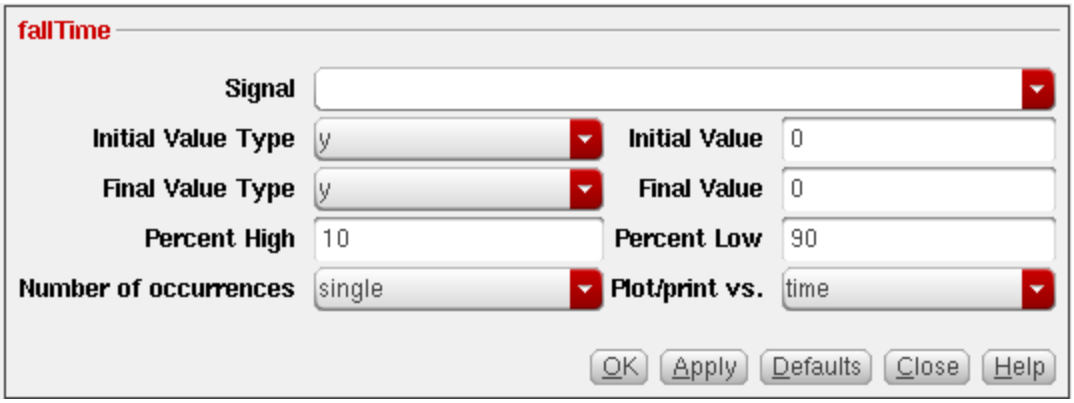
Also, the Vref and eyeAperture scalar values are displayed in CIW:

```
Loading paraplot.cxt
vref = 0.000000e+00. Eye aperture width = 9.357741e-11
net /net3 selected but not highlighted
net /net2 selected but not highlighted
Loading stats.cxt
```

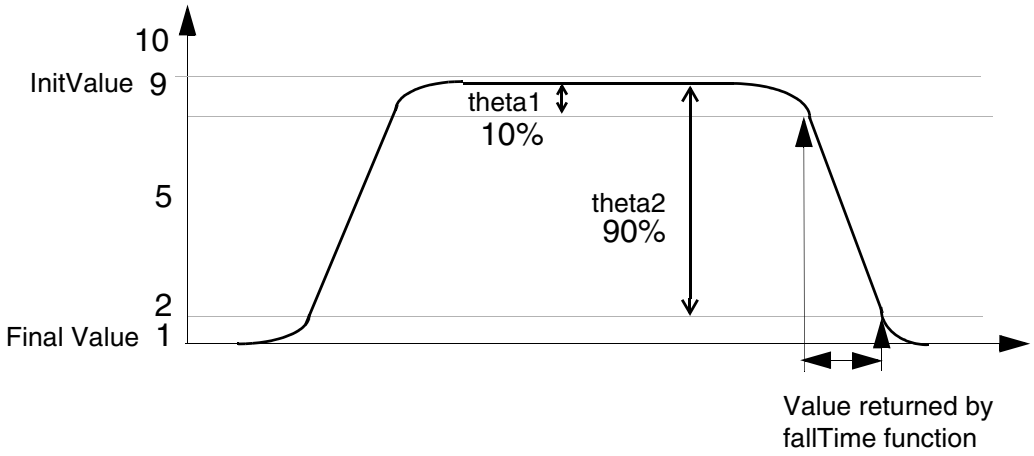

© 2023 Cadence Design Systems, Inc.

fallTime

Returns the fall time measured between *theta1* (percent high) to *theta2* (percent low) of the difference between the initial value and the final value.



The `fallTime` function can also be used to compute the rise time if *initVal* is lower than *finalVal*.



This function includes the following fields:

- *Signal*—Name of the waveform.
- *Initial Value Type*—Initial value type used to start the computation. The options are y at x or y.
- *Initial Value*—Initial value at which the computation is to be started.

© 2023 Cadence Design Systems, Inc.

- *Final Value Type*—Final value type used to end the computation. The options are y at x or y.
- *Final Value*—Final value at which the computation is to be ended.
- *Percent High*—Percentage high difference between initial and final values.
- *Percent Low*—Percentage low difference between initial and final values.
- *Number of occurrences*—Number of occurrences of falls to be calculated. It can be single or multiple.
- *Plot/print vs.*—All falltime values to be calculated per cycle or time.

Additional Information

Consider the following equations:

```
Val1 = theta1 / 100.0 * diff + initVal

Val2 = theta2 / 100.0 * diff + initVal
```

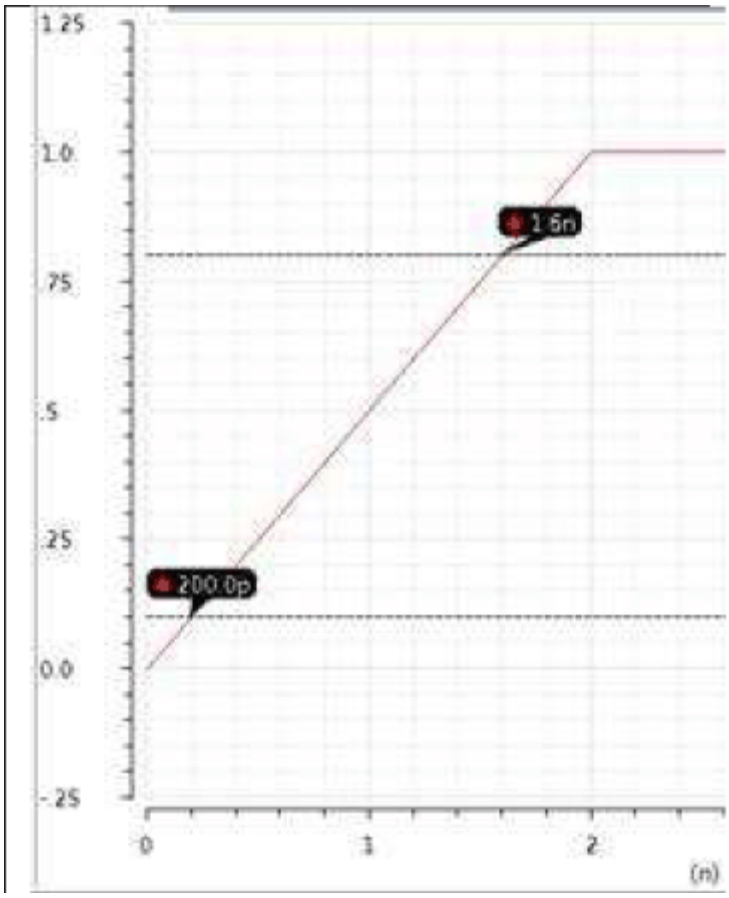

© 2023 Cadence Design Systems, Inc.

The following table shows how the *fallTime* function works when you apply the above equations:

Function *FallTime(w initVal nil finalVal nil theta1 theta2)*	**initVal**	**finalVal**	**theta1**	**theta2**	**Val1**	**Val2**	**Output**
Case 1	0	1	10	80	0.1	0.8	1.4n (time taken to rise from 200.0p ns to 1.6ns) See Figure for Case 1.
Case 2	0	1	80	10	0.1	0.8	0.7n (time taken to fall from 4.2ns to 4.9ns) See Figure for Case 2.
Case 3	1	0	10	80	0.9	0.2	0.7n (time taken to fall from 4.1ns to 4.8ns) See Figure for Case 3.
Case 4	1	0	80	10	0.2	0.9	1.4n (time taken to rise from 400.0p s to 1.8ns) See Figure for Case 4

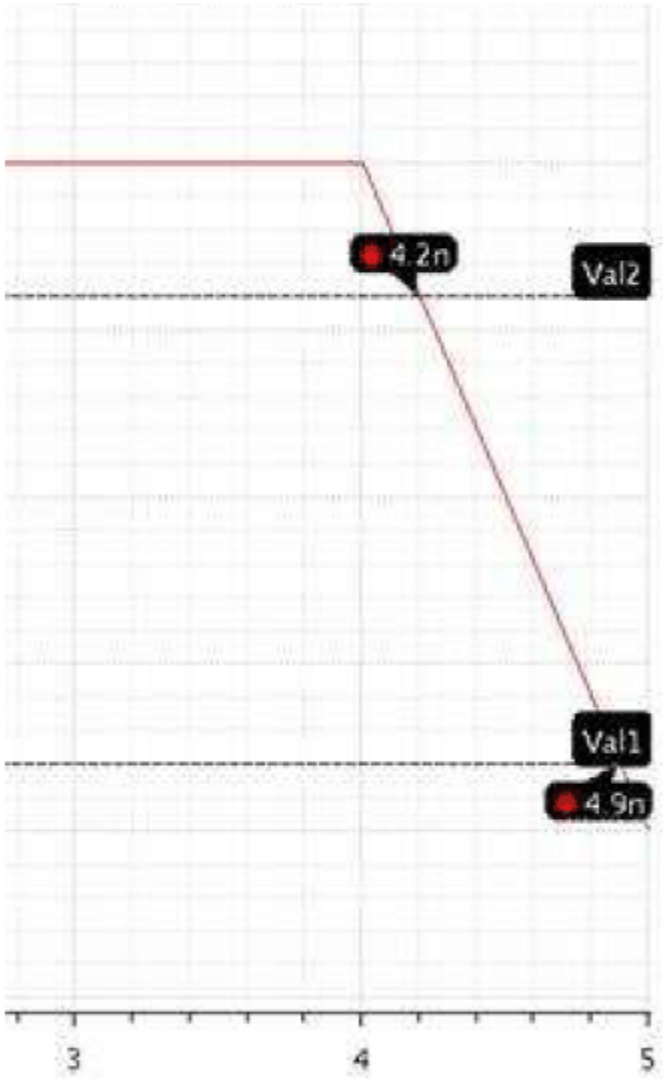
© 2023 Cadence Design Systems, Inc.

Figure for Case 1



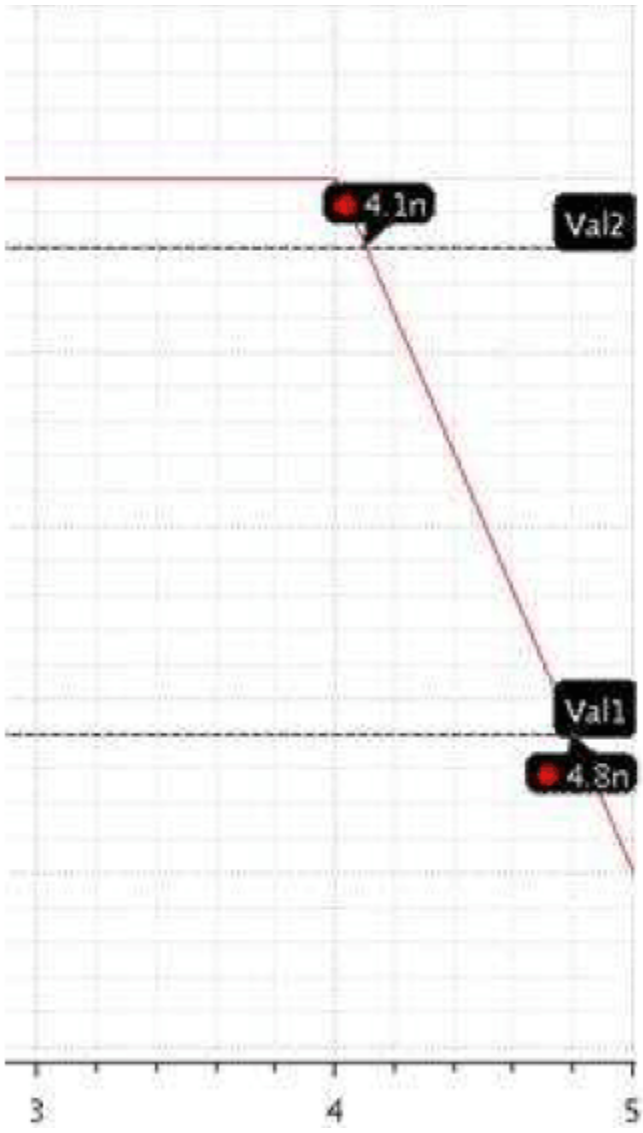
© 2023 Cadence Design Systems, Inc.

Figure for Case 2



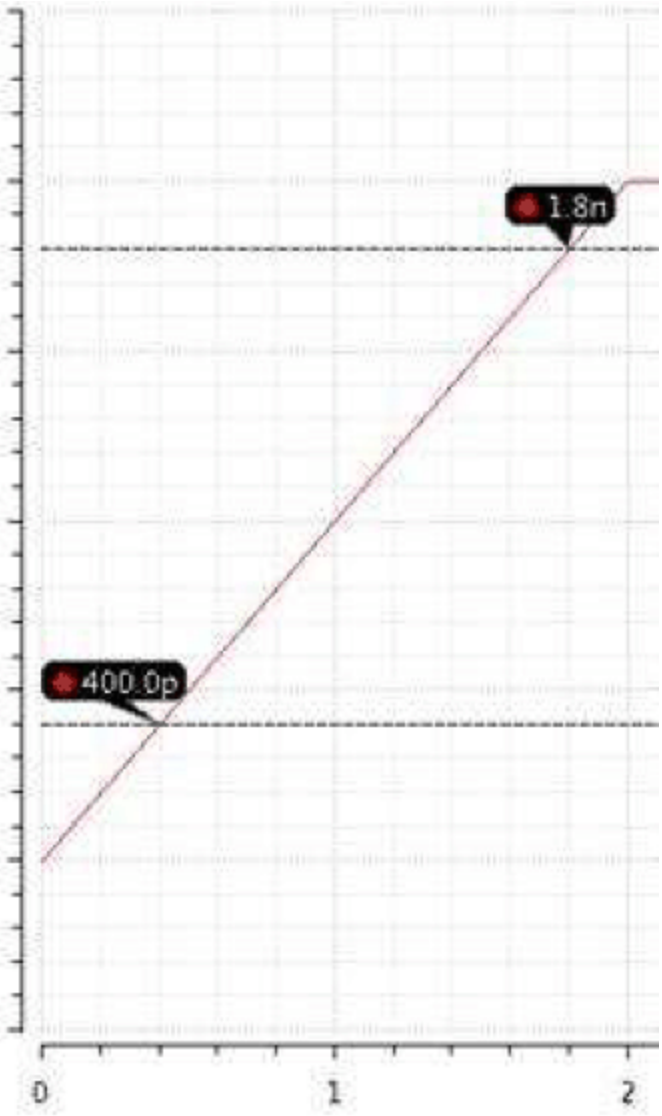
© 2023 Cadence Design Systems, Inc.

Figure for Case 3



© 2023 Cadence Design Systems, Inc.

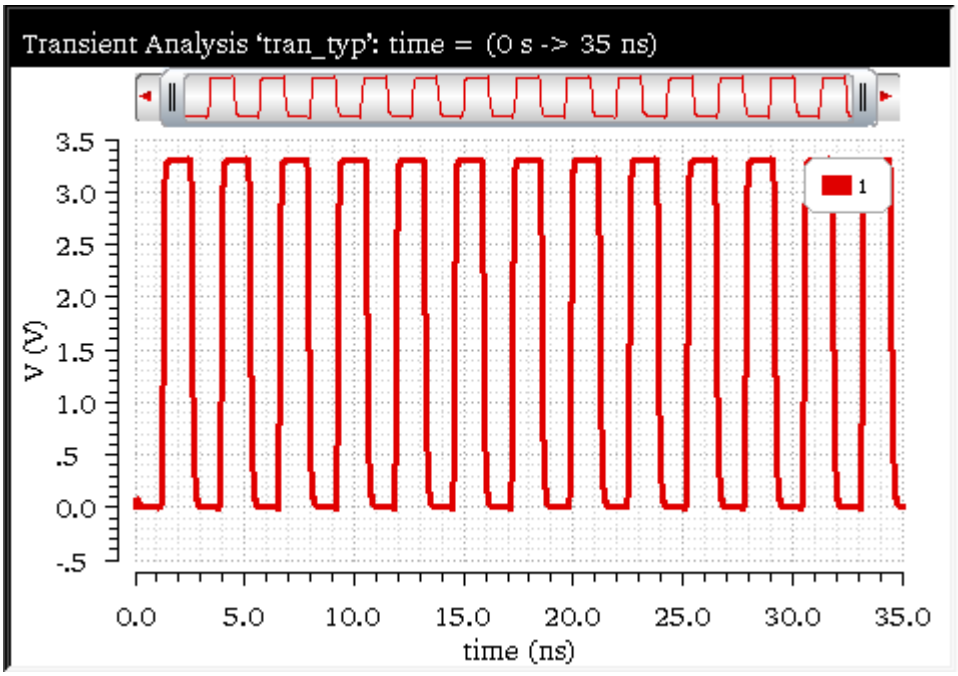
Figure for Case 4



© 2023 Cadence Design Systems, Inc.

Example

This example shows the output waveform genearted when you apply the `fallTime` function on the following input signal:



The following arguments are specified in this example:

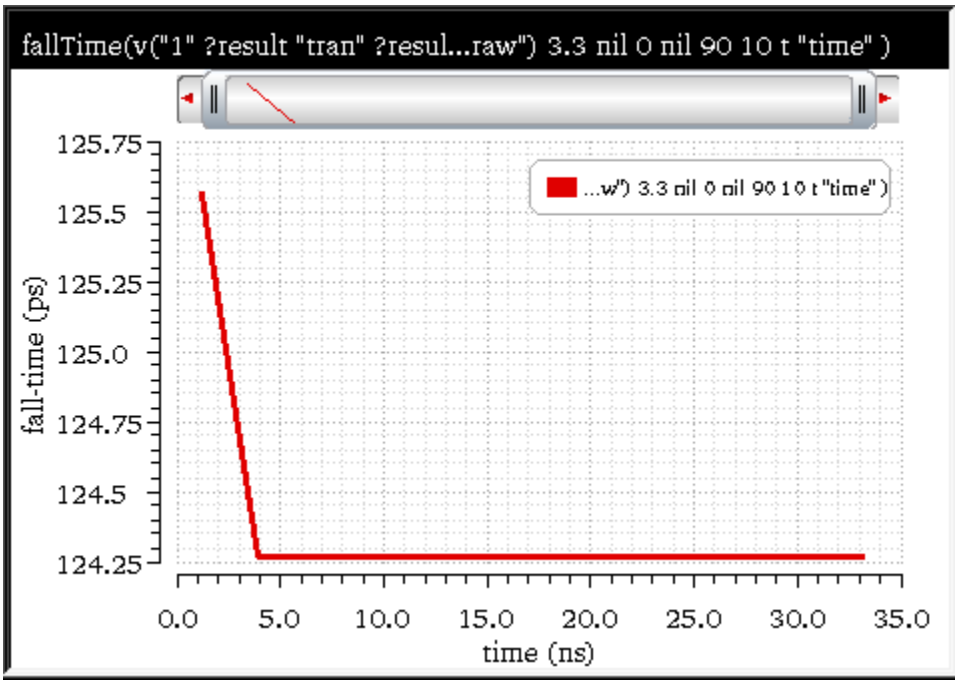
- *Signal*—`v("1" ?result "tran")`
- *Initial Value Type*—`y`
- *Initial Value*—`3.3`
- *Final Value Type*—`y`
- *Final Value*—`0.0`
- *Percent High*—`90`
- *Percent Low*—`10`
- *Number of occurrences*—`multiple`
- *Print/plot vs.*—`time`

The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
fallTime(v("1" ?result "tran" ?resultsDir "./nand2_ring.raw") 3.3 nil
0 nil 90 10 t "time" )
```

Now, when you evaluate this expression, the following output waveform showing the falltime is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `fallTime` is:

```
fallTime( o_waveform n_initVal g_initType
     n_finalVal g_finalType n_theta1 n_theta2
     [g_multiple [s_Xname][g_histoDisplay][x_noOfHistoBins] ] )
     => o_waveform/n_value/nil
```

For more information, see fallTime in *OCEAN Reference*.

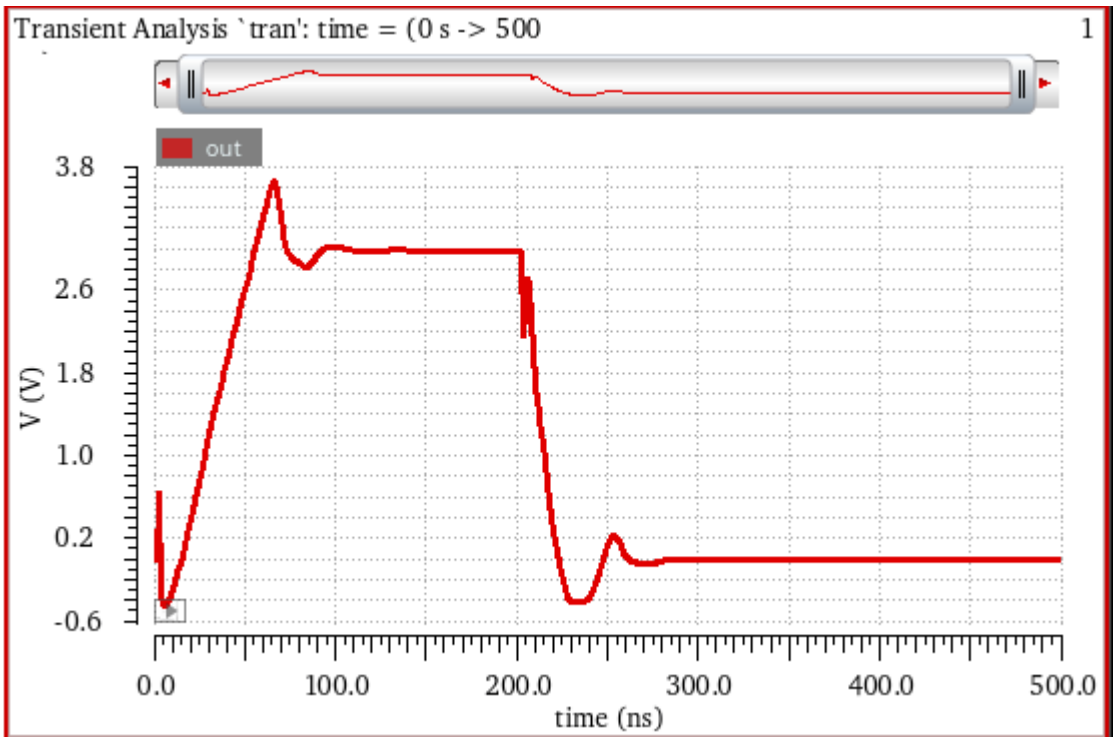
© 2023 Cadence Design Systems, Inc.

firstVal

Returns the first value from where the waveform starts on the X-axis.

Example

Consider the following example in which you run the `firstVal` function on the following transient signal:



When you send this signal to Calculator, the following expression is created in the Buffer:

```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you select the `firstVal` function in Function Panel, it is applied directly on the expression in the Buffer:

```
firstVal(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```

Now, when you evaluate this expression, the function returns the starting value of the `out` signal on the X-axis, as shown in the figure below:

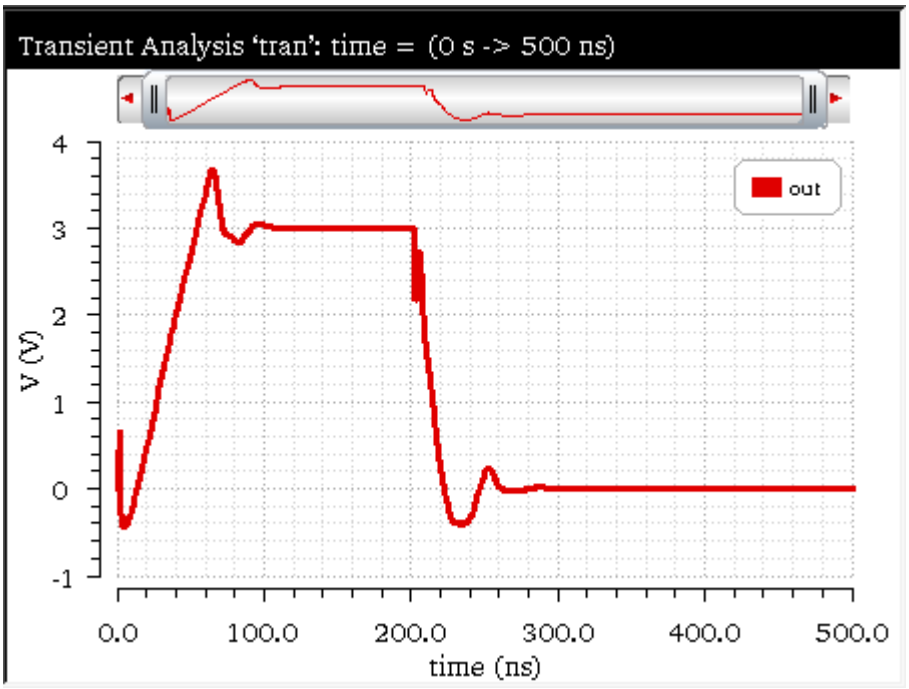
© 2023 Cadence Design Systems, Inc.

flip

Returns a reversed version of a signal (rotates the signal along the Y-axis).

Example

This example displays the output waveform generated when you apply the `flip` function on the following input signal:

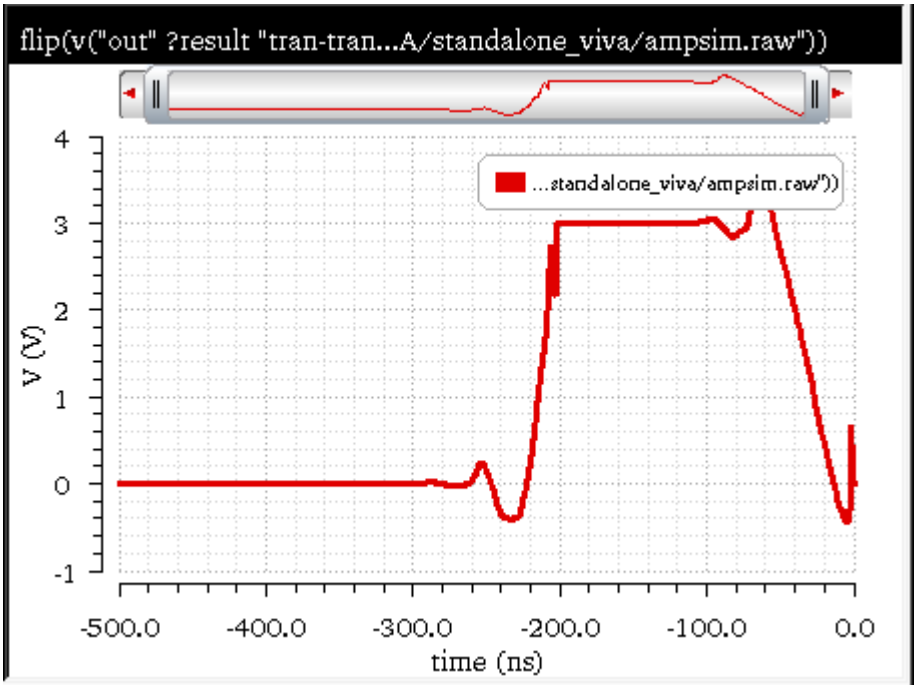


When you apply the `flip` function on this signal, the following expression is created in the Buffer:

```
flip(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform, which is the inverse of the input signal, is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for the `flip` function is:

```
flip( o_waveform )
     => o_waveform/nil
```

For more information about the OCEAN function, see flip in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

fourEval

Evaluates the Fourier series represented by the buffer expression. This function is an inverse Fourier transformation and thus the inverse of the dft (discrete fourier transform) function. It transforms the buffer expression from the frequency domain to the time domain.This function is available only in the SKILL mode.

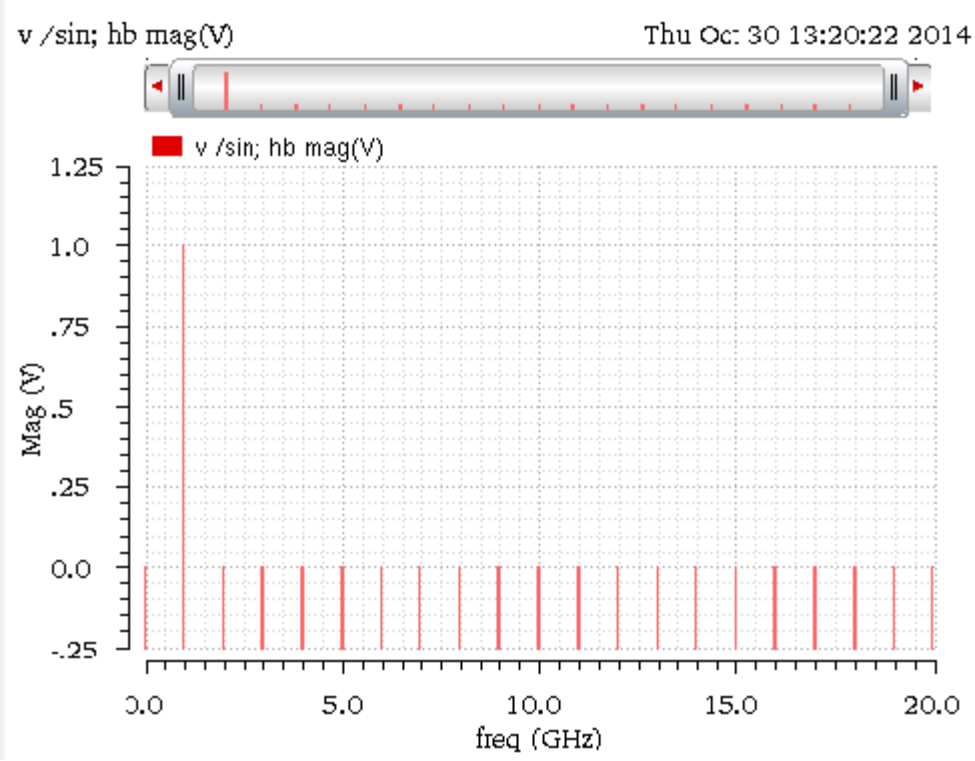
This function includes the following arguments:

- *Signal*—Name of the signal.
- *From*—Time at which you want to begin evaluating the series.
- *To*—Time till which you want to evaluate the series.
- *By*—Increment.
- *Baseband*—Evaluates the baseband version of the inverse of the `dft` function by converting the unsymmetrical spectrum to a symmetrical one (when set to `on`).

© 2023 Cadence Design Systems, Inc.

Example 1

The first example converts the spectrum of a pure sinusoidal signal `v/sin` from SpectreRF Harmonic Balance (hb) analysis to the time domain.



The following argument values are specified in this example:

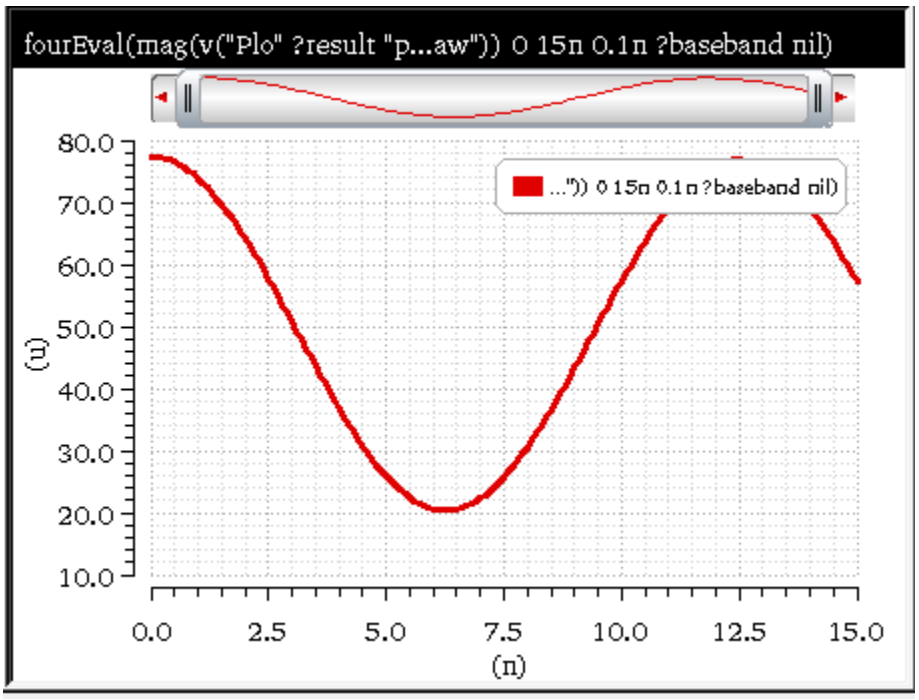
- *Signal*—`v("/sin" ?result "hb" ?resultsDir "./pss_ne600.raw")`
- *From*—0n
- *To*—`15n`
- *By*—`0.1n`
- *Baseband*—`off`

The following expression is created in the Buffer:

```
fourEval(mag(v("Plo" ?result "pss_fd" ?resultsDir "./
pss_ne600.raw")) 0 15n 0.1n ?baseband nil)
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the output waveform for the Fourier series is displayed in the graph window (as shown in the figure below):



Example 2

This example converts the spectrum of a pure sinusoid from SpectreRF Harmonic Balance analysis to the time domain. The frequency of the sinusoid is `1`GHz and evaluated from `0` to `2`ns in increments of `0.01`ns.

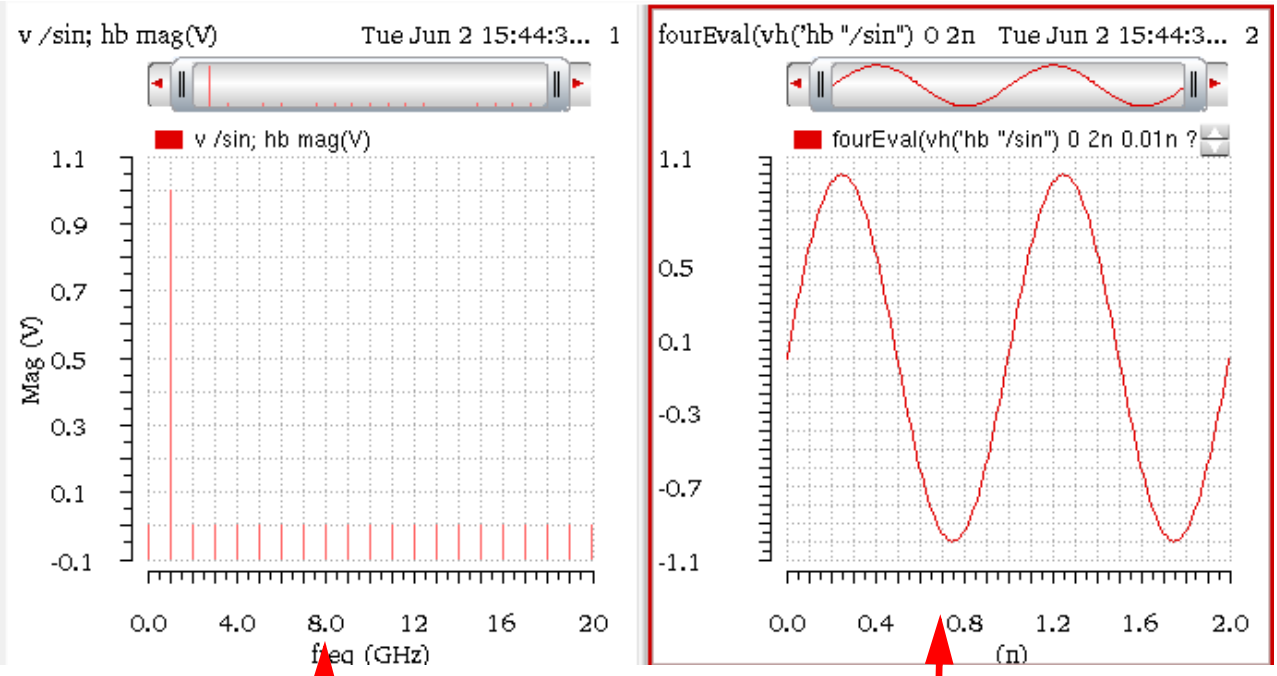
Consider the `test_sin` library in the `fourtest` test case. The following outputs are displayed window when you load the cellview `test_sin`:

Outputs

	Name/Signal/Expr	Value	Plot	Save	Save Options
1	v /sin; hb mag(V)	wave	☑	☐	
2	fourEval(vh("hb "/sin") 0 2...	wave	☑	☐	

© 2023 Cadence Design Systems, Inc.

When you run the simulation and plot the outputs, the following waveforms are displayed in the graph window:



Sinusoidal spectrum from HB analysis

Converting a 1GHz sinusoid to the time domain over two periods using fourEval(vh('hb "/sin") 0 2n 0.01n ?baseband nil)

Example 3

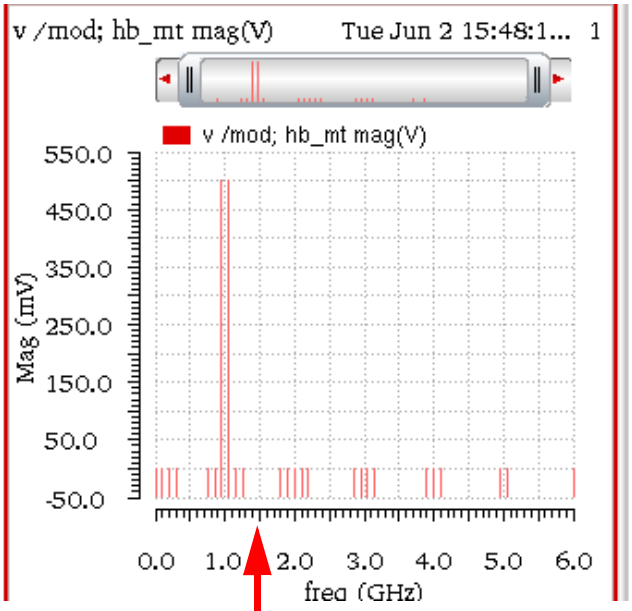
This example demonstrates that the frequencies in the spectrum are not required to be equally-spaced. Consider the example of two closely spaced sinusoids:

$$x(t) = 0.5 \times \sin((\omega_0 - \Delta\omega)t) + 0.5 \times \sin((\omega_0 + \Delta\omega)t) = \cos(\Delta\omega t)\sin(\omega_0 t) \quad (3)$$

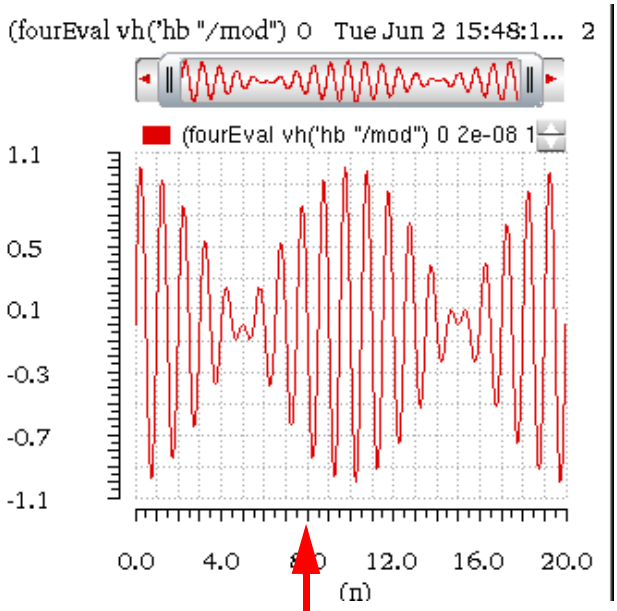
Consider the `test_mod` library in the `fourtest` test case. The following outputs are displayed when you load the cellview `test_mod`:

© 2023 Cadence Design Systems, Inc.

When you run the simulation and plot the outputs, the following waveforms are displayed in the graph window:



The spectrum of two closely-spaced sinusoids from SpectreRF HB analysis.



The result of fourEval(vh'hb "/mod") 0 20n 0.01n ?baseband nil), showing a high-frequency carrier modulated by a 50MHz "beat" signal.

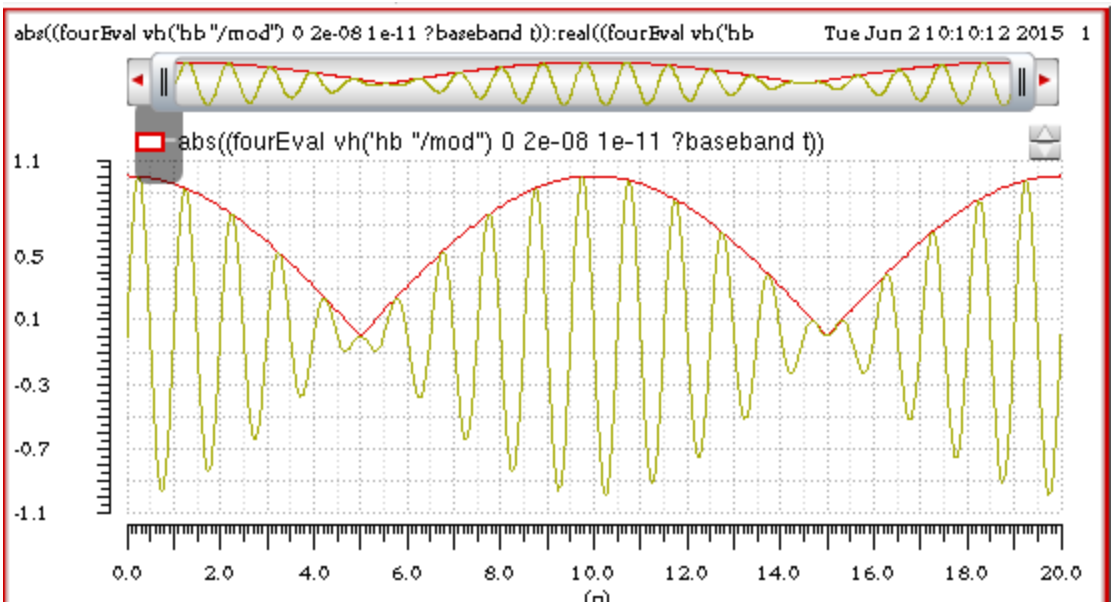
Example 4

This example applies `fourEval()` to the problem of envelope detection. The signal shown in Example 2 is amplitude-modulated by a `50` MHz sinusoid, and its envelope is `|cos((2pi)t*50MHz)|`

Consider the `test_bb` library in the `fourtest` test case. The following outputs are displayed when you load the cellview `test_bb`:

© 2023 Cadence Design Systems, Inc.

When you run the simulation and plot the output, the output waveform is displayed in the graph window. In this output waveform, the magnitude of the analytic signal (produced by `fourEval()` when baseband=`t`) represents the envelope of the signal.



Related OCEAN Function

The equivalent OCEAN command for `fourEval` function is:

```
fourEval( o_waveform n_from n_to n_by [?g_baseBand] )
     => o_waveform/nil
```

For more information about the OCEAN function, see fourEval in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

freq

In the SKILL mode, `freq` returns a waveform representing the frequency of a signal versus time or cycle.

Unlike the frequency function, which takes only waveform as input and returns the average frequency value, the `freq` function plots multiple occurrences of frequency based on the specified edge types. You can use this function to plot frequencies against time or cycle when the waveform includes a varying period.

This function includes the following arguments:

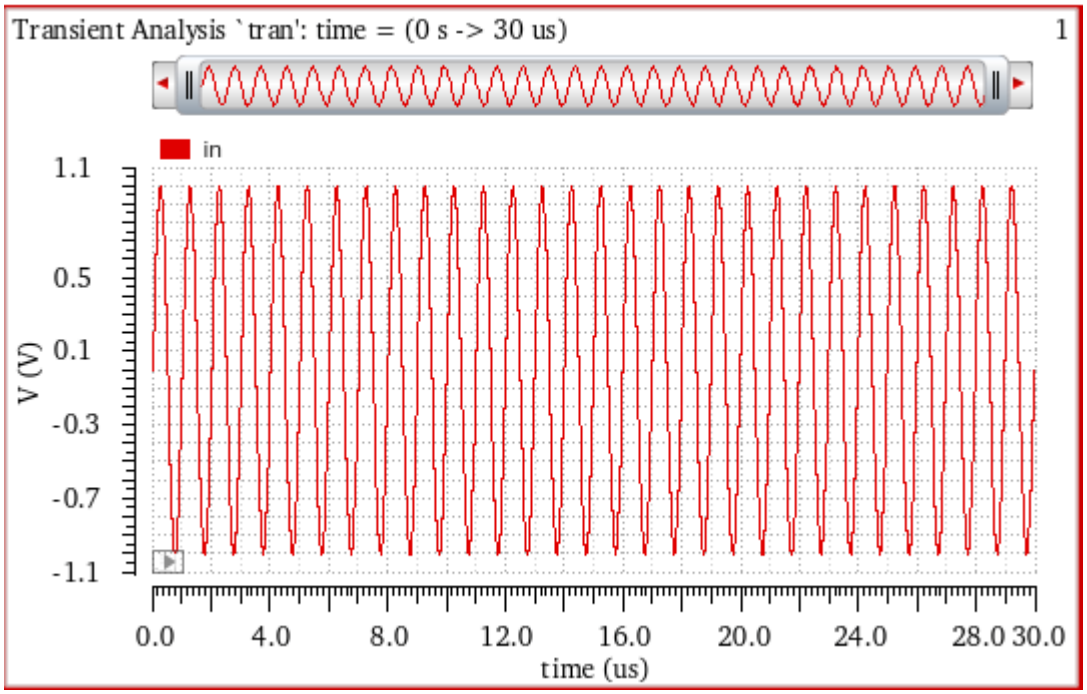
- *Signal*—Name of the input signal.
- *Edge Type*—Direction of the crossing event.
  `rising` directs the function to look for crossings where the Y value is increasing
  `falling` directs the function to look for crossings where the Y value is decreasing
- *Plot vs.*—Specifies whether you want to retrieve frequency against *time* (or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform.
- *Threshold Mode*—Specifies the mode used to calculate the threshold value. If you want to specify the threshold value, select `user`. If you want to automatically calculate the threshold value, select `auto`. The *auto* threshold is calculated as:

  Auto Threshold Value = integral of the waveform divided by the X range.
- *User Threshold Value*—Threshold Y-axis value to be crossed. In SKILL mode, this field is available only if the *Threshold Mode* is *user*.

© 2023 Cadence Design Systems, Inc.

Example 1

This example calculates the multi-occurences of frequency on the rising edges of the input transient waveform and plotted the output waveform against time. The `freq` function is applied on the following input signal:



Specify the following argument values:

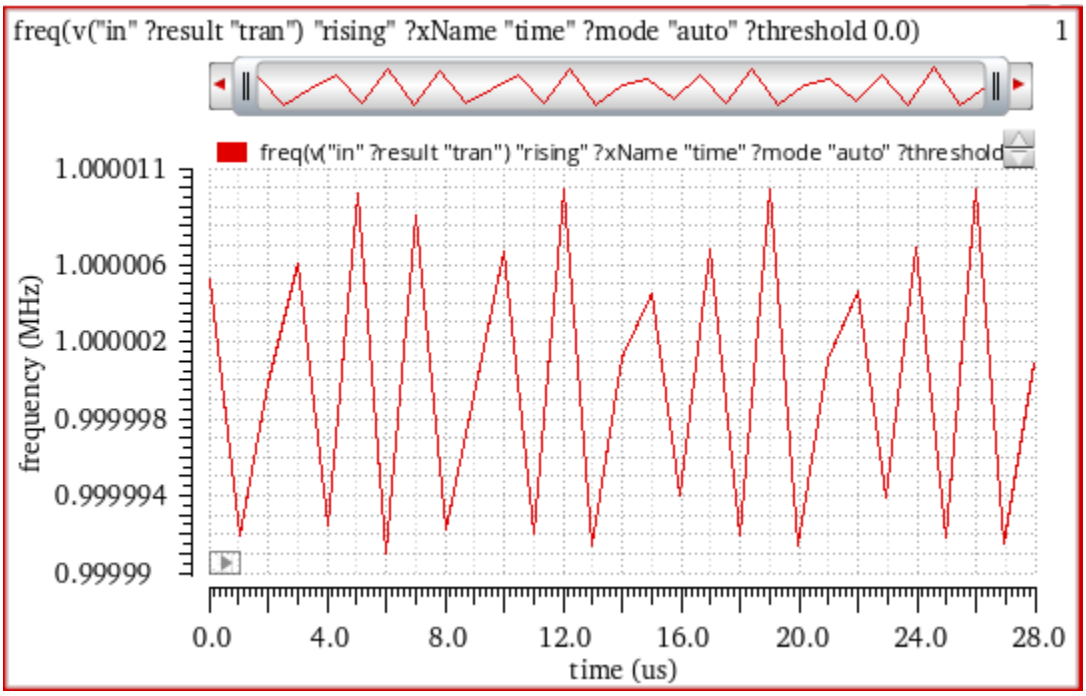
- *Signal*—`v("in" ?results "tran" )`
- *Edge Type*—`rising`
- *Plot Vs.*—`time`
- *Threshold Mode*—`auto`
- *User Threshold Value*—`0.0`

The auto threshold mode is specified in this example. You can also set the threshold mode as user and specify a threshold value.The following expression is created in the Buffer:

```
freq("v("in" ?results "tran" ?resultsDir "vsin_sim.raw")" "rising"
?xName "time" ?mode "auto" ?threshold 0.0)
```


© 2023 Cadence Design Systems, Inc.

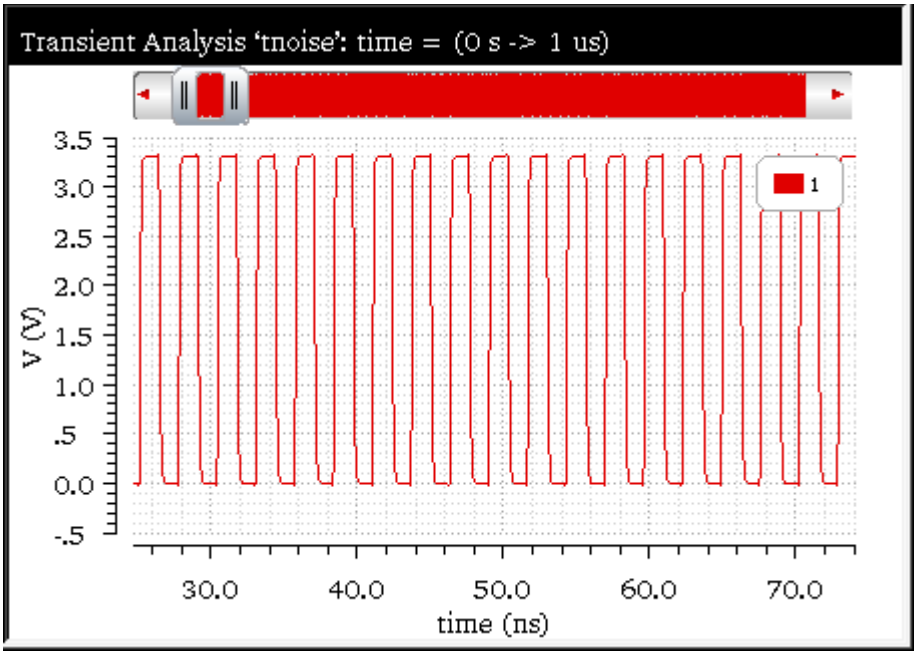
When you evaluate this expression, the frequency on the rising edges of the input waveform is calculated and plotted against time, as shown in the figure below.



© 2023 Cadence Design Systems, Inc.

Example 2

This example shows the output generated when you apply the `freq` function on the input signal plotted in the graph below:



Specify the following argument values:

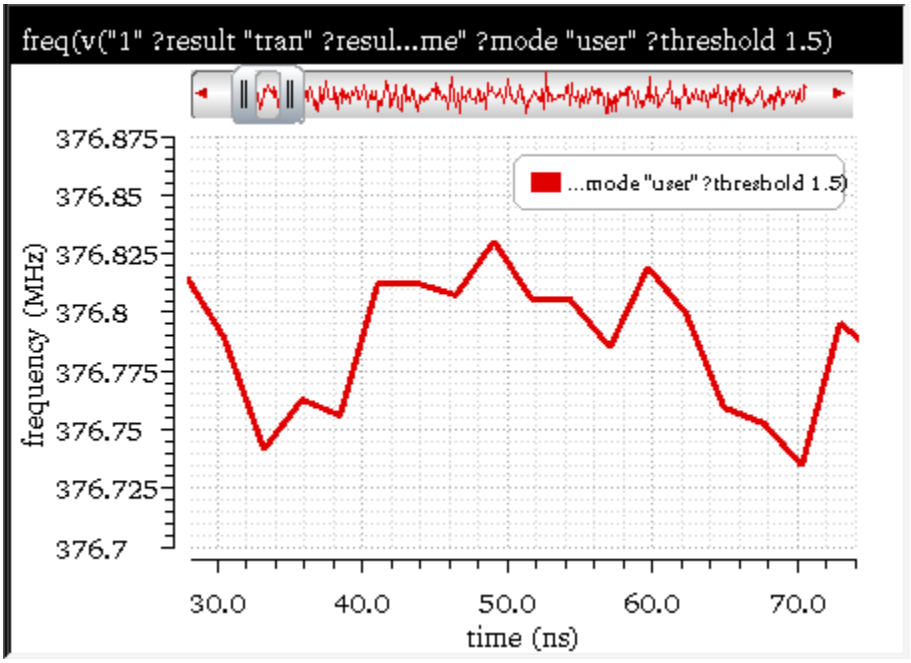
- *Signal*—`v("1" ?results "tran")`
- *Edge Type*—`rising`
- *Plot Vs.*—`time`
- *Threshold Mode*—`user`
- *User Threshold Value*—`1.5`

The following expression is created in the Buffer:

```
freq("v("1" ?results "tran" ?resultsDir "nand2_ring_tnoise.raw")"
"rising" ?xName "time" ?mode "user" ?threshold 1.5)
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN function for `freq` is:

```
freq( o_waveform t_crossType
      [ ?threshold n_threshold ]
      [ ?mode t_mode ]
      [ ?xName xName ]
      [ ?histoDisplay g_histoDisplay ]
      [ ?noOfHistoBins x_noOfHistoBins ] )
     => o_outputWave/nil
```

For more information about this OCEAN function, see freq in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

freq_jitter

Returns a waveform representing the deviation from the average frequency.

This function includes the following arguments:

- *Waveform*—Name of the signal, expression, or family of waveforms.
- *Cross Type*—Points at which the curves of the waveform intersect with the threshold. While intersecting, the curve may be either rising (`rising`) or falling (`falling`).
- *mode*—Specifies the mode used to calculate the threshold value. If you want to specify the threshold value, select `user`. If you want that the Virtuoso Visualization and Analysis XL calculates the threshold value, select `auto`. The auto threshold is calculated as:

  Auto Threshold Value = integral of the waveform divided by the X range.
- *Threshold*—Threshold value against which the frequency is to be calculated. You need to specify the threshold value only if the *mode* is `user`.

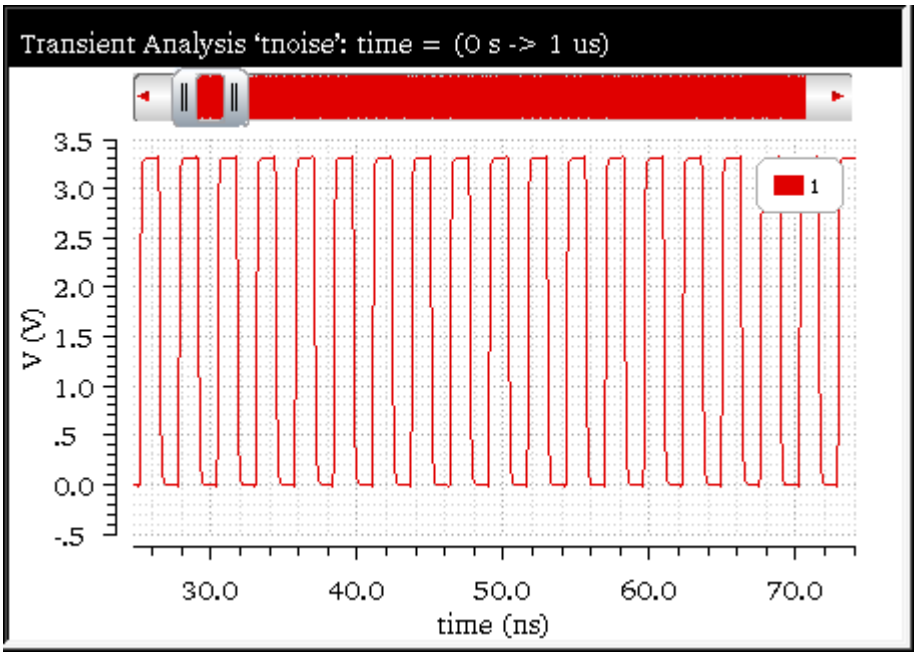
  Note: A threshold tolerance of `0.1*stddev(signal)` is used when detecting crossings.
- *Bin Size*—Width of the moving average window, The deviation of value at the particular point from the average of this window is the jitter.
  If *binsize*=0, all frequencies are used to calculate the average.
  If *binsize*=*N*, the last *N* frequencies are used to calculate the average.
- *Plot/print vs.*—Specifies whether you want to retrieve the frequency jitter against *time* (or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform.

© 2023 Cadence Design Systems, Inc.

- *Output Type*—Type of output. If set to `sd`, the output is a standard deviation jitter. If set to `plot`, the output is a waveform. The default value is `plot`.

Example

This example shows the output waveform generated when you apply the `freq_jitter` function on the following waveform:



The arguments specified in this example are as follows:

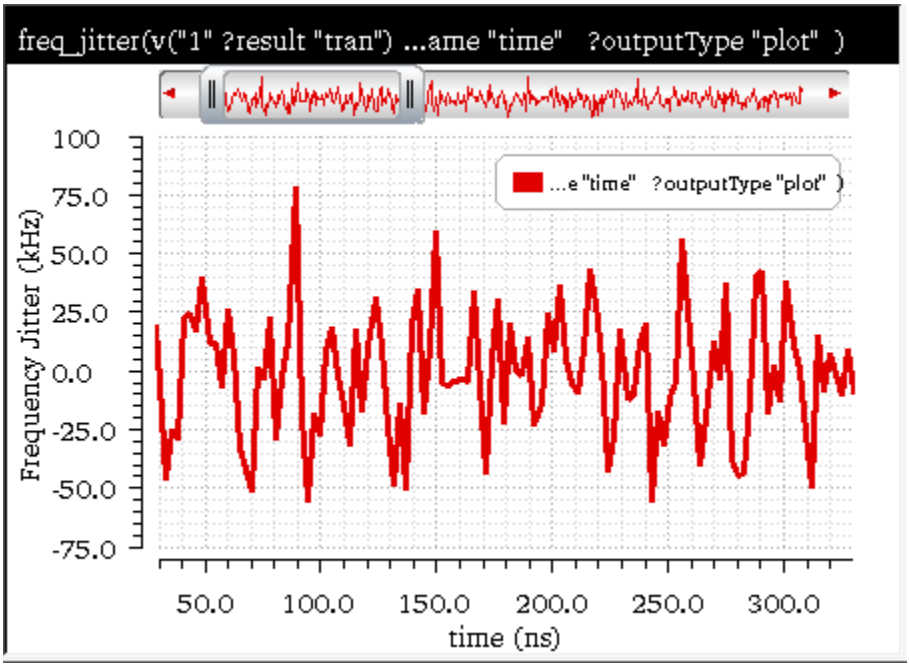
- *Signal*—`v("1" ?result "tran")`
- *Cross Type*—`rising`
- *mode*—`auto`
- *Threshold*—`1.5`
- *Bin Size*—`30`
- *Plot/print vs.*—`time`
- *Output Type*—`plot`

The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
freq_jitter(v("1"?result "tran" ?resultsDir "nand2_ring_tnoise.raw")
"rising"  ?mode "auto"   ?binSize 30 ?xName "time"   ?outputType
"plot" )
```

When you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN function for `freq_jitter` is:

```
freq_jitter( o_waveform t_crossType
     [?mode t_mode] [?threshold n_threshold] [binSize n_binSize]
     [?xName t_xName] [?outputType t_outputType] )
     => o_waveform/f_val/nil
```

For more information about this OCEAN function, see freq_jitter in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

frequency

Returns the average frequency of all cycles plotted. This function is available only in the SKILL mode.

The following equation describes how the average frequency can be calculated:

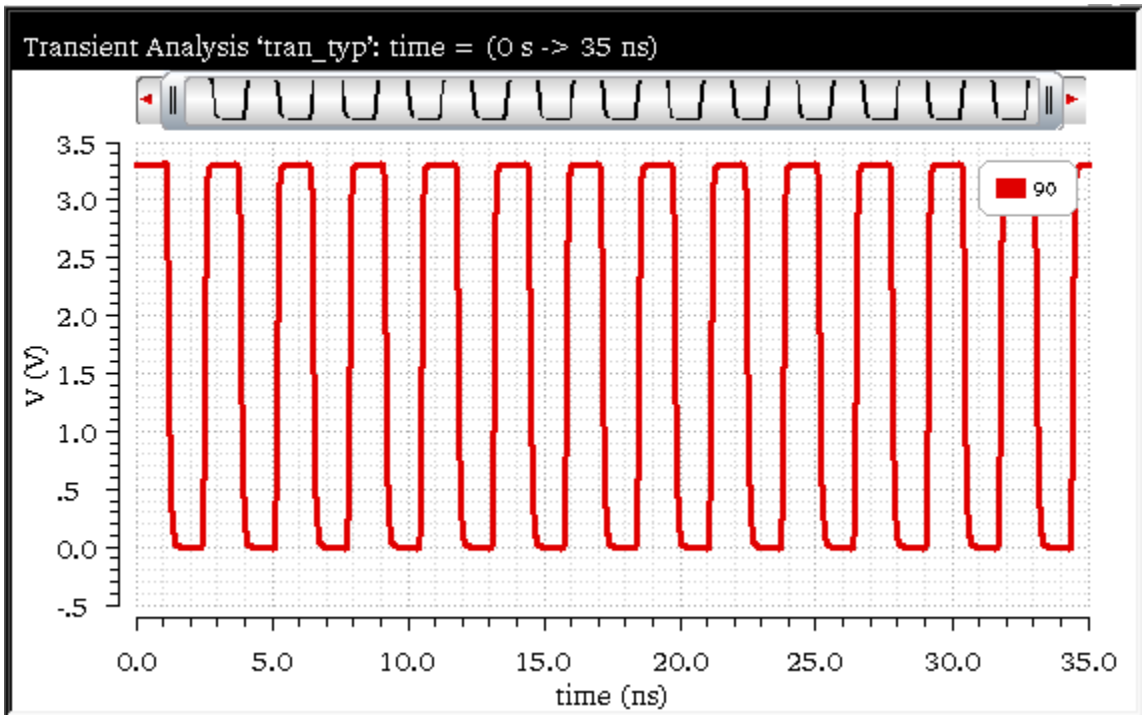
$$avgFreq = \frac{\text{Total number of crosses} - 1}{\text{Total X range between first and last cross}}$$

Here, threshold is defined as:

$$Threshold = \left(\frac{Y_{max} + Y_{min}}{2}\right) \text{of the input waveform}$$

Example

This example shows the output generated when you apply the `frequency` function on the following input waveform (shown in the figure below):



The following expression is created in the Buffer when you apply the `frequency` function:

© 2023 Cadence Design Systems, Inc.

```
frequency(v("90" ?result "tran" ?resultsDir "nand2_ring.raw))
```

When you evaluate this expression, the following output value is displayed in the Buffer (shown in the figure below):

Related OCEAN Function

The equivalent OCEAN function for `frequency` is:

```
frequency( o_waveform )
     => o_waveform/n_value/nil
```

For more information about this OCEAN function, see frequency in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

gainBwProd

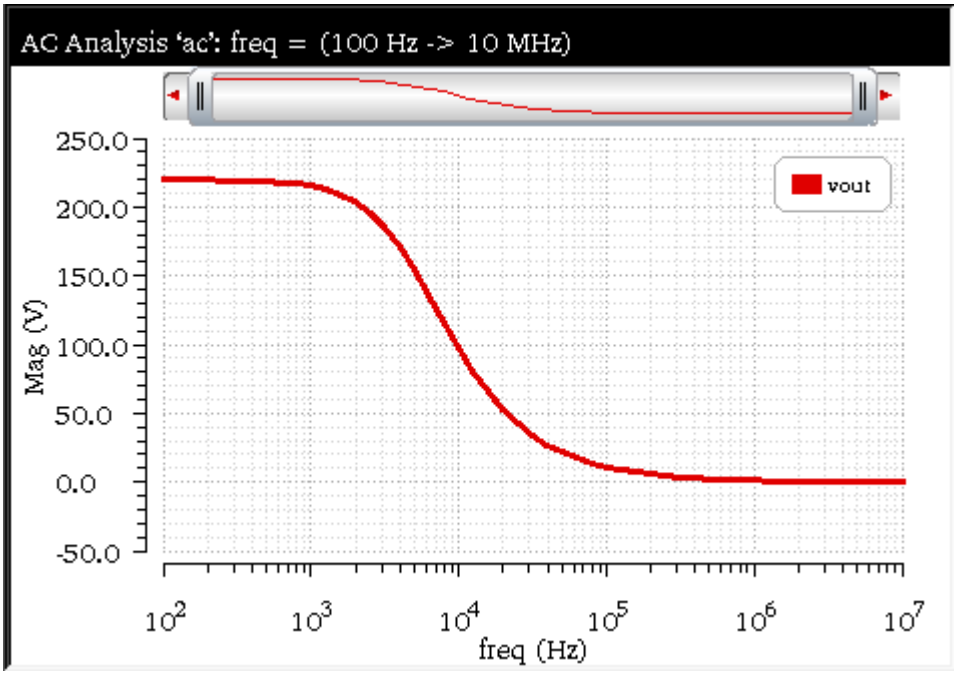
Returns the product of DC gain and upper cutoff frequency for a low-pass type filter or amplifier. (Returns the product of the zero-frequency-gain and 3dB-gain-frequency). Calculates the gain-bandwidth product of a waveform representing the frequency response of interest over a sufficiently large frequency range.

$$gainBwProd(gain) = A_0 \times f_2$$

The gain-bandwidth product is calculated as the product of the DC gain and the critical frequency. The critical frequency is the smallest frequency for which the gain equals 21 times the DC gain 0A.

Example

This example shows the gain bandwidth value generated when you apply the `gainBwProd` function on the following input waveform:

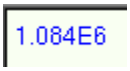


When you apply the function, the following expression is created in the Buffer:

```
gainBwProd(mag(v("vout" ?result "ac" ?resultsDir "./
openloop_ampSim.raw")))
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following result is displayed in the Buffer:



Related OCEAN Function

The equivalent OCEAN command for `gainBwProd` is:

```
gainBwProd( o_waveform )
     => o_waveform/n_value/nil
```

For more information about this OCEAN function, see gainBwProd in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

gainMargin

Computes the gain margin of the loop gain of an amplifier.

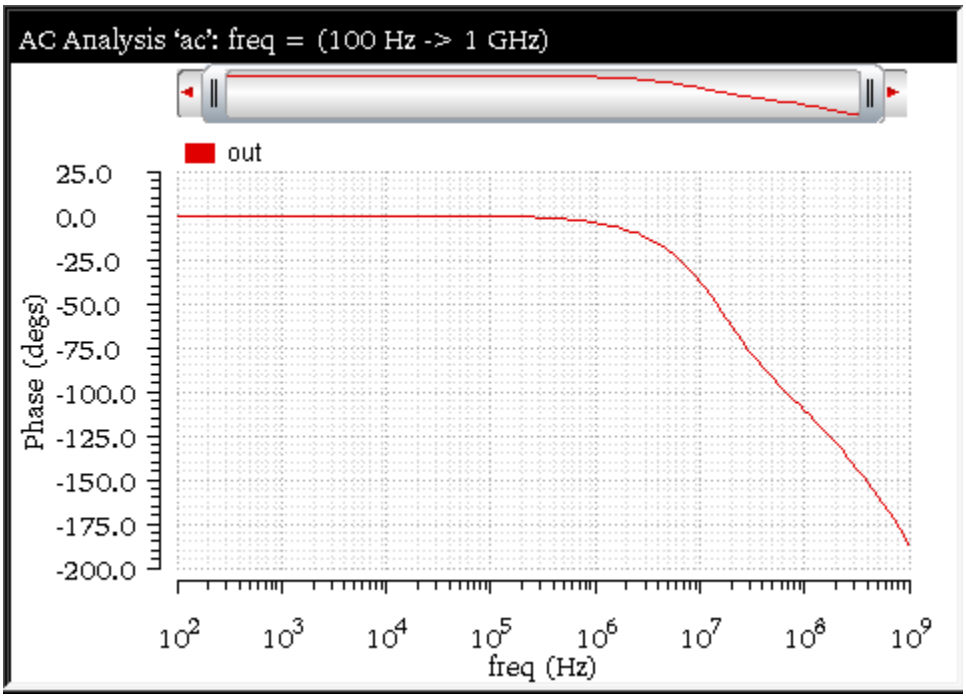
The gain margin is calculated as the magnitude (in dB) of the gain at f0. The frequency f0 is the smallest frequency in which the phase of the gain provided is -180 degrees. For stability, the gain margin must be positive.

$$\text{gainMargin}(\text{gain}) = 20\log 10(value(gain\ f_0))$$

The first argument is a waveform representing the loop gain of interest over a sufficiently large frequency range. This command returns the dB value of the waveform when its phase crosses negative pi.

Example

This example shows the gain margin value generated when you apply the `gainMargin` function on the following AC signal:



When you apply this function, the following expression is created in the Buffer:

```
gainMargin(v("out" ?result "ac" ?resultsDir "./ampTest.raw"))
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output value is displayed in the Buffer:

Related OCEAN Function

The equivalent OCEAN command for `gainMargin` is:

```
gainMargin( o_waveform [g_stable])
     => o_waveform/n_value/nil
```

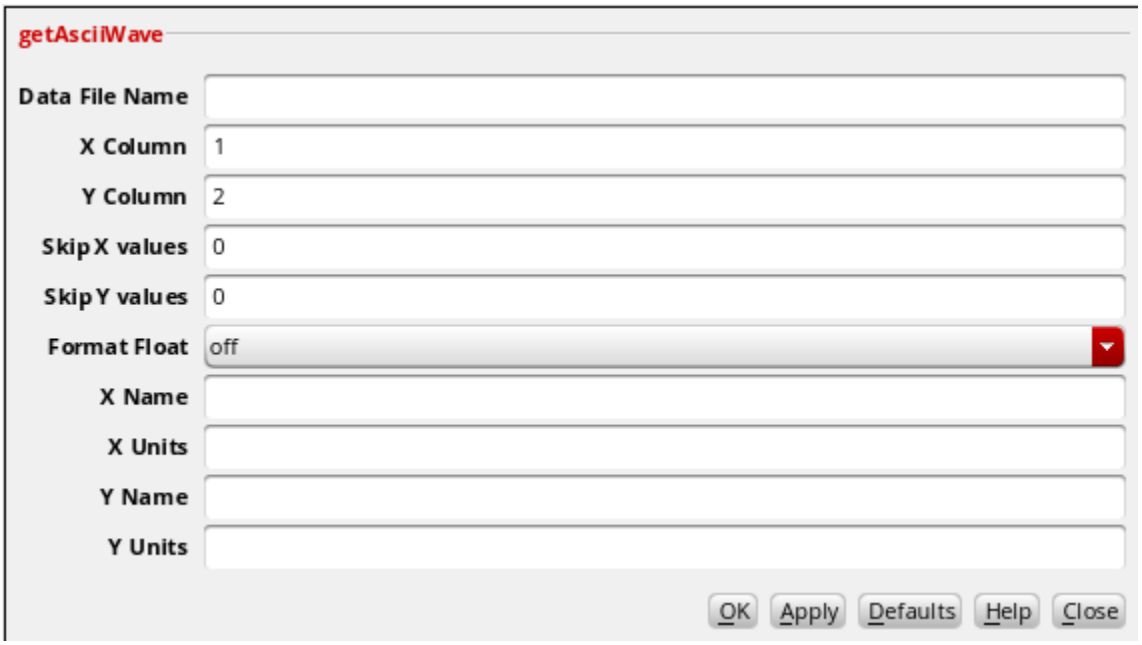
For more information about this OCEAN function, see gainMargin in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

getAsciiWave

Returns a piece-wise linear function from a column of Xand Y values in a file. The `getAsciiWave` function is available only in the SKILL mode.

This function reads an ASCII data file and generates a waveform object from the specified data. The X-axis data must contain real numbers, whereas the Y-axis data can be real or complex values. Complex values are represented as `(real imag)` or `complex(real imag)`. This function skips blank lines and comment lines. Comments are defined as the lines beginning with a semicolon.



The function includes the following arguments:

- *Data File Name*—Path of the ASCII data filename.
- *X Column*—Number of columns to be read as X-axis.
- *Y Column*—Number of columns to be read as Y-axis.
- *Skip X values*—The number of lines to be skipped in X column.
- *Skip Y values*—The number of lines to be skipped in Y column.
- *Format Float*—When set to `on`, converts the integer data into float values. Default value: `off`.
- *X Name*—Name of the X-axis.

© 2023 Cadence Design Systems, Inc.

- *X Units*—Units for the X-axis values.
- *Y Name*—Name of the Y-axis
- *Y Units*—Units for the Y-axis values.

Example 1

This example shows the graph plotted when you apply the `getAsciiWave` function on the following data file:

```
./vout_tran_xy.csv
```

The text file contains the data in the following format:

Time	**V(OUT)**
0	1.333820380642945
1e-06	1.335676012809686
:	:
:	:

Specify the following field values:

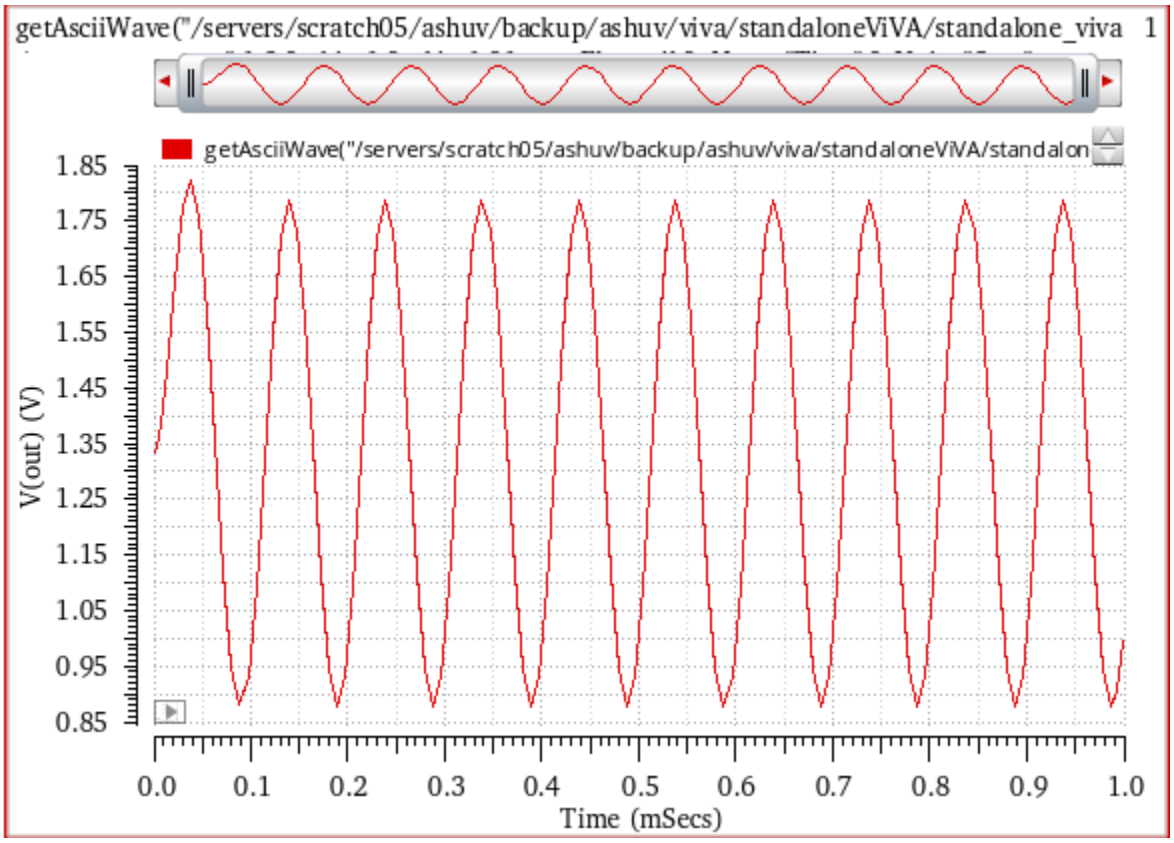
- *Data File Name*—`./vout_tran_xy.csv`
- *X Column*—`1`
- *Y Column*—`2`
- *Skip X values*—`1`
- *Skip Y values*—`1`
- *Format Float*—`off`
- *X Name*—`Time`
- *X Units*—`Secs`. The X-axis name is displayed as `mSecs`.
- *Y Name*—`V(out)`
- *Y Units*—`V`

The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
getAsciiWave("./vout_tran_xy.csv" 1 2 ?xskip 1 ?yskip 1 ?formatFloat
nil ?xName "Time" ?xUnits "Secs" ?yName "V(Out)" ?yUnits "V")
```

When you evaluate this expression, the following output waveform is displayed in the graph window. The X- and Y-axis name and units are displayed accordingly to the specified values.



Example 2

This example shows the graph plotted when you apply the `getAsciiWave` function on the following data file:

```
./data.txt
```

This text file contains the data in the following format:

Time	V(OUT)
2150000000	10

© 2023 Cadence Design Systems, Inc.

3000000000	20
4050000000	30
5055000000	40

Specify the following field values:

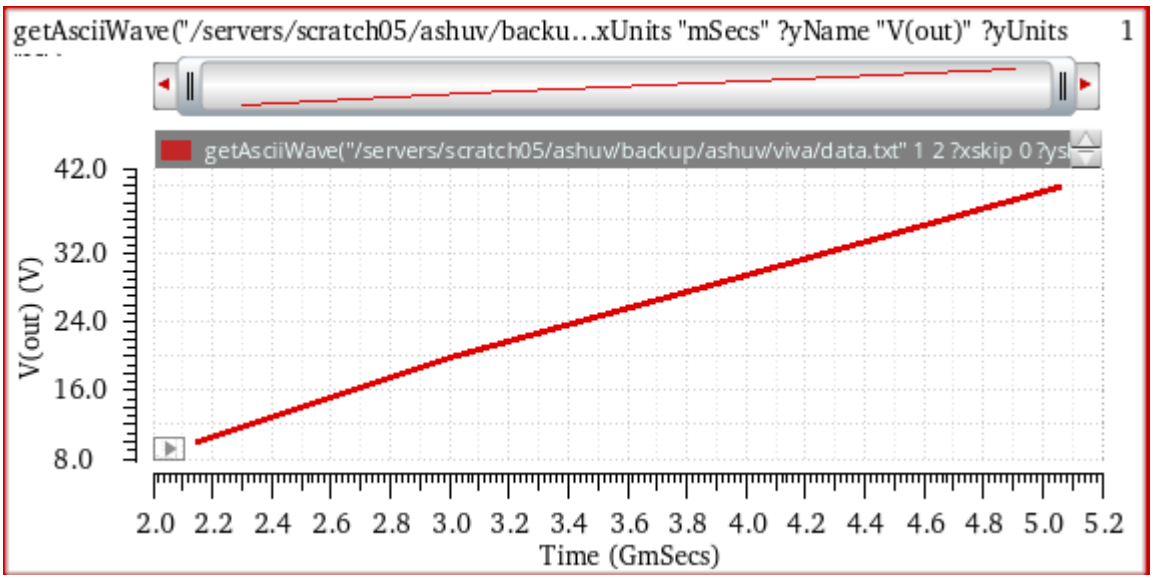
- *Data File Name*—`./data.txt`
- *X Column*—`1`
- *Y Column*—`2`
- *Skip X values*—`0`
- *Skip Y values*—`0`
- *Format Float*—`on`
- *X Name*—Time
- *X Units*—`Secs`. The X-axis name is displayed as `mSecs`.
- *Y Name*—`V(out)`
- *Y Units*—`V`

The following expression is created in the Buffer:

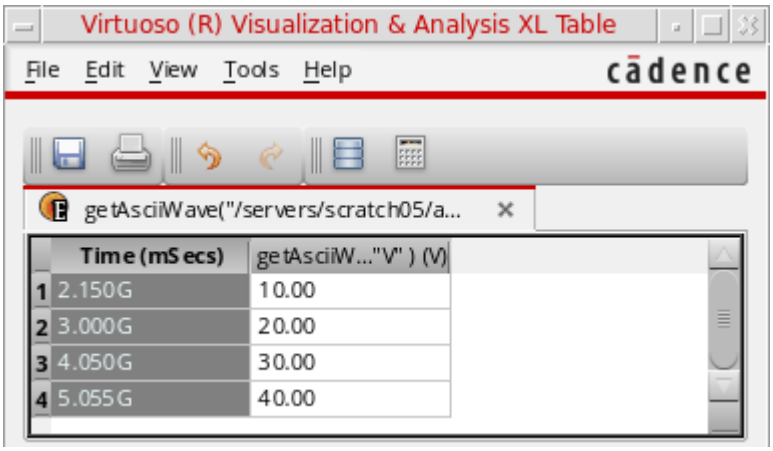
```
getAsciiWave("./data.txt" 1 2 ?xskip 0 ?yskip 0 ?formatFloat t ?xName
"Time" ?xUnits "Secs" ?yName "V(Out)" ?yUnits "V")
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform is displayed in the graph window.



Now, if you send this trace to the Virtuoso Visualization and Analysis XL Table, the following X- and Y-axis values are displayed. The integer value format has been changed to float values.



Example 3

This example shows the graph plotted when you apply the `getAsciiWave` function on the following data file that includes complex numbers:

```
./complex.txt
```


© 2023 Cadence Design Systems, Inc.

This text file contains the data in the following format:

Time	**V(OUT)**
0.25,-5.840062602800402e-08	3.407650965478429e-09
0.28,-5.839796008205289e-08	3.816413160905035e-09
0.3100000000000001,-5.839499666338922e-08	4.225122682762201e-09
0.3400000000000001,-5.839173715241753e-08	4.633774036876914e-09
0.3700000000000001,-5.838818306378015e-08	5.042361774051216e-09
0.4000000000000001,-5.838433604460925e-08	5.45088049412205e-09
0.4300000000000002,-5.838019787263262e-08	5.859324849942298e-09
0.4600000000000002,-5.837577045413842e-08	6.267689551277e-09
0.4900000000000002,-5.837105582180227e-08	6.675969368609005e-09

Specify the following field values:

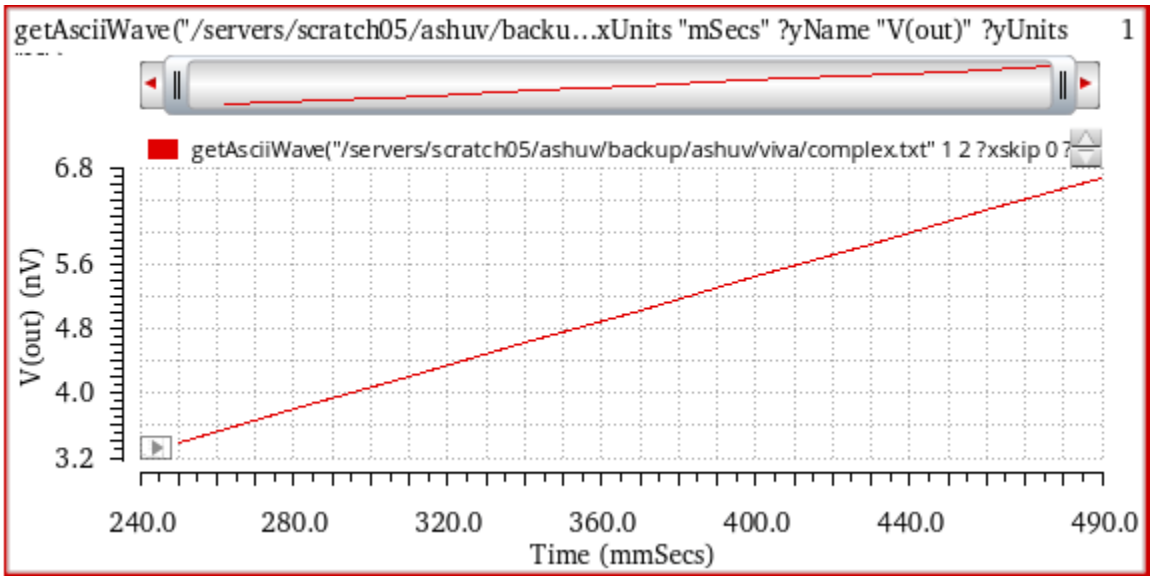
- *Data File Name*—`./complex.txt`
- *X Column*—`1`
- *Y Column*—`2`
- *Skip X values*—`0`
- *Skip Y values*—`0`
- *Format Float*—`on`
- *X Name*—`Time`
- *X Units*—`Secs`. The X-axis name is displayed as `mSecs`.
- *Y Name*—`V(out)`
- *Y Units*—`V`

The following expression is created in the Buffer:

```
getAsciiWave("./complex.txt" 1 2 ?xskip 0 ?yskip 0 ?formatFloat t
?xName "Time" ?xUnits "Secs" ?yName "V(Out)" ?yUnits "V")
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform is displayed in the graph window.



Now, if you send this trace to the Virtuoso Visualization and Analysis XL Table, the following X- and Y-axis values are displayed. The complex values have been changed to float values.

getAsciiWave("/servers/scratch05/a...

	Time (mSecs)	getAsciiW..."V") (V)
1	250.0m	3.408E-9
2	280.0m	3.816E-9
3	310.0m	4.225E-9
4	340.0m	4.634E-9
5	370.0m	5.042E-9
6	400.0m	5.451E-9
7	430.0m	5.859E-9
8	460.0m	6.268E-9
9	490.0m	6.676E-9

Related OCEAN Function

The equivalent OCEAN command for `getAsciiWave` is:

```
getAsciiWave(
     t_filename
     x_xColumn
     x_yColumn
     [ ?xskip x_x_skip ]
     [ ?yskip x_yskip ]
```


© 2023 Cadence Design Systems, Inc.

```
[ ?formatFloat g_formatFloat ]
[ ?xName t_xName ]
[ ?xUnits t_xUnits ]
[ ?yName t_yName ]
[ ?yUnits t_yUnits ]
)
=> o_wave / nil
```

For more information about the OCEAN function, see getAsciiWave in *OCEAN Reference* manual.

© 2023 Cadence Design Systems, Inc.

getData

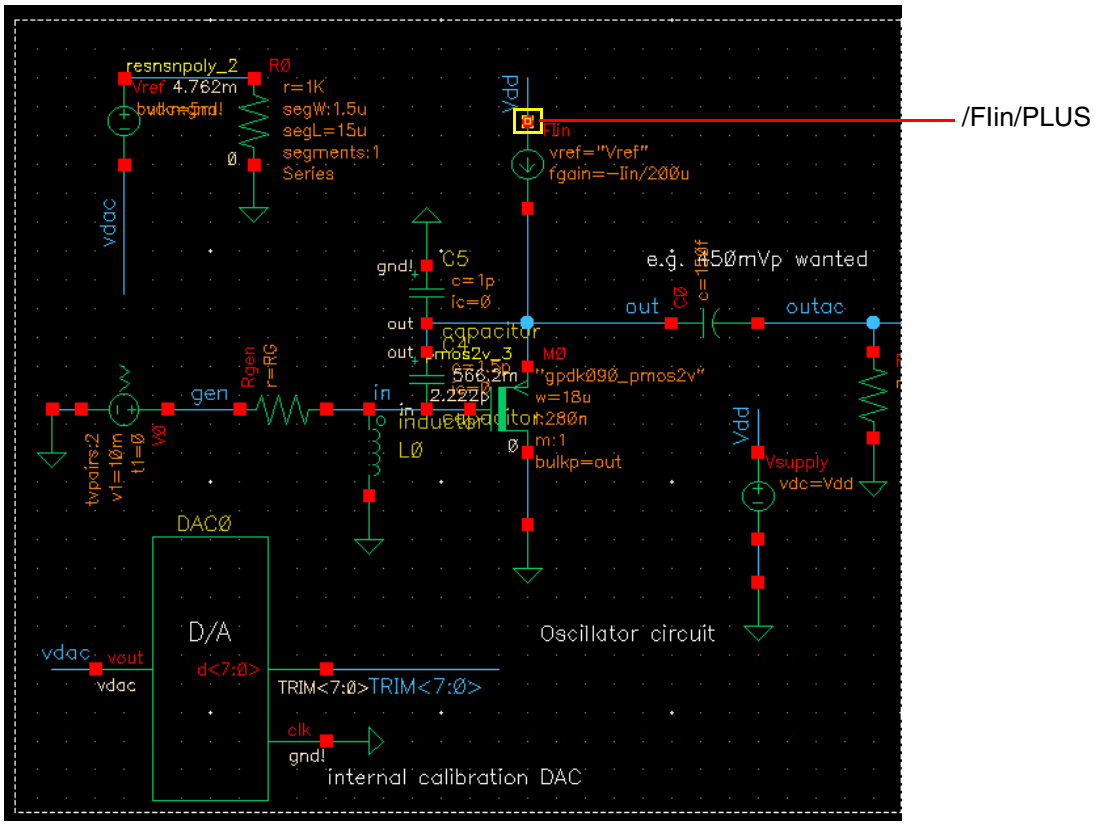
Returns the number or the waveform for the specified signal.

This function includes the following arguments:

- *Signal name*—The name of the signal.
- *Analysis Type*—The type of analysis. For example, `ac` and `tran`.

Example 1

Consider the signal, `/FIin/PLUS`, in the following design:



© 2023 Cadence Design Systems, Inc.

This example shows the voltage calculation for the signal, `/FIin/PLUS`, using the `getData` function. Specify the following arguments:

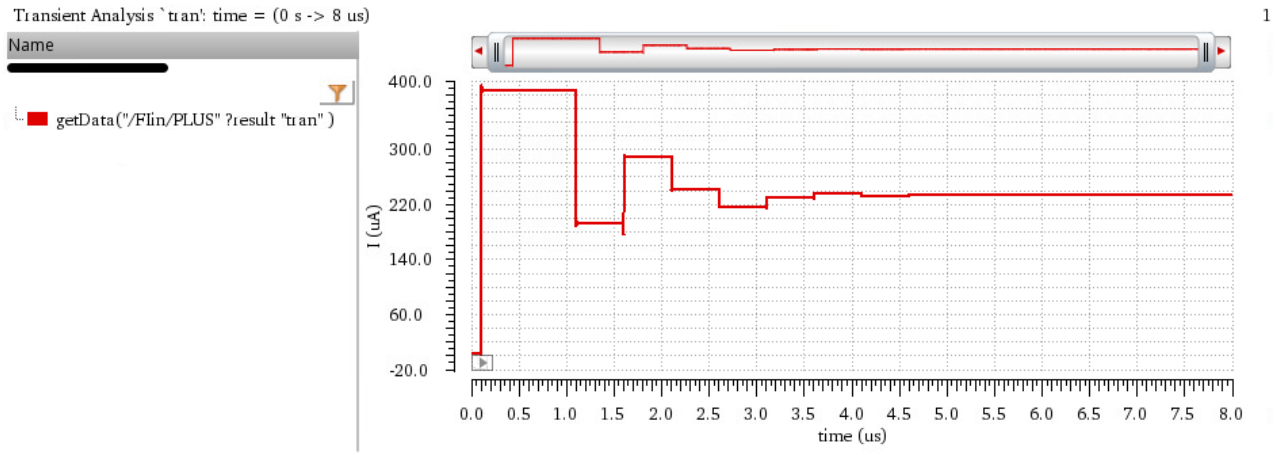
- *Signal name*—`/FIin/PLUS`
- *Analysis Type*—`tran`

Note: Before using this function, you need to set the results database context to the results database that contains the input signal, such as `/FIin/PLUS` used in this example. For more information about how to change the in-context results directory, see Setting the In-Context Results Directory.

The following expression is created in the Buffer:

```
getData("/FIin/PLUS" ?result "tran" )
```

When you evaluate this expression, the function returns the following waveform for the specified signal.



Related OCEAN Function

The equivalent OCEAN command for `getData` is:

© 2023 Cadence Design Systems, Inc.

```
getData(
     t_name
     [ ?result s_resultName [ ?resultsDir t_resultsDir ] ]
     )
     => x_number / o_waveform / nil
```

For more information about this OCEAN function, see getData in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

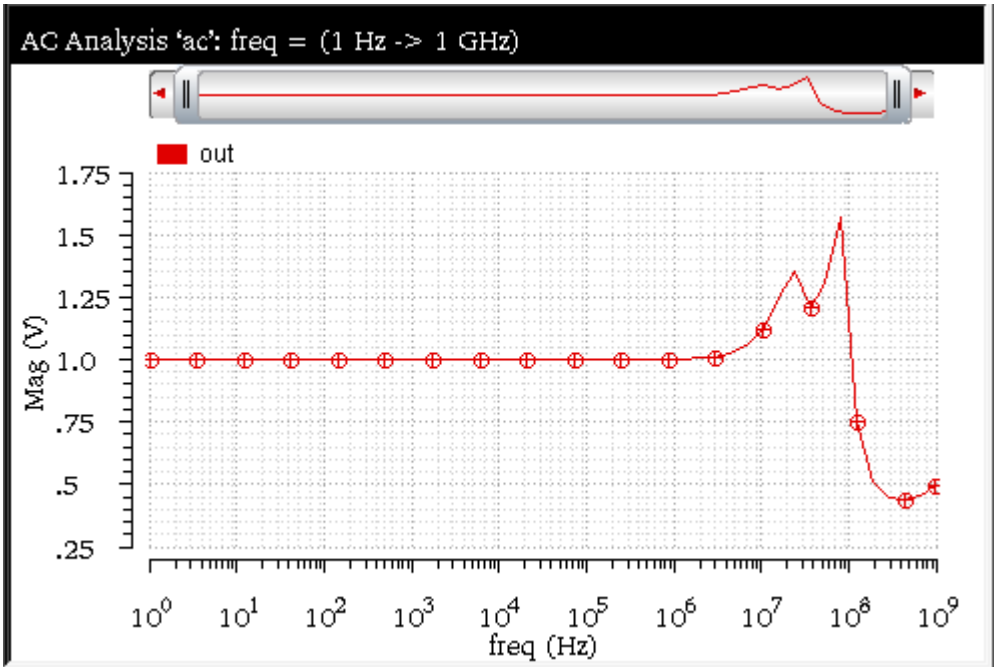
groupDelay

Computes the group delay of the expression in the buffer. Group delay is defined as the derivative of the phase with respect to frequency. Group delay is expressed in seconds. The following equation is used to calculate this function:

$$GroupDelay = \frac{d\phi}{d\omega} = \left(-\frac{d}{df}\right)\left[\frac{phase(/netX)}{360}\right]$$

Example

Consider the following input signal form the AC analysis:



When you send this signal to Calculator, the expression for this signal is displayed in the Buffer. By default, this expression is based on the magnitude of the signal and represented as shown below:

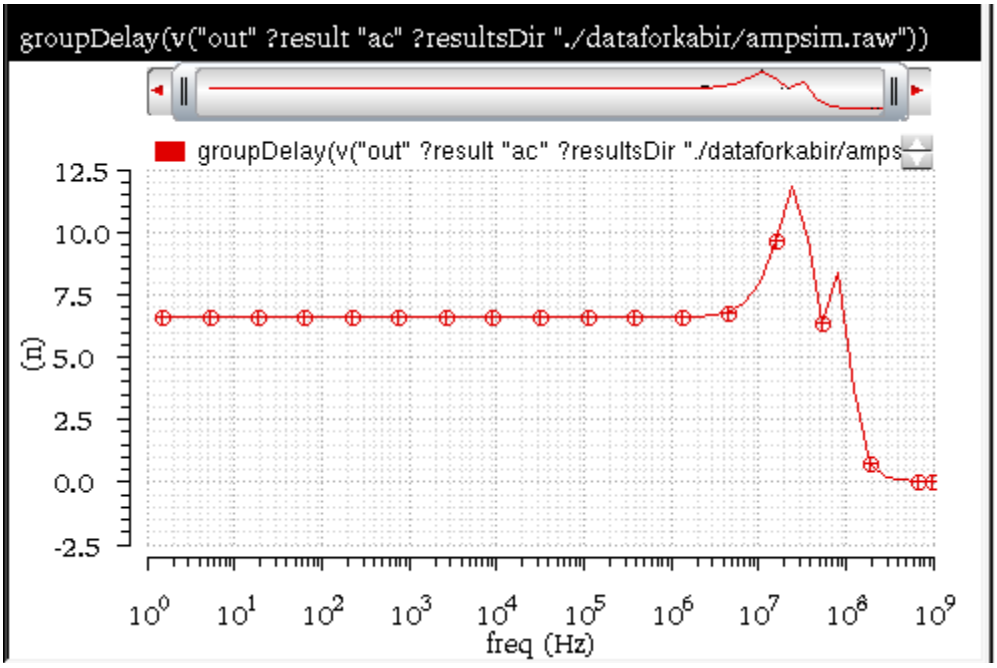
```
mag(v("out" ?result "ac" ?resultsDir "./ampsim.raw"))
```

Edit this expression to remove the magnitude and select the `groupDelay` function in the Function Panel. The function is applied to the signal expression in the Buffer. The corresponding expression created in the Buffer is as follows:

© 2023 Cadence Design Systems, Inc.

```
groupDelay(v("out" ?result "ac" ?resultsDir "./ampsim.raw"))
```

When you evaluate this expression, the group delay is calculated and the following output waveform is displayed in a new graph window.



Related OCEAN Function

The equivalent OCEAN command for `groupDelay` is:

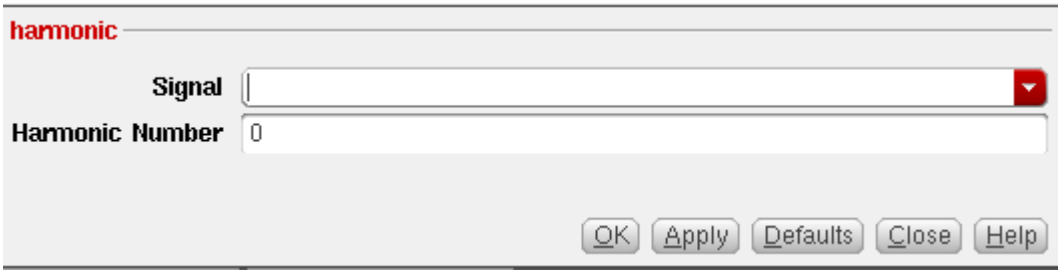
```
groupDelay( o_waveform )
     => o_waveform/nil
```

For more information, see `groupDelay` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

harmonic

Returns the harmonic waveform of the specified harmonic. This function is available only in the SKILL mode.



This function includes the following arguments:

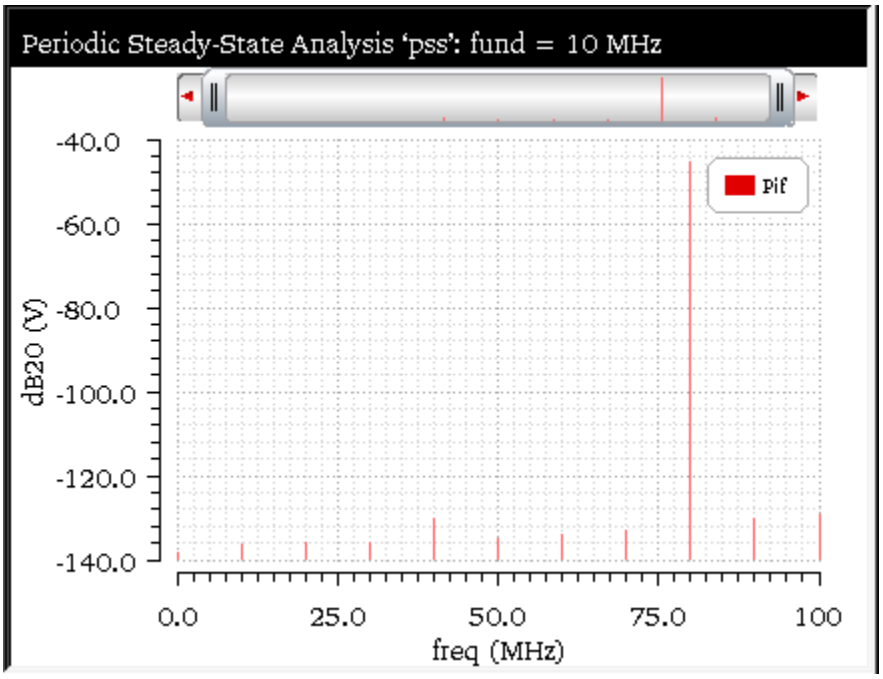
- *Signal*—Name of the signal.
- *Harmonic Number* is the index number that designates the harmonic information to be returned. For the pss, pac, and pxf analyses, the index is an integer. For the pdisto analysis, the index is a list of integers that correspond to the frequency names listed in the funds analysis parameter in the netlist. You can specify more than one harmonic number at a time.

Example

This example shows the harmonic value generated when you apply the `harmonic` function on the following input waveform:

© 2023 Cadence Design Systems, Inc.

Note: Ensure that the dependent modifier displayed on Y-axis of the input waveform is `dB20`. For more details about how to change the dependent modifier, see Plotting Traces Using Different Modifiers.



In this waveform, the dB20 of the input signal is plotted against frequency on X-axis.The following arguments are specified in this example:

- *Signal*—`db20(v("Pif" ?results "pss_fd"))`
- *Harmonic Number*—`8`

The following expression is created in the Buffer:

```
harmonic(db20(v("Pif" ?result "pss_fd" ?resultsDir "./
pss_ne600.raw")) 8 )
```

When you evaluate this expression, the following output value is displayed in the Buffer:

```
-45.46
```

Related OCEAN Function

The equivalent OCEAN function for `harmonic` is:

© 2023 Cadence Design Systems, Inc.

```
harmonic( o_waveform h_index )
     => o_waveform/g_value/nil
```

For more information about this OCEAN function, see harmonic in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

harmonicFreq

Returns the frequency(s) of the harmonic waveform for the specified harmonic. This function is available only in the SKILL mode.

This function includes the following arguments:

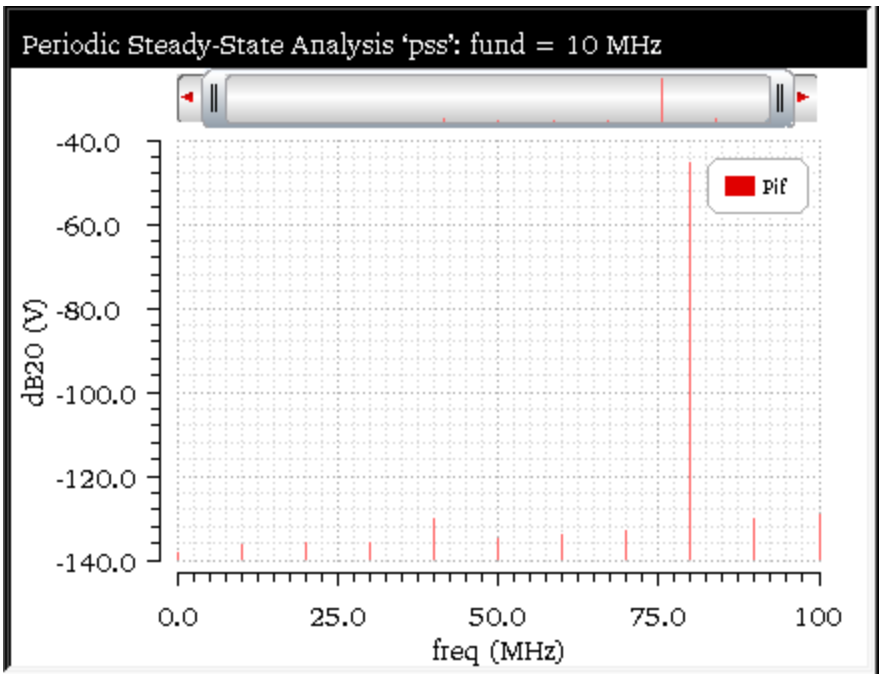
- *Signal*—Name of the signal.
- *Harmonic Number*—Index number that designates the harmonic information to be returned. For the pss, pac, and pxf analyses, the index is an integer. For the pdisto analysis, the index is a list of integers that correspond to the frequency names listed in the funds analysis parameter in the netlist. You can specify more than one harmonic number at a time.

Example

This example shows the harmonic frequency calculated when you apply the `harmonicFreq` function on the following input signal. Results Directory: `./pss_ne600.raw`

© 2023 Cadence Design Systems, Inc.

Note: Ensure that the dependent modifier displayed on Y-axis of the input waveform is `dB20`. For more details about how to change the dependent modifier, see Plotting Traces Using Different Modifiers.



The following arguments are specified in this example:

- *Signal*—`db20(v("Pif" ?result "pss_fd"))`
- *Harmonic Number*—`8`

The following expression is created in the Buffer:

```
harmonicFreq(xval(db20(v("Pif" ?result "pss_fd" ))), 8)
```

When you evaluate this expression, the following output value is displayed in the Buffer:

```
80.0E6
```

Related OCEAN Function

The equivalent OCEAN command for `harmonicFreq` is:

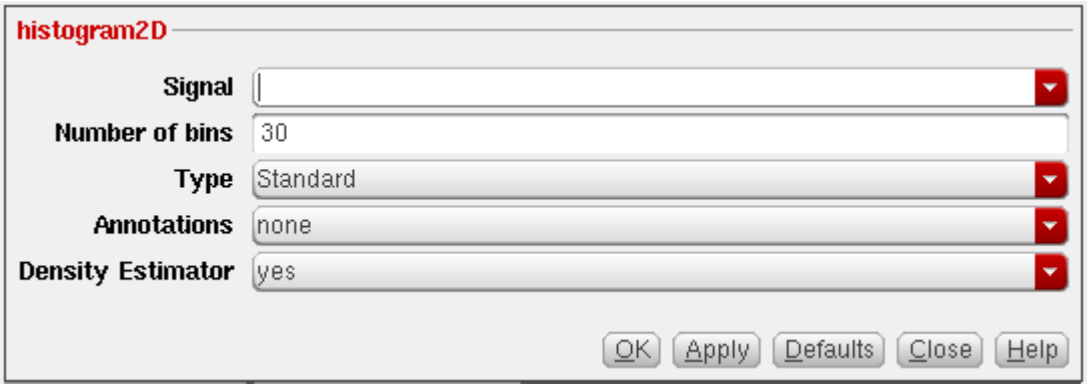
```
harmonic( xval(o_waveform) h_index )
```

For more information about this OCEAN function, see harmonic in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

histogram2D

Returns a waveform that represents the statistical distribution of input data in the form of a histogram. The height of the bars (or bins) in the histogram represents the frequency of the occurrence of values within a specific period.



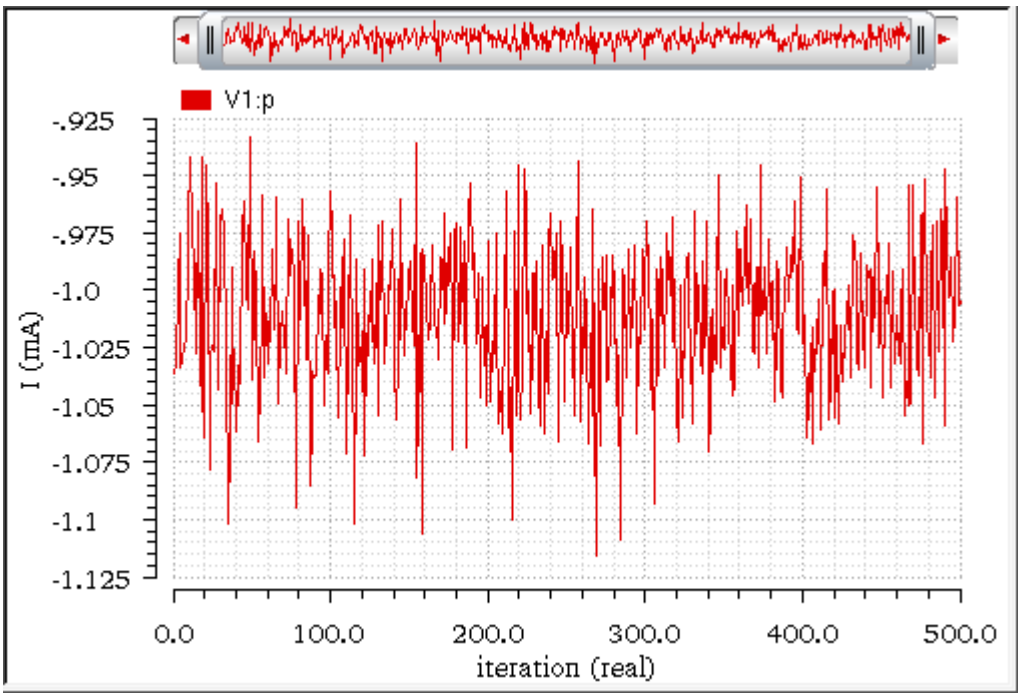
This function includes the following arguments:

- *Signal*—Name of the waveform, expression, or family of waveforms. The expression displayed in the Buffer can be added to this field by selecting `buffer`.
- *Number of bins*—Specifiy how many bars you want to plot in the resulting histogram plot.
  Default value:`10`.
- *Type*—Specify the type of histogram to be plotted.
  Valid values: `Standard` and `Cumulative box`.
  Default value: `Standard`.
- *Annotations*—Select the annotations that you want to display in the resulting histogram plot. The available options are: `none`, `Std`, `Dev Lines` (indicating the mean, (mean - standard deviation), and (mean + standard deviation) values)), `% Markers`, and `Std Dev Lines and % Markers`.
  Default value: `none`
- *Density Estimator*—Select `yes` from the drop-down if you want that the resulting histogram plot displays a curve that estimates the distribution concentration.
  Default value: no.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from Monte Carlo analysis:



The `histogram2D` function is applied on this input waveform with the following arguments:

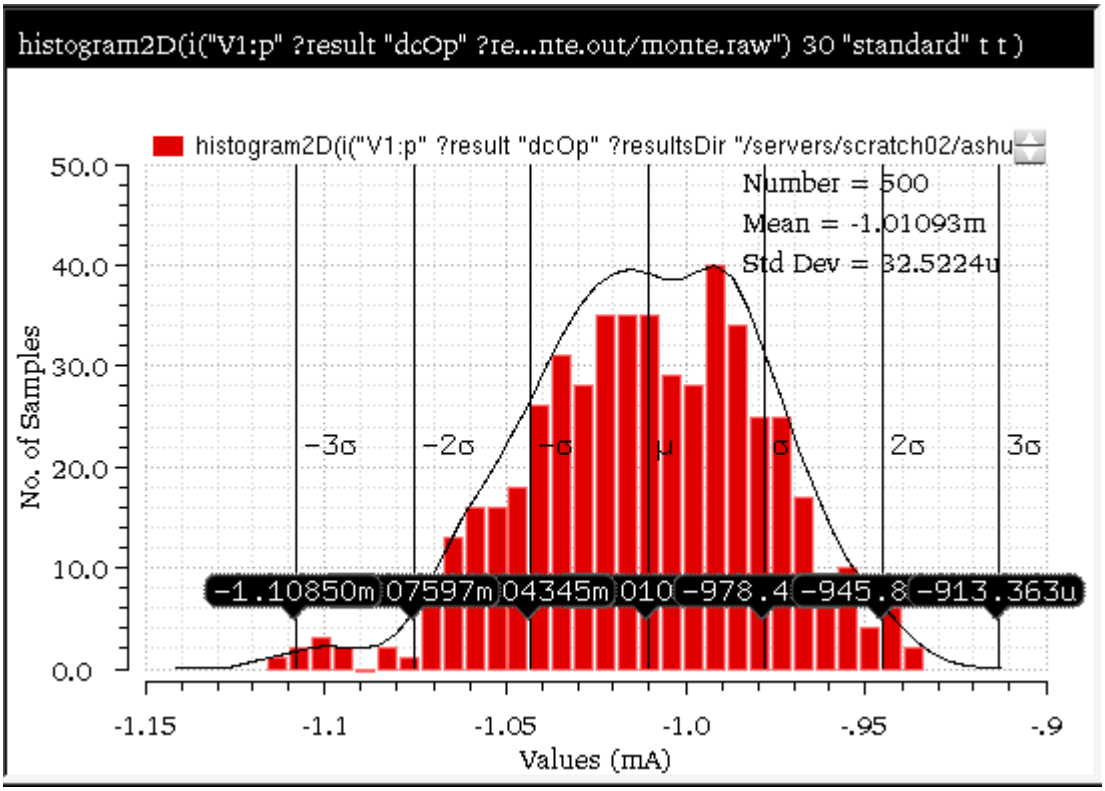
- *Signal*—`i("V1:p" ?result "dcOp" ?resultsDir "./monte/monte.out/monte.raw")`
- *Number of bins*—`30`
- *Type*—`standard`
- *Annotations*—`Std Dev Lines`
- *Density Estimator*—`yes`

The corresponding expression created in the Buffer is as follows:

```
histogram2D(i("V1:p" ?result "dcOp" ?resultsDir "./monte/monte.out/
monte.raw") 30 "standard" "stddev"t )
```

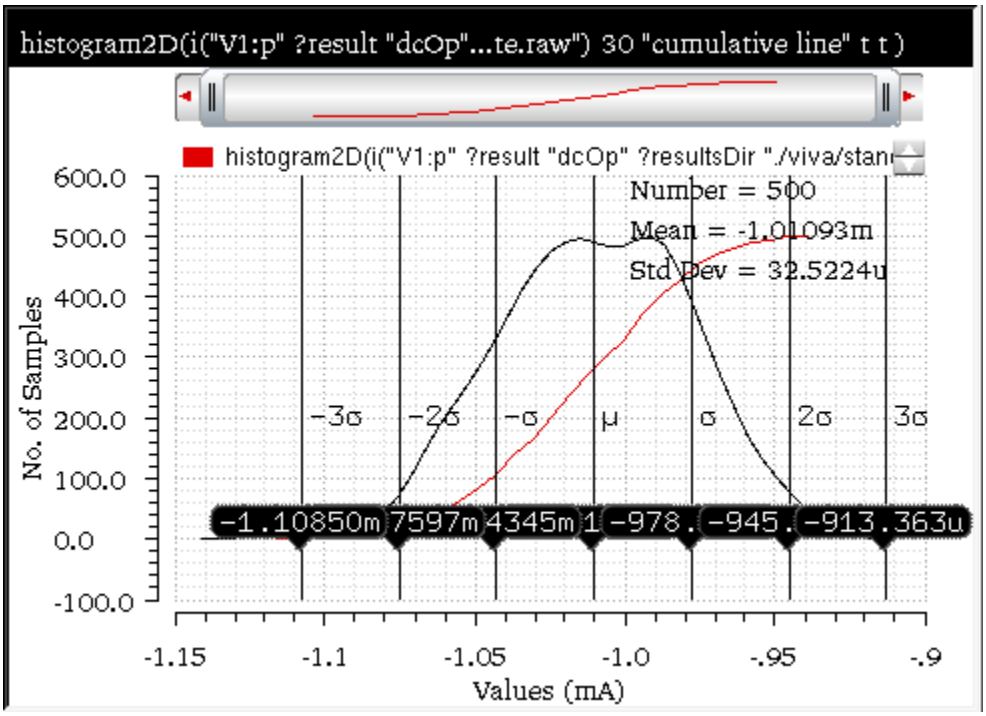

© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform representing the data in histogram is displayed in a new graph window.

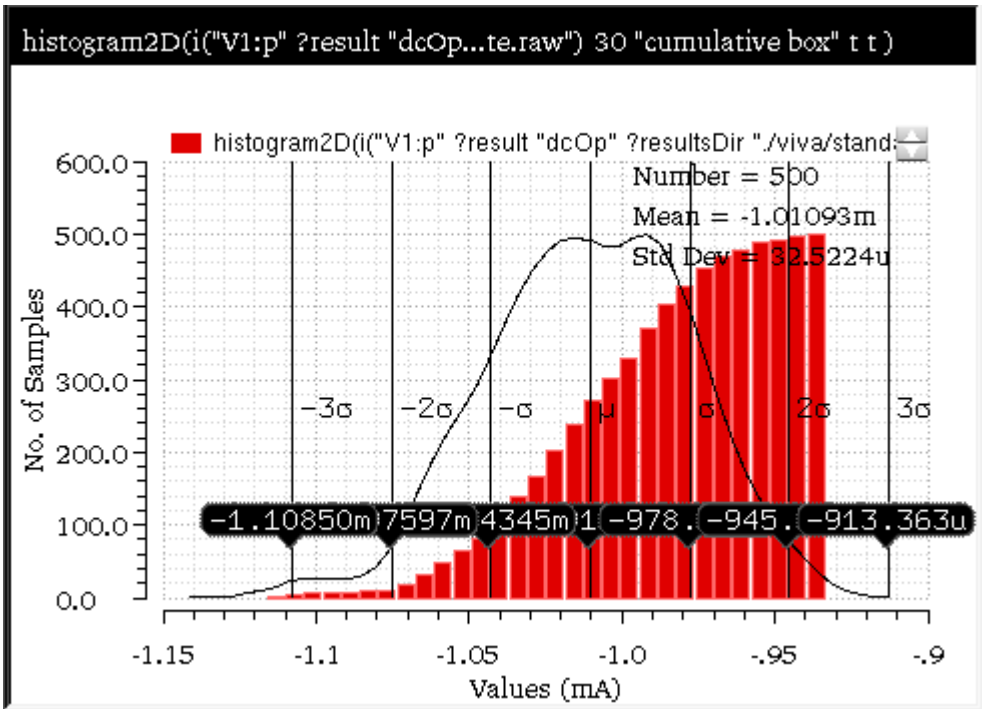


© 2023 Cadence Design Systems, Inc.

The following figure shows the output generated if you select *Type* as `Comulative line`:



The following figure shows the output generated if you select *Type* as `Comulative box`:



© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN command for `histogram2D` is:

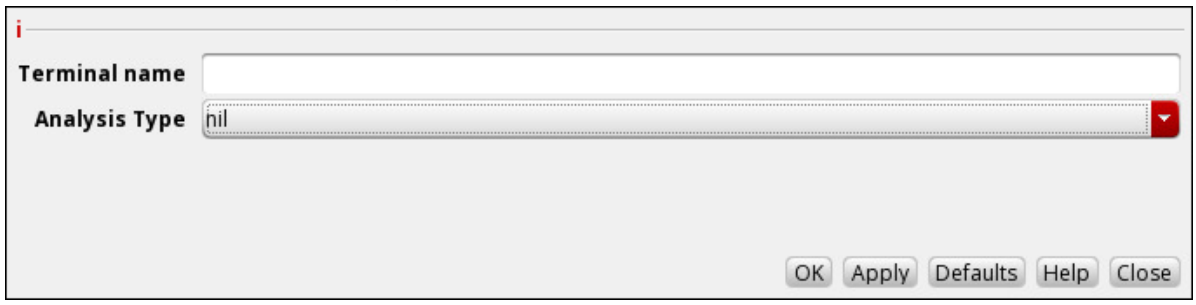
```
histogram2D(o_waveform x_nbins s_type g_setAnnotation g_setDensityEstimator)
     => o_waveform/nil
```

For more information, see `histogram2D` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

i

Calculates the current through the specified terminal.

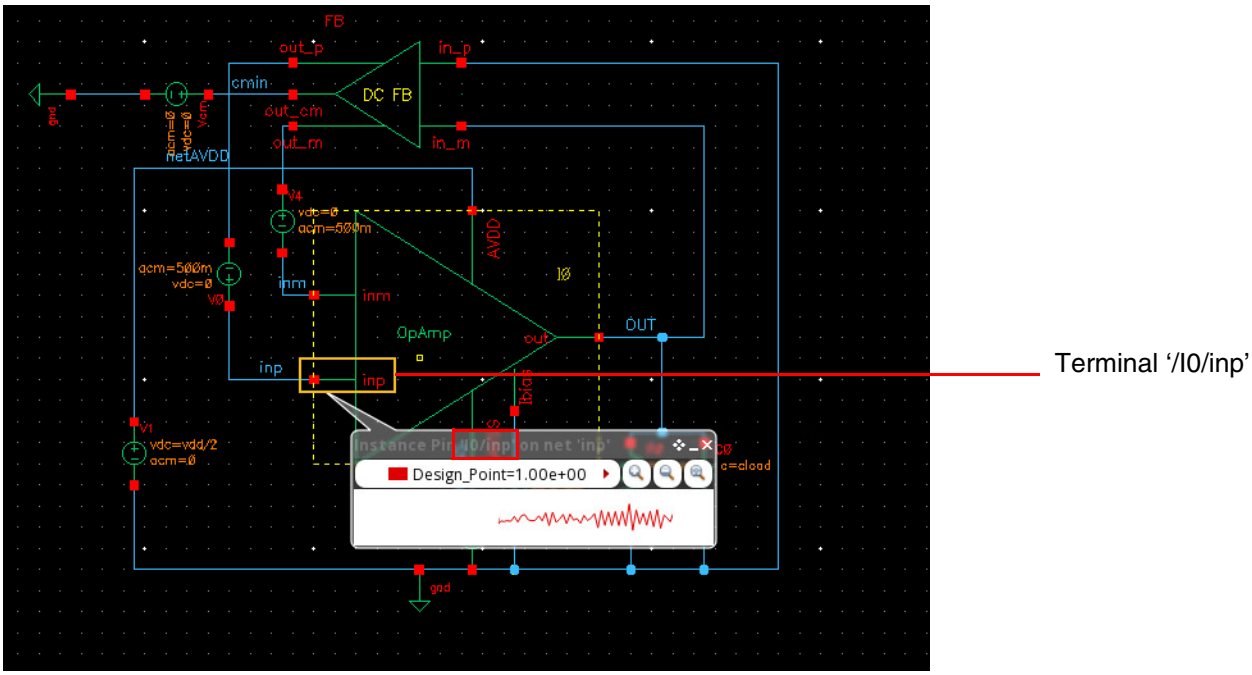


This function includes the following arguments:

- *Terminal name*—The name of the terminal.
- *Analysis Type*—The type of analysis. For example, `ac` and `tran`.

Example

Consider the terminal, `/I0/inp`, in the following design.



This example shows the current calculation for the terminal, `/I0/inp`, using the `i` function.

Specify the following arguments:

© 2023 Cadence Design Systems, Inc.

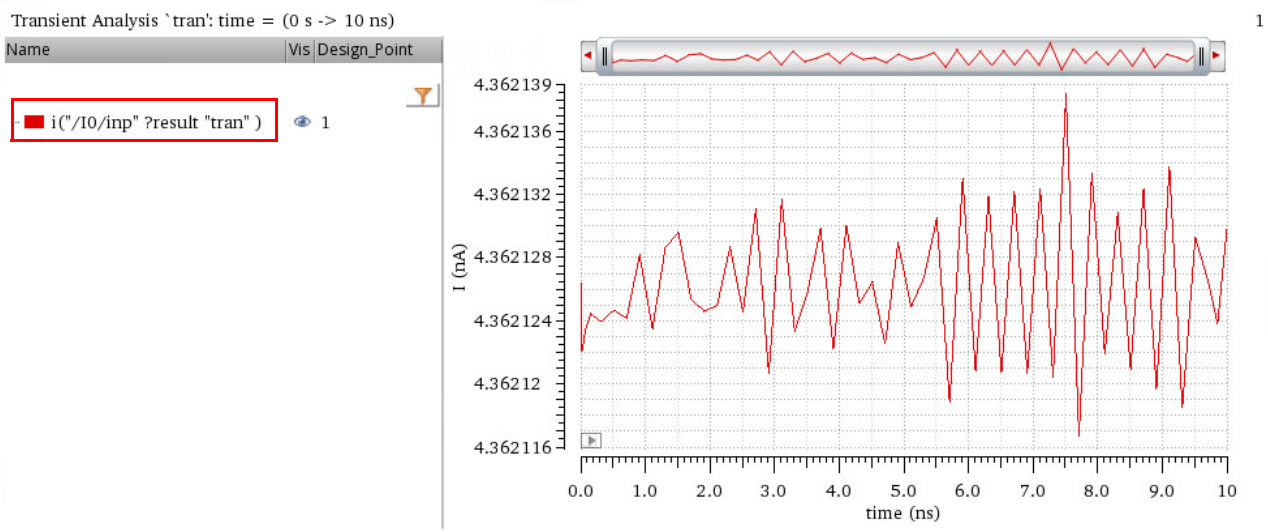
- *Terminal name*—`/I0/inp`
- *Analysis Type*—`tran`

Note: Before using this function, you need to set the results database context to the results database that contains the input terminal, such as `/I0/inp` used in this example. For more information about how to change the in-context results directory, see Setting the In-Context Results Directory.

The following expression is created in the Buffer:

```
i("/I0/inp" ?result "tran" )
```

When you evaluate this expression, the function returns the following waveform, which displays the variation of current values with time through the specified terminal.



Related OCEAN Function

The equivalent OCEAN command for `i` is:

© 2023 Cadence Design Systems, Inc.

```
i (
     t_component
     [ ?result s_resultName ]
     [ ?resultsDir t_resultsDir ]
     )
     => o_waveform / nil
```

For more information about this OCEAN function, see i in *OCEAN Reference*.

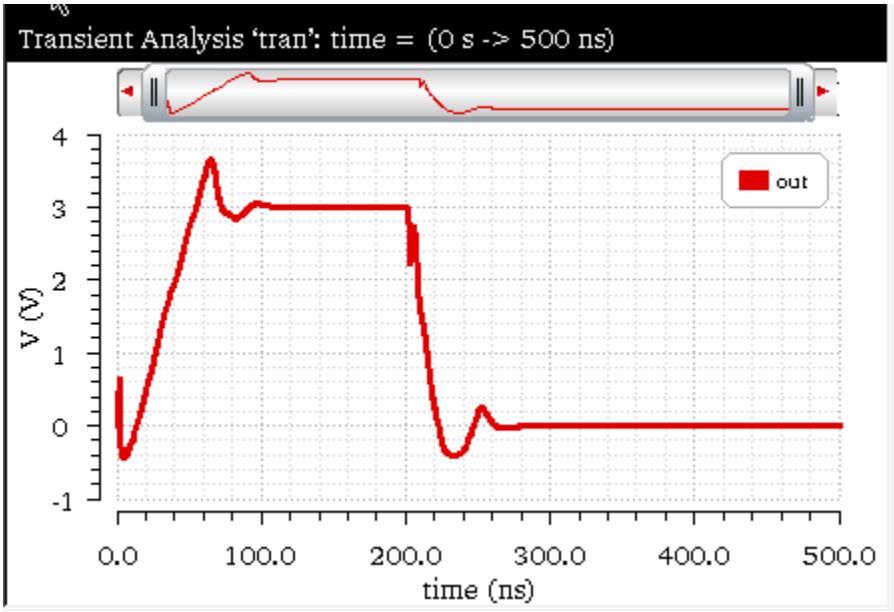
© 2023 Cadence Design Systems, Inc.

iinteg

Returns the incremental area under the waveform.

Example

This example calculates the incremental area under the following input waveform:

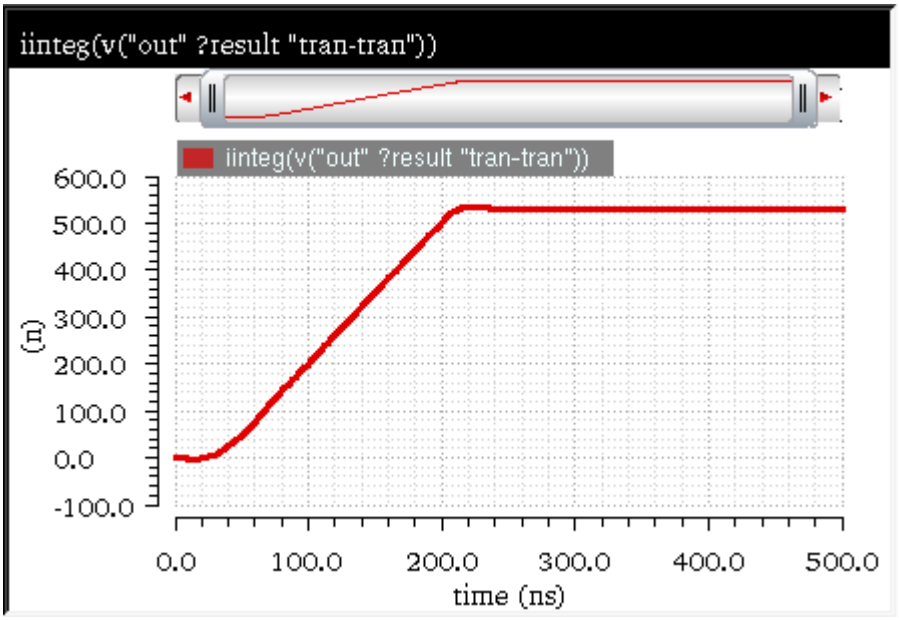


When you apply the `iinteg` function on this waveform, the following expression is created in the Buffer:

```
iinteg(v("out" ?result "tran-tran" ?resultsDir "ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform appears:



Related OCEAN Function

The equivalent OCEAN command for `iinteg` is:

```
iinteg( o_waveform )
     => o_waveform/nil
```

For more information about this OCEAN command, see iinteg in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

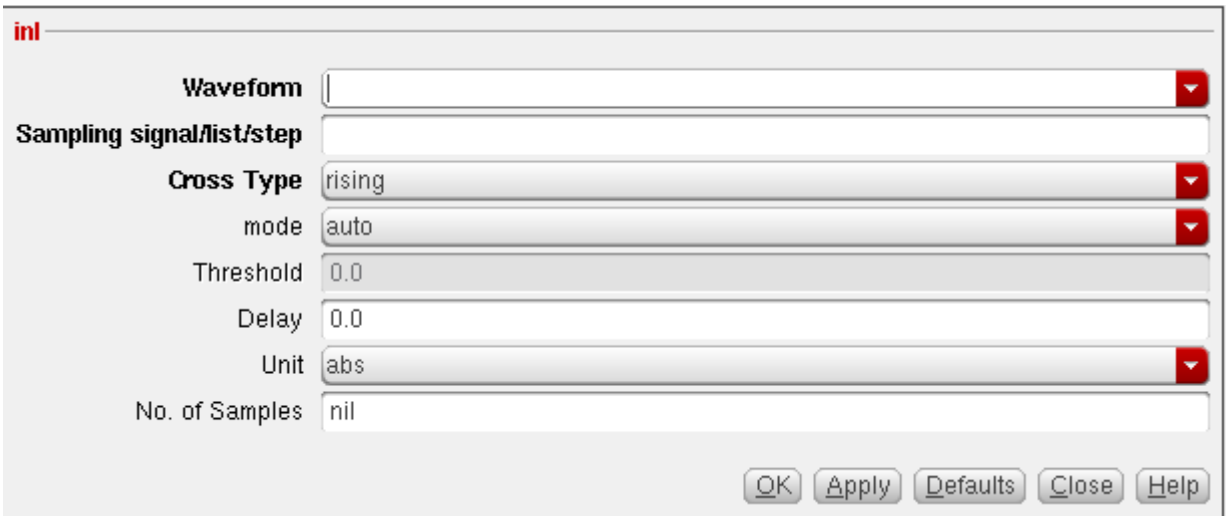
inl

Computes the integral non-linearity of a transient simple or parametric waveform. INL can be calculated by using the following expression:

$$INL(i) = DNL(i) + INL(i-1)$$

The following table shows how the value of INL varies with that of DNL.

i	Voltage	$INL_{(i-1)}$	$DNL_{(i)}$	$INL_{(i)}=DNL_{(i)}+INL_{(i-1)}$
1	3.269	0	-1.671E-3	-1.671E-3
2	3.283	-1.671E-3	11.70E-3	10.03E-3
3	3.282	10.03E-3	-1.718E-3	8.314E-3
4	3.296	8.314E-3	11.73E-3	20.04E-3



This function includes the following arguments:

- *Waveform*—Name of the signal.
- *Samplingsignal/list/step*—Signal used to obtain the points for sampling the *Waveform (*points at which the waveform crosses the threshold while either `rising` or `falling` as specified in the *cross Type* field with the *delay* added to them), list of domain values at which the sample points are obtained from the *Waveform,* or the sampling interval.

© 2023 Cadence Design Systems, Inc.

- *Cross Type*—Specifies the points at which the curves of the waveform intersect with the threshold. While intersecting, the curve may be either rising (`rising`) or falling (`falling`).
- *mode*—Specifies whether the threshold value is to be calculated by Virtuoso Visualization and Analysis XL (`auto`) or specified by you (`user`). The auto threshold is calculated as:

  Auto Threshold Value = integral of the waveform divided by the X range
- *Threshold*—Threshold value against which the frequency is to be calculated. You need to specify the threshold value only if the *mode* is `auto`
- *Delay*—Delay time after which the sampling begins.
- *Unit*—Specifies whether the output waveform is to be output as an absolute value (`abs`) or multiples of least significant bit (`lsb`).
- *No. of Samples*—Specifies the samples used for calculating the non-linearity. If not specified, the samples are taken against the entire data window.

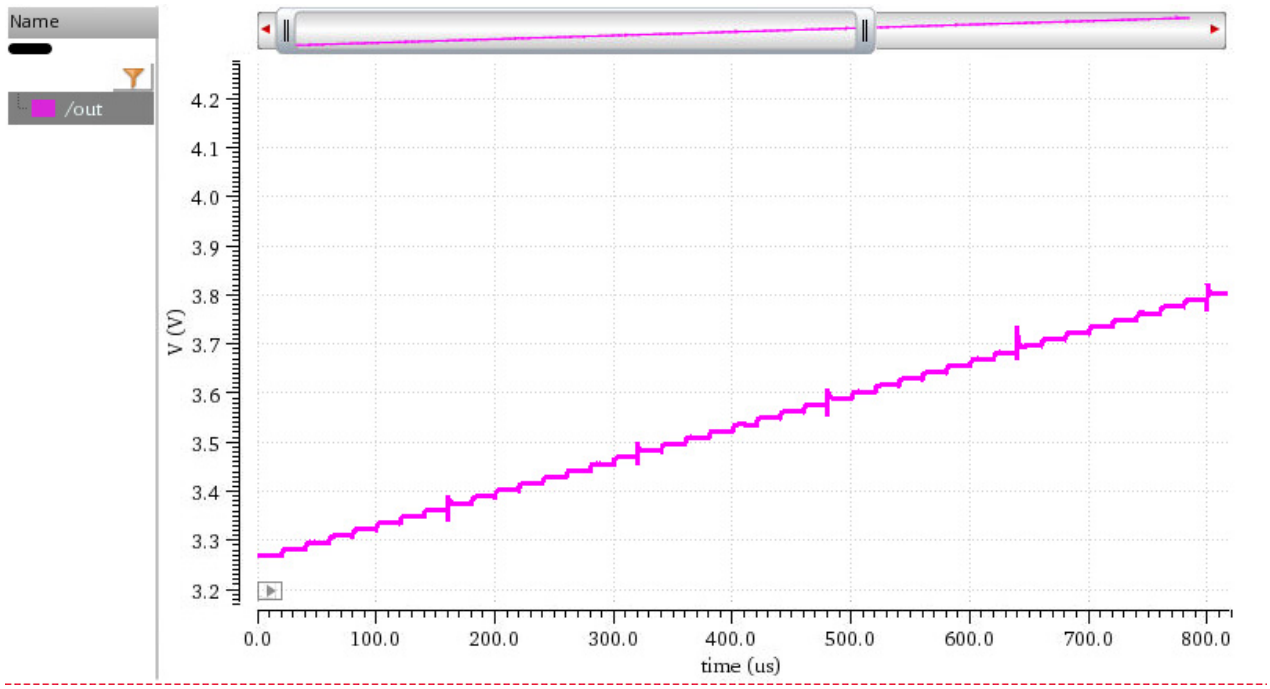
Note: For each of the three ways in which the sample points can be specified, only a few of the other optional arguments are meaningful,

- For *Sampling signal*, the fields *Cross Type*, *mode*, *Threshold*, *Delay*, *Method*, and *Units* are meaningful.
- For *list*, the fields *Method* and *Units* are meaningful.
- For *step*, the fields *Method*, *Units*, and *No. of samples* are meaningful.

© 2023 Cadence Design Systems, Inc.

Example

This example shows the output waveform generated when you apply the `inl` function on the following input waveform:



The following arguments are specified in this example:

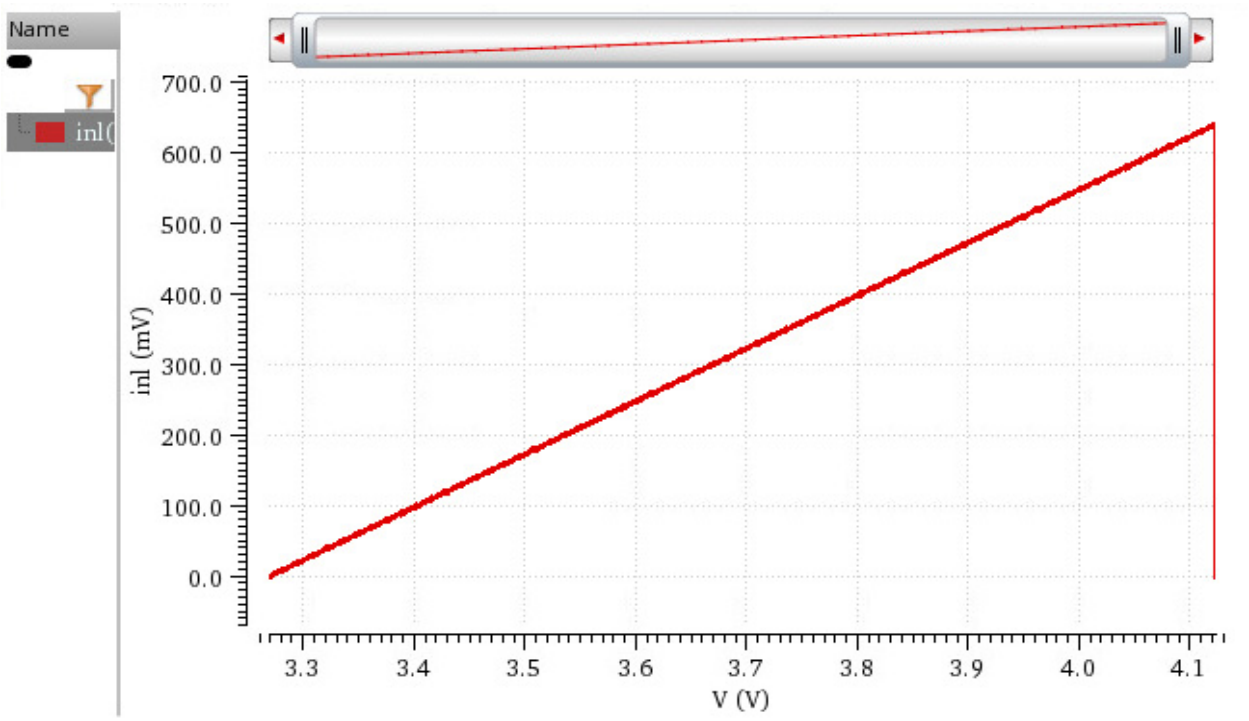
- *Waveform*—`v("/out" ?result "tran")`
- *Sampling signal/list/step*—`1e-05`
- *Cross Type*—`rising`
- *mode*—`auto`
- *Threshold*—`0.0`
- *Delay*—`4.5e-06`
- *Unit*—`abs`
- *No. of Samples*—`512`

The following expression is created in the Buffer with the specified argument

```
inl(v("/out" ?result "tran") 1e-05 ?mode "auto" ?crossType "rising" ?delay 4.5e-06
?units "abs" ?nbsamples 512)
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluated this expression, the following output waveform is displayed in the graph window:



Note: Since `inl` is calculated from the end-point `dnl`, the initial and final values are `0`.

Related OCEAN Function

The equivalent OCEAN command for `inl` is:

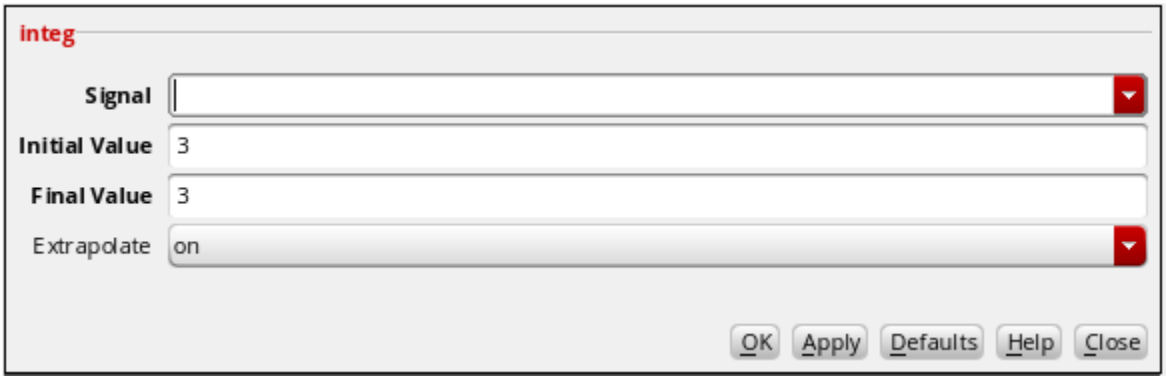
```
inl( o_dacSignal o_sample|o_pointList|n_interval
     [?mode t_mode] [?threshold n_threshold] [?crossType t_crossType]
     [?delay f_delay] [?units x_units] [?nbsamples n_nbsamples] )
     => n_inl/nil
```

For more information about this OCEAN function, see inl in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

integ

Returns the area bounded under the curve. Computes the definite integral of the waveform with respect to a range specified on the X-axis of the waveform. The result is the value of the area under the curve over the range specified on the X-axis.



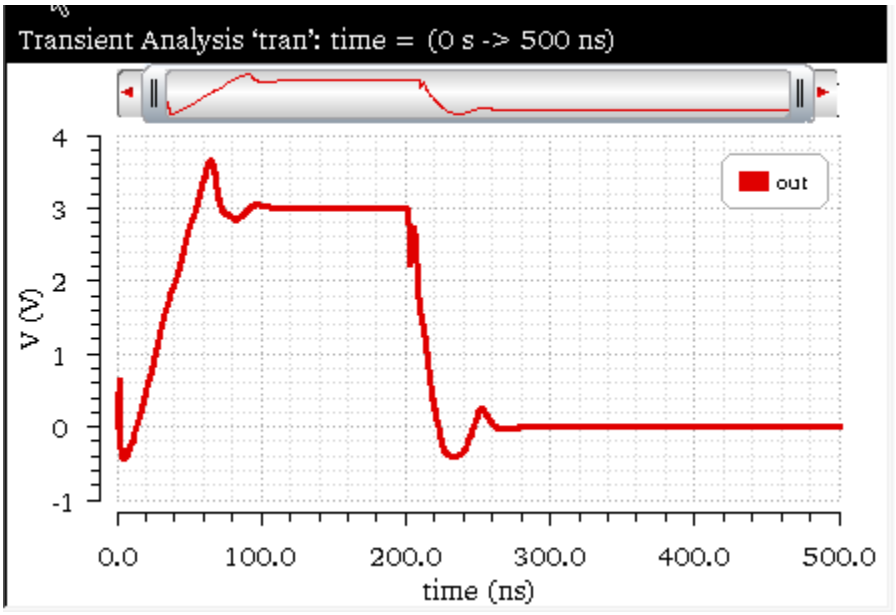
This function includes the following arguments:

- *Signal*—Name of the signal.
- *Initial Value*—Initial Value is the X start point of area.
- *Final Value*—Final Value is the X end point of area.
- *Extrapolate*—When this argument is set to `on`, extrapolation takes place, which is a process of taking data values at points x1, ..., xn and approximating a value outside the range of the given points. When this argument is set `off`, and initial or final values are outside of the range of the given waveform, the `integ` function returns `nil` and displays a warning message in CIW.

© 2023 Cadence Design Systems, Inc.

Example

This example shows the definite integral value calculated when you apply the `integ` function on the following input waveform:



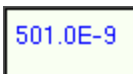
The following arguments are specified in this example:

- *Signal*—`v("out" ?result "tran")`
- *Initial Value*—`0`
- *Final Value*—`200n`
- *Extrapolate*—`no`

The following expression is created in the Buffer:

```
integ(v("out" ?result "tran-tran" ?results "./ampsim.raw") 0 200n "")
```

When you evaluate this expression, the following result value is displayed in the Buffer:



Related OCEAN Function

The equivalent OCEAN command for `integ` is:

© 2023 Cadence Design Systems, Inc.

```
integ( o_waveform,[n_intial_limit,n_final_limit] )
     => o_waveform/n_value/nil
```

For more information about this OCEAN function, see integ in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

intersect

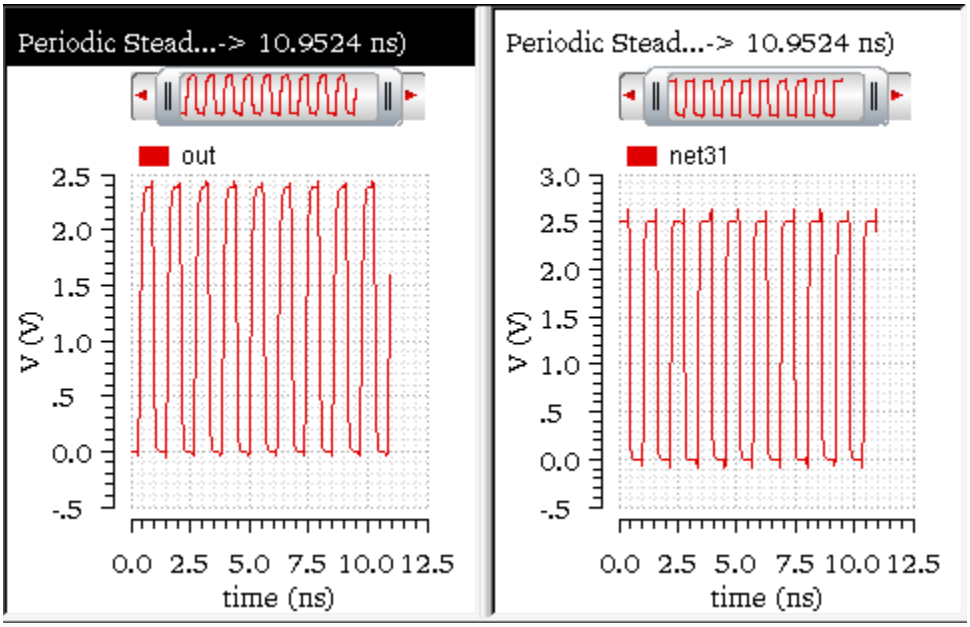
Returns all the points at which two waveforms intersect each other. The intersect function can be used on families of traces swept on the same parameter names and values.

This function includes the following arguments:

- *Signal1*—Name of the first waveform.
- *Signal2*—Name of the second waveform.

Example

This example shows the output waveform generated when you apply the `intersect` function on the following input waveforms:



The following arguments are specified in this example:

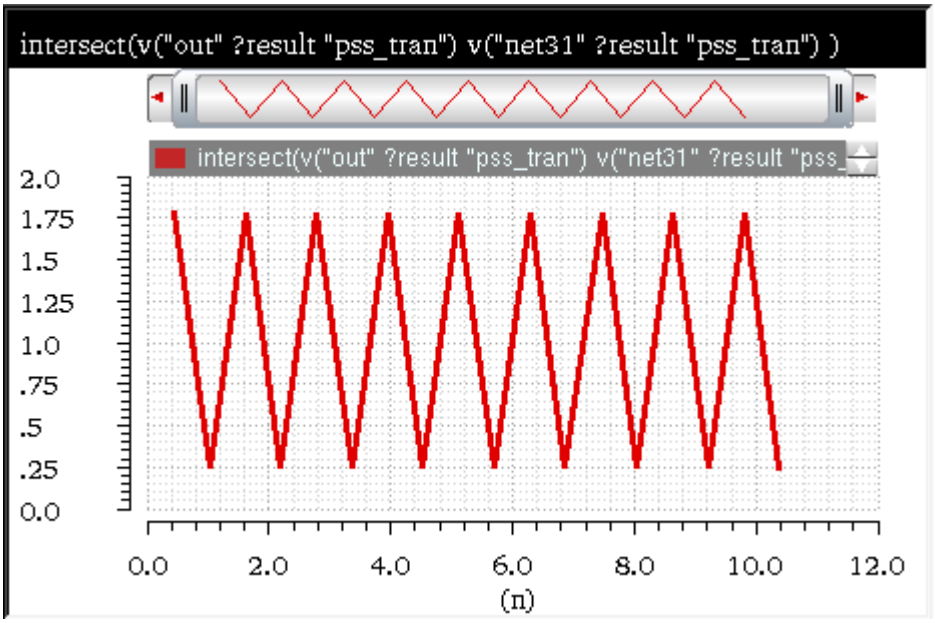
© 2023 Cadence Design Systems, Inc.

- *Signal1*—`v("out" ?result "pss_tran")`
- *Signal2*—`v("net31" ?result "pss_tran")`

The following expression is created in the Buffer:

```
intersect(v("out" ?result "pss_tran" ?resultsDir "osc13.raw")
v("net31" ?result "pss_tran" ?resultsDir "osc13.raw"))
```

Now when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `intersect` is:

```
intersect( o_waveform1 o_waveform2 )
     => o_wave/nil
```

For more information about this OCEAN function, see intersect in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

ipn

Plots the *N*th order intercept between two harmonics of a waveform that you define. This function is available only in the SKILL mode.

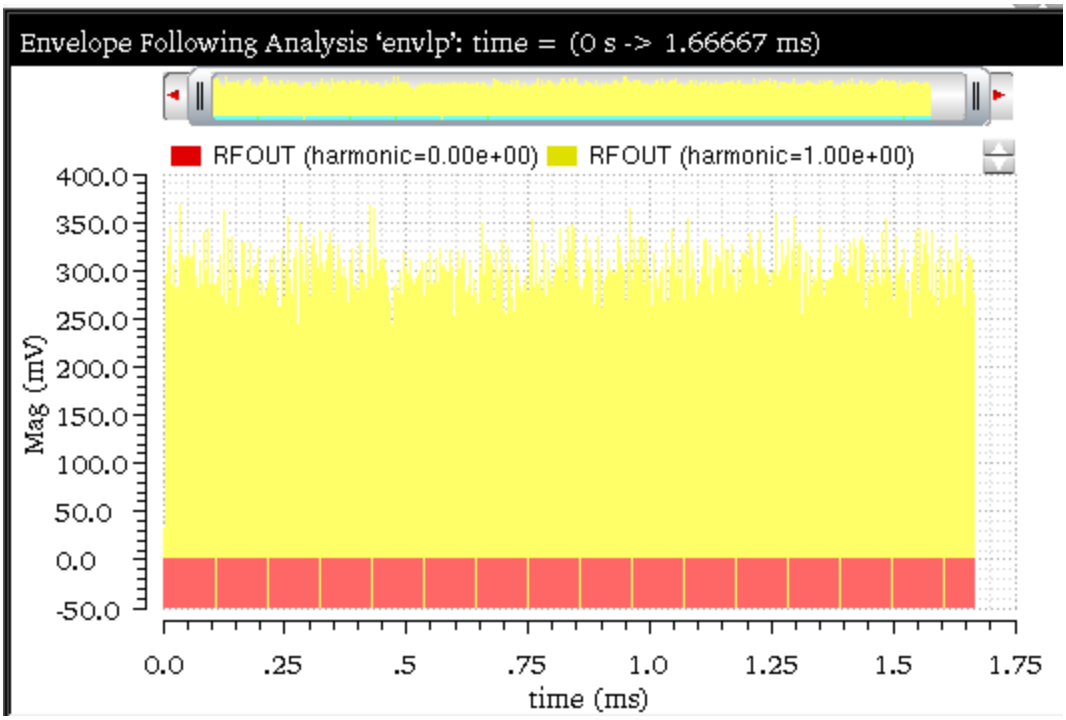
This function includes the following arguments:

- *Signal*—Name of the signal.
- *Spur Order*—Determines what order of interference is calculated for the spurious and reference waves. The default value `3` corresponds to the IP3 function. If you use a value other than 3, that order of interference is calculated between those two waves.
- *Spur Harmonic*—Harmonic number for the spurious waveform.
- *Extrapolation Point*—Extrapolation point for the IPN function. This is the X-axis value.
- *Reference Harmonic*—Harmonic number for the reference waveform.

© 2023 Cadence Design Systems, Inc.

Example

This example shows the output generated when you apply the `ipn` function on the following input signal:



The following arguments are specified in this example:

- *Signal*—`(v("/RFOUT" ?result "envlp_fd"))`
- *Spur Order*—`3`
- *Spur Harmonic*— `-2`
- *Extrapolation Point*— `-50`
- *Reference Harmonic*—`0`

The following expression is created in the Buffer:

```
ipn(dB20(harmonic(v("/RFOUT" ?result "envlp_fd" ?resultsDir "./
rfworkshop/simulation/EF_example_envlp/spectre/schematic/psf"),-
2)), dB20(harmonic(v("/RFOUT" ?result "envlp_fd" ?resultsDir "./
rfworkshop/simulation/EF_example_envlp/spectre/schematic/psf"),
0)), 3, 1, -50, -50)
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output value is displayed in the Buffer:

Related OCEAN Function

The equivalent OCEAN function for `ipn` is:

```
ipn( o_spurious o_reference
     [ f_ordspur f_ordref f_epspur f_epref g_psweep s_measure ] )
     => o_waveform/f_number/nil
```

For more information about this OCEAN function, see ipn in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

ipnVRI

Performs an intermodulation *N*th-order intercept point measurement

You can use this function to simplify the declaration of an IPN measurement. This function extracts the spurious and reference harmonics from the input waveform(s), and uses dBm(spectralPower((i or v/r),v)) to calculate the respective powers. The function then passes these power curves or numbers and the remaining arguments to the IPN function to complete the measurement.

From each of the spurious and reference power waveforms (or points), the IPN function extrapolates a line of constant slope (dB/dB) according to the specified order and input power level. These lines represent constant small-signal power gain (ideal gain). The IPN function calculates the intersection of these two lines and returns the value of either the x coordinate (input referred) or y coordinate (output referred).

This function includes the following arguments:

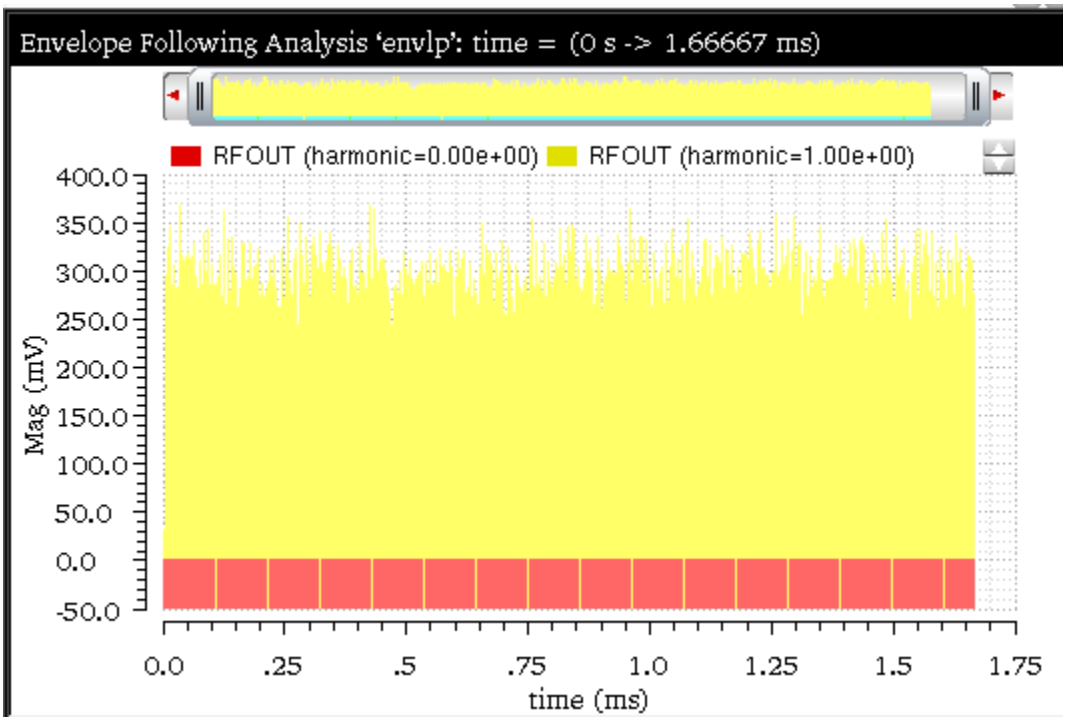
- *Signal*—Name of the signal.
- *Spur Harmonic*—Harmonic index for the spurious waveform.
- *Reference Harmonic*—Harmonic index for the reference waveform.
- *Spur Order*—Determines what order of interference is calculated for the spurious and reference waves. The default value `3` corresponds to the IP3 function. If you use a value other than 3, that order of interference is calculated between those two waves.
- *Extrapolation Point*—Extrapolation point for the `ipn` function. This is the X-axis value. The default is the minimum X-axis value of the input voltage waveform.

© 2023 Cadence Design Systems, Inc.

- *Load Resistance*—Resistance into the output port.
- *Measurement Type*—Select the *Input Referred IPN* option to get the X-coordinate of the intercept. Select *Output Referred IPN* to get the Y-coordinate of the intercept.
- *Circuit Input Power*—Specifies whether the input power is a variable sweep or a single point.

Example

This example shows the output generated when you apply the `ipnVRI` function on the following input signal:



The following arguments are specified in this example:

- Signal—`v("/RFOUT" ?result "envlp_fd")`
- *Spur Harmonic*—`-2`
- *Reference Harmonic*—`0`
- *Spur Order*— `3`
- *Extrapolation Point*—`-50`
- *Load Resistance*—`50`

© 2023 Cadence Design Systems, Inc.

- *Measurement Type*—`Input Referred IPN`
- *Circuit Input Power is:*—`Variable Sweep`

The following expression is created in the Buffer:

```
ipnVRI(v("/RFOUT" ?result "envlp_fd" ?resultsDir "./
EF_example_envlp/spectre/schematic/psf") -2 0 ?ordspur 3 ?epoint -50
?rport 50 )
```

Now, when you evaluate this expression, the following output value is dispalyed in the Buffer:

Related OCEAN Function

The equivalent OCEAN command for `ipnVRI` is:

```
ipnVRI( o_vport x_harmspur x_harmref
     [?iport o_iport] [?rport f_rport] [?ordspur f_ordspur]
     [?epoint f_epoint] [?psweep g_psweep] [?epref f_epref]
     [?ordref f_ordref] [?measure s_measure] )
     => o_waveform/f_number/nil
```

For more information about this OCEAN function, see ipnVRI in *OCEAN Reference*.

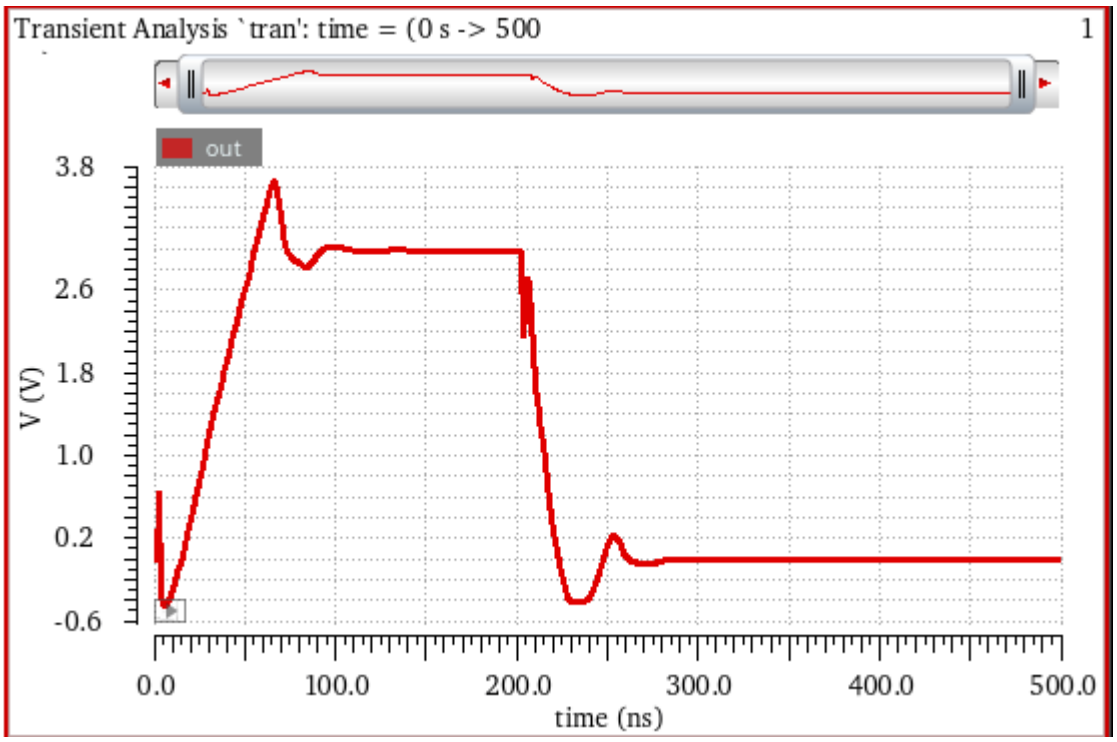
© 2023 Cadence Design Systems, Inc.

lastVal

Returns the last value at which the waveform ends on the X-axis.

Example

Consider the below example in which you run the `lastVal` function on the following transient signal:



When you send this signal to Calculator, the following expression is created in the Buffer:

```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you select the `lastVal` function in Function Panel, it is applied directly on the expression in the Buffer:

```
lastVal(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```

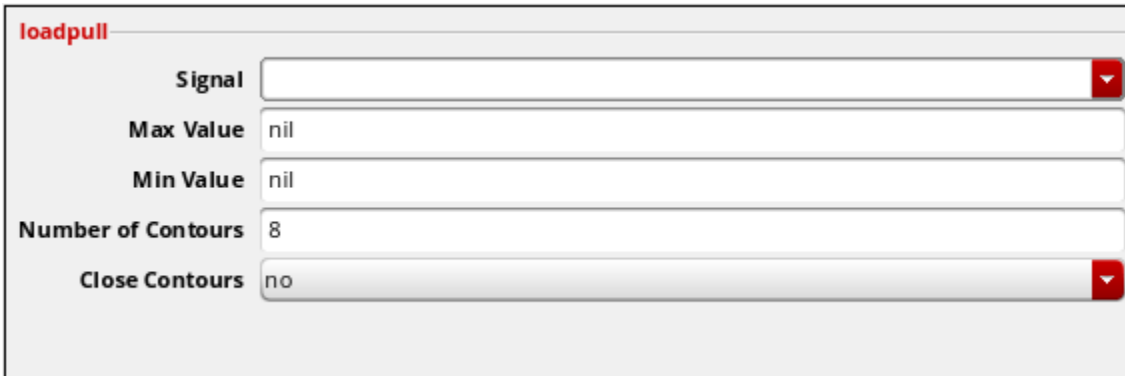
Now, when you evaluate this expression, the function returns the end value of the `out` signal on the X-axis, as shown in the figure below:

500.0n

© 2023 Cadence Design Systems, Inc.

loadpull

Plots load pull contour for the given waveform of PSS analysis. This function works only for two-dimensional sweep PSS results. The inner sweep should be phase and the outer sweep should be magnitude.



This function includes the following fields:

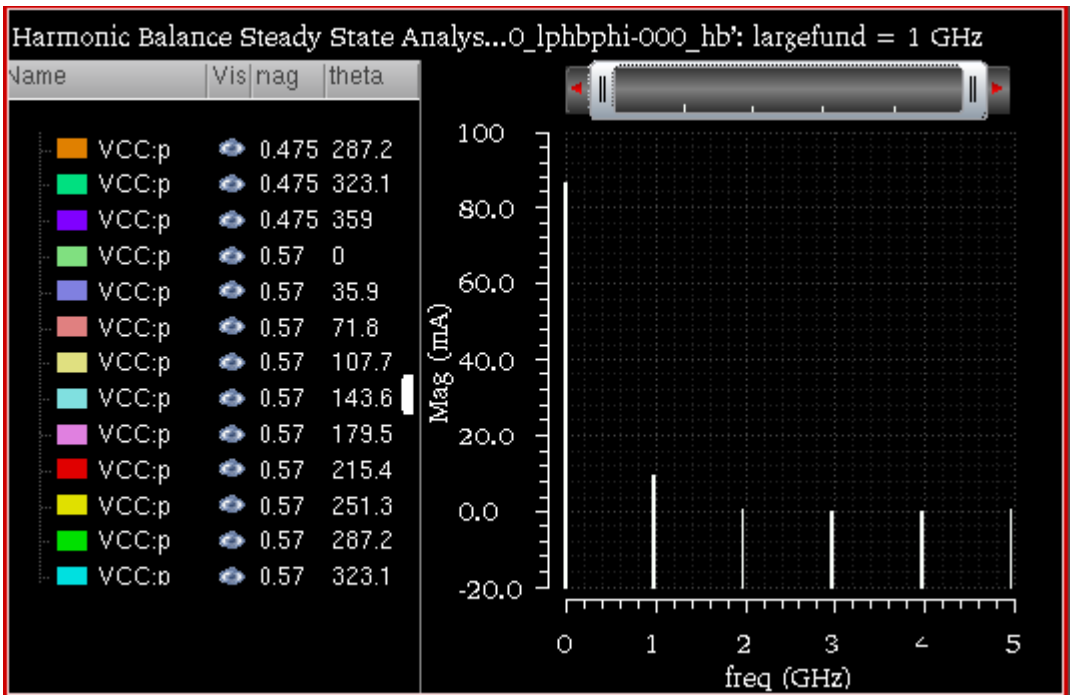
- *Signal*—Name of the input waveform.
- *Max Value*—Largest value of the contour to be drawn. Default value is `nil`, which specifies that the largest value is to be taken from the results.
- *Min Value*—Smallest value of the contour to be drawn. Default value is `nil`, which specifies that the smallest value is to be taken from the results.
- *Number of Contours*—Number of points on the contour. Default value is `8`.
- *Close Contours*—Boolean flag that specifies if a closed or open contour is to be drawn. If this field is set to `yes`, it specifies that a closed contour is to be drawn. If set to `no`, it specifies that an open contour is to be drawn.

Example

Consider the following input signal from the `hb_fd` analysis:

© 2023 Cadence Design Systems, Inc.

```
i("/VCC/PLUS" ?result "hb_fd" ?resultsDir "./viva/rfworkshop/
simulation/EF_example_loadpull/spectre/schematic/psf")
```



The following arguments are specified in this example:

- *Signal*—`i("/VCC/PLUS" ?result "hb_fd" ?resultsDir "./viva/rfworkshop/simulation/EF_example_loadpull/spectre/schematic/psf")`
- *Max Value*—`5`
- *Min Value*—`0`
- *Number of Contours*—`9`
- *Close Contours*—`no`

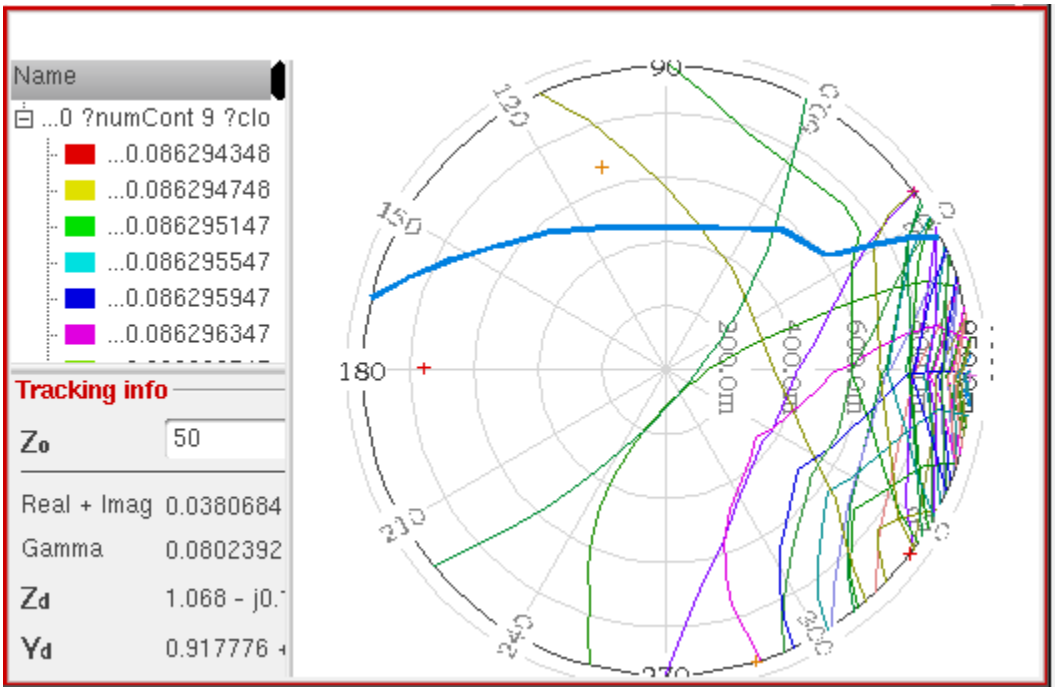
The following expression is created in the Buffer:

```
awvRfLoadPull(mag(i("/VCC/PLUS" ?result "hb_fd" ?resultsDir "./viva/
rfworkshop/simulation/EF_example_loadpull/spectre/schematic/
psf"))?maxValue 5 ?minValue 0 ?numCount 9 ?closeCont nil)
```

Before you evaluate this expression, ensure that the `Polar` option is selected in the Plot Destination drop-down.

© 2023 Cadence Design Systems, Inc.

Now, when you evaluate the expression, the following output waveform is displayed in a new window.




```
awvRfLoadPull(i("V2:p" ?maxValue nil ?minValue nil ?numCont 9
?closeCont nil)
```


© 2023 Cadence Design Systems, Inc.

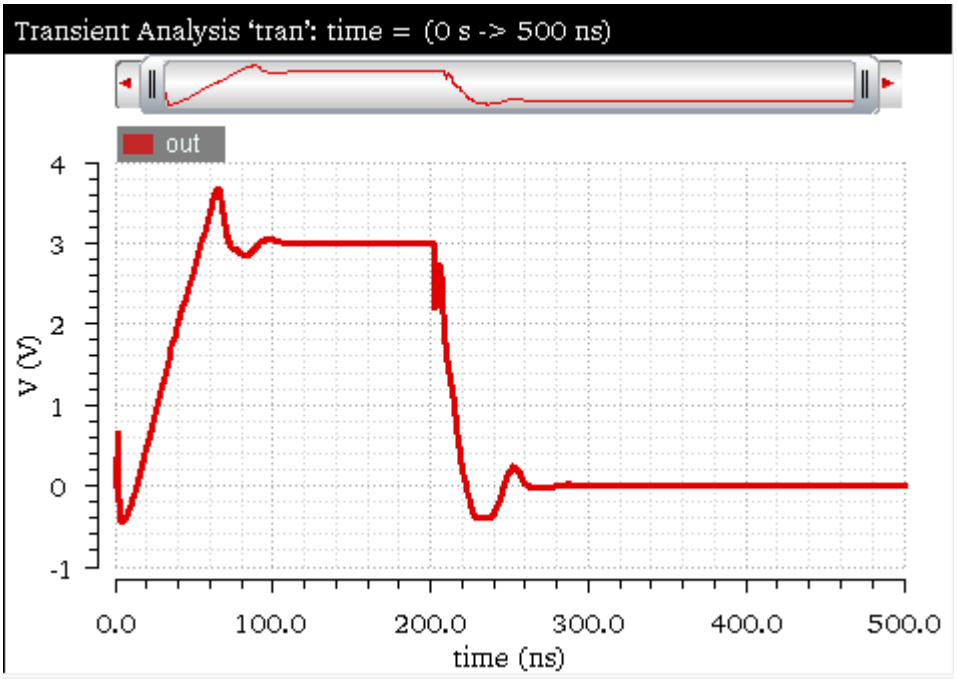
lshift

Shifts the data in the Graph Window to the left by the specified amount. A negative value shifts the data to the right. This function is available only in the SKILL mode.

- *Signal*—Name of the signal.
- *Delta X*—Amount by which you want to shift the data.

Example

This example shows the output waveform is left shifted by the specified amount when you apply the `lshift` function on the following input waveform:



The following arguments are specified in this example:

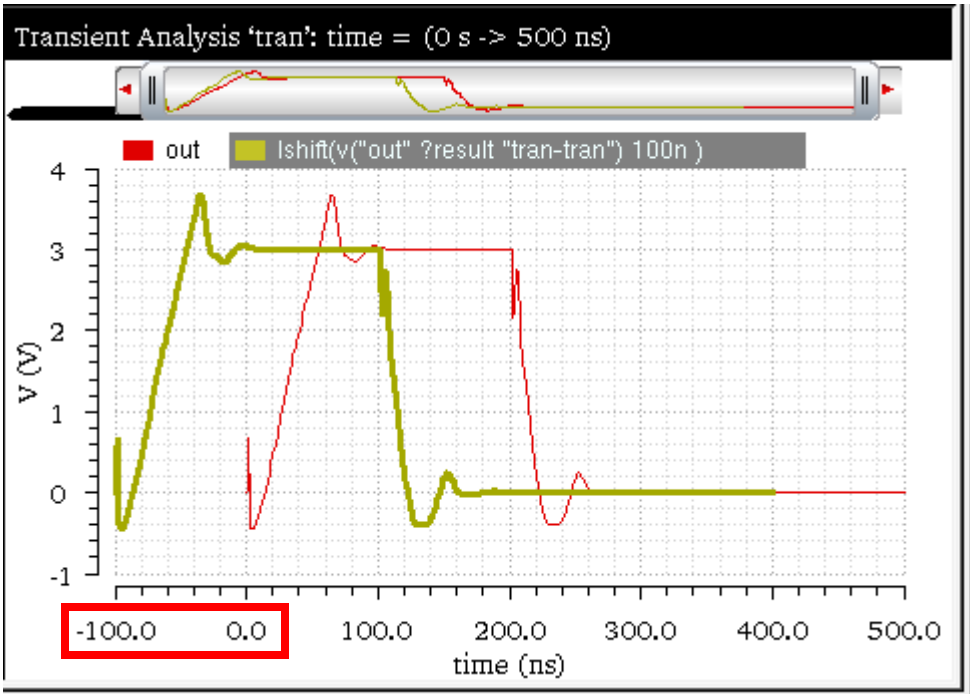
© 2023 Cadence Design Systems, Inc.

- *Signal*—`v(out ?result "tran-tran")`
- *Delta X*—`100n`

The following expression is created in the Buffer:

```
lshift(v(out ?result "tran-tran" ?resultsDir "./ampsim.raw") 100n)
```

Now, when you evaluate this expression, the output waveform is shifted by a value of 100n on the X-axis as shown in the figure below.



Related OCEAN Function

The equivalent OCEAN command for `lshift` is:

```
lshift( o_waveform n_delta )
     => o_waveform/nil
```

For more information about this OCEAN function, see lshift in *OCEAN Reference*.

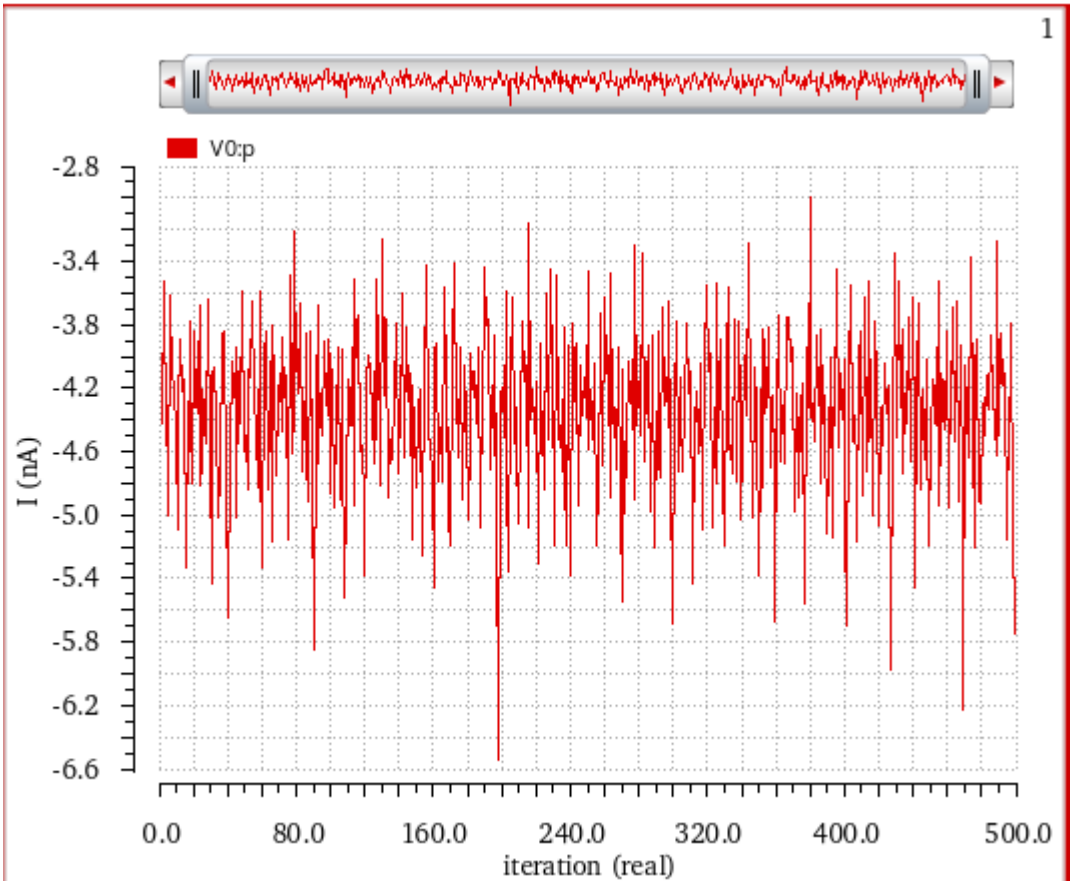
© 2023 Cadence Design Systems, Inc.

normalQQ

Returns a quantile-quantile plot of the sample quantiles versus theoretical quantiles from a normal distribution. If the distribution is normal, the plot is close to a linear waveform.

Example

Consider the following waveform from the Monte Carlo analysis:



The `histogram2D` function is applied on this input waveform with the following arguments:

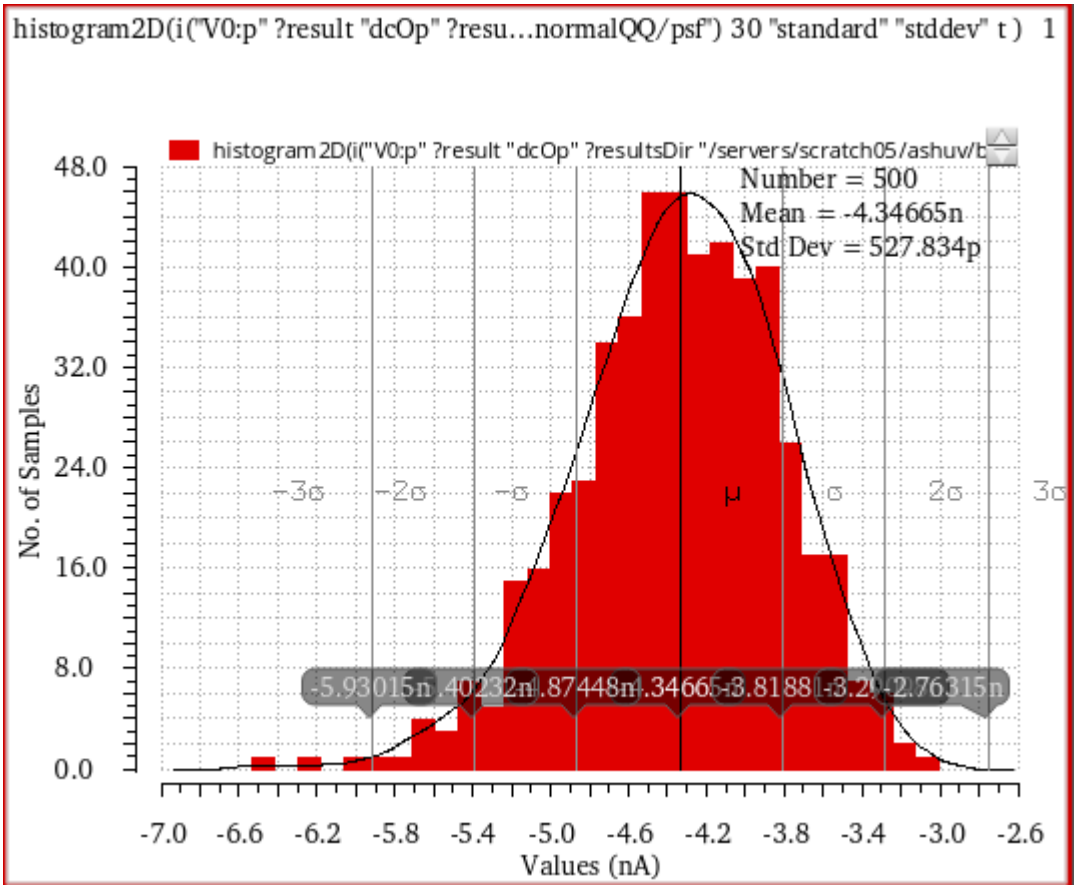
- *Signal*—`i("V0:p" ?result "dcOp" ?resultsDir "../normalQQ/psf")`
- *Number of bins*—`30`
- *Type*—`standard`
- *Std Dev Lines*—`yes`
- *Density Estimator*—`yes`

© 2023 Cadence Design Systems, Inc.

The corresponding expression created in the Buffer is as follows:

```
histogram2D(i("V0:p" ?result "dcOp" ?resultsDir "../normalQQ/psf")
30 "standard" "stddev" t )
```

When you evaluate this expression, the following output waveform representing the data in histogram is displayed in a new graph window.

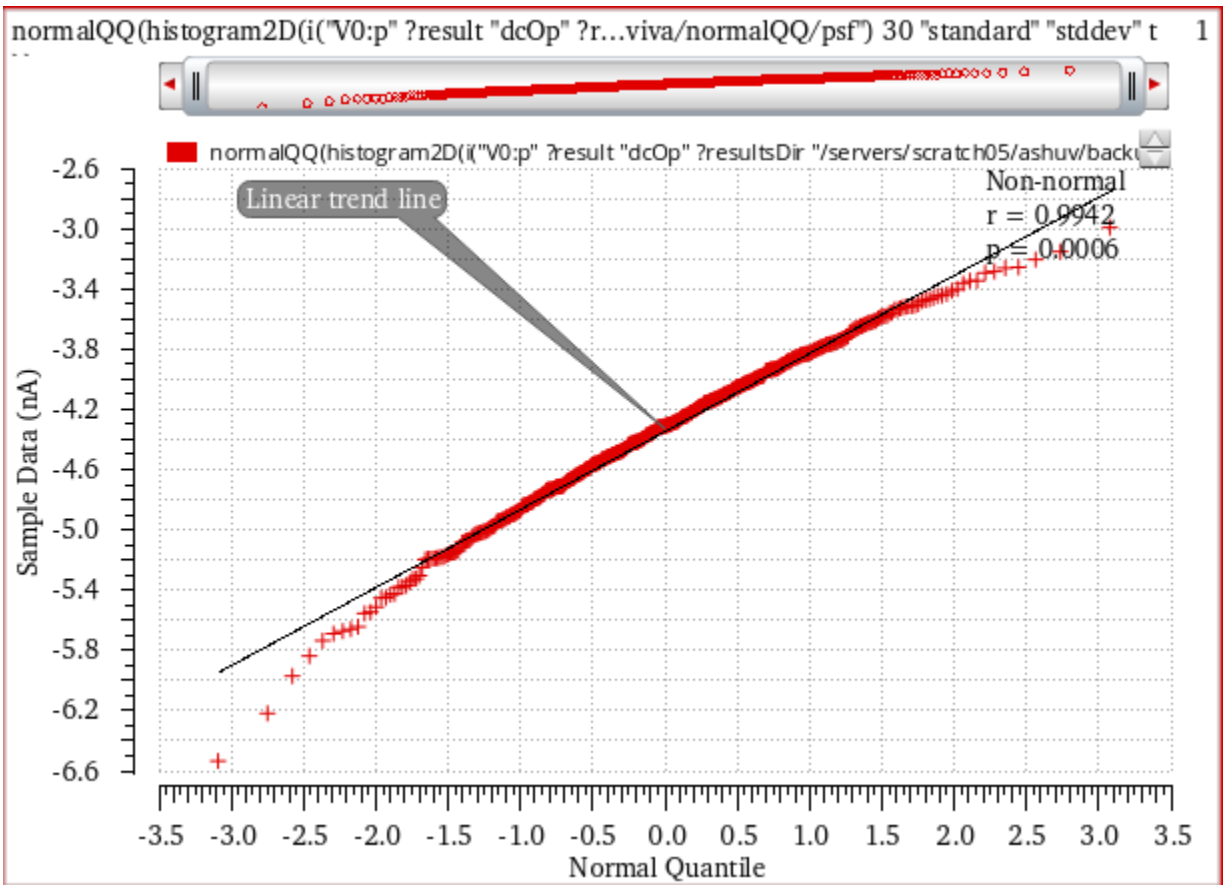


Now, apply the `normalQQ` function on the expression created for the `histogram2D` function. When you select normalQQ function from the Function Panel, the function is applied on the expression in the Buffer. The corresponding expression created for `normalQQ` is as follows:

normalQQ(histogram2D(i("V0:p" ?result "dcOp" ?resultsDir "../normalQQ/psf") 30 "standard" "stddev" t))

© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the quantile-quantile waveform is displayed in a new graph window, as shown in the figure below:



On the quantile-quantile plot, `r` stands for the correlation coefficient value between the Sample Data and Normal Quantile, which is a number between `-1` and `1`. A hypothesis test is then performed on the correlation coefficient, and the null hypothesis is that the distribution is normal. `p` stands for the p-value of this hypothesis test, and it means the probability a correlation coefficient smaller or equal to `r` can be generated under a normal distribution. Small `p` indicates that the distribution is unlikely to be normal, while large `p` means the distribution is close to normal. Conventionally, a distribution is considered to be non-normal when `p` is less than `0.05`, and a label `Non-normal` is shown in this case. A label `Normal` is shown when `p` is significantly larger than `0.05`, and a label `Near-normal` is shown when `p` is larger but close to `0.05`.

Related OCEAN Function

The equivalent OCEAN command for `normalQQ` is:

© 2023 Cadence Design Systems, Inc.

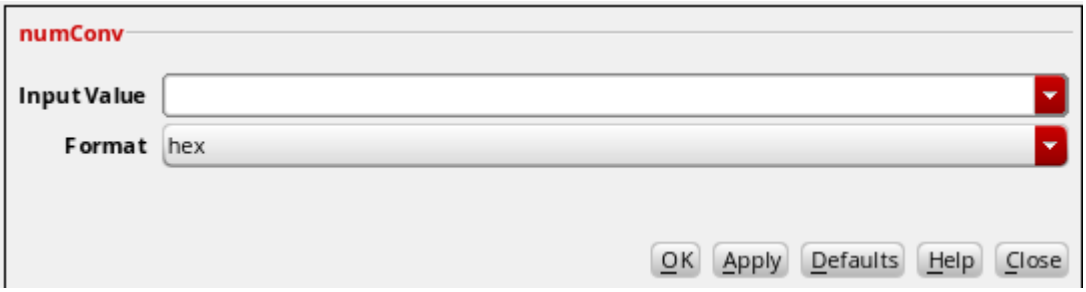
```
normalQQ( o_waveform )
     => o_waveform/nil
```

For more information, see `normalQQ` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

numConv

Converts a given bus value string into the specified format.



This function includes the following arguments:

- *Input Value*—The input number in string format that you want to convert.

  Note: The specified input value must be enclosed in double quotes, otherwise, an error message is displayed in CIW.

- *Format*—The type of format in which you want to convert the specified number. The default value is `hex` (hexadecimal). The other available values are `bin` (binary), `oct` (octal), `dec` (decimal), `sdec` (signed decimal), and `udec` (unsigned decimal).

Example

The example below converts a binary number, `0b1100`, into hexadecimal, octal, decimal, signed decimal, and unsigned decimal formats:

- `hex`: Specify the following arguments:
  - *Input Value*—`"0b1100"`
  - *Format*—`hex`

  The following expression is created in the Buffer: `numConv("0b1100" "hex" t)`

  When you click the evaluate button, the number is converted into a hexadecimal number and the following output is displayed in Buffer: `0xc`.

- `oct`: If you specify the *Input Value* as `"0b1100"` and select *Format* as `oct`, the following expression is created in the Buffer: `numConv("0b1100" "oct" t)`

  When you click the evaluate button, the number is converted into a octal number and the following output is displayed in Buffer: `014`

© 2023 Cadence Design Systems, Inc.

- `dec`: If you specify the *Input Value* as `"0b1100"` and select *Format* as `dec`, the following expression is created in the Buffer: `numConv("0b1100" "dec" t)`

  When you click the evaluate button, the number is converted into a decimal number and the following output is displayed in Buffer: `12`

- `sdec`: If you specify the *Input Value* as `"0b1100"` and select *Format* as `sdec`, the following expression is created in the Buffer: `numConv("0b1100" "sdec" t)`

  When you click the evaluate button, the number is converted into a signed decimal number and the following output is displayed in Buffer: `-4`

- `udec`: If you specify the *Input Value* as `"0b1100"` and select *Format* as `udec`, the following expression is created in the Buffer: `numConv("0b1100" "udec" t)`

  When you click the evaluate button, the number is converted into an unsigned decimal number and the following output is displayed in Buffer: `12`

© 2023 Cadence Design Systems, Inc.

overshoot

Returns the overshoot/undershoot of a signal as a percentage of the difference between initial and final values. Overshoot is calculated as:

$$OvershootOut = \frac{MaximumValue - FinalValue}{FinalValue - InitialValue}$$

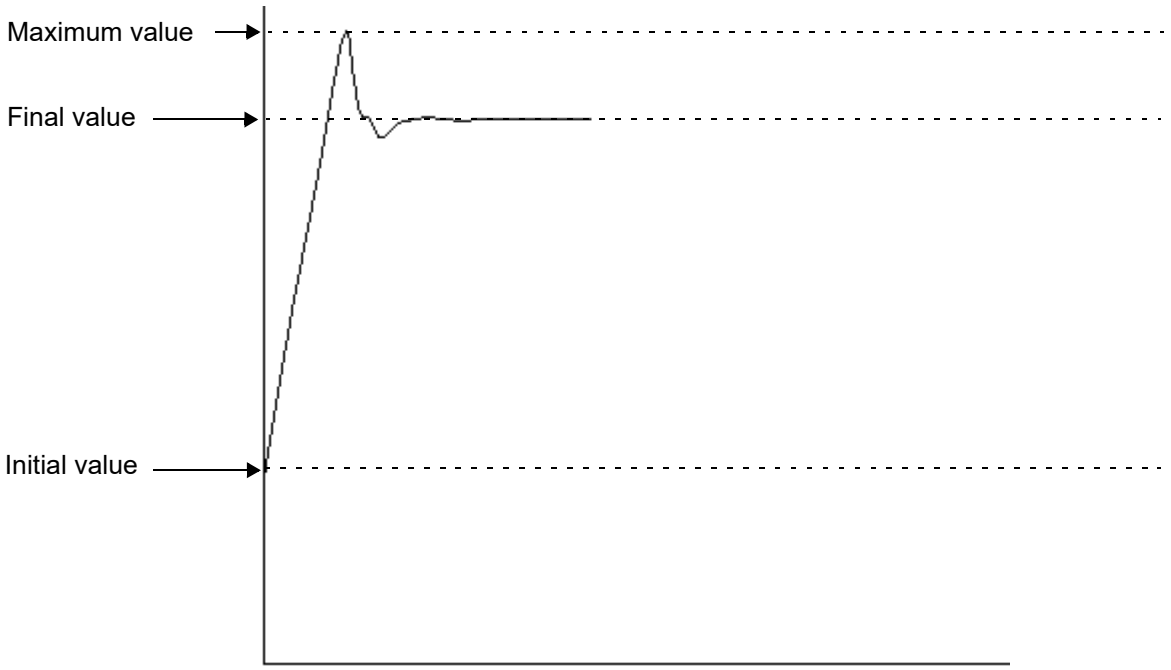
This function includes the following arguments in the SKILL mode:

- *Signal*—Name of the signal.
- *Initial Value Type*—Specifies whether the initial value is the Y-axis value at the specified X-axis value (`y at x`) or Y-axis value (`y`).
- *Initial Value*—Initial value. To calculate the undershoot of a signal, the Initial Value should be higher than Final Value.
- *Final Value Type*—Specifies whether the final value is the Y-axis value at the specified X-axis value (`y at x`) or Y-axis value (`y`).
- *Final Value* is the final value.
- *Number of Occurences*—Specifies whether you want to retrieve only one occurrence of an overshoot event for the given waveform (*single*), or all occurrences of overshoot for the given waveform which you can later plot or print (*multiple*).
- *Plot/print vs*—Specifies whether you want to retrieve overshoot data against *time* (or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform.
  The value in this field is ignored when you specify *Number of Occurences* as *single*.

© 2023 Cadence Design Systems, Inc.

Example

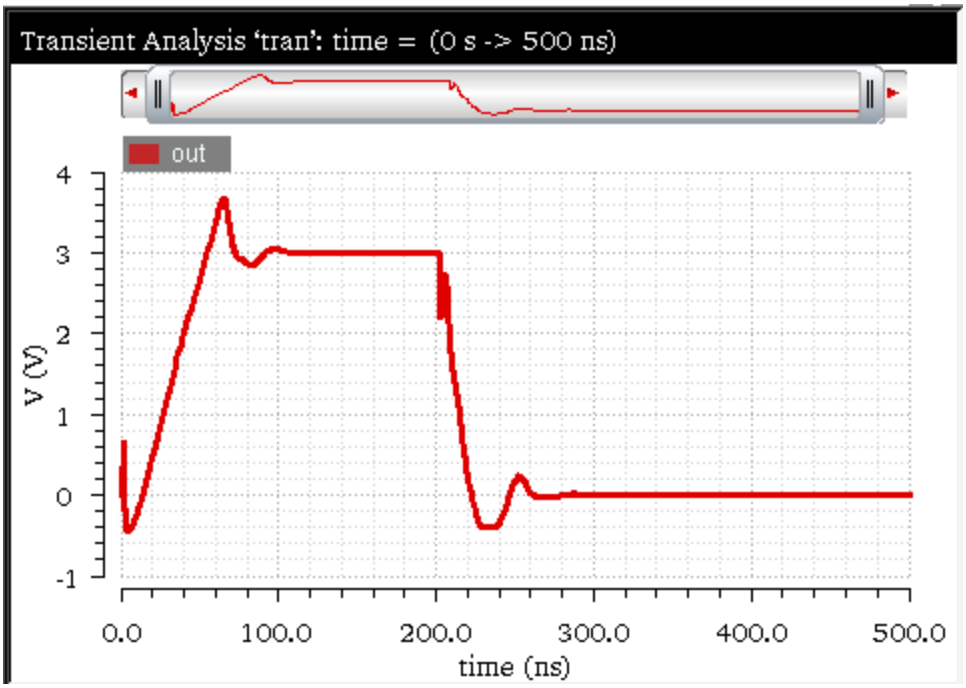
The following diagram illustrates how the result is obtained with the values *signal*=`V(out)`, *Initial Value Type*=`y`, *Final Value Type*=`y`, *Initial Value*=`1`, and *Final Value*=`3`.



© 2023 Cadence Design Systems, Inc.

Example

This example shows the overshoot value generated when you apply the `overshoot` function on the following input waveform:



The following arguments are specified in this example:

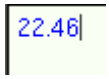
- *Signal*—`v(out ?result "tran-tran")`
- *Initial Value Type*—`y`
- *Initial Value*—`0`
- *Final Value Type*—y
- *Final Value*—3.0
- *Number of occurrences*—single
- *Plot/print vs.*—time

The following expression is created in the Buffer:

```
overshoot(v(out ?result "tran-tran" ?resultsDir "ampsim.raw") 0 nil
3.0 nil nil "time")
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output is displayed in the Buffer:



Related OCEAN Function

The equivalent OCEAN command for `overshoot` is:

```
overshoot( o_waveform n_initVal g_initType n_finalVal g_finalType
     [g_multiple [s_Xname]][g_histoDisplay][x_noOfHistoBins] )
     => o_waveform/n_value/nil
```

For more information about this OCEAN function, see overshoot in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

pavg

Returns the periodic average of a family of signals for each time point.

This function includes the following arguments:

- *Signal*—Name of the signal for which you want to calculate the periodic average.
- *From*—Start time from where you want to calculate the periodic average.
- *To*—End time till you want to calculate the periodic average.
- *Period*—Difference between the end time and start time. This is an optional argument.
- *Sampling Factor*—Sampling factor, which can be increased to increase the accuracy of the output. Default value is `1`.

Additional Information

The periodic average of an input waveform is calculated by using the following equations:

```
pavg (signal from to period (sfactor 1) )
```

```
pavg(tk) = average(sample( o_waveform(t), from tk, to N*T, linear, by T))
```

where,

`o_waveform(t)` is the input waveform

`N` = `floor((to-from) / T)`

`T` = period

© 2023 Cadence Design Systems, Inc.

If the input waveform is a multi-dimensional waveform or a family of waveforms, the output is calculated as:

$$\text{pavg(tk)} = 1/\text{M} \sum_{j=0}^{M-1} \text{pavg(tk)[o_waveform_family[j]]}$$

where,

`o_waveform_family` is the input waveform family

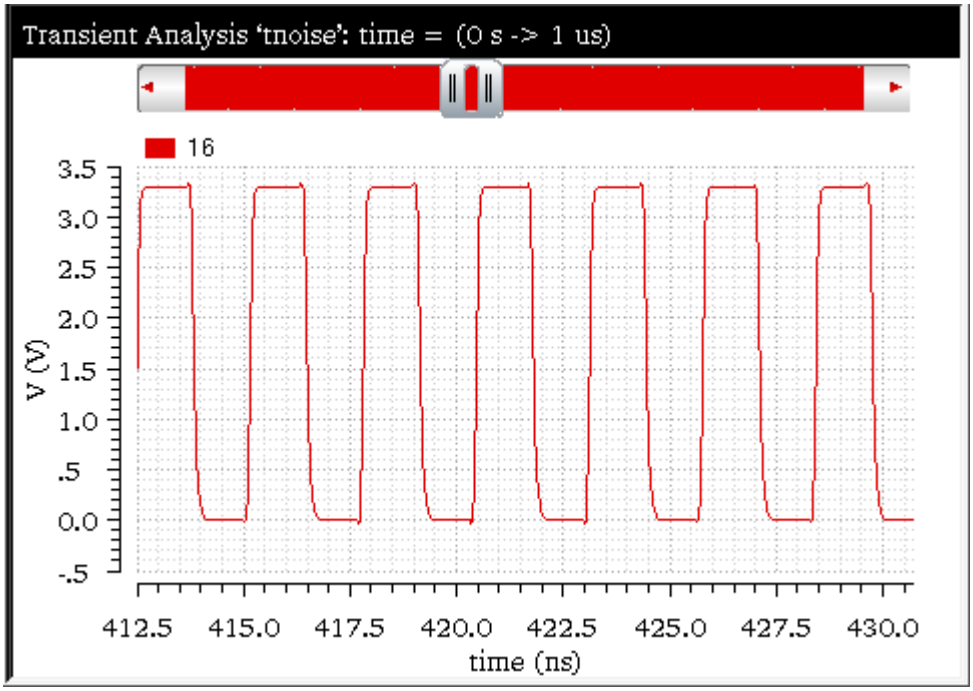
`M` is the number of leaf waveforms in the family.

`o_waveform_family[j]` is the `j`th leaf waveform of the family of waveform represented by the input waveform.

It calculates the `pavg` on each leaf, and then averages over the number of leaves in the family of waveform. In this case the resultant is not a family of waveforms, but a normal waveform of dimension 1.

Example

This example shows the output plot generated when you apply the `pavg` function on the following input waveform:



© 2023 Cadence Design Systems, Inc.

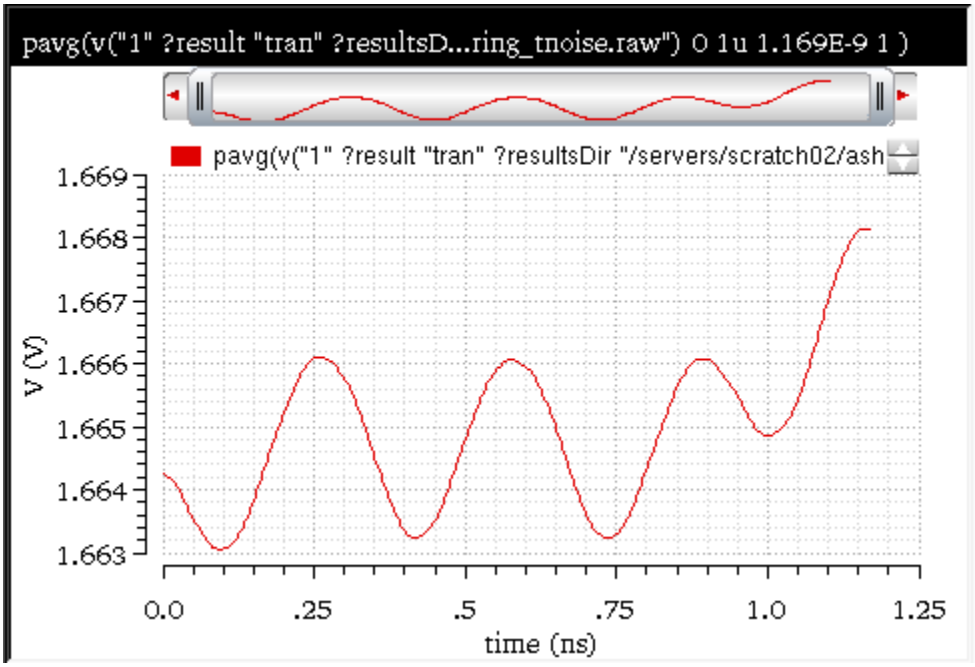
The following arguments are specified in this example:

- *Signal*—`v("16" ?result "tran")`
- *From*—`0`
- *To*—`1u`
- *Period*—`1.169E-9`
- *Sampling Factor*—`1`

The following expression is created in the Buffer:

```
pavg(v("16" ?result "tran" ?resultsDir "./nand2_ring_tnoise.raw") 0
1u 1.169E-9 1 )
```

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `pavg` is:

```
pavg( o_waveform n_from n_to [n_period [n_sfactor]])
     => o_waveform/nil
```

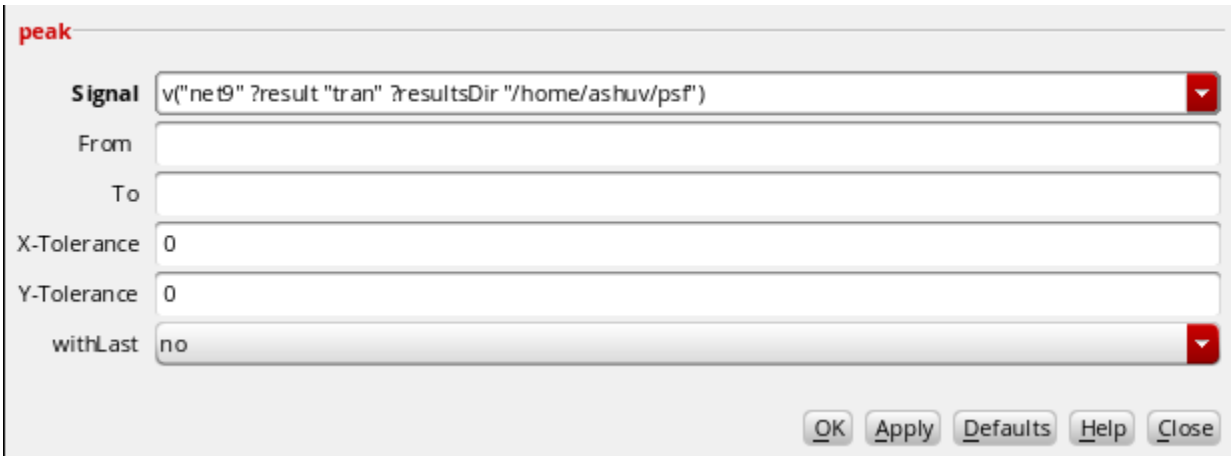

© 2023 Cadence Design Systems, Inc.

For more information about this OCEAN function, see pavg in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

peak

Detects the peaks in the input waveform and returns the X and Y coordinates of these peak points in the form of a waveform. The peak function will not work for waveforms that comprise of complex numbers.



This function is available only in the SKILL mode.

- *Signal*—Name of the signal.
- *From*—Initial point on the specified waveform to start determining the peaks. By default, the first point of the waveform is the starting point.
- *To*—Final point on the specified waveform up to which the peaks are to be determined. By default, the last point of the waveform is the end point.
- *X-Tolerance*—Distance on the X-axis within which all peaks are to be filtered. The default value is `0.0`.
- *Y-tolerance*—Distance on the Y-axis within which all peaks are to be filtered. The default value is `0.0`.
- *withLast*—Determines whether to include the last point in the peak calculation or not. When this argument is set to `yes`, the last point is included in the plot if it is a higher than the previous point. When set to `no`, the plot stops at the last peak. The default value is `no`.

If only the *X-Tolerance* is specified, the peaks are filtered in the X-direction.

If only the *Y-Tolerance* is specified, the peaks are filtered in the Y-axis direction.

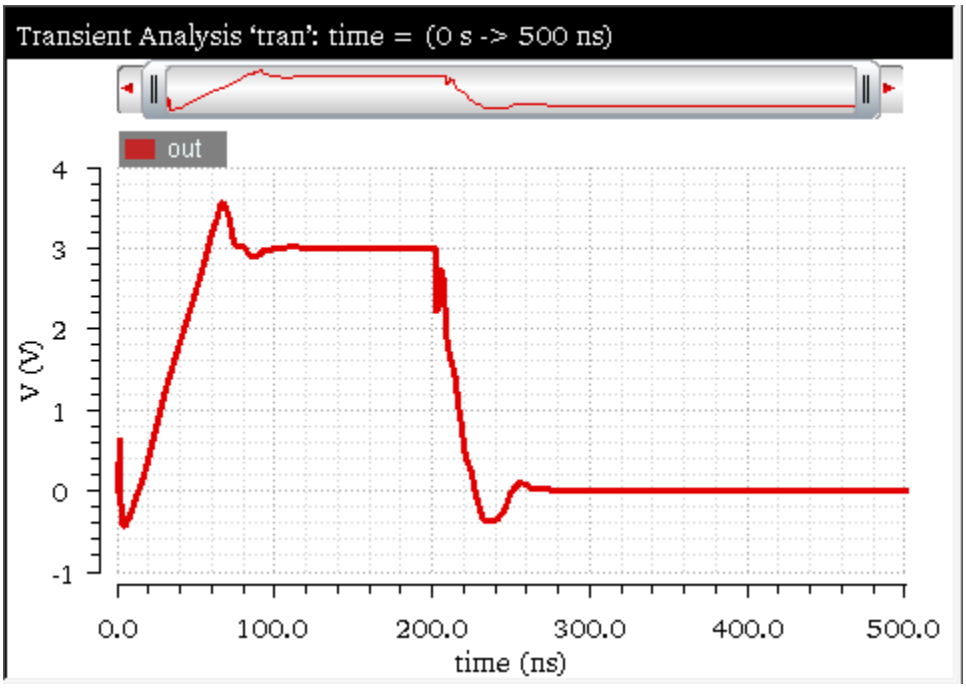
If both *X-Tolerance* and *Y-Tolerance* are specified, the filtering mechanism operates as follows:

© 2023 Cadence Design Systems, Inc.

1. The maximum peak is selected first.
2. All adjacent peaks in the neighborhood of both *X-Tolerance* in the X-axis direction and *Y-Tolerance* in the Y-axis direction are then filtered.
3. All the peaks in the rectangular window thus formed are filtered based on both *X-Tolerance* and *Y-Tolerance*.

Example

This example shows the output waveform generated when you apply the `peak` function on the following input signal:



The following arguments are specified in this example:

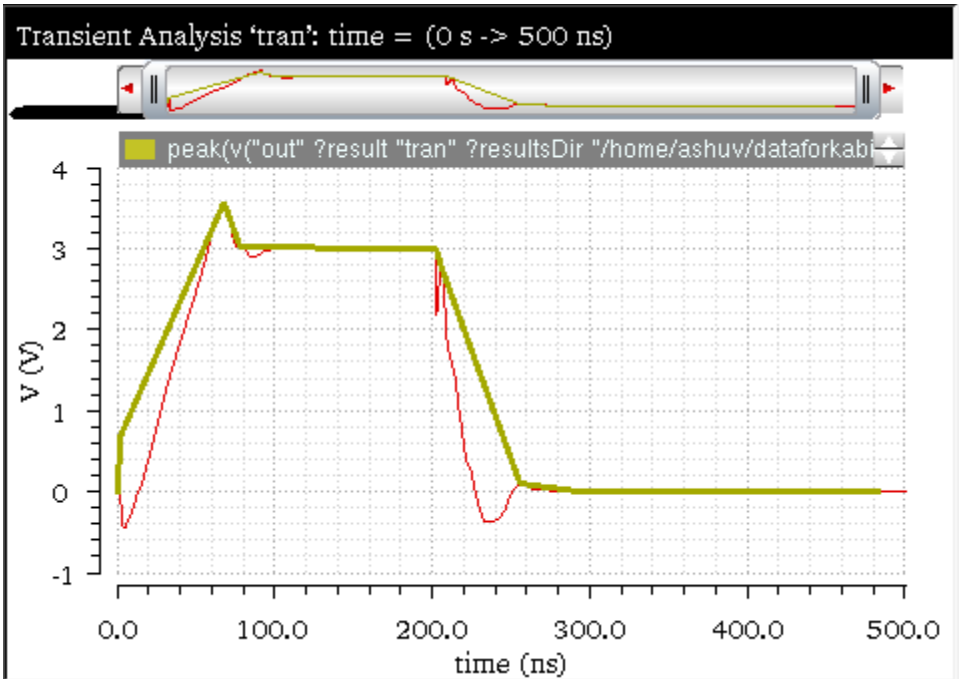
- *Signal*—`v("out" ?result "tran")`
- *From*—`0`
- *To*—`500n`
- *X-Tolerance*—`0`
- *Y-Tolerance*—`0`
- *withLast*—`no`

© 2023 Cadence Design Systems, Inc.

The following expression is created in the Buffer:

```
peak(v("out" ?result "tran" ?resultsDir "./ampsim.raw") ?from 0 ?to
500n ?xtol 0 ?ytol 0 ?withLast nil)
```

Now, when you evaluate this expression and plot the results in append mode, the following waveform is appended to the existing input waveform. This waveform shows the peak points of the input waveform.



Related OCEAN Function

The equivalent OCEAN command for `peak` is:

```
peak( o_waveform ?from f_from ?to f_to ?xtol f_xtol ?ytol f_ytol )
     => o_waveform/nil
```

For more information about this OCEAN function, see peak in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

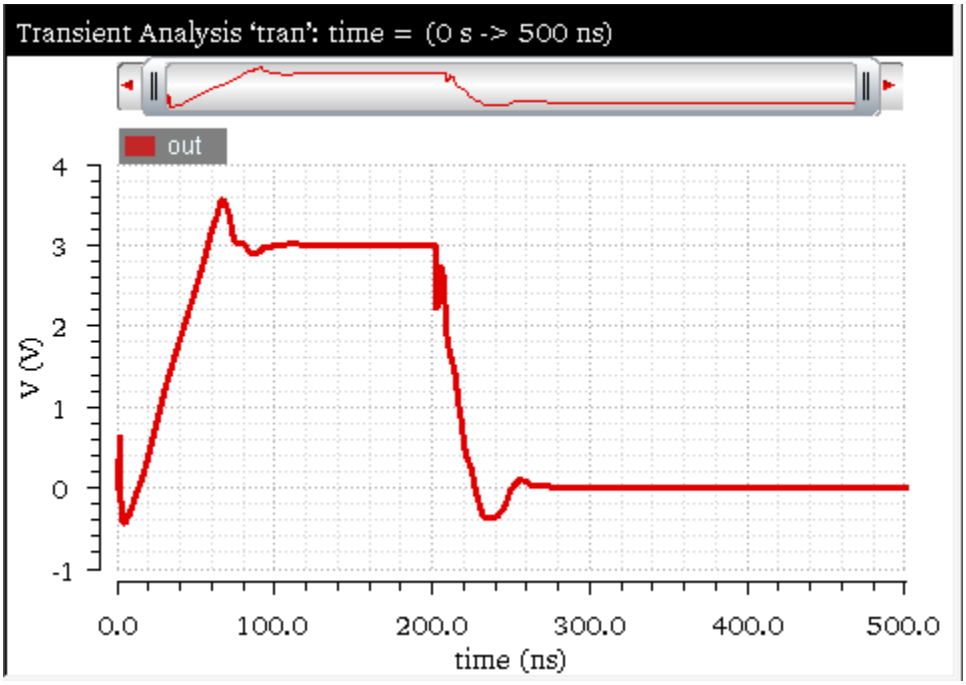
peakToPeak

Returns the difference between the highest and lowest values of a signal.

When you use this function to evaluate data for measurement across corners, a new key argument, `overall`, is added to this function. If the input is a scalar value and the `overall` argument is set to `t`, the `peakToPeak` function returns zero.

Example

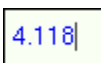
This example calculates the peak to peak value when you apply the `peakToPeak` function on the following input signal:



The following expression is created in the Buffer when you apply this function:

```
peakToPeak(v("out" ?result "tran" ?resultsDir "./ampsim.raw"))
```

Now, when you evaluate this expression, the following output value is displayed in the Buffer:



© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN command for `peakToPeak` is:

```
peakToPeak( o_waveform )
     => o_waveform/n_value/nil
```

For more information about this OCEAN function, see peakToPeak in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

period_jitter

Returns a waveform or a value representing the deviation from the average period.

This function includes the following arguments in the SKILL mode,

- *Waveform* is the name of the signal, expression, or family of waveforms.
- *Cross Type* is the points at which the curves of the waveform intersect with the threshold. While intersecting, the curve may be either rising (`rising`) or falling (`falling`).
- *mode* specifies whether the threshold value is to be calculated by the Virtuoso Visualization and Analysis XL (`auto`) or specified by you (`user`). The auto threshold is calculated as:

  Auto Threshold Value = integral of the waveform divided by the X range.
- *Threshold* is the threshold value against which the period is to be calculated. You need to specify the threshold value only if the *mode* is `user`.

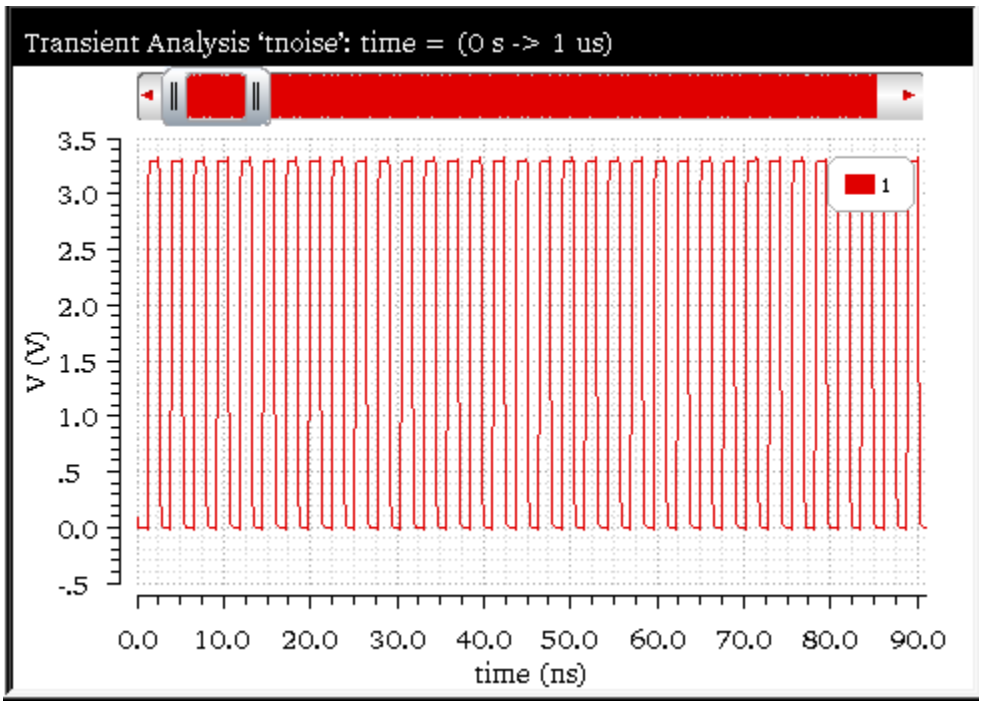
  Note: A threshold tolerance of `0.1*stddev(signal)` is used when detecting crossings.
- *Bin Size* is the width of the moving average window. The deviation of value at the particular point from the average of this window is the jitter.
  If *binsize*=0, all periods are used to calculate the average.
  If *binsize*=*N*, the pervious *N* periods are used to calculate the average.
- *Plot/print vs.* specifies whether you want to retrieve the period jitter against *time* (or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform.

© 2023 Cadence Design Systems, Inc.

- *Output Type* is the type of output. If set to `plot`, the output is a jitter waveform. If set to `sd`, the output is a standard deviation of the jitter waveform. The default value is `plot`.

Example

This example displays the period jitter waveform generated when you apply the `period_jitter` function on the following input waveform:



The following arguments are specified in this example:

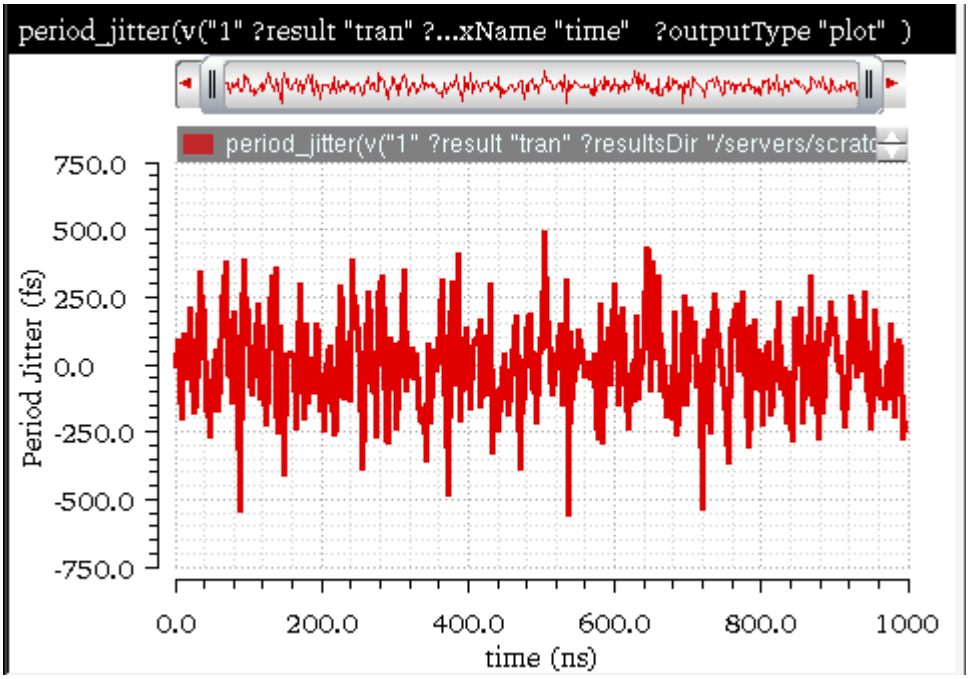
- *Waveform*—`v("1" ?result "tran")`
- *Cross Type*—`rising`
- *mode*—`user`
- *Threshold*—`1.5`
- *Bin Size*—`30`
- *Plot/print vs.*—`time`
- *Output Type*—`plot`

The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
period_jitter(v("1" ?result "tran" ?resultsDir "./
nand2_ring_tnoise.raw") "rising"  ?mode "user" ?threshold 1.5
?binSize 30 ?xName "time"   ?outputType "plot" )
```

Now, when you evaluate this expression and plot the results, the following output waveform is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `period_jitter` is:

```
period_jitter( o_waveform t_crossType
     [?mode t_mode] [?threshold n_threshold] [?binSize n_binSize]
     [?xName t_xName] [?outputType t_outputType] )
     => o_waveform/f_val/nil
```

For more information about this OCEAN function, see period_jitter in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

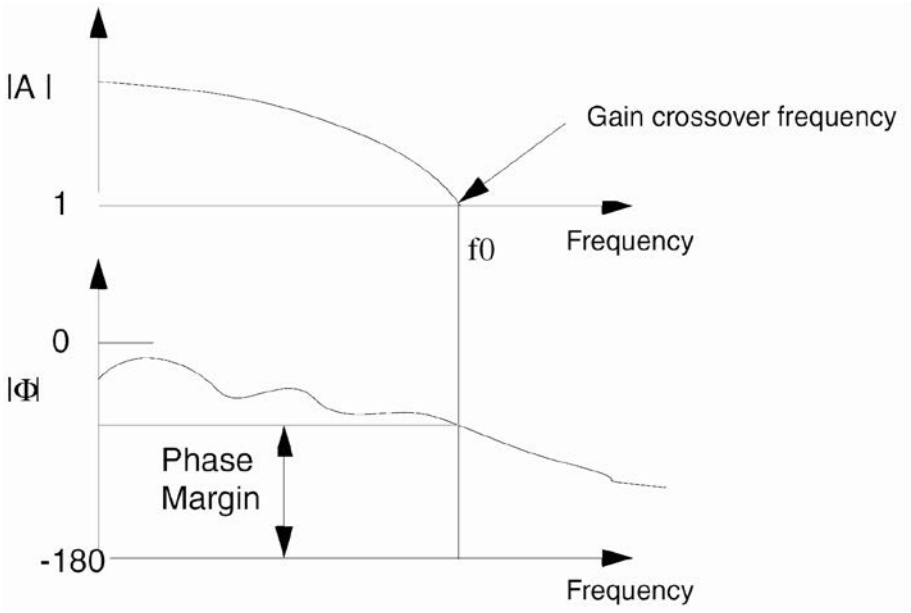
phaseMargin

Computes the phase margin of the loop gain of an amplifier. You supply a waveform representing the loop gain of interest over a sufficiently large frequency range.

$$\text{phaseMargin(gain)} = 180 + \text{phase(value(gain } f_0\text{))}$$

The phase margin is calculated as the difference between the phase of the gain in degrees at f0 and at -180 degrees. The frequency f0 is the smallest frequency where the gain is 1. For stability, the phase margin must be positive.

small-sig is the loop gain of interest over a sufficiently large frequency range.



Linear interpolation is performed in the phaseMargin function in the following two cases:

- When computing the frequency with 0dB magnitude in the magnitude plot. (the below 'cross')
- When computing the phase at that frequency in the phase plot. (the below 'value')

The interpolation is performed on the linear scale to find frequency `1`, that is `0dB`.

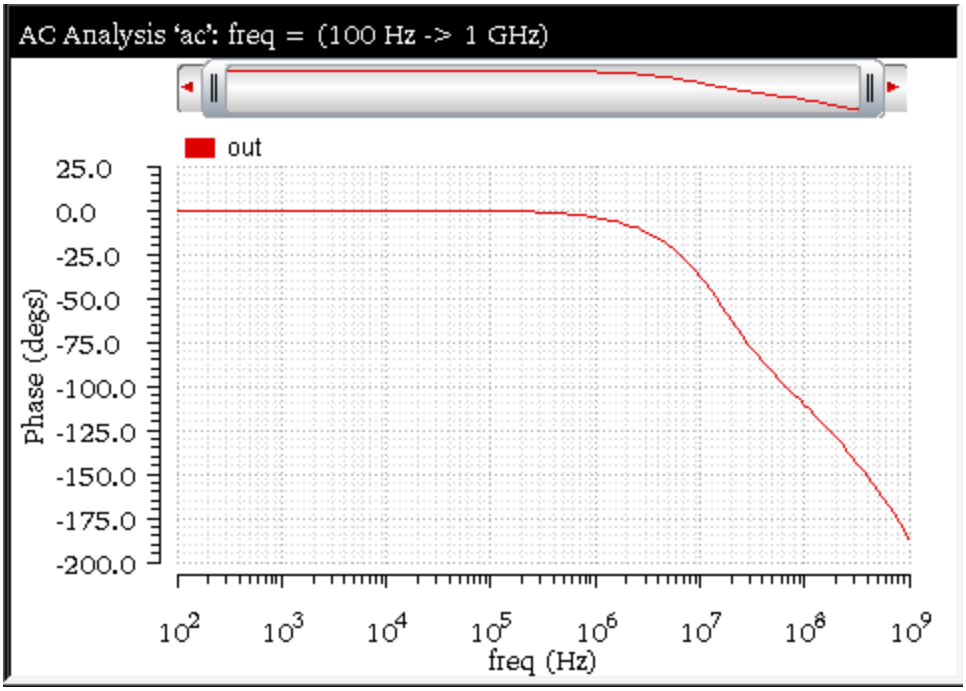
The phaseMargin function is equivalent to running the following expression:

```
wf = awvDigital2Analog(wf 5.0 0.0 2.5) freq = cross( abs(wf) 1)
phaseDeg( -value(wf freq))
```


© 2023 Cadence Design Systems, Inc.

Example

This example shows the phase margin value calculated when you apply the `phaseMargin` function on the following AC signal:



When you apply this function, the following expression is created in the Buffer:

```
phaseMargin(v("out" ?result "ac" ?resultsDir "/.ampTest.raw"))
```

Now, when you evaluate this expression, the following output value is displayed in the Buffer:

Related OCEAN Function

The equivalent OCEAN command for `phaseMargin` is:

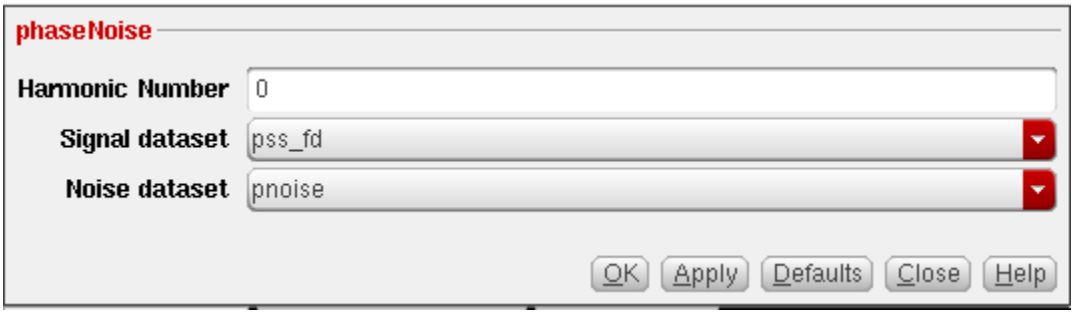
```
phaseMargin( o_waveform )
     => o_waveform/n_value/nil
```

For more information about this OCEAN function, see phaseMargin in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

phaseNoise

Plots the phase noise waveform for noise analysis results.



This function is available only in the SKILL mode.

This function includes the following arguments:

- *Harmonic*—Specify a harmonic number.
- *Signal dataset*—Select the signal data set from the drop-down list.
  Available options: `pss_fd`, `hb_fd`, `hb_mt_fi`, and `qpss_fi`
- *Noise dataset*—Select the noise dataset from the drop-down list.
  Available options: `pnoise`, `pnoise_corr`, `pnoise_src`, `pnoise_xfersrc`, `pnoise_pm`, `hbnoise`, `hbnoise_pm`, `hbnoise_xf`, `hbnoise_mt_xf`, `pnoise_hbnoise`, `qpnoise`, and `qpnoise_hbnoise`

Important

You need to set the database context to the `pss` results directory for which you want to plot the phase noise. For more information about how to set a results directory as in-context results directory, see Setting the In-Context Results Directory.

Example

This example shows the output generated when you apply the `phaseNoise` function on the `./osc13.raw/pss_fd` results database.

Firstly, ensure that the database context is set to the `pss_fd` results directory.

The following arguments are specified in this example:

- *Harmonic number*—`1`
- *Signal dataset*—`pss_fd`

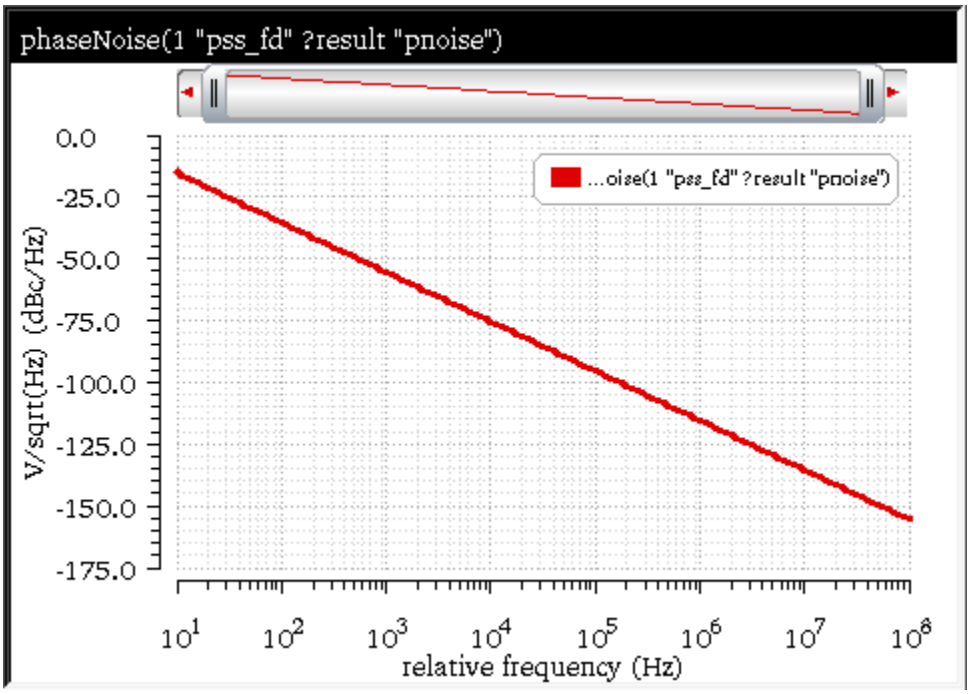
© 2023 Cadence Design Systems, Inc.

- *Noise dataset*—`pnoise`

The following expression is created in the Buffer:

```
phaseNoise(1 "pss_fd" ?result "pnoise")
```

Now, when you evaluate this expression, the following waveform is displayed in the graph window.



Related OCEAN Function

The equivalent OCEAN command for `phaseNoise` is:

```
phaseNoise( g_harmonic S_signalResultName
     [?result s_noiseResultName [?resultsDir t_resultsDir]] )
     => o_waveform/nil
```

For more information, see `phaseNoise` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

PN

Returns a waveform for the transient phase noise of the input waveforms in decibels (`dBc/Hz`). Phase noise is defined as the power spectral density of the absolute jitter of an input waveform. This function is available only in the SKILL mode.

This function includes the following arguments:

- *Signal*—Name of the waveform, expression, or family of waveforms. The expression displayed in the Buffer can be added to this field by selecting *buffer*.
- *Cross Type*—Points at which the curves of the waveform intersect with the threshold. While intersecting, the curve may be either rising (`rising`) or falling (`falling`). The default value is `rising`.
- *Threshold*—Threshold value against which the phase noise is to be calculated.

  Note: A threshold tolerance of `0.1*stddev(signal)` is used when detecting crossings.
- *Tnom*—Nominal time period of the input waveform.
  Default value: `nil`.
- *Window Type*—Window you want to use. The following window types are supported—`Blackman`, `BlackmanHarris`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hanning`, `Kaiser`,

© 2023 Cadence Design Systems, Inc.

`Nuttall`, `Parzen`, `Rectangular`, and `Triangular`. The default window type is `Rectangular`.

- *Smoothing Factor*—Smoothing factor applicable to only Kaiser window. The *Smoothing Factor* field accepts values from `0` to `15`. The value `0` implies no smoothing and is equivalent to a rectangular window. The default value is `1`.
- *Window Size*—Number of frequency domain points to be used while calculating the power spectral density. A larger window size results in an expected operation over fewer samples, which can result in larger variations in the phase noise. A small window size can smear out sharp steps in the phase noise that might be present. Default value is `256`. For more information, see psd.

  The window size affects the size of the `psd` which determines the frequency resolution. It should not exceed the number of samples.
- *Detrending Mode*—Determines the expected trend for the underlying data while calculating the power spectral density. The default value is `None`. For more information, see psd.
- *Coherent Gain*—The coherent gain of a window is the zero frequency gain (or the DC gain) of the window. It is calculated by normalizing the maximum amplitude of the window to one and then summing the values of the window amplitudes over the duration of the window. The result is then divided by the length of the window (that is, the number of samples).
  When you do a dft, the applied Window Type (aside from the rectangular type) changes the signal amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.
  Valid Values: `none`, `default`, `magnitude`, `dB20`, or `dB10`
  Default value: `default`
- *Coherent Gain Factor*—This is the scaling factor. A non-zero value that scales the power spectral density by 1/(Coherent Gain).
  Valid values : 0 < coherent_gain_factor < 1. You can use 1 if you do not want this scaling factor to be used.
  Default value : `0`.
- *Method Type*—Determines the algorithm you want to use to calculate the phase noise, which can be `Absolute Jitter Method` or `Direct Power Spectral Density Method`.
  Default value: `Absolute Jitter Method`.

  The `Absolute Jitter Method` calculates PN as the `psd` of the absolute jitter in radians for *Tnom* at the specified threshold.

  The `Direct Power Spectral Density` method calculates the `psd` of the input waveform, divides it by `2`, and takes the frequency range `nominal frequency ±`

© 2023 Cadence Design Systems, Inc.

`nominal frequency`. The values on the two sides of the nominal frequency are then summed and averaged. The return value is one side of this result given in terms of offset frequency.

Note: When the method type is selected as `Power Spectral Density Method`, ensure that a valid nominal time period value is specified in the *Tnom* field, otherwise, an error message is displayed.

Note: The *Window Type, Smoothing Factor, Window Size, Detrending Mode*, and *Coherent Gain Factor*, *Coherent Gain* arguments are used to calculate the power spectral density of the absolute jitter to obtain the phase noise.

Defining the Phase Noise

For a given waveform $v(t), t_{start} \le t \le t_{end}$, that has the following properties:

- Oscillating with expected nominal period, `T`
- Between minimum and maximum values, $v_{min} \le v(t) \le v_{max}$,
- Rising and falling through a given threshold, v_{th}
- In time intervals, $t_k; 0 \le k \le N$

The phase noise, `P(f)`, can be represented as the spectral density (in decibels) of absolute jitter in phase units as a function of reference clock time:

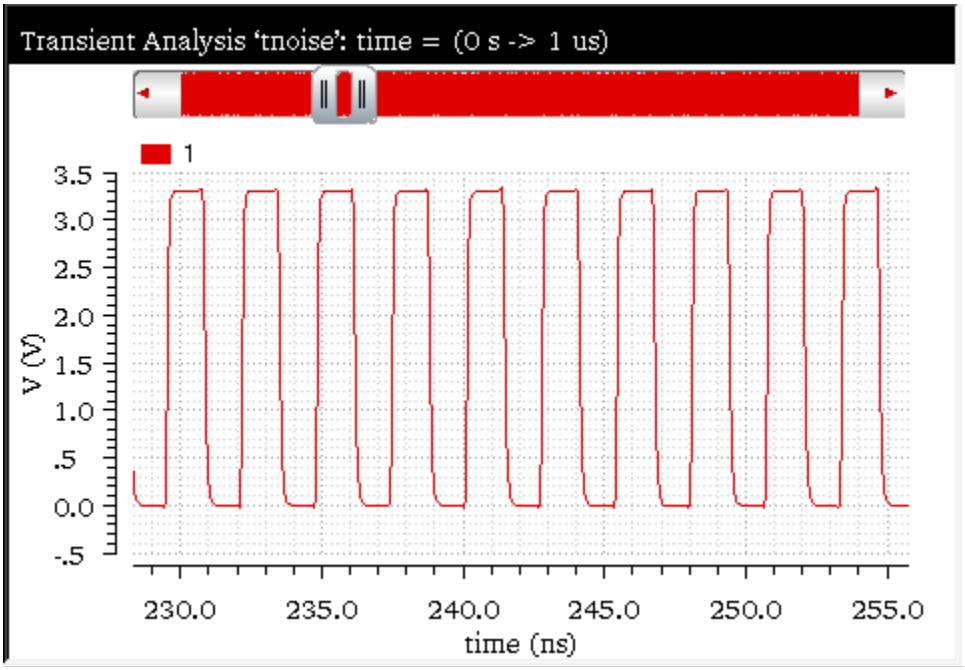
$$P(f) = PSD\left\{\frac{J_a(k \cdot T)}{T}\right\}$$

where, `PSD` is the power spectral density.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input signal from the transient analysis:



The PN function is applied on this input waveform with the following arguments:

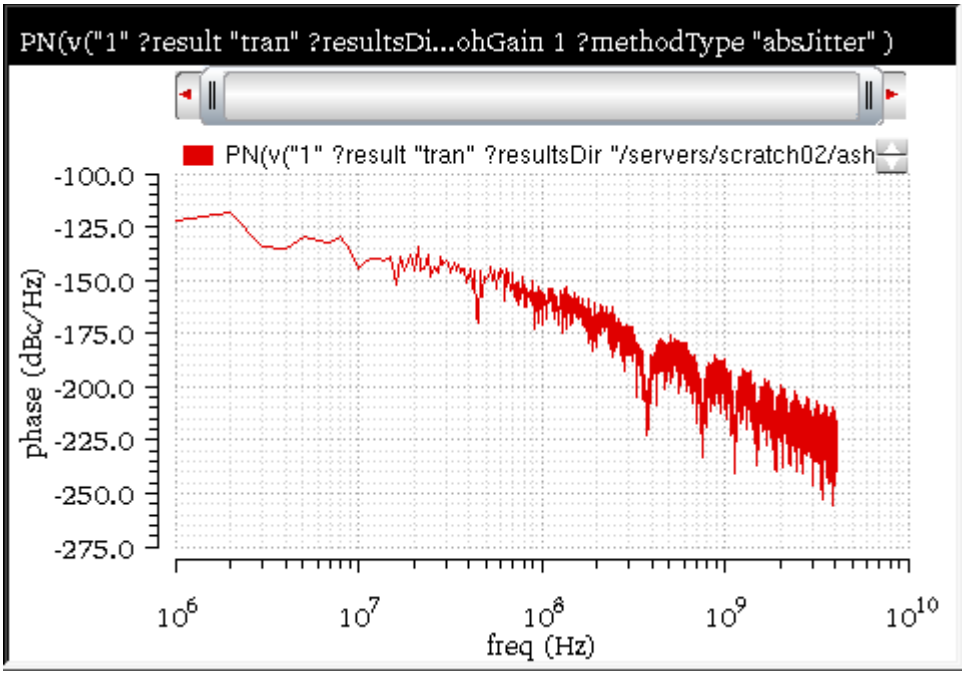
- *Signal*=`v("1" ?result "tran" ?resultsDir "nand2_ring_tnoise.raw")`
- *Cross Type*=`rising`
- *Threshold*=`1.0`
- *Tnom*=`nil`
- *Window Type*=`Rectangular`
- *Smoothing Factor*=`1`
- *Window Size*=`8192`
- *Detrending Mode*=`None`
- *Coherent Gain*=`(magnitde)`
- *Coherent Gain Factor*=`1`
- *Method Type*=`Absolute Jitter Method`

The corresponding expression created in the Buffer is as follows:

© 2023 Cadence Design Systems, Inc.

```
PN(v("out" ?result "tran" ?resultsDir "nand2_ring_tnoise.raw")
"rising" 1.0 ?Tnom nil ?windowName "Rectangular" ?smooth 1
?windowSize 8192 ?detrending "None" ?cohGain 1 ?methodType
"absJitter" )
```

When you evaluate this expression, the following output waveform showing the transient phase noise is displayed in a new graph window:



Related OCEAN Function

The equivalent OCEAN command for `PN` is:

```
PN( o_waveform t_crossType n_threshold ?Tnom n_tnom ?windowName t_windowName
     ?smooth x_smooth ?windowsize x_windowsize ?detrending t_detrending)
     ?cohGain f_cohGain ?methodType t_methodType)
     => o_waveform/nil
```

For more information, see `PN` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

pow

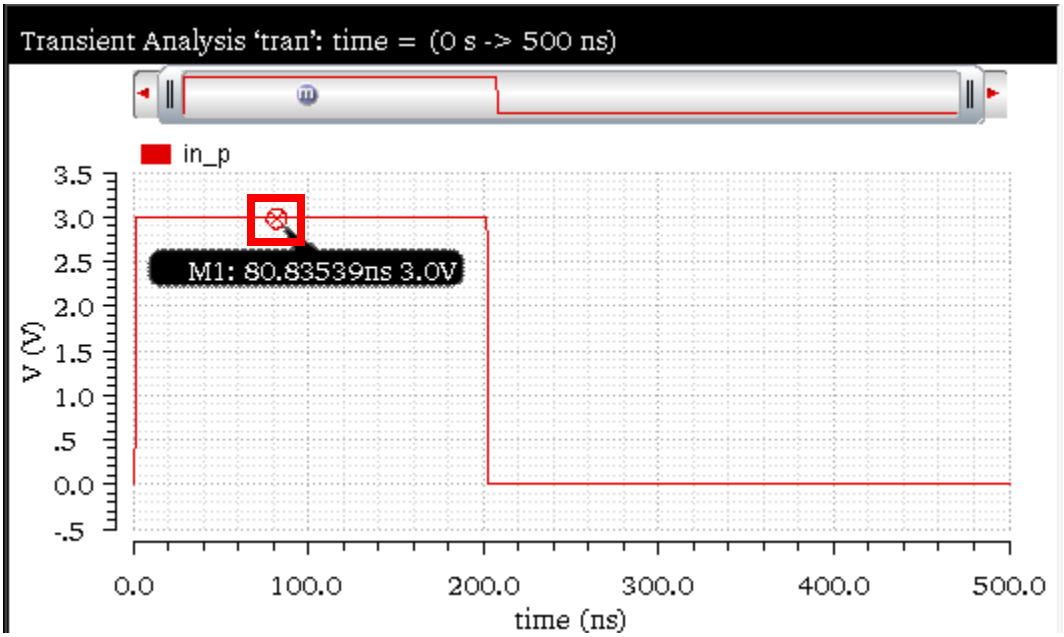
Returns the value of base raised to the power of exponent ($base^{exponent}$).

This function includes the following arguments:

- *Base*—Name of the signal
- *Exponent*—Power value, which indicates how many times you want to multiply the signal. The default value is `1`.

Example

Consider the following input signal from the transient analysis. A point marker is placed on the input waveform at the voltage value `3.0V` on Y-axis.



The `pow` function is applied to this input signal with the following arguments:

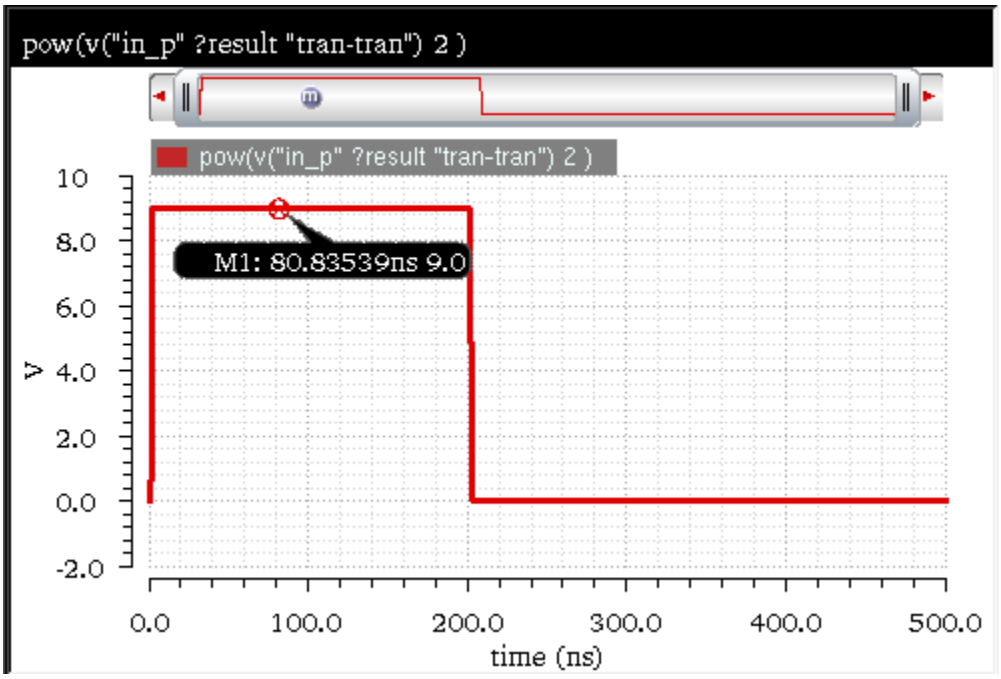
© 2023 Cadence Design Systems, Inc.

- *Base*—`v("in_p" ?result "tran-tran" ?resultsDir "./ampsim.raw")`
- *Exponent*—`2`

The corresponding expression created in the Buffer is as follows:

```
pow(v("in_p" ?result "tran-tran" ?resultsDir "./ampsim.raw")2)
```

When you evaluate this expression, the power is calculated and the following output waveform is displayed in a new graph window. The voltage value for the point marker has been changed to `9.0V` in the output waveform, which represents the `pow(3,2)` calculation.



Below are additional examples to indicate the power calculation:

- `pow( average( v( "/net9" ) ) 0.5 )`

  Returns the square root of the average value of the voltage at `"/net9"`.

- `pow( 2 3 )=> 8`

  Returns the value of `2` to the third power, or `8`.

- `pow( -2 2 )=> 4`

  Returns the value of `-2` to the second power.

- `pow( 2.5 -1.2 )=> 0.3330213`

© 2023 Cadence Design Systems, Inc.

Returns the value of `2.5` to the power of `-1.2`

Related OCEAN Function

The equivalent OCEAN command for `pow` is:

```
pow( {o_waveformBase | n_numberBas} {o_waveformExpn | n_numberExpn} )
     => o_waveform/n_result/nil
```

For more information, see `pow` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

prms

(Periodic Root Mean Square)

Returns the periodic root mean square of a family of signals for each time point, which is the square root of the periodic average of the square of input waveform and is represented as follows:

```
prms(o_waveform n_from n_to n_period n_sfactor) =
sqrt(pavg(o_waveform*o_waveform) from to period sfactor)
```

For more information about the `pavg` function, see pavg.

This function includes the following arguments:

- *Signal*—Name of the signal for which you want to calculate the root mean square.
- *From*—Start time from where you want to calculate the periodic root mean square.
- *To*—End time till you want to calculate the periodic root mean square.
- *Period*—Difference between the end time and start time.
- *Sampling Factor*—Controls the output accuracy, You can increase the sampling factor to increase the accuracy in the output.
  Default value is `1`.

Additional Information

The periodic root mean square is represented by the following equation:

```
prms(t)= sqrt( pavg( fam(t)**2  start_time end_time period sfactor))
```

Here,

© 2023 Cadence Design Systems, Inc.

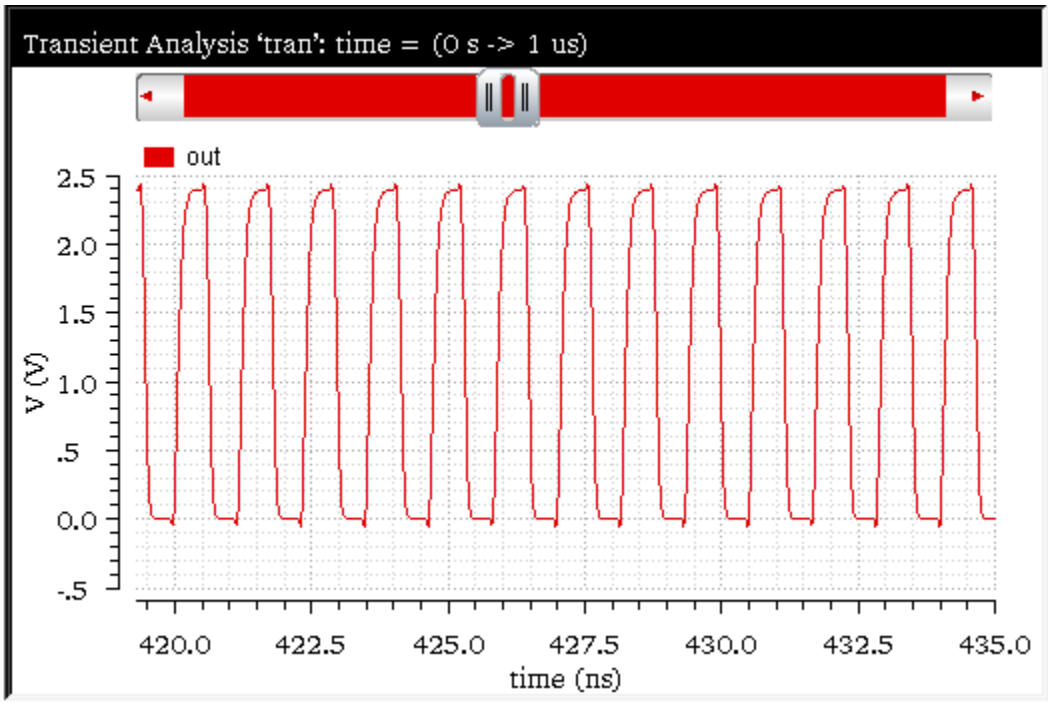
`pavg` is the periodic average

`fam(t)` is the family of `t` waveforms

`period=end_time-start_time`

Example

Consider the following input waveform from the transient analysis:



The `prms` function is applied on this input waveform with the following arguments:

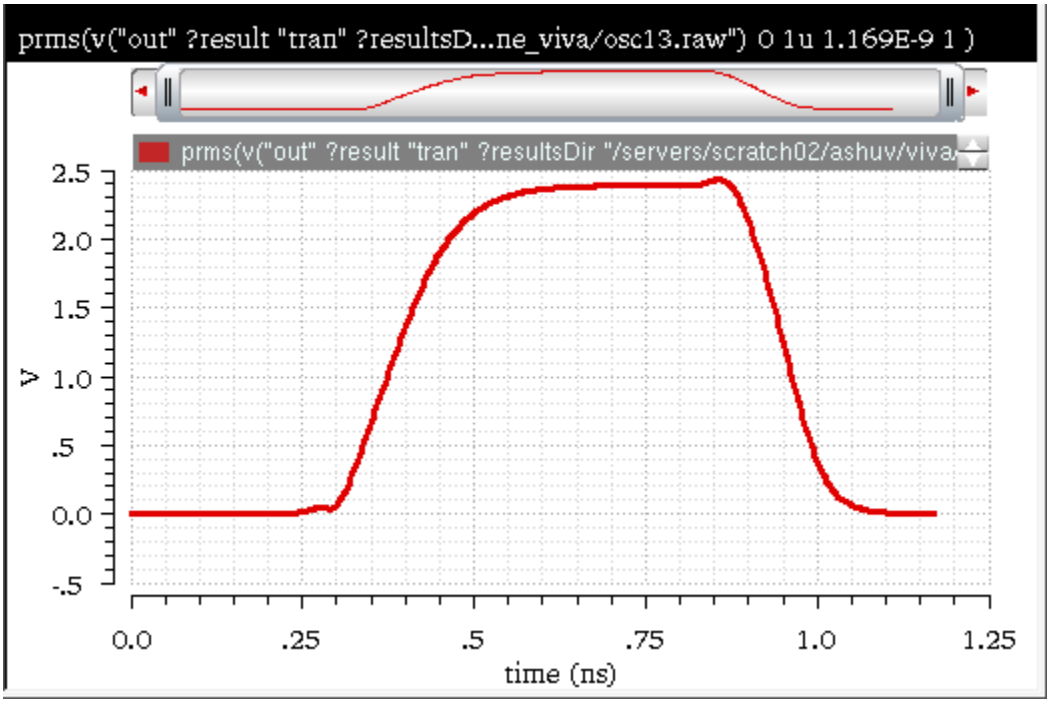
- *Signal*—`v("out" ?result "tran" ?resultsDir "./osc13.raw")`
- *From*—`0`
- *To*—`1u`
- *Period*—`1.169E-9`
- *Sampling Factor*—`1`

The corresponding expression created in the Buffer is as follows:

```
prms(v("out" ?result "tran" ?resultsDir "./osc13.raw") 0 1u 1.169E-9
1 )
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform is displayed in a new graph window to represent the periodic root mean square values:



Related OCEAN Function

The equivalent OCEAN command for `prms` is:

```
prms( o_waveform n_from n_to [n_period [n_sfactor]])
     => o_waveform/nil
```

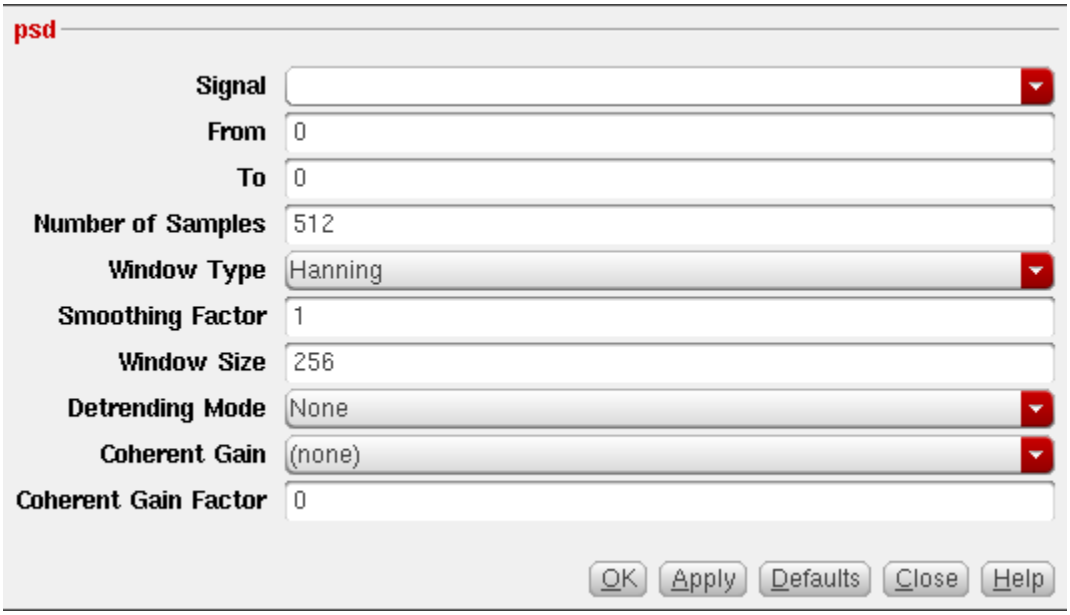
For more information, see `prms` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

psd

Power Spectral Density

Describes how the power (or variance) of a time series (signal) is distributed with frequency. Mathematically, it is defined as the Fourier Transform of the auto correlation sequence of the time series (signal). The waveform is first interpolated to generate evenly spaced data points in time. The spacing of the data points is the inverse of the *dft* sampling frequency. The *psd* is computed by first breaking up the time interval into overlapping segments. Each segment is multiplied, time point by time point, by the specified windowing function. The *dft* is performed on each windowed segment of the baseband waveform. At each frequency, the *dft*s from all segments are averaged together and the squared modulus of these averages gives the *psd*.



This function is available only in the SKILL mode and includes the following arguments:

- *Signal*—Name of the signal.
- *From*—Starting time for the spectral analysis interval.
- *To*—Ending time for the spectral analysis interval.
- *Number* of Samples—Number of time domain points to be used.
  Default value: `512`
- *Window Type*—Window you want to use.
  Default value: `Hanning`
  Valid values: `Blackman, BlackmanHarris, Cosine2, Cosine4, ExtCosBell,`

© 2023 Cadence Design Systems, Inc.

`HalfCycleSine, HalfCycleSine3, HalfCycleSine6, Hanning, Kaiser, Nuttall, Parzen, Rectangular, and Triangular`
If you select *Kaiser*, type in a value for the Kaiser smoothing factor. The smoothing factor must be in the range 0 <= factor <= 15, where 0 is the same as using a rectangular window.

- *Smoothing Factor*—Controls the output accuracy. This argument applies only to the *Kaiser* window type.
  Default value: `1`
  It is applicable to the Kaiser window only. The Smoothing Factor accepts values from `0` to `15`. The value `0` implies no smoothing and is equivalent to a rectangular window.

- *Window Size*—Number of frequency domain points to use in the Fourier analysis. A larger window size results in an expectation operation over fewer samples, which leads to larger variations in the power spectral density. A small window size can smear out sharp steps in the power spectral density that might be present.
  Default value: `256`

- *Detrending Mode*—Specifies the trend mode.
  Valid values: `Linear`, `Mean`, `None`
  Default value: `None`
  The `psd` function works by applying a moving windowed FFT to time-series data. If there is a deterministic trend to the underlying data, you may want to remove the trend before performing the spectral analysis. For example, consider analyzing phase noise in a VCO model. Without the noise, the phase increases more or less linearly with time, so it is appropriate to set the detrending mode to `Linear`. To subtract an average value, set the detrending mode to `Mean`. Where the spectrum of raw data is desired, set the detrending mode to `None`.

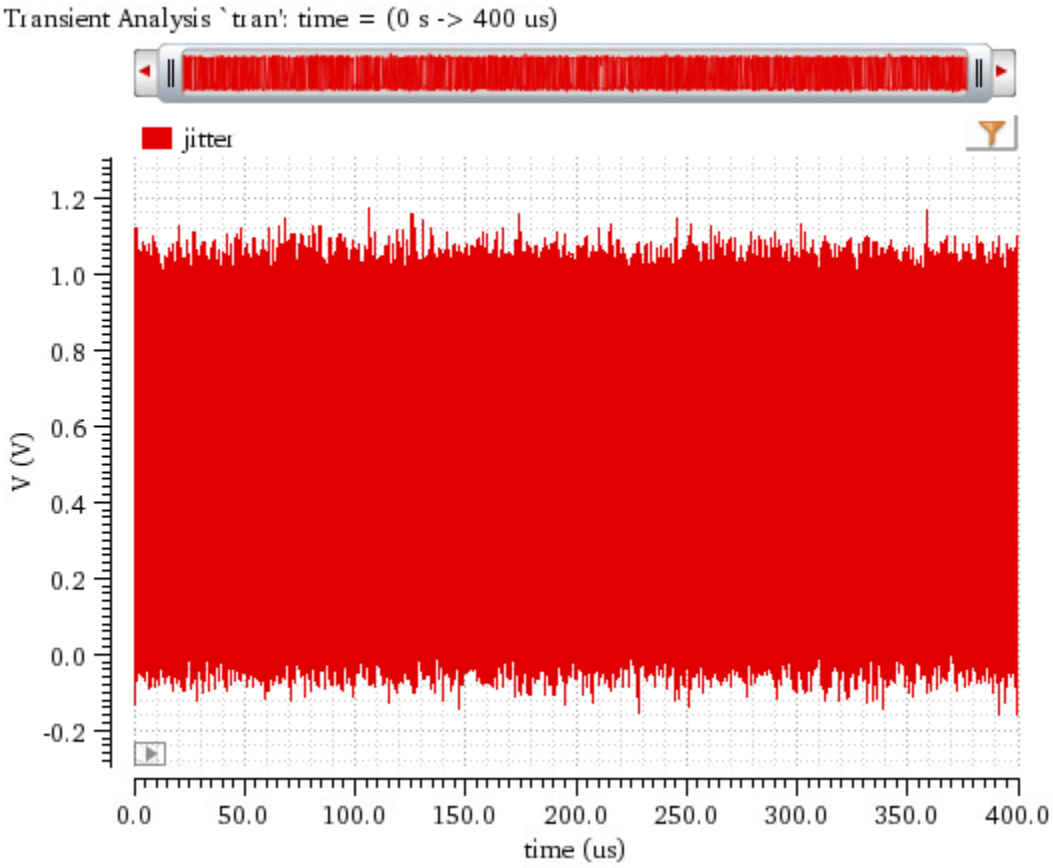
- *Coherent Gain*—The coherent gain of a window is the zero frequency gain (or the DC gain) of the window. It is calculated by normalizing the maximum amplitude of the window to one and then summing the values of the window amplitudes over the duration of the window. The result is then divided by the length of the window (that is, the number of samples).
  When you do a dft, the applied Window Type (aside from the rectangular type) changes the signal amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.
  Valid Values: `none`, `default`, `magnitude`, `dB20`, or `dB10`
  Default value: `default`

- *Coherent Gain Factor*—This is the scaling factor. A non-zero value that scales the power spectral density by 1/(Coherent Gain).
  Valid values : 0 < coherent_gain_factor < 1. You can use 1 if you do not want this scaling factor to be used.
  Default value : `0`

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input signal from the transient analysis:



The `psd` function is applied on this input waveform with the following arguments:

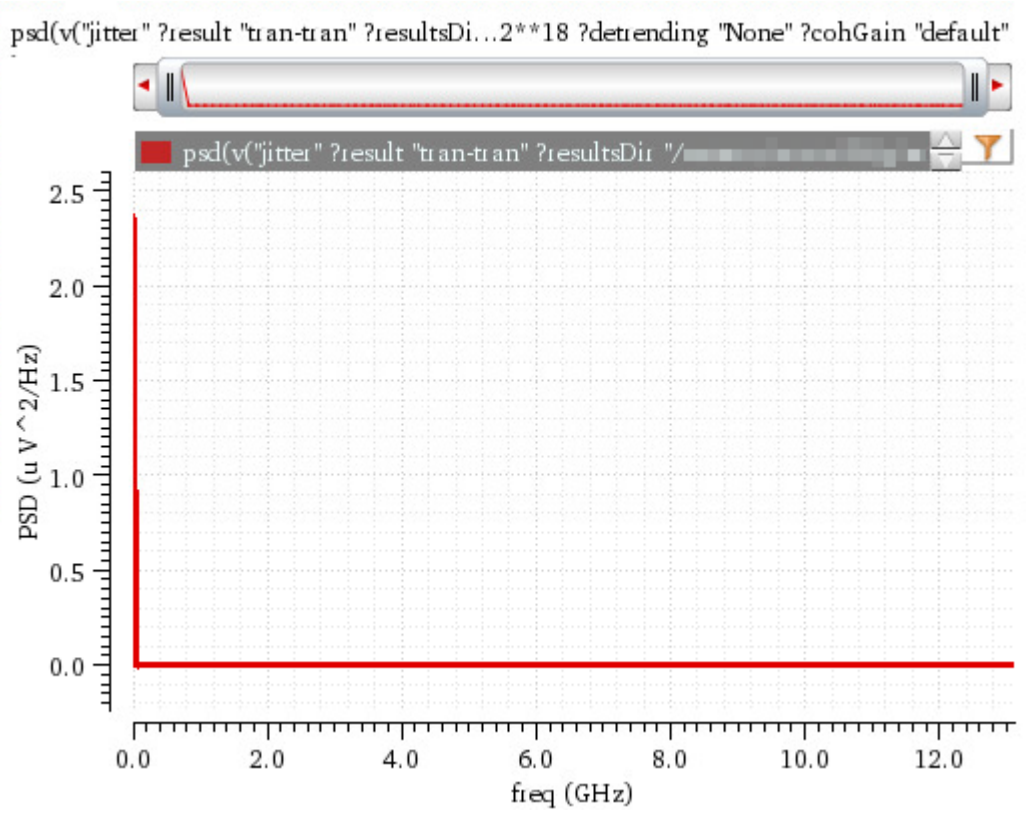
- *Signal*—`v("jitter" ?result "tran-tran" ?resultsDir "./prbs.raw")`
- *From*—`0`
- *To*—`40u`
- *Number of Samples*—`2**20`
- *Window Type*—`Rectangular`
- *Smoothing Factor*—`1`
- *Window Size*—`2**18`
- *Detrending Mode*—`None`

© 2023 Cadence Design Systems, Inc.

- *Coherent Gain*—`(default)`
- *Coherent Gain Factor*—`0`

The corresponding expression created in the Buffer is as follows:

```
psd(v("jitter" ?result "tran-tran" ?resultsDir "./prbs.raw") 0 40u 2**20
?windowName "Rectangular" ?smooth 1 ?windowSize 2**18 ?detrending "None" ?cohGain
"default" )
```

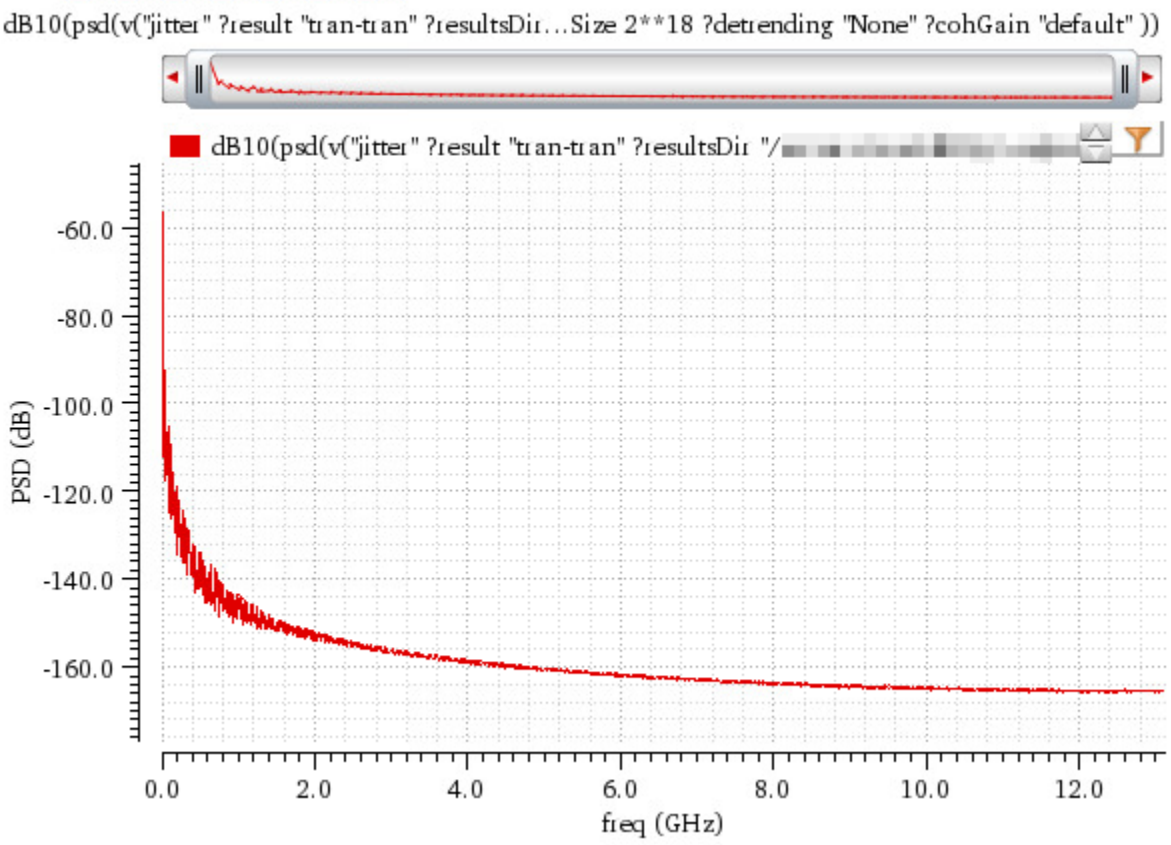


Then, the `dB10` function is applied on this expression to modify the power spectral density calculation. The following expression is created in the Buffer.

```
dB10(psd(v("jitter" ?result "tran-tran" ?resultsDir "./prbs.raw") 0 40u 2**20
?windowName "Rectangular" ?smooth 1 ?windowSize 2**18 ?detrending "None" ?cohGain
"default" ))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the power spectral density is calculated and the spectrum is displayed, as shown in the following output waveform:



Related OCEAN Function

The equivalent OCEAN command for `psd` is:

```
psd( o_waveform f_timeStart f_timeEnd x_num
     ?windowName t_windowName ?smooth x_smooth ?cohGain f_cohGain
     ?windowsize x_windowsize ?detrending t_detrending)
     => o_waveformReal/nil
```

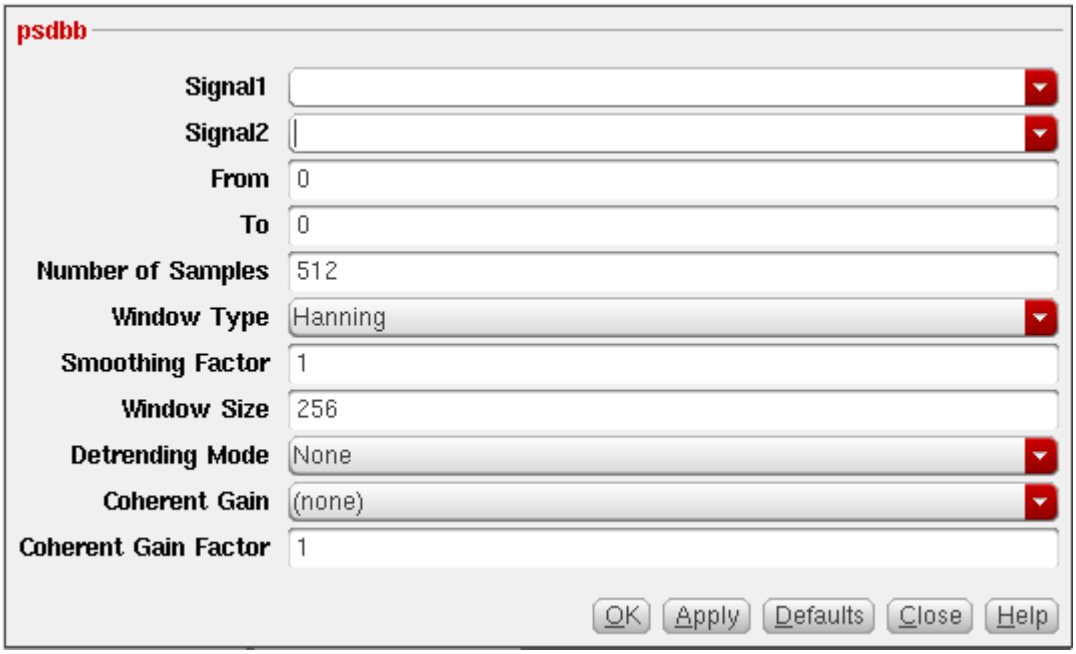
For more information, see `psd` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

psdbb

(Power Spectral Density Baseband)

Returns an estimate for the power spectral density of a `waveform1+j * waveform2`. This function is available only in the SKILL mode.



This function includes the following arguments:

- *Signal1*—First waveform.
- *Signal2*—Second waveform.
- *From*—Starting time for the spectral analysis interval.
- *To*—Ending time for the spectral analysis interval.
- *Number of Samples*—Specifies how many times the domain points are to be used. The maximum frequency in the Fourier analysis is proportional to the *Number of Samples* parameter and inversely proportional to the difference between the starting time and the ending time. The default value is `512`.
- *Window Type*—Window you want to use.
  Default value: `Hanning`
  Valid values: `Blackman`, `BlackmanHarris`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hanning`, `Kaiser`, `Nuttall`, `Parzen`, `Rectangular`, and `Triangular`

© 2023 Cadence Design Systems, Inc.

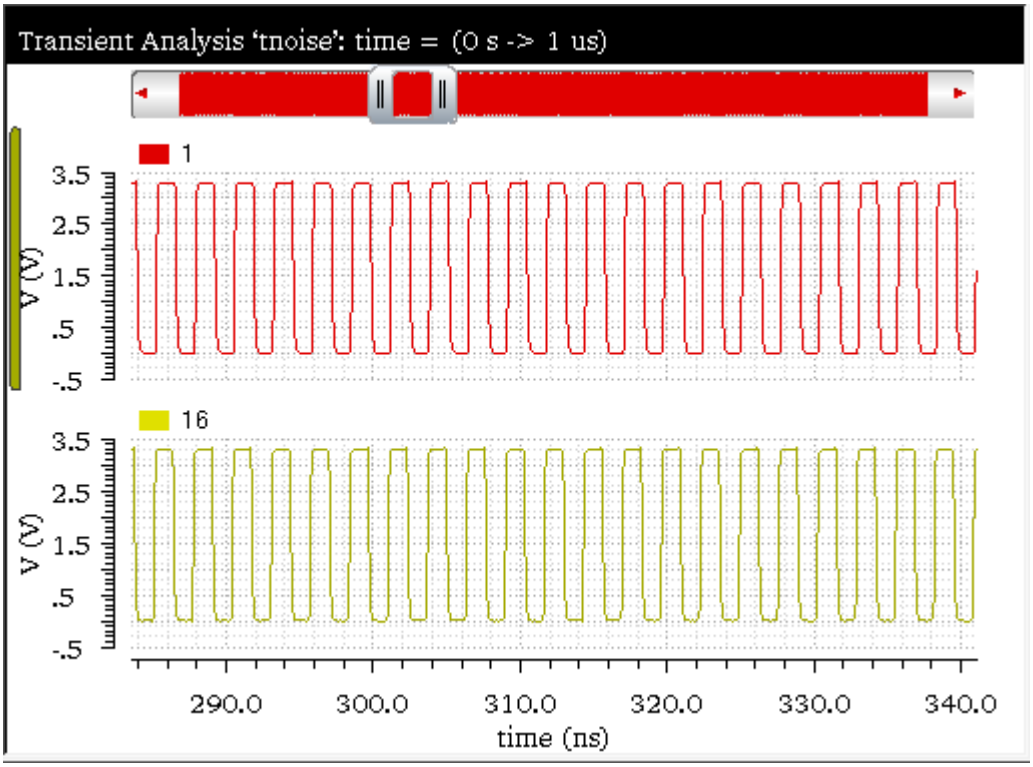
If you select *Kaiser*, type in a value for the Kaiser smoothing factor. The smoothing factor must be in the range 0 <= factor <= 15, where 0 is the same as using a rectangular window.

- *Smoothing Factor*—Specify the smoothing factor only if you have selected the *Window Type* as `Kaiser`.
- *Window Size*—Number of frequency domain points to use in the Fourier analysis. A larger window size results in an expectation operation over fewer samples, which leads to larger variations in the power spectral density. A small window size can smear out sharp steps in the power spectral density that might be present. The default value is `256`.
- *Detrending Mode*—Specifies the trend mode.
  Valid values: `Linear`, `Mean`, `None`
  Default value: `None`
  The `psd` function works by applying a moving windowed FFT to time-series data. If there is a deterministic trend to the underlying data, you may want to remove the trend before performing the spectral analysis. For example, consider analyzing phase noise in a VCO model. Without the noise, the phase increases more or less linearly with time, so it is appropriate to set the detrending mode to `Linear`. To subtract an average value, set the detrending mode to `Mean`. Where the spectrum of raw data is desired, set the detrending mode to `None`.
- *Coherent Gain*—The coherent gain of a window is the zero frequency gain (or the DC gain) of the window. It is calculated by normalizing the maximum amplitude of the window to one and then summing the values of the window amplitudes over the duration of the window. The result is then divided by the length of the window (that is, the number of samples).
  When you do a dft, the applied Window Type (aside from the rectangular type) changes the signal amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.
  Valid Values: `none`, `default`, `magnitude`, `dB20`, or `dB10`
  Default value: `default`
- *Coherent Gain Factor*—This is the scaling factor. A non-zero value that scales the power spectral density by 1/(Coherent Gain).
  Valid values : 0 < coherent_gain_factor < 1. You can use 1 if you do not want this scaling factor to be used.
  Default value : `1`

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from the transient analysis:



The `psdbb` function is applied on this input waveform with the following arguments:

- *Signal1*—`v("16" ?result "tran" ?resultsDir "./nand2_ring_tnoise.raw")`
- *Signal2*—`v("1" ?result "tran" ?resultsDir "./nand2_ring_tnoise.raw")`
- *From*—`0`
- *To*—`1.2u`
- *Number of Samples*—`1024`
- *Window Type*—`Hanning`
- *Smoothing Factor*—`1`
- *Window Size*—`512`
- *Detrending Mode*—`None`

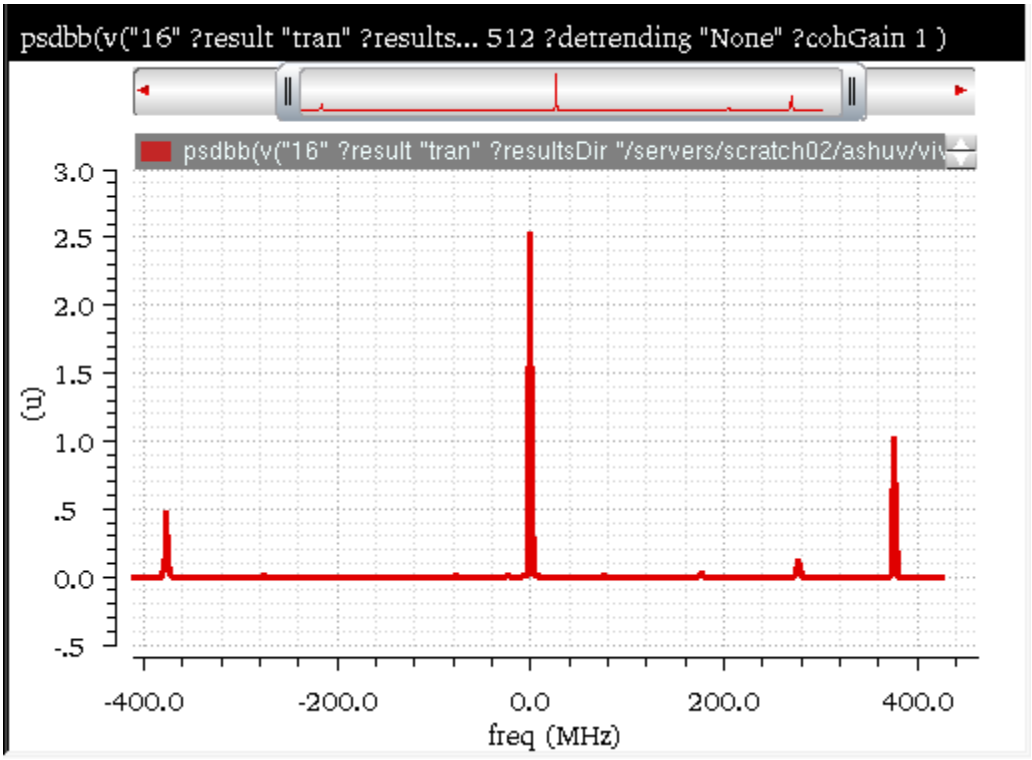
© 2023 Cadence Design Systems, Inc.

- *Coherent Gain*—`(none)`
- *Coherent Gain Factor*—`1`

The corresponding expression created in the Buffer is as follows:

```
psdbb(v("16" ?result "tran" ?resultsDir "./nand2_ring_tnoise.raw")
v("1" ?result "tran" ?resultsDir "./nand2_ring_tnoise.raw") 0 1.2u
1024 ?windowName "Hanning" ?smooth 1 ?windowSize 512 ?detrending
"None" ?cohGain 1 )
```

When you evaluate this expression, the power spectral density is calculated and a spectrum is displayed, as shown in the following output waveform:



Related OCEAN Function

The equivalent OCEAN command for `psdbb` is:

```
psdbb( o_waveform1 o_waveform2 f_timeStart f_timeEnd x_num
     ?windowName t_windowName ?smooth x_smooth ?cohGain f_cohGain
     ?windowsize x_windowsize ?detrending t_detrending)
     => o_waveformReal/nil
```


© 2023 Cadence Design Systems, Inc.

For more information, see `psdbb` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

pstddev

(Periodic Standard Deviation)

Returns the periodic standard deviation of a family of signals for each time point.

This function includes the following arguments:

- *Signal*—Name of the signal for which you want to calculate the periodic standard deviation.
- *From*—Start time from where you want to calculate the periodic standard deviation.
- *To*—End time till you want to calculate the periodic standard deviation.
- *Period*—Difference between the end time and start time. This is an optional argument.
- *Sampling Factor*—Determines the output accuracy. Increase its value in order to increase the accuracy of the outputs.
  Default value is `1`.

Additional Information

The periodic standard deviation is represented by the following equation:

```
pstddev(tk) = stddev(sample( o_waveform(t), from tk, to N*T, linear,
by T))
```

where,

`o_waveform(t)` is the input waveform

`N` = `floor((to-from) / T)`

`T` = period

© 2023 Cadence Design Systems, Inc.

If the input waveform is a multi-dimensional waveform or a family of waveforms, the output is calculated as:

$$\text{pstddev(tk)} = 1/\text{M} \sum_{j=0}^{M-1} \text{pstddev(tk)[o_waveform_family[j]]}$$

where,

`o_waveform_family` is the input waveform family

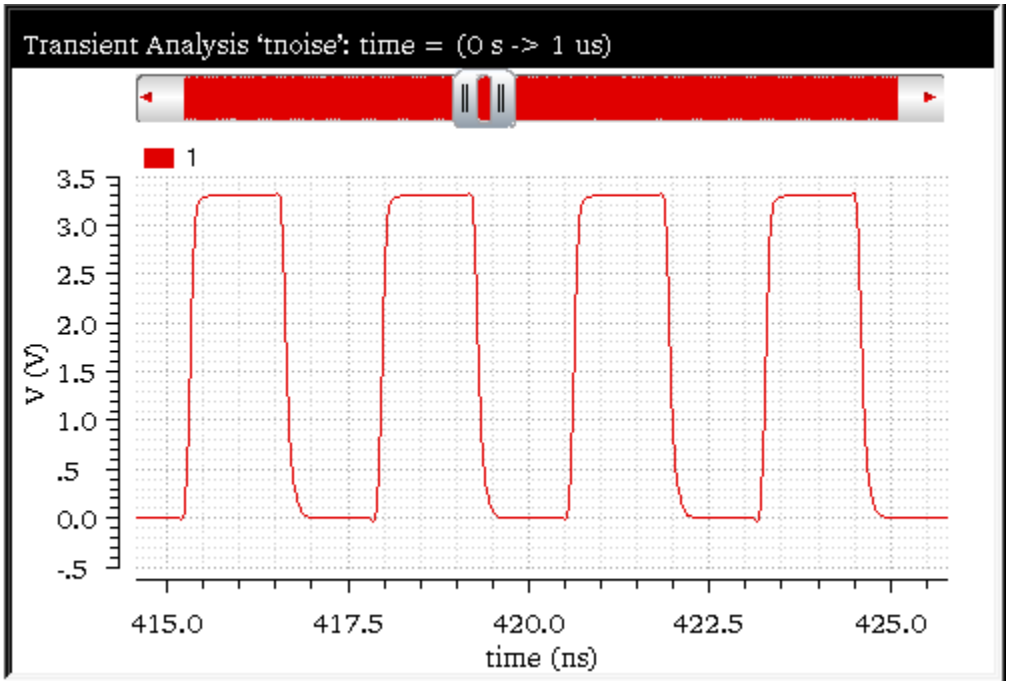
`M` is the number of leaf waveforms in the family.

`o_waveform_family[j]` is the `j`th leaf waveform of the family of waveform represented by the input waveform.

It calculates the `pstddev` on each leaf, and then averages over the number of leaves in the family of waveform. In this case the resultant is not a family of waveforms, but a normal waveform of dimension 1.

Example

Consider the following input waveform from transient analysis:



The `pstddev` function is applied on this input waveform with the following arguments:

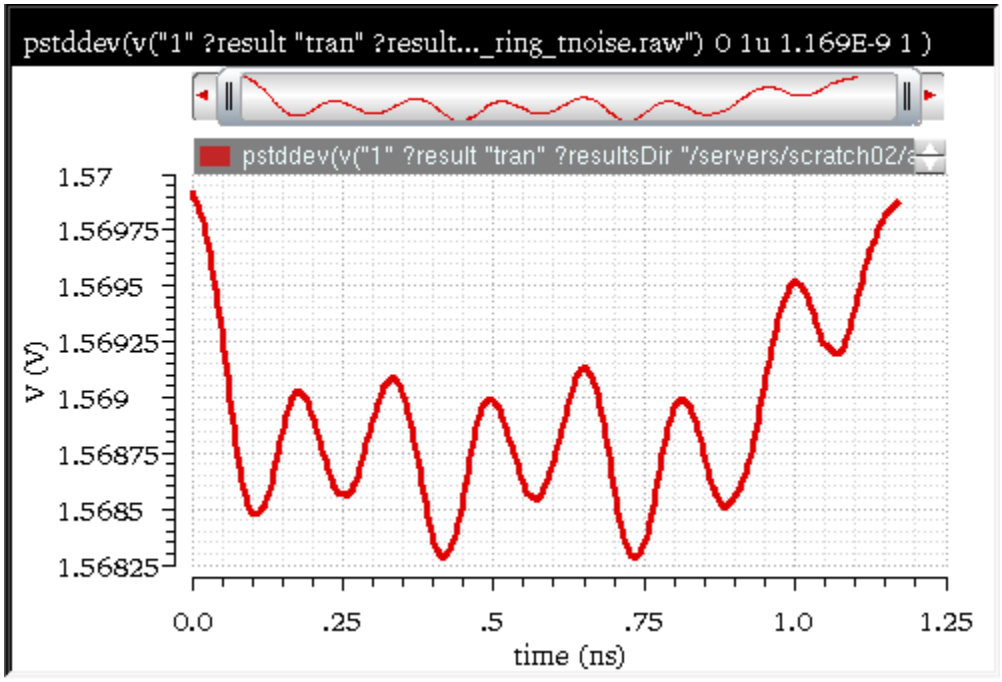
© 2023 Cadence Design Systems, Inc.

- *Signal*—`v("1" ?result "tran" ?resultsDir "./nand2_ring_tnoise.raw")`
- *From*—`0`
- *To*—`1u`
- *Period*—`1.169E-9`
- *Sampling Factor*—`1`

The corresponding expression created in the Buffer is as follows:

```
pstddev(v("1" ?result "tran" ?resultsDir "./nand2_ring_tnoise.raw")
0 1u 1.169E-9 1)
```

When you evaluate this expression, the following output waveform is displayed in a new graph window, which indicates the periodic standard deviation at each time point.



Related OCEAN Function

The equivalent OCEAN command for `pstddev` is:

```
pstddev( o_waveform n_from n_to [n_period [n_sfactor]])
     => o_waveform/nil
```


© 2023 Cadence Design Systems, Inc.

For more information, see `pstddev` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

pzbode

Calculates and plots the transfer function for a circuit from pole zero simulation data.This function also works on the parametric or sweep data.

This function includes the following arguments:

- *DC Gain*—Transfer gain constant.
- *Min. Frequency*—Minimum frequency for the bode plot.
- *Max. Frequency*—Maximum frequency for the bode plot.
- *No. of Points*—Frequency interval for the bode plot, in points per decade.
- *Poles*—Poles from the simulation data.
- *Zeroes*—Zeroes from the simulation data.

Important

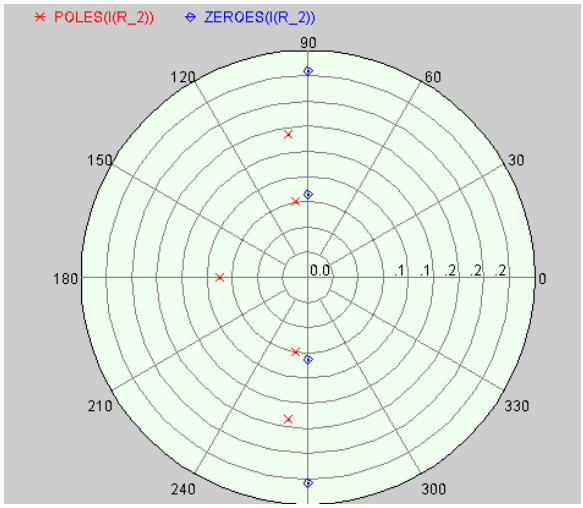
You need to set the database context to the `pz` results directory for which you want to plot the pzbode plot. For more information about how to set a results directory as in-context results directory, see Setting the In-Context Results Directory.

© 2023 Cadence Design Systems, Inc.

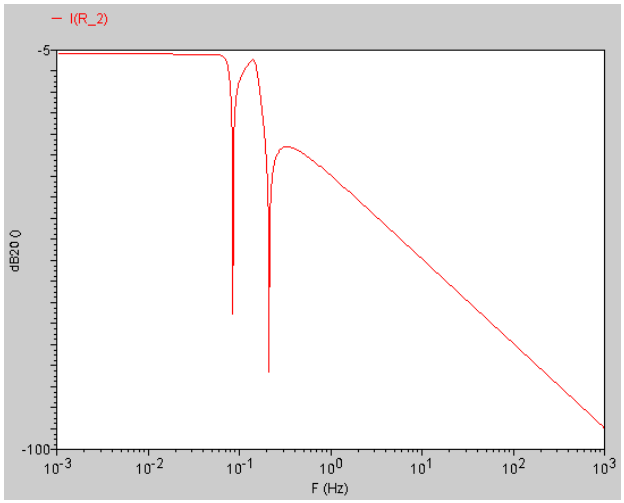
Example 1

The following diagram illustrates how the result with the values *poles*=`POLES<I<R_1>>`, *zeroes*=`ZEROES<I<R_1>>`, *c*=`I<R_1>\[K\]`, *minfreq*=`1e-3`, *maxfreq*=`1e3`, and *npoints*=`1000` is determined.

Polar Plot



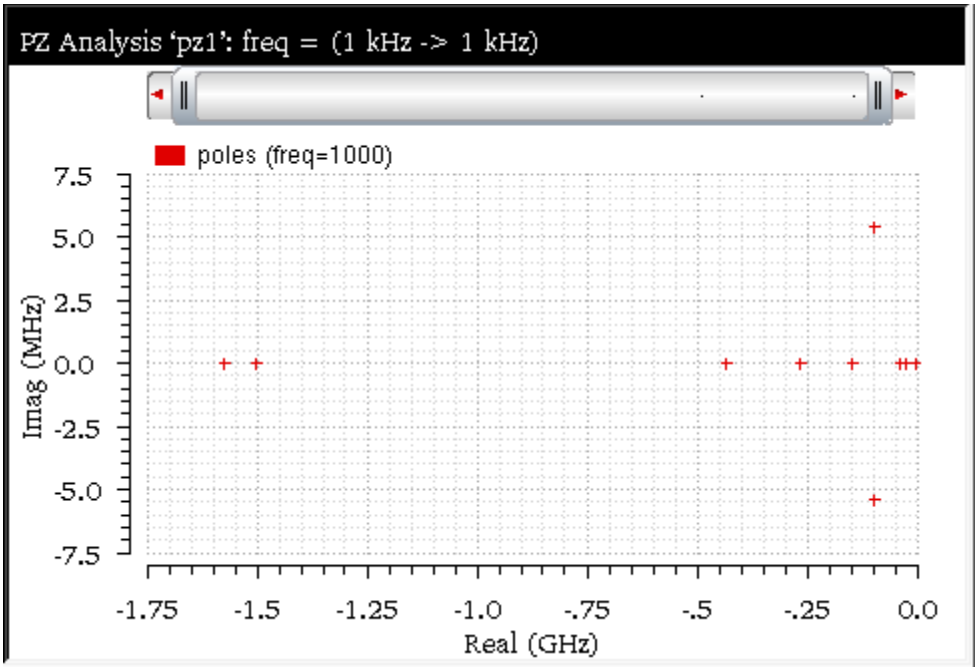
Corresponding bode plot



© 2023 Cadence Design Systems, Inc.

Example 2

This example shows the bode plot generated when you apply the `pzbode` function on the following pole-zero results from the o`penloop_ampSim.raw` database:



The following arguments are specified in this example:

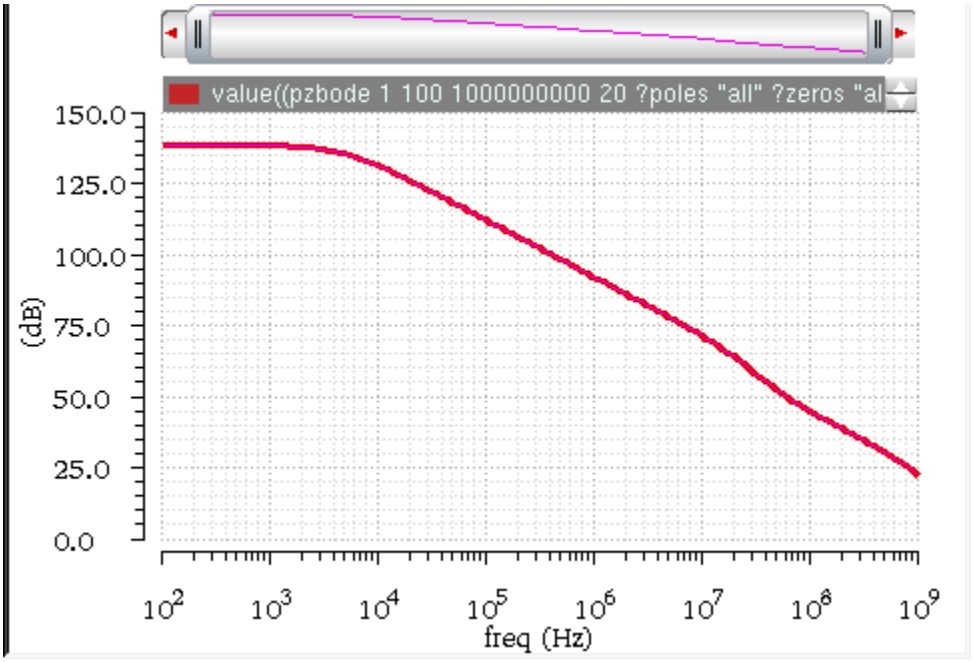
- *DC Gain*—`1`
- *Min. Frequency*—`100`
- *Max. Frequency*—`1G`
- *No. of Points*—`20`
- *Poles*—`all`
- *Zeros*—`all`

The following expression is created in the Buffer:

```
pzbode(1 100 1G 20 ?poles "all" ?zeros "all")
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform showing the bode plot is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `pzbode` is:

```
pzbode( f_c f_minf f_maxf x_nponits ?poles o_waveform1 ?zeros o_waveform2)
     => o_waveform/nil
```

For more information about this OCEAN function, see pzbode in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

pzfilter

Filters the poles and zeroes according to the specified criteria. The `pzfilter` function works only on pole zero simulation data. In addition, this function also works on the parametric or sweep data.

This function includes the followign arguments:

- *maxfreq*—Frequency upto which the poles and zeroes are plotted.
- *reldist*—Relative distance between the pole and zero. Pole-zero pairs with a relative distance lower than the specified value are not plotted.
- *absdist*—Absolute distance between the pole and zero. Pole-zero pairs with an absolute distance lower than the specified value are not plotted.
- *minq*—Minimum Q-factor. Pole-zero pairs with a Q-factor less than the specified value are not canceled. The equations that define the Q-factor of a complex pole or zero are described in the section below.

Note: If you do not specify *maxfreq*, *reldist*, *absdist*, or *minq*, `pzfilter` filters out the poles and zeroes with a frequency higher than 10 GHz (default value of *maxfreq*).

Important

You need to set the database context to the `pz` results directory for which you want to plot the pzfilter plot. For more information about how to set a results directory as in-context results directory, see Setting the In-Context Results Directory.

© 2023 Cadence Design Systems, Inc.

Additional Information

Equations Defining the Q-Factor of a Complex Pole or Zero

$$Re(X) < 0.0 \qquad Q = 0.5 \times \sqrt{[lm(X)/Re(X)]^2 + 1}$$

$$Re(X) = 0 \qquad UNDEFINED$$

$$Re(X) > 0.0 \qquad Q = -0.5 \times \sqrt{[lm(X)/Re(X)]^2 + 1}$$

Filtration Rules

- Real poles can be canceled only by real zeros. A real pole `P` is canceled by a real zero `Z` if the following equation is satisfied:

$$|P - Z| < absdist + \frac{|P + Z|}{2} \times reldist$$

- Complex poles and zeros always occur in conjugated pairs. A pair of conjugated poles can only be canceled by a pair of conjugated zeros. A pole pair `P1=a+jb`, `P2=a-jb` is canceled by a zero pair `Z1=c+jd`, `Z2=c-jd`, if the following equation is satisfied:

$$|P1 - Z1| = |P2 - Z2| = \sqrt{(a-c)^2 + (b-d)^2} < absdist + \frac{|a + c|}{2} \times reldist$$

- Poles in the right-half plane are never canceled because they show the instability of the circuit.

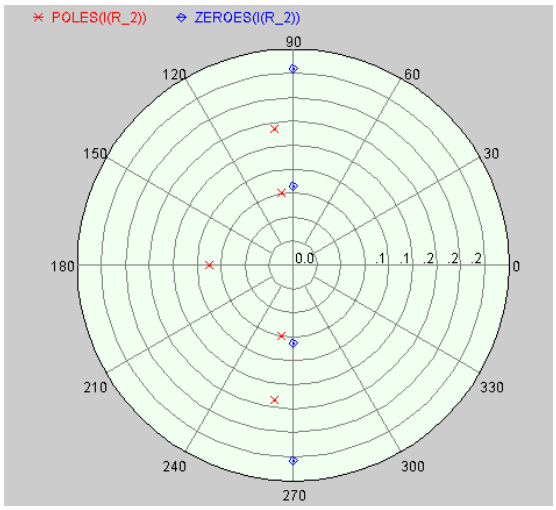
Example

The values *poles*=`POLES<I<R_2>>`, *zeroes*=`ZEROES<I<R_2>>`, *absdist*=`0.05`, and *minq*=`10000` filters pole-zero pairs with a relative distance of less than 0.05 Hz from the plot

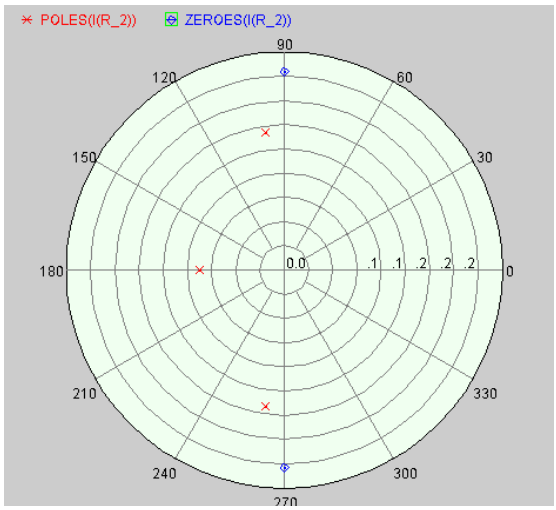
© 2023 Cadence Design Systems, Inc.

on the left side. In the filtered plot shown on the right side, two pole-zero pairs have been filtered out.

Original polar Plot



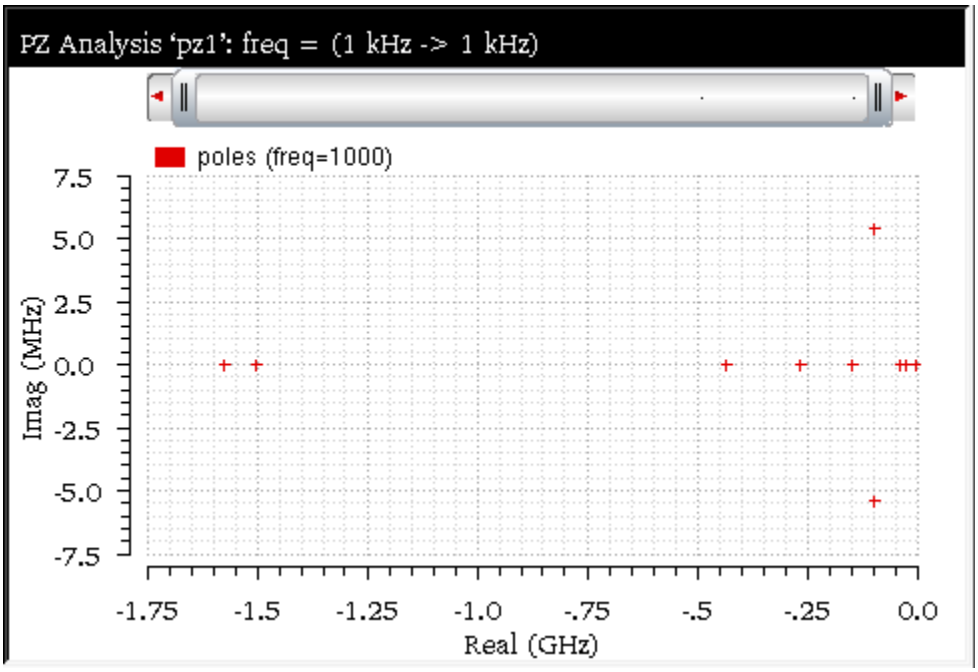
Filtered polar plot



© 2023 Cadence Design Systems, Inc.

Example

This example shows the output plot generated with the filtered pole-zero results when you apply the `pzfilter` function on the following pole-zero data:



Only the following argument is specified in this example:

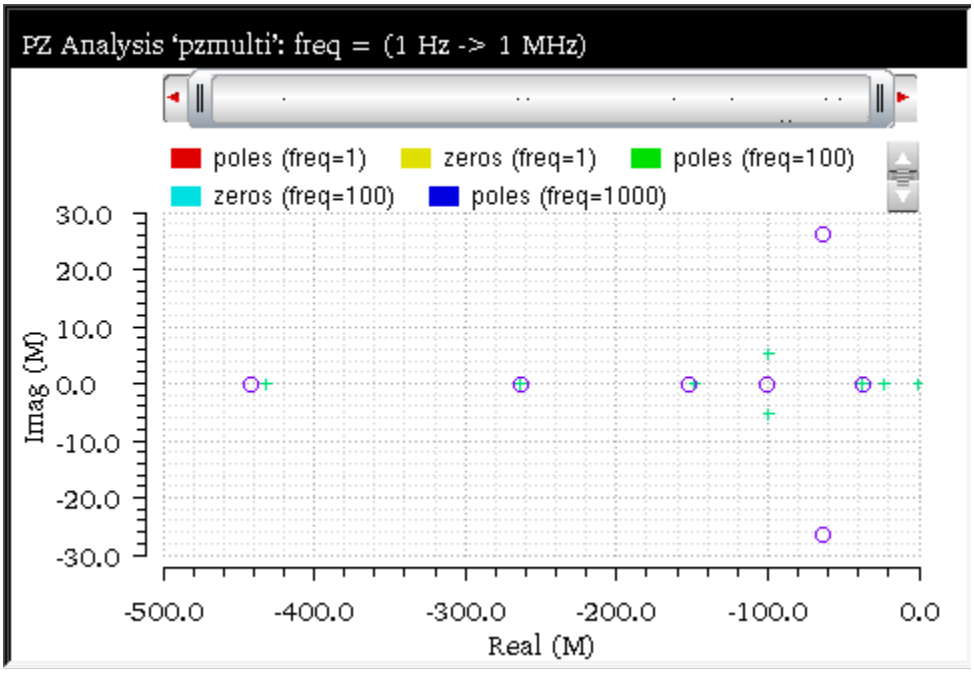
- *maxfreq*—`1G`

The following expression is created in the Buffer:

```
pzfilter(?maxfreq 1G )
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output plot is displayed in the graph window:



Related OCEAN Function

The equivalent OCEAN command for `pzfilter` is:

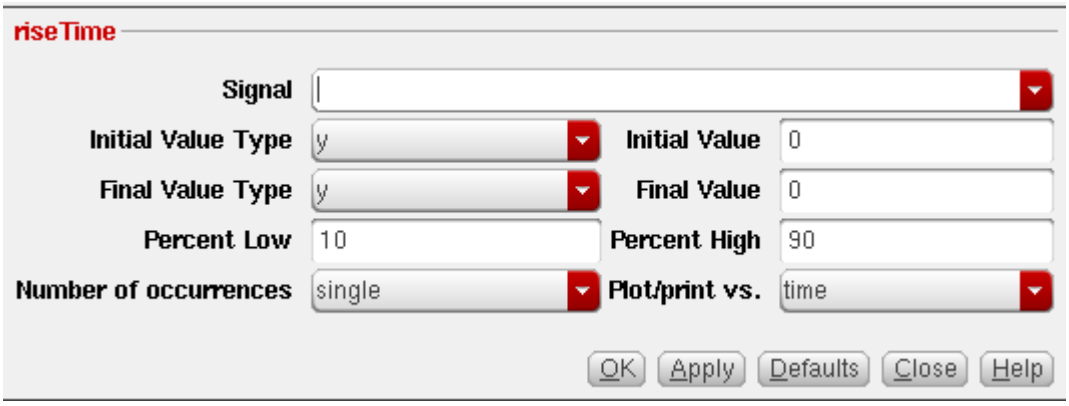
```
pzfilter( [o_PoleWaveform] [o_ZeroWaveform]
     [?maxfreq t_maxfreq] [?reldist n_relDist] [?absdist n_absdist]
     [?minq n_minq] [?output_type o_output] )
     => o_waveform/nil
```

For more information about this OCEAN function, see pzfilter in *OCEAN Reference*.

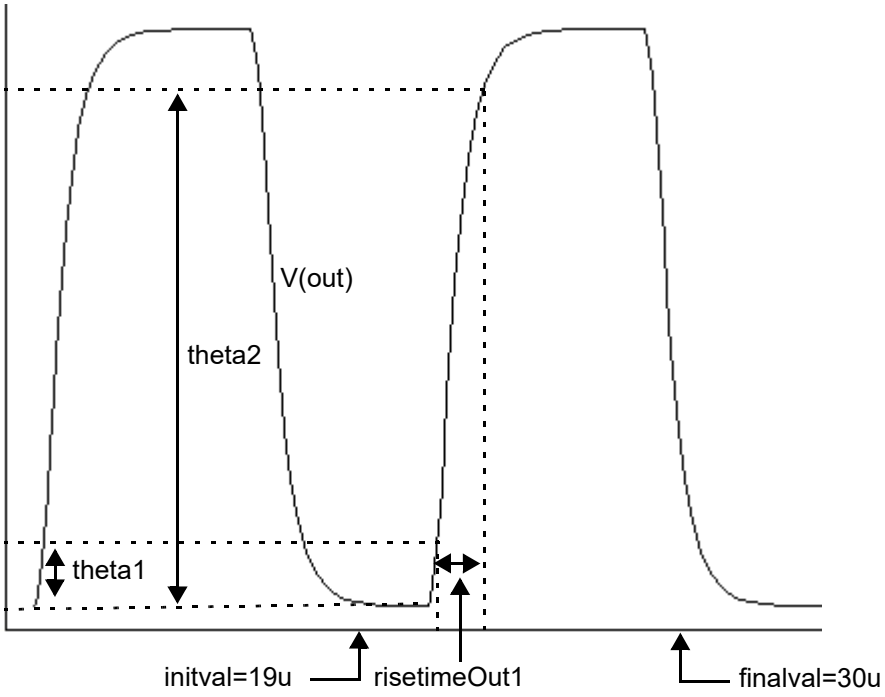
© 2023 Cadence Design Systems, Inc.

riseTime

Returns the rise time for a signal, which is the time taken by a signal to change from a specified low value to a specified high value. Typically, these values are 10% and 90% of the step height.



The following illustration shows how the rise time value is calculated:



This function includes the following arguments in the SKILL mode:

- *Signal*—Name of the signal.

© 2023 Cadence Design Systems, Inc.

- *Initial Value Type*—*S*pecify whether the initial value is the Y-axis value at the specified X-axis value (`y at x`) or Y-axis value (`y`). The default value is `y`.
- *Initial Value*—Value at which the rise time interval is started.
- *Final Value Type*—Specify whether the final value is the Y-axis value at the specified X-axis value (`y at x`) or Y-axis value (`y`). The default value is `y`.
- *Final Value*—Value at which the rise time interval ends.
- *Percent Low*—Percentage low. The default value is `10`.
- *Percent High*—Percentage high. The default value is `90`.
- *Number of Occurences*—Specify whether you want to retrieve only one occurrence of a risetime event for the given waveform (*single*), or all occurrences of risetime for the given waveform which you can later plot or print (*multiple*). The default value is `single`.
- *Plot/print vs*—Specify whether you want to retrieve risetime data against *time* (or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform. The default value is `time`. The value in this field is ignored when you specify *Number of Occurrences* as *single*.

Additional Information

Consider the following equations:

```
Val1 = theta1 / 100.0 * diff + initVal

Val2 = theta2 / 100.0 * diff + initVal
```


© 2023 Cadence Design Systems, Inc.

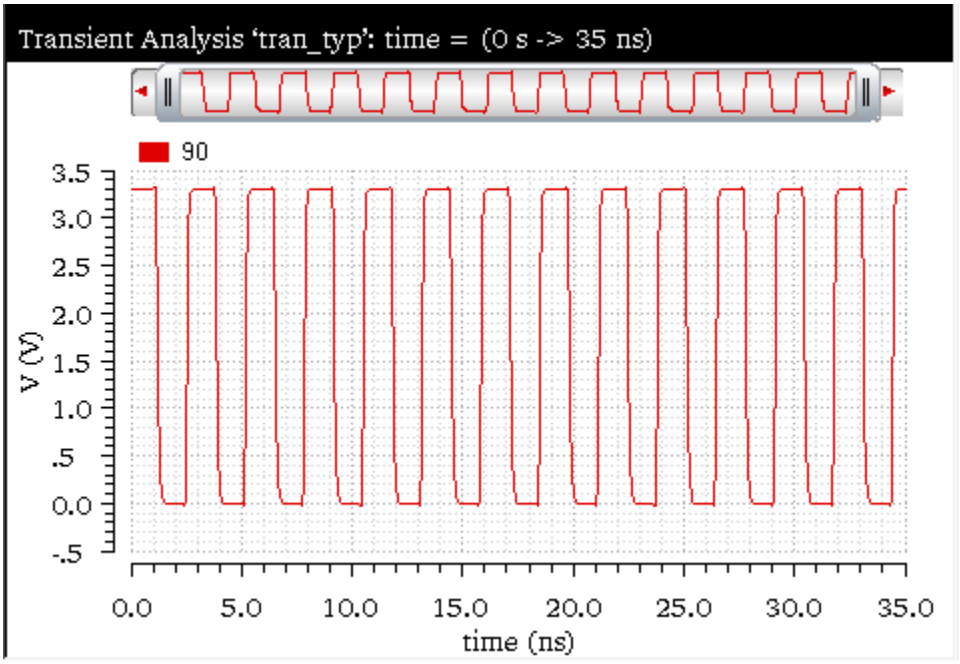
The following table shows how the *riseTime* function works when you apply the above equations:

Function *riseTime(w initVal nil finalVal nil theta1 theta2)*	**initVal**	**finalVal**	**theta1**	**theta2**	**Val1**	**Val2**	**Output**
Case 1	0	1	10	80	0.1	0.8	1.4n (time taken to rise from 200.0p ns to 1.6ns) See Figure for Case 1.
Case 2	0	1	80	10	0.1	0.8	0.7n (time taken to fall from 4.2ns to 4.9ns) See Figure for Case 2.
Case 3	1	0	10	80	0.9	0.2	0.7n (time taken to fall from 4.1ns to 4.8ns) See Figure for Case 3.
Case 4	1	0	80	10	0.2	0.9	1.4n (time taken to rise from 400.0p s to 1.8ns) See Figure for Case 4.

© 2023 Cadence Design Systems, Inc.

Example 3

Consider the following input waveform from transient analysis:



The `riseTime` function is applied on this input waveform with the following arguments:

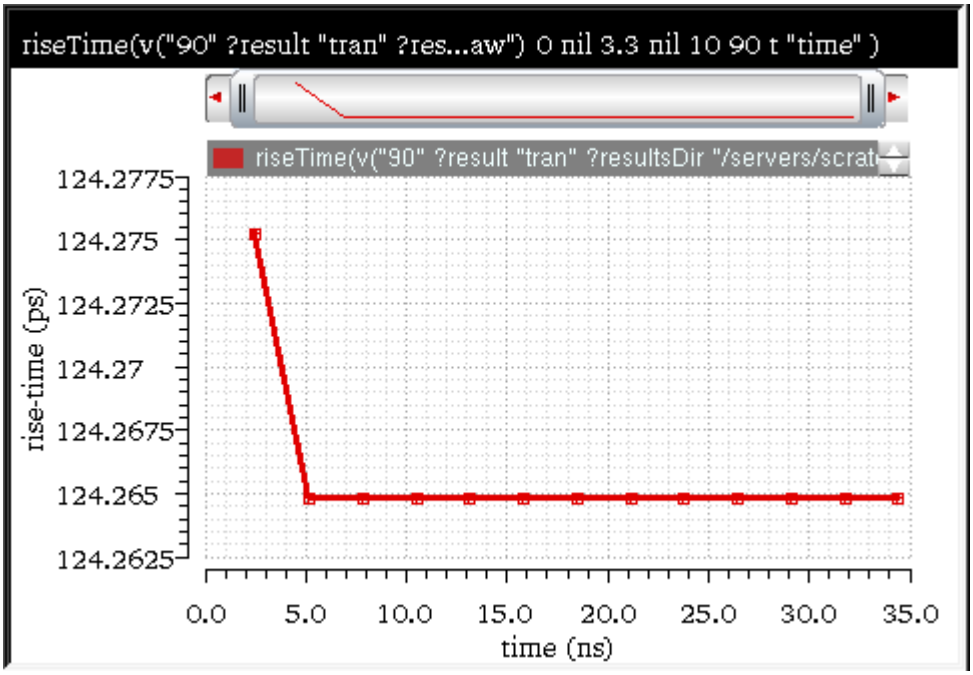
- *Signal*—`v("90" ?result "tran" ?resultsDir "./nand2_ring.raw")`
- *Initial Value Type*—`y`
- *Initial Value*—`0`
- *Final Value Type*—`y`
- *Final Value*—`3.3`
- *Percent Low*—`10`
- *Percent High*—`90`
- *Number of occurrences*—`multiple`
- *Print/plot vs.*—`time`

The corresponding expression created in the Buffer is as follows:

```
riseTime(v("90" ?result "tran" ?resultsDir "./nand2_ring.raw") 0 nil
3.3 nil 10 90 t "time" )
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the rise time is calculated and the following output waveform is displayed in a new graph window:



Related OCEAN Function

The equivalent OCEAN command for `riseTime` is:

```
riseTime( o_waveform n_initVal g_initType n_finalVal g_finalType
     n_theta1 n_theta2
     [g_multiple [s_Xname][g_histoDisplay][x_noOfHistoBins] ] )
     => o_waveform/n_value/nil
```

For more information, see `riseTime` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

rms

(Root Mean Square)

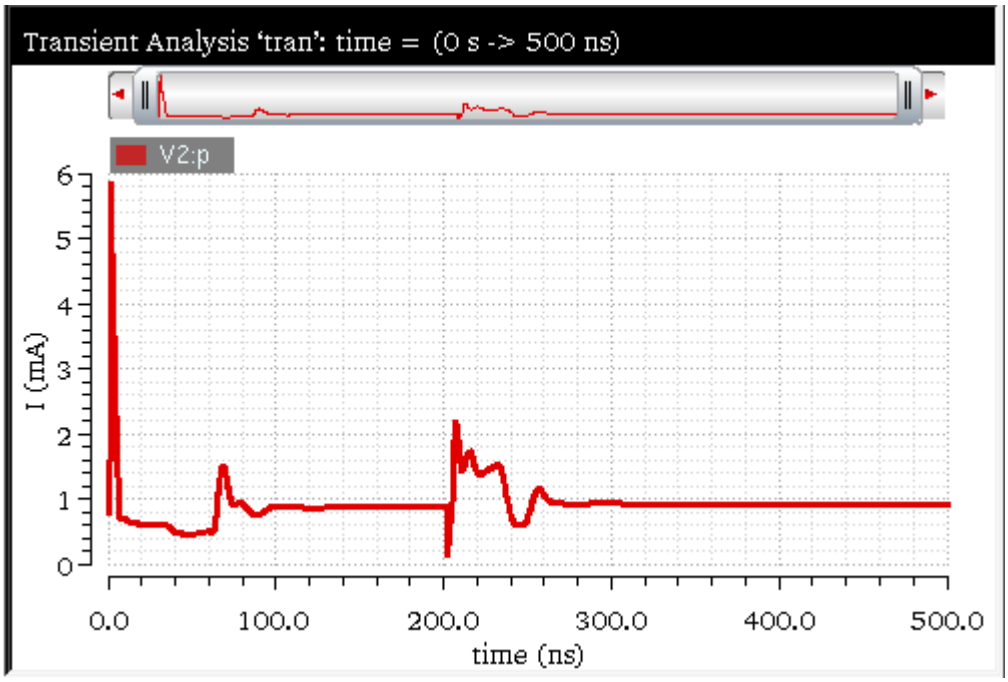
Returns the root mean square of a signal. The equation for `rms` is:

```
rms = sqrt(average(integ(f(x)**2)))
```

Note: For harmonic data, `rms` is calculated by dividing the value at each non-DC frequency by `sqrt(2)`.

Example

Consider the following input waveform from the transient analysis:



When you send this input signal to Calculator and apply the `rms` function, the function is applied on the input waveform expression displayed in the Buffer. The corresponding expression created in the Buffer is as follows:

```
rms(i("V2:p" ?result "tran-tran" ?resultsDir "ampsim.raw")))
```

When you evaluate this expression, the following scalar output is displayed in the Buffer:

```
985.2E-6
```


© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN command for `rms` is:

```
rms( o_waveform )
     => o_waveform/n_value/nil
```

For more information, see `rms` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

rms_jitter

Returns the root mean square jitter value in the specified frequency range. The RMS jitter value is calculated based on the following equation:

```
sqrt(integ(10**(PN/10) fmin fmax))/(2*pi/Tnom)
```

This function includes the following fields:

- *Signal*—Name of the waveform, expression, or family of waveforms. The expression displayed in the Buffer can be added to this field by selecting *buffer*.
- *From*—Starting frequency range.
- *To*—End frequency range.
- *Cross Type*—Points at which the curves of the waveform intersect with the threshold. While intersecting, the curve may be either rising (`rising`) or falling (`falling`). The default value is `rising`.
- *Threshold*—Threshold value against which the rms jitter value is to be calculated. Default value: `nil`.

  Note: A threshold tolerance of `0.1*stddev(signal)` is used when detecting crossings.

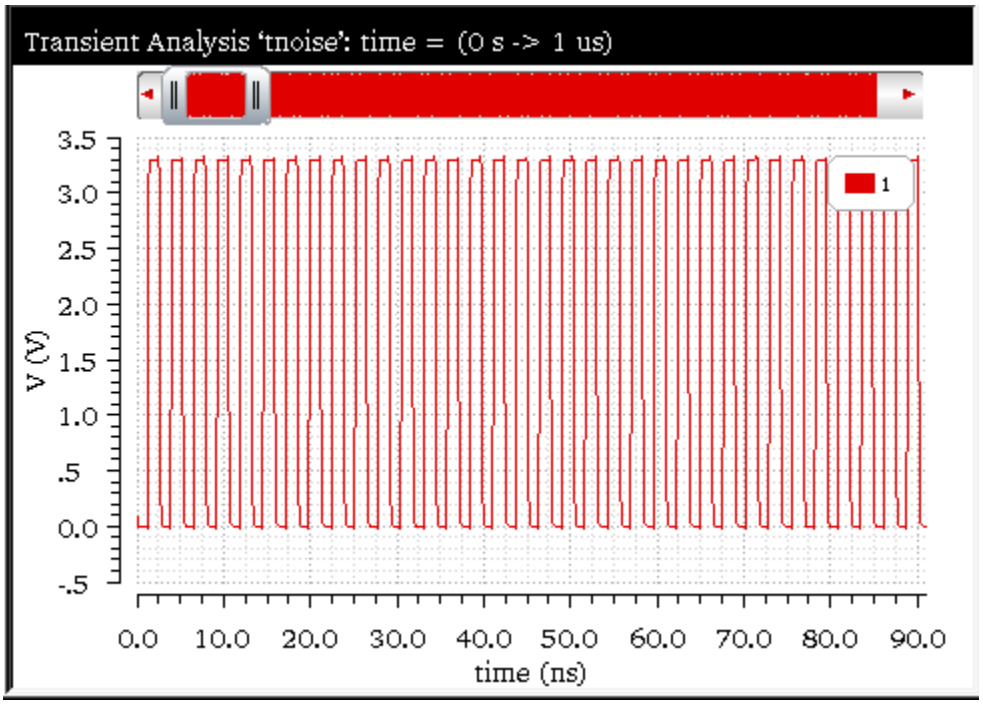
© 2023 Cadence Design Systems, Inc.

- *Tnom*—Nominal time period of the input waveform.
  Default value: `nil`.
- *Window Type*—Window you want to use. The following window types are supported—`Blackman`, `BlackmanHarris`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hanning`, `Kaiser`, `Nuttall`, `Parzen`, `Rectangular`, and `Triangular`. The default window type is `Hanning`.
- *Smoothing Factor*—Smoothing factor is applicable to only Kaiser window. The *Smoothing Factor* field accepts values from `0` to `15`. The value `0` implies no smoothing and is equivalent to a rectangular window. The default value is `1`.
- *Window Size*—Number of frequency domain points to be used while calculating the power spectral density. Default value is `nil`. For more information, see psd.
- *Detrending Mode*—Determines the expected trend for the underlying data while calculating the power spectral density.
  Valid values: `None`, `Mean`, or `Linear`.
  The default value is `Linear`. For more information, see psd.
- *Coherent Gain*—The coherent gain of a window is the zero frequency gain (or the DC gain) of the window. It is calculated by normalizing the maximum amplitude of the window to one and then summing the values of the window amplitudes over the duration of the window. The result is then divided by the length of the window (that is, the number of samples).
  When you do a dft, the applied Window Type (aside from the rectangular type) changes the signal amplitude. Applying a coherent gain factor is a way to get consistent results regardless of the window type.
  Valid Values: `none`, `default`, `magnitude`, `dB20`, or `dB10`
  Default value: `default`
- *Coherent Gain Factor*—This is the scaling factor. A non-zero value that scales the power spectral density by 1/(Coherent Gain).
  Valid values : 0 < coherent_gain_factor < 1. You can use 1 if you do not want this scaling factor to be used.
  Default value : `0`.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following waveform from the transient analysis (Results Directory: `nand2_ring_tnoise.raw`)



The `rms_jitter` function is applied to this waveform with the following arguments:

- *Signal*—`v("1" ?result "tran")`
- *From*—`1M`
- *To*—`376.79M`
- *Cross Type*—`rising`
- *Threshold*—`nil`
- *Tnom*—`nil`
- *Window Type*—`Hanning`
- *Smoothing Factor*—`1`
- *Window Size*—`376`
- *Detrending Mode*—`Linear`
- *Coherent Gain*—`(default)`

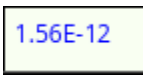
© 2023 Cadence Design Systems, Inc.

■ *Coherent Gain Factor*—`0`

The following expression is created in the Buffer:

```
rms_jitter(v("1" ?result "tran") 1M 376.79M ?crossType "rising"
?windowName "Hanning" ?smooth 1 ?cohGain "default" ?windowSize 376
?detrending "Linear" )
```

When you evaluate this expression, the following `rms_jitter` value is displayed in the Buffer:



In addition, the `Threshold`, `Tnom`, and `Window size` values are displayed in CIW.

```
*Info*    Threshold=1.666e+00
*Info*    Tnom=2.654e-09
*Info*    Window size=376
```

Important

Note the following important points:

- The starting frequency range provided in the *From* field should be the inverse of the time window. For example, in this case, `From=1/Tstop=1/1us=1MHz`. If you provide a value smaller than this, it results in an error. However, the valid values should be smaller than the *To* frequency.
- The end frequency range provided in the *To* field should be half of the oscillator frequency. In this case, `To=0.5/Tnom=376.79MHz`. The larger value results in an error and smaller value is unusual.
- The *Coherent Gain* field should be set as `default`, so that the appropriate window gain factor is applied in PSD calculation.
- The *Window Size* field should not exceed the total number of samples (oscillator cycles) in the waveform: `WindowSize<Tstop/Tnom=376`. A good value is number of samples within the From frequency cycle: `WindowSize = From*Tnom`.

© 2023 Cadence Design Systems, Inc.

rmsNoise

Computes the integrated root-mean-square of the total output noise over the bandwidth specified in Hertz in the *From* and *To* fields.This function is available only in the SKILL mode.

The formula used to calculate `rmsNoise` is as follows:

```
rmsNoise = sqrt(integ(f(x)**2))
```

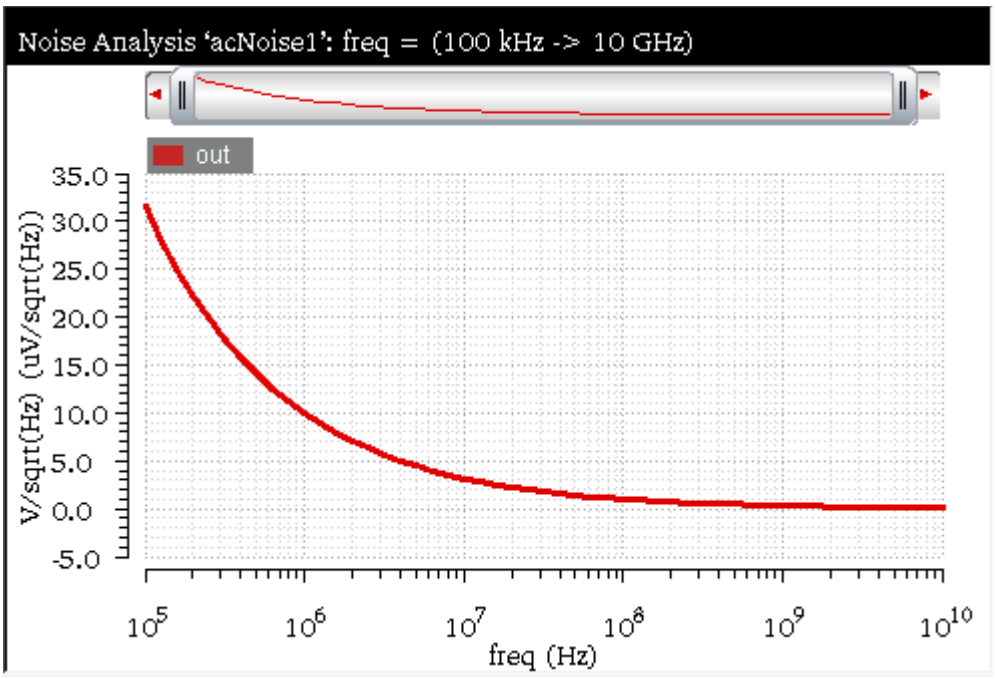
This function includes the following arguments:

- *From*—Starting time for the measurement.
- *To*—Ending time for the measurement.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from the noise analysis: `v("out" ?result "noise" ?resultsDir "./linear.raw")`



Note: Before using this function, you need to set the results database context to the noise analysis results database that contains the input signal, such as `linear.raw` used in this example. For more information about how to change the in-context results directory, see Setting the In-Context Results Directory.

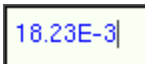
The `rmsNoise` function is applied on this input waveform with the following arguments:

- *From*—`10`
- *To*—`1M`

The corresponding expression created in the Buffer is as follows:

```
rmsNoise(10 1M)
```

When you evaluate this expression, the following scalar output indicating the root mean square is displayed in the Buffer:



© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent command for `rmsNoise` is:

```
rmsNoise( n_from n_to )
     => o_waveform/n_value/nil
```

For more information, see `rmsNoise` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

root

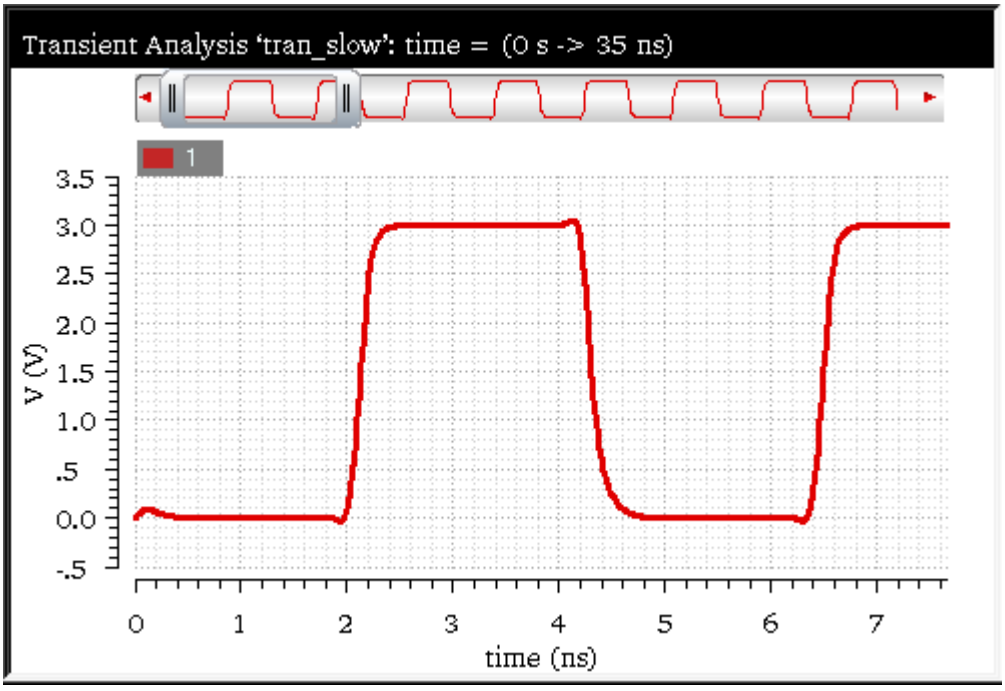
Computes the value of x at which f(x) equals the specified threshold. This function is available only in the SKILL mode.

This function includes the following arguments:

- *Signal*—Name of the signal.
- *Threshold*—Waveform value at which the root value is to be computed.
- *Nth Root*—Root value you want to calculate.

Example

Consider the following input waveform from transient analysis:



© 2023 Cadence Design Systems, Inc.

The `root` function is applied on this input waveform with the following arguments:

- *Signal*—`v("1" ?result "tran_slow-tran" ?resultsDir "./nand2_ring.raw")`
- *Threshold*—`2.5`
- *Nth Root*—`2`

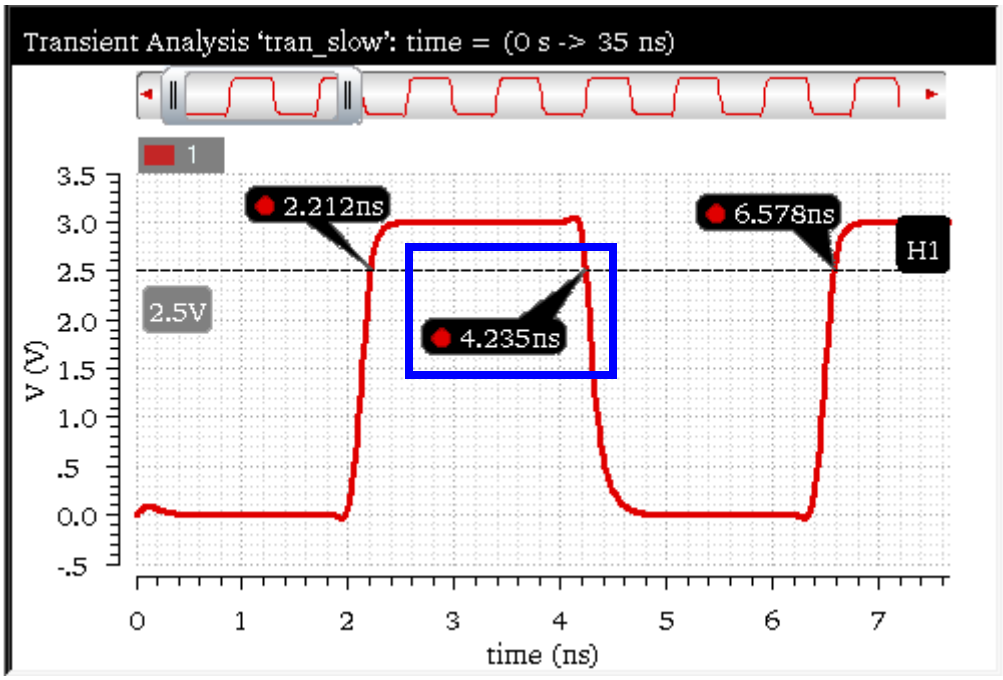
The corresponding expression created in the Buffer is as follows:

```
root(v("1" ?result "tran_slow-tran" ?resultsDir "./nand2_ring.raw")
2.5 2 )
```

When you evaluate this expression, the scalar output is displayed in the Buffer that indicates the second crossing point value at which the input signal crosses the threshold `2.5V`.

```
4.235E-9
```

You can also verify the output generated by placing a horizontal marker in the graph at `Y-axis = 2.5V`, as displayed in the figure below:



Related OCEAN Function

The equivalent OCEAN command for `root` is:

© 2023 Cadence Design Systems, Inc.

```
root( o_waveform n_rootVal x_n )
     => o_waveform/n_value/l_value/nil
```

For more information, see `root` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

rshift

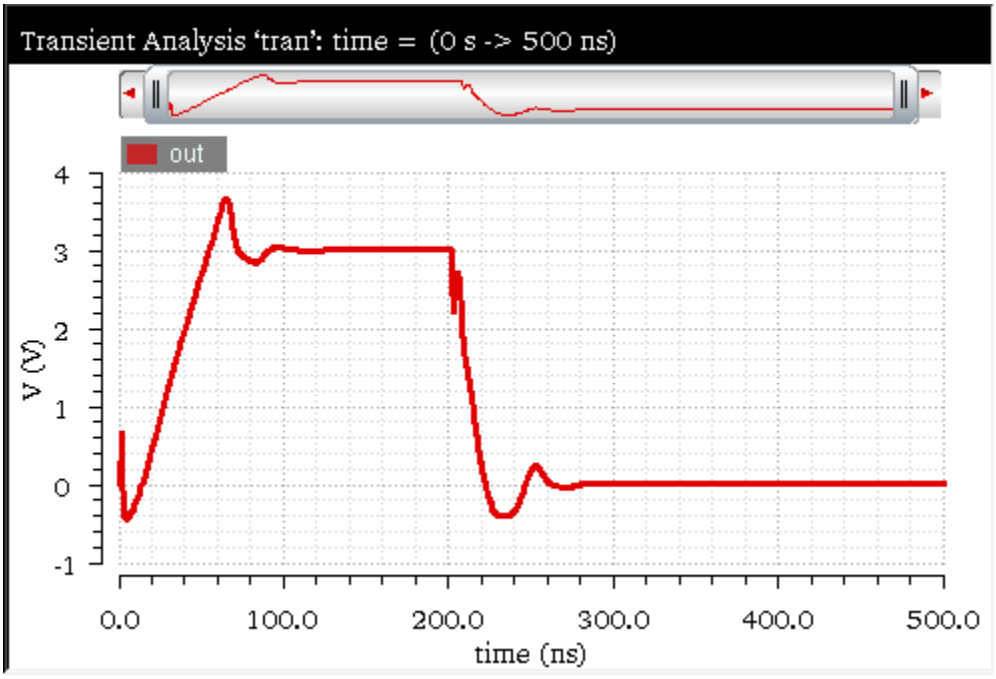
Shifts the data in the graph window to the right by the specified amount. A negative value shifts the data to the left. This function is available only in the SKILL mode.

This function includes the following arguments:

- *Signal*—Name of the signal you want to right shift.
- *Delta X*—Amount on X-axis by which you want to shift the data.

Example

Consider the following input waveform from transient analysis:



The `rshift` function is applied to this input waveform with the following arguments:

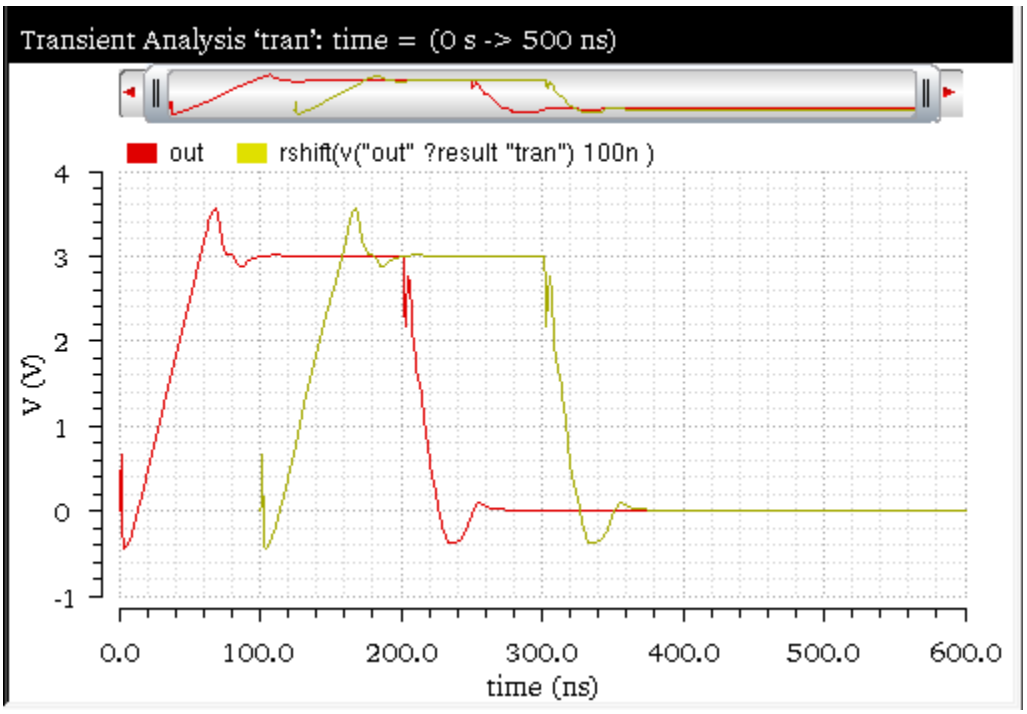
© 2023 Cadence Design Systems, Inc.

- *Signal*—`v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")`
- *Delta X*—`100n`

The corresponding expression created in the Buffer is as follows:

```
rshift(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw") 100n )
```

When you evaluate this expression and plot the result in the same graph window in the append mode, the following output waveform is displayed in the graph window:



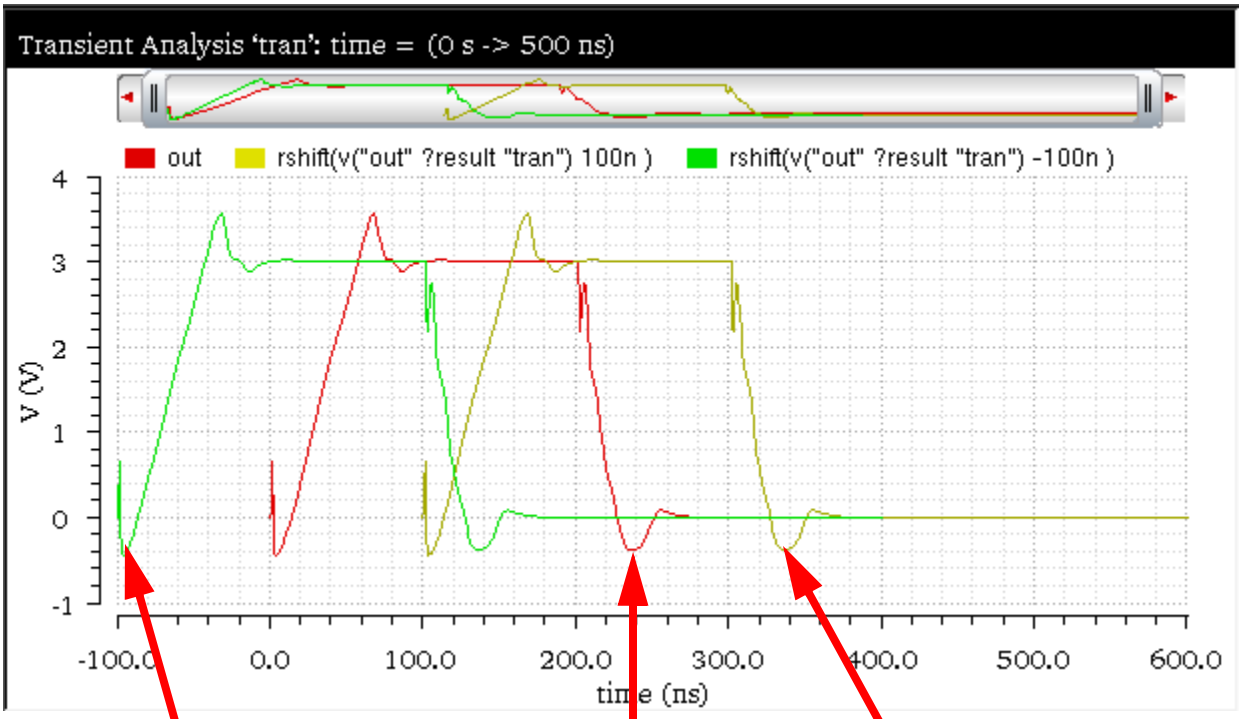
The output waveform starts from time=`100n`, which indicates that the X-axis has been shifted by `100n` from the starting point `0.0n`.

Now, again apply the rshift function on the input signal with Delta=-100. The expression created in the Buffer is as follows:

```
rshift(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw") -100n
)
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression and plot the results in an appned mode, the following output waveform appears in the graph window.



Waveform in green is the second output waveform that has been shifted by -100n on X-axis.

Waveform in red is the input waveform.

Waveform in yellow is the first output waveform that has been shifted by 100n on X-axis.

The output waveform starts from time=`-100n`, which indicates that the X-axis has been shifted by `-100n` from the starting point `-100n`.

Related OCEAN Function

The equivalent OCEAN command for `rshift` is:

```
rshift( o_waveform n_delta )
     => o_waveform/nil
```

For more information, see `rshift` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

sample

Samples the given waveform over the specified interval.

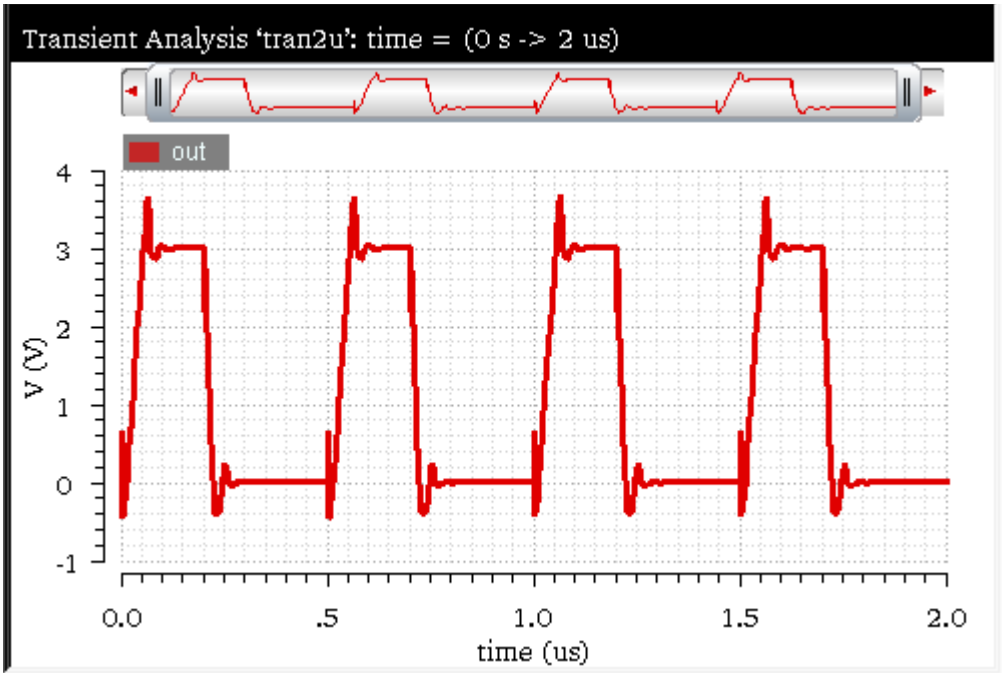
This function includes the following arguments in SKILL mode:

- *Signal*—Name of the signal.
- *From*—X-axis value at which the sampling begins.
- *To*—X-axis value at which the sampling stops
- *Type*—Specifies whether the sample should be linear or logarithmic. The default value is `linear`.
- *By*—Specifies the step size for the sample (if *type* is `linear`) or the points per decade (if *type* is `logarithmic`).

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input signal from the transient analysis:



The `sample` function is applied on this input waveform with the following arguments:

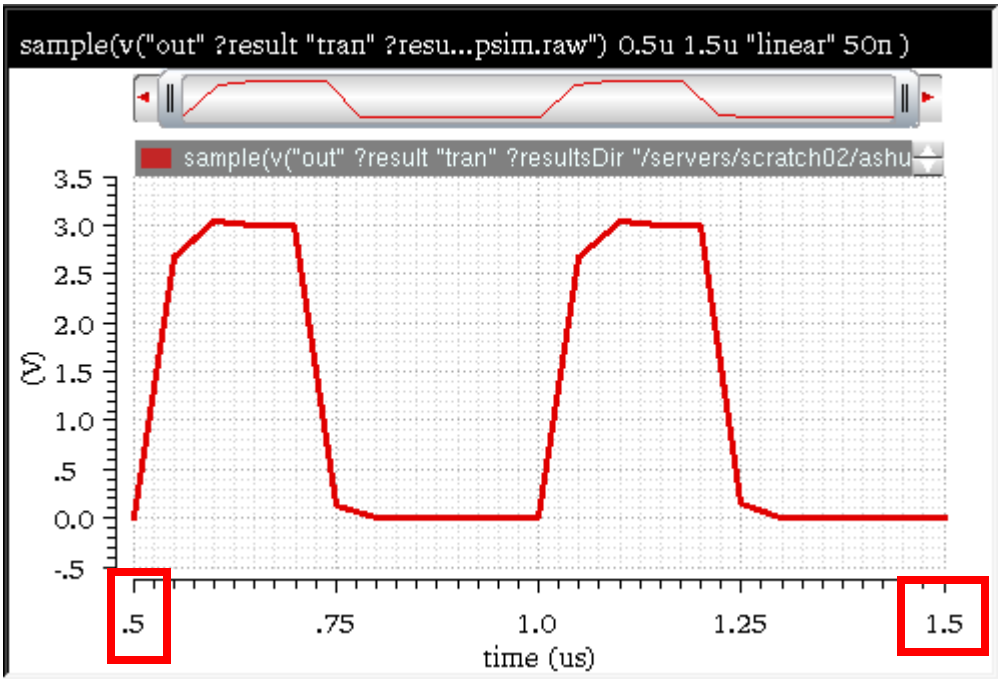
- *Signal*—`v("out" ?result "tran" ?resultsDir "./ampsim.raw")`
- *From*—`0.5u`
- *To*—`1.5u`
- *Type*—`linear`
- *By*—`50n`

The corresponding expression created in the Buffer is as follows:

```
sample(v("out" ?result "tran" ?resultsDir "/ampsim.raw") 0.5u 1.5u
"linear" 50n )
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the sample of the signal is calculated in the X-axis range `0.5u` to `1.5u`. The following output waveform is displayed in a new graph window:



Related OCEAN Function

The equivalent OCEAN command for `sample` is:

```
sample( o_waveform n_from n_to t_type n_by )
     => o_waveform/n_number/nil
```

For more information, see `sample` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

settlingTime

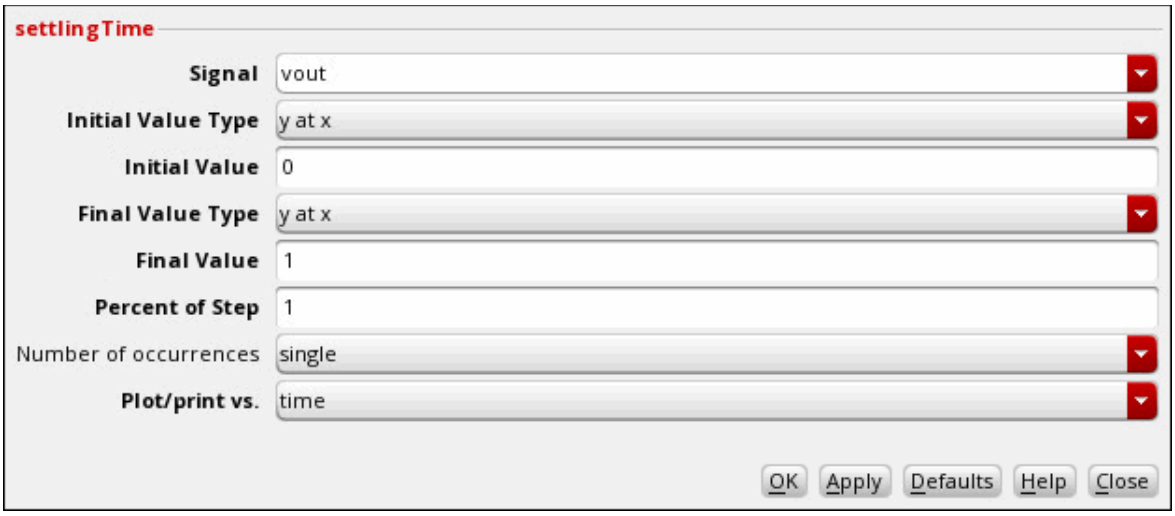
Calculates the time taken by a signal before it reaches to a value that remains within a certain percent of its final value.

The formula to calculate maximum deviation is:
`maxDeltaY = ((theta/100.0) * abs(FinalVal -InitVal))`
where, `theta` is the percentage of step, which is used as one of the function arguments that are passed to calculate the settling time.

For example, if the initial value of the signal is 0V, final value of the signal is 1V (1000mV), and the percent of step is 1:

```
maxDeltaY = ((1/100.0) * (1 -0)) = 0.01V = 10 mV
```

And the range within which the signal has to remain after the settling time = Final value± maxDeltaY = 1000±10 = 990mV to 1010mV



This function includes the following arguments in the SKILL mode:

- *Signal*—Name of the signal.
- *Initial Value Type*—Specifies whether the initial value is the y-axis value at the specified x-axis value (`y at x`) or the absolute y-axis value (`y`). The default value is `y`.
- *Initial Value*—Starting value for the measurement.
- *Final Value Type*—Specifies whether the final value is the y-axis value at the specified x-axis value (`y at x`) or the absolute y-axis value (`y`). The default value is `y`.
- *Final Value*—Final value for the measurement.

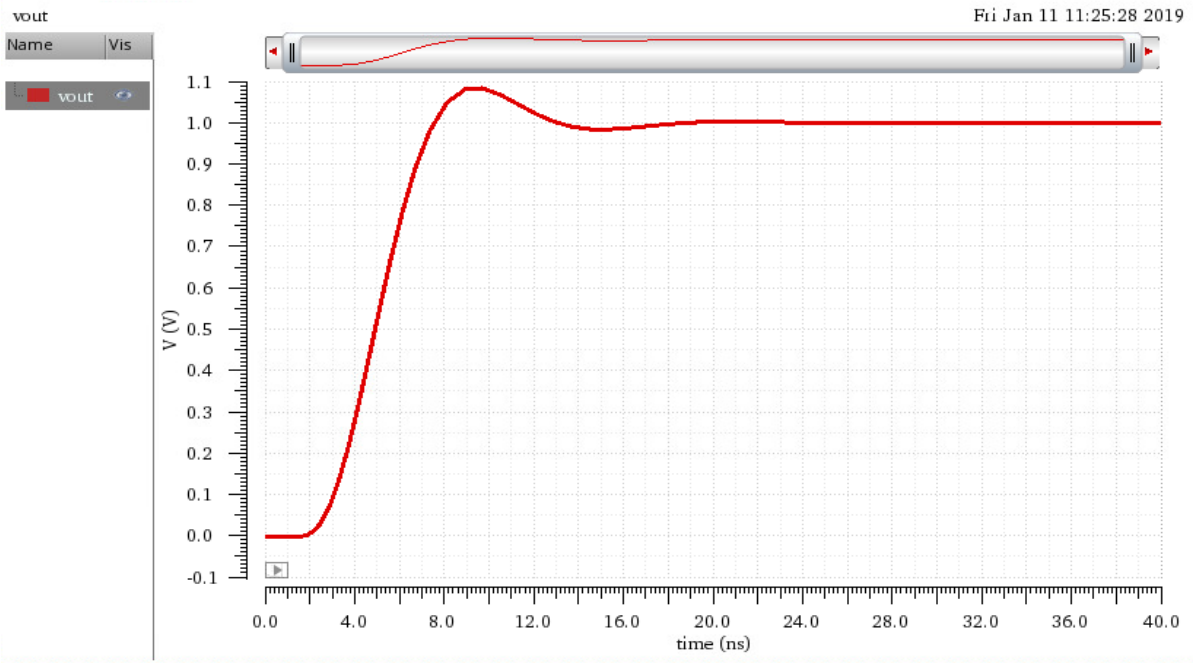
© 2023 Cadence Design Systems, Inc.

- *Percent of Step*—Percentage (*Final value - Initial Value*) within which the signal has to settle. The default value is `5`.
- *Number of Occurences*—Specifies whether you want to retrieve only one occurrence of a settling time event for the given waveform (*single*), or all occurrences of settling time for the given waveform which you can later plot or print (*multiple*). The default value is `single`.

  Note: When *Number of Occurrences* is selected as *multiple*, it is possible that the last settling time value returned by this function is outside of the specified range.
- *Plot/print vs.*—Specifies whether you want to retrieve settling time data against *time* (or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform. The default value is `time`. The value in this field is ignored when you specify *Number of Occurrences* as *single*.

Example 1

Consider the following example in which `settlingTime` function is applied on the `vout` signal as shown in the figure below:



The initial and final values of this signal are 0 (at time, t = 0ns) and 1 (at time, t = 40ns), respectively. If the settling time for this signal is calculated at percent of step = 1, the signal must remain within 990mV to 1010mV after reaching its settling time.

© 2023 Cadence Design Systems, Inc.

Apply `settlingTime` function on this signal with the following argument values:

- *Signal*—`vout`
- *Initial Value Type*—`y at x`
- *Initial Value*—`0`
- *Final Value Type*—`y at x`
- *Final Value*—`1`
- *Percent of Step*—`1`
- *Number of occurrences*—`single`
- *Plot/print vs.*—`time`

The corresponding expression created in the Buffer is as follows:

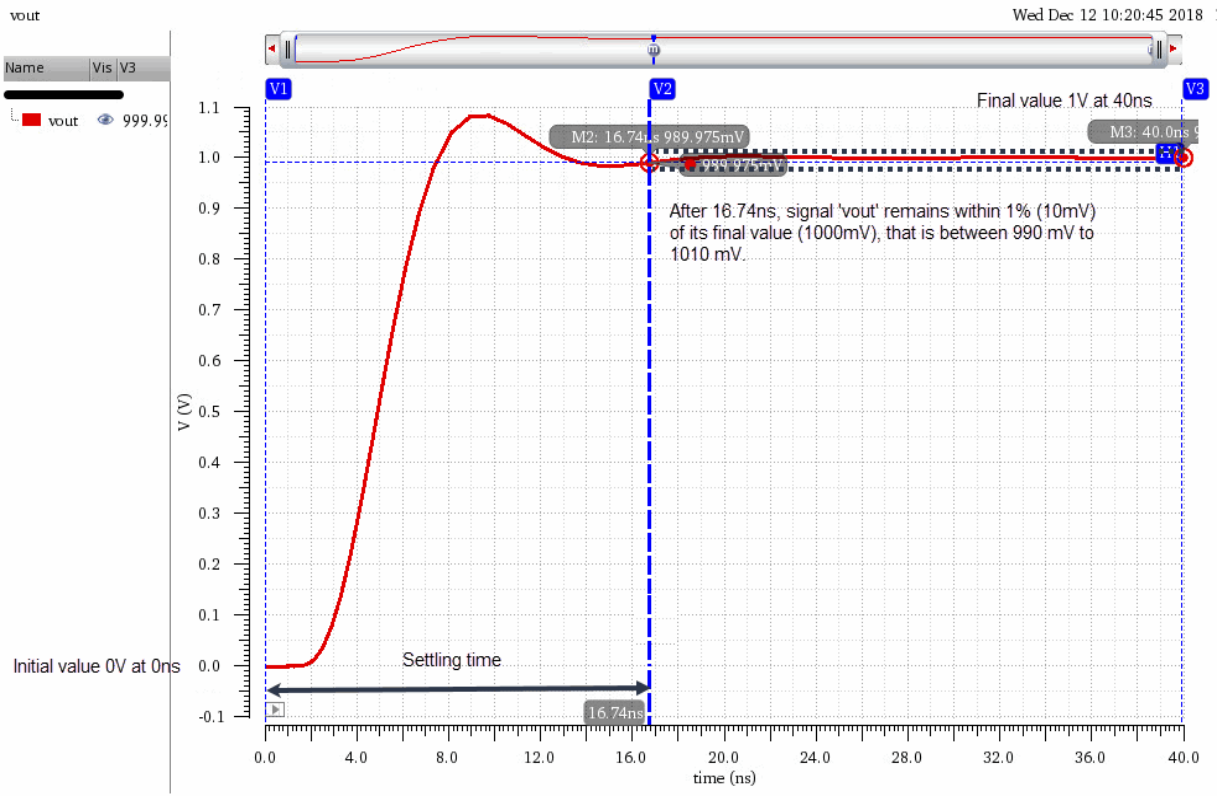
```
settlingTime(vout 0 t 1 t 1 nil "time")
```

Now, when you evaluate this expression, the following scalar value is displayed in the Buffer that indicates the time required by this signal to settle down.

```
16.74E-9
```

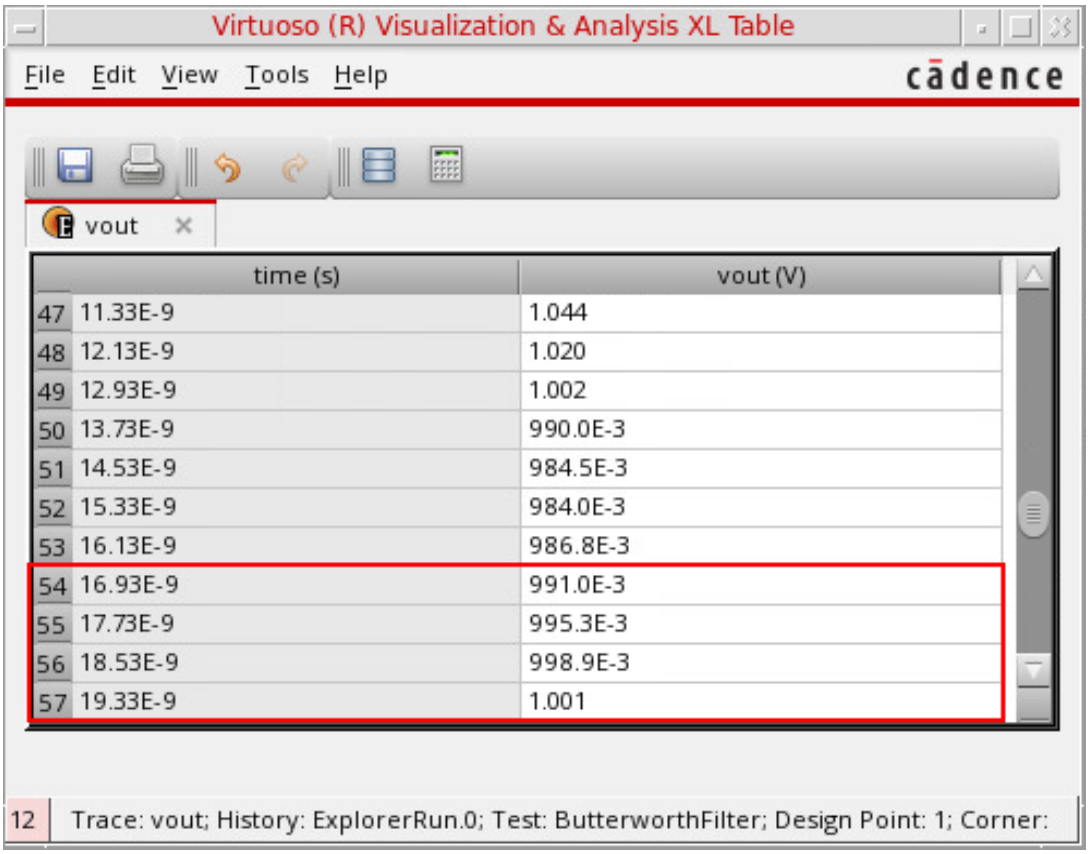

© 2023 Cadence Design Systems, Inc.

See the following figure to observe that after time, t = 16.74ns, the signal remains within 990mV to 1010mV range.



© 2023 Cadence Design Systems, Inc.

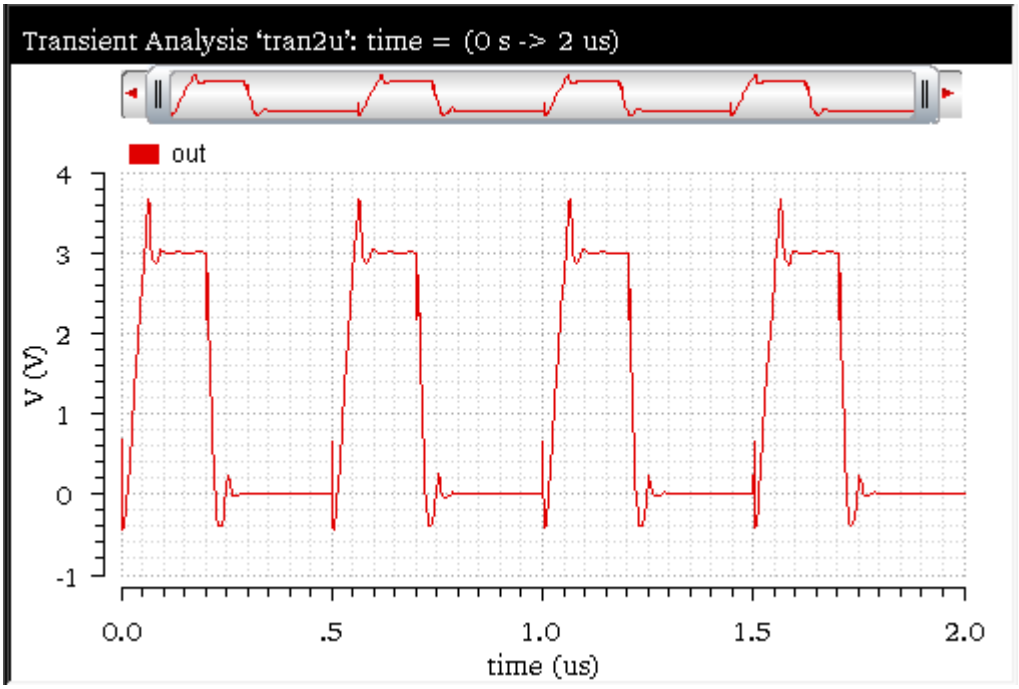
You can also analyze the signal by sending it to the table. To do so, right-click the signal and choose *Send To – Table – New Window*. The signal values are displayed in the table as shown in the following figure.



© 2023 Cadence Design Systems, Inc.

Example 2

Consider the following input waveform from the transient analysis:



The `settlingTime` function is applied on this input waveform with the following arguments:

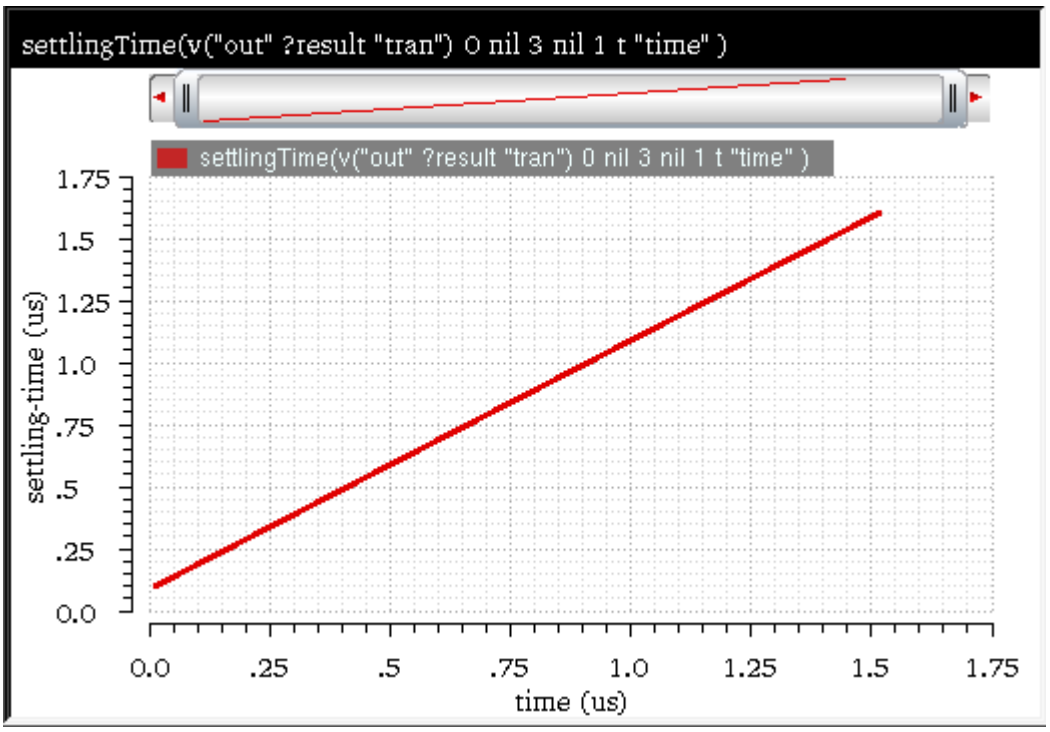
- *Signal*—`v("out" ?result "tran" ?resultsDir "./ampsim.raw"))`
- *Initial Value Type*—`y`
- *Initial Value*—`0`
- *Final Value Type*—`y`
- *Final Value*—`3`
- *Percent of Step*—`1`
- *Number of occurrences*—`multiple`
- *Plot/print vs.*—`time`

The corresponding expression created in the Buffer is as follows:

```
settlingTime(v("out" ?result "tran") 0 nil 3 nil 1 t "time" )
```

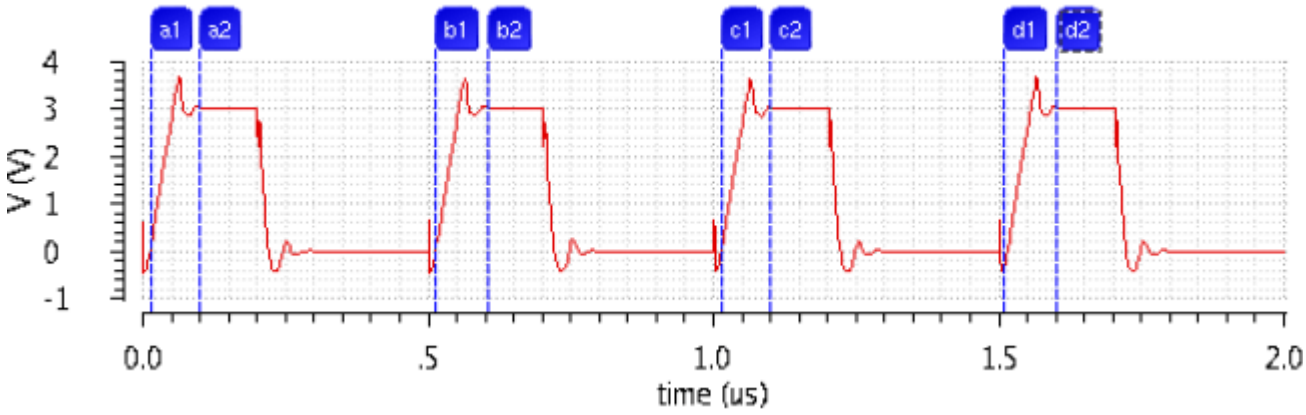

© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the settling time is calculated and following output waveform is displayed in the graph window:



The above example calculates multiple settling time values because the *Number of occurrences* field has been set to `multiple`.

Perform the steps listed below to measure the individual settling times for the curves, such as `a2-a1,...,d2-d1` marked on the input waveform shown below,



© 2023 Cadence Design Systems, Inc.

1. Apply the `xval` function on the settling time waveform expression to measure the X values from the settling time signal. The corresponding expression created in the Buffer is as follows:

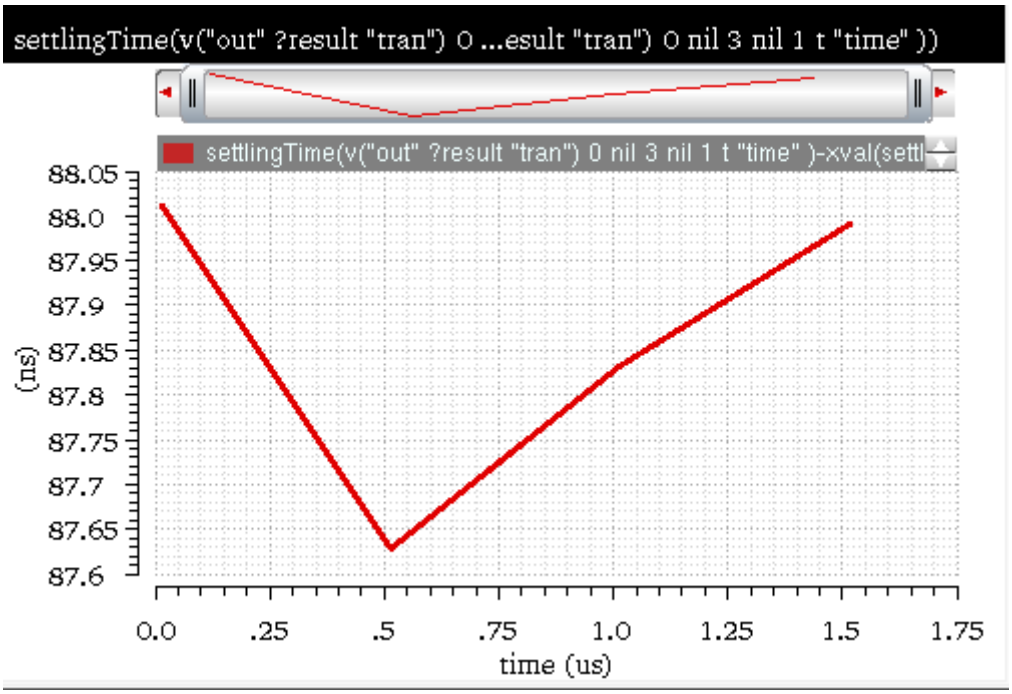
   ```
   xval(settlingTime(v("out" ?result "tran") 0 nil 3 nil 1 t "time"
   ))
   ```

2. Calculate the difference between the two waveforms by subtracting the two waveforms. The corresponding expression created in the Buffer is as follows:

   ```
   settlingTime(v("out" ?result "tran") 0 nil 3 nil 1 t "time" )-
   xval(settlingTime(v("out" ?result "tran") 0 nil 3 nil 1 t "time"
   ))
   ```

3. Evaluate the expression.

   The following output waveform appears in the graph window that indicates the individual settling time for each curve in the waveform.



Related OCEAN Function

The equivalent OCEAN command for `settlingTime` is:

```
settlingTime( o_waveform n_initVal g_initType n_finalVal g_finalType
     n_theta [g_multiple [s_Xname]] )
     => o_waveform/n_value/nil
```


© 2023 Cadence Design Systems, Inc.

For more information, see `settlingTime` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

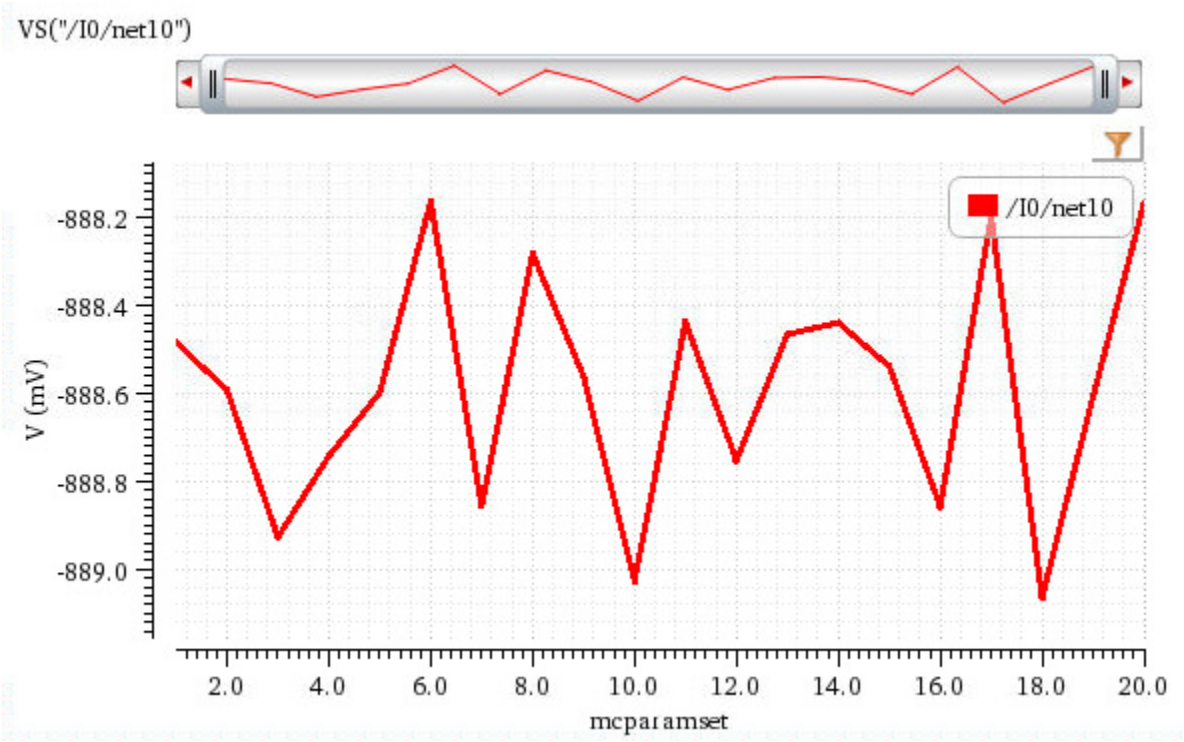
kurtosis

Calculates the kurtosis of the specified waveform.

Kurtosis is a measurement of whether the data is heavy-tailed or light-tailed relative to a normal distribution. That is, data sets with high kurtosis tend to have heavy tails or outliers. Data sets with low kurtosis tend to have light tails or lack of outliers.

Example

This example shows how the kurtosis is calculated when you apply the `kurtosis` function on the following input signal:



When you apply the `kurtosis` function on this signal, the following expression is created in the Buffer:

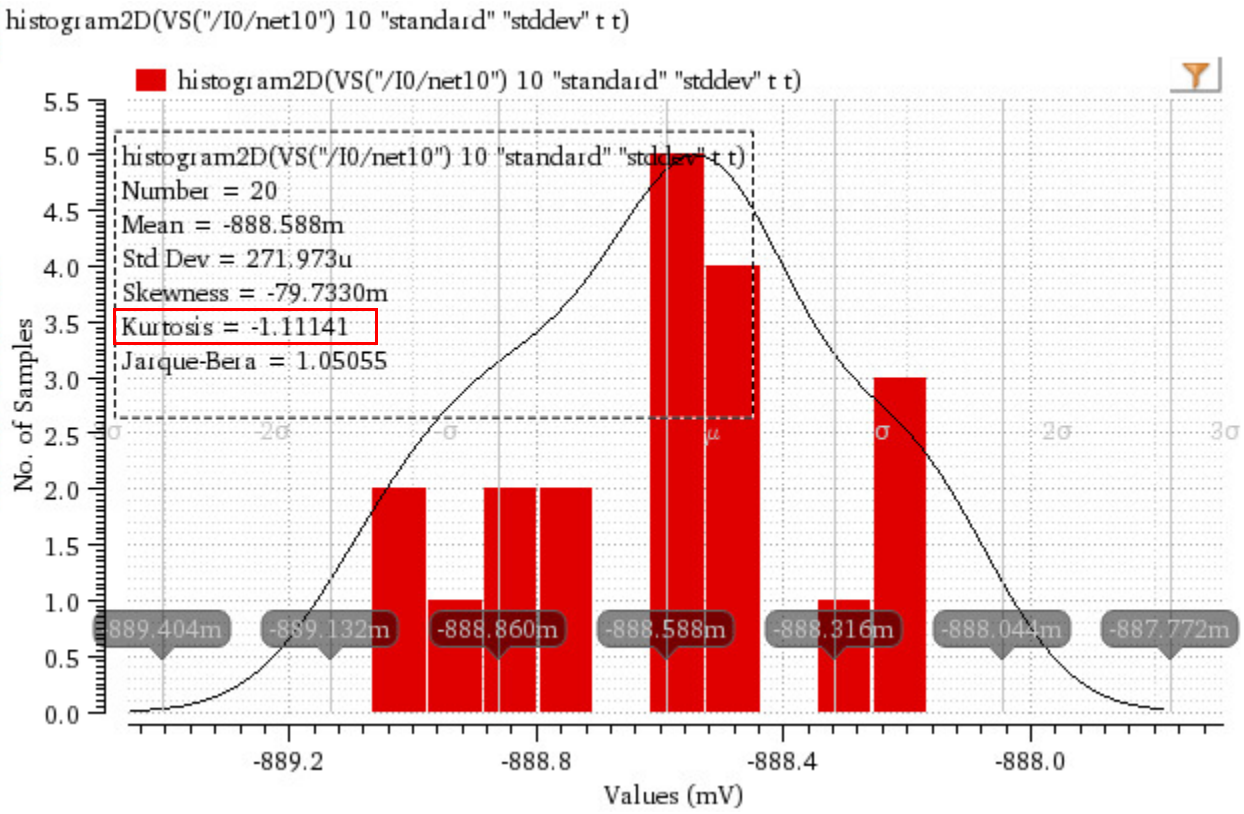
```
kurtosis(VS("/I0/net10"))
```

When you evaluate this expression, the following output is displayed in the Buffer:

```
-1.111
```


© 2023 Cadence Design Systems, Inc.

You can also annotate kurtosis on the histogram plotted using the same input signal.



© 2023 Cadence Design Systems, Inc.

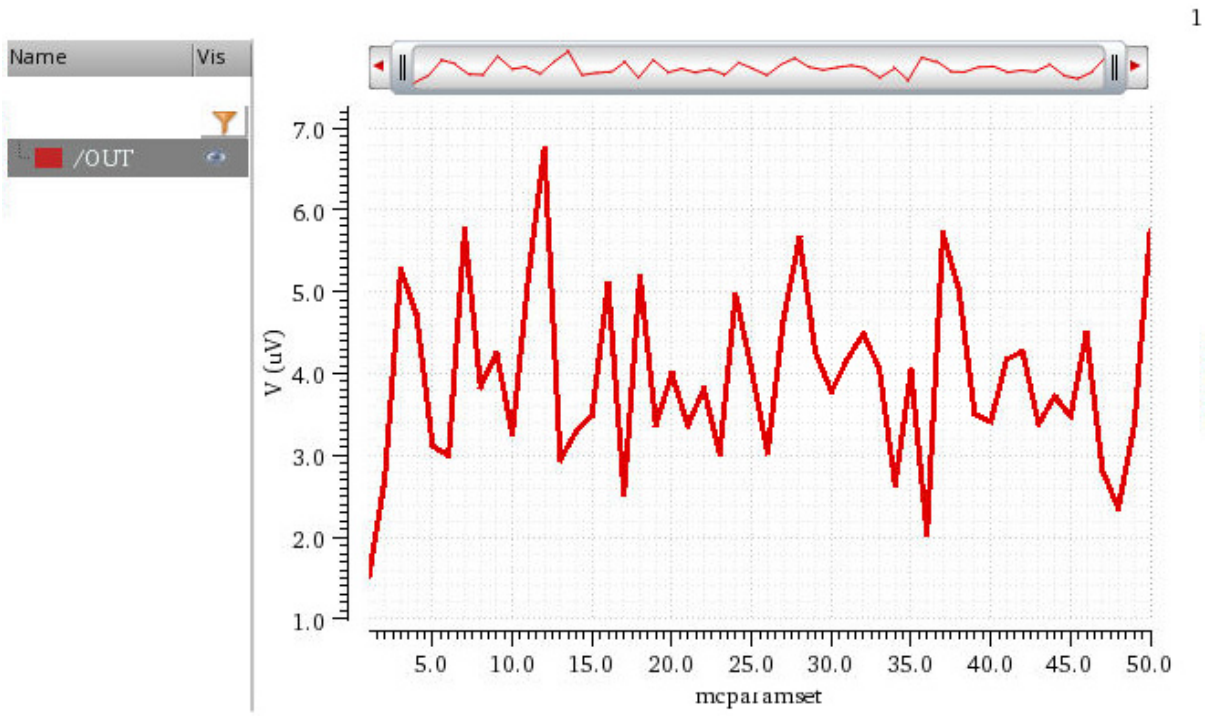
normalQQPValue

Returns the p-value of a QQ (quantile-quantile) plot. A QQ plot draws the correlation between a given sample and the normal distribution.

A small p-value indicates that the distribution is unlikely to be normal, while a large p-value indicates that the distribution is close to normal.

Example

This example shows how the p-value is calculated when you apply the `normalQQPValue` function on the following input signal:



When you apply the `normalQQPValue` function on this signal, the following expression is created in the Buffer:

```
normalQQPValue(v("/OUT" ?result "dcOp"))
```

When you evaluate this expression, the following output is displayed in the Buffer:

```
714.9E-3
```


© 2023 Cadence Design Systems, Inc.

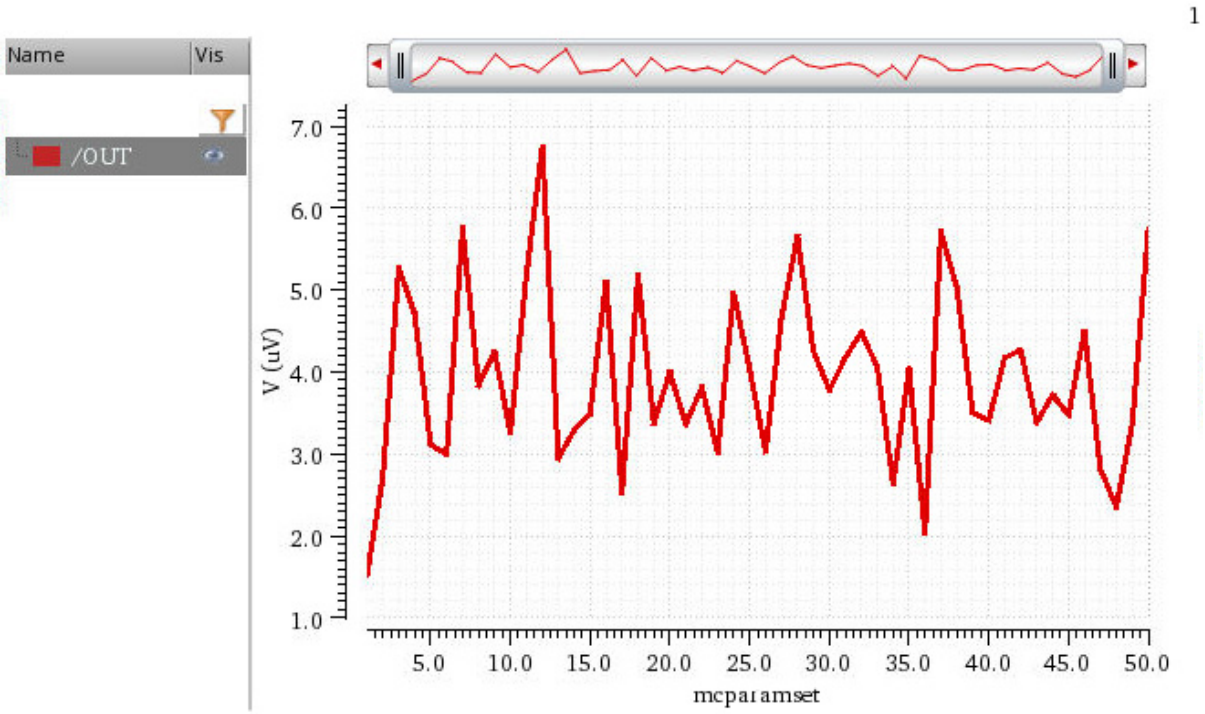
skewness

Calculates the skewness of the specified waveform.

Skewness is a measurement of symmetry or more precisely, the lack of symmetry. A perfectly symmetric waveform has zero skewness.

Example

This example shows how the skewness is calculated when you apply the `skewness` function on the following input signal:



When you apply the `skewness` function on this signal, the following expression is created in the Buffer:

```
skewness(v("/OUT" ?result "dcOp"))
```

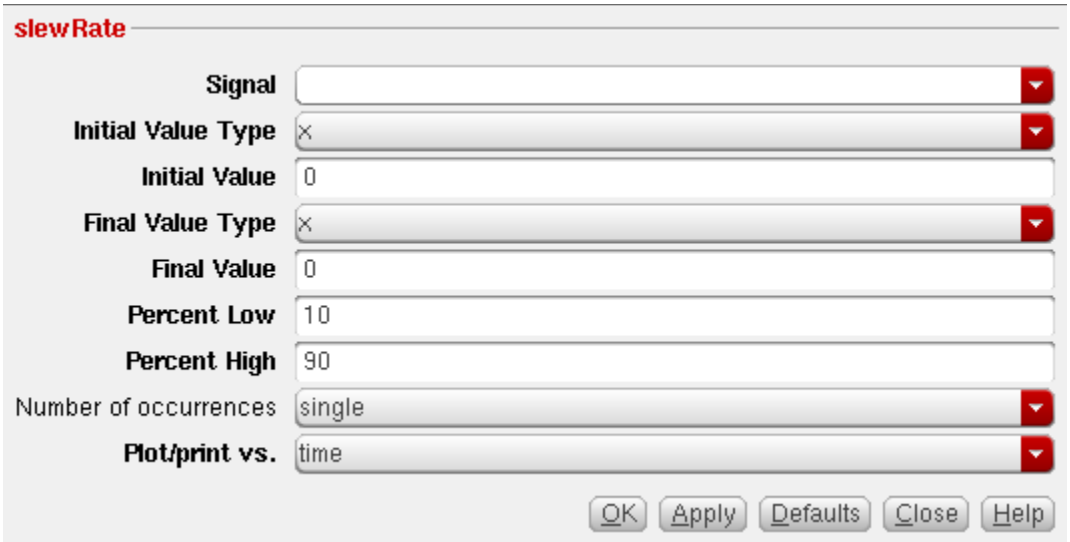
When you evaluate this expression, the following output is displayed in the Buffer:

```
294.2E-3
```


© 2023 Cadence Design Systems, Inc.

slewrate

Computes the average rate at which the Buffer expression changes from percent low to percent high of the difference between the initial value and the final value.



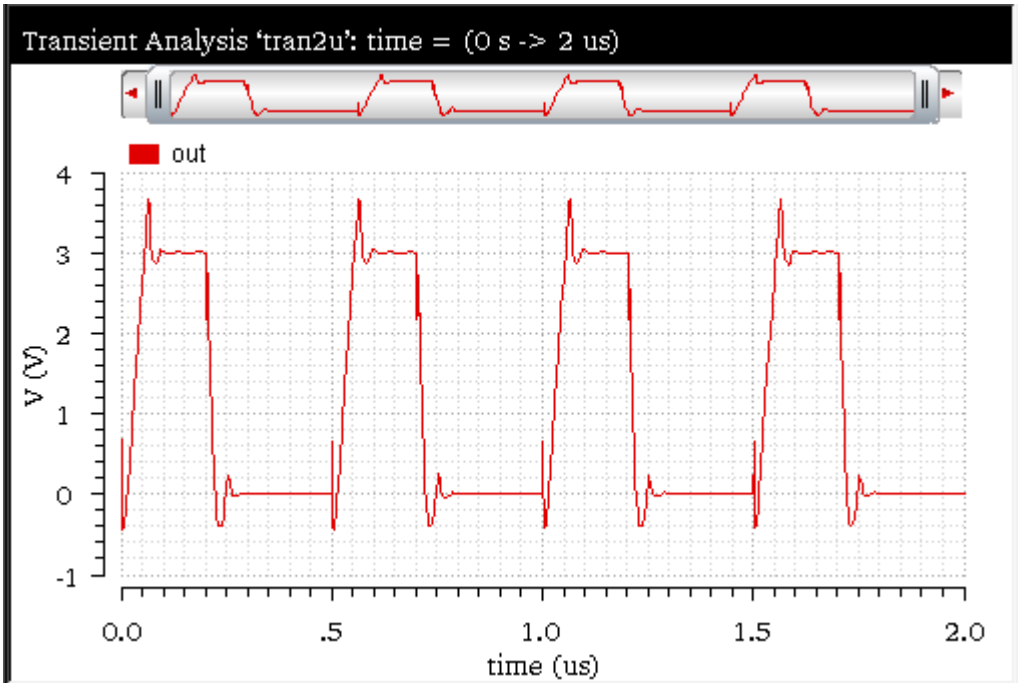
This function includes the following arguments in the SKILL mode,

- *Signal*—Name of the signal.
- *Initial Value Type*—Specifies whether the initial value is an X-axis (`x`) or Y-axis value (`y`). The default value is `y`.
- *Initial Value*—Starting value for the measurement.
- *Final Value Type*—Specifies whether the final value is an X-axis (`x`) or Y-axis value (`y`). The default value is `y`.
- *Final Value*—Final value for the measurement.
- *Percent Low*—Percent low. The default value is `10`.
- *Percent High*—Percent high. The default value is `90`.
- *Number of Occurences*—Specifies whether you want to retrieve only one occurrence of a slewrate event for the given waveform (*single*), or all occurrences of slewrate for the given waveform which you can later plot or print (*multiple*). The default value is `single`.
- *Plot/print vs*—Specifies whether you want to retrieve slewrate risetime data against *time*(or another X-axis parameter for non-transient data) or *cycle*. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform. The default value is `time`. The value in this field is ignored when you specify *Number of Occurrences* as *single*.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from transient analysis:



The `slewrate` function is applied on this input waveform with the following arguments:

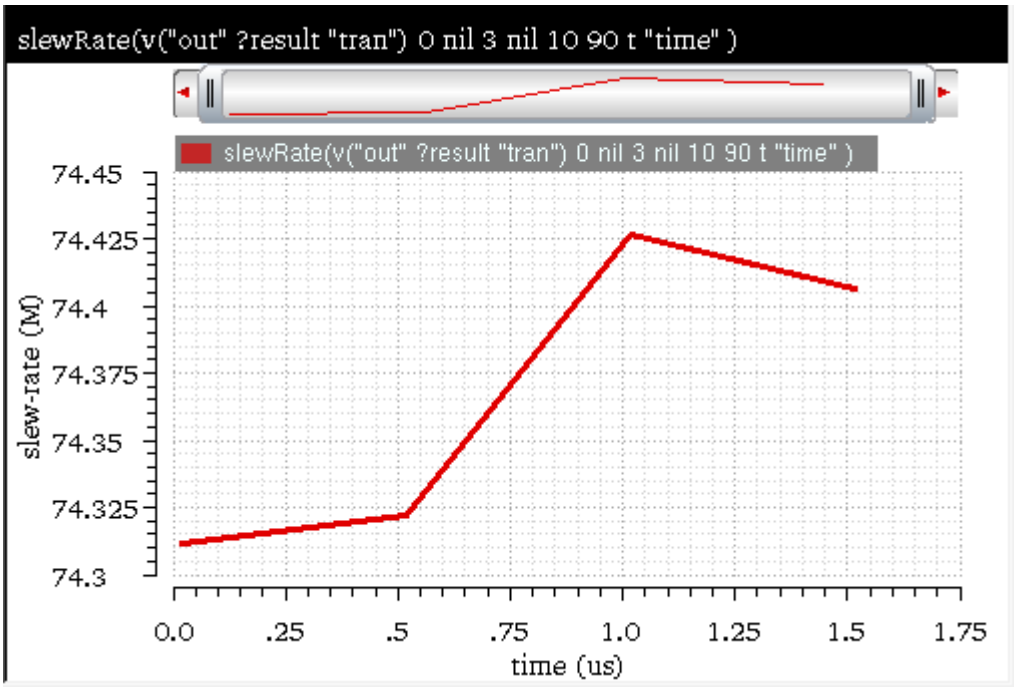
- *Signal*—`v("out" ?result "tran" ?resultsDir "ampsim.raw")`
- *Initial Value Type*—`y`
- *Initial Value*—`0`
- *Final Value Type*—`y`
- *Final Value*—`3`
- *Percent Low*—`10`
- *Percent High*—`90`
- *Number of occurrences*—`multiple`
- *Plot/print vs.*—`time`

The corresponding expression created in the Buffer is as follows:

```
slewRate(v("out" ?result "tran") 0 nil 3 nil 10 90 t "time" )
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the slewrate is calculated and the output waveform is displayed in a new graph window. Notice that multiple slewrate values are calculated in this example because the *Number of occurrences* field has been set to `multiplte`.



Related OCEAN Function

The equivalent OCEAN command for `slewRate` is:

```
slewRate( o_waveform n_initVal g_initType n_finalVal g_finalType
     n_theta1 n_theta2
     [g_multiple [s_Xname]][g_histoDisplay][x_noOfHistoBins] )
     => o_waveform/n_value/nil
```

For more information, see `slewRate` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

spectralPower

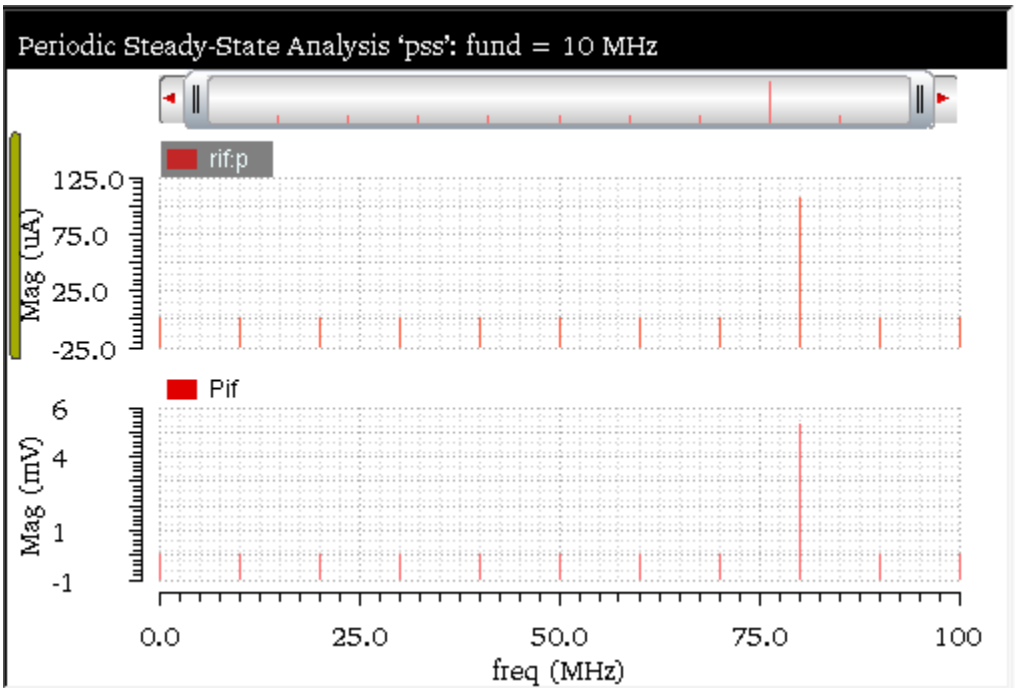
Plots the spectral power for the specified current waveform and voltage waveform. This function is available only in the SKILL mode.

This function includes the following arguments:

- *Current waveform*—Waveform for the current signal which you which you want to calculate the spectral power.
- *Voltage waveform*—Waveform for the voltage signal for which you which you want to calculate the spectral power.

Example

Consider the following input waveforms from PSS analysis:



© 2023 Cadence Design Systems, Inc.

When you send these voltage and current waveforms to the Calculator, the waveform expression that is created in the Buffer is based on their magnitude, as shown below:

```
mag(i("rif:p" ?result "pss_fd" ?resultsDir "./pss_ne600.raw"))
```

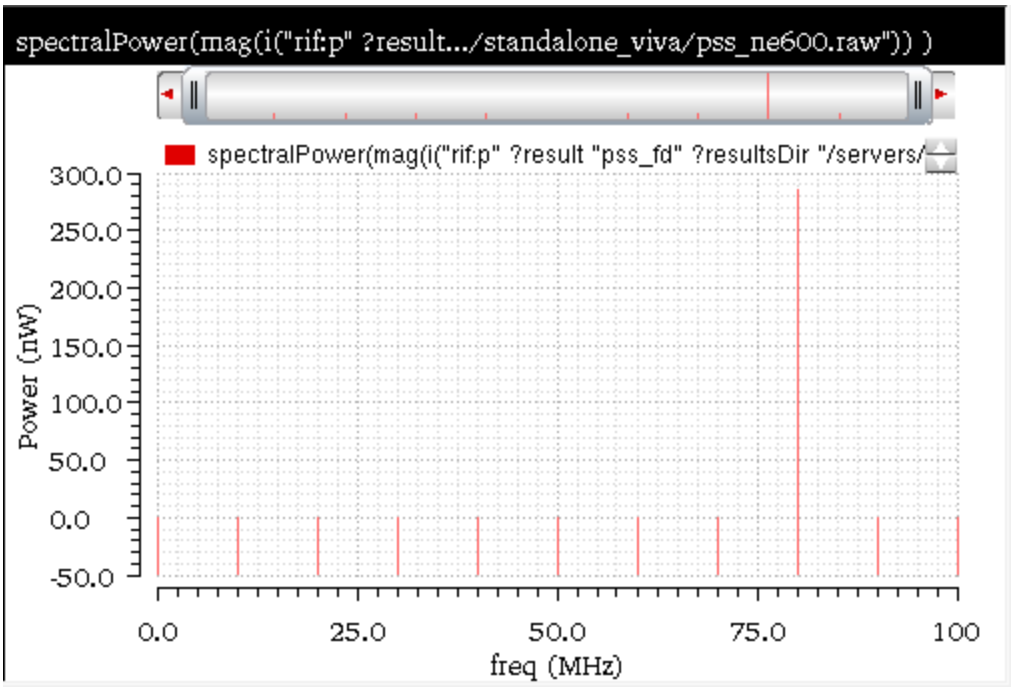
Edit this expression to remove the magnitude and apply the `spectralPower` function with the following arguments:

- *Current Waveform*—`i("rif:p" ?result "pss_fd" ?resultsDir "./pss_ne600.raw")`
- *Voltage Waveform*—`v("Pif" ?result "pss_fd" ?resultsDir "./pss_ne600.raw")`

The corresponding expression created in the Buffer is as follows:

```
spectralPower(i("rif:p" ?result "pss_fd" ?resultsDir "./
pss_ne600.raw") v("Pif" ?result "pss_fd" ?resultsDir "./
pss_ne600.raw") )
```

When you evaluate this expression, the spectral power for these voltage and current signals is calculated and the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN command for `spectralPower` is:

```
spectralPower( o_current o_voltage )
     => o_power/nil
```

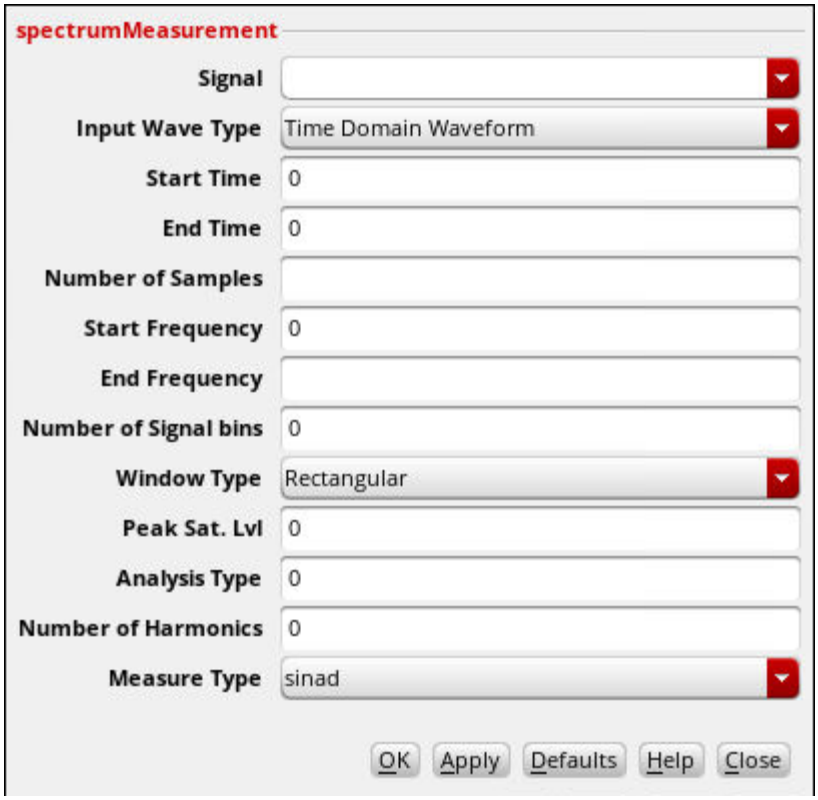
For more information about, see `spectralPower` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

spectrumMeasurement

Calculates Signal-to-Noise-and-Distortion Ratio (SINAD), Spurious Free Dynamic Range (SFDR), Effective Number of Bits (ENOB), and Signal-to-Noise Ratio (without distortion) by using Fast Fourier Transform (FFT) of any given input signal.

The spectrum measure is used for characterizing A-to-D converters and is typically supported for transient simulation data.This function is available only in the SKILL mode



This function includes the following arguments:

- *Signal*—Signal to be measured.
- *Input Wave Type*—Type of the input waveform: *Time Domain waveform* or *Frequency Domain Waveform*.
- *Start Time*—Time to start clipping the signal in time domain.
- *End time*—Time to end clipping in time domain.
- *Number of Samples*—Number of sampled points used for the FFT.
  Valid values: Any integer greater than zero.
  Default value: Number of data points in the *Signal*.

© 2023 Cadence Design Systems, Inc.

- *Start Frequency*—Lower limit of frequency range for the spectrum measures. Default value: First frequency point of the FFT.
- *End Frequency*—Upper limit of frequency range for the spectrum measures. Default value: Last frequency point of the FFT.
- *Number of Signal bins*—Number of signal bins. When you select a window type, this field displays the default number of bins for the selected window type.

  For example, if you select the *Window Type* as `Kaiser` that has two signal bins, this field displays `2`. You can increase the number of signal bins to up to half the value of the sample count.

  For example, if the sample count is `16` for the window type `Kaiser`, you can increase the signal bin count in the *Number of Signal Bins* field up to `8`. You cannot decrease the displayed signal bin value.Valid values: `0–99`.

  Default value: `0`.

  If the signal is not centrally located in a bin, the window spreads the main beam by an amount which is window dependent. In this case, you can designate more bins as signal rather than noise. This field must be set to its default value 0 if the signal is centrally located because the function has stored the nominal spreading due to each window.
- *Window Type*—Windowing function applied to the input waveform.
  Valid values: `Blackman`, `BlackmanHarris`, `Cosine2`, `Cosine4`, `ExtCosBell`, `HalfCycleSine`, `HalfCycleSine3`, `HalfCycleSine6`, `Hanning, Hamming`, `Kaiser`, `Parzen`, `Rectangular`, and `Triangular`.
  Default value: `Rectangular`.
- *Peak Sat. Lvl*—Peak saturation level of the input FFT waveform. Magnitude of the FFT wave is divided by the peak saturation level before using it in calculations. Peak saturation level is the full-scale span ignoring any DC offsets and used in ENOB calculation.Valid values: Any floating point number.

  Default value: `0`
- *Analysis Type*—Specifies whether the analysis type is `Signal Analysis` or `Noise Analysis`.
- *Number of Harmonics*—Number of harmonics for the waveform that you want to plot.

  For example, If this variable is `n`, where `n` is greater than `1` and the fundamental frequency is harmonic `1`, the `n` harmonics are considered for the harmonic power calculation. The signal bins are used for calculating the harmonic power.

  For example, to calculate the total harmonic distortion (THD), if you set the Harmonics value to `n`, where `n` is greater than `1`, and the fundamental frequency is harmonic `1`, the

© 2023 Cadence Design Systems, Inc.

number of harmonics used to calculate THD is `2, . . . , n`. If `n=3`, the `2nd` and `3rd` harmonics are used to calculate THD.

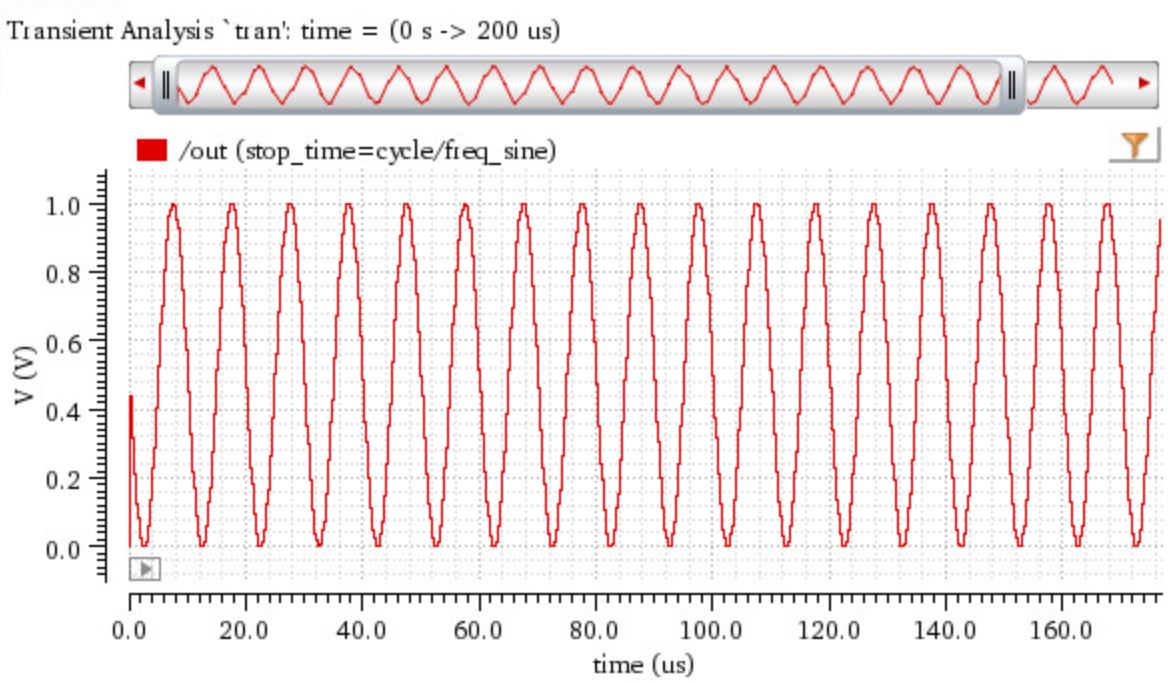
- *Measure Type*—Result specifier.

  Valid values: `sinad`, `sfdr(db)`, `enob`, and `snhr`.

  Default value: `sinad`

Example

Consider the following input waveform from a transient analysis:



The `spectrumMeasurement` function is applied on this waveform with the following arguments:

- *Signal*—`VT("/out")`
- *Input Wave Type*—`Time Domain Waveform`
- *Start Time*—`0`
- *End Time*—`3`
- *Number of Samples*—`1024`
- *Start Frequency*—`0`

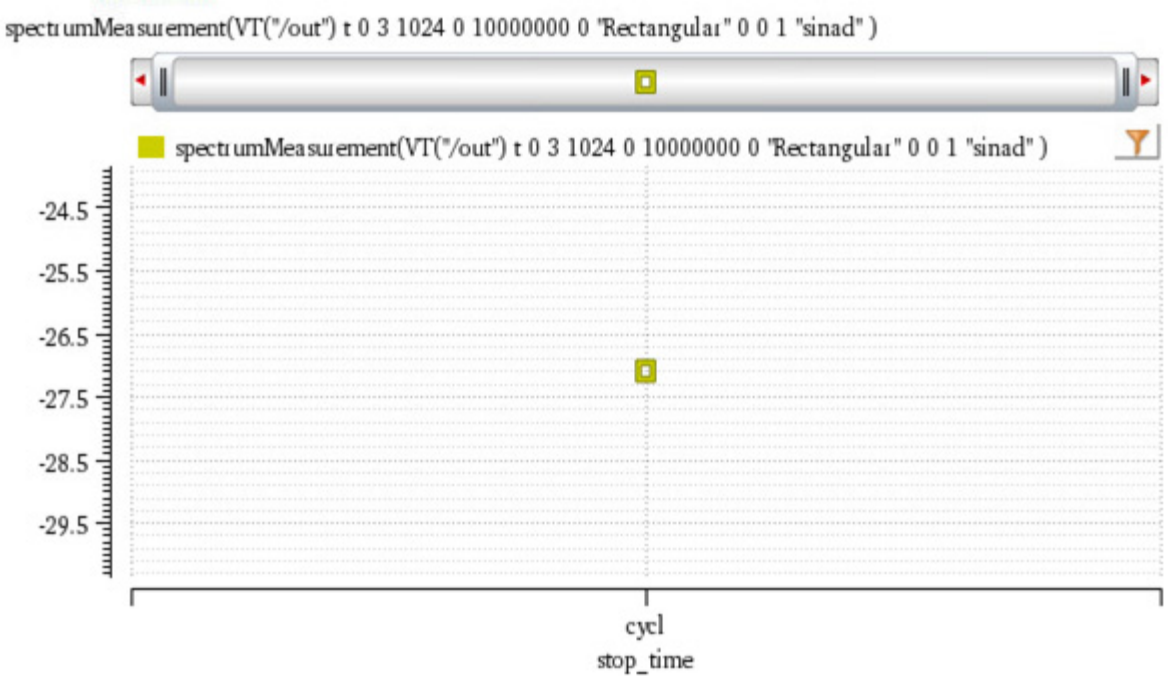
© 2023 Cadence Design Systems, Inc.

- *End Frequency*—`10000000`
- *Number of Signal bins*—`0`
- *Window Type*—`Rectangular`
- *Peak Sat. Lvl*—`0`
- *Analysis Type*—`0`
- *Number of Harmonics*—`1`
- *Measure Type*—`sinad`

The expression created in the Buffer is as follows:

```
spectrumMeasurement(VT("/out") t 0 3 1024 0 10000000 0 "Rectangular" 0 0 1 "sinad" )
```

When you evaluate this expression, the following output value of the spectrum measurement `sinad` is displayed:



Related OCEAN Function

The equivalent OCEAN command for `spectrumMeasurement` is:

© 2023 Cadence Design Systems, Inc.

```
spectrumMeasurement( o_waveform g_isTimeWave n_from n_to x_numSamples n_startFreq
     n_endFreq x_signalBins t_windowName n_satLvl g_isNoiseAnalysis
     x_noOfHarmonics t_measType)

=> g_value / nil
```

For more information, see `spectrumMeasurement` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

stddev

Computes the standard deviation of a waveform (or a family of waveforms) over its entire range. This function can be applied to both scalar and vector inputs.

You can calculate the standard deviation for a population as well as for a sample.

The population standard deviation is a parameter, which is a fixed value calculated from every individual in the population.

A sample standard deviation is a statistic, which means that it is calculated from only some of the individuals in a population.

Since the sample standard deviation depends upon the sample, it has greater variability. Thus, the standard deviation of the sample is greater than that of the population.

If `y=f(x)`:

Standard deviation for a population is calculated as follows:

$$stddev(y) = \sqrt{\frac{\sum_{i=1}^{N} (y_i - average(y))^2}{N}}$$

Standard deviation for a sample is calculated as follows:

$$stddev(y) = \sqrt{\frac{\sum_{i=1}^{N} (y_i - average(y))^2}{N-1}}$$

Where N is the total number of data points.

© 2023 Cadence Design Systems, Inc.

When you use this function to evaluate data for measurement across corners, a new key argument, `overall`, is added to this function. If the input is a scalar and the `overall` argument is set to `t`, the `stddev` function returns zero.

This function includes the following arguments:

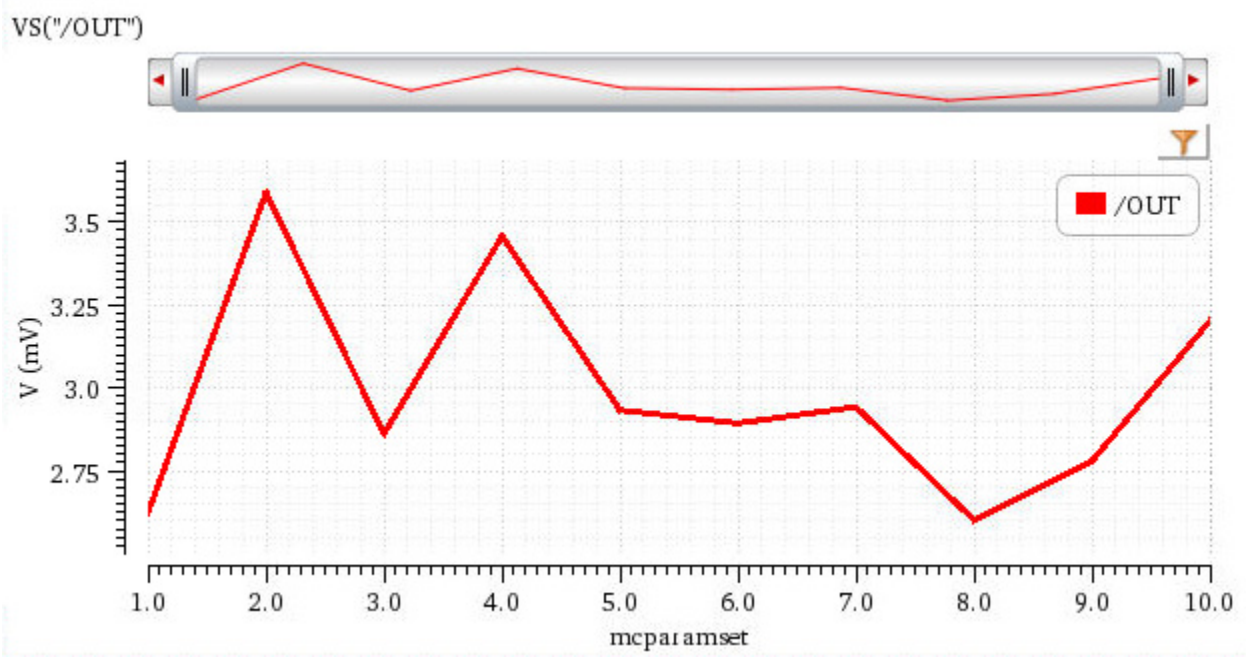
- *Signal*—Displays the name of the signal.
- *type*—Specifies the type of standard deviation. Possible values are *population* and *sample*.
- *continuous*—Specifies whether to consider the signal as a continuous signal. If the signal is continuous, average is calculated using the integral. If you set it to *no*, the signal is considered as a discrete signal. The default is *yes*.

  This argument controls how the average is used in the calculation of standard deviation. For more information about how average is calculated for a continuous and a discrete signal, see the average function.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform:



The following arguments are specified in this example:

- *Signal*—`VS("/OUT")`
- *type*—`sample`
- *continuous*—`no`

When you apply the `stddev` function on this signal, the following expression is created in the Buffer:

```
stddev(VS("/OUT") ?type "sample" ?continuous nil )
```

When you evaluate this expression, the standard deviation is calculated and the following output is displayed in the Buffer:

```
329.9E-6
```

If you specify the *type* argument as `population` and the *continuous* argument as `yes`, and then apply the `stddev` function on the same signal, the following expression is created in the Buffer:

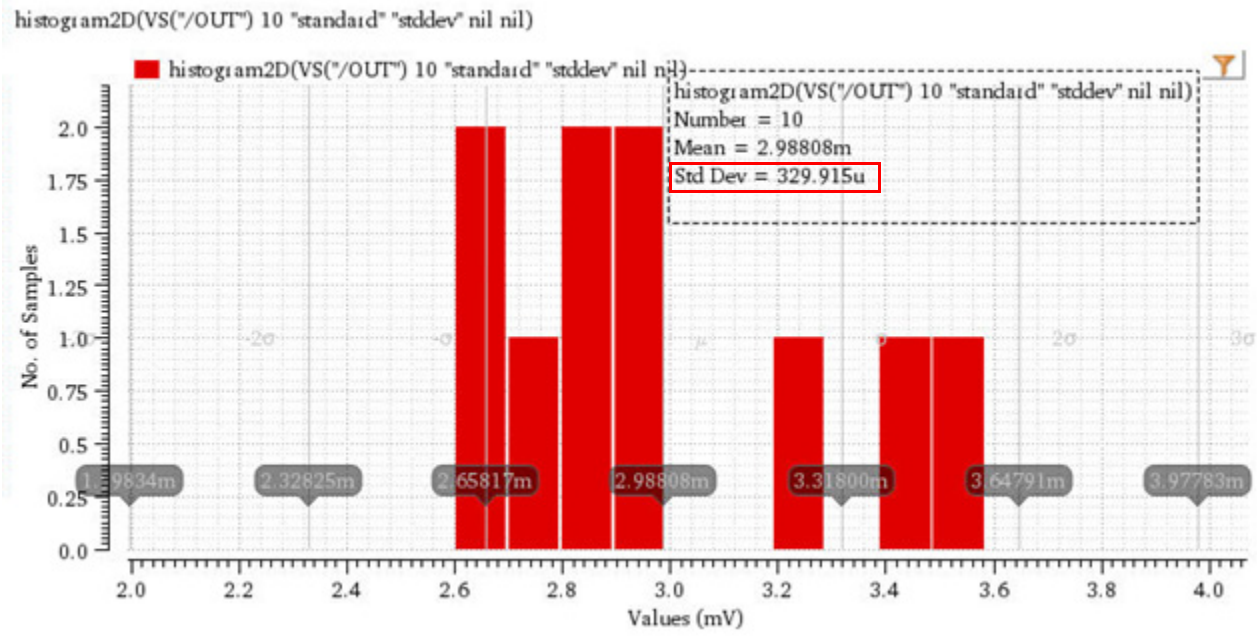
© 2023 Cadence Design Systems, Inc.

```
stddev(VS("/OUT") )
```

When you evaluate this expression, the standard deviation is calculated and the following output is displayed in the Buffer:

```
314.4E-6
```

Now, if you annotate the standard deviation on the histogram plotted using the same input signal `VS("/OUT")`, you will observe that the value is same as calculated by Calculator with the *type* argument specified as `sample` and the *continuous* argument specified as `no`.



Related OCEAN Function

The equivalent OCEAN command for `stddev` is:

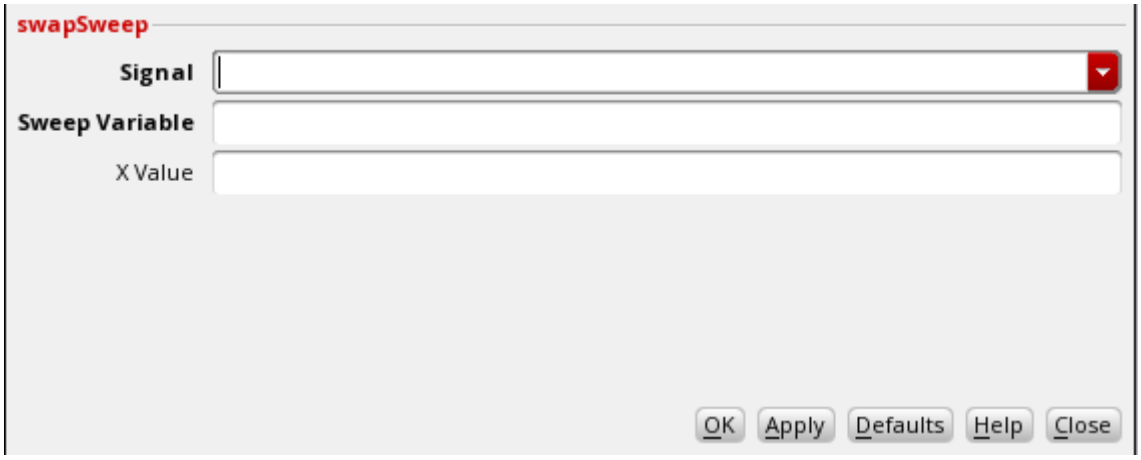
```
stddev( o_waveform )
     => n_stddev/o_waveformStddev/nil
```

For more information, see `stddev` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

swapSweep

Swaps the X-axis value with the specified sweep variable.



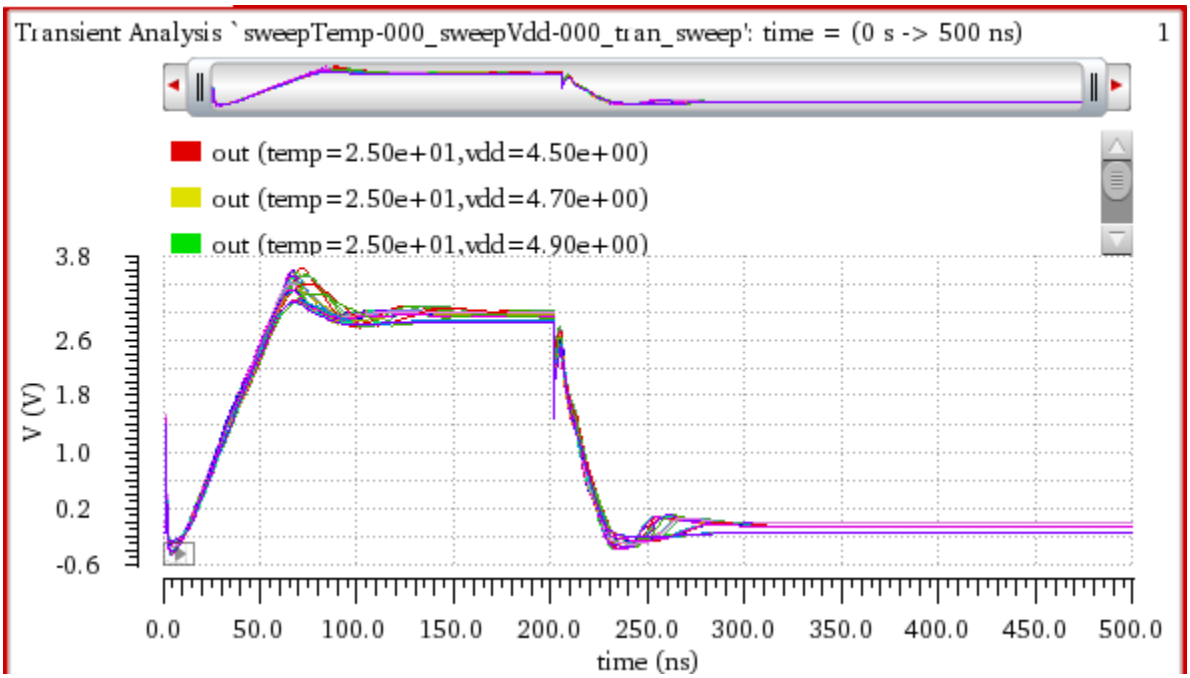
This function includes the following arguments:

- *Signal*—The output waveform whose X-axis values you want to swap.
- *Sweep Variable*—Sweep variable with which the X-axis of the given waveform is to be swapped. For example, `CAP`, `temp`.
- *X Value*—X-axis value at which the waveform is swapped with the specified sweep variable. This is an optional argument.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input signal from a parametric data simulated using the sweep variables, `vdd` and `temp`.



Now, specify the following argument values:

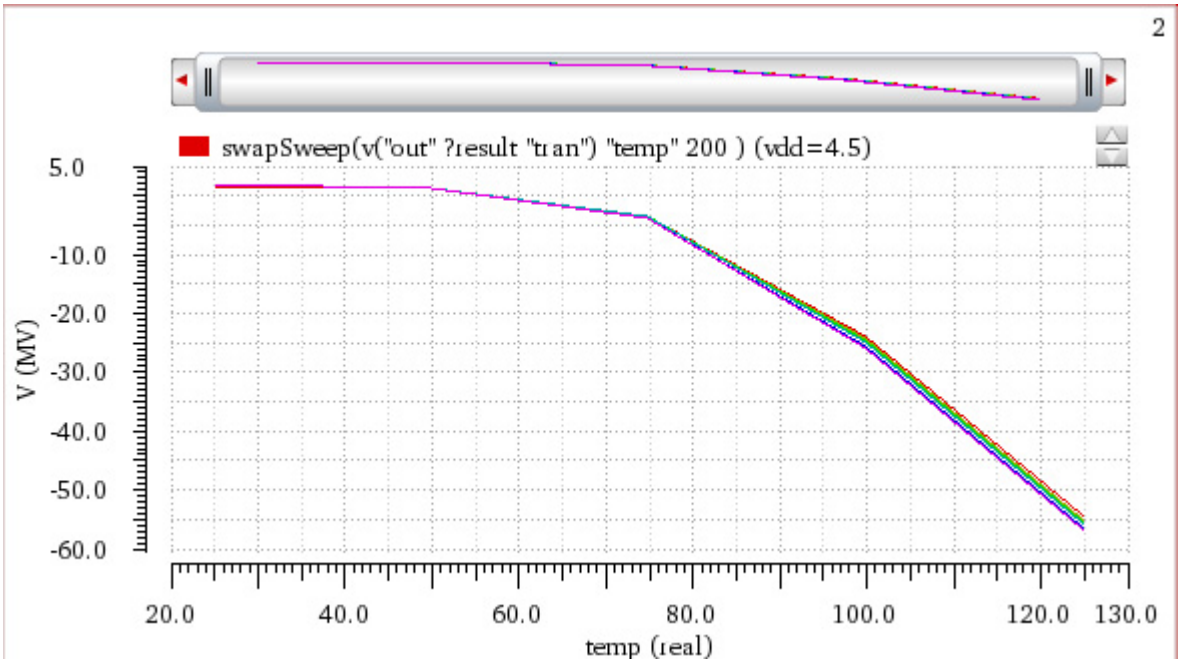
- *Signal*—`v("out"?result "tran" ?resultsDir "./sweeptran.raw)`
- *Sweep Variable*—`temp`
- *X Value*—`200`

The following expression is created in the buffer:

```
SwapSweep(v("out" ?result "tran") "temp" 200)
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform is displayed that has the X-axis values swapped with the specified variable, `temp`.



Related SKILL Function

```
swapSweep(
     o_waveform
     t_sweepVar
     [ ?xValue n_xValue ]
     )
     => o_waveform / nil
```

For more information see swapSweep.

© 2023 Cadence Design Systems, Inc.

tangent

(Tangent Line)

Plots a line that passes through x and y coordinates and the slope that you specify. This function is available only in the SKILL mode.

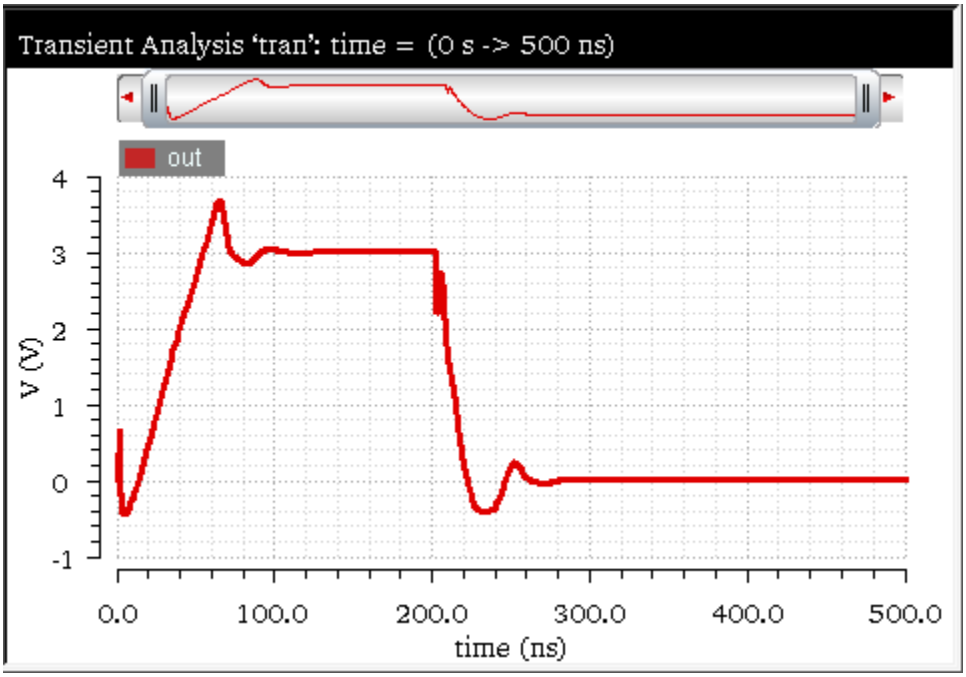
This function includes the following fields:

- *Signal*—Name of the signal.
- *X Point*—X-axis value you specify.
- *Y Point*—Y-axis value you specify.
- *Slope*—Slope you specify.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from the transient analysis:



The tangent function is applied on this input waveform with the following arguments:

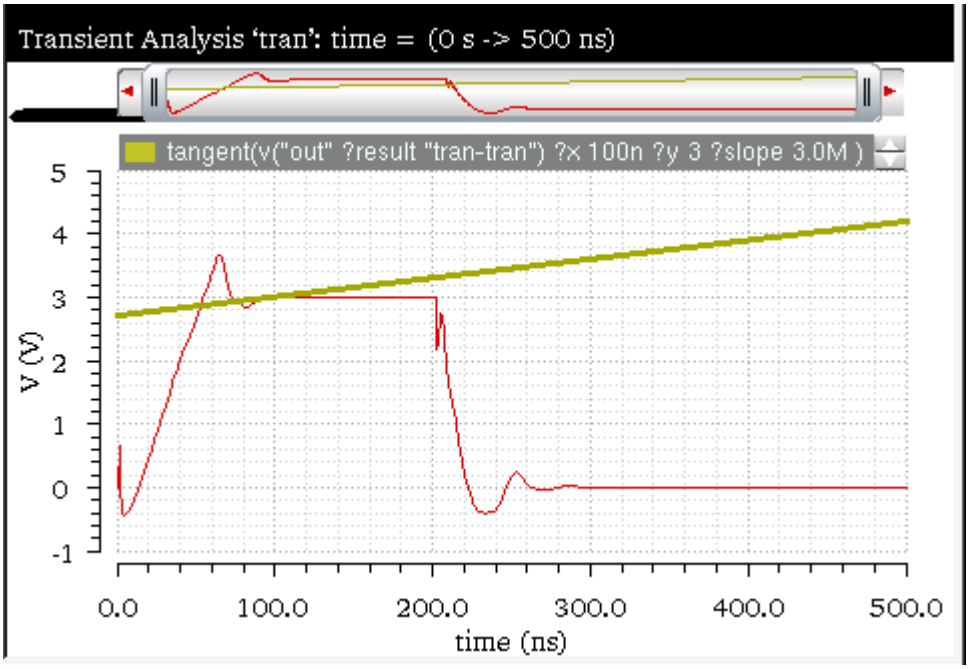
- *Signal*—`v("out" ?result "tran-tran" ?resultsDir "ampsim.raw")`
- *X Point*—`100n`
- *Y Point*—`3`
- *Slope*—`3.0M`

The corresponding expression created in the Buffer is as follows:

```
tangent(v("out" ?result "tran-tran" ?resultsDir "ampsim.raw") ?x 100n
?y 3 ?slope 3.0M )
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression and plot results in the append mode, a tangent line is displayed in the graph as shown in the figure below:



Related OCEAN Function

The equivalent OCEAN command for `tangent` is:

```
tangent( o_waveform [ ?x n_x ] [ ?y n_y ] [ ?slope n_slope ] )
     => o_waveform/nil
```

For more information, see tangent in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

thd

(Total Harmonic Distortion)

Computes the percentage of THD of a signal with respect to the fundamental frequency and is expressed as a voltage percentage.

This function is available only in the SKILL mode.

- *Signal*—Name of the signal.
- *From*—Starting time for the DFT sample window.
- *To*—Ending time of the DFT sample window.
- *Number* of Samples—Number of time domain points to be used.
- *Fundamental (Hz)*—Fundamental frequency of the signal.

Additional Information

The computation uses the dft function. Assume that the *dft* function returns complex coefficients A0, A1..., A*f* , ... The fundamental frequency *f* is the frequency contributing to the largest power in the signal. A0 is the complex coefficient for the DC component and Ai is the complex coefficient for the *i*th harmonic where *i≠0,f*. Then, total harmonic distortion is computed as:

$$THD = \frac{\sqrt{\sum_{i=1, i\neq 0} f^{|A_i|^2}}}{|A_f|} \times 100\%$$

© 2023 Cadence Design Systems, Inc.

The accuracy of the total harmonic distortion measurement depends on simulator options and the analysis parameters. To view the simulator options, do the following:

- Choose *Simulation – Options – Analog*.

  The *Simulator Options* form appears.

If you are using the Spectre simulator, the options listed in the table below are displayed in the main tab of the Simulator Options form. For an accurate measurement, set the following simulation options:

Option	Description
reltol	Relative convergence criterion. Default value is `1e-3`
residualtol	Tolerance ratio for residual (multiplies reltol).
vabstol	Voltage absolute tolerance convergence criterion. Default value is `1e-6`
iabstol	Current absolute tolerance convergence criterion. Default value is `1e-12`

Set the option `strobeperiod` to the period of the signal to be analyzed divided by `128`, and simulate enough cycles of this signal so that circuit can reach a steady-state. If that occurs in the 10th cycle, end the simulation just after the 10th cycle.

For the spectre, and some other simulator interfaces, the Options buttons in each Choosing Analyses form let you set options that apply to the specific analysis.

If you are using the HSPICE simulator, the following Tolerance options are displayed:

Option	Suggested Value
reltol	`1e-5`
abstol	`1e-13`
vntol	`3e-8`
trtol	`1`
method	`gear`
maxord	`3`

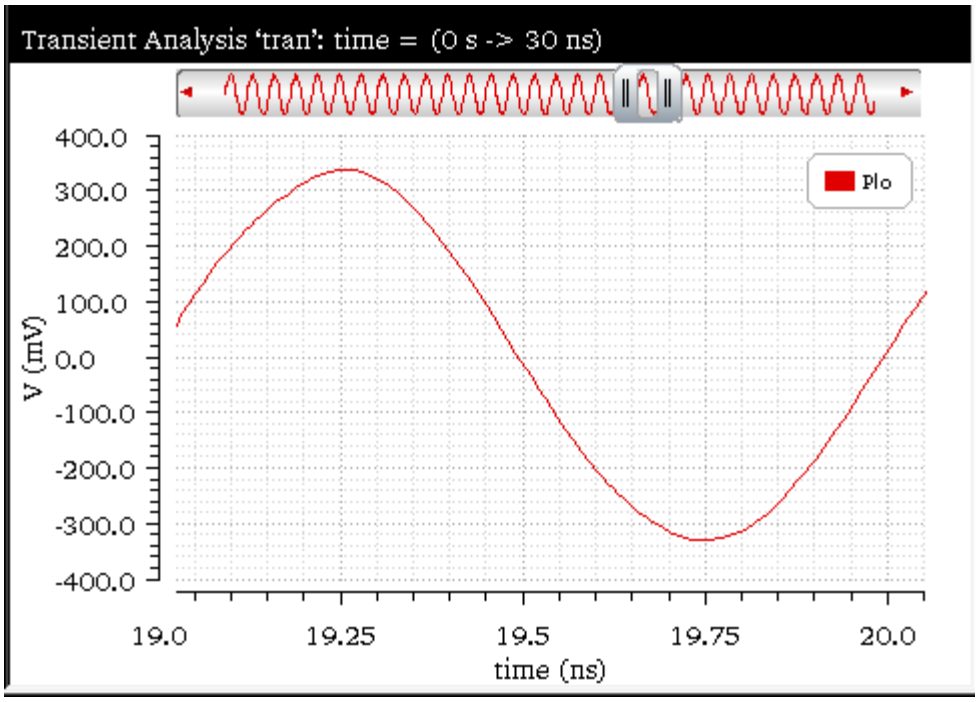
© 2023 Cadence Design Systems, Inc.

When you use transient analysis, it is difficult to generate the accurate values for THD because you need to be careful to avoid interpolation error in the dft() function and make sure that the waveform reaches a steady state. Also, you need to ensure that transient samples are at equal time, steps, and so on. The interpolation error occurs when the number of samples are large and cannot be fully processed by the simulator. Therefore, the waveform is interpolated for the remaining points and causes small numeric errors, which thereby, results in a non-zero value for THD.

In such cases, it is recommended to use RF simulations whenever possible (except for mixed-signal and verilog-A behavioral models). The direct plot includes the RF version of THD measurements and returns THD=0 with more accurate measurements of small distortions.

Example

This example displays the total harmonic value generated when you apply the `thd` function on the following input waveform:



The following arguments are specified in this example:

- *Signal*—`v("Plo" ?result "tran")`
- *From*—`19n`
- *To*—`20n`

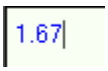
© 2023 Cadence Design Systems, Inc.

- *Number of Samples*—`8192`
- *Fundamental (HZ)*—`1G`

The following expression is created in the Buffer:

```
thd(v("Plo" ?result "tran" ?resultsDir "./pss_ne600.raw") 19n 20n
8192 1G )
```

When you evaluate this expression, the output value generated is displayed in the Buffer (as shown in the figure below):



Related OCEAN Function

The equivalent OCEAN function for `thd` is:

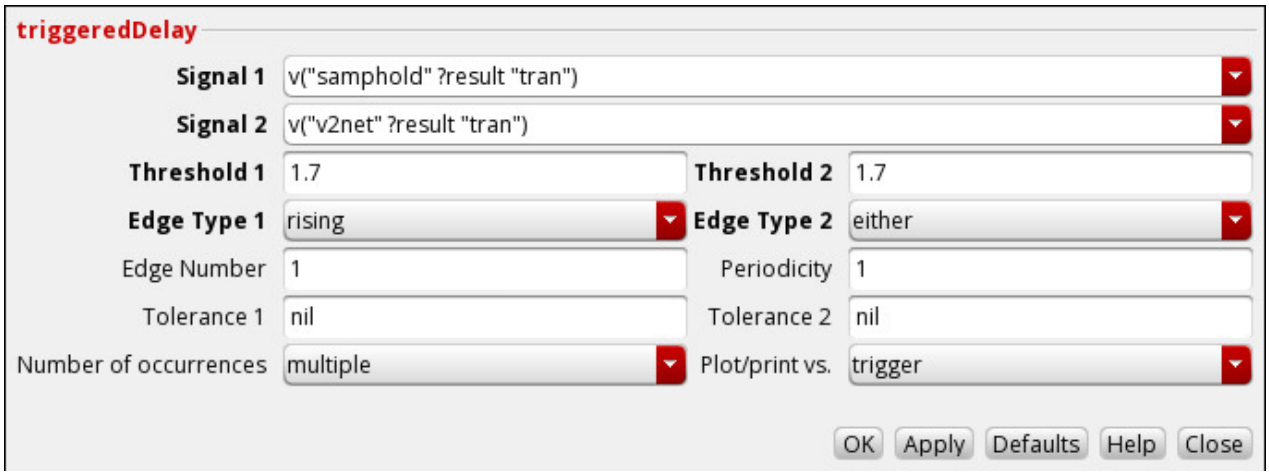
```
thd( o_waveform n_from n_to x_num n_fund)
     => o_waveform/n_thdValue/nil
```

For more information about the OCEAN function, see thd in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

triggeredDelay

Calculates the delay from the trigger point on the edge (either rising or falling) of a triggering signal to the next edge (either rising or falling) of the target signal. This function is available only in the SKILL mode.



This function includes the following arguments:

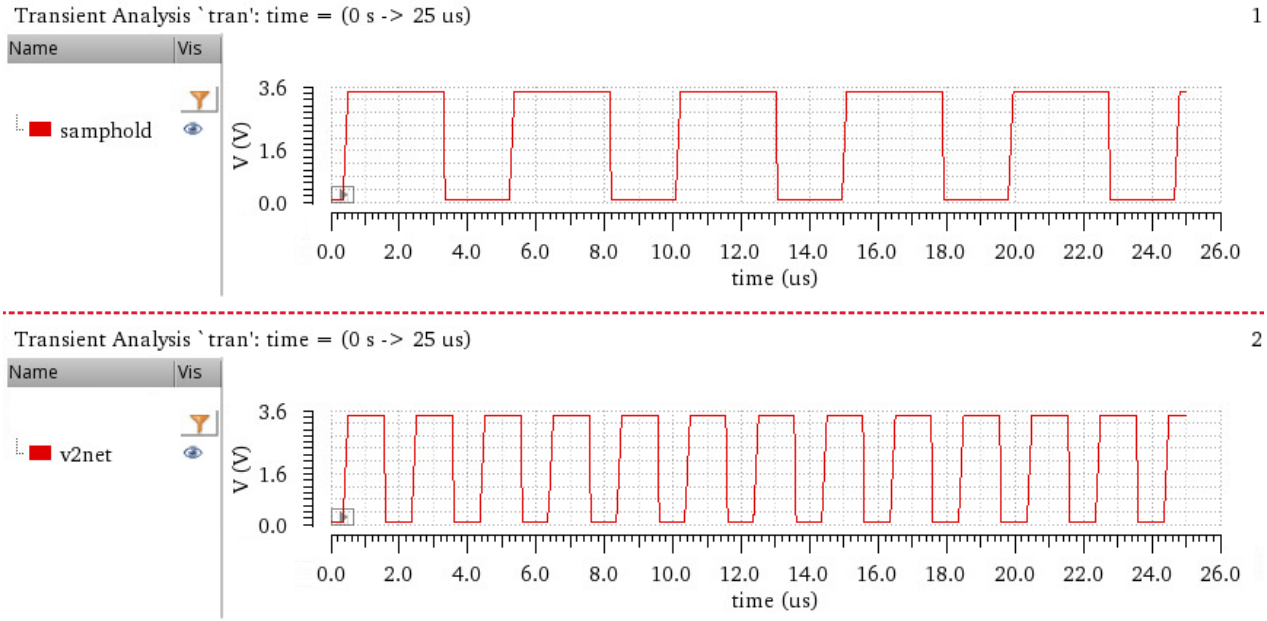
- *Signal1*—Name of the triggering signal.
- *Signal2*—Name of the target signal.
- *Threshold 1*—Threshold value for the triggering signal.
- *Threshold 2*—Threshold value for the target signal.
- *Edge Type 1*—Direction of the first crossing event. `rising` directs the function to look for crossings where the Y value is increasing, `falling` for crossings where the Y value is decreasing, and `either` for crossings in either direction.
- *Edge Type 2*—Direction of the second crossing event. `rising` directs the function to look for crossings where the Y value is increasing, `falling` for crossings where the Y value is decreasing, and `either` for crossings in either direction.
- *Edge Number*—Number that specifies from which edge of the triggering signal you want to calculate the delay. For example, if *Edge Number=1*, the delay is calculated from the first edge of the triggering signal.
- *Periodicity*—Periodic interval for the triggering signal.
- *Tolerance 1*—Tolerance value to detect the threshold crossings for the triggering signal.
- *Tolerance 2*—Tolerance value to detect the threshold crossings for the target signal.

© 2023 Cadence Design Systems, Inc.

- *Number of occurrences*—Specifies whether you want to retrieve only one occurrence of a delay event for the given waveform (`single`), or all occurrences of an overshoot for the given waveform, which you can later plot or print (`multiple`).
- *Plot/print vs.*—Specifies whether you want to retrieve the delay data against trigger time, target time (or another X-axis parameter for non-transient data) or cycle. Cycle numbers refer to the n'th occurrence of the delay event in the input waveform.The value in this field is ignored when you specify *Number of occurrences* as `single`.

Example

Consider the following example, where `samphold` is the triggering signal and `v2net` is the target signal.



To calculate the delay from the trigger point on the first rising edge of the triggering signal, `samphold`, to the next edge (either rising or falling) of the target signal, `v2net`, specify the following arguments:

- *Signal1*—`v("samphold" ?result "tran")`
- *Signal2*—`v("v2net" ?result "tran")`
- *Threshold 1*—`1.7`
- *Threshold 2*—`1.7`
- *Edge Type 1*—`rising`

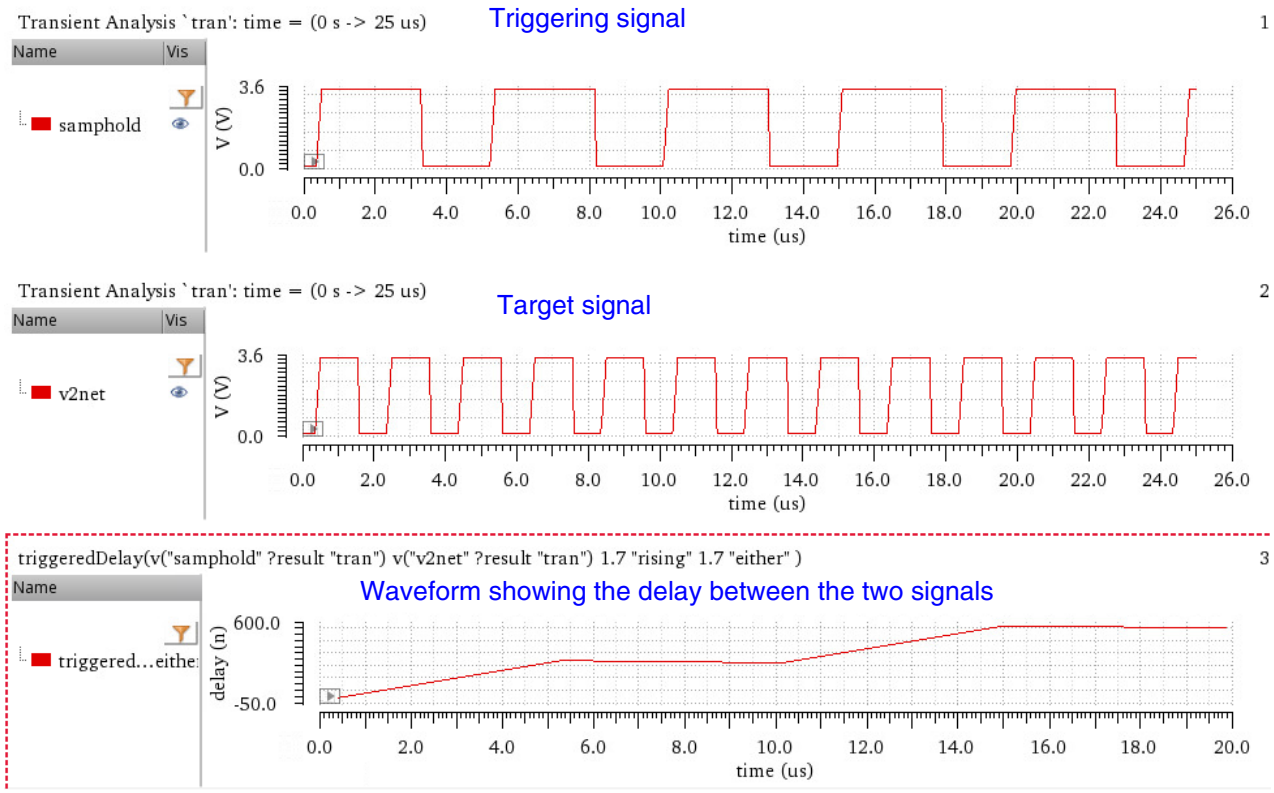
© 2023 Cadence Design Systems, Inc.

- *Edge Type 2*—`either`
- *Edge Number*—`1`
- *Periodicity*—`1`
- *Tolerance1*—`nil`
- *Tolerance2*—`nil`
- *Number of occurrences*—`multiple`
- *Plot/print vs.*—`trigger`

The following expression is created in the Buffer:

```
triggeredDelay(v("samphold" ?result "tran") v("v2net" ?result
"tran") 1.7 "rising" 1.7 "either" )
```

Now, when you evaluate this expression, the following output waveform showing the delay in two signals is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

Related OCEAN Function

The equivalent OCEAN command for `delay` is:

```
delay(
     [ ?wf1 o_waveform1 ]
     [ ?value1 n_value1 ]
     [ ?edge1 s_edge1 ]
     [ ?nth1 x_nth1 ]
     [ ?td1 n_td1 ]
     [ ?wf2 o_waveform2 ]
     [ ?value2 n_value2 ]
     [ ?edge2 s_edge2 ]
     [ ?nth2 x_nth2
     [ ?td2 n_td2 ]
     [ ?td2r0 n_td2r0 ]}
     [ ?stop n_stop ]
     [ ?histoDisplay g_histoDisplay ]
     [ ?noOfHistoBins x_noOfHistoBins ]
     [ ?period1 period1 ]
     [ ?period2 period2 ]
     [ ?multiple multiple ]
     [ ?xName xName ]
     )
     => o_waveform / n_value / nil
```

For more information about this OCEAN function, see delay in *OCEAN Reference*.

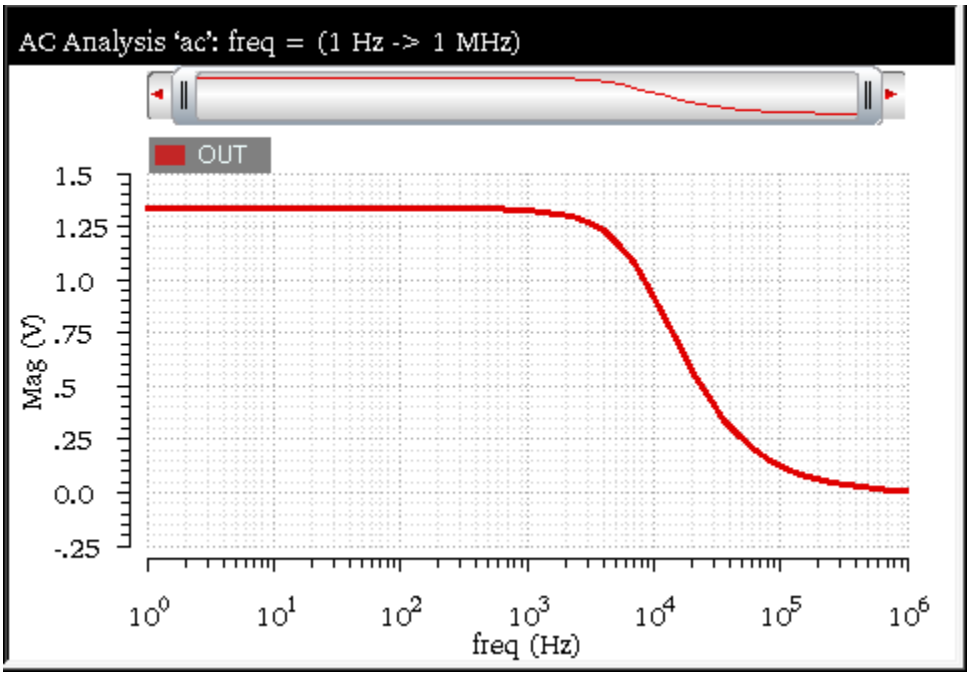
© 2023 Cadence Design Systems, Inc.

unityGainFreq

Computes and reports the frequency at which the gain is unity. This function is available only in the SKILL mode.

Example

Consider the following input waveform from AC analysis:



The expression for this signal is displayed in the Buffer. Now, when you select the `unityGainFreq` function from the Function Panel, the function is applied to the signal expression in the Buffer. The corresponding expression created in the Buffer is as follows:

```
unityGainFreq(v("OUT" ?result "ac" ?resultsDir "./aExamples.raw"))
```

When you evaluate this expression, the following output value indicating the frequency is displayed in the Buffer:

```
8.446E3
```

Related OCEAN Function

The equivalent OCEAN command for `unityGainFreq` is:

© 2023 Cadence Design Systems, Inc.

```
unityGainFreq( o_gainFreqWaveform )
     => n_frequency/nil
```

For more information, see `unityGainFreq` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

v

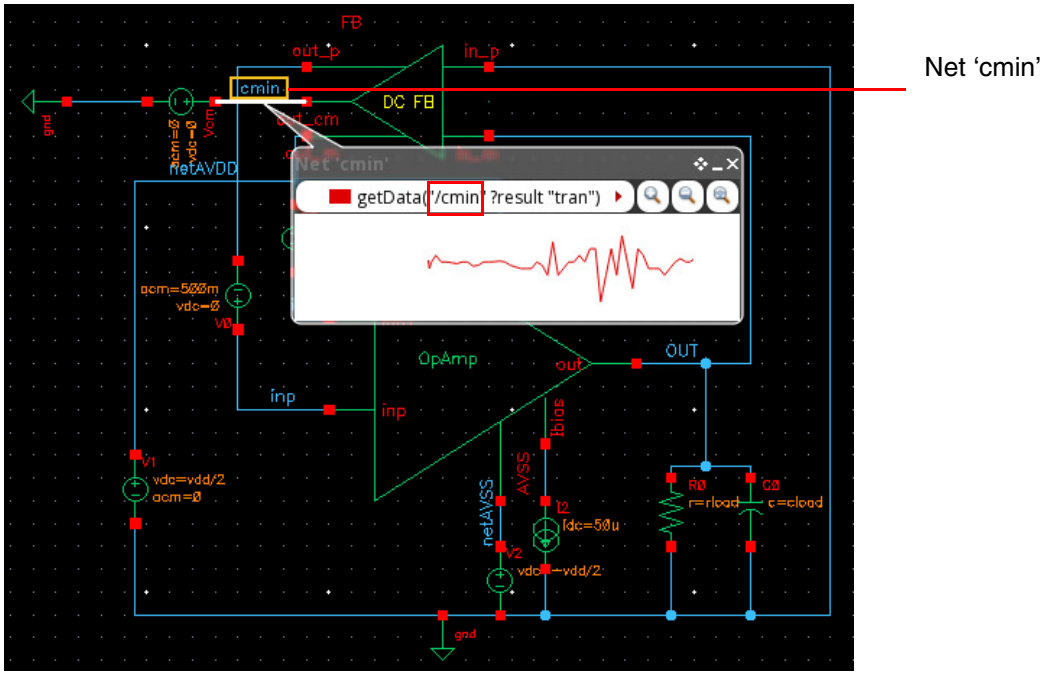
Calculates the voltage of the specified net or terminal.

This function includes the following arguments:

- *Signal name*—The name of the signal.
- *Analysis Type*—The type of analysis. For example, `ac` and `tran`.
- *Signal Type*—Specifies whether the specified signal is a terminal voltage or a net.

Example 1

Consider the net, `cmin`, in the following design.



Net ‘cmin’

© 2023 Cadence Design Systems, Inc.

This example shows the voltage calculation for the net, `cmin`, using the `v` fuction. Specify the following arguments:

- *Signal name*—`cmin`

  Note: Specify this argument by selecting the signal from the schematic.

- *Analysis Type*—`tran`
- *Signal Type*—`net`

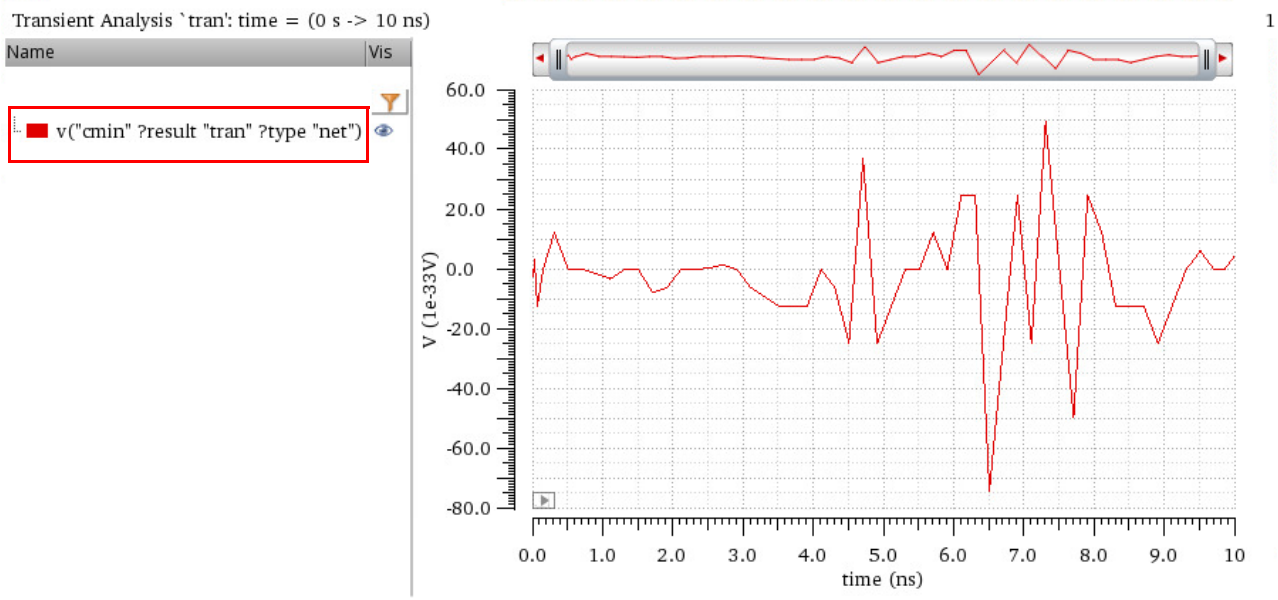
Note: Before using this function, you need to set the results database context to the results database that contains the input signal, such as `cmin` used in this example. For more information about how to change the in-context results directory, see Setting the In-Context Results Directory.

The following expression is created in the Buffer:

```
v("cmin" ?result "tran" ?type "net")
```

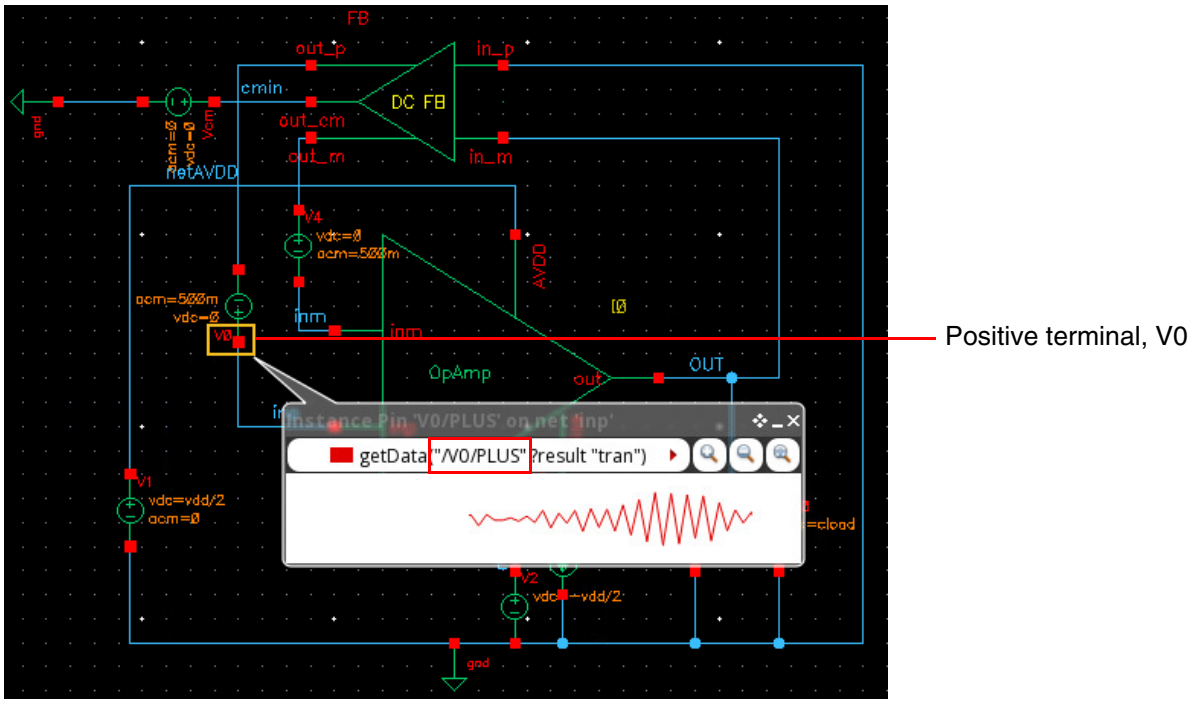

© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the following waveform, which displays the variation of voltage values with time at the specified net.



Example 2

This example shows the voltage calculation for the positive terminal of `V0`, as shown in the following design.



© 2023 Cadence Design Systems, Inc.

This example shows the voltage calculation for the positive terminal of `V0` using the `v` function. Specify the following arguments:

- *Signal name*—`/V0/PLUS`

  Note: Specify this argument by selecting the signal from the schematic.

- *Analysis Type*—`tran`
- *Signal Type*—`terminalV`

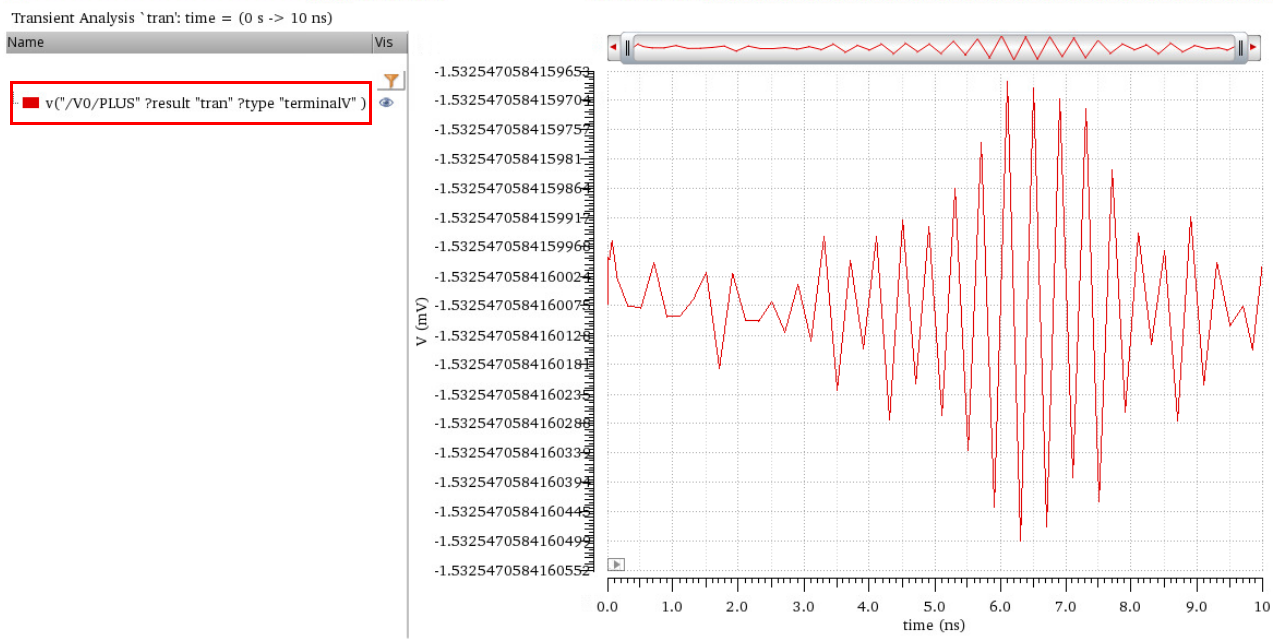
Note: Before using this function, you need to set the results database context to the results database that contains the input signal, such as `/V0/PLUS` used in this example. For more information about how to change the in-context results directory, see Setting the In-Context Results Directory.

The following expression is created in the Buffer:

```
v("/V0/PLUS" ?result "tran" ?type "terminalV")
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the following waveform, which displays the variation of voltage values with time at the specified terminal.



Related OCEAN Function

The equivalent OCEAN command for `v` is:

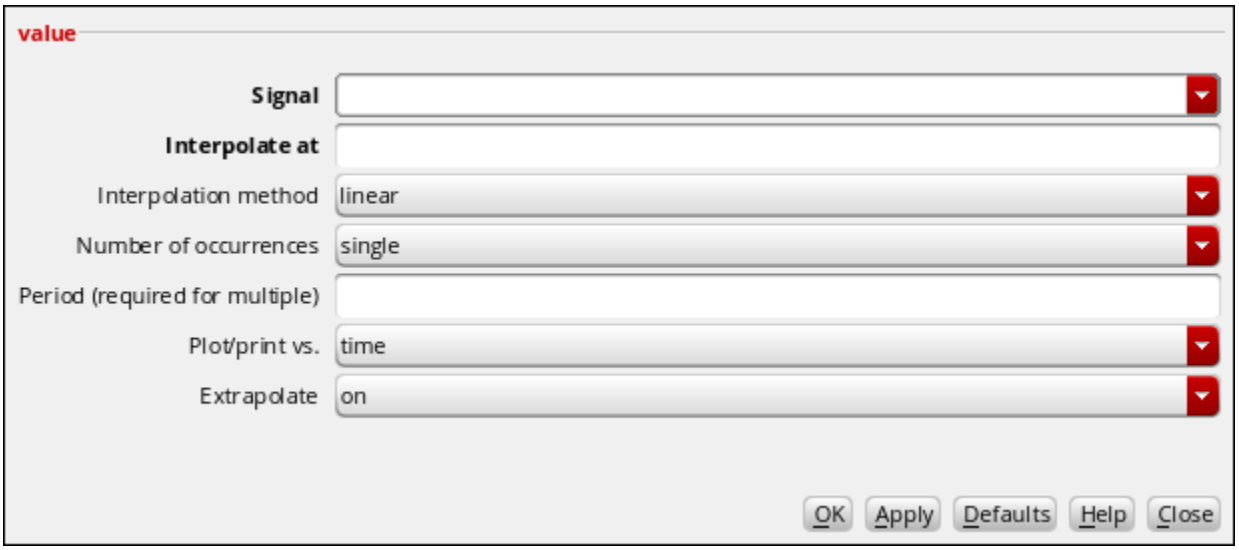
```
v (
     t_net
     [ ?result s_resultName ]
     [ ?resultsDir t_resultsDir ]
     )
     => o_waveform / nil
```

For more information about this OCEAN function, see v in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

value

Computes the Y-axis value of the waveform at the specified X-axis value.



This function includes the following fields:

- *Signal*—Name of the signal.
- *Interpolate At*—X-axis value for which you want the Y-axis value to be computed.
- *Interpolation Method*—Select the interpolation method as `linear` or `roundDown`. When the method is set to `linear`, this function performs the linear interpolation. When set to `roundDown`, the function returns the lower bound of an interval that contains a value to interpolate.
- *Number of occurrences*—Specifies whether you want to retrieve only one occurrence of an interpolated value for the given waveform (`single`) or all interpolated values for the given waveform (`multiple`), which you can later plot or print. The default value is `single`.
- *Period (required for multiple)*—Time between samples that result in the multiple Y values. You must specify this if you selected multiple in the *Number of Occurrences* field.
- *Plot/print vs.*—Specifies whether you want to retrieve value data against time (or another X-axis parameter cycle). The default value is `time`.
- *Extrapolate*—When this argument is set to `on` or not specified, the extrapolation takes place. For example, if a function has `1,2,3,...10` points. Now, if you run `value` function for point `15` (which is greater than `10`) with extrapolation set to default value `yes`

© 2023 Cadence Design Systems, Inc.

(or not specified), the function returns the last point, `10`. When this argument is set to `no`, the extrapolation does not happen and a warning message is displayed. For more information, see Example 2 below.

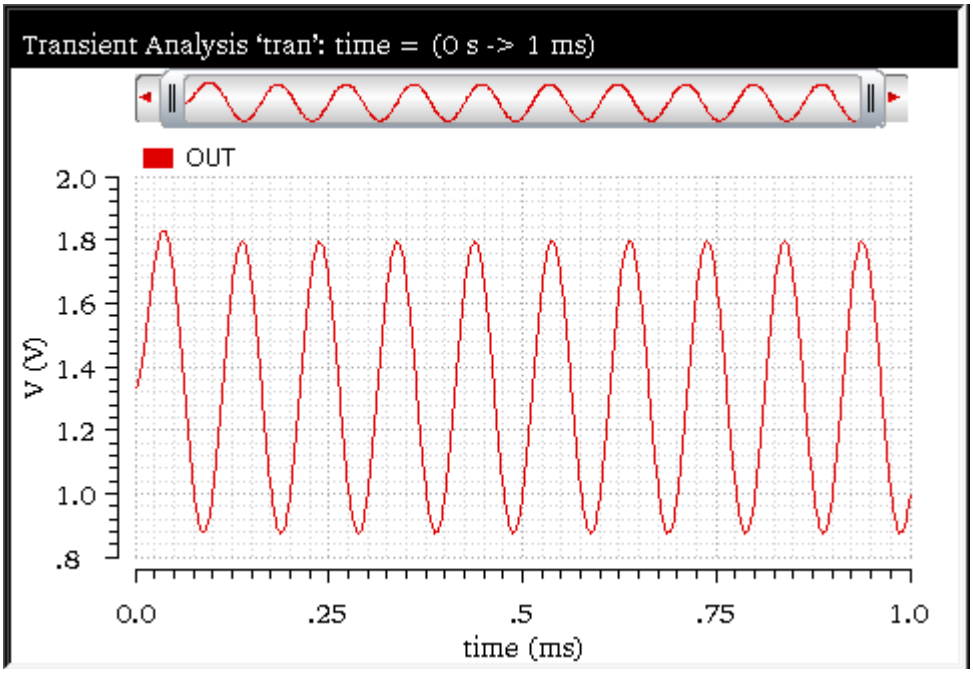
Note: When the extrapolation is on, the `value` function always returns a value. However, when extrapolation is off, an error message is displayed if the specified X-axis value is outside the range of the waveform.

Additional Information

The X-axis interpolation is linear across the independent axis of the waveform.

Example 1

Consider the following input waveform from transient analysis:



The value function is applied on this input signal with the following arguments:

- *Signal*—`v("OUT" ?result "tran" ?resultsDir "./aExamples.raw")`
- *Interpolate At*—`0.1m`
- *Number of occurrences*—`multiple`
- *Period (required for multiple)*—`0.2m`

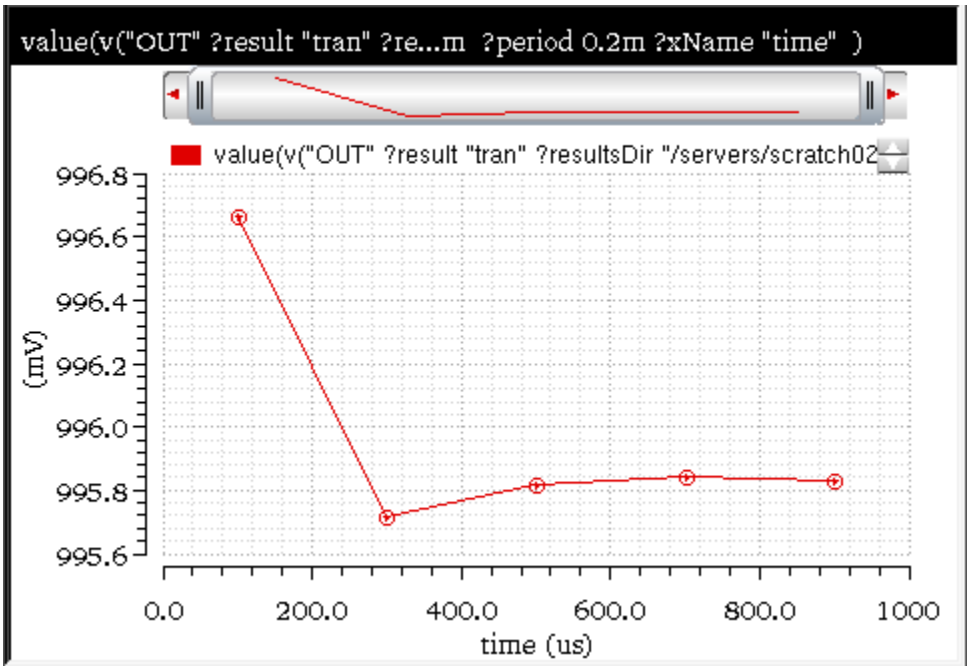
© 2023 Cadence Design Systems, Inc.

- *Plot/print vs.*—`time`
- *Extrapolate*—`no`

The corresponding expression created in the Buffer is as follows:

```
value(v("OUT" ?result "tran" ?resultsDir "./aExamples.raw") 0.1m
?period 0.2m ?xName "time" )
```

When you evaluate this expression, the Y points are measured and the following output waveform is displayed in a new graph window (symbols are enabled in this output waveform).

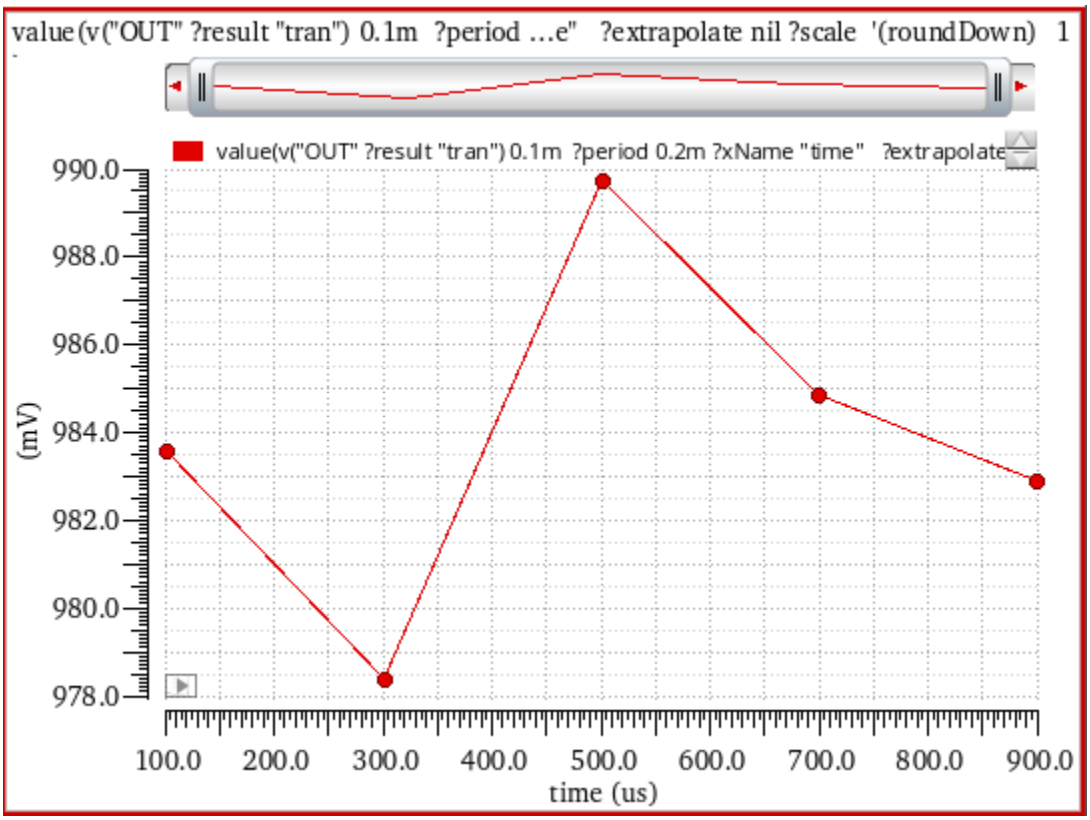


When you specify the Interpolation Method as `roundDown`, the following expression is created in the Buffer:

```
value(v("OUT" ?result "tran") 0.1m  ?period 0.2m ?xName "time"
?extrapolate nil ?scale  '(roundDown) )
```


© 2023 Cadence Design Systems, Inc.

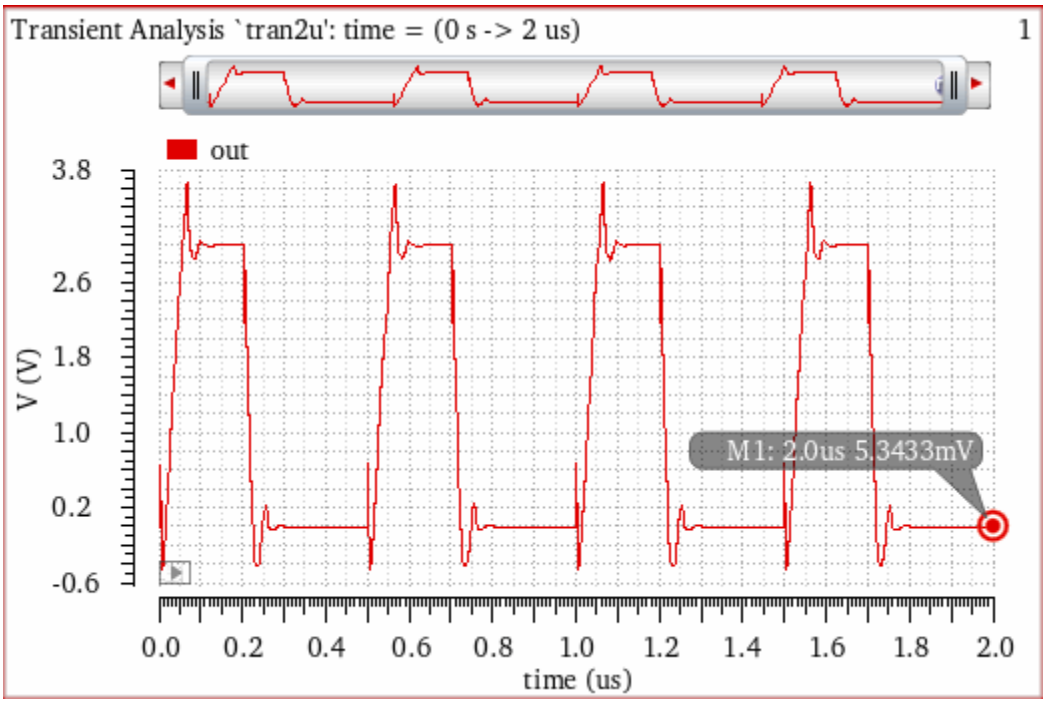
When you evaluate this expression, the following output waveform appears:



© 2023 Cadence Design Systems, Inc.

Example 2

Consider the following waveform from the transient analysis to find out the Y result at `x=50u`, which is past the end of the graph.



Now, apply the `value` function on this waveform with the following arguments:

- *Signal*—`v("OUT" ?result "tran" ?resultsDir "./ampsim.raw")`
- *Interpolate At*—`50u`
- *Interpolation Method*—`linear`
- *Number of occurrences*—`multiple`
- *Period (required for multiple)*—`nil`
- *Plot/print vs.*—`time`
- *Extrapolate*—`no`

The following expression is created in the Buffer:

```
value(v("out" ?result "tran") 50u   ?extrapolate nil )
```

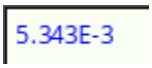

© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the expression evaluation fails and an error message is displayed in CIW because the X-axis value specified in *Interpolate At* argument is outside the range of the given signal.

In this case, if you set the *Extrapolate* argument to its default value, `yes`, the following expression is created in the Buffer:

```
value(v("out" ?result "tran") 50u )
```

Now, when you evaluate this expression, the `value` function returns the last point of the signal waveform in the Buffer, as shown below:



Notice that this output value matches with the Y-axis value given in the marker M1 that displays the X and Y-axis values of the last point.

Related OCEAN Function

The equivalent OCEAN command for `value` is:

```
value( o_waveform [s_name] g_value ?period n_period
     [g_multiple [s_Xname]] [g_histoDisplay][x_noOfHistoBins])
     => o_waveform/g_value/nil
```

For more information, see `value` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

waveVsWave

Computes the YvsY function on the given input waveforms and plots the output waveform for which X- and Y- axes are created using the Y-axes of the specified input waveforms. When the specified input waveforms have different X-axes, this function performs the interpolation. You can also use this function to compare the Y-axis of a family of waveforms with the Y-axis of a single-leaf waveform.

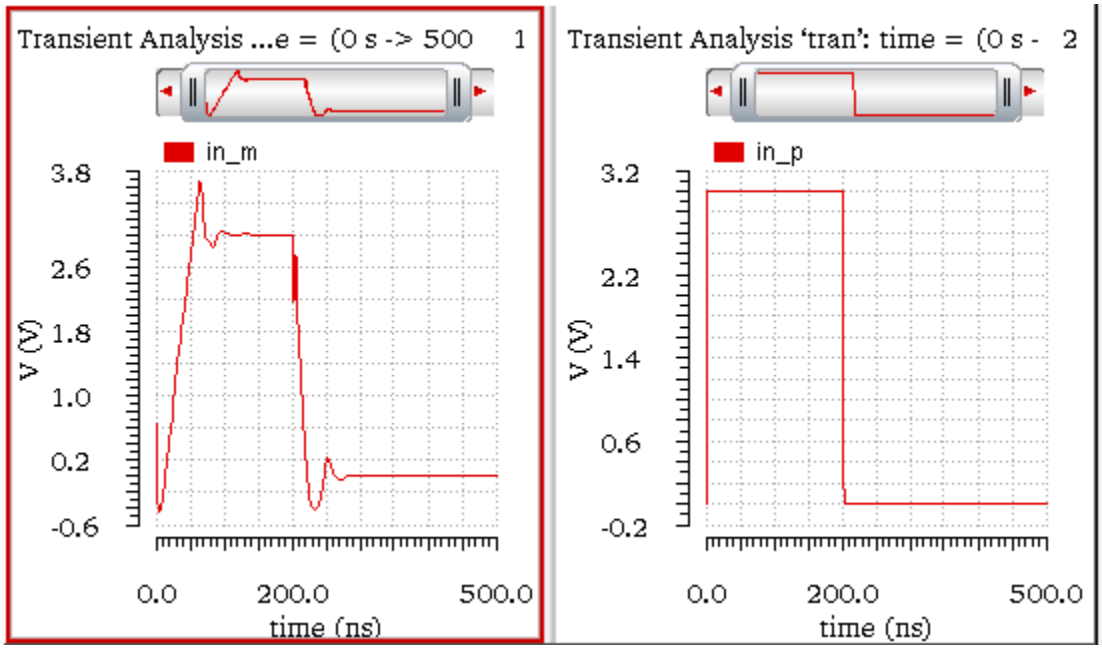
This function includes the following fields:

- *Select xTrace*—Specify input waveform whose Y-axis you want to use as the Y-axis of the output waveform.
- *Select yTrace*—Specify input waveform whose Y-axis you want to use as the X-axis of the output waveform.

© 2023 Cadence Design Systems, Inc.

Example 1 (Wave versus Wave)

Consider the following input signals from the transient analysis: `in_m` and `in_p`:



Now, specify the following argument values in the `waveVsWave` function:

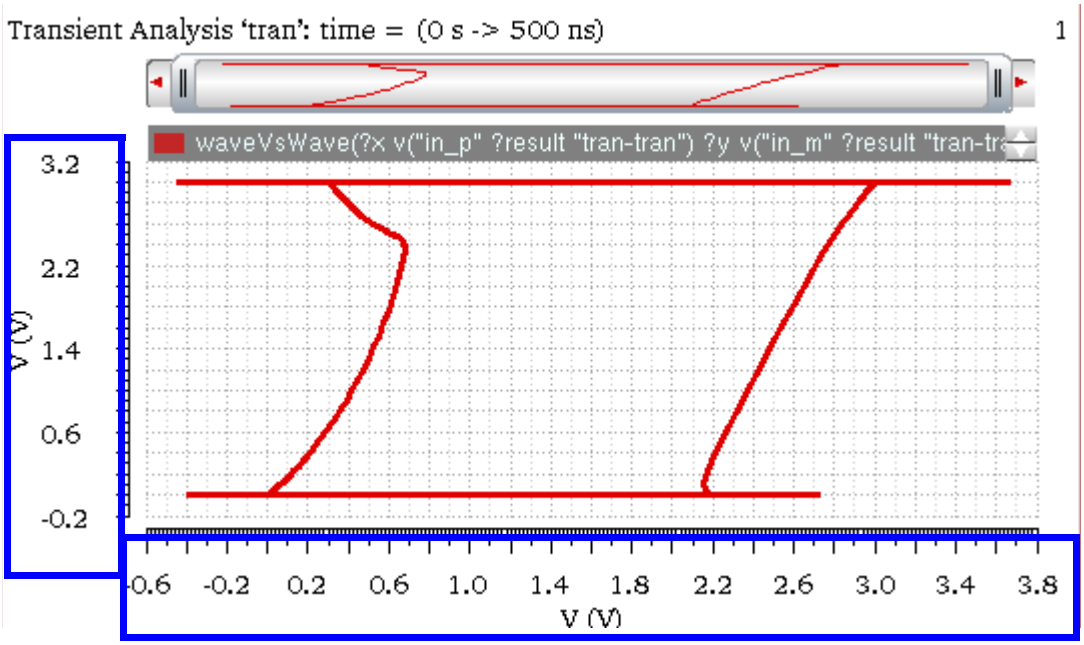
- *Select xTrace*—`v("in_p" ?result "tran-tran" ?resultsDir ../ampsim.raw)`
- *Select yTrace*—`v("in_m" ?result "tran-tran" ?resultsDir ../ampsim.raw)`

The following expression is created in the Buffer:

```
waveVsWave(?x v("in_p" ?result "tran-tran") ?y v("in_m" ?result
"tran-tran"))
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the YvsY of these two input waveforms is computed and the output waveform, as shown in the below figure, is plotted in the graph window. The X-axis of this output waveform is same as that of `in_m` and Y-axis is same as `in_p`.

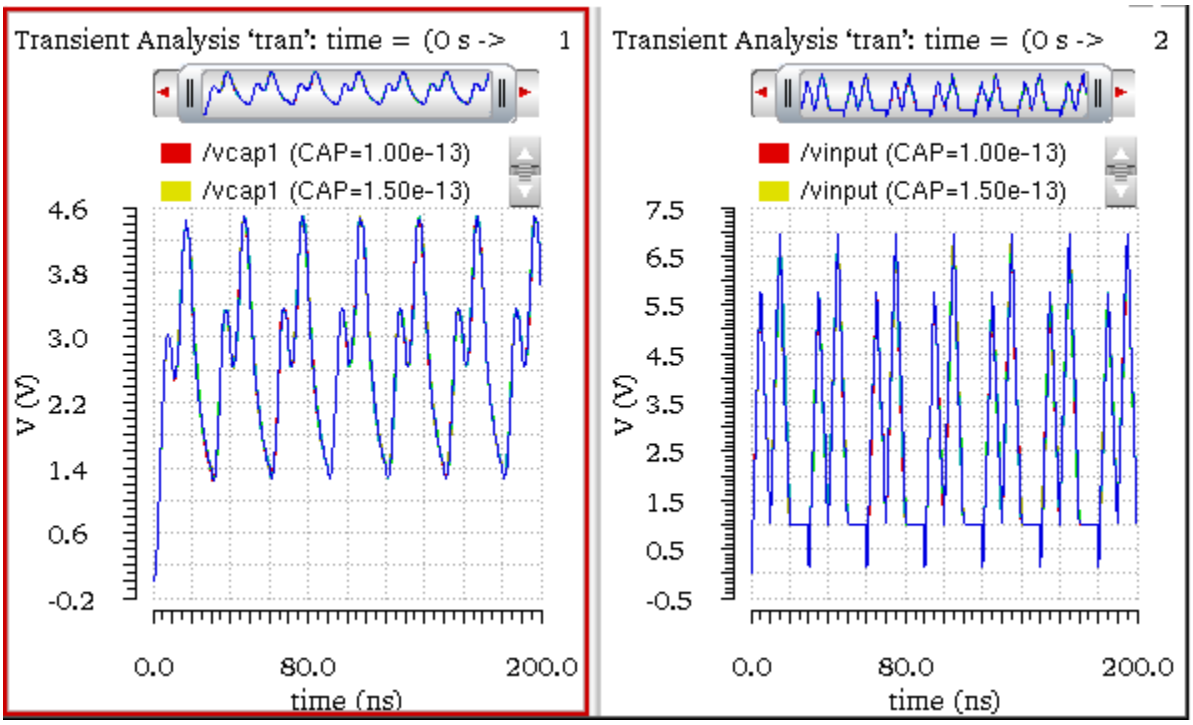


Note: You can also use the expressions signals, such as `BW` and `Abs`, to calculate the YvsY of two waveforms.

© 2023 Cadence Design Systems, Inc.

Example 2 (Wave versus Wave)

This example shows how the `waveVsWave` function is applied on two transient signals. Consider the following input signals from a transient waveform: `v("/vinput" ?result "tran")` and `v("/vcap1" ?result "tran")`



The following arguments are specified in this example:

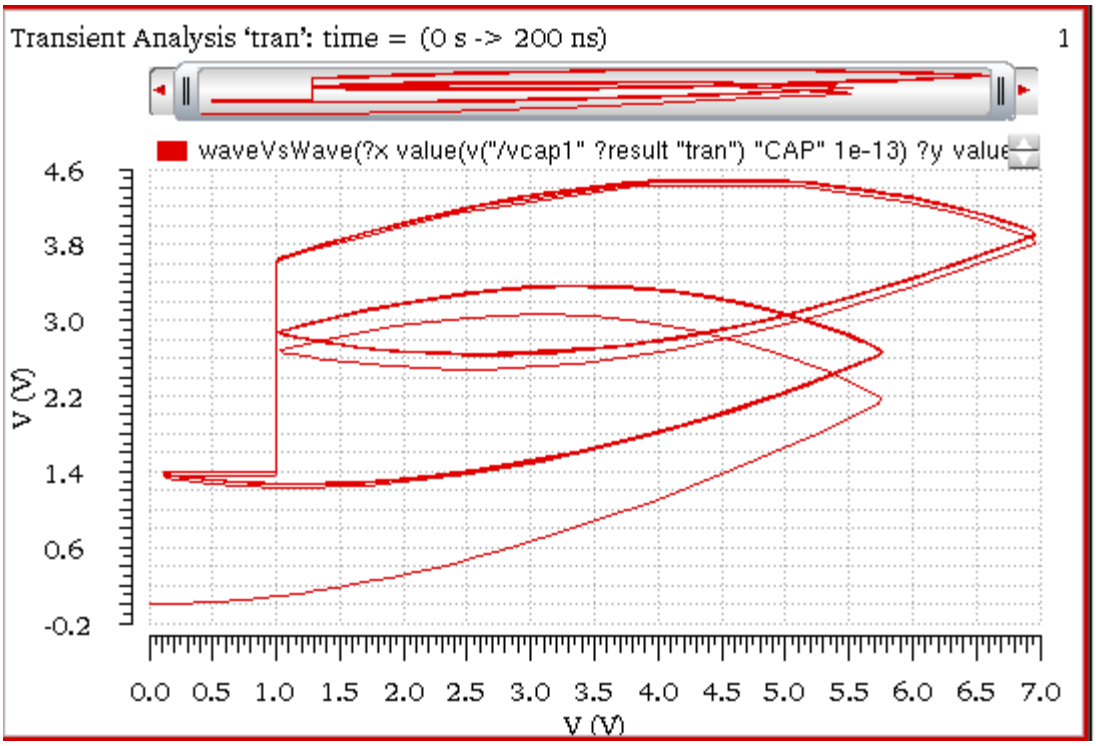
- *Select xTrace*—`value(v("/vcap1" ?result "tran" ?resultsDir ../peakTest/spectre/schematic/psf) "CAP" 1e-13)`
- *Select yTrace*—`value(v("/vinput" ?result "tran" ?resultsDir ../peakTest/spectre/schematic/psf) "CAP" 1e-13)`

The following expression is created in the Buffer:

```
waveVsWave(?x value(v("/vcap1" ?result "tran") "CAP" 1e-13) ?y
value(v("/vinput" ?result "tran") "CAP" 1e-13))
```


© 2023 Cadence Design Systems, Inc.

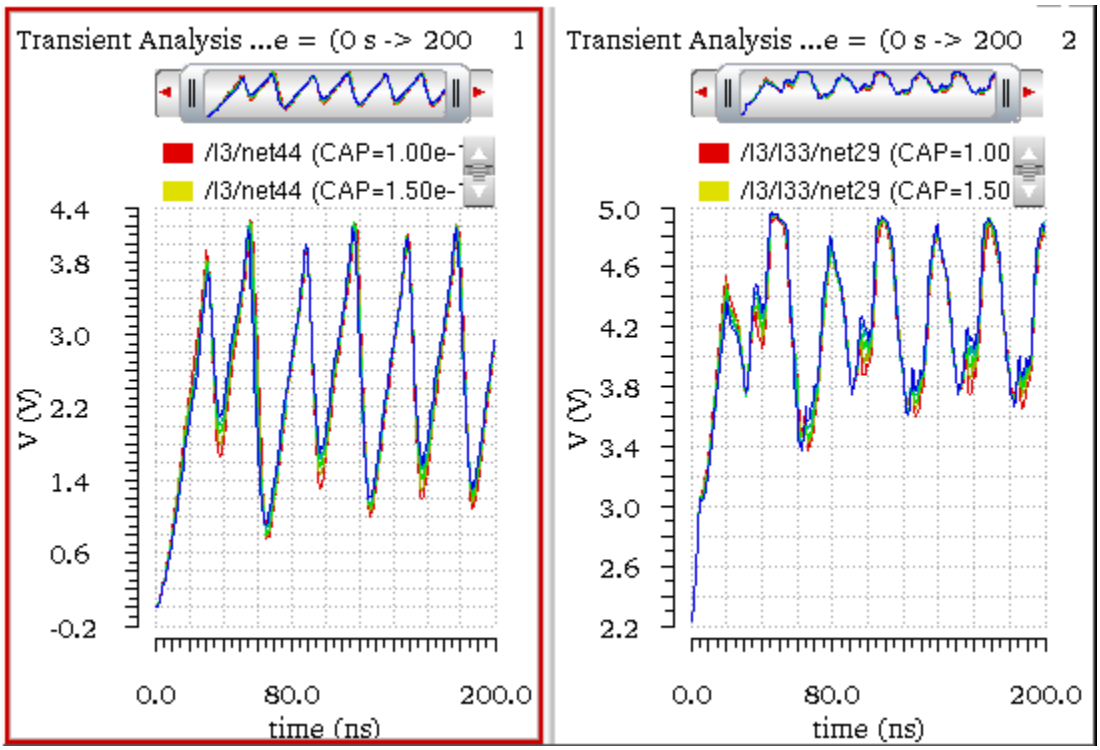
When you evaluate this expression, the YvsY of these two input waveforms is computed and the output waveform, as shown in the below figure, is plotted in the graph window. The X-axis of this output waveform is same as that of `vinput` and Y-axis is same as `vcap1`.



© 2023 Cadence Design Systems, Inc.

Example 3 (Family versus Family)

This example shows how the `waveVsWave` function is used to calculate the YvsY of two waveform families. Consider the following input signals from a transient waveform: `v("/I3/net44" ?result "tran")` and `v("/I3/net29" ?result "tran")`



The following arguments are specified in this example:

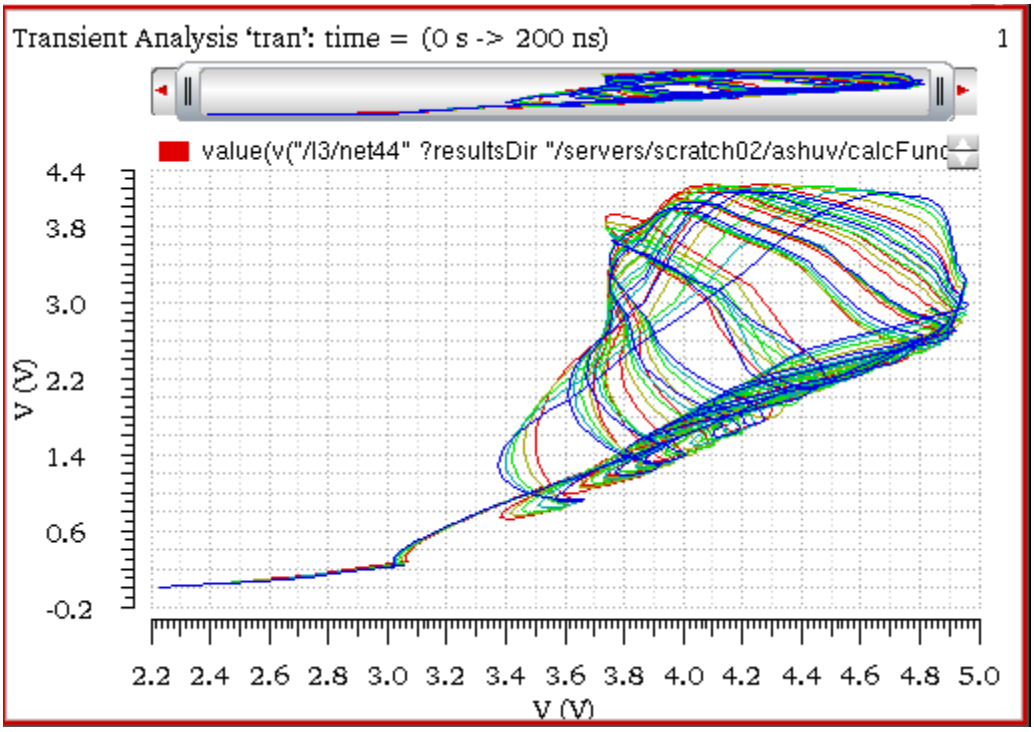
- *Select xTrace*—`v("/I3/net44" ?result "tran" ?resultsDir ../peakTest/spectre/schematic/psf)`
- *Select yTrace*—`v("/I3/net29" ?result "tran" ?resultsDir ../peakTest/spectre/schematic/psf)`

The following expression is created in the Buffer:

```
waveVsWave(?x v("/I3/net44" ?result "tran") ?y v("/I3/I33/net29"
?result "tran"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the YvsY of these two families is computed and the output waveform, as shown in the below figure, is plotted in the graph window. The X-axis of this output waveform is same as that of `/I3/net29` and Y-axis is same as `/I3/net44`.



Example 4 (Family versus Leaf)

This example show how `waveVsWave` function is used to calculate YvsY of a waveform family versus leaf waveform. Consider the input waveforms from the following two waveform families shown in Example 2:

- A leaf waveform from `vinput` family: `value(v("/vinput" ?result "tran") "CAP" 2.5e-13)`
- Waveform family: `/vcap1`

The following arguments are specified in this example:

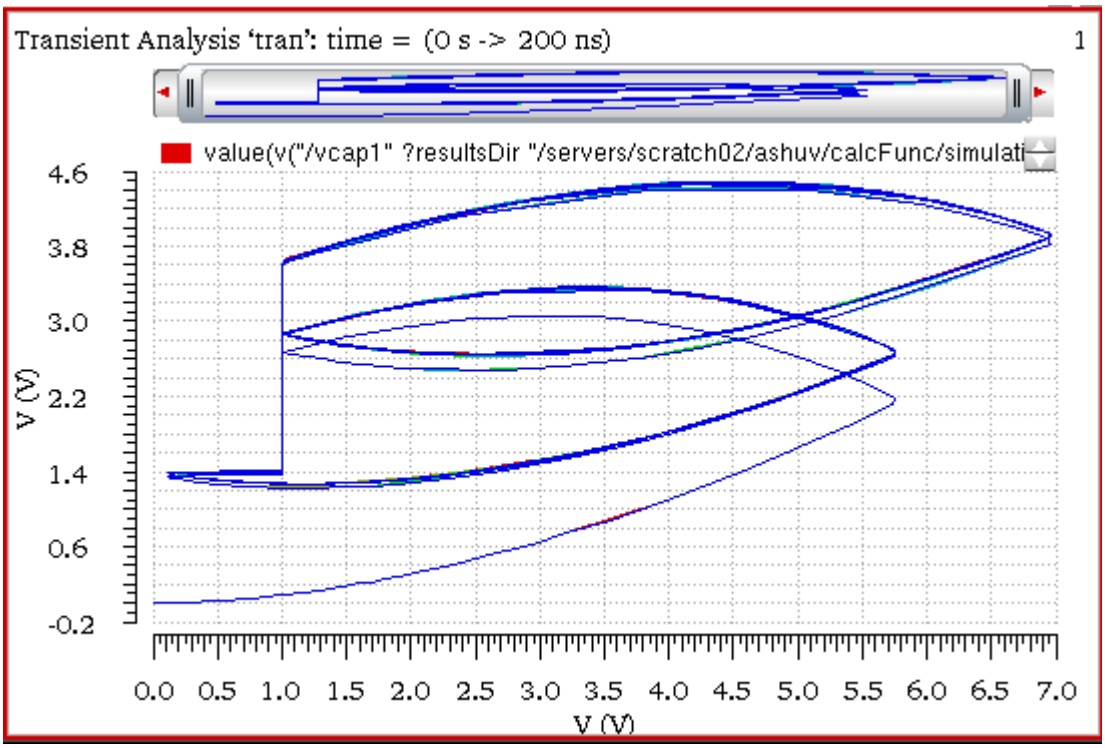
- *Select xTrace*—`value("/vcap1" ?result "tran" ?resultsDir ../peakTest/spectre/schematic/psf)`
- *Select yTrace*—`value(v("/vinput" ?result "tran" resultsDir ../peakTest/spectre/schematic/psf) "CAP" 2.5e-13)`

The following expression is created in the Buffer:

© 2023 Cadence Design Systems, Inc.

```
waveVsWave(?x v("/vcap1" ?result "tran") ?y value(v("/vinput" ?result
"tran") "CAP" 2.5e-13))
```

When you evaluate this expression, the YvsY of these two families is computed and the output waveform, as shown in the below figure, is plotted in the graph window. The X-axis of this output waveform is same as that of leaf waveform `/vinput` and Y-axis is same as family `/vcap1`.



Example 4 (Scalar versus Scalar)

When you apply the `waveVsWave` function on expression waveforms that result in a scalar value, the resultant output is a waveform.

Consider the following example in which you have created the following output in ADE for the `waveVsWave` function

```
waveVsWave(?x SettlingTime ?y SlewRate)
```

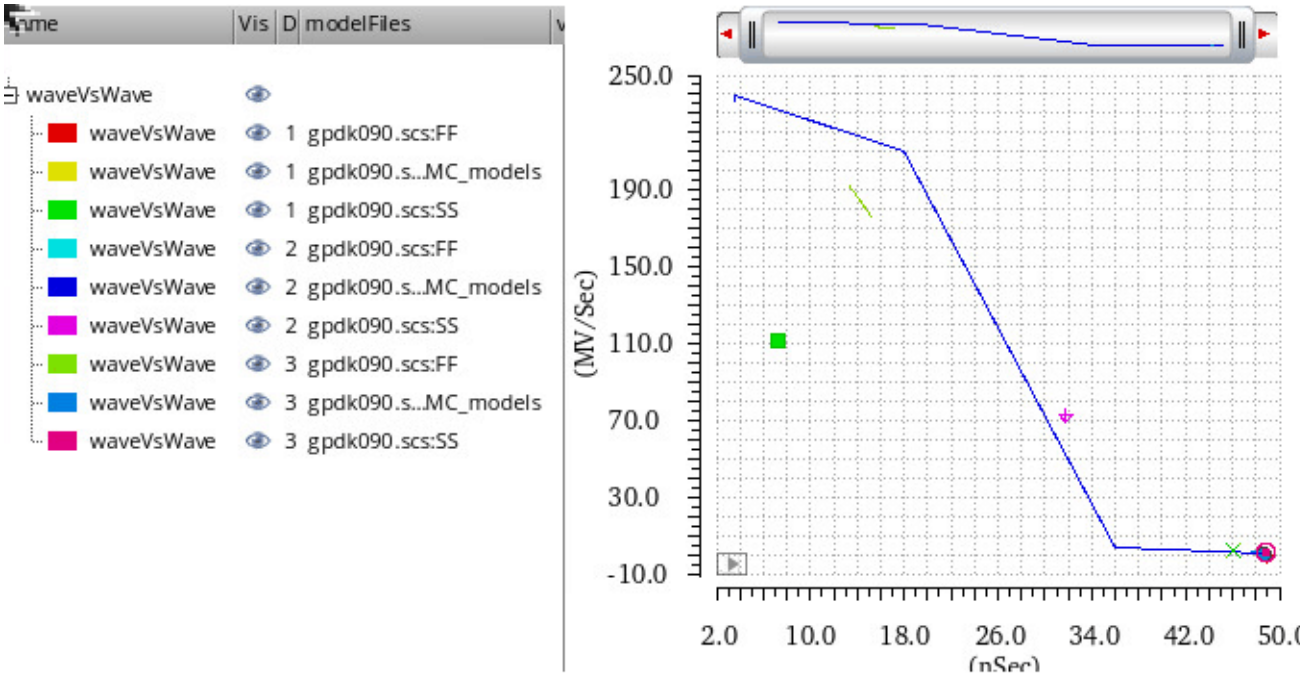
Ensure that the evaluation type for this expression should be selected as `corners`.

© 2023 Cadence Design Systems, Inc.

The `settlingTime` and `slewrate` functions generate a scalar value when they are evaluated individually.

AN:1	waveVsWave	expr	waveVsWave(?x SettlingTime ?y SlewRate)	corners

Now, after the simulation is run in ADE, this expression is evaluated and the following output waveform is plotted in the graph window.



© 2023 Cadence Design Systems, Inc.

xmax

Computes the value of the independent variable x at which the expression attains its maximum value, that is, the value of x that maximizes y=f(x).

The maximum might occur at more than one point on the X-axis, so you must choose (in the *Nth Maximizer* field) which maximum value you want to see. The Calculator returns the value of the *Nth Maximizer* counting from the left, that is, toward increasing X-axis values. If you enter a negative integer, the direction of search is reversed toward decreasing X-axis values (counting from the right). This function is available only in the SKILL mode

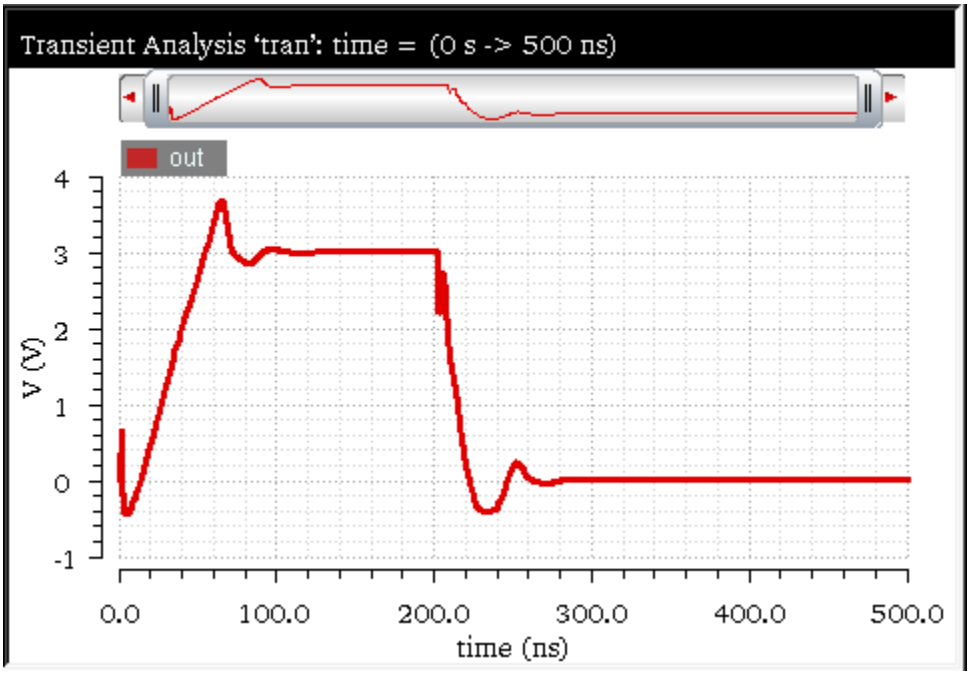
This function includes the following arguments:

- *Signal*—Name of the signal.
- *Nth Maximizer*—Maximum value of the expression.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from the transient analysis:



The `xmax` function is applied on this input waveform with the following arguments:

- *Signal*—`v("out" ?result "tran-tran" ?resultsDir "ampsim.raw")`
- *Nth Maximizer*—`1`

The corresponding expression created in the Buffer is as follows:

```
xmax(v("out" ?result "tran-tran") 1 )
```

When you evaluate this expression, the following output indicating the maximum X-value is displayed in the Buffer.

```
65.0E-9
```

Related OCEAN Function

The equivalent OCEAN command for `xmax` is:

```
xmax( o_waveform x_numberOfPeaks )
     => o_waveform/g_value/l_value/nil
```


© 2023 Cadence Design Systems, Inc.

For more information, see `xmax` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

xmin

Computes the value of the independent variable x at which the expression has its minimum value, that is, the value of x that minimizes y=f(x).

The minimum might occur at more than one point on the x axis, so you must choose (in the *Nth Minimizer* field) which minimum value you want to see. The calculator returns the value of the *Nth Minimizer*, counting from the left, that is, toward increasing X-axis values. If you enter a negative integer, the direction of search is reversed toward decreasing X-axis values (counting from the right). This function is available only in the SKILL mode.

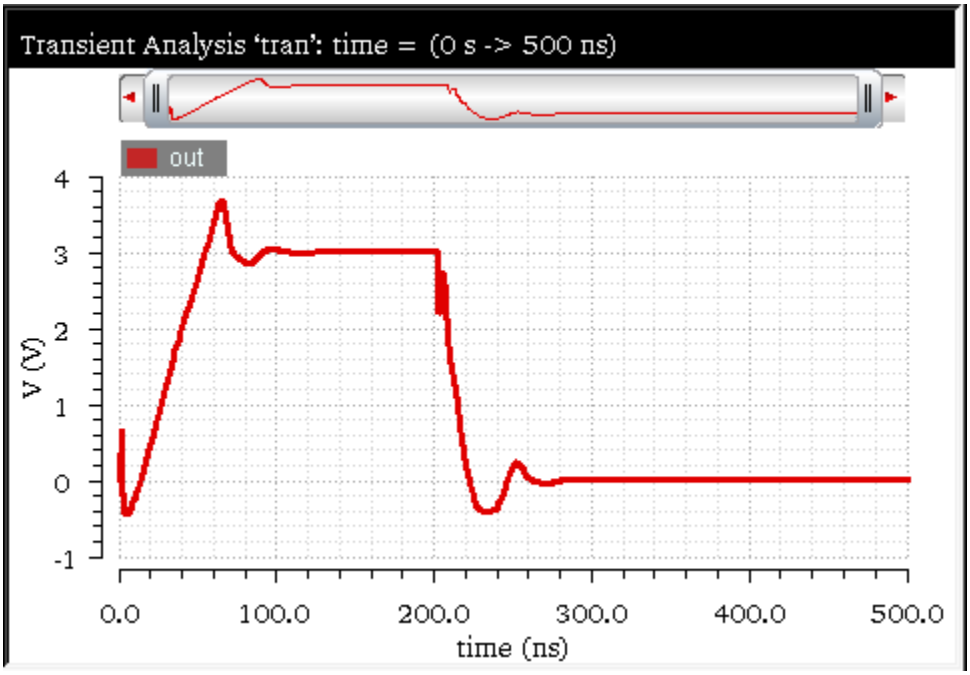
This function includes the following arguments:

- *Signal*—Name of the signal.
- *Nth Minimizer*—Minimum value of the expression.

© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from the transient analysis:



The `xmin` function is applied on this input waveform with the following arguments:

- *Signal*—`v("out" ?result "tran-tran" ?resultsDir "ampsim.raw")`
- *Nth Maximizer*—`1`

The corresponding expression created in the Buffer is as follows:

```
xmin(v("out" ?result "tran-tran") 1 )
```

When you evaluate this expression, the following output indicating the minimum X value is displayed in the Buffer.

```
4.346E-9
```

Related OCEAN Function

The equivalent OCEAN command for `xmin` is:

```
xmin( o_waveform x_numberOfValleys )
     => o_waveform/g_value/l_value/nil
```

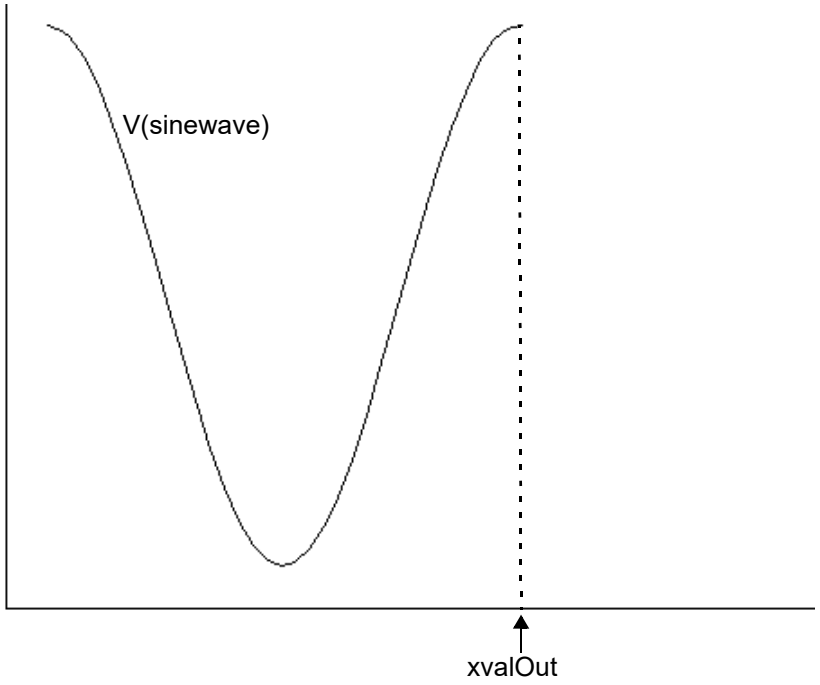

© 2023 Cadence Design Systems, Inc.

For more information, see `xmin` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

xval

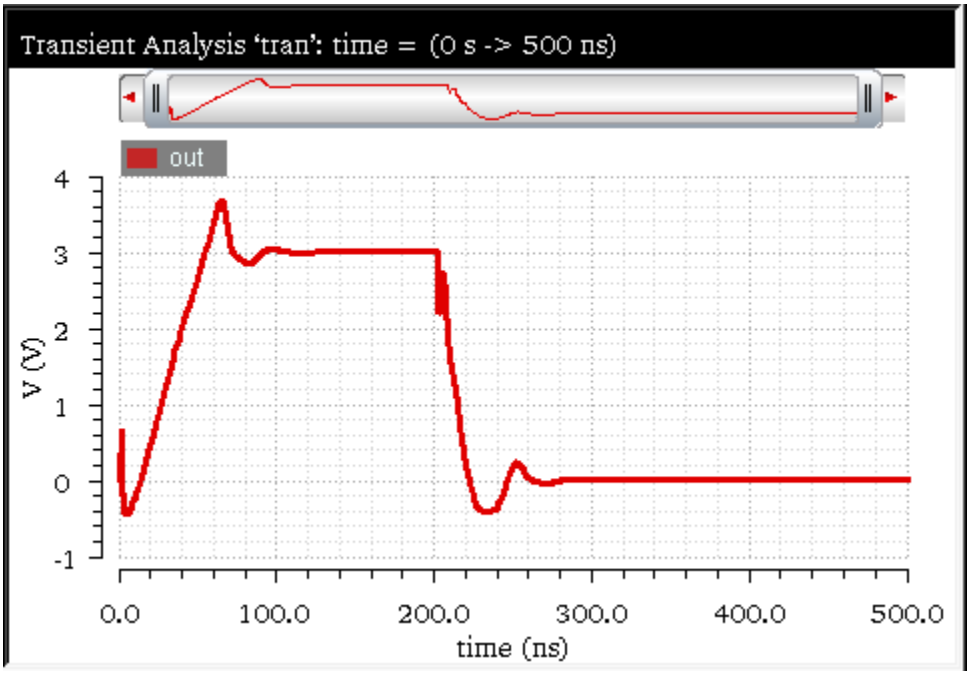
Returns the vector consisting of the X-axis values of the points in the signal. The following figure illustrates this function.



© 2023 Cadence Design Systems, Inc.

Example

Consider the following input waveform from the transient analysis:

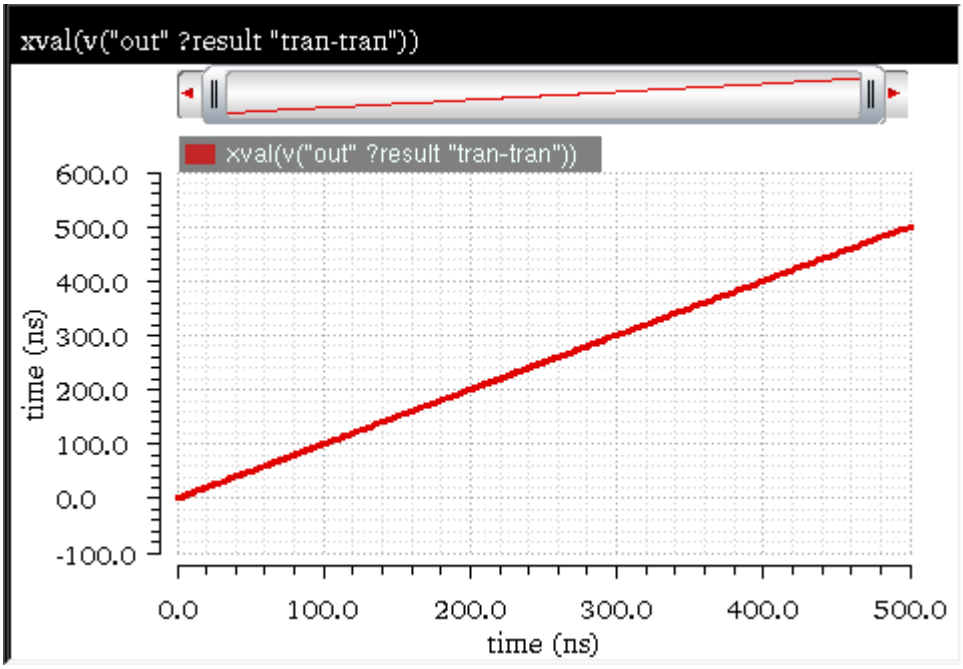


The expression for this input signal is displayed in the Buffer. Now, when you select the `xval` function from the Function Panel, the function is applied to the signal expression in the Buffer. The corresponding expression created in the Buffer is as follows:

```
xval(v("out" ?result "tran-tran"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the corresponding X vector values are plotted in a new graph window as shown in the figure below:



Related OCEAN Function

The equivalent OCEAN command for `xval` is:

```
xval( o_waveform )
     => o_waveform/nil
```

For more information, see `xval` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

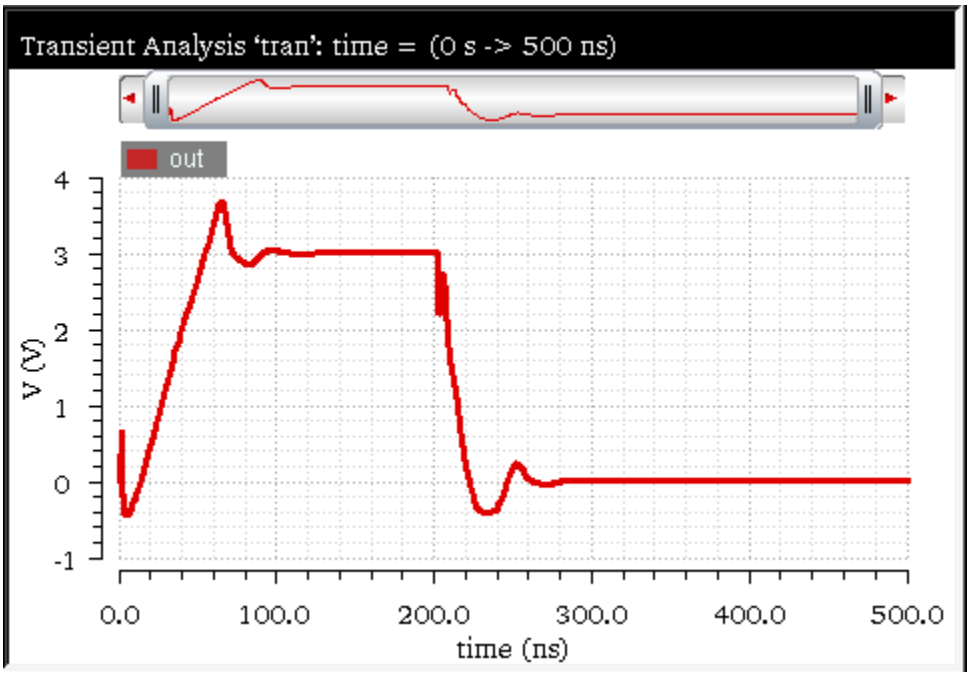
ymax

Computes the maximum Y-axis value of the expression y=f(x). This function is available only in the SKILL mode.

This function can be applied to both scalar and vector inputs.

Example

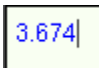
Consider the following input waveform from the transient analysis:



The expression for this input signal is displayed in the Buffer. Now, when you select the `ymax` function from the Function Panel, the function is applied to the signal expression in the Buffer. The corresponding expression is displayed in the Buffer as follows:

```
ymax(v("out" ?result "tran-tran"))
```

When you evaluate this expression, the following output showing maximum Y-axis value is displayed in the Buffer:



© 2023 Cadence Design Systems, Inc.

When you use this function to evaluate data for measurement across corners, a new key argument, `overall`, is added to this function. If the input is a scalar and the `overall` argument is set to `t`, the `ymax` function returns the input scalar value.

Related OCEAN Function

The equivalent OCEAN command for `ymax` is:

```
ymax( o_waveform )
     => n_max/o_waveformMax/nil
```

For more information, see `ymax` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

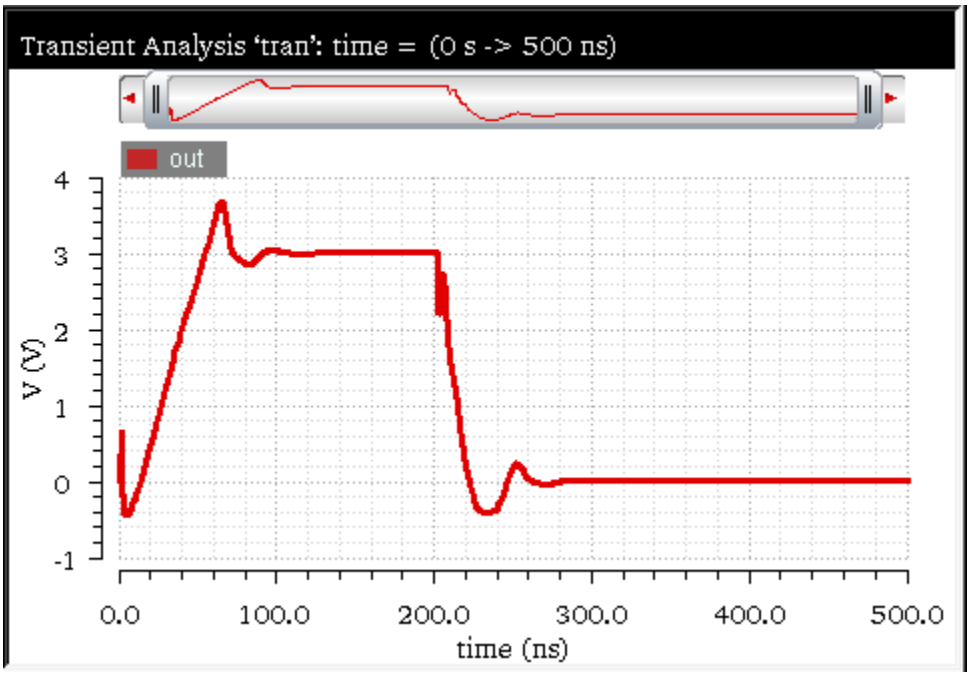
ymin

Computes the minimum Y-axis value of the expression y=f(x). This function is available only in the SKILL mode.

This function can be applied to both scalar and vector inputs.

Example

Consider the following input waveform from the transient analysis:



The expression for this input signal is displayed in the Buffer. Now, when you select the `ymin` function from the Function Panel, the function is applied to the expression in the Buffer. The corresponding expression created in the Buffer is as follows:

```
ymin(v("out" ?result "tran-tran"))
```

When you evaluate this expression, the following output showing minimum Y-axis value is displayed in the Buffer:

```
-444.3E-3
```


© 2023 Cadence Design Systems, Inc.

When you use this function to evaluate data for measurement across corners, a new key argument, `overall`, is added to this function. If the input is a scalar and the `overall` argument is set to `t`, the `ymin` function returns the input scalar value.

Related OCEAN Function

The equivalent OCEAN command for `ymin` is:

```
ymin( o_waveform )
     => n_min/o_waveformMin/nil
```

For more information, see `ymin` in *OCEAN Reference.*

© 2023 Cadence Design Systems, Inc.

Modifier Functions

This section describes the following modifier functions available in the Virtuoso Visualization and Analysis XL Calculator:

- conjugate
- dB10
- dB20
- imag
- real
- mag
- phase
- phaseRad
- phaseDeg
- phaseDegUnwrapped
- phaseRadUnwrapped

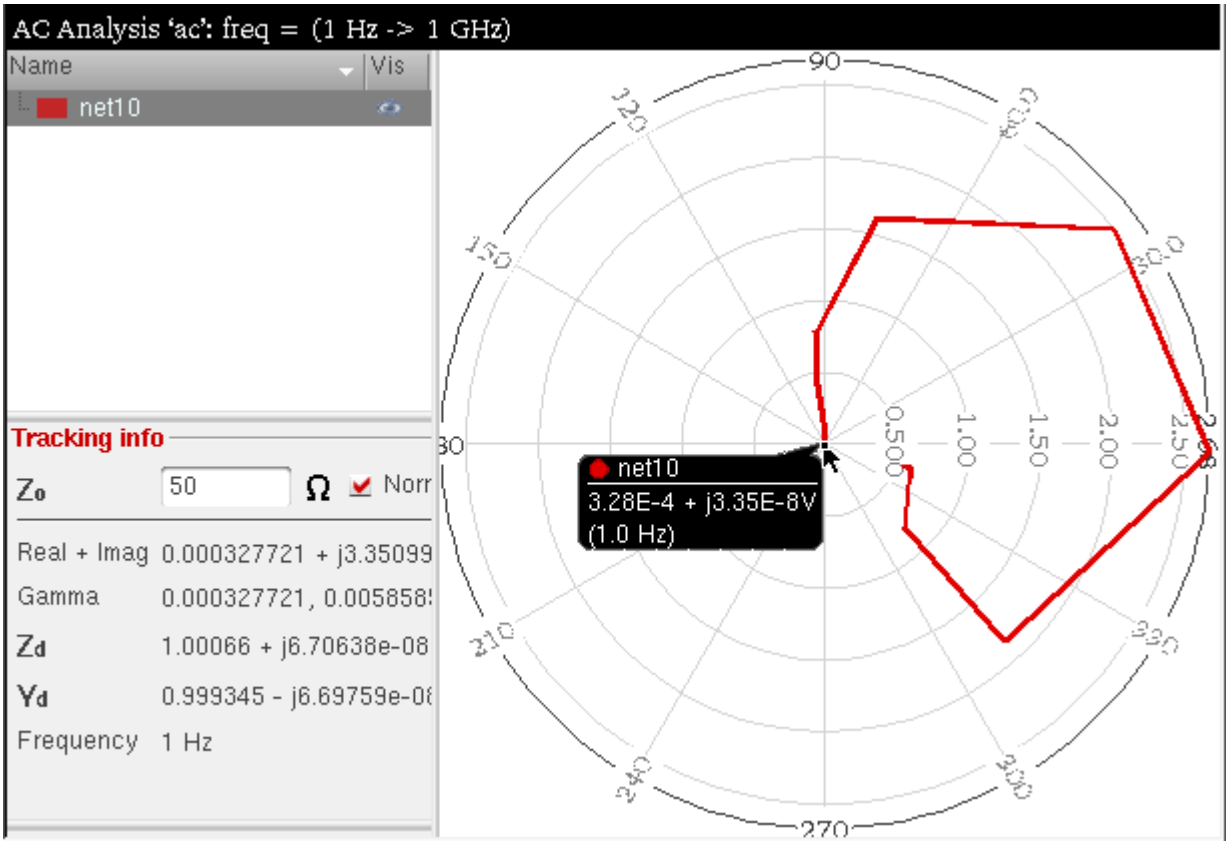
© 2023 Cadence Design Systems, Inc.

conjugate

Returns the conjugate of a complex number. In the conjugate, the magnitudes of real and imaginary parts of the complex conjugate are same as that of the given complex number, and the imaginary part is of the opposite sign. For example, the conjugate of `i+jk` is `i-jk`.

Example

Consider the following signal expression from the AC analysis for the complex data:



When you send this signal to Calculator, the following expression is displayed in the Buffer:

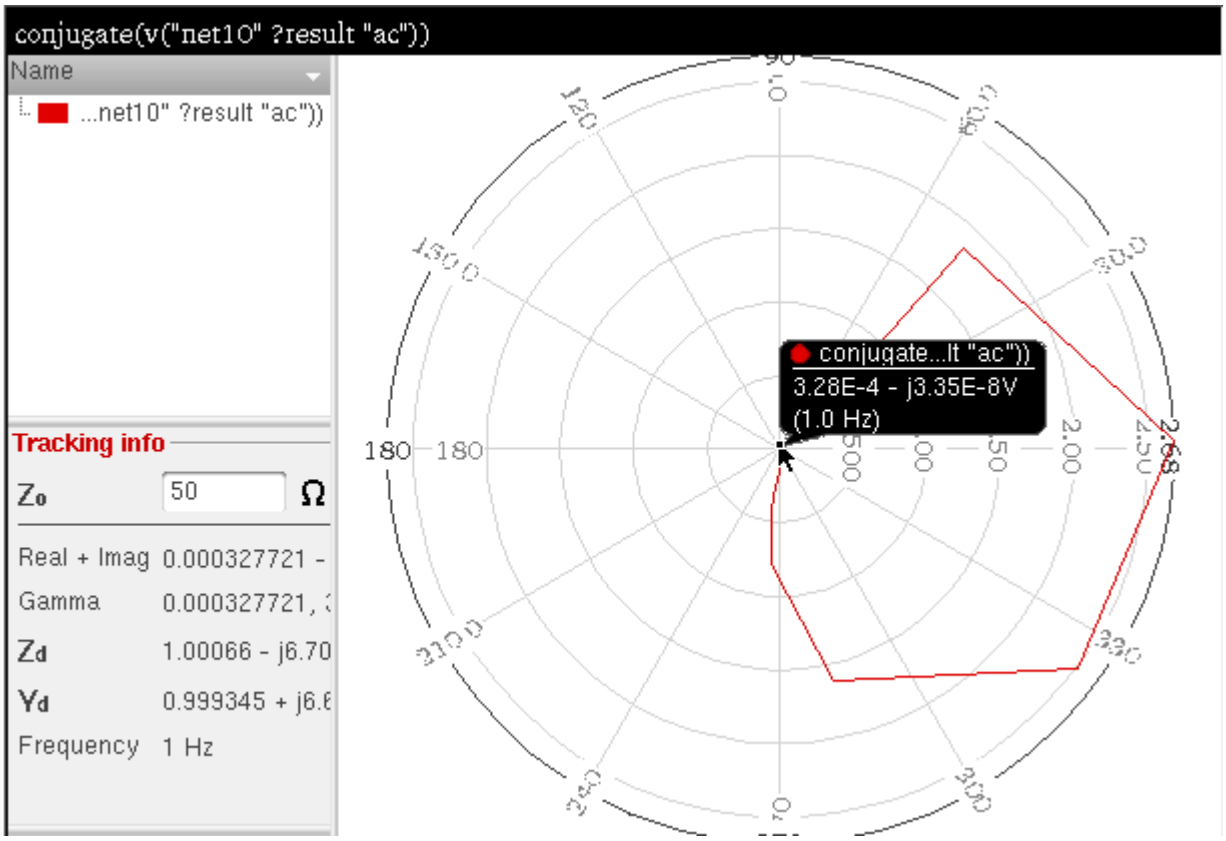
```
mag(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```

By default, the expression is displayed with the `mag` function. You need to remove this function from the Buffer expression manually When you select the `conjugate` function from the Function Panel, the function is applied on the expression in the Buffer. The expression created in the Buffer is as follows:

© 2023 Cadence Design Systems, Inc.

```
conjugate(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```

When you evaluate this expression, the following waveform is displayed in a new graph window that represents the conjugate of the input waveform.

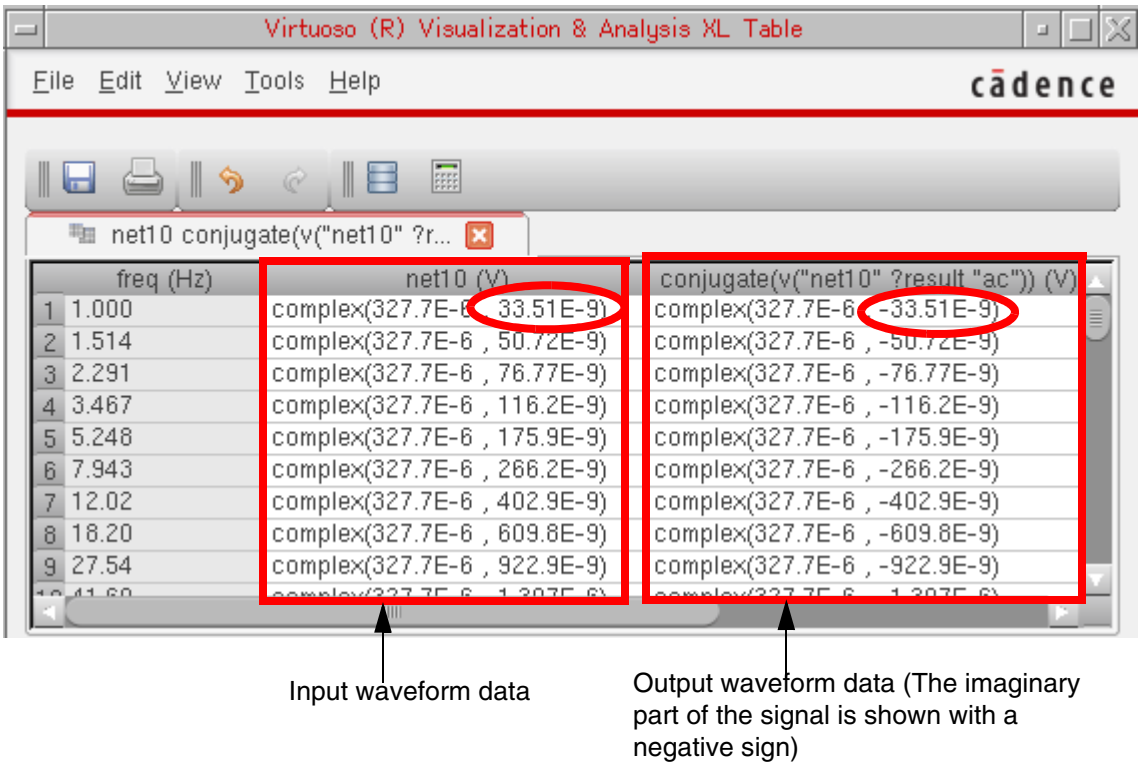


To analyze these generated output values, you can display the input and output signals in the Virtuoso Visualization and Analysis XL Table. To send the input signal to the table, right-click the signal and choose *Send To – Table – New Window*. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

To view both the real and imaginary values in the table, right-click the table row and choose *Display Complex As – Real and Imaginary*. The following table contents are displayed when you send the input and output signals described in this example to the table and display the values in the complex number format.

© 2023 Cadence Design Systems, Inc.

The following figure displays the tabular values for input and output signals.



Related OCEAN Function

The equivalent OCEAN command for `conjugate` is:

```
conjugate( {o_waveform | n_x} )
     => o_waveform/n_y/nil
```

For more information, see `conjugate` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

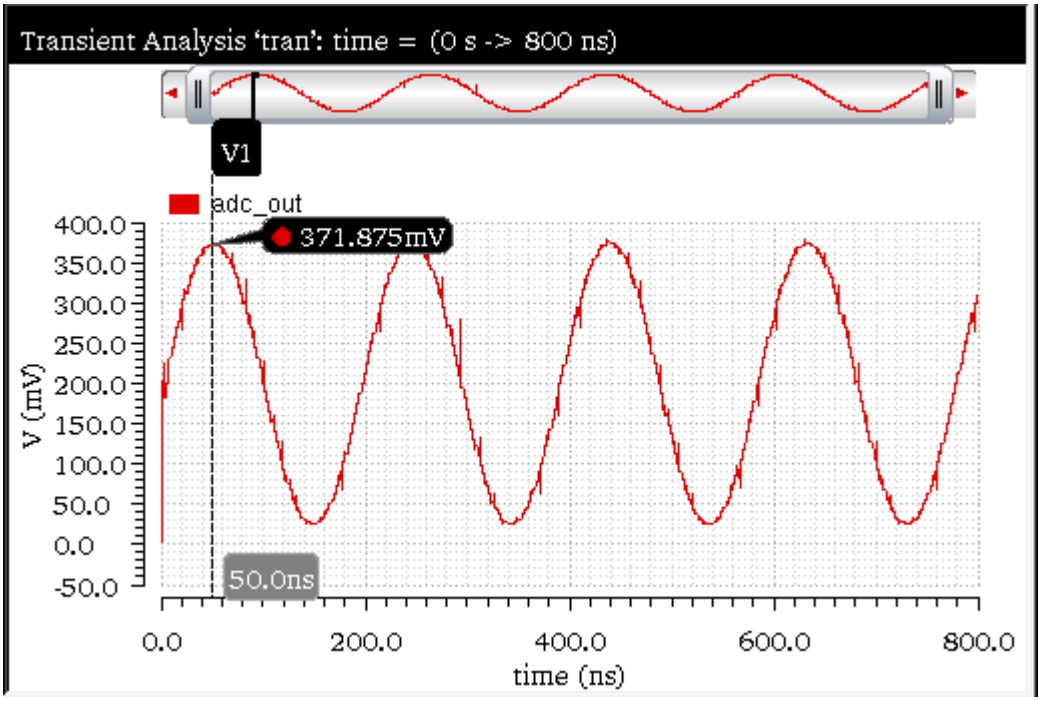
dB10

Converts a signal to dB by using the following equation:

```
dB10(x)=10/log(10)*log(mag(x))
```

Example

Consider the following input signal waveform from the transient analysis:



A vertical marker `V1` is placed at `time=50ns` that displays the voltage value on Y-axis as `371.875mv`.

When you send this signal to Calculator, the following expression is displayed in the Buffer:

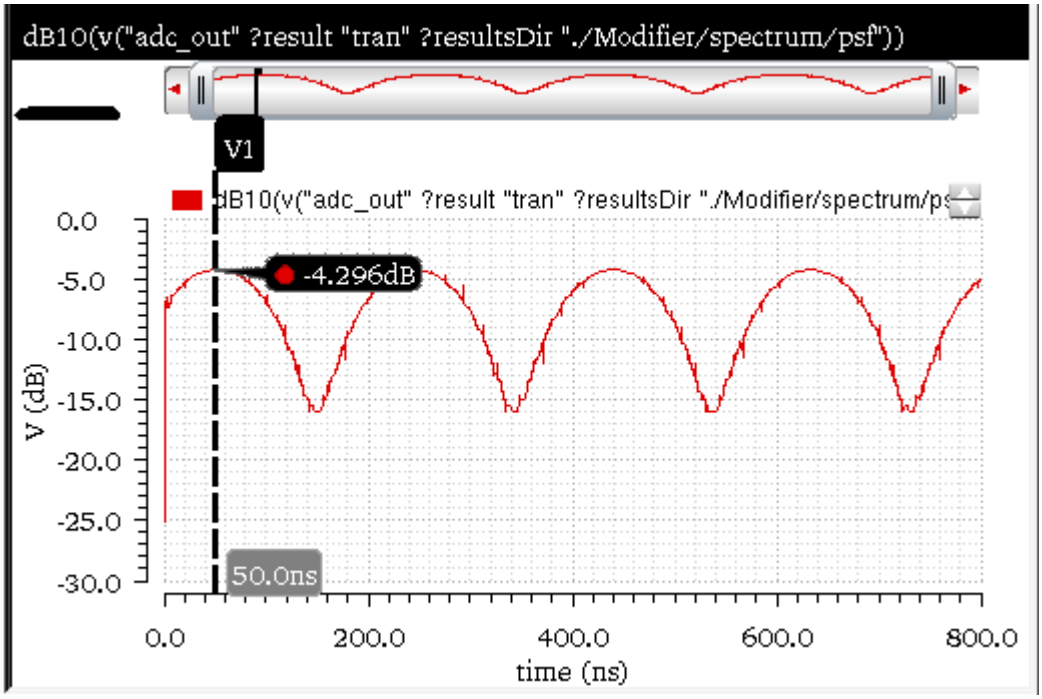
```
v("adc_out" ?result "tran" ?resultsDir "./Modifier/spectrum/psf")
```

When you select the `dB10` function from the Function Panel, the function is applied on the expression in the Buffer. The expression created in the Buffer is as follows:

```
dB10(v("adc_out" ?result "tran" ?resultsDir "./Modifier/spectrum/
psf"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following waveform is displayed in a new graph window that represents the dB10 calculation of the input waveform.

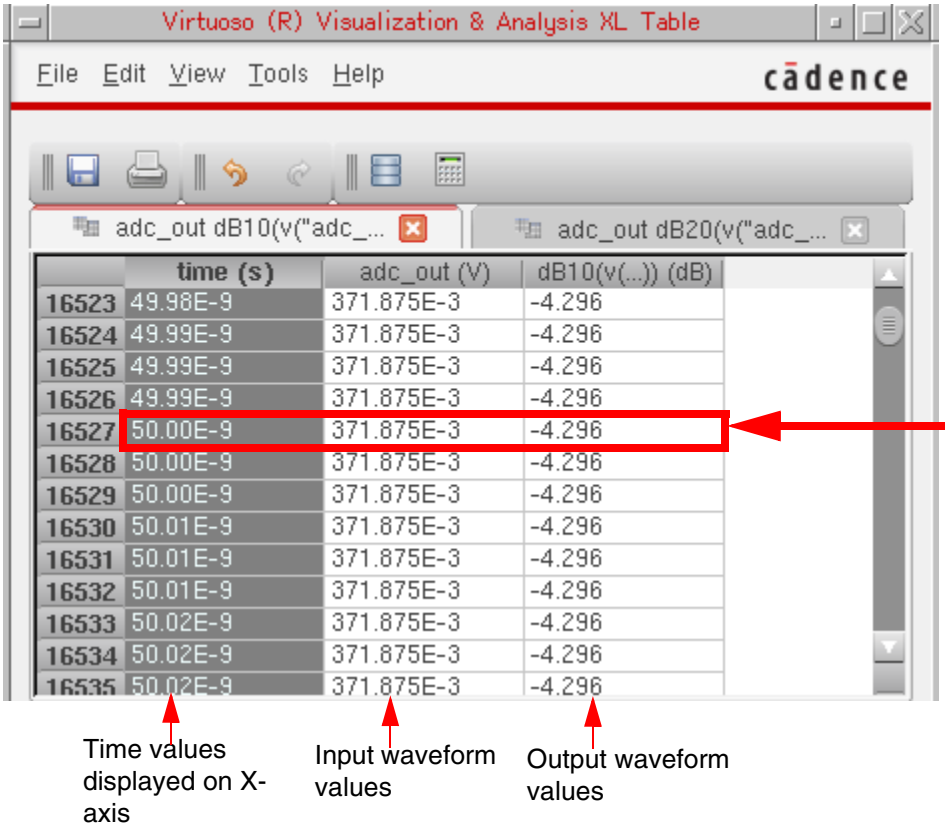


A point marker `V2` is placed at `time=50ns` (where the marker `V1` is placed in the input waveform) and after dB10 calculation it displays voltage as `-4.296dB` on the Y-axis.

To analyze these generated output values, you can display the input and output signals in the Virtuoso Visualization and Analysis XL Table. To send the input signal to the table, right-click the signal and choose *Send To – Table – New Window*. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=50ns, the voltage value in input waveform is 371.875E-3 (as shown in the above input waveform figure) and the voltage value in output waveform (after dB10 function is applied) is -4.296, as shown in the above output waveform figure.

Related OCEAN Function

The equivalent OCEAN command for `dB10` is:

```
db10( {o_waveform | n_number} )
     => o_waveform/n_number/nil
```

For more information, see `dB10` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

dB20

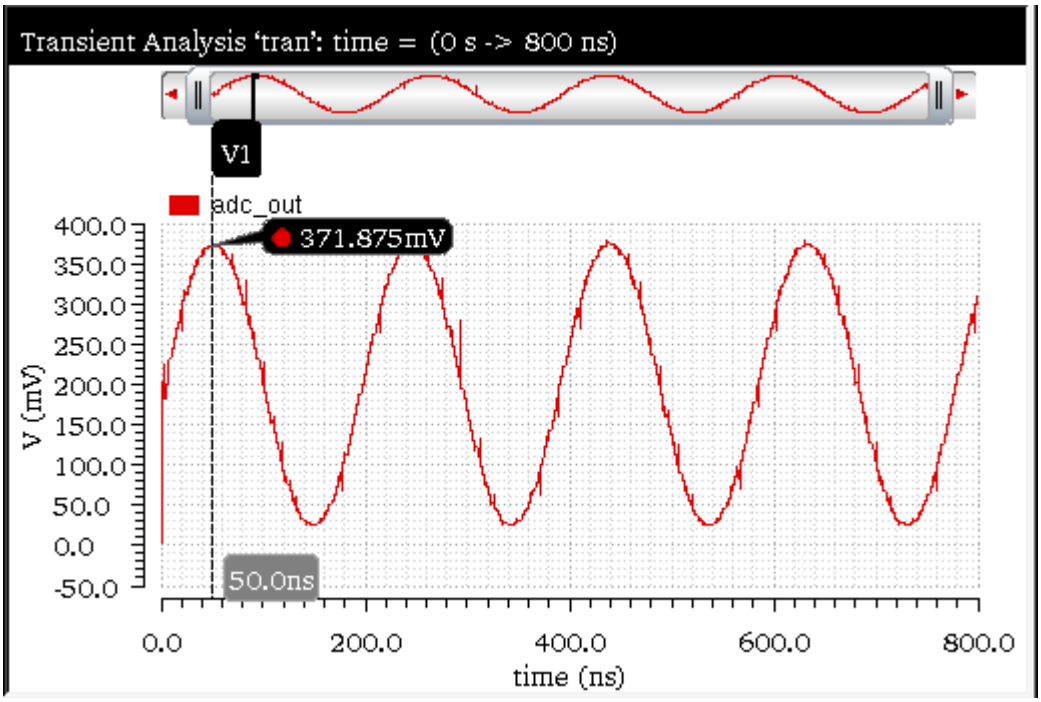
Converts a signal to dB by using the following equation:

```
dB20(x)= 20/log(10)*log(mag(x))
```

where, `x` is any number, waveform, or family of waveforms

Example

Consider the following input signal waveform from the transient analysis:



A vertical marker `V1` is placed at `time=50ns` that displays the voltage value on Y-axis as `371.875mv`.

When you send this signal to Calculator, the following expression is displayed in the Buffer:

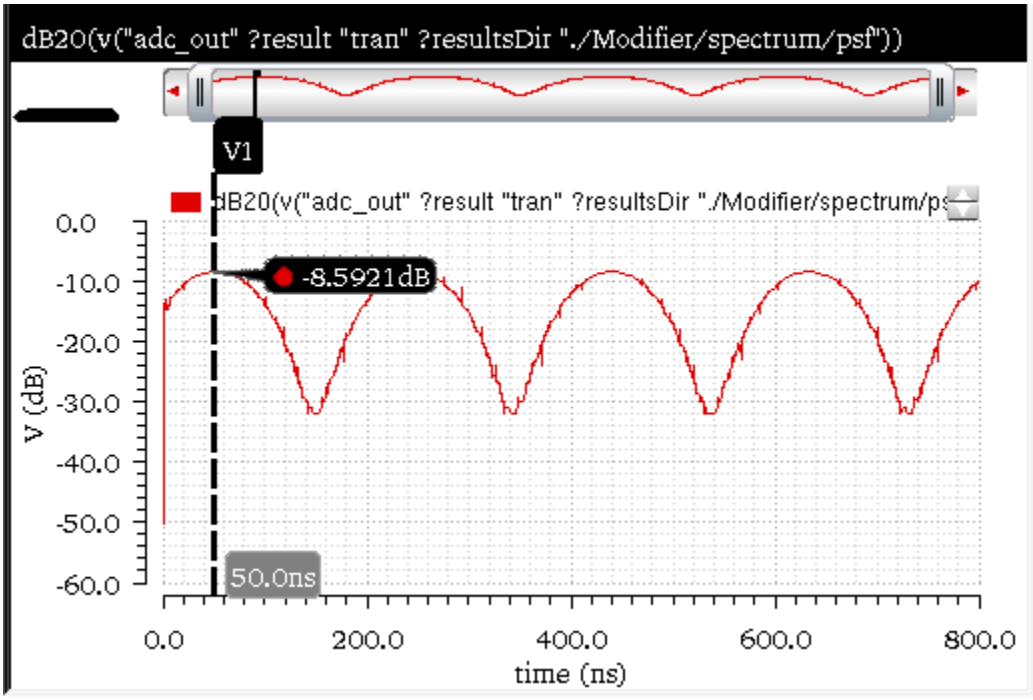
```
v("adc_out" ?result "tran" ?resultsDir "./Modifier/spectrum/psf")
```

When you select the `dB20` function from the Function Panel, the function is applied on the expression in the Buffer. The expression created in the Buffer is as follows:

```
dB20(v("adc_out" ?result "tran" ?resultsDir "./Modifier/spectrum/
psf"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following waveform is displayed in a new graph window that represents the dB20 calculation of the input waveform.

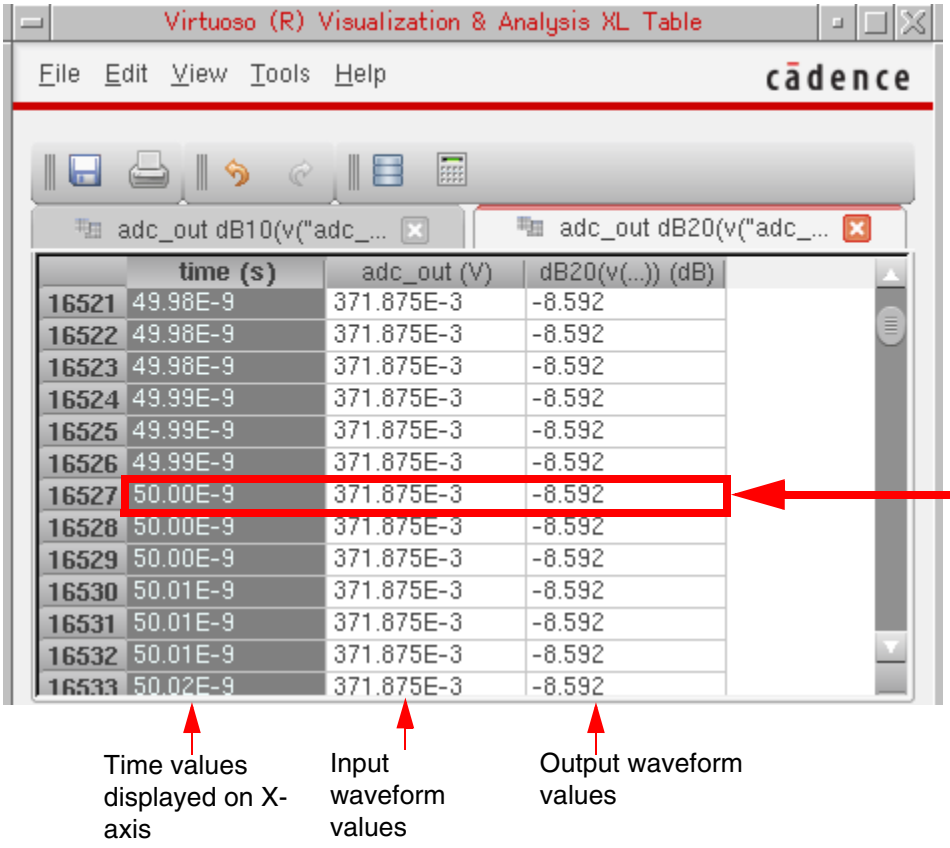


A point marker `M2` is placed at `time=50ns` (where the marker `M1` is placed in the input waveform) and after dB20 calculation it displays voltage as `-8.5921dB` on the Y-axis.

To analyze these generated output values, you can display the input and output signals in the Virtuoso Visualization and Analysis XL Table. To send the input signal to the table, right-click the signal and choose *Send To – Table – New Window*. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=50ns, the voltage value in input waveform is 371.875E-3 (as shown in the above input waveform figure) and the voltage value in output waveform (after dB20 function is applied) is -8.592, as shown in the above output waveform figure.

Related OCEAN Function

The equivalent OCEAN command for `dB20` is:

```
db20( {o_waveform | n_number} )
     => o_waveform/n_number/nil
```

For more information, see `dB20` in *OCEAN Reference*.

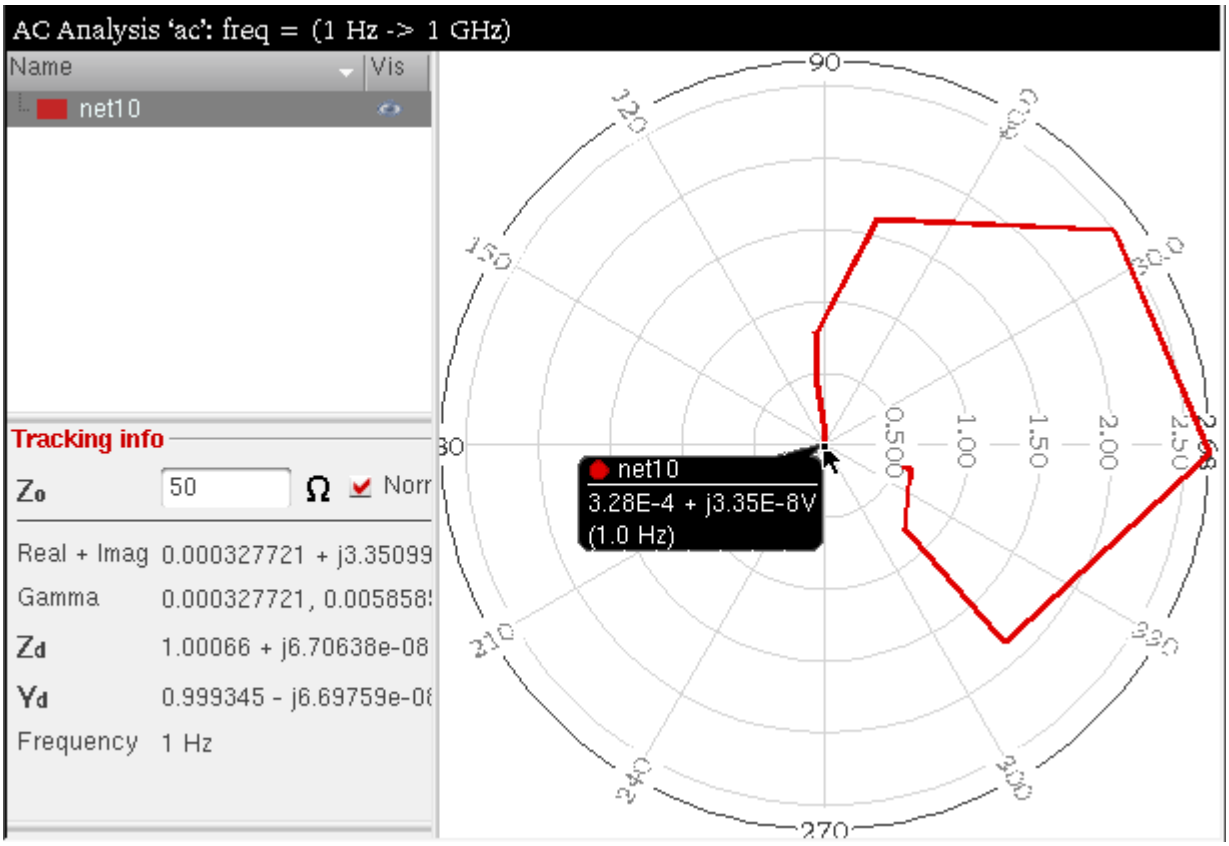
© 2023 Cadence Design Systems, Inc.

imag

Returns the imaginary component of the input waveform. This function is available only in the SKILL mode.

Example

Consider the following signal expression from the AC analysis for the complex data:



When you send this signal to Calculator, the following expression is displayed in the Buffer:

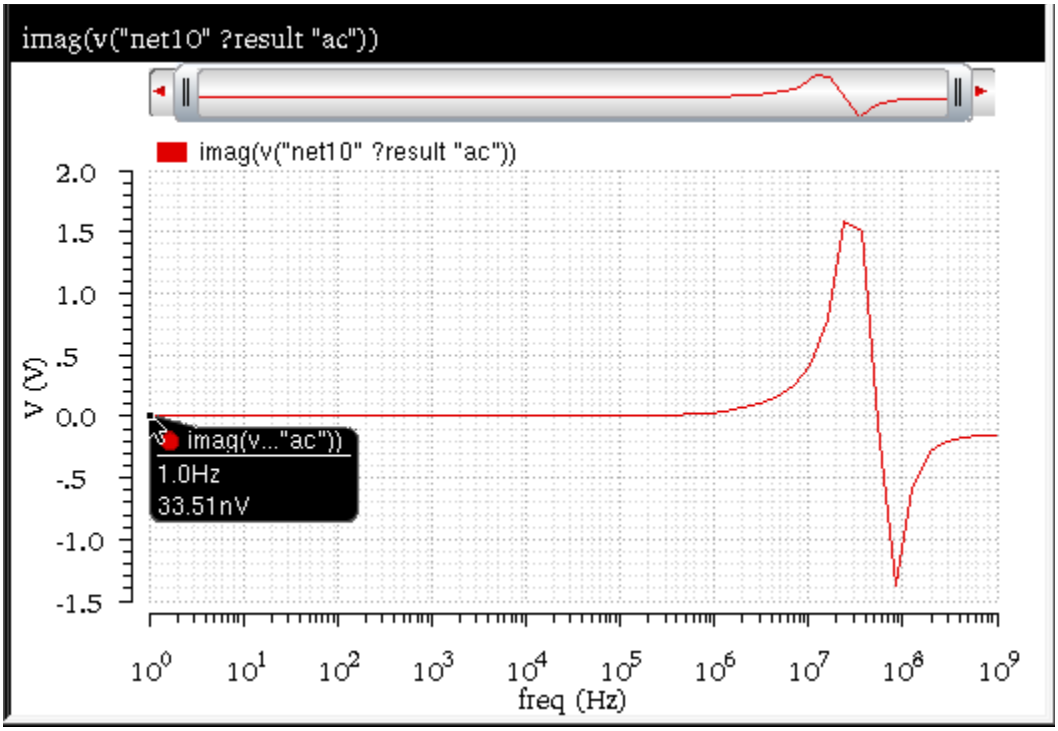
```
mag(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```

By default, the expression is displayed with the `mag` modifier function. You need to remove this `mag` function from the Buffer expression manually. When you select the `imag` function from the Function Panel, the function is applied on the signal expression in the Buffer. The new expression created in the Buffer is as follows:

```
imag(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following waveform is displayed in a new graph window that represents the conjugate of the input waveform.

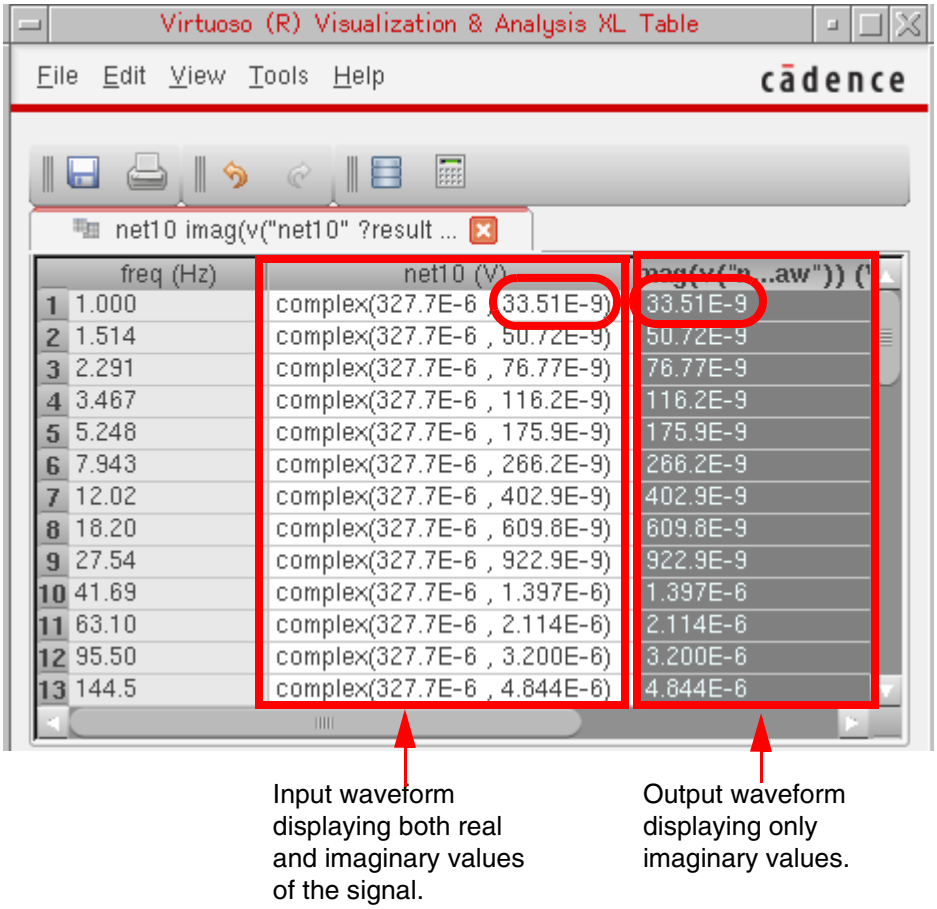


To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

To view both the real and imaginary values in the table, right-click the table row and choose *Display Complex As – Real and Imaginary*. The following table contents are displayed

© 2023 Cadence Design Systems, Inc.

when you send the input and output signals described in this example to the table and display the values in the complex number format.



Related OCEAN Function

The equivalent OCEAN command for `imag` is:

```
imag( {o_waveform | n_input} )
     => o_waveformImag/n_numberImag/nil
```

For more information, see `imag` in *OCEAN Reference*.

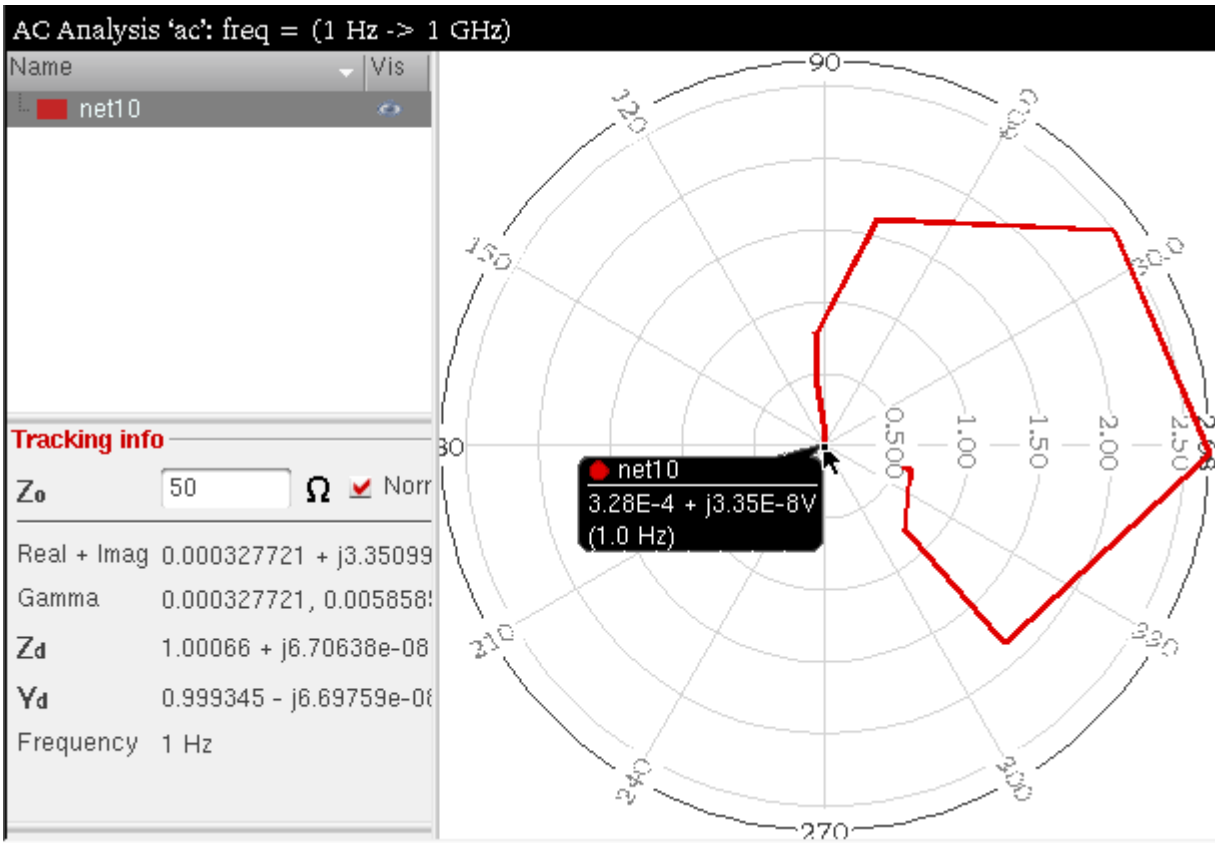
© 2023 Cadence Design Systems, Inc.

real

Returns the real component of the input signal.

Example

Consider the following signal expression from the AC analysis for the complex data:



When you send this signal to Calculator, the following expression is displayed in the Buffer:

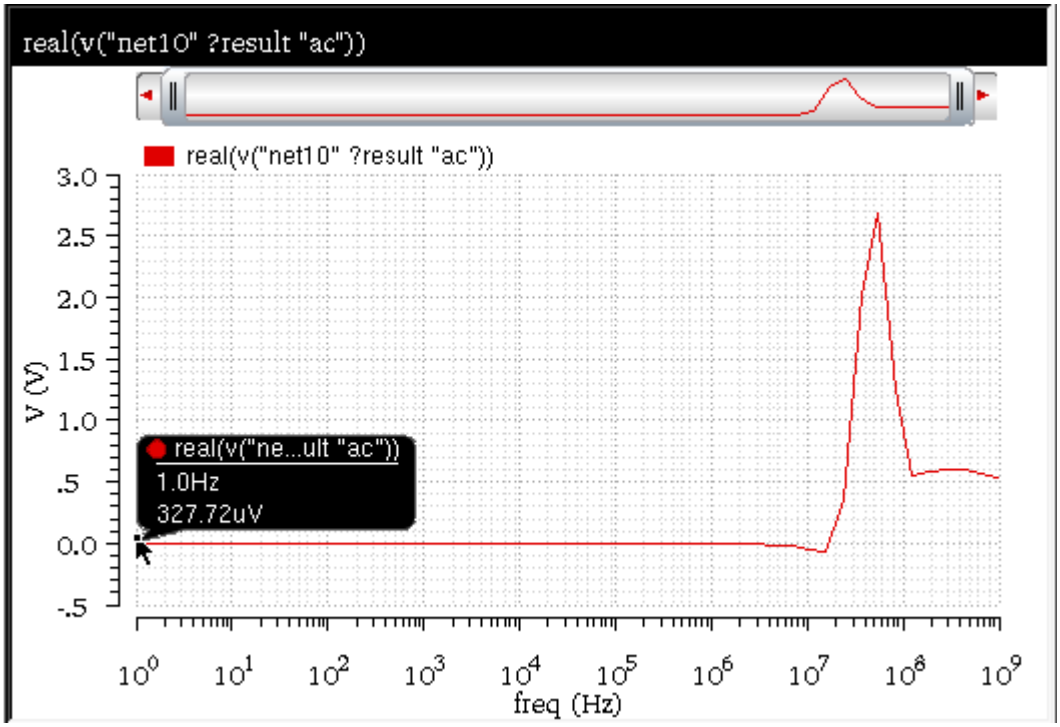
```
mag(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```

By default, the expression is displayed with the `mag` modifier function. You need to remove this `mag` function from the Buffer expression manually. When you select the `real` function from the Function Panel, the function is applied on the signal expression in the Buffer. The new expression created in the Buffer is as follows:

```
real(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following waveform is displayed in a new graph window that represents the conjugate of the input waveform.

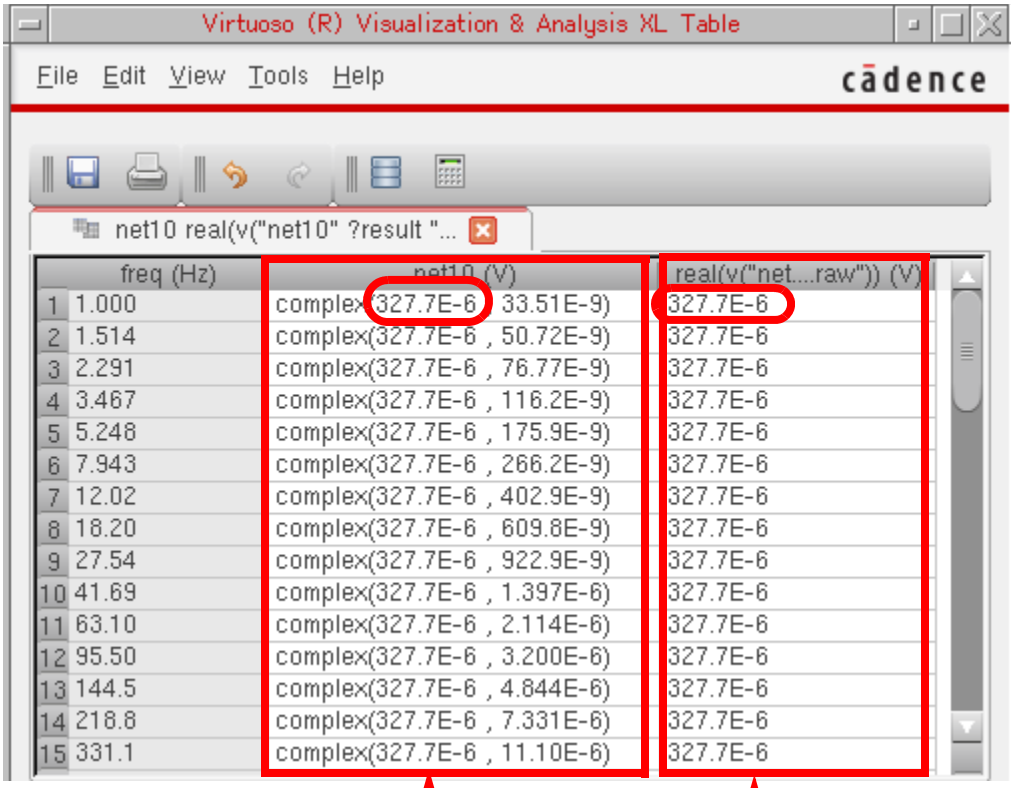


To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

To view both the real and imaginary values in the table, right-click the table row and choose *Display Complex As – Real and Imaginary*. The following table contents are displayed

© 2023 Cadence Design Systems, Inc.

when you send the input and output signals described in this example to the table and display the values in the complex number format.



Related OCEAN Function

The equivalent OCEAN command for `real` is:

```
real( {o_waveform | n_input} )
     => o_waveformReal/n_numberReal/nil
```

For more information, see `real` in *OCEAN Reference*.

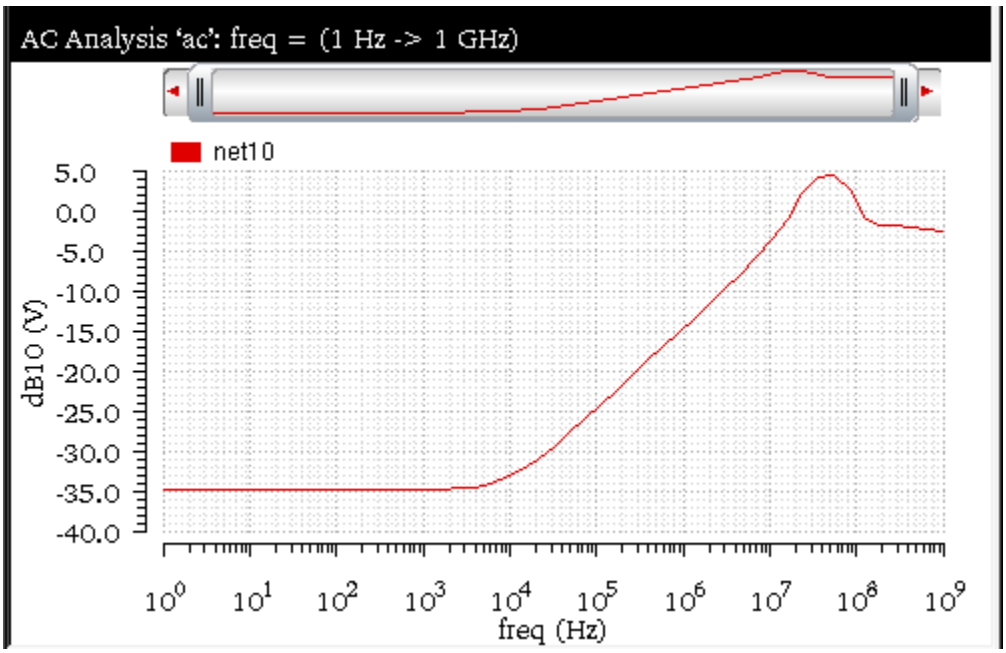
© 2023 Cadence Design Systems, Inc.

mag

Returns the magnitude of a signal.

Example

When you plot a signal from AC analysis, by default it displays the magnitude values of the signal on Y-axis. This function is basically used when you have another modifier function, such as `real`, `imag`, `dB10`, plotted on Y-axis and you want to view the magnitude of the output signal. For example, consider the following waveform from AC analysis in which the dB10 values are plotted on the Y-axis of the waveform.



When you send this signal to Calculator, the expression displayed in the Buffer is as follows:

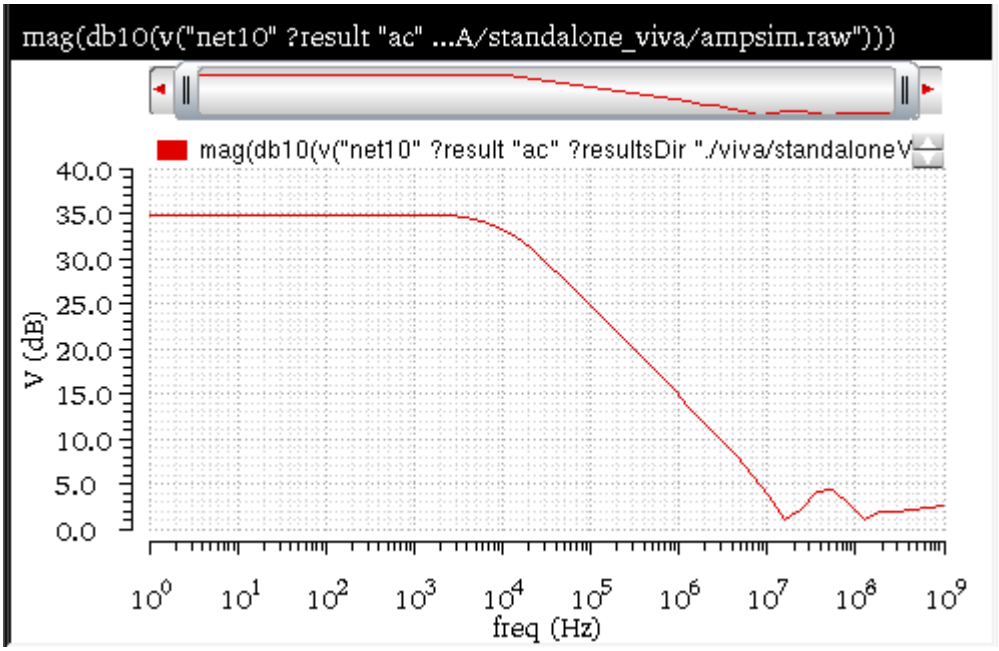
```
db10(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```

By default, the expression is displayed with the `dB10` (`db10`) function. When you select the `mag` function from the Function Panel, the function is applied on the signal expression in the Buffer. The new expression created in the Buffer is as follows:

```
mag(db10(v("net10" ?result "ac" ?resultsDir "./ampsim.raw")))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform showing the magnitude of the db10 of the voltage signal (`net10`) is displayed in a new graph window.



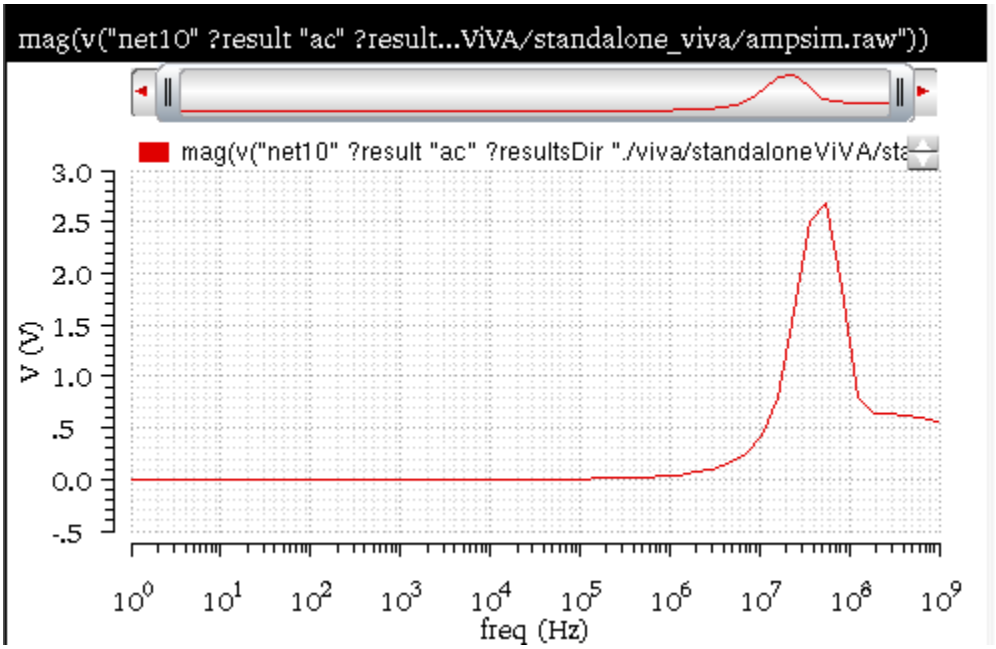
If you manually remove the `db10` function from the input signal and then apply the `mag` function, the following expression is created in the Buffer:

```
mag(v("net10" ?result "ac" ?resultsDir "./ampsim.raw"))
```

This expression calculates the magnitude of the voltage signal, `net10`, where as the previous expression calculated the magnitude of the waveform after the `dB10` modifier function is applied.

© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform appears in a new graph window.



Related OCEAN Function

The equivalent OCEAN command for `mag` is:

```
mag( {o_waveform | n_number} )
     => o_waveform/n_number/nil
```

For more information, see `mag` in *OCEAN Reference*.

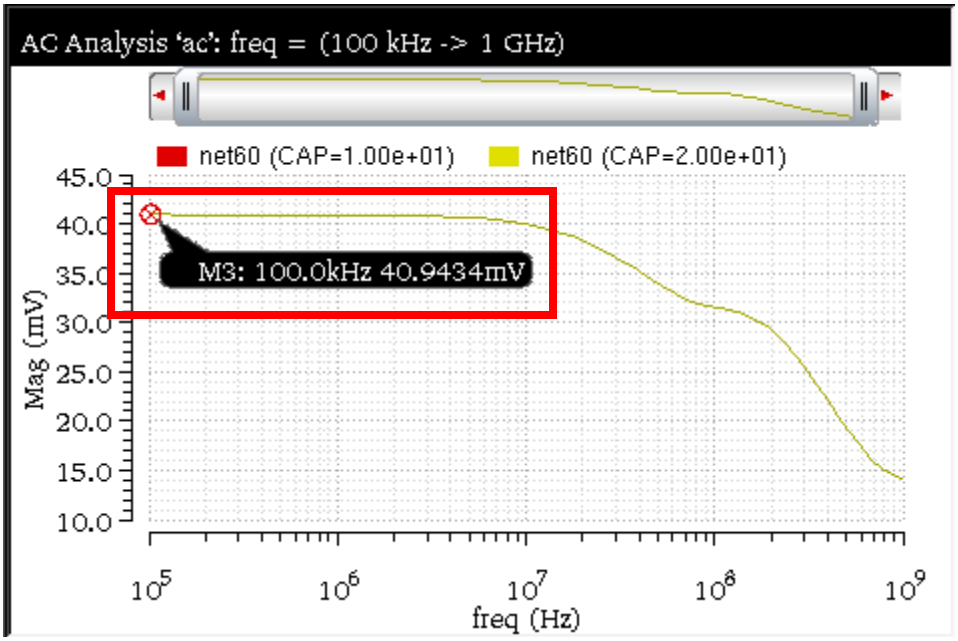
© 2023 Cadence Design Systems, Inc.

phase

Returns the phase of a signal in degrees.

Example

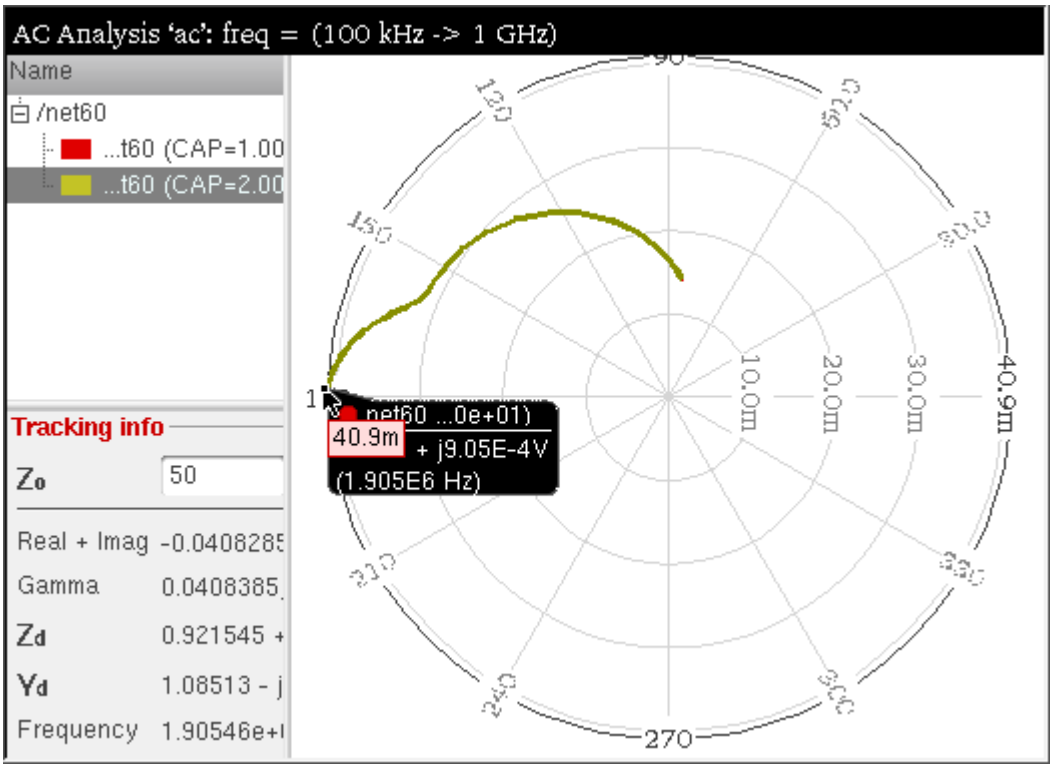
Consider the following input waveform from a AC analysis:



A point marker, `M3`, is placed at `freq=100KHz` and the `voltage=40.9434mV`.

© 2023 Cadence Design Systems, Inc.

The following graph appears when you plot this input signal in a circular graph (plor plot):



When you send this signal to Calculator, the following expression is displayed in the Buffer:

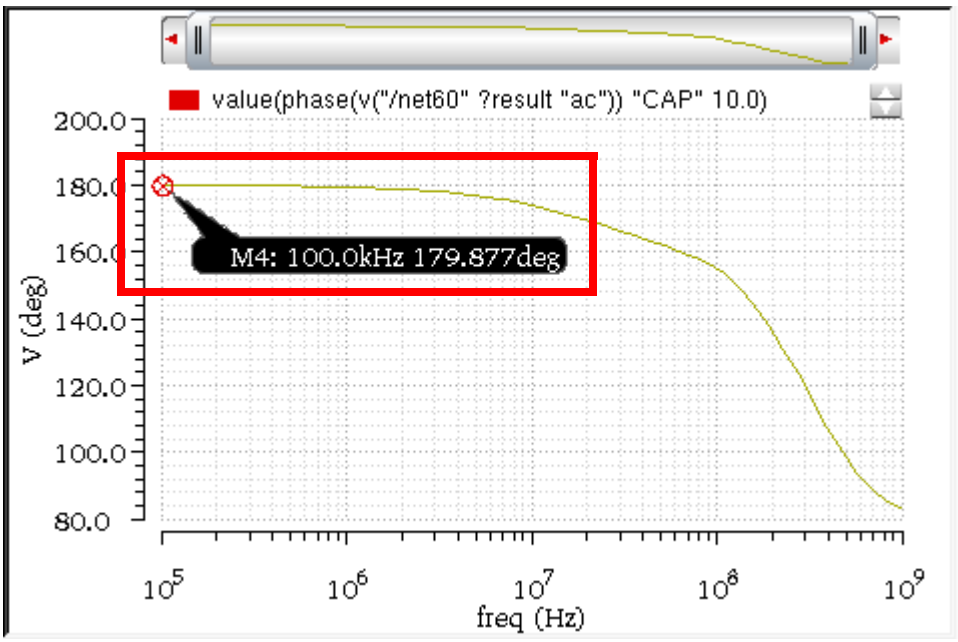
```
mag(v("net60" ?result "ac" ?resultsDir "./Modifier/peakTest/spectre/
schematic/psf"))
```

By default, the expression is displayed with the `mag` modifier function. You need to remove this `mag` function from the Buffer expression manually. When you select the `phase` function from the Function Panel, the function is applied on the signal expression in the Buffer. The new expression created in the Buffer is as follows:

```
phase(v("net60" ?result "ac" ?resultsDir "./Modifier/peakTest/
spectre/schematic/psf"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following waveform is displayed in a new graph window that represents the phase of the input waveform.

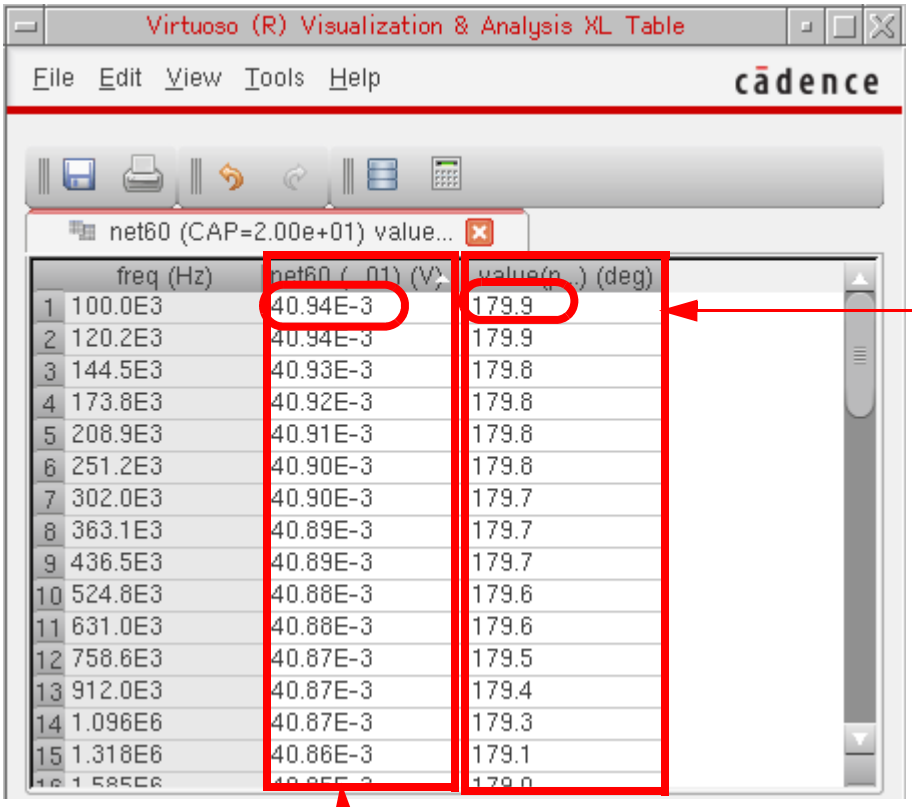


A point marker `M4` is placed at `freq=100.0KHz` (where the point marker `M3` was placed in the input waveform) and after phase calculation, it displays the voltage as `179.877` degrees.

To analyze these generated output voltage values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table.



Output waveform values

Input waveform values

At time=0.0ns, the voltage value in input waveform is 40.94E-3V(as shown in the above input waveform figure) and the voltage value in output waveform (after dB20 function is applied) is 179.9 deg, as shown in the above output waveform figure.

Related OCEAN Function

The equivalent OCEAN command for `phase` is:

```
phase( {o_waveform | n_number} )
     => o_waveform/n_number/nil
```

For more information, see phase in *OCEAN Reference*.

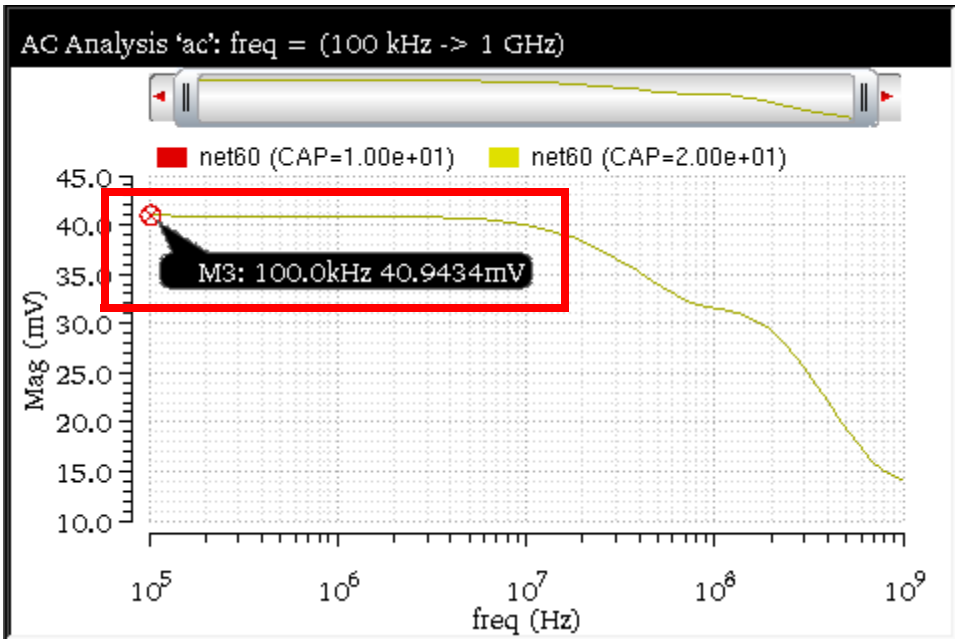
© 2023 Cadence Design Systems, Inc.

phaseRad

Calculates the wrapped (discontinuous) phase in radians of a waveform.

Example

Consider the following input waveform from a AC analysis:



A point marker, `M3`, is placed at `freq=100KHz` and `voltage=40.9434mV`.

When you send this signal to Calculator, the following expression is displayed in the Buffer:

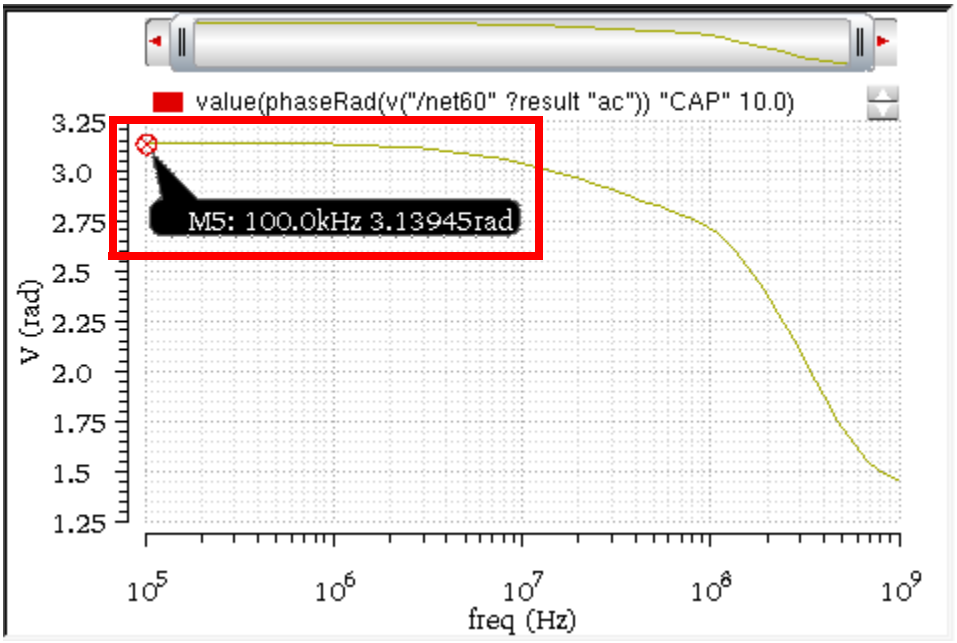
```
mag(v("net60" ?result "ac" ?resultsDir "./Modifier/peakTest/spectre/
schematic/psf"))
```

By default, the expression is displayed with the `mag` modifier function. You need to remove this `mag` function from the Buffer expression manually. When you select the `phaseRad` function from the Function Panel, the function is applied on the signal expression in the Buffer. The new expression created in the Buffer is as follows:

```
phaseRad(v("net60" ?result "ac" ?resultsDir "./Modifier/peakTest/
spectre/schematic/psf"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following waveform is displayed in a new graph window that represents the phase (in radians) of the input waveform.

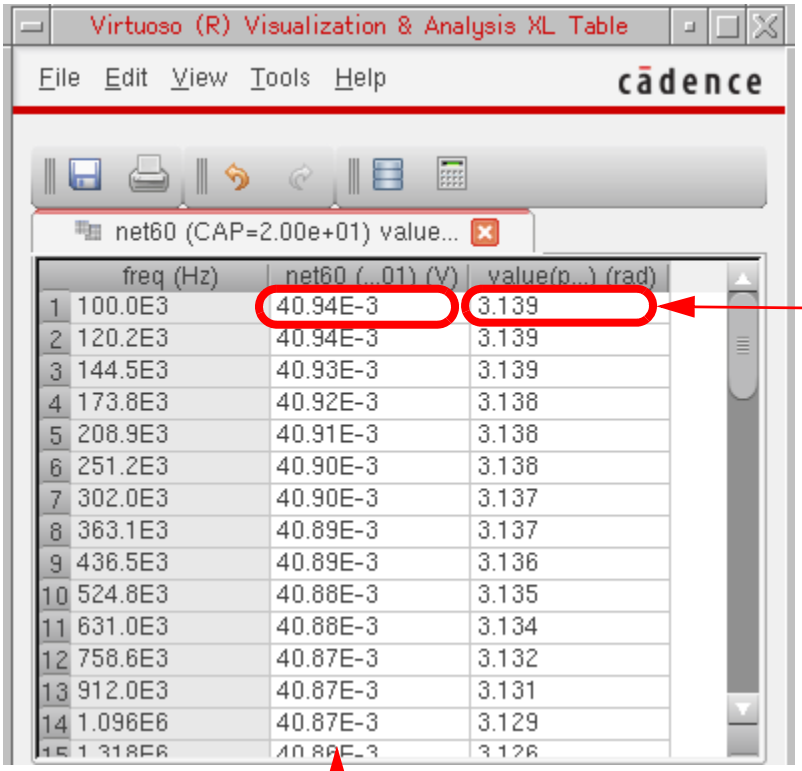


A point marker `M5` is placed at `freq=100.0KHz` (where the point marker `M3` was placed in the input waveform) and after phase calculation, it displays the voltage as `3.13945` degrees.

To analyze these generated output voltage values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table.



Output waveform values

Input waveform

At time=0.0ns, the voltage value in input waveform is 40.94E-3V(as shown in the above input waveform figure) and the voltage value in output waveform (after dB20 function is applied) is 3.139 rad, as shown in the above output waveform figure.

Related OCEAN Function

The equivalent OCEAN command for `phase` is:

```
phaseRad( {o_waveform | n_number} )
     => o_waveform/n_number/nil
```

For more information, see `phase` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

phaseDeg

Calculates the wrapped phase in degrees of a waveform and returns a waveform.

Example: `phaseDeg(v("net9" ?result "tran"))`

© 2023 Cadence Design Systems, Inc.

phaseDegUnwrapped

Calculates the unwrapped phase in degrees of a waveform and returns a waveform.

Example: `phaseDegUnwrapped(v("net9" ?result "tran"))`

© 2023 Cadence Design Systems, Inc.

phaseRadUnwrapped

Calculates the unwrapped (continuous) phase in radians of a waveform and returns a waveform.

Example: `phaseRadUnwrapped(v("net9" ?result "tran"))`

Takes the input waveform, representing the voltage of `net9`, and returns the waveform object representing the unwrapped phase in degrees.

© 2023 Cadence Design Systems, Inc.

Trigonometric Functions

This section covers the following Trigonometeric functions in Calculator:

- cos
- acos
- cosh
- acosh
- sin
- asin
- sinh
- asinh
- tan
- atan
- tanh
- atanh

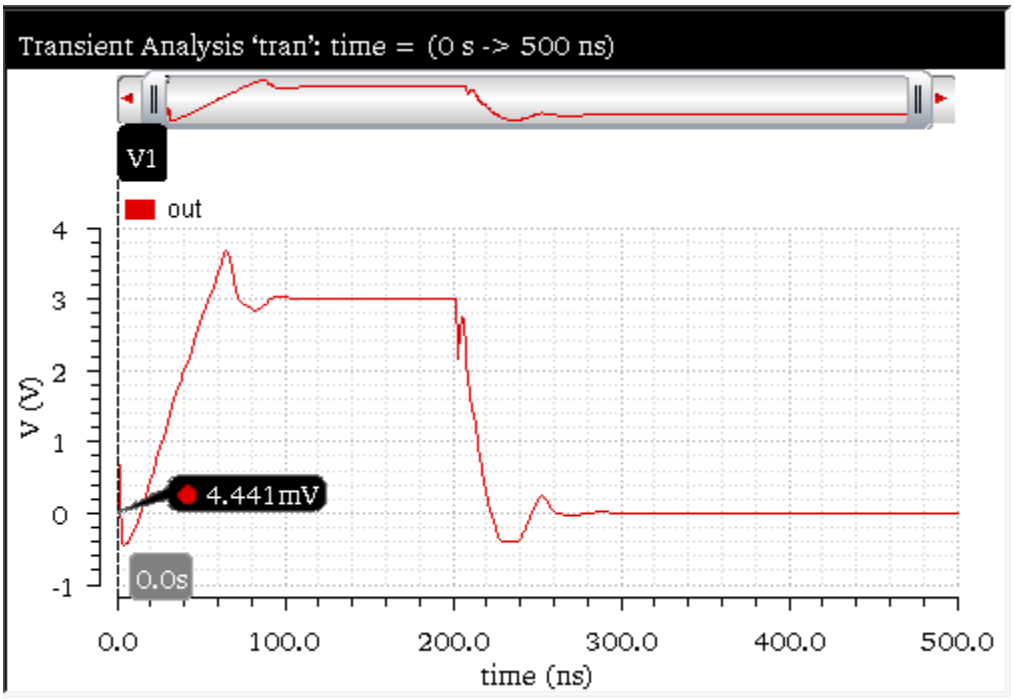
© 2023 Cadence Design Systems, Inc.

cos

Returns the cosine of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

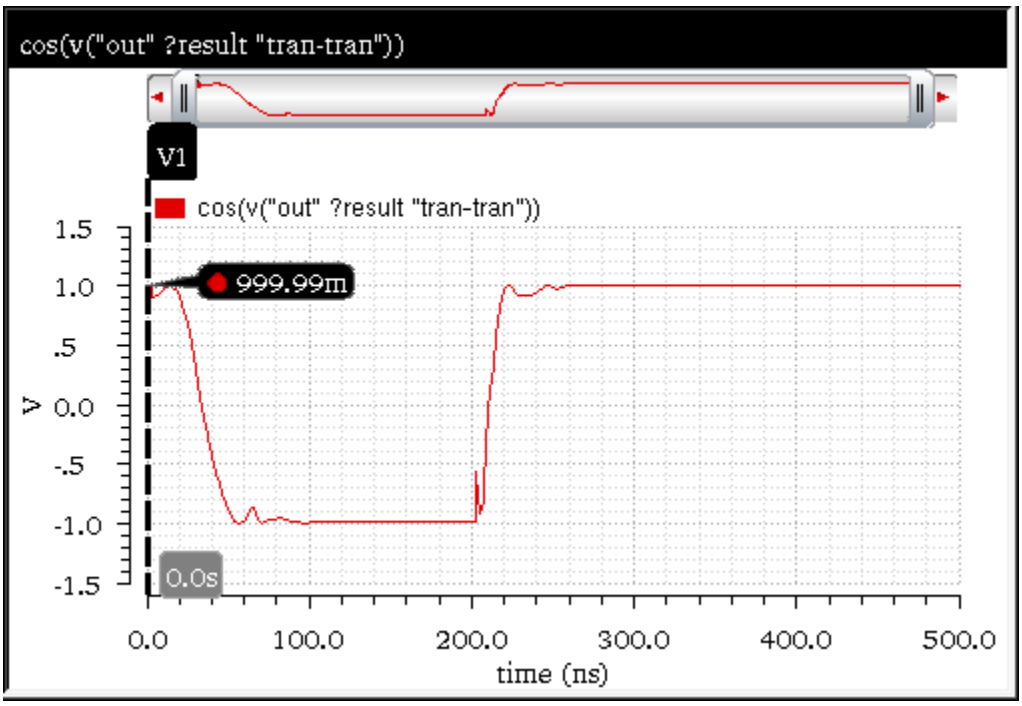
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `cos` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
cos(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the integer voltage values of the input waveform. The following output waveform is displayed in a new window:

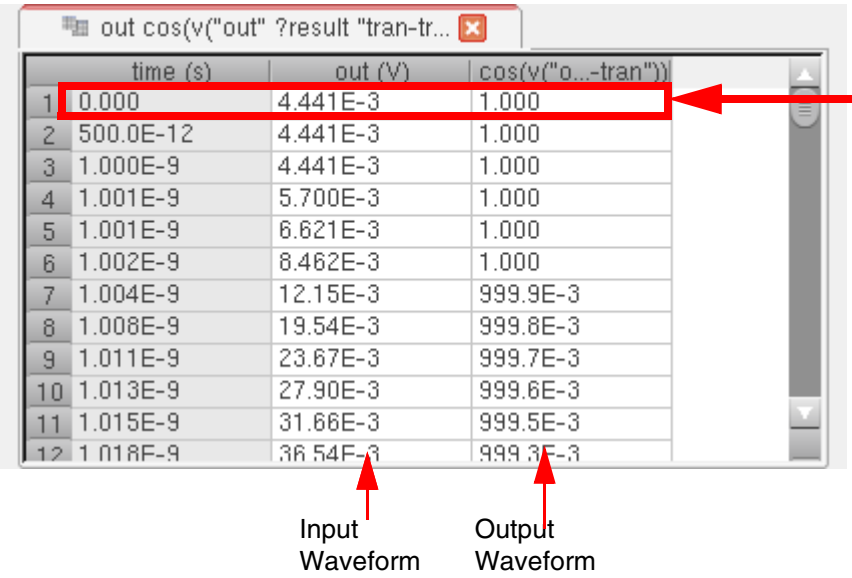


In the output waveform, the vertical maker `V1` shows the voltage=`999.99`mV when placed at time=`0.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.0ns, the value of voltage in the input signal1, out(V) is 4.441E-3. After the cos function is applied, the resultant voltage value at time=0.0ns in output signal is 1.000V. This is the cosine value of 4.441E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

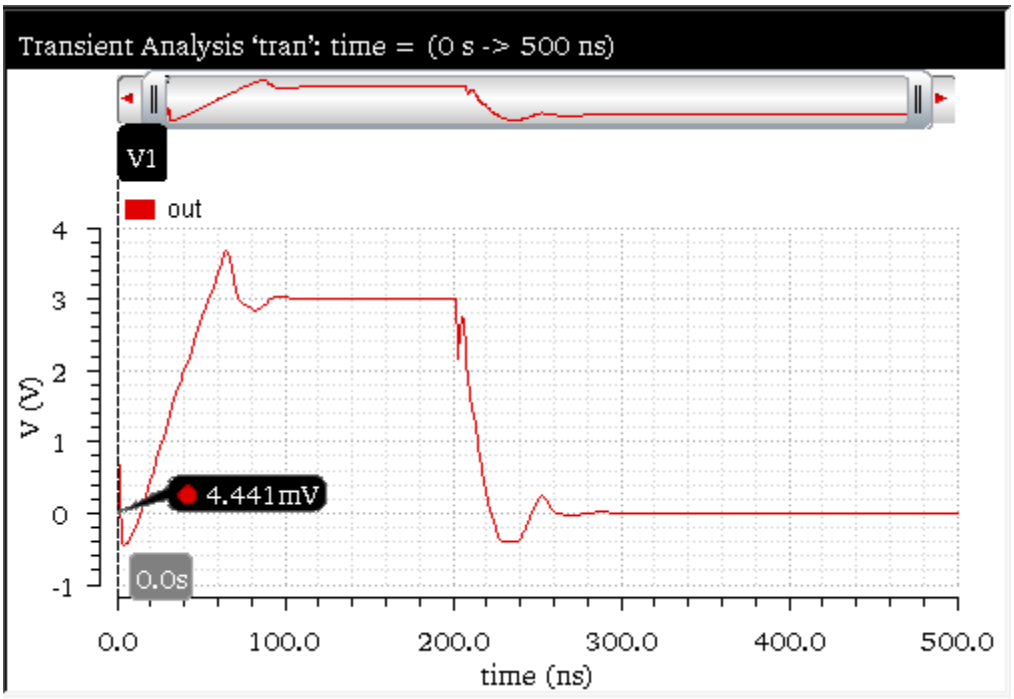
© 2023 Cadence Design Systems, Inc.

acos

Returns the inverse cosine of a ignal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

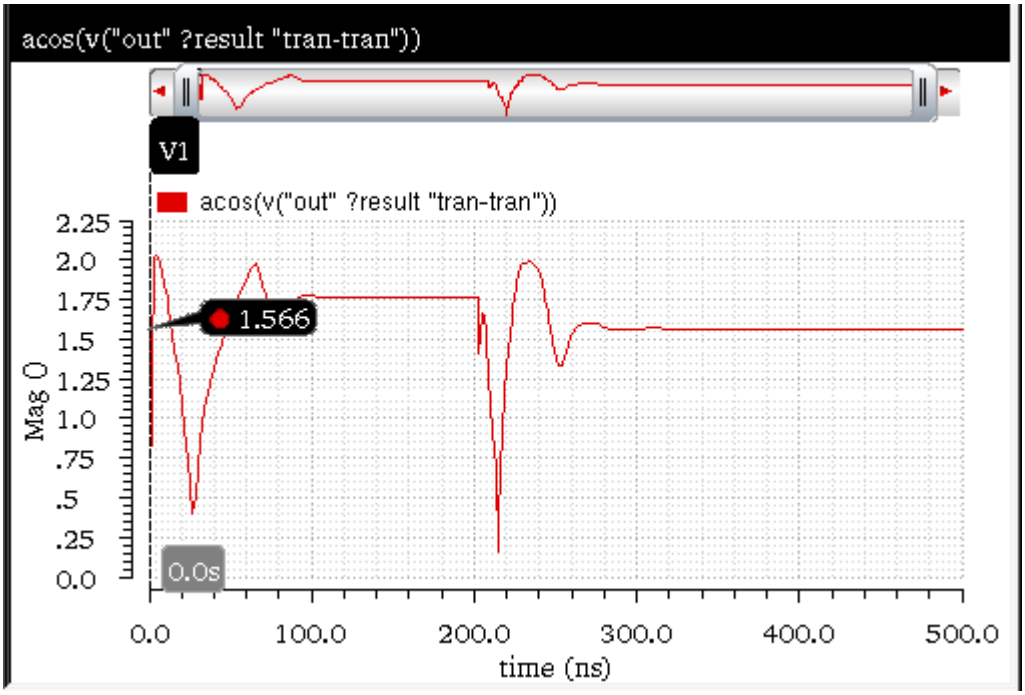
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `acos` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
acos(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the inverse cosine values of the input waveform. The following output waveform is displayed in a new window:

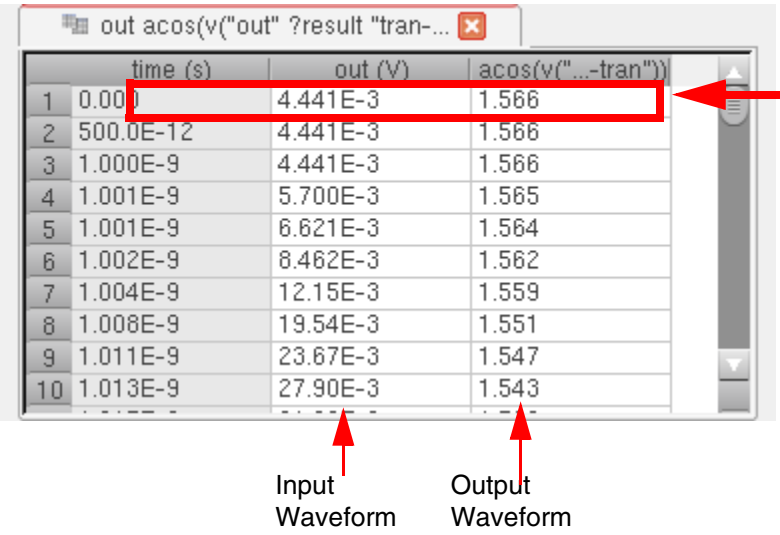


In the output waveform, the vertical maker `V1` shows the voltage=1.566 when placed at time=`0.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.0ns, the value of voltage in the input signal, out(V), is 4.441E-3. After the acos function is applied, the resultant voltage value at time=0.0ns in output signal is 1.566V. This is the inverse cosine value of 4.441E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

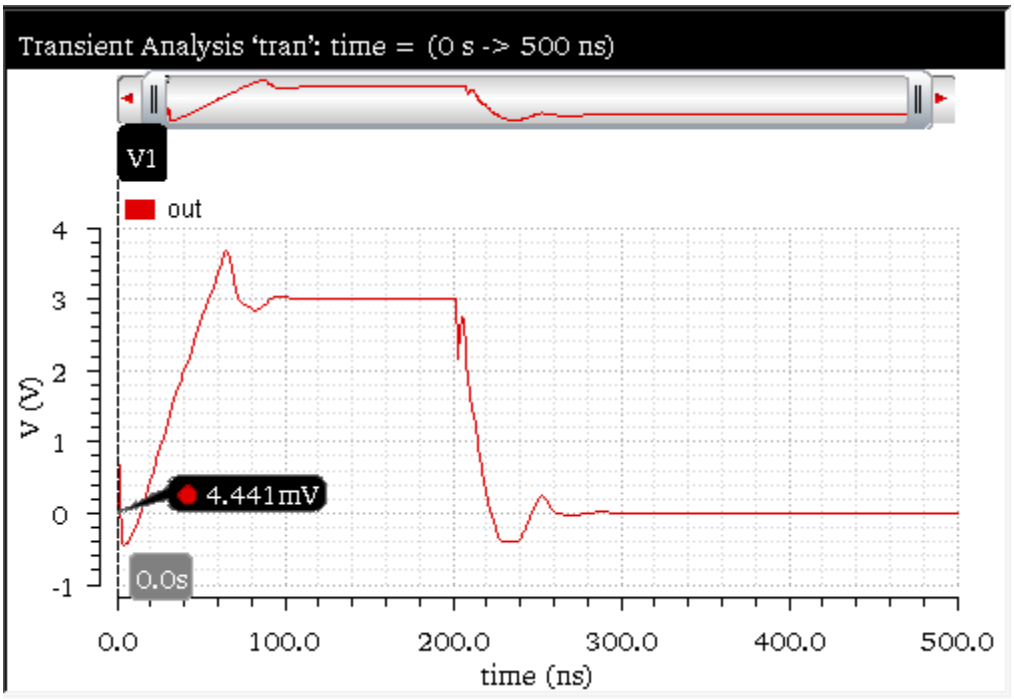
© 2023 Cadence Design Systems, Inc.

cosh

Returns the hyperbolic cosine of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

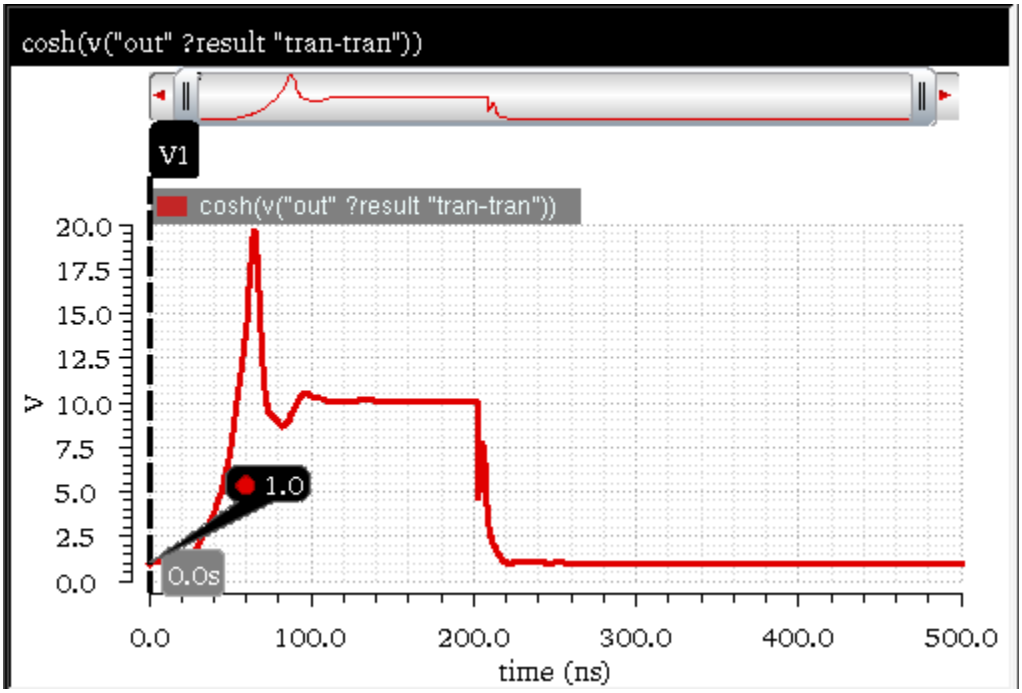
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `cosh` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
cosh(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the hyperbolic cosine values of the input waveform. The following output waveform is displayed in a new window:

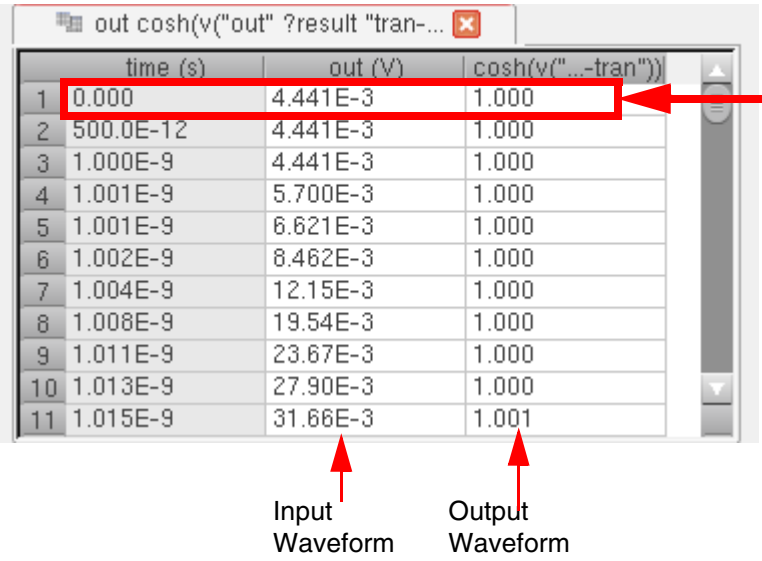


In the output waveform, the vertical maker `V1` shows the voltage=`1.0V` when placed at time=`0.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.0ns, the value of voltage in the input signal, out(V), is 4.441E-3. After the cosh function is applied, the resultant voltage value at time=0.0ns in output signal is 1.0V. This is the hyperbolic cosine value of 4.441E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

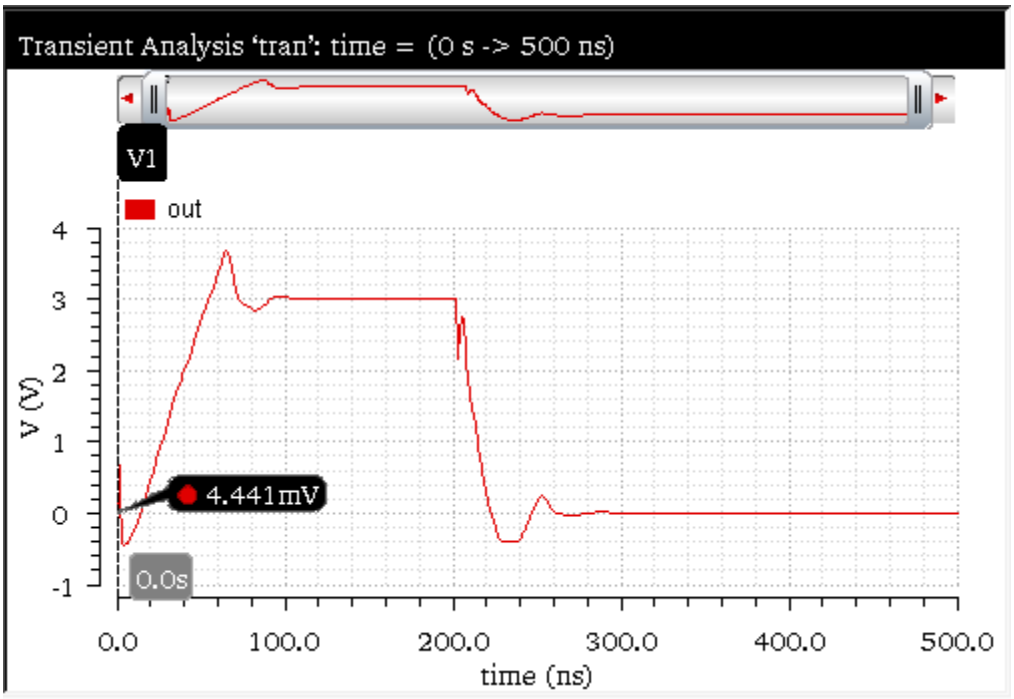
© 2023 Cadence Design Systems, Inc.

acosh

Returns the inverse hyperbolic cosine or hyperbolic arc-cosine of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`0.0s` and voltage=`4.441mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

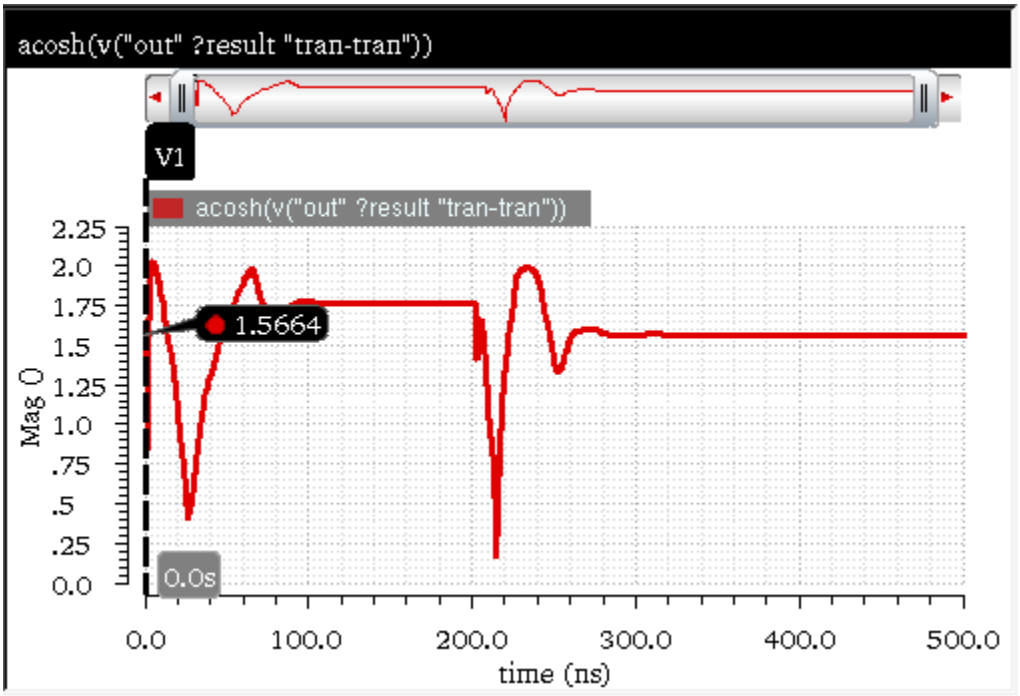
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `acosh` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
acosh(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the hyperbolic arc-cosine values of the input waveform. The following output waveform is displayed in a new window:



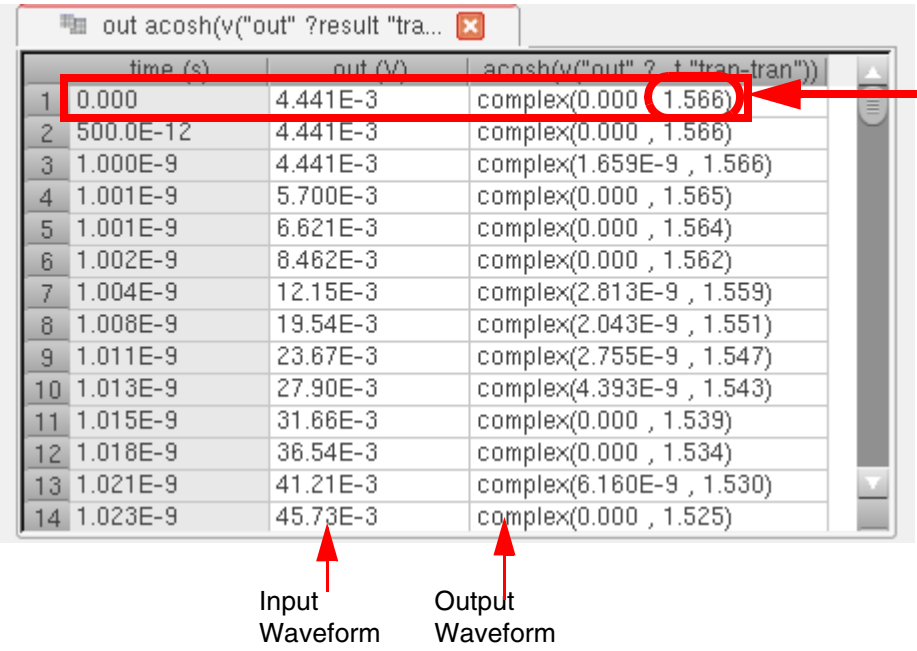
In the output waveform, the vertical maker `V1` shows the voltage=`1.5664V` when placed at time=`0.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

To view both the real and imaginary values in the table, right-click the table row and choose *Display Complex As – Real and Imaginary*. The following table contents are displayed when you send the input and output signals described in this example to the table and display the values in the complex number format.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.0ns, the value of voltage in the input signal, out(V), is 4.441E-3. After the acosh function is applied, the resultant voltage value at time=0.0ns in output signal is a complex number with real part=1.566V. This is the hyperbolic arc cosine value of 4.441E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

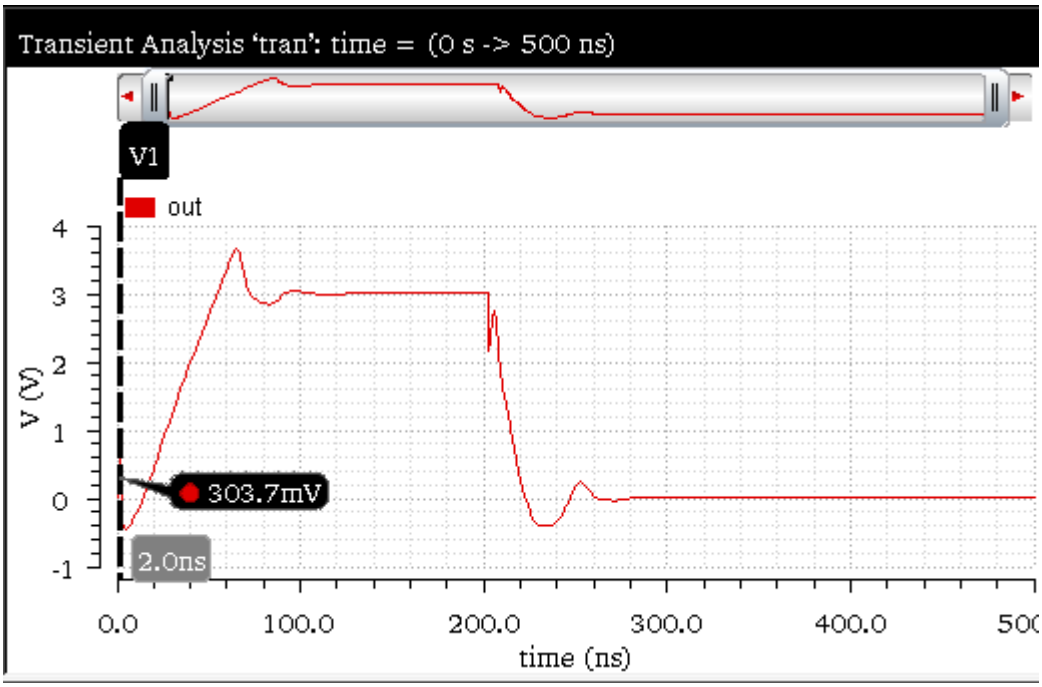
© 2023 Cadence Design Systems, Inc.

sin

Returns the sine of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

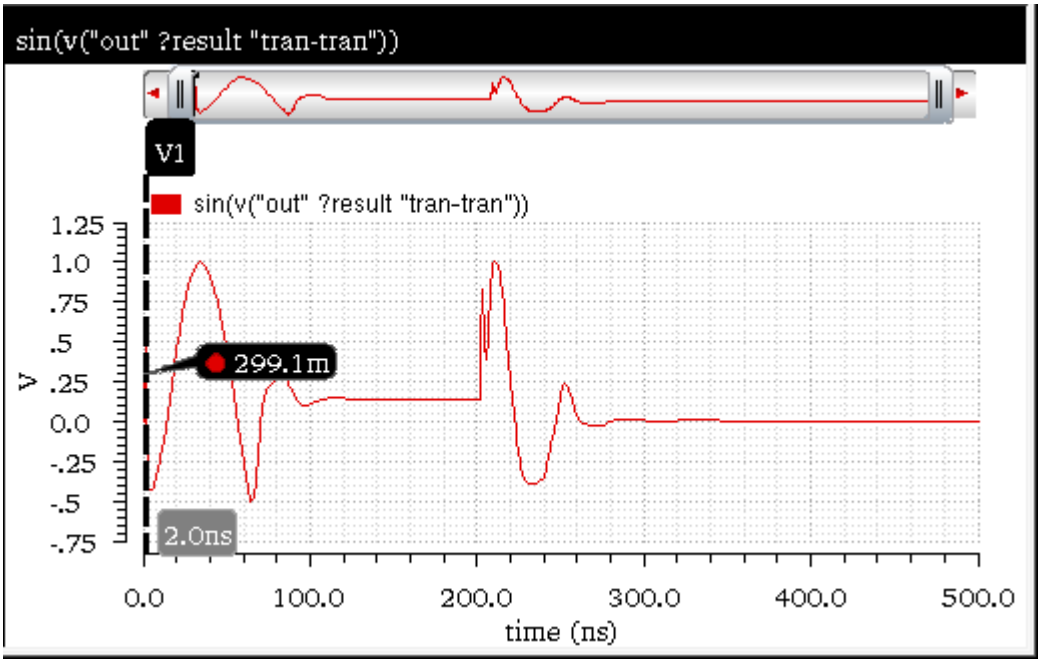
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `sin` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
sin(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the sine values of the input waveform. The following output waveform is displayed in a new window:

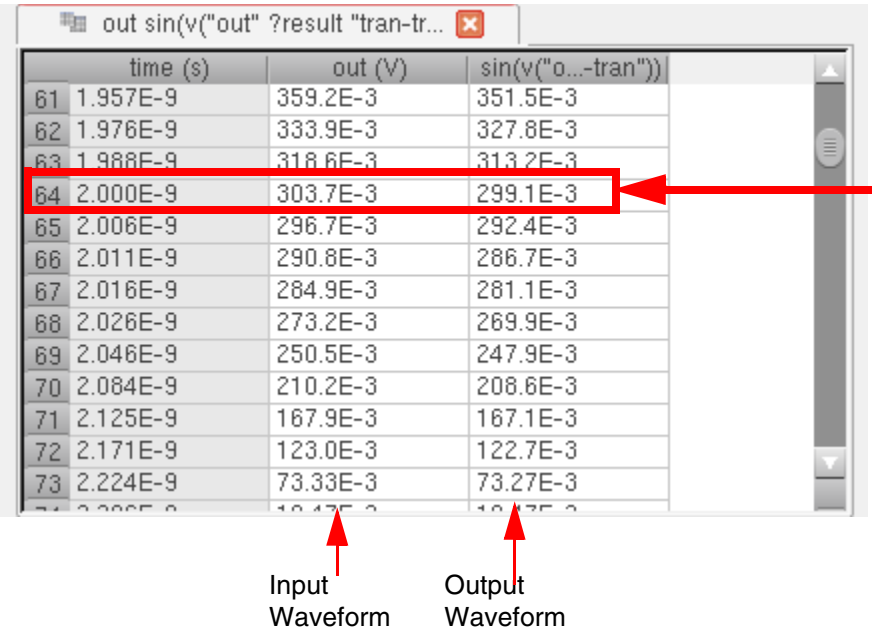


In the output waveform, the vertical maker `V1` shows the voltage=`299.1mV` when placed at time=`2.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the voltage value in the input signal1, out(V), is 303.7E-3V After the sin function is applied, the resulted voltage value at time=2.00E-9s in output signal is 299.1E-3V, which is the sine value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

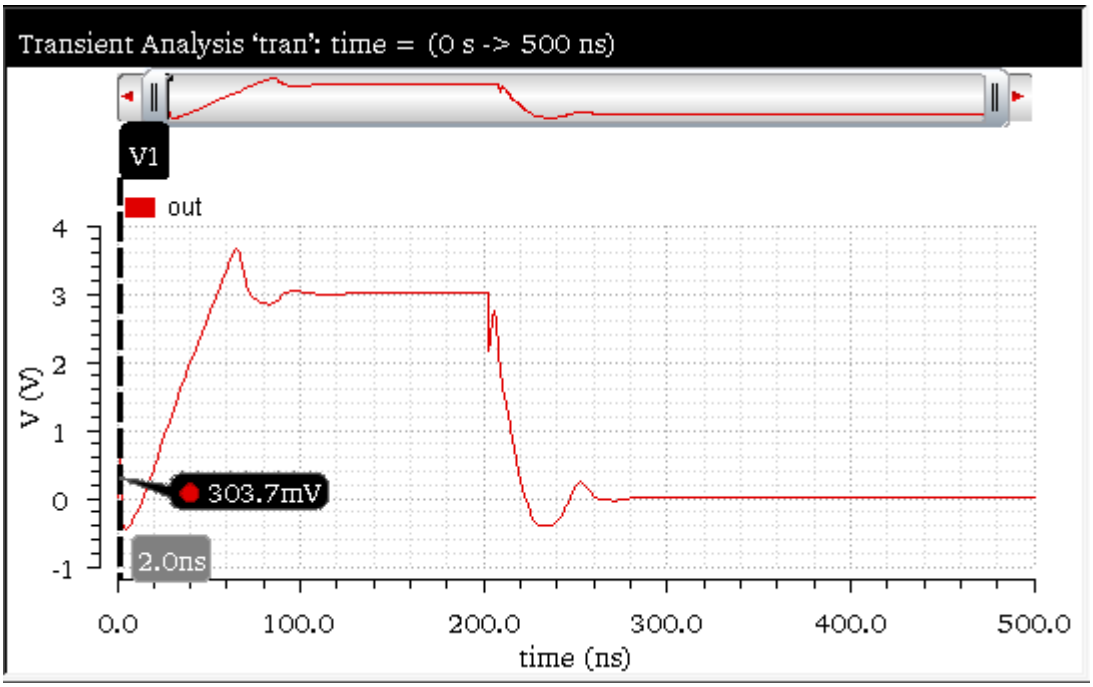
© 2023 Cadence Design Systems, Inc.

asin

Returns the inverse sine of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

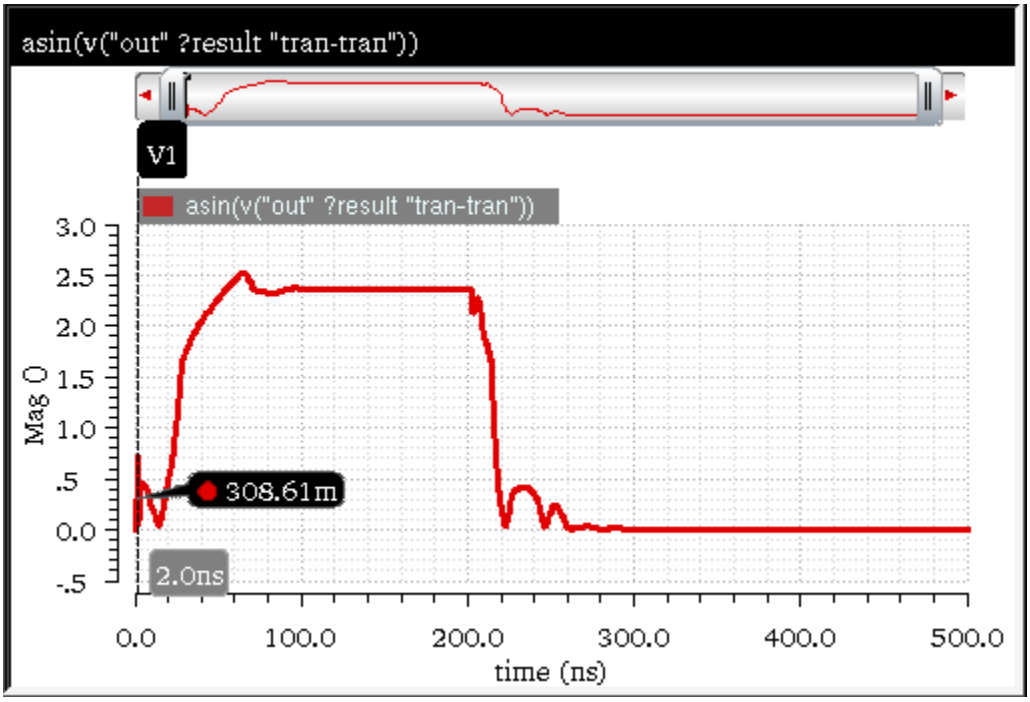
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `asin` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
asin(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the inverse sine values of the input waveform. The following output waveform is displayed in a new window:

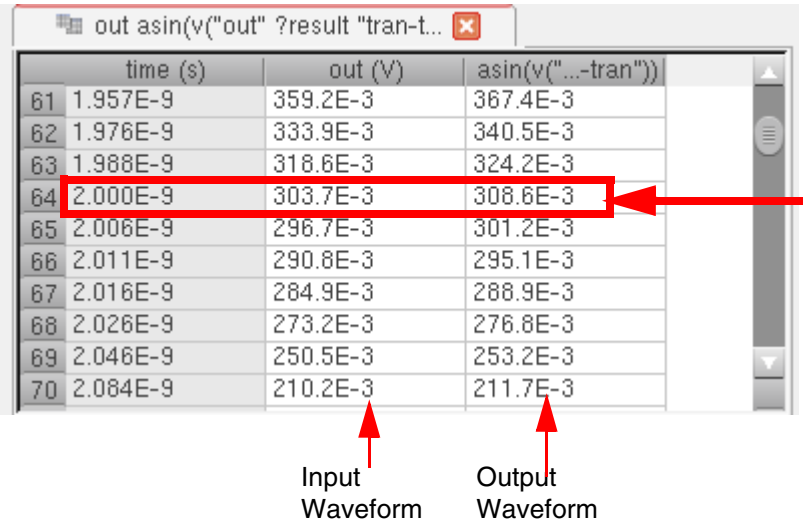


In the output waveform, the vertical maker `V1` shows the voltage=`308.61mV` when placed at time=`2.0ns`

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the value of voltage in the input signal, out(V), is 303.7E-3V After the asin function is applied, the resultant voltage value at time=2.00E-9s in output signal is 308.6E-3V. This is the inverse sine value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

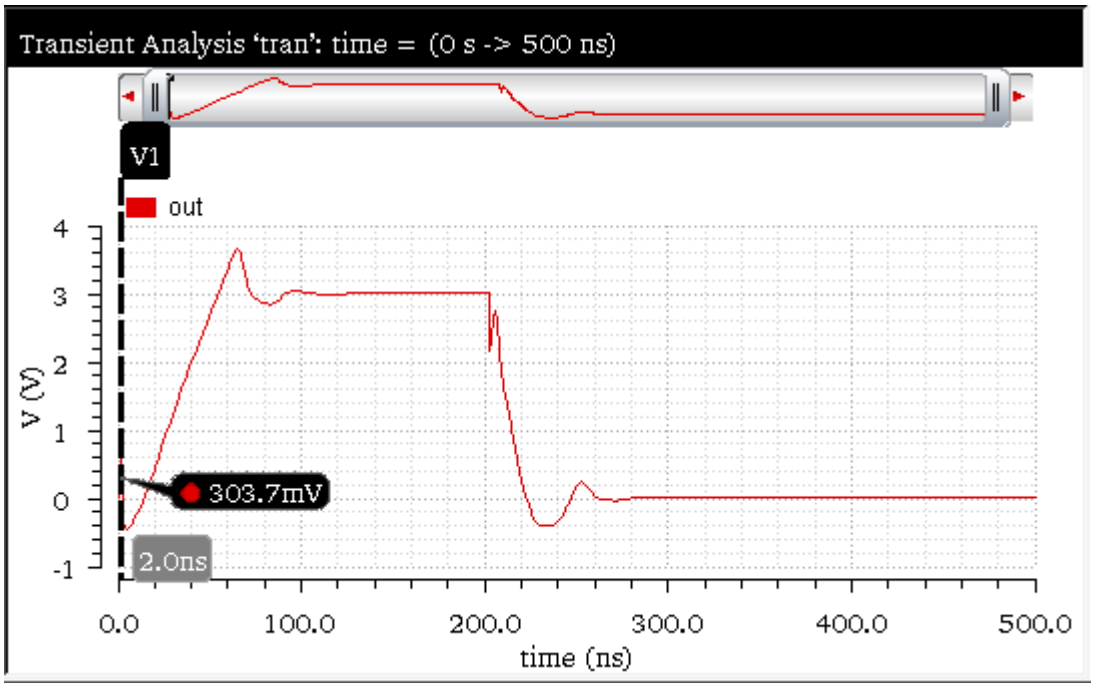
© 2023 Cadence Design Systems, Inc.

sinh

Returns the hyperbolic sine of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

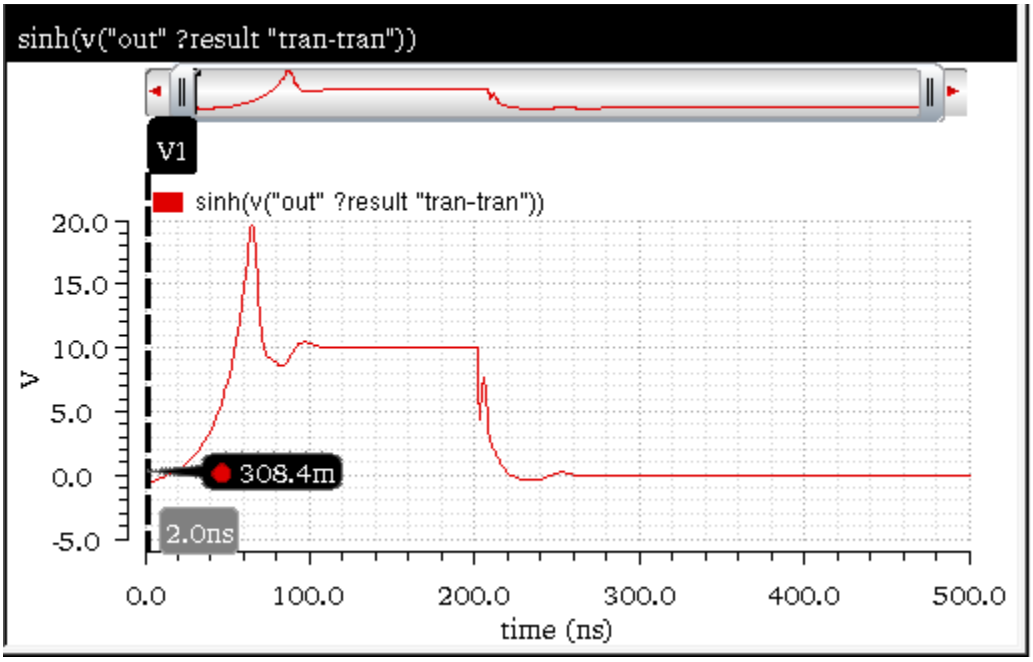
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `sinh` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
sinh(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the hyperbolic sine values of the input waveform. The following output waveform is displayed in a new window:

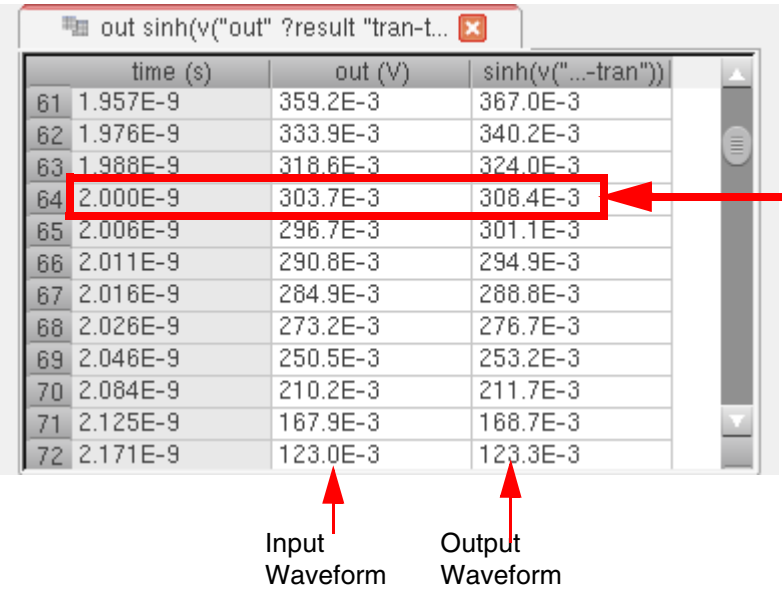


In the output waveform, the vertical maker `V1` shows the voltage=`308.4mV` when placed at time=`2.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the voltage value in the input signal1, out(V), is 303.7E-3V After the sinh function is applied, the resulted voltage value at time=2.00E-9s in output signal is 308.4E-3V, which is the hyperbolic sine value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

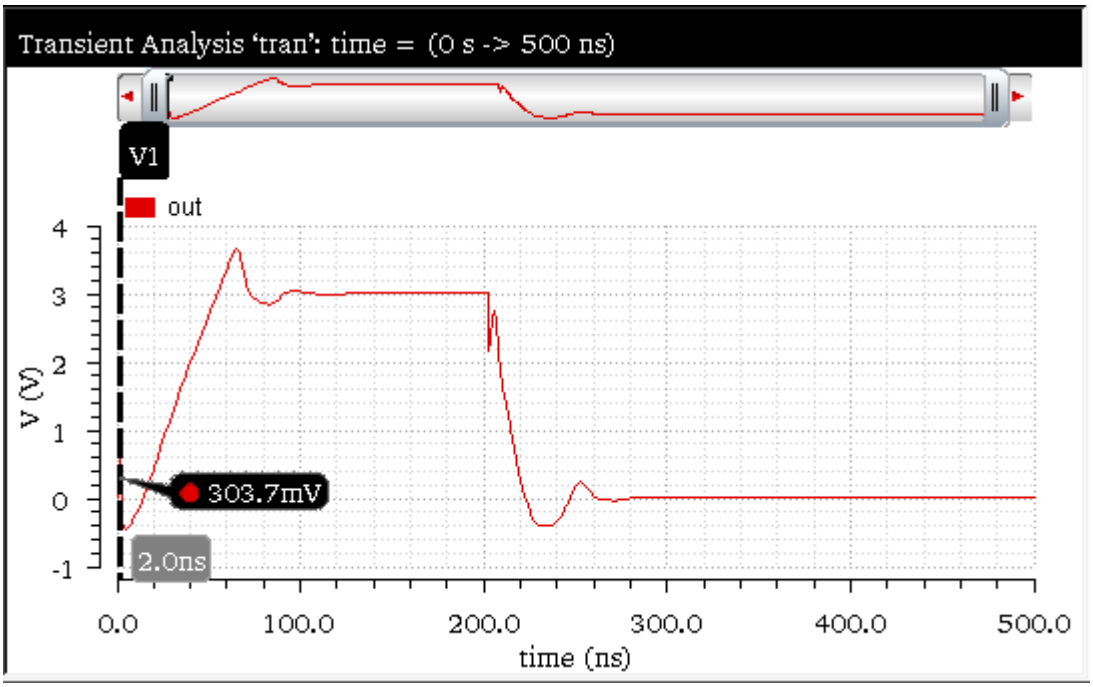
© 2023 Cadence Design Systems, Inc.

asinh

Returns the inverse hyperbolic sine or hyperbolic arc-sine of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

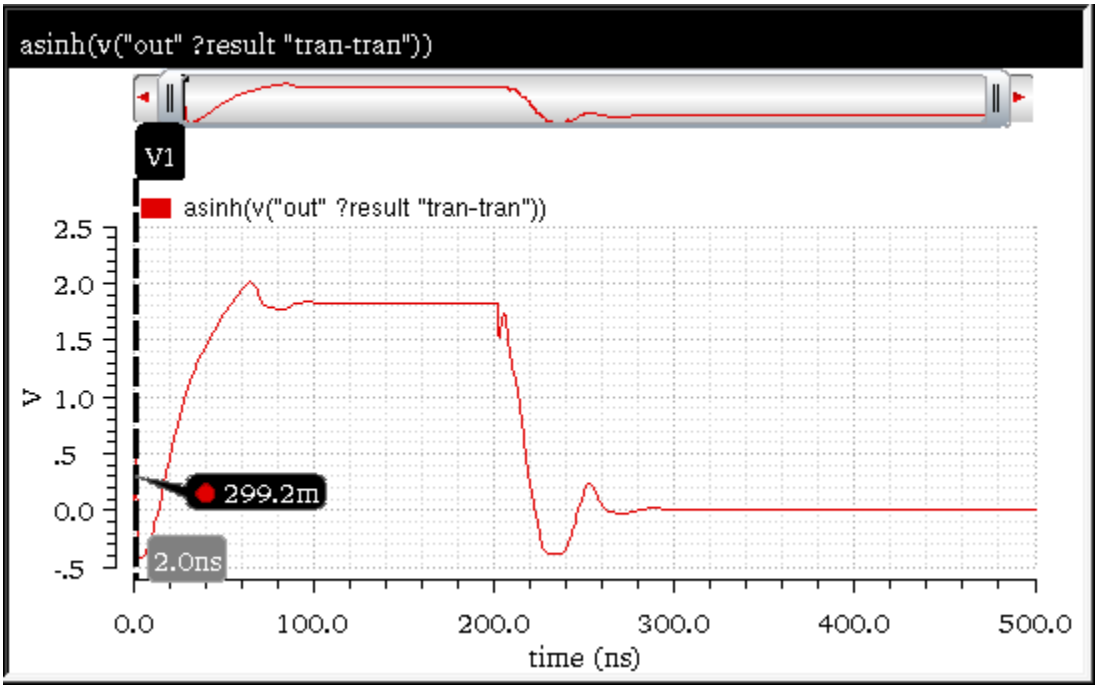
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `asinh` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
asinh(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the hyperbolic arc-sine values of the input waveform. The following output waveform is displayed in a new window:

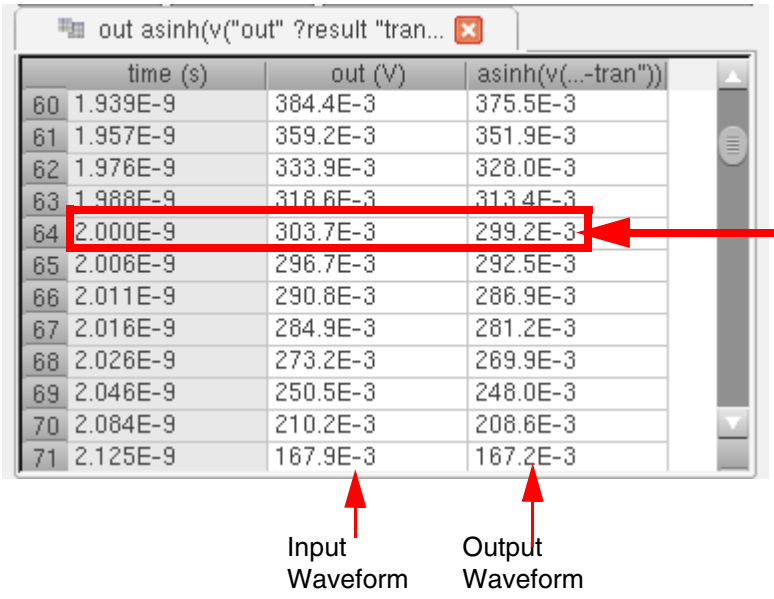


In the output waveform, the vertical maker `V1` shows the voltage=`299.2mV` when placed at time=`2.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the value of voltage in the input signal, out(V), is 303.7E-3V After the asinh function is applied, the resultant voltage value at time=2.00E-9s in output signal is 299.2E-3V. This is the arc hyperbolic sine value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

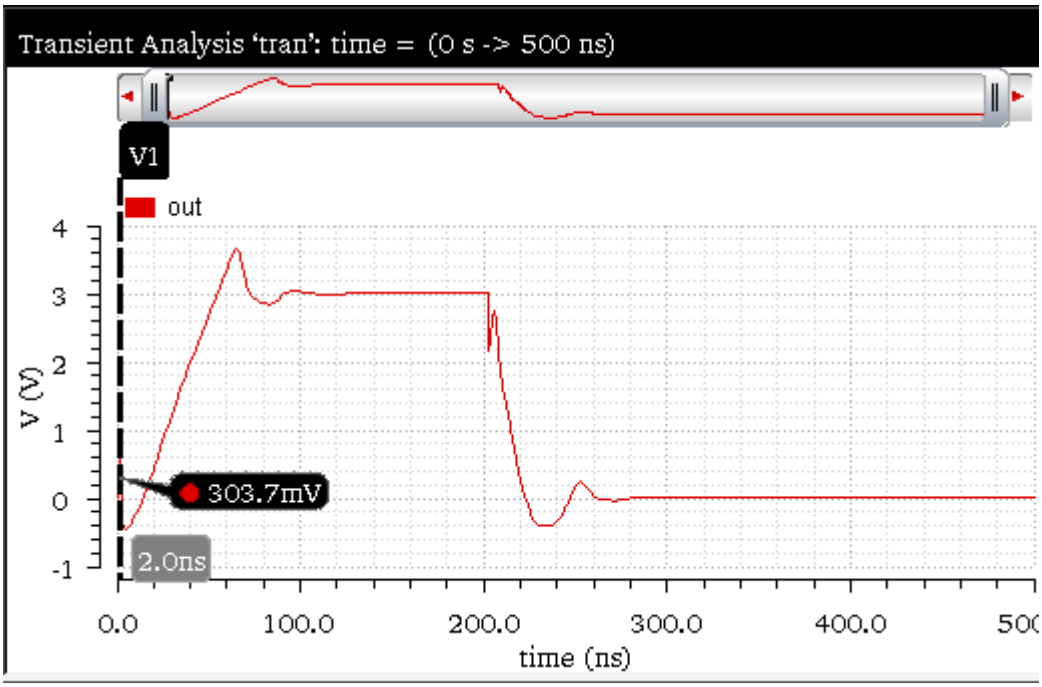
© 2023 Cadence Design Systems, Inc.

tan

Returns the tangent of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

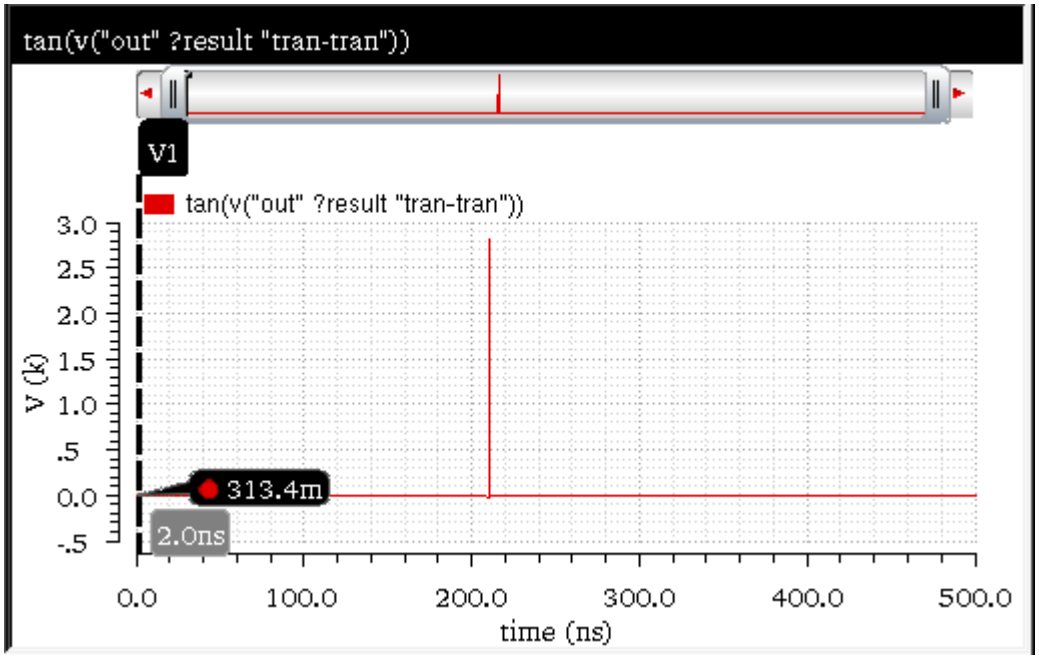
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `tan` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
tan(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the tangent values of the input waveform. The following output waveform is displayed in a new window:

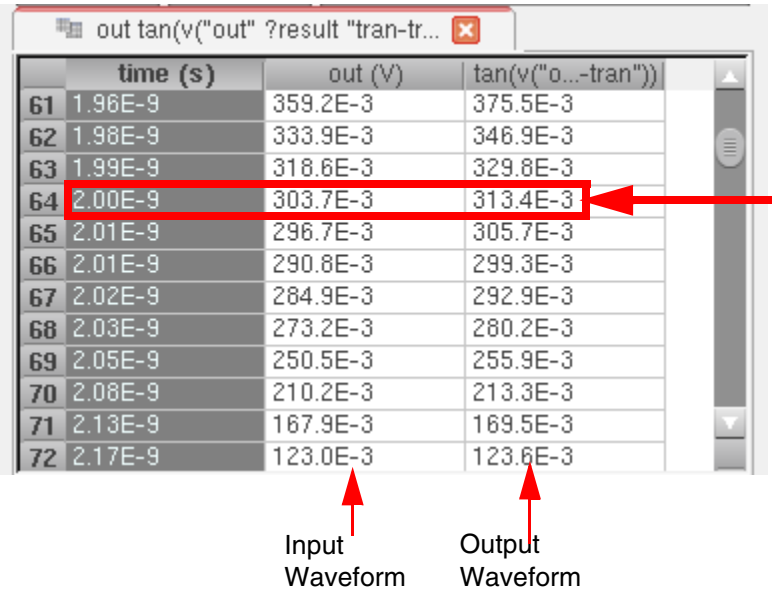


In the output waveform, the vertical maker `V1` shows the voltage=`313.4mV` when placed at time=`2.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the value of voltage in the input signal, out(V), is 303.7E-3V After the tan function is applied, the resultant voltage value at time=2.00E-9s in output signal is 313.4E-3. This is the tangent value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

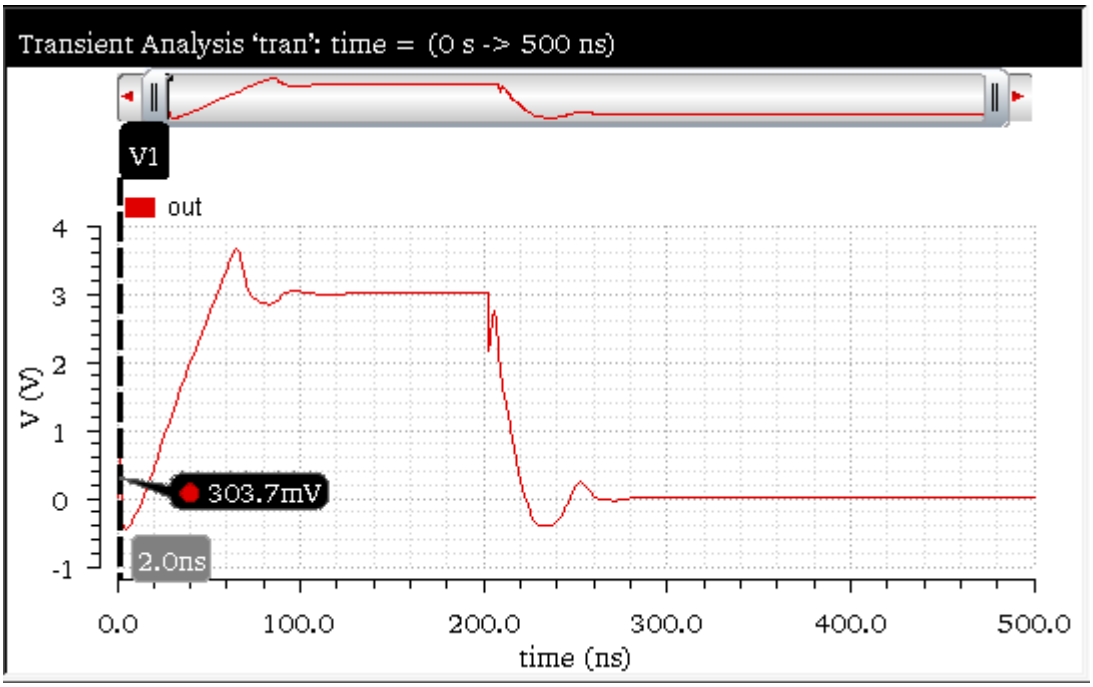
© 2023 Cadence Design Systems, Inc.

atan

Returns the nverse tangent of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

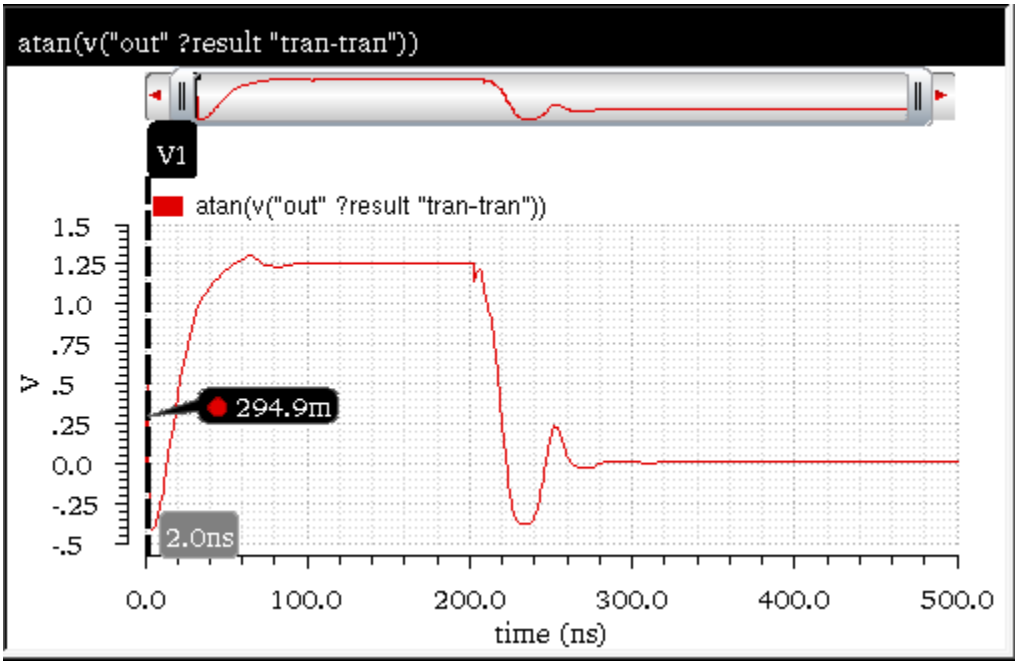
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `atan` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
atan(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the inverse tangent values of the input waveform. The following output waveform is displayed in a new window:

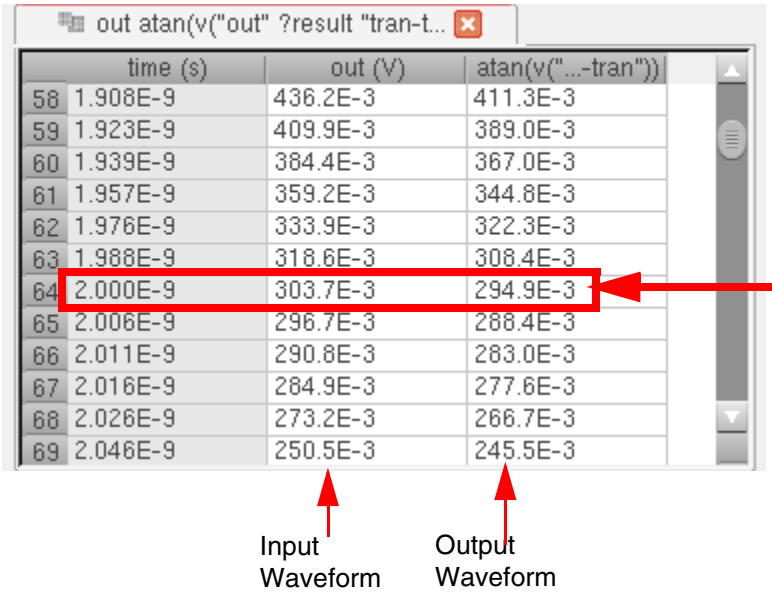


In the output waveform, the vertical maker `V1` shows the voltage=`294.9mV` when placed at time=`2.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the value of voltage in the input signal, out(V), is 303.7E-3V. After the atan function is applied, the resultant voltage value at time=2.00E-9s in output signal is 294.9E-3. This is the inverse tangent value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

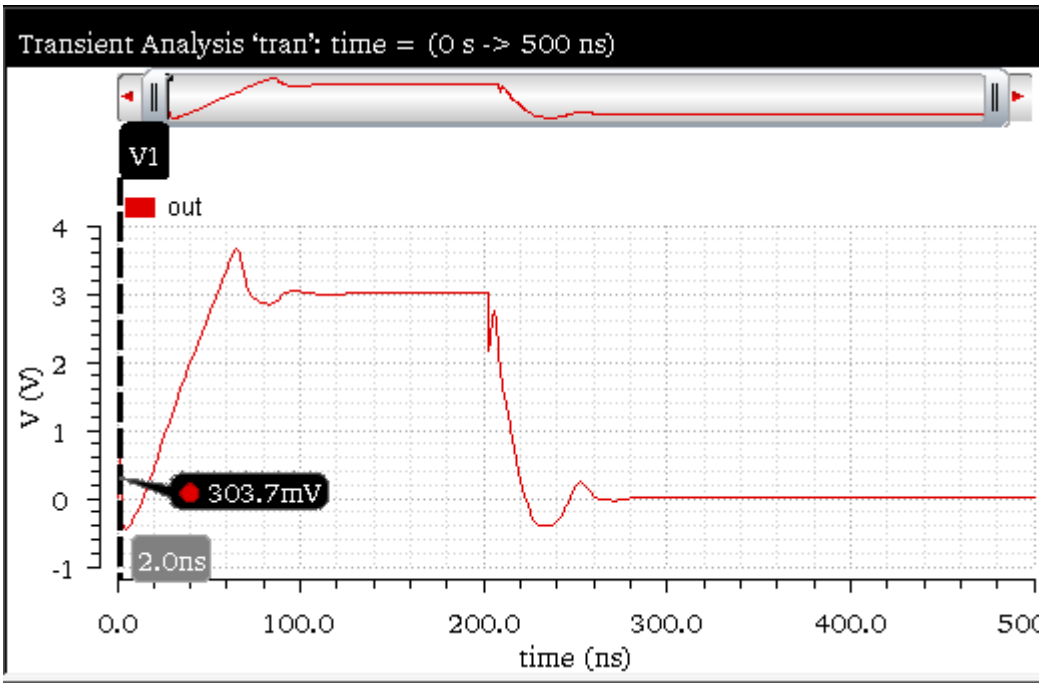
© 2023 Cadence Design Systems, Inc.

tanh

Returns the hyperbolic tangent of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

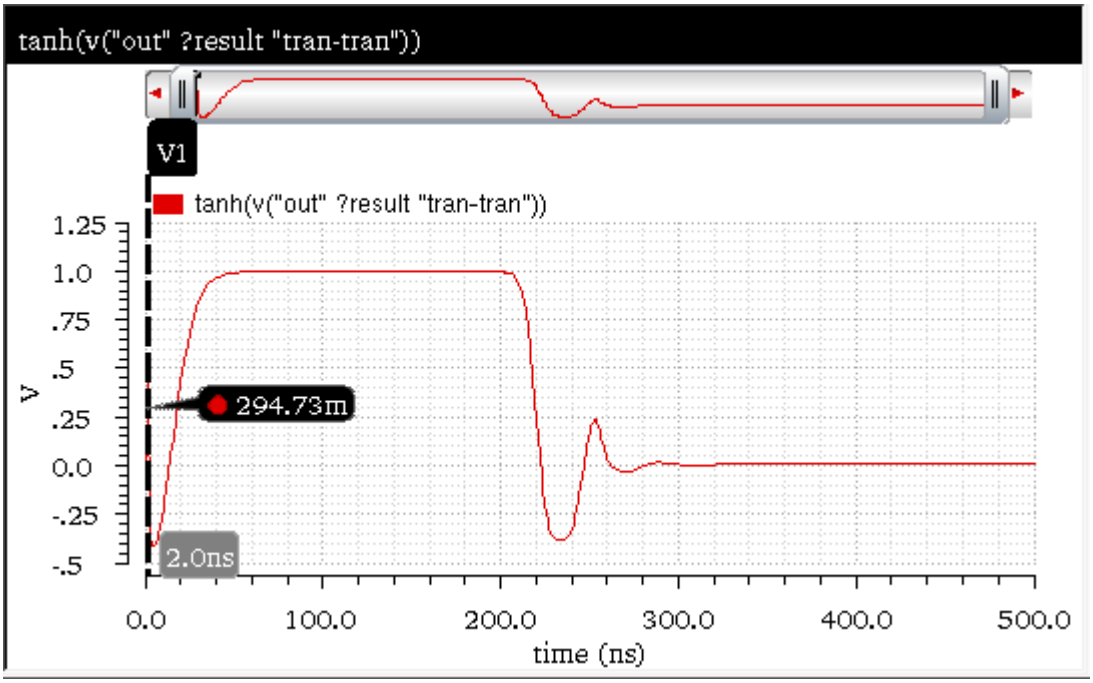
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `tanh` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
tanh(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the hyperbolic tangent values of the input waveform. The following output waveform is displayed in a new window:

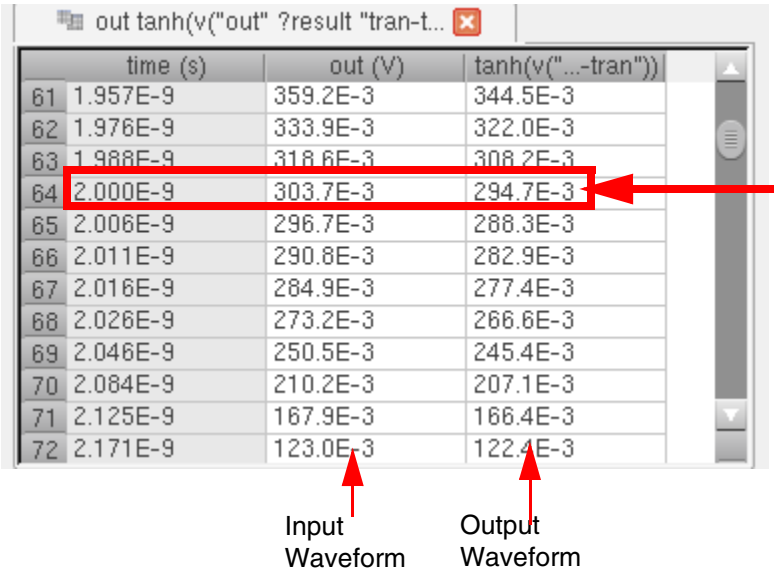


In the output waveform, the vertical maker `V1` shows the voltage=`294.93mV` when placed at time=`2.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the value of voltage in the input signal, out(V), is 303.7E-3V .After the tanh function is applied, the resultant voltage value at time=2.00E-9s in output signal is 294.7E-3. This is the hyperbolic tangent value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and

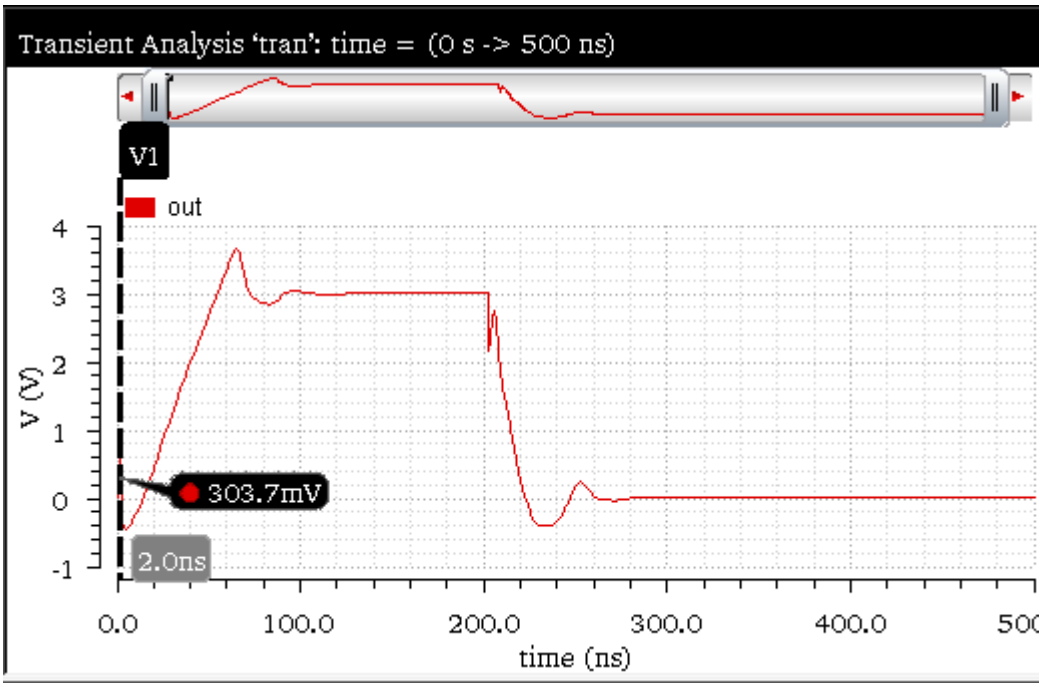
© 2023 Cadence Design Systems, Inc.

atanh

Returns the inverse hyperbolic tangent or hyperbolic arc tangent of a signal.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`2.0ns` and voltage=`303.7mV`.

When you send this signal to Calculator, the following expression is created in the Buffer:

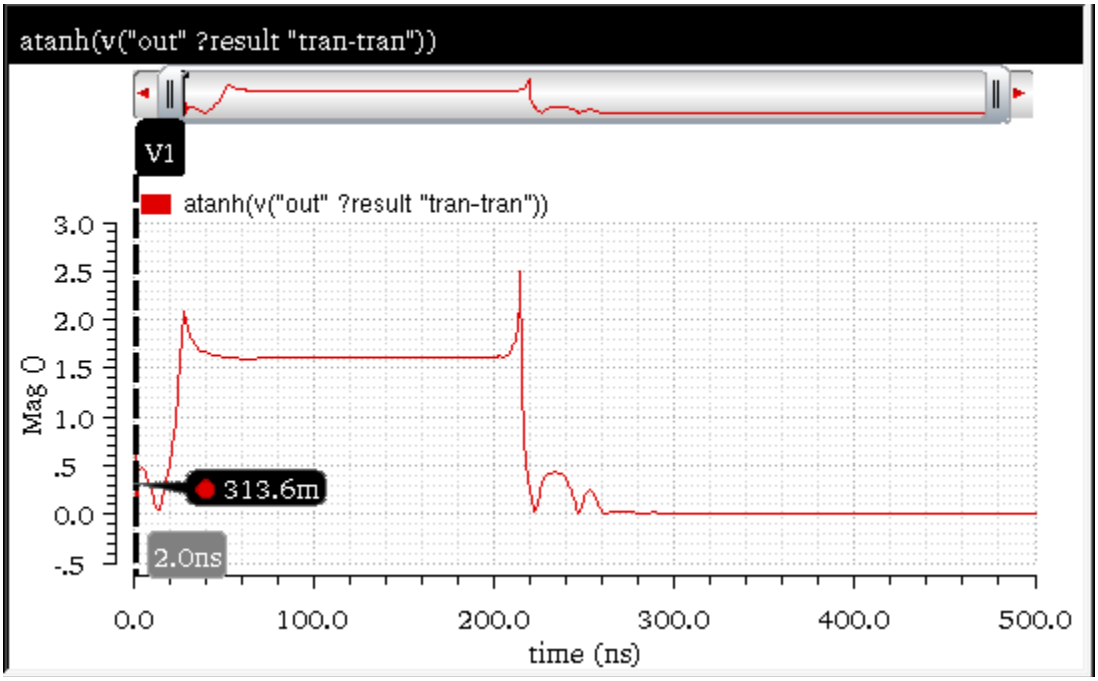
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `atanh` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in the Buffer is as follows:

```
atanh(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the function returns the arc hyperbolic tangent values of the input waveform. The following output waveform is displayed in a new window:

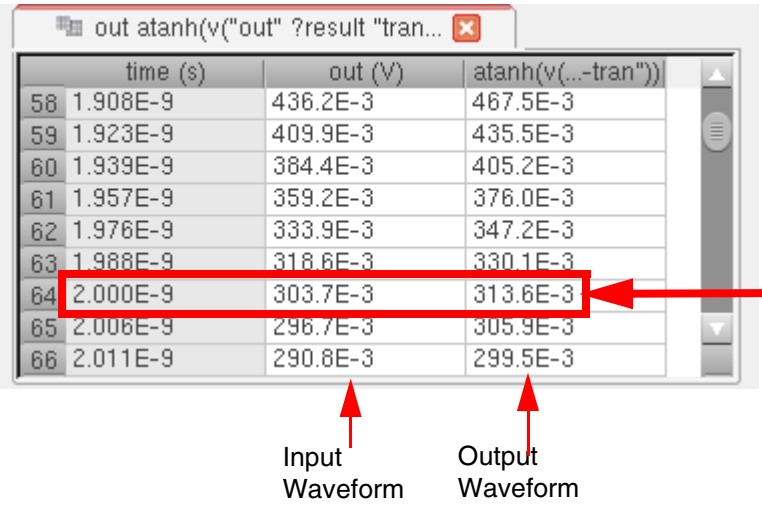


In the output waveform, the vertical maker `V1` shows the voltage=`313.6mV` when placed at time=`2.0ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=2.00E-9s, the value of voltage in the input signal, out(V), is 303.7E-3V. After the tanh function is applied, the resultant voltage value at time=2.00E-9s in output signal is 313.6E-3. This is the arc hyperbolic tangent value of 303.7E-3. (You can also refer to vertical markers in the above figures for input and output waveforms)

© 2023 Cadence Design Systems, Inc.

Math Functions

This section covers the following Math functions in Calculator:

- 1/x
- 10**x
- abs
- exp
- int
- ln
- log10
- sqrt
- x**2
- y**x

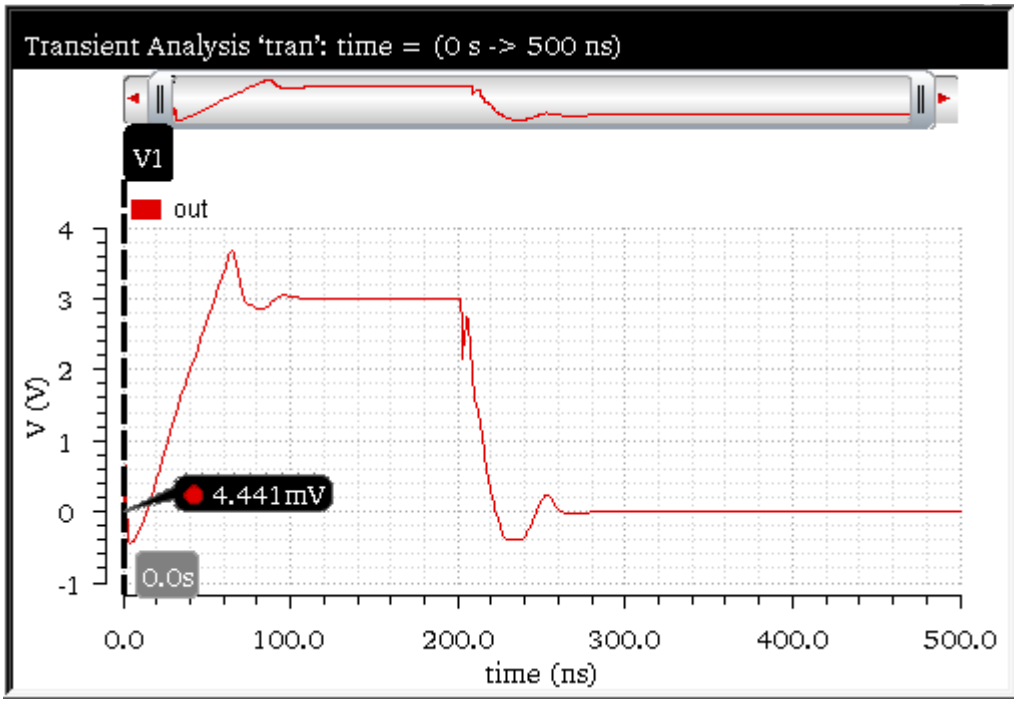
© 2023 Cadence Design Systems, Inc.

1/x

Returns the inverse value. This function is available only in the SKILL mode.

Example

Consider the following signal from the transient analysis:



When you send this signal to Calculator, the following expression is created in the Buffer:

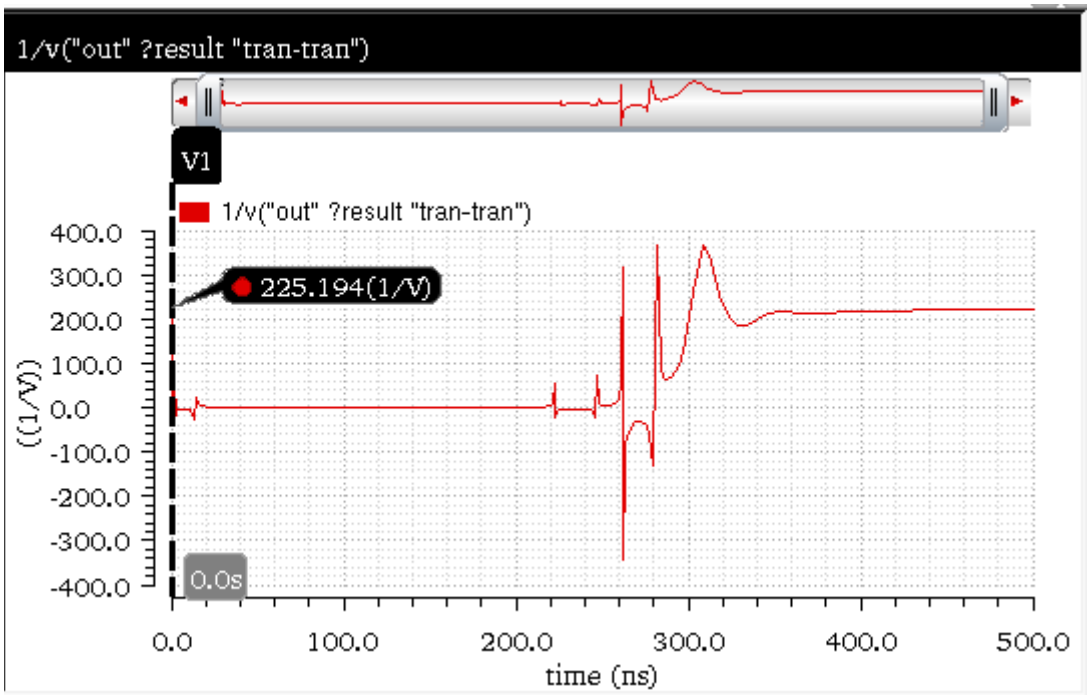
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `1/x` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
1/v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the inverse of the signal is calculated and the following output signal is displayed in a new graph window:

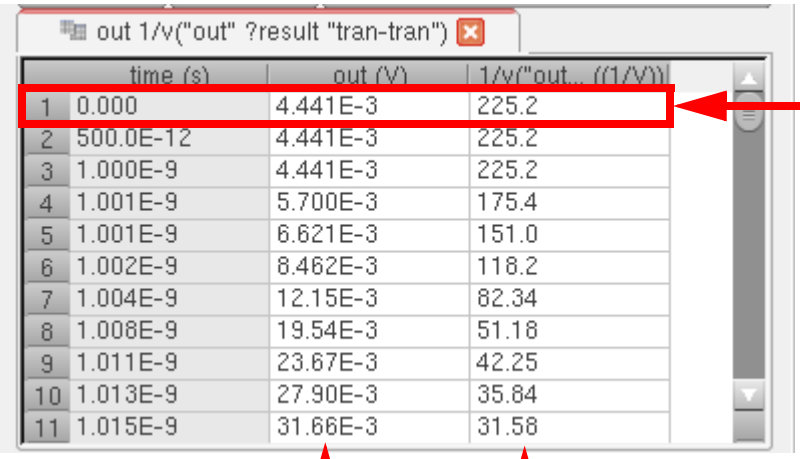


In the output waveform, the vertical maker `V1` shows the voltage=`225.194V` when placed at time=`0.0ns`, which is the inverse of the voltage value (`1/4.441mV`) shown in the input waveform.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal, out(V), is 4.441E-3. After the 1/x function is applied, the inverse voltage value at time=0.00ns in output signal is calculated as 225.2V. (Also, see vertical markers in the above figures for input and output waveforms)

Input waveform values

Output waveform values

Related OCEAN Function

The equivalent OCEAN command for `1/x` is:

For more information, see 1/x in *OCEAN Reference*.

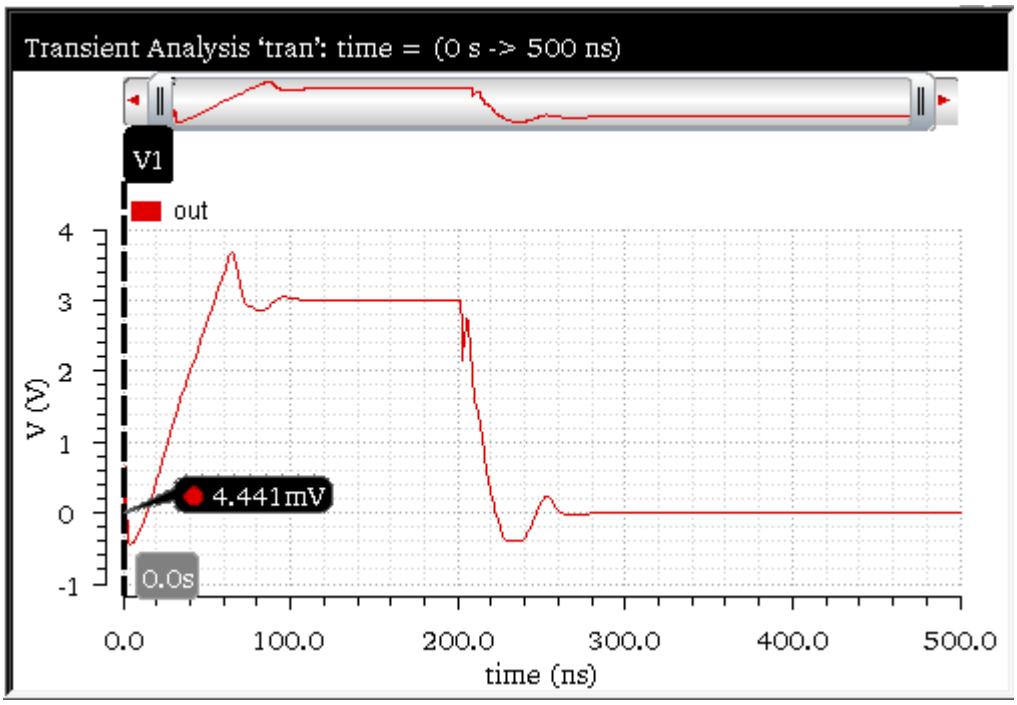
© 2023 Cadence Design Systems, Inc.

10**x

Returns the 10^x value of the input signal. This function is available only in the SKILL mode.

Example

Consider the following input signal form a transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mv`.

When you send this signal to Calculator, the following expression is created in the Buffer:

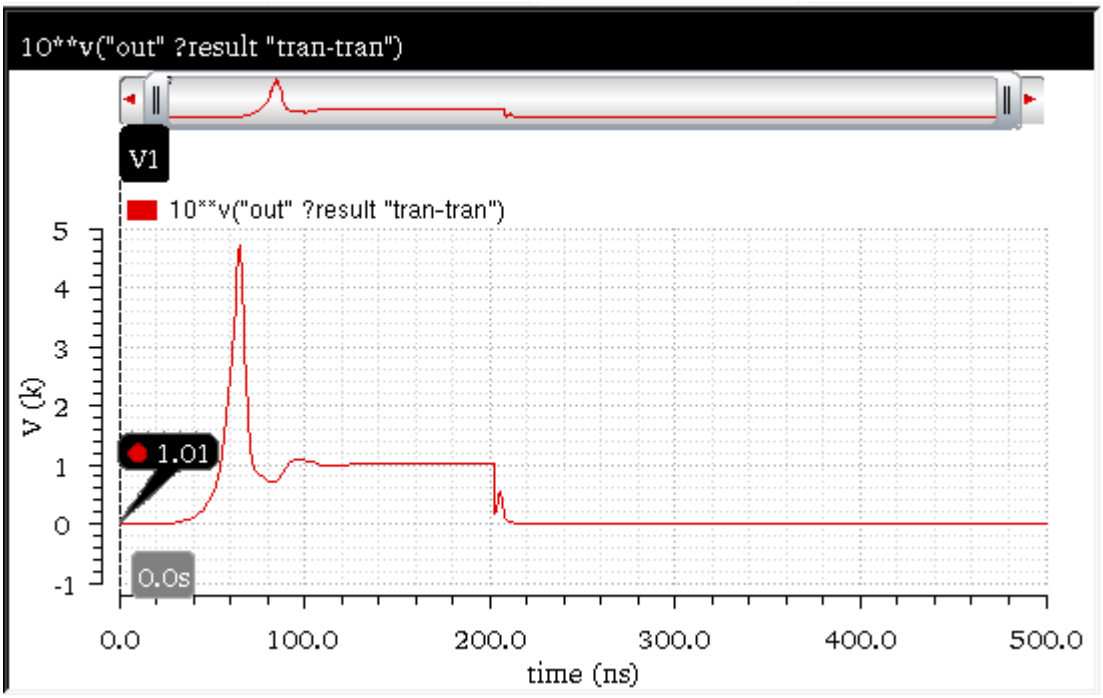
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `10**x` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
10**v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the 10^x of the signal is calculated and the following output is displayed in a new graph window:

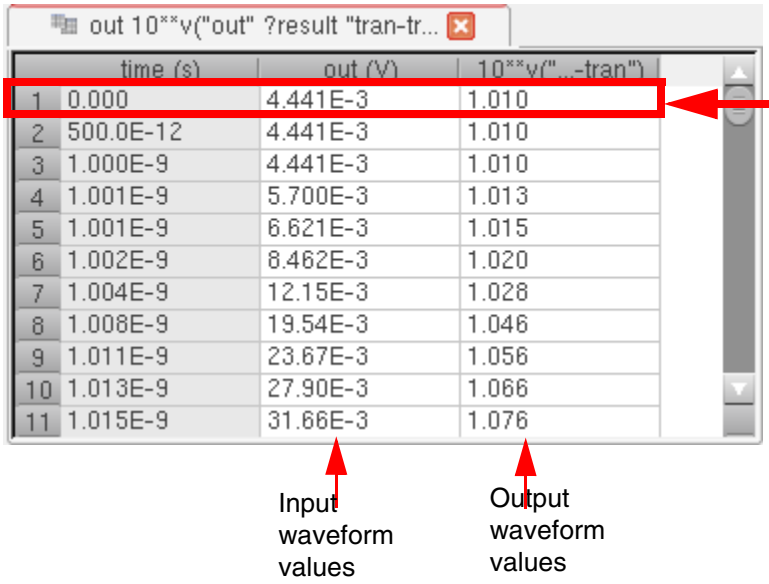


In the output waveform, the vertical maker `V1` shows the voltage=`1.01mV` when placed at time=`0.0ns`, which is the 10^x of the voltage value (`4.441mV`) shown in the input waveform.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal, out(V), is 4.441E-3. After the 10**x function is applied, the voltage value at time=0.00ns in output signal is shown as 1.010 mV (Refer to vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `10**x` is:

For more information, see `10**x` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

abs

Returns the absolute value of the specified value or signal.

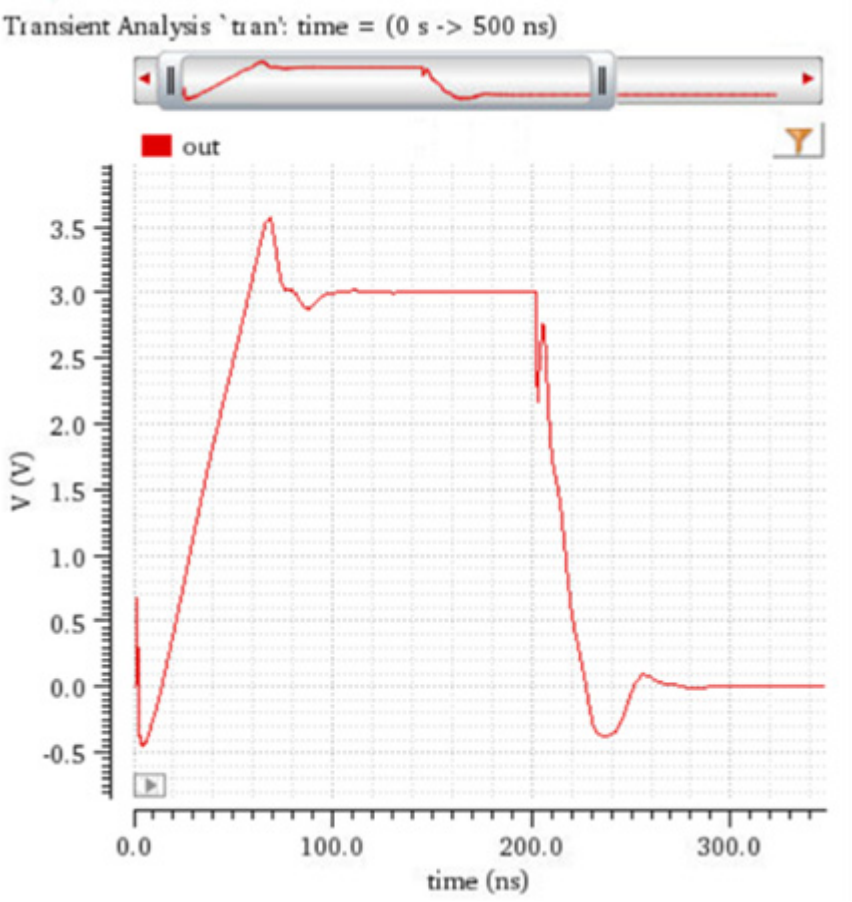
The absolute value of y is represented by |y| and it is calculated using the following formula:

$$|y| = \begin{cases} y & y \geq 0 \\ -y & y < 0 \end{cases}$$

Therefore, the absolute value of y is always either a positive number or zero, but never a negative number.

Example

Consider the following input signal from a transient analysis.



The following expression is created in the Buffer when you send this signal to the Calculator.

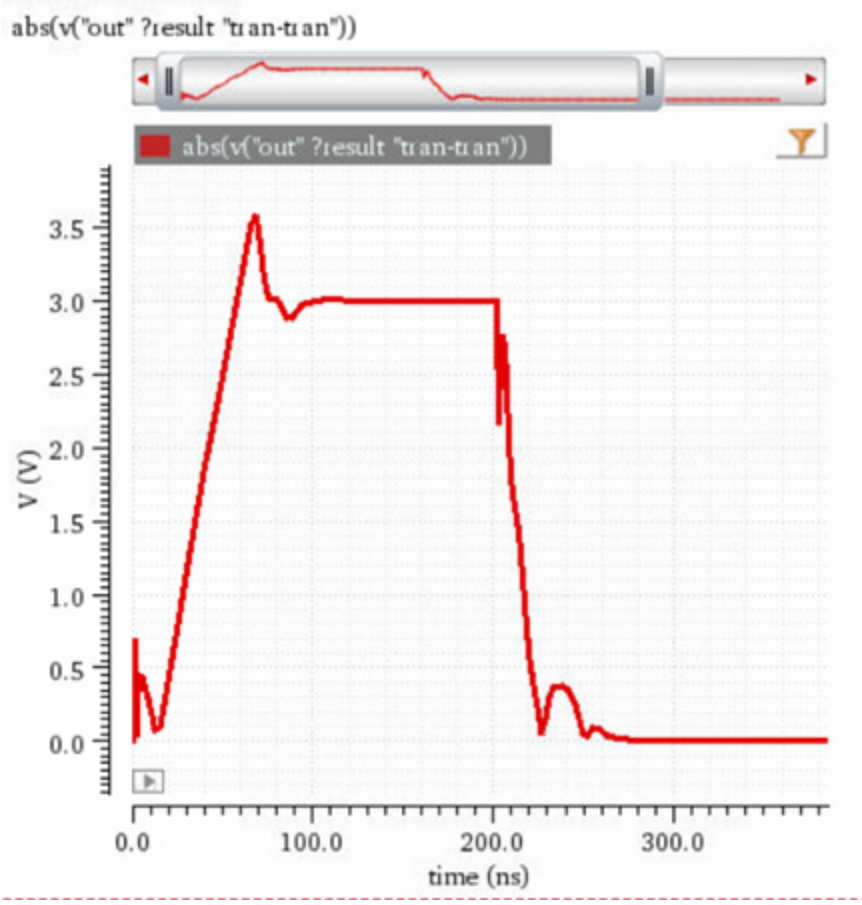
© 2023 Cadence Design Systems, Inc.

```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the function abs to this signal from the Function Panel, the following expression is created in the Buffer.

```
abs(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```

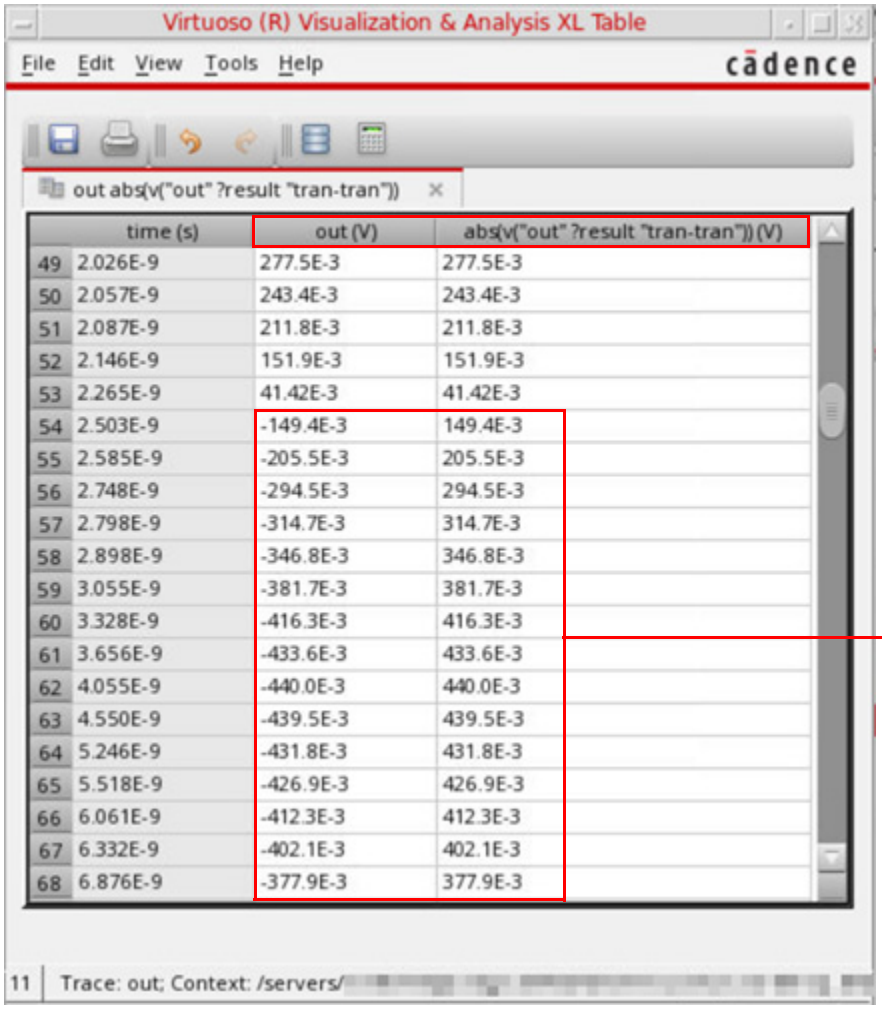
When you evaluate this expression, the absolute value of the signal is calculated and the resultant waveform is plotted in the graph window.



To analyze the output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal

© 2023 Cadence Design Systems, Inc.

to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.



Negative y values in the input signal are converted to positive y values in the output signal

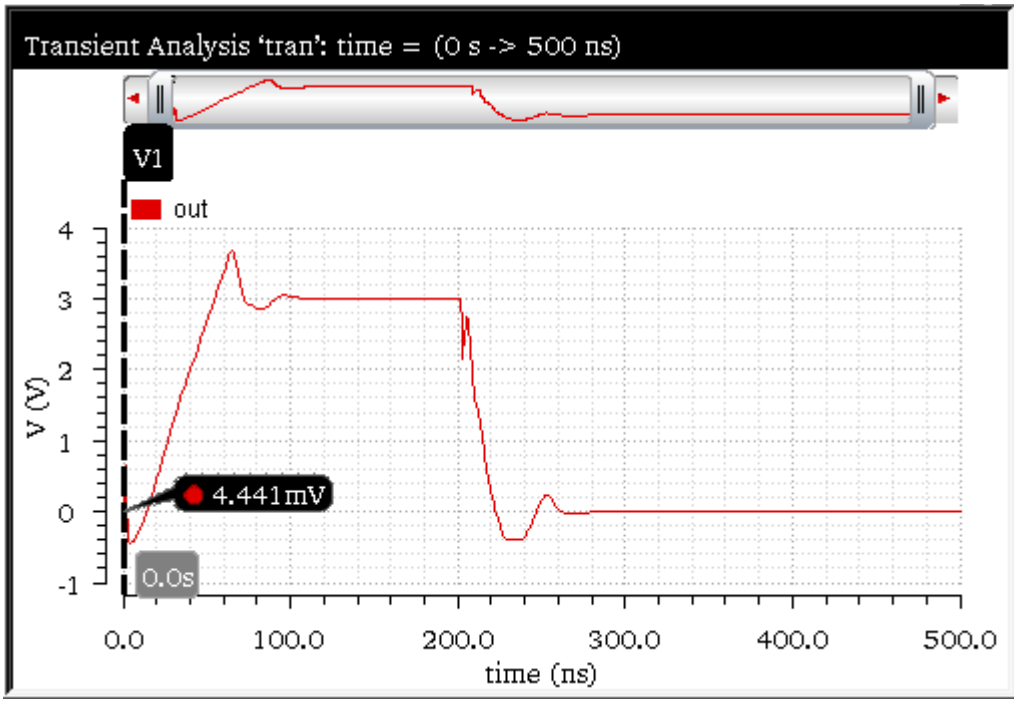
© 2023 Cadence Design Systems, Inc.

exp

Returns the exponential value (e^x) of the signal.

Example

Consider the following input signal from transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mv`.

When you send this signal to Calculator, the following expression is created in the Buffer:

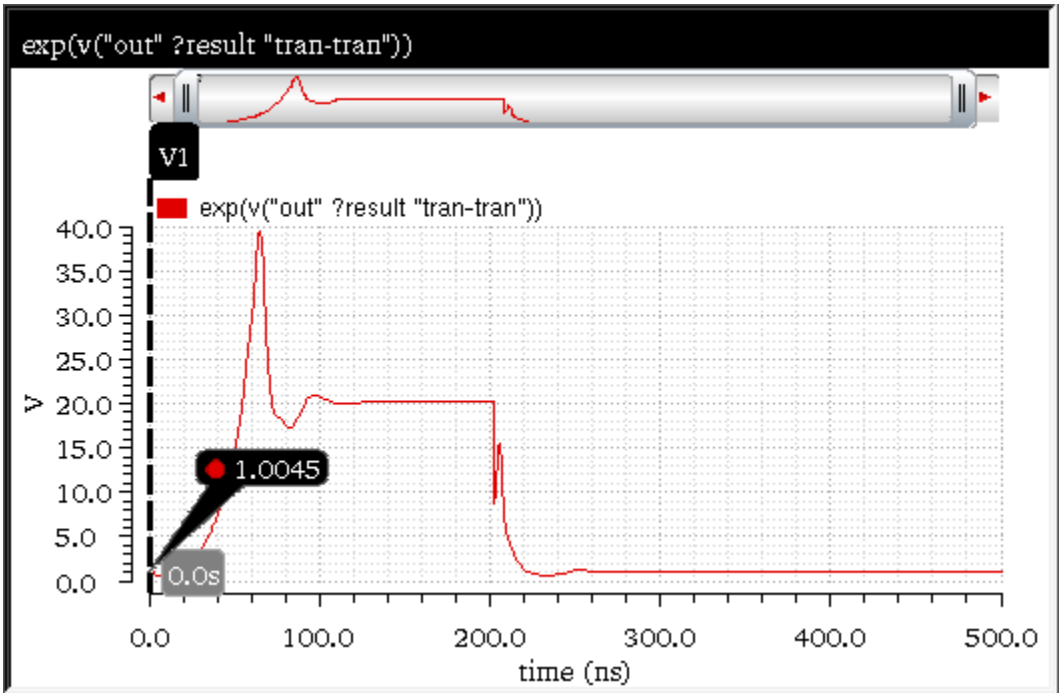
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `exp` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
exp(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the exponential value of the input signal is calculated and the following output waveform is plotted in a new graph window:

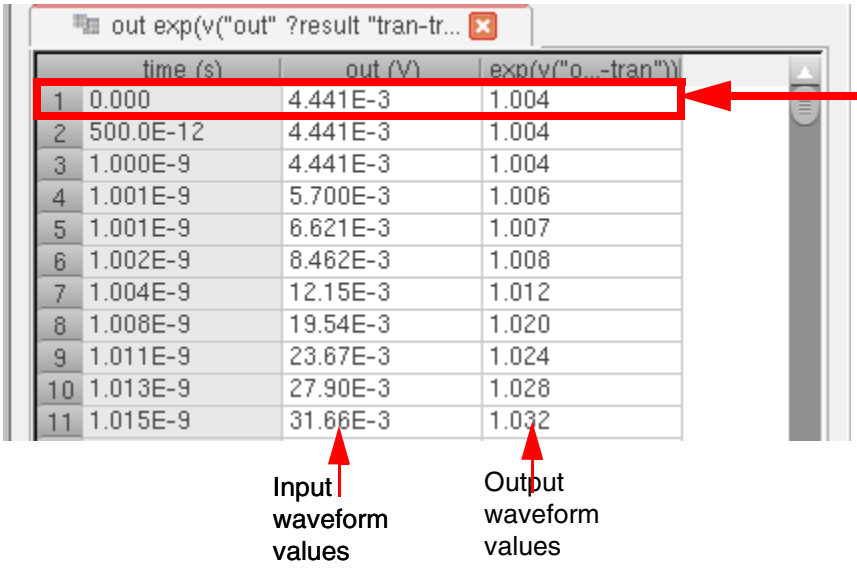


In the output waveform, the vertical maker `V1` shows the voltage=`1.0045mV` when placed at time=`0.0ns`, which is the exponential of the voltage value (`4.441mV`) shown in the input waveform.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal, out(V), is 4.441E-3. After the exp function is applied, the voltage value at time=0.00ns in output signal is shown as 1.004. (Also, see vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `exp` is:

```
exp( n_number )
     => f_result
```

For more information, see `exp` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

int

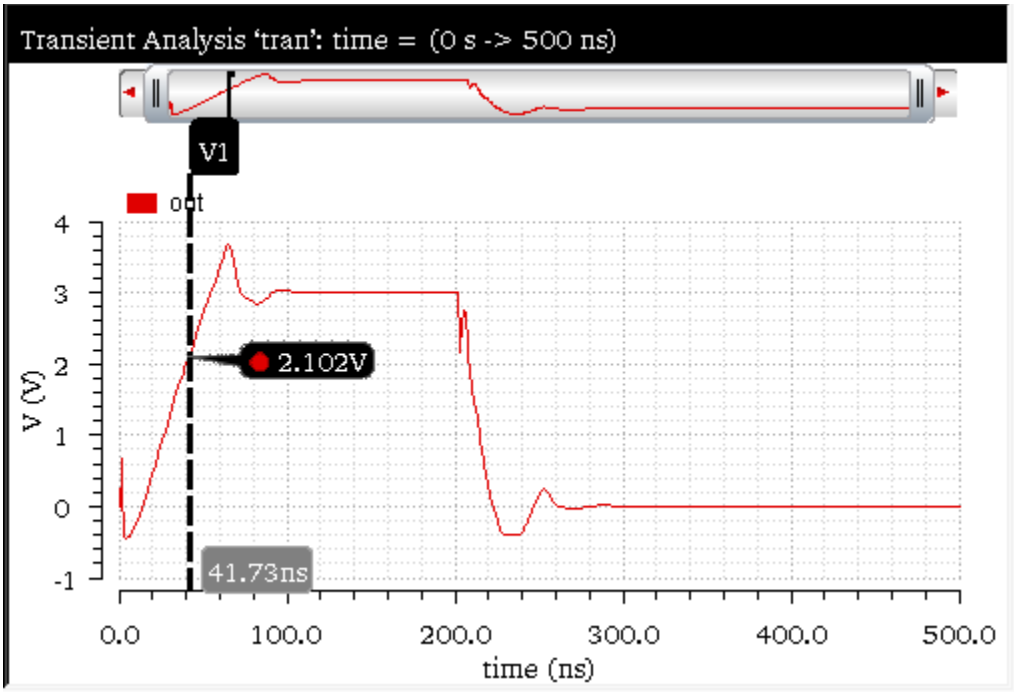
Returns the integer portion of a real value.

Note: While performing the division of two integer or floating point values, if denominator is greater than numerator, the division results are truncated. For example, if you perform the division of `1/2`, the `int` function displays the result as `0` instead of `0.5`.

For more information about how the integer and floating point division is performed using SKILL, see Integer vs. Floating-Point Division in the in the Arithmetic and Logical Expressions chapter of *Cadence SKILL Language User Guide*.

Example

Consider the following input signal from the transient analysis:



A vertical marker `V1` is placed at time=`41.73ns` and voltage=`2.102V`.

When you send this signal to Calculator, the following expression is created in the Buffer:

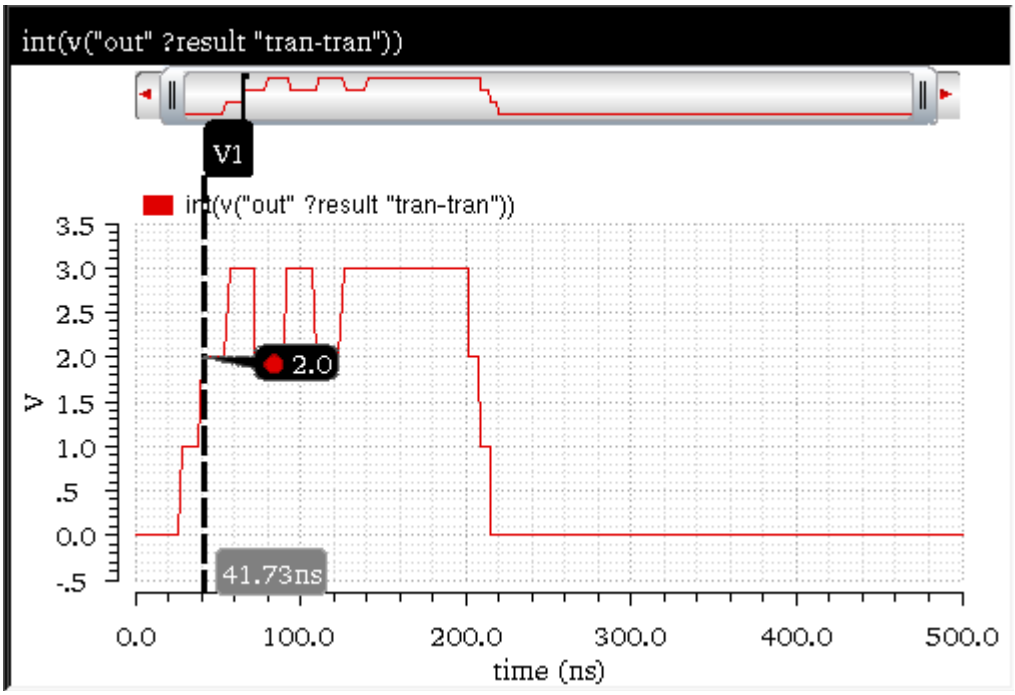
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `int` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

© 2023 Cadence Design Systems, Inc.

```
int(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```

When you evaluate this expression, the function returns the integer voltage values of the input waveform. The following output waveform is displayed in a new window:

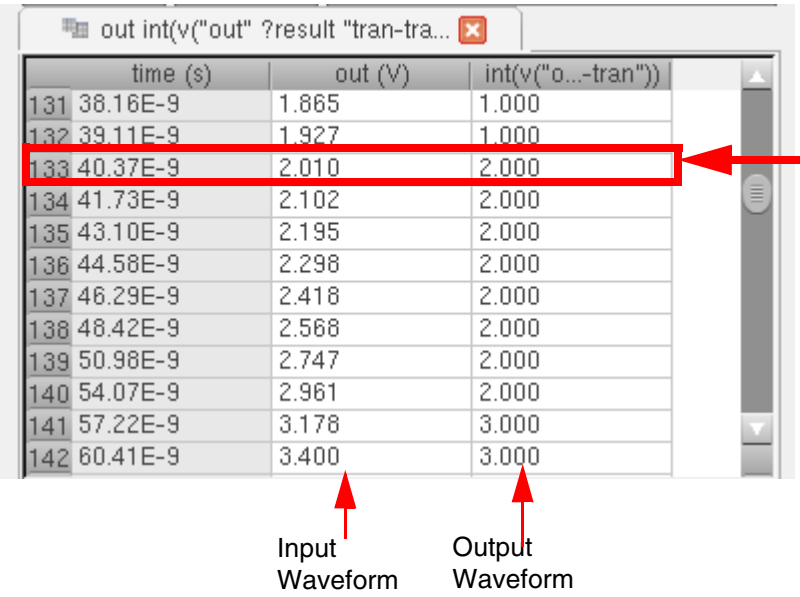


In the output waveform, the vertical maker `V1` shows the voltage=`2.0V` when placed at time=`41.73ns`.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=41.73ns, the voltage value in the input signal1, out(V), is 2.102. After the int function is applied, the resulted voltage value at time=41.73ns in output signal is shown as 2.000, which shows the integer value from 2.102. (Refer to vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `int` is:

For more information, see `int` in *OCEAN Reference*.

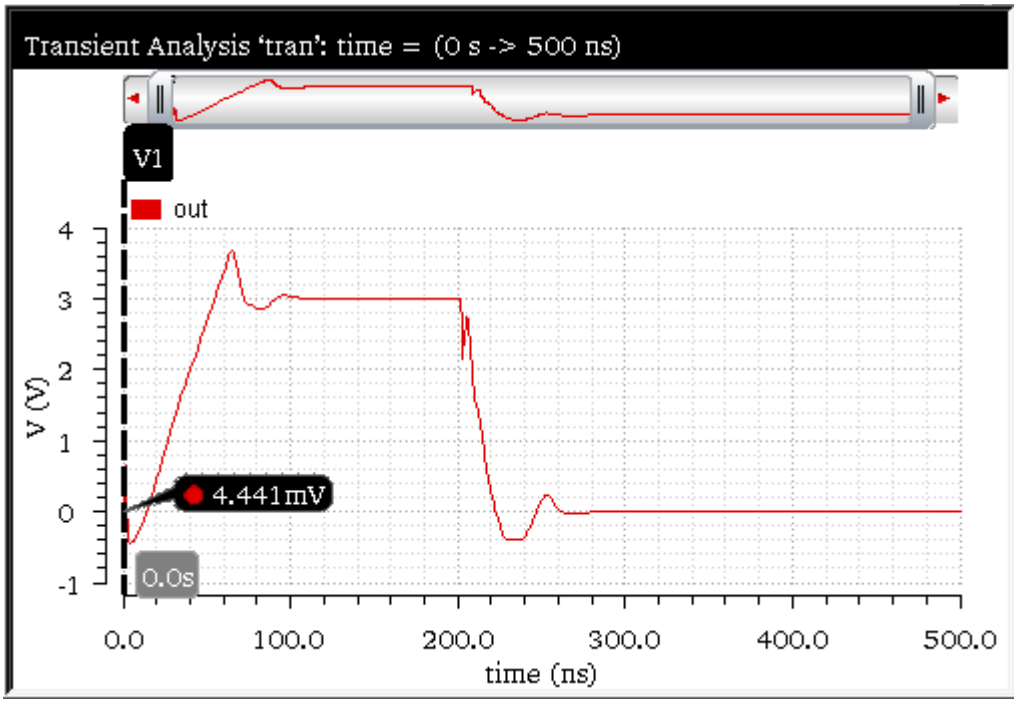
© 2023 Cadence Design Systems, Inc.

ln

Returns the natural logarithm value of the given signal.

Example

Consider the following input signal from transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mv`.

When you send this signal to Calculator, the following expression is created in the Buffer:

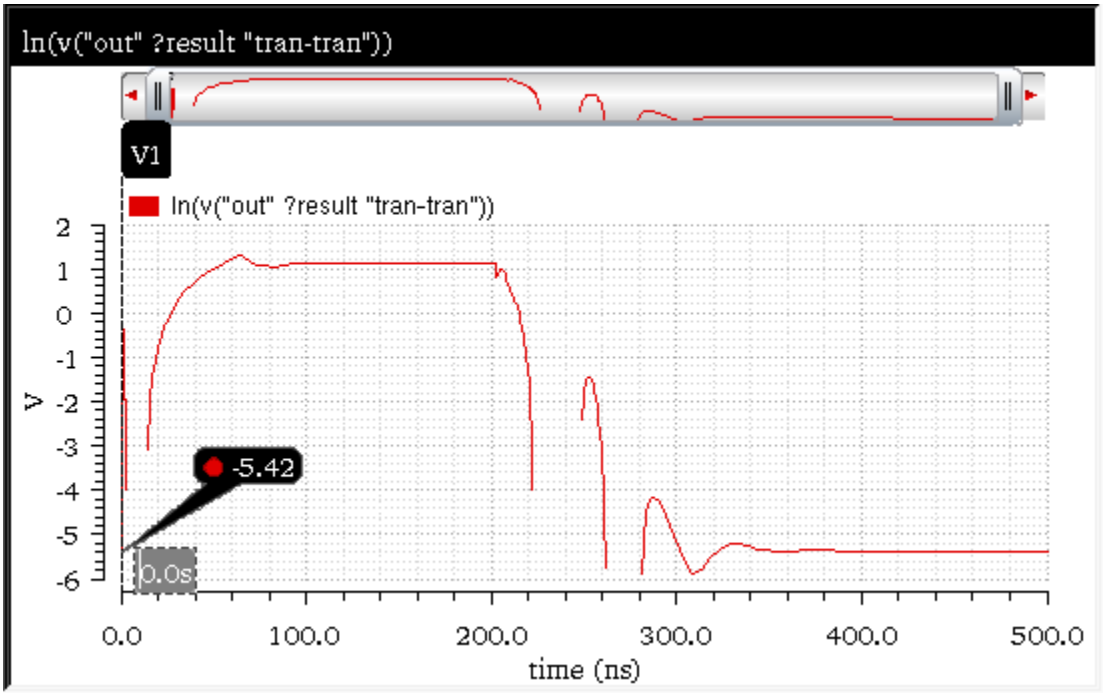
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `ln` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
ln(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the exponential value of the input signal is calculated and the following output waveform is plotted in a new graph window:

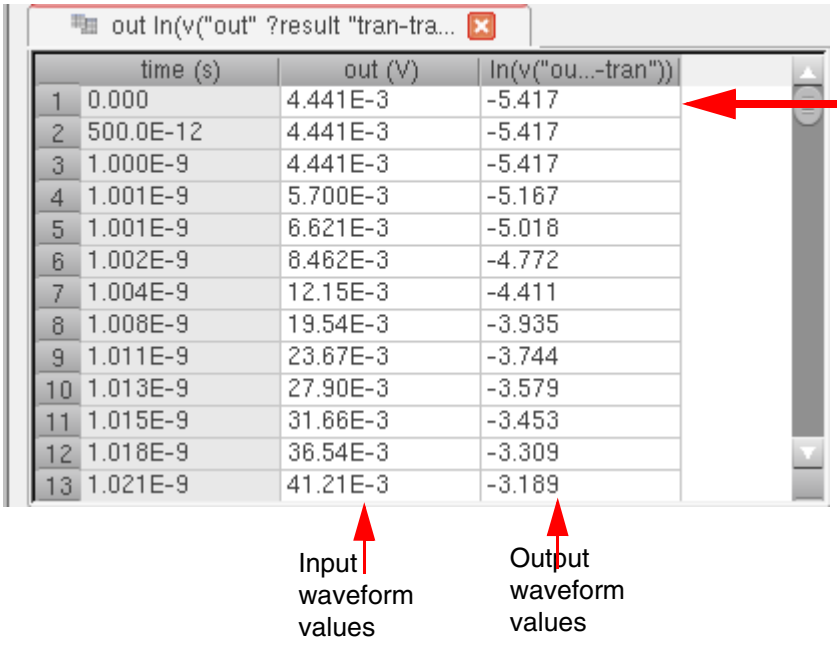


In the output waveform, the vertical maker `V1` shows the voltage=`-5.42mV` when placed at time=`0.0ns`, which is the natural logarithmic value of the voltage value (`4.441E-3mV`) shown in the input waveform.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal, out(V), is 4.441E-3. After the ln function is applied, the voltage value at time=0.00ns in output signal is shown as -5.417. (Refer to see vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `ln` is:

For more information, see `ln` in *OCEAN Reference*.

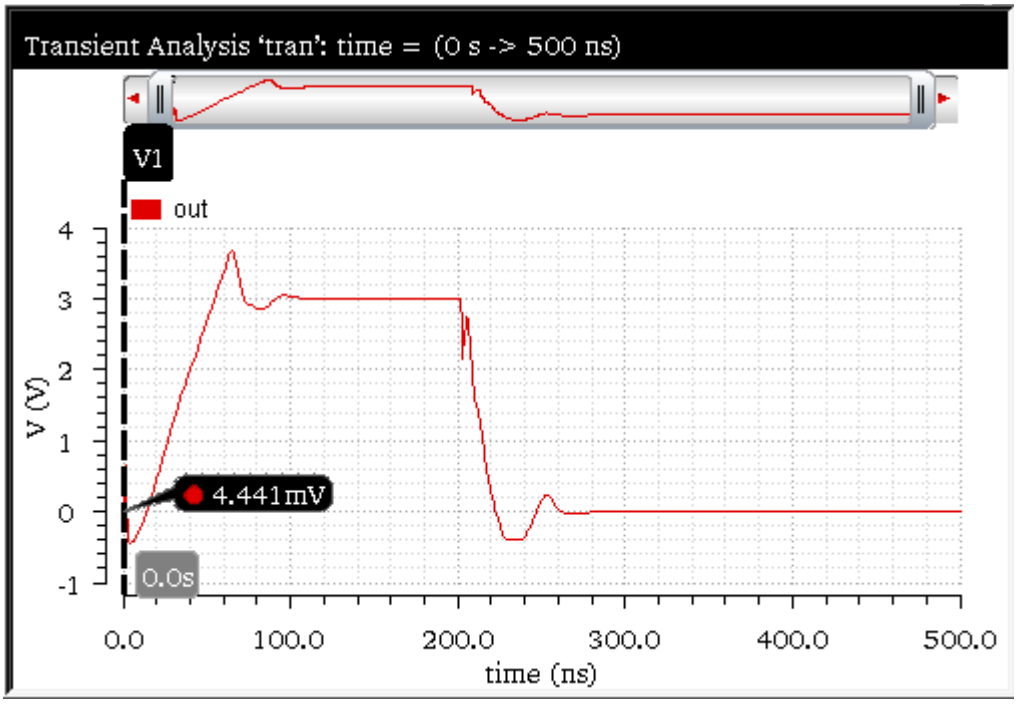
© 2023 Cadence Design Systems, Inc.

log10

Returns the base 10 logarithm of a signal.

Example

Consider the following input signal from transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mv`.

When you send this signal to Calculator, the following expression is created in the Buffer:

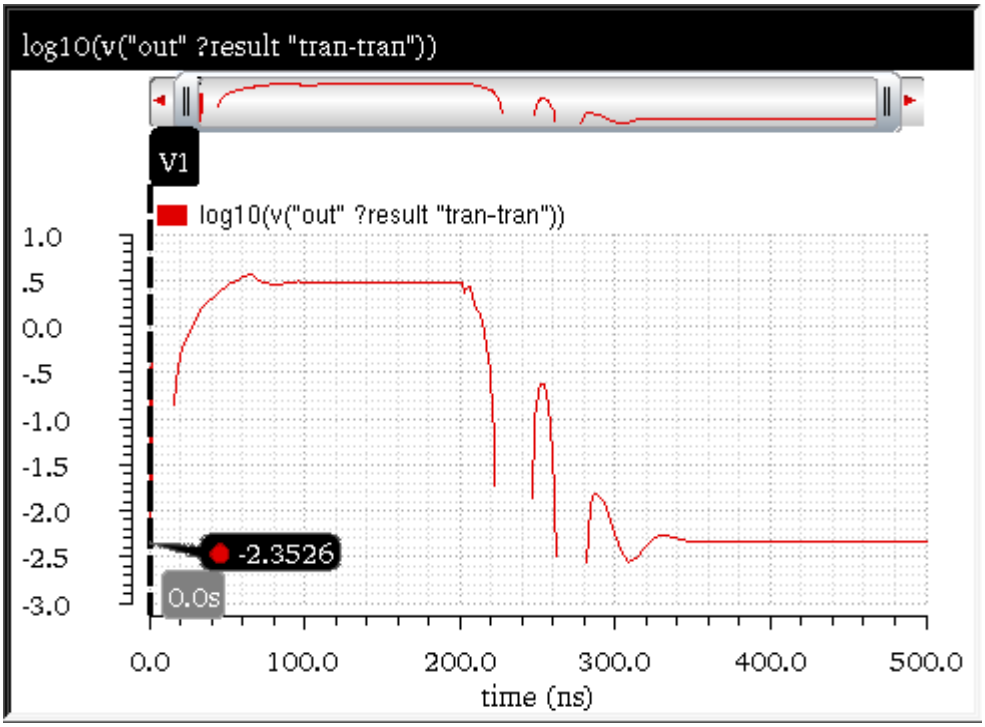
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `log10` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
log10(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the log10 value of the input waveform is calculated and the following output signal is displayed in a new window:

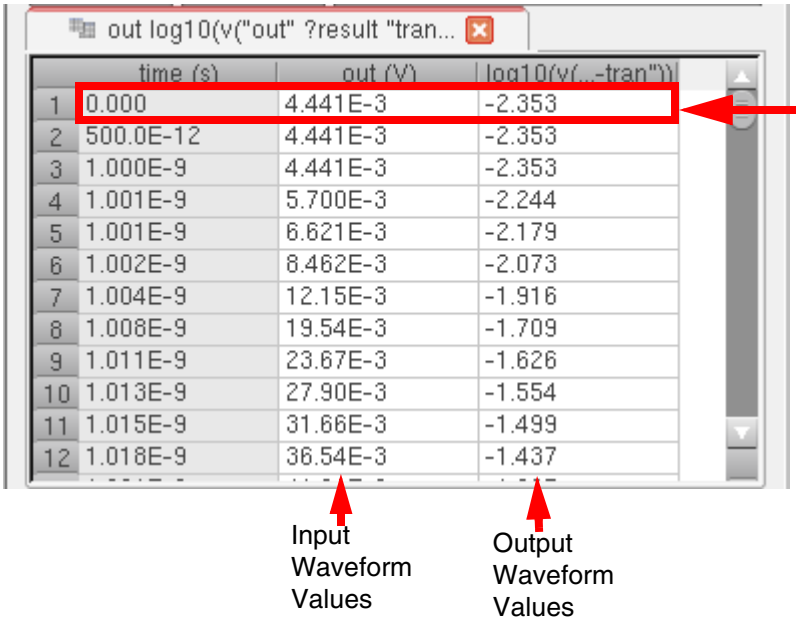


In the output waveform, the vertical maker `V1` shows the voltage=`-2.3526mV` when placed at time=`0.0ns`, which is the logarithmic to the base 10 value of the voltage value (`4.441E-3mV`) shown in the input waveform.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal, out(V), is 4.441E-3. After the log10 function is applied, the voltage value at time=0.00ns in output signal is shown as -2.353. (Refer to vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `log10` is:

For more information, see `log10` in *OCEAN Reference*.

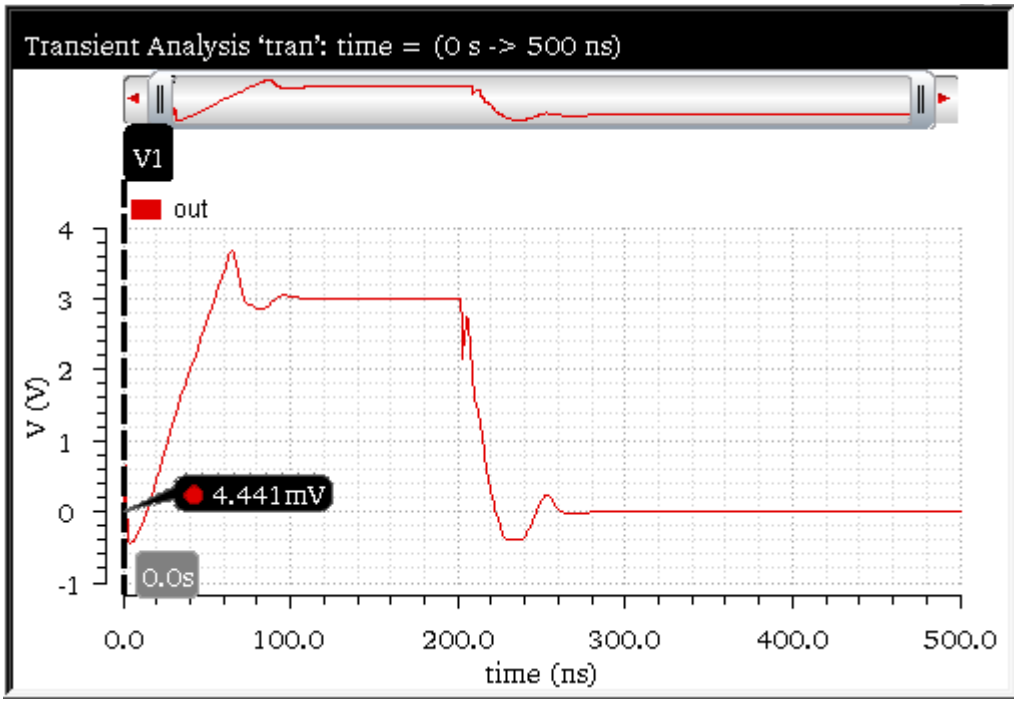
© 2023 Cadence Design Systems, Inc.

sqrt

Returns the square root of the given input waveform.

Example

Consider the following input signal from transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mv`.

When you send this signal to Calculator, the following expression is created in the Buffer:

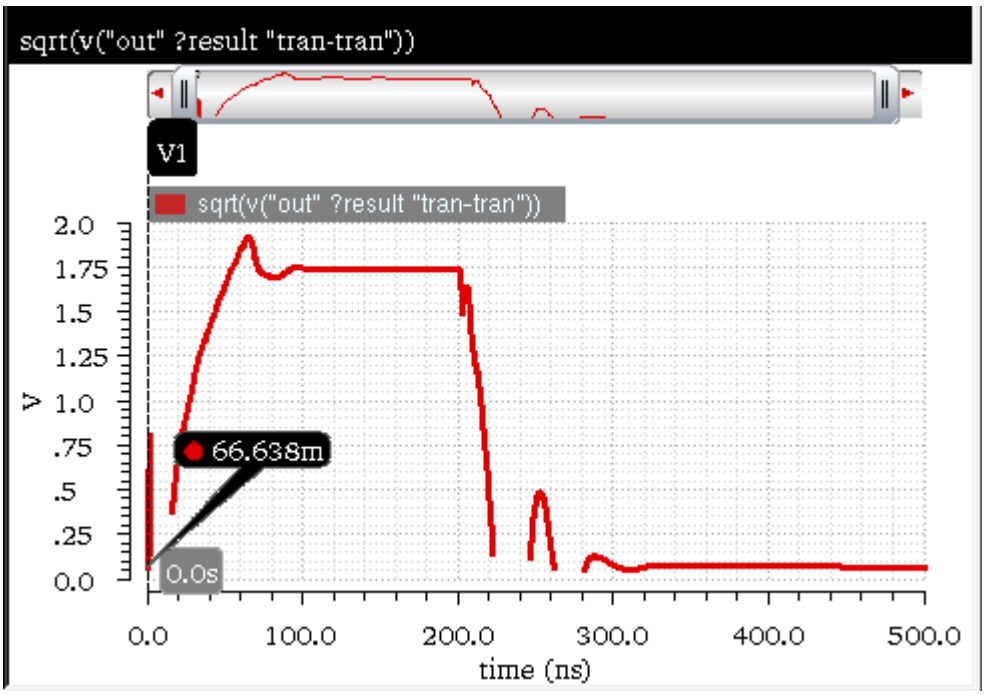
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `sqrt` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
sqrt(v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw"))
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the square root of the input waveform is calculated and the following output signal is displayed in a new window:

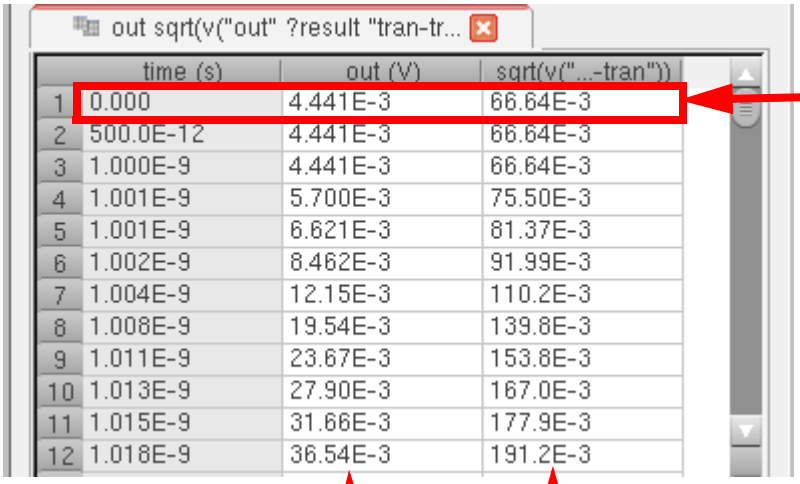


In the output waveform, the vertical maker `V1` shows the voltage=`-2.3526mV` when placed at time=`0.0ns`, which is the square root of the voltage value (`4.441E-3mV`) shown in the input waveform.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal, out(V), is 4.441E-3. After the sqrt function is applied, the voltage value at time=0.00ns in output signal is shown as 66.64E-3. (Refer to vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `sqrt` is:

```
sqrt( n_number )
     => f_result
```

For more information, see `sqrt` in *OCEAN Reference*.

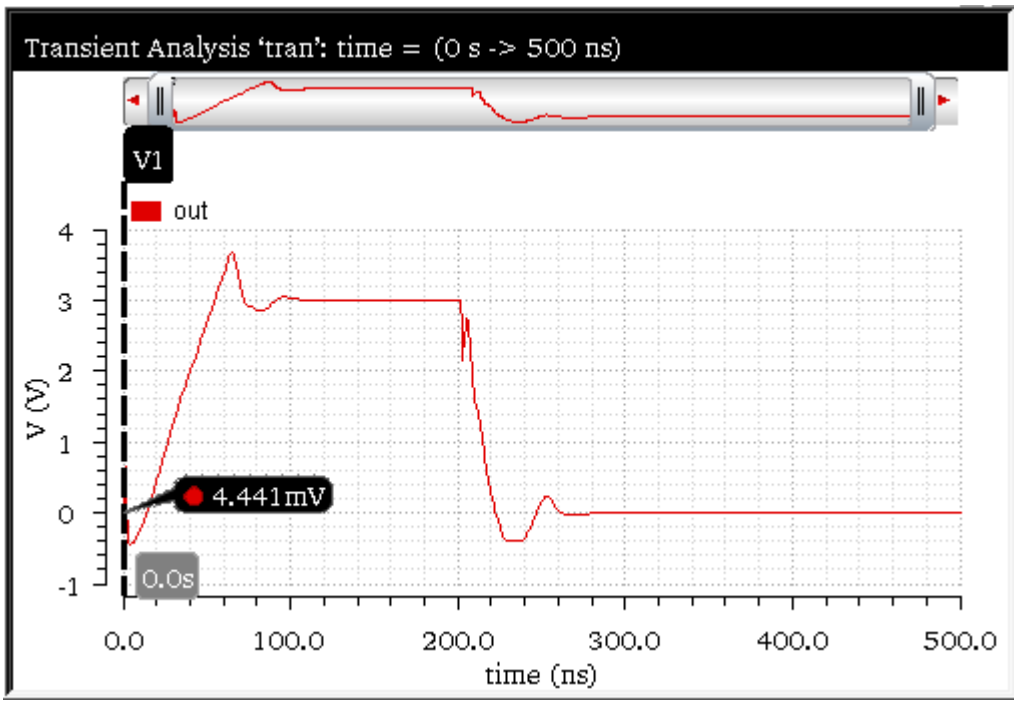
© 2023 Cadence Design Systems, Inc.

x**2

Returns the x^2 (second power) value of the input signal. This function is available only in the SKILL mode.

Example

Consider the following input signal from transient analysis:



A vertical marker `V1` is placed at time=`0.0ns` and voltage=`4.441mv`.

When you send this signal to Calculator, the following expression is created in the Buffer:

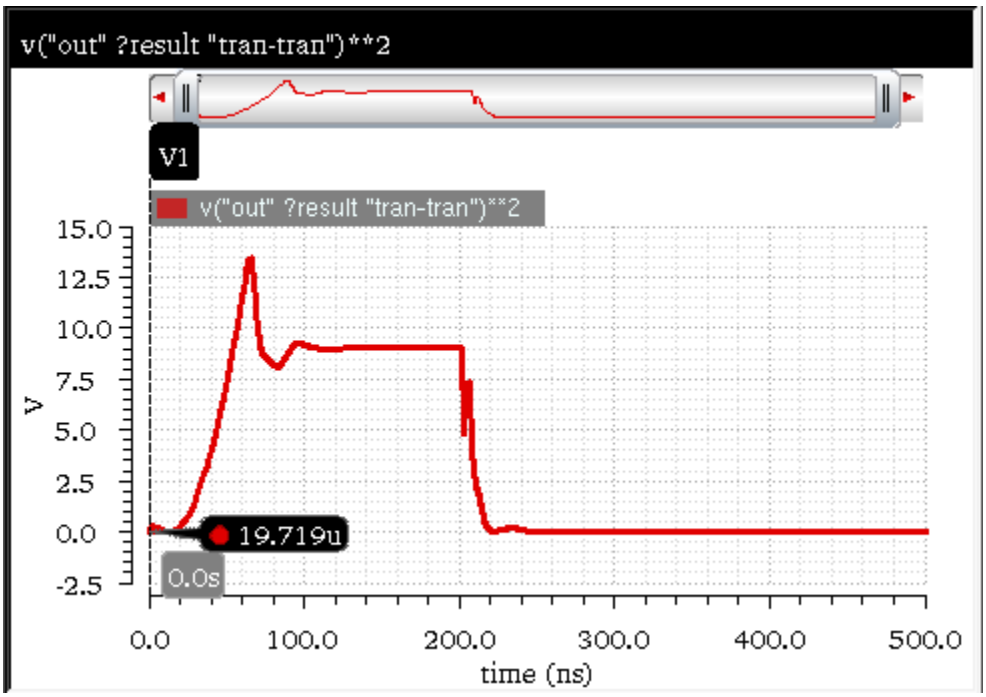
```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `x**2` function on this input signal, the function is directly applied on the signal expression in Buffer. The expression created in Buffer is as follows:

```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")**2
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the second power of the input waveform is calculated and the following output signal is displayed in a new window:

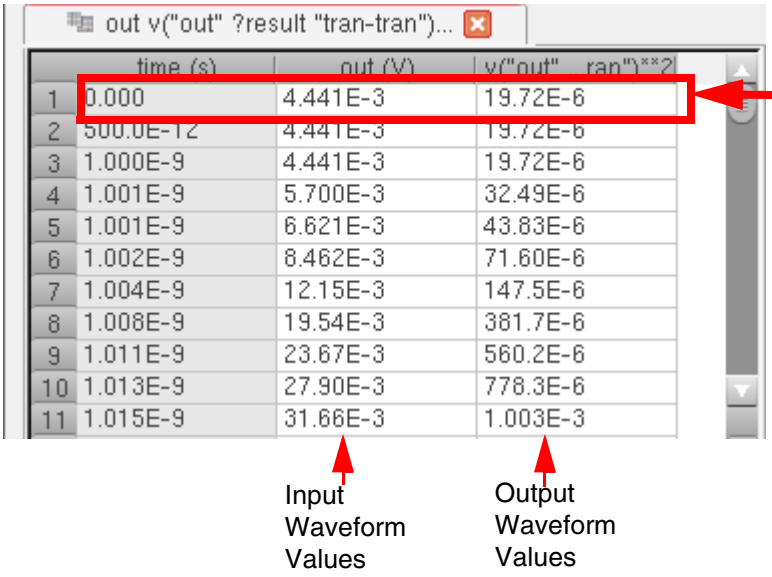


In the output waveform, the vertical maker `V1` shows the voltage=`19.71mV` when placed at time=`0.0ns`, which is the square root of the voltage value (`4.441E-3mV`) shown in the input waveform.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal, out(V), is 4.441E-3. After the sqrt function is applied, the voltage value at time=0.00ns in output signal is shown as 19.72E-6 (Refer to vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `x**2` is:

For more information, see `x**2` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

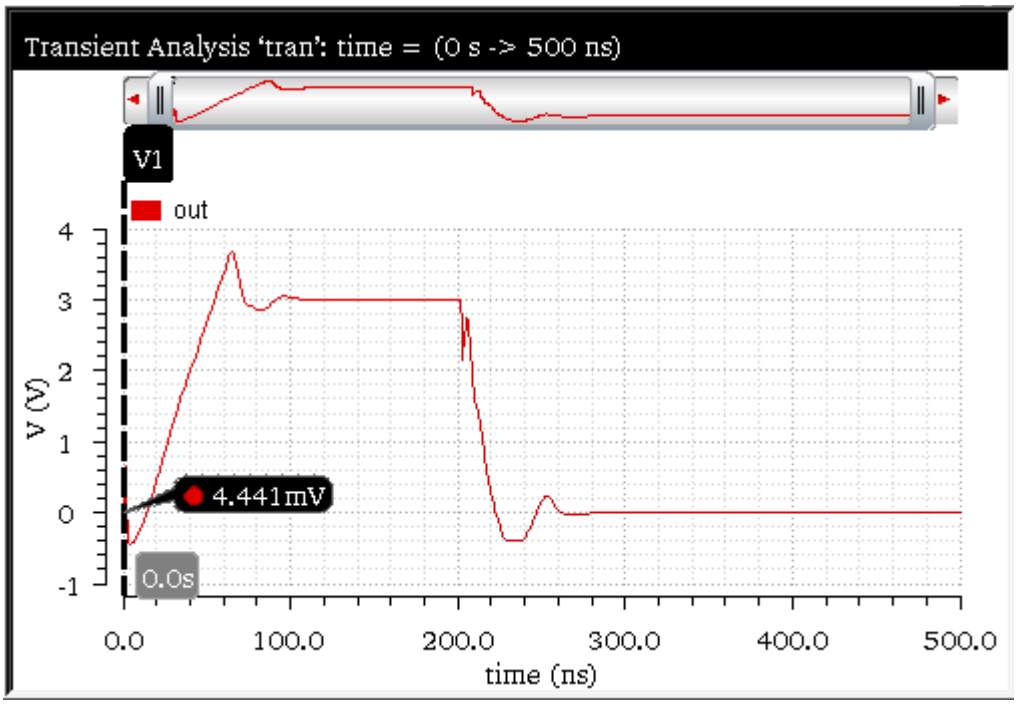
y**x

Returns the y^x (y to the power x) value. This function is available only in the SKILL mode.

Example

Consider the following two input signals from transient analysis. The first signal `v("out" ?result "tran-tran")` denotes the y value whose power is to be calculated and second signal `v("net10" ?result "tran-tran")` denotes the power (x value in y^x).

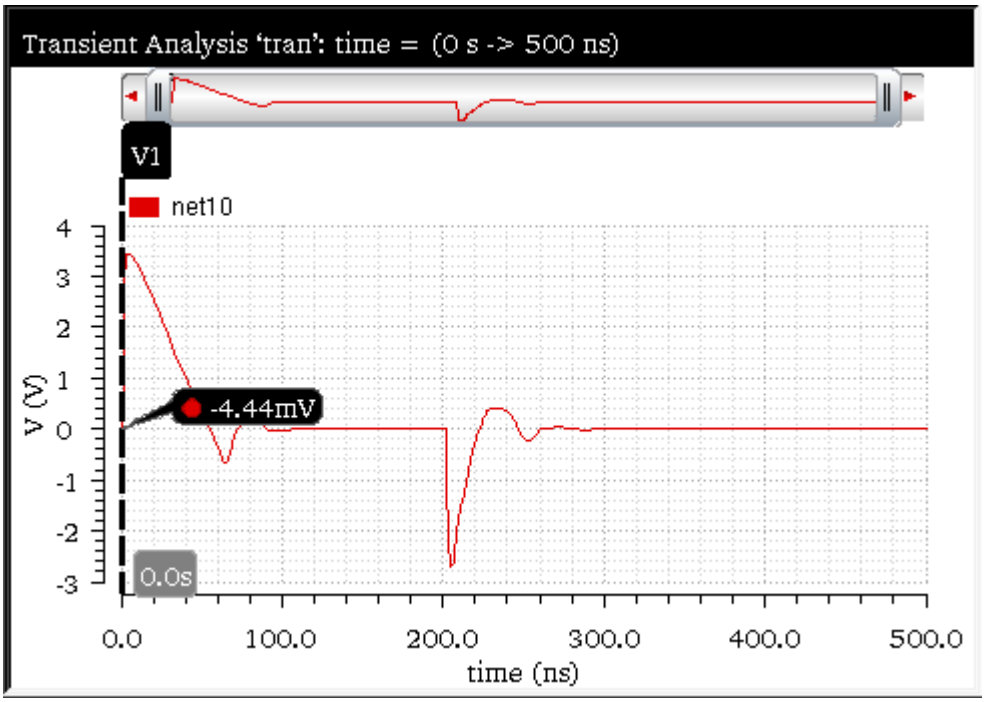
First Input Signal: `v("out" ?result "tran-tran")`



A vertical marker `V1` is placed on the first signal at time=`0.0ns` and voltage=`4.441mv`.

© 2023 Cadence Design Systems, Inc.

Second Input Signal: `v("net10" ?result "tran-tran")`



A vertical marker `V1` is placed on the second input signal at time=`0.0ns` and voltage=`-4.44mv`.

When you send the first input signal to Calculator, the following expression is created in the Buffer:

```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

To add second signal to the Calculator, you need to send the expression for the first input signal in the Stack. Now, when you send the second input signal to Calculator, the following expression is created in the Buffer:

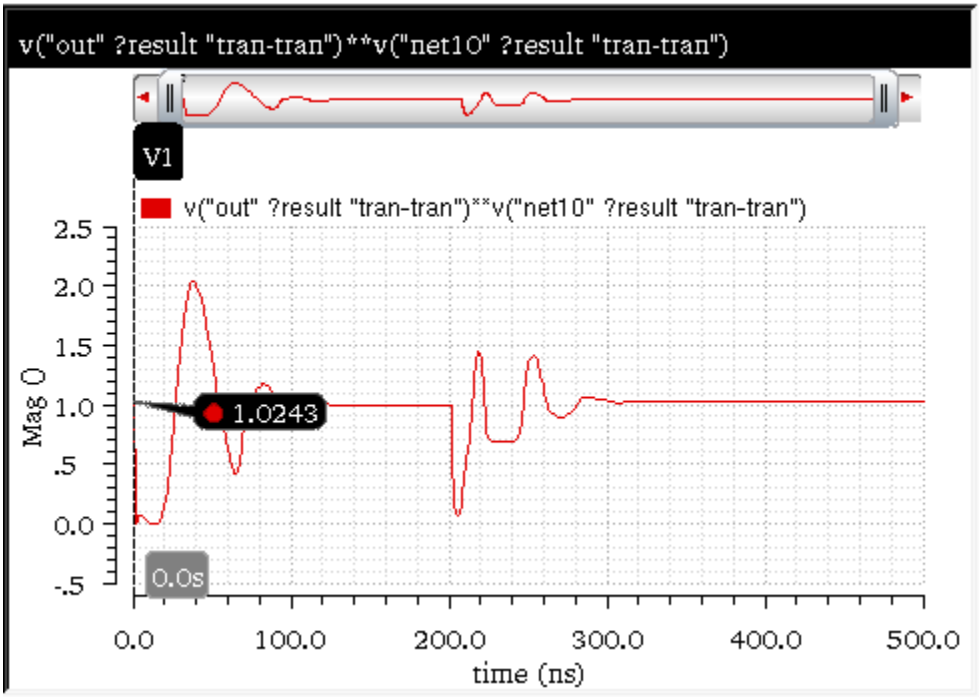
```
v("net10" ?result "tran-tran" ?resultsDir "./ampsim.raw")
```

When you apply the `y**x` function on this input signal, the function pops out the first signal expression from the Stack and uses the second signal expression from the Buffer. The new expression created in Buffer is as follows:

```
v("out" ?result "tran-tran" ?resultsDir "./ampsim.raw")**v("net10"
?result "tran-tran" ?resultsDir "./ampsim.raw")
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the second power of the input waveform is calculated and the following output signal is displayed in a new window:

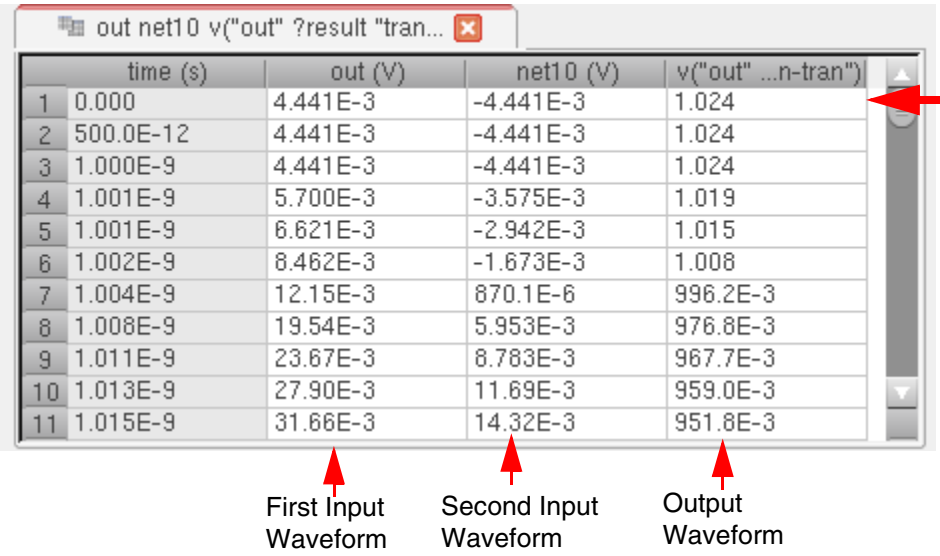


In the output waveform, the vertical maker `V1` shows the voltage=`1.0243mV` when placed at time=`0.0ns`, which is the (`4.441E-3`)th power of the voltage value `-4.44E-3` as marked by vertical markers in the input waveforms.

To analyze the generated output values, you can send the input and output signals to the Virtuoso Visualization and Analysis XL Table. To compare the input and output values, it is required to display both the signals in the same table, side by side. Therefore, to send the output signal to the table, right-click the output signal and choose *Send To – Table – Append*. The output signal is appended to the existing table.

© 2023 Cadence Design Systems, Inc.

The following table contents are displayed when you send the input and output signals described in this example to the table:



At time=0.00ns, the voltage value in the input signal1(x), out(V), is 4.441E-3 and input signal2 (y) is -4.441E-3. After the y**x function is applied, the resulted voltage value at time=0.00ns in output signal is shown as 1.024 (Refer to vertical markers in the above figures for input and output waveforms)

Related OCEAN Function

The equivalent OCEAN command for `y**x` is:

For more information, see `y**x` in *OCEAN Reference*.

© 2023 Cadence Design Systems, Inc.

RF Functions

This section describes the following RF functions available in the Virtuoso Visualization and Analysis XL Calculator:

- Rn
- B1f
- gac_freq
- gac_gain
- Gmax
- Gmin
- Gmsg
- GP
- gpc_freq
- gpc_gain
- GT
- Gumx
- Kf
- loadStability
- nc_freq
- nc_gain
- NF
- NFmin
- rn
- s11
- s12
- s21
- s22
- sourceStability

© 2023 Cadence Design Systems, Inc.

How to Run Examples?

The examples explained in these functions are created on the S-parameter analysis of the Low Noise Amplifier (LNA) design. Follow the following steps to run the examples:

1. Download and untar the test case `rfworkshop.tar.Z` from the latest MMSIM hierarchy saved at the following location: `./tools.lnx86/spectre/examples/SpectreRF_workshop/rfworkshop.tar.Z`
2. Run Virtuoso and in the Library Manager, choose library as `Rfworkshop`, cell as `Diff_LNA_test`, and view as `Lab1_sp`.
3. Run the simulation.
4. Choose *Tools – Calculator* to open the Virtuoso Visualization and Analysis XL Calculator.
5. Click the function you want to apply in the Function Panel under the RF functions category. The related expression is displayed in the Buffer.
6. Click the Evaluate button to generate the output plot. The output waveform is displayed in the graph window.

   Alternatively, you can evaluate these functions by performing the following steps:

   a. After step 4, send the Buffer expression to ADE outputs by choosing *Tools – Send Buffer to ADE Outputs*.

   b. The expression is displayed in the ADE L Outputs section. Ensure that the *Plot* check box is selected and run the simulation. After the simulation is complete and results are generated, the output waveform is displayed in the Virtuoso Visualization and Analysis XL graph window.

Note the following:

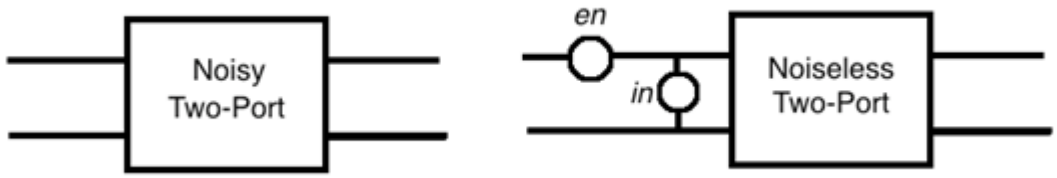
- For some functions, the output plot is a circular graph (Smith chart). Therefore, ensure that the *Impedance* plot type is selected in the Selection toolbar in the Calculator for these functions.
- For some functions, you need to convert the Y-axis into dB units. Therefore, `dB10` or `dB20` functions are applied on these functions.

© 2023 Cadence Design Systems, Inc.

Rn

Returns the equivalent noise resistance as a function of frequency. This function is available only on a 2-port network from that uses a S-Parameter analysis.

Consider a two-port network shown below. A noisy network can be converted into a noiseless network by connecting equivalent noise sources with the port.



In this case, `Rn` is defined as:

$$R_n = \frac{\overline{e_n^2}}{4kT\Delta f}$$

Here,

- `k` = Boltzmann constant = `1.38*10-23 J/K`
- `T` = Absolute temperature, degrees or Kelvin.
- `df` = System's bandwidth

Example

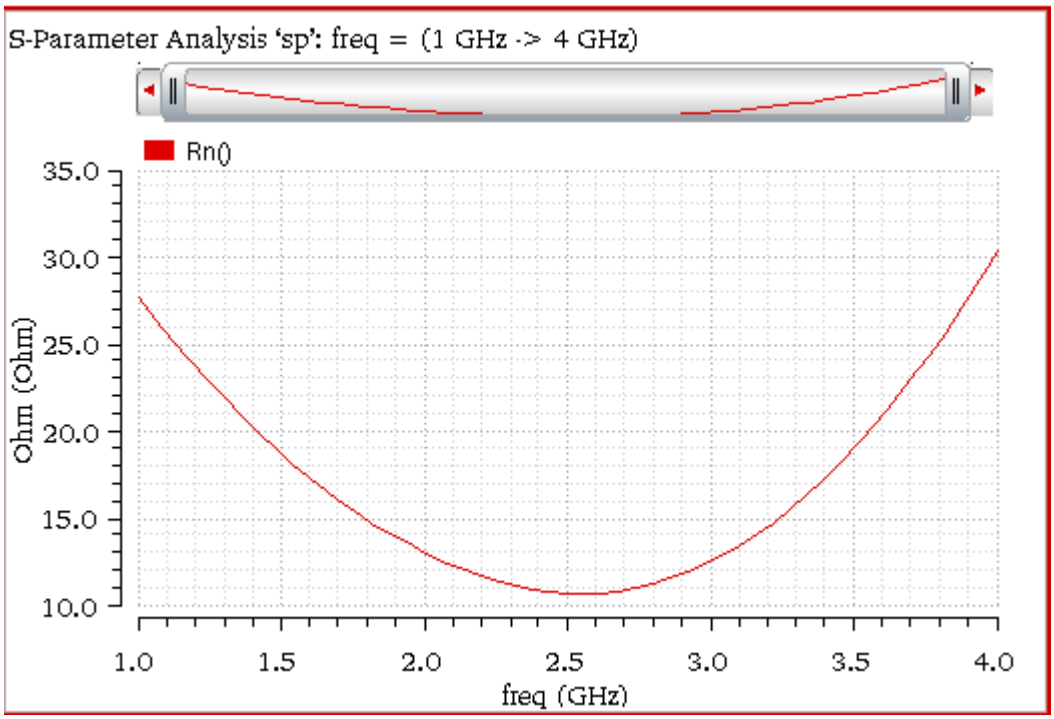
This example applies the `Rn` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `Rn` function in the Function Panel, the following expression is displayed in the Buffer:

```
Rn()
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

B1f

Returns the alternative stability factor.

`Kf` and `B1f` plot the Rollet stability factor and its intermediate term. These parameters can be used only for two-port circuits.

$$Kf = \frac{1 - |S_{11}|^2 - |S_{22}|^2 + |\Delta|^2}{2|S_{12}S_{21}|}$$

Here,

$$\Delta = S_{11}S_{22} - S_{12}S_{21}$$

$$B_{1f} = 1 + |S_{11}|^2 - |S_{22}|^2 - |\Delta|^2$$

Note: When `Kf>1` and `B1f>0`, the circuit becomes unconditionally stable.

Example

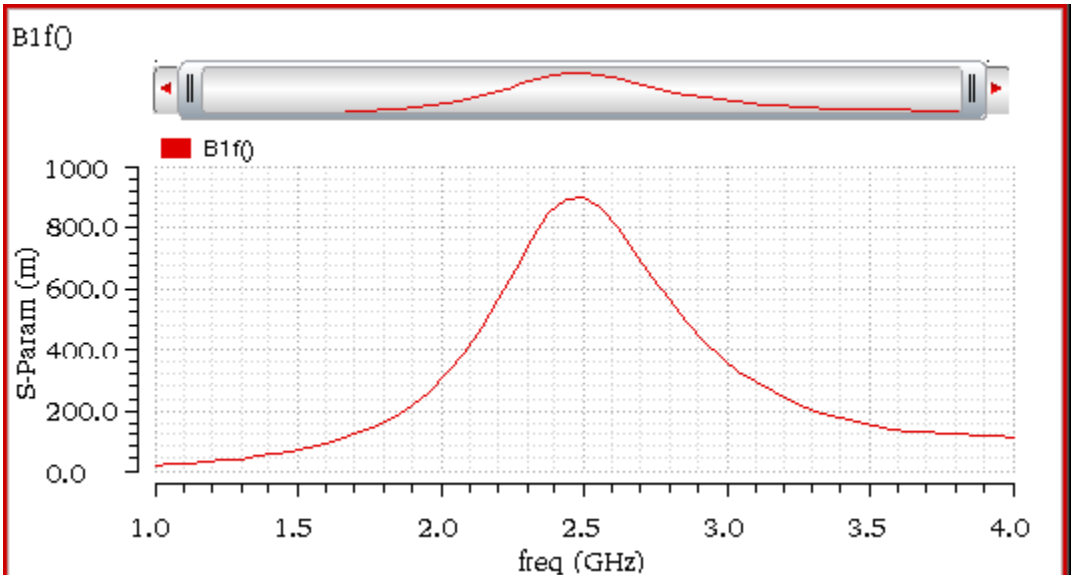
This example applies the `B1f` function on Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `B1f` function in the Function Panel, the following expression is displayed in the Buffer:

```
B1f()
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

GA

Returns the available power gain.

The available power gain, `GA`, is defined as the ratio between the power available from the network and the power available from the source.

$$G_A = \frac{1-|\Gamma_S|^2}{|1-S_{11}\Gamma_S|^2}|S_{21}|^2\frac{1}{1-|\Gamma_{out}|^2}$$

When source and load have impedance `Z0`, the equations are as follows:

$$\Gamma_L = \Gamma_S = 0$$

$$G_A = |S_{21}|^2\frac{1}{1-|S_{22}|^2}$$

Example

This example applies the `ga` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `ga` function in the Function Panel, the following expression is displayed in the Buffer:

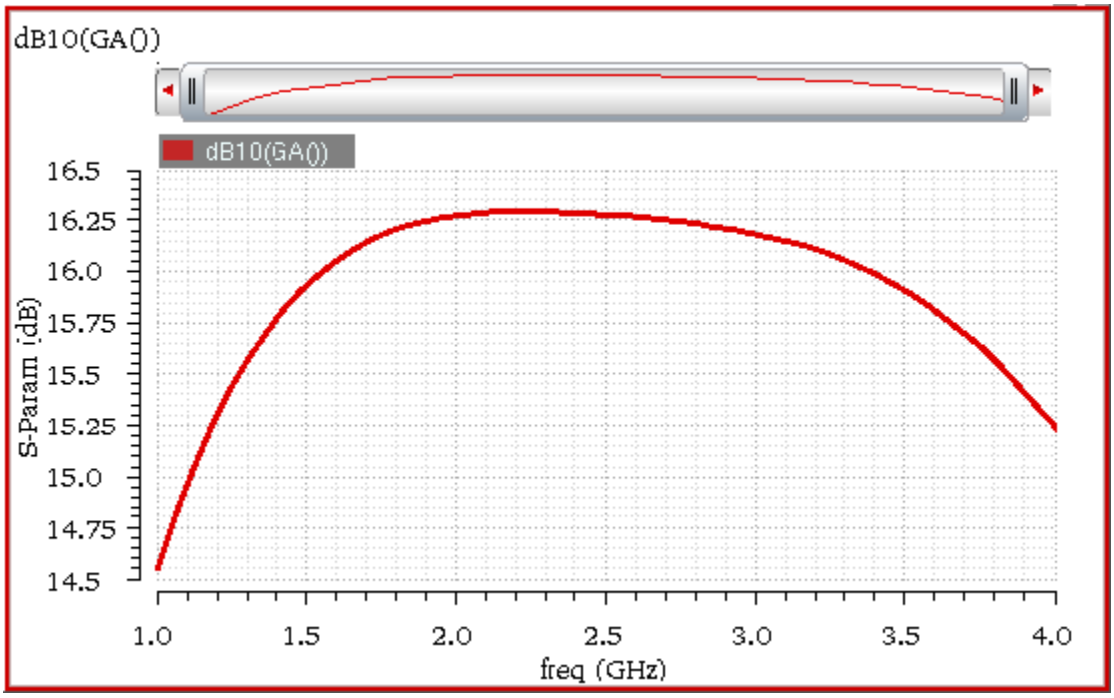
```
GA()
```

Now, convert this signal into dB by applying the `dB10` function on the above expression. The new expression in the Buffer is as follows:

```
dB10(GA())
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

gac_freq

Returns the available power gain circles where the gain is fixed and frequency is swept.

This function includes the following arguments:

- *Gain (dB)*—Gain value in dB.
- *Start*—Starting frequency.
- *Stop*—End frequency.
- *Step*—Frequency step size to be used.
- *Results Directory*—(Optional) Results directory path.

Note: The default value is `0` for all the fields.

Example

This example applies the `gac_freq` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

The following values for the function arguments has been specified in this example:

- *Gain (dB)*—`16`
- *Start*—`2G`
- *Stop*—`3G`
- *Step*—`100M`

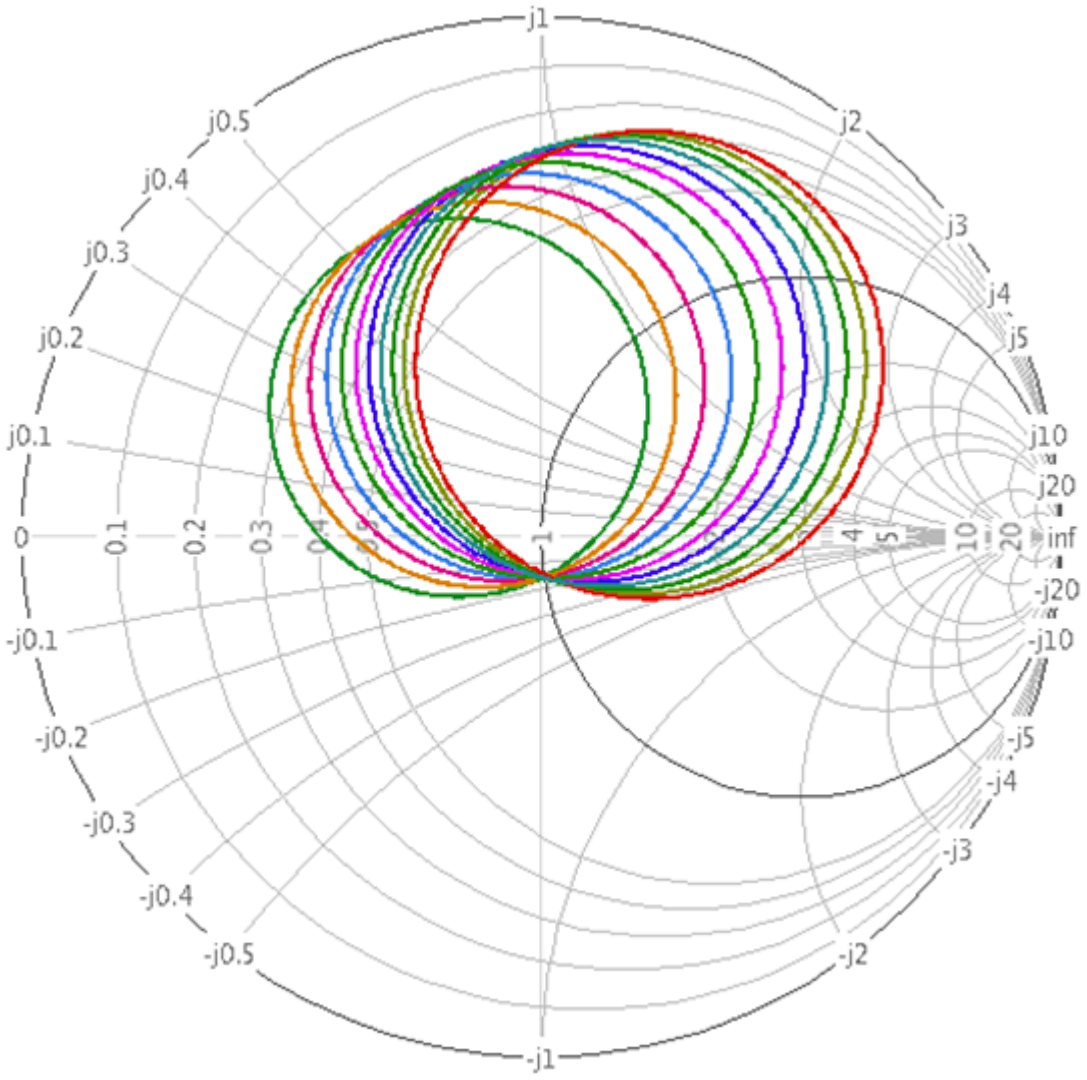
© 2023 Cadence Design Systems, Inc.

The corresponding expression created in the Buffer is as follows:

```
gac_freq(16 2G 3G 100M)
```

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

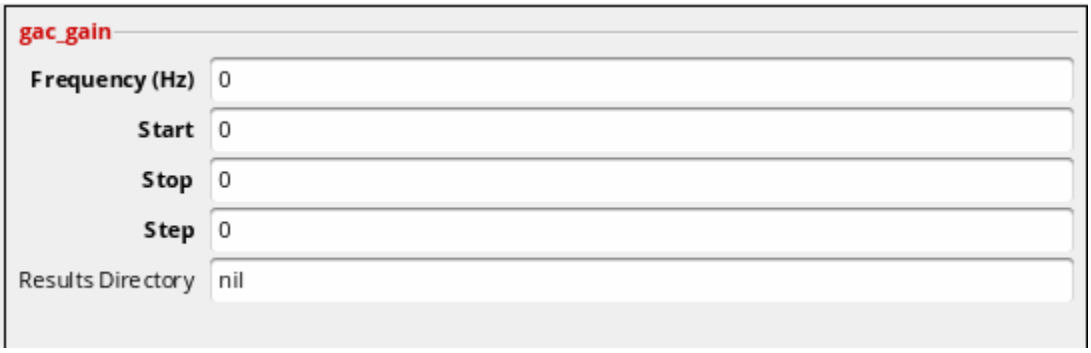
Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

gac_gain

Returns the available power gain circles where the frequency is fixed and gain is swept.



This function includes the following arguments:

- *Frequency (Hz)*—Frequency in Hertz.
- *Start*—Starting gain value
- *Stop*—End gain value
- *Step*—Gain step size to be used
- *Results Directory*—(Optional) Results directory path.

Note: The default value is `0` for all the fields.

Example

This example applies the `gac_gain` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?

The following values for the function arguments has been specified in this example:

- *Frequency (Hz)*—`2.4G`
- *Start*—`14`
- *Stop*—`18`
- *Step*—`0.5`

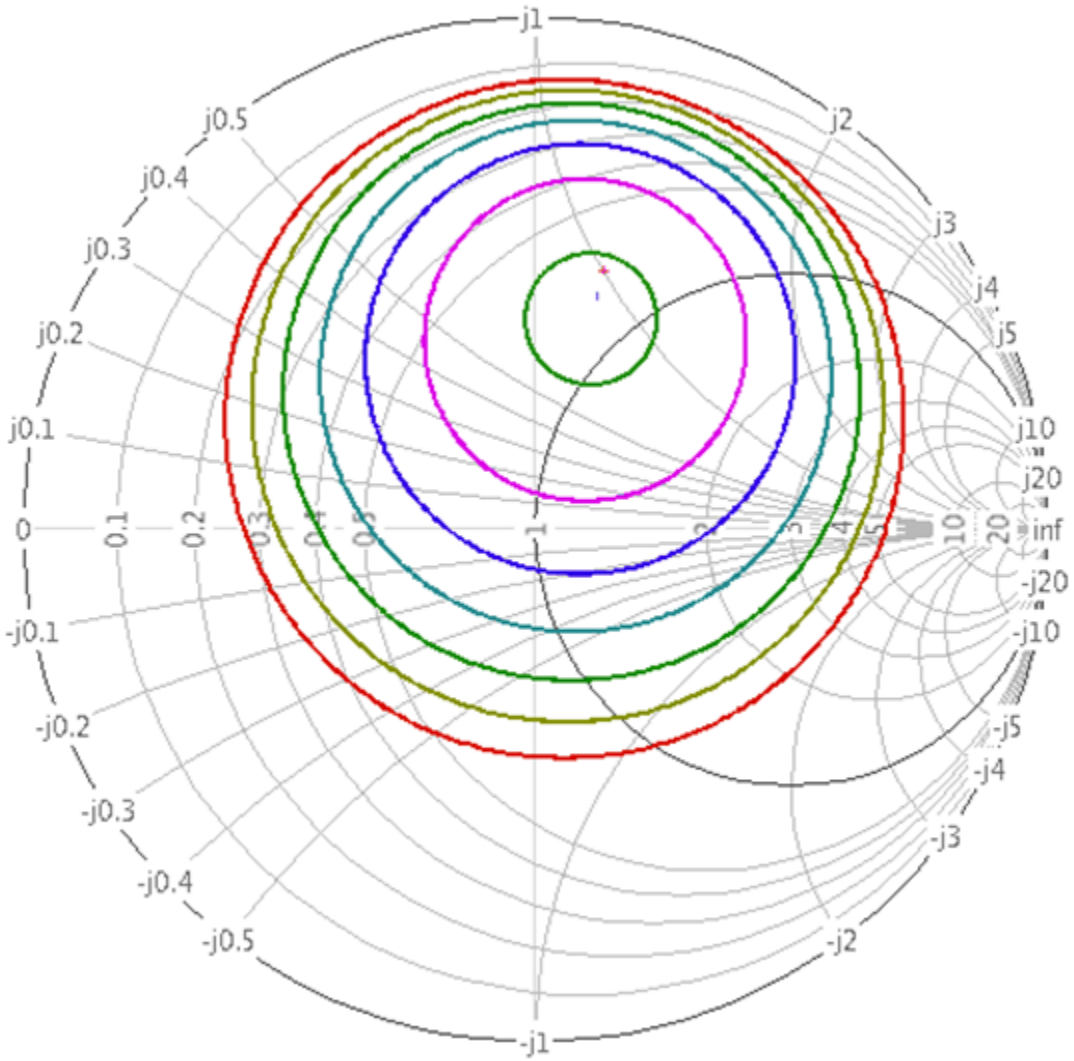
The corresponding expression created in the Buffer is as follows:

```
gac_gain(2.4G 14 18 0.5)
```


© 2023 Cadence Design Systems, Inc.

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

Gmax

Returns the maximum available gain for a two port.

The maximum transducer power gain, `Gmax`, is the simultaneous conjugate matching power gain when both the input and output are conjugately matched.

$$\Gamma_s = \Gamma_{in}$$

$$\Gamma_L = \Gamma_{out}$$

$$G_{max} = \frac{|S_{21}|}{|S_{12}|}\left(Kf - \sqrt{Kf^2 - 1}\right)$$

Here, `kf` is the stability factor. For more information, see Kf.

Example

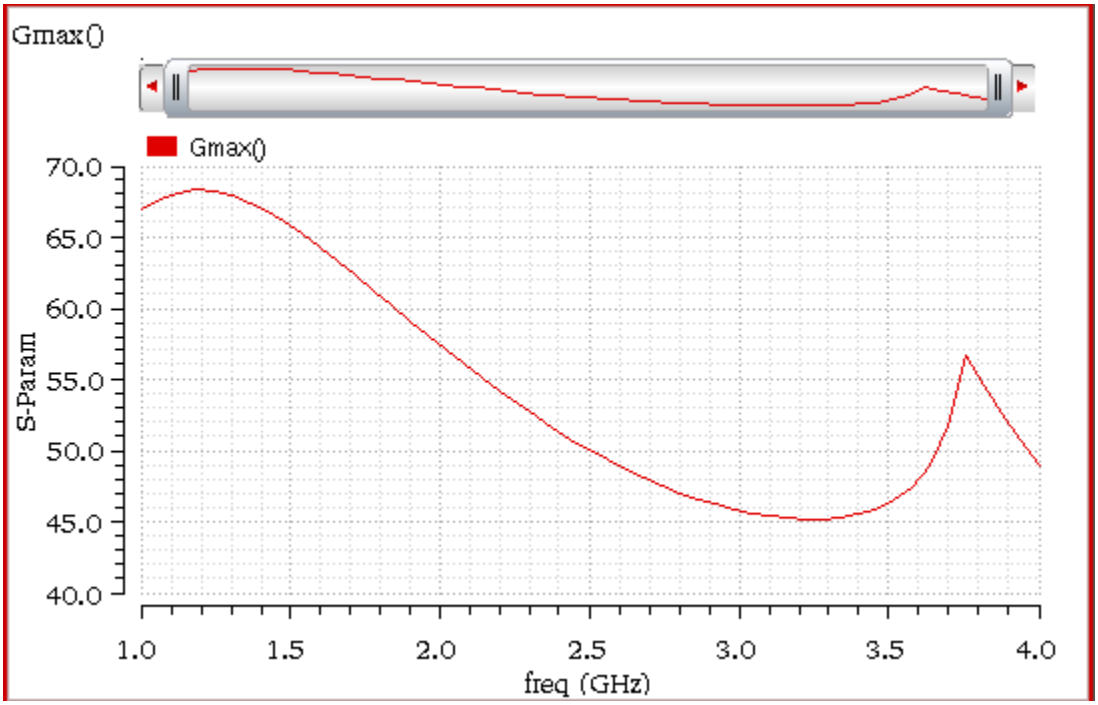
This example applies the `gmax` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `gmax` function in the Function Panel, the following expression is displayed in the Buffer:

```
Gmax()
```


© 2023 Cadence Design Systems, Inc.

When you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

Gmin

Returns the optimum noise reflection coefficient for NFmin.

Example

This example applies the `gmin` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

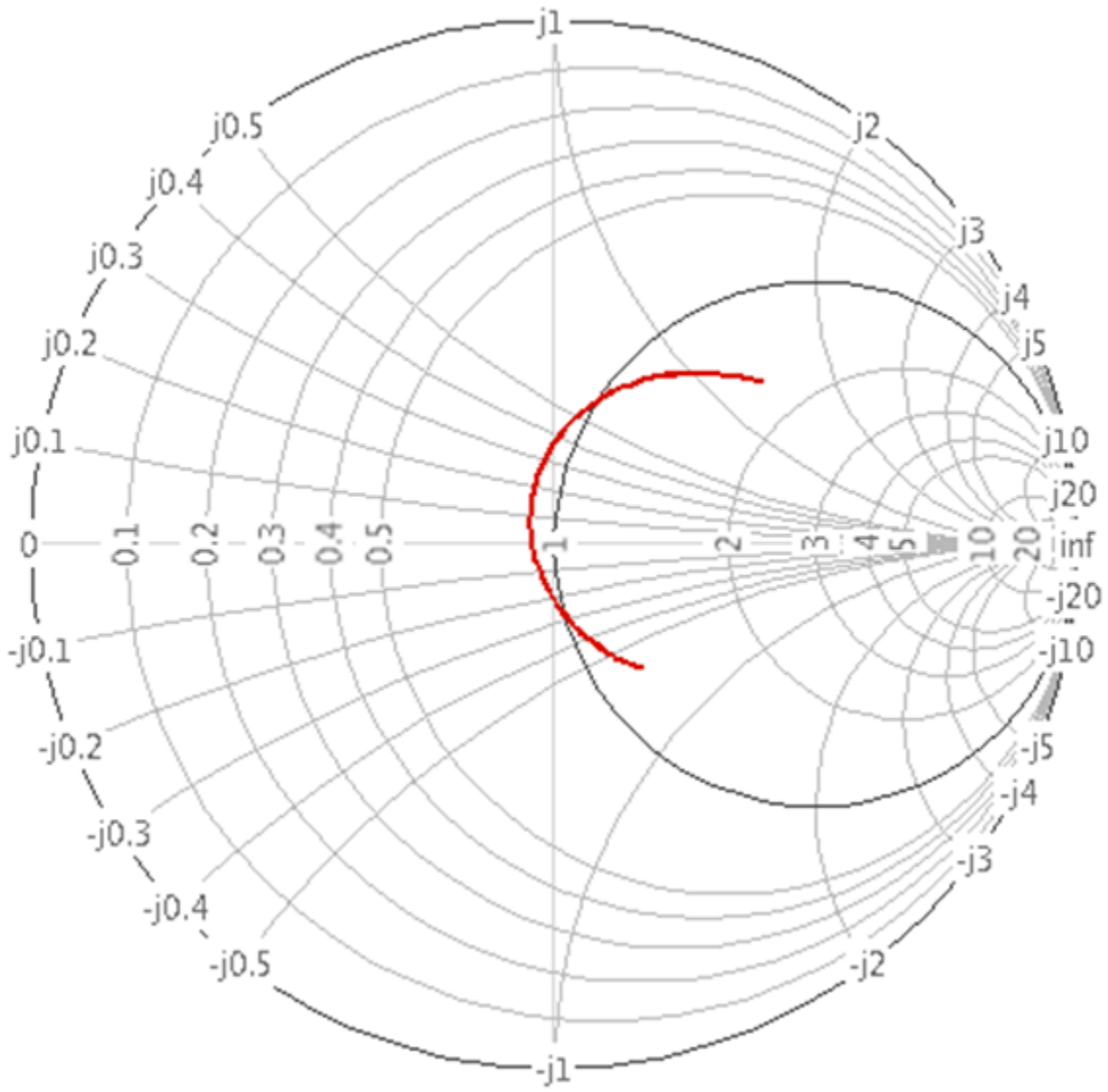
When you select the `gmin` function in the Function Panel, the following expression is displayed in the Buffer:

```
Gmin()
```

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

Gmsg

Returns the maximum stable power gain for a two port.

Maximum stability gain, `Gmsg`, is the maximum of `Gmax` when the stability condition, `Kf > 1`, is satisfied.

$$G_{msg} = \frac{|S_{21}|}{|S_{12}|}$$

Example

This example applies the `gmsg` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `gmsg` function in the Function Panel, the following expression is displayed in the Buffer:

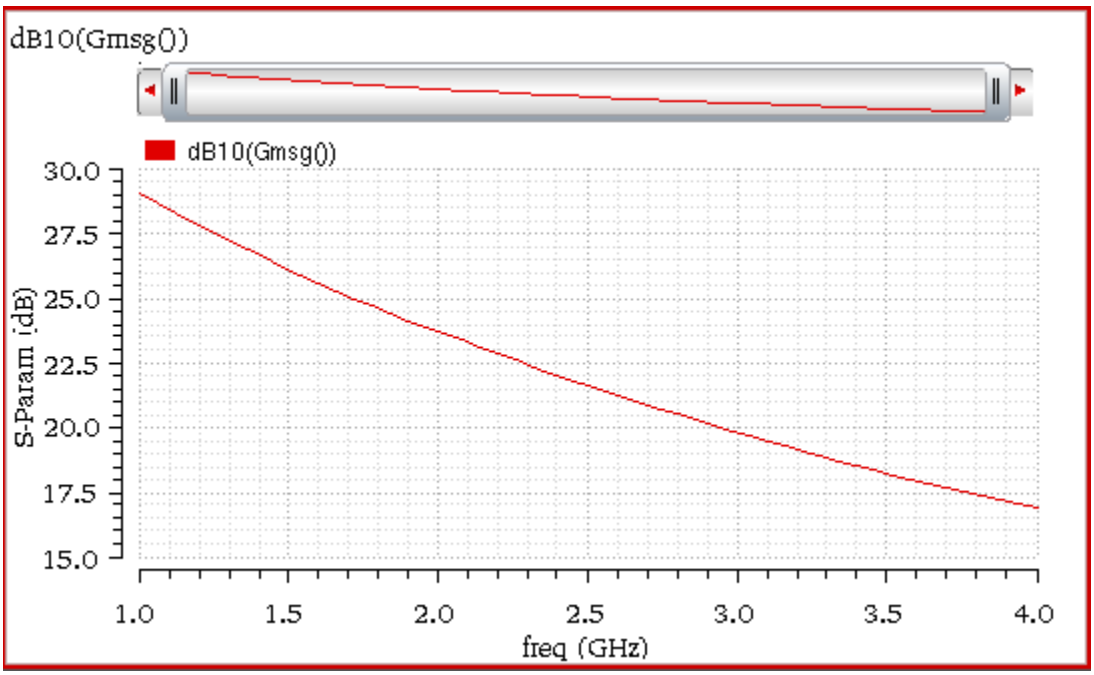
```
Gmsg()
```

Now, convert this signal into dB by applying the `dB10` function on the above expression. The new expression in the Buffer is as follows:

```
dB10(Gmsg())
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

GP

Returns the power gain.

Operating power gain, `GP`, is defined as the ratio between the power delivered to the load and the power input to the network.

$$G_P = \frac{|S_{21}|^2}{1-|\Gamma_{in}|^2}\frac{1-|\Gamma_L|^2}{|1-S_{22}\Gamma_L|^2}$$

When source and load have impedance `Z0`, the equations are as follows:

$$\Gamma_L = \Gamma_S = 0$$

$$G_P = \frac{|S_{21}|^2}{1-|S_{11}|^2}$$

Example

This example applies the `gp` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `gp` function in the Function Panel, the following expression is displayed in the Buffer:

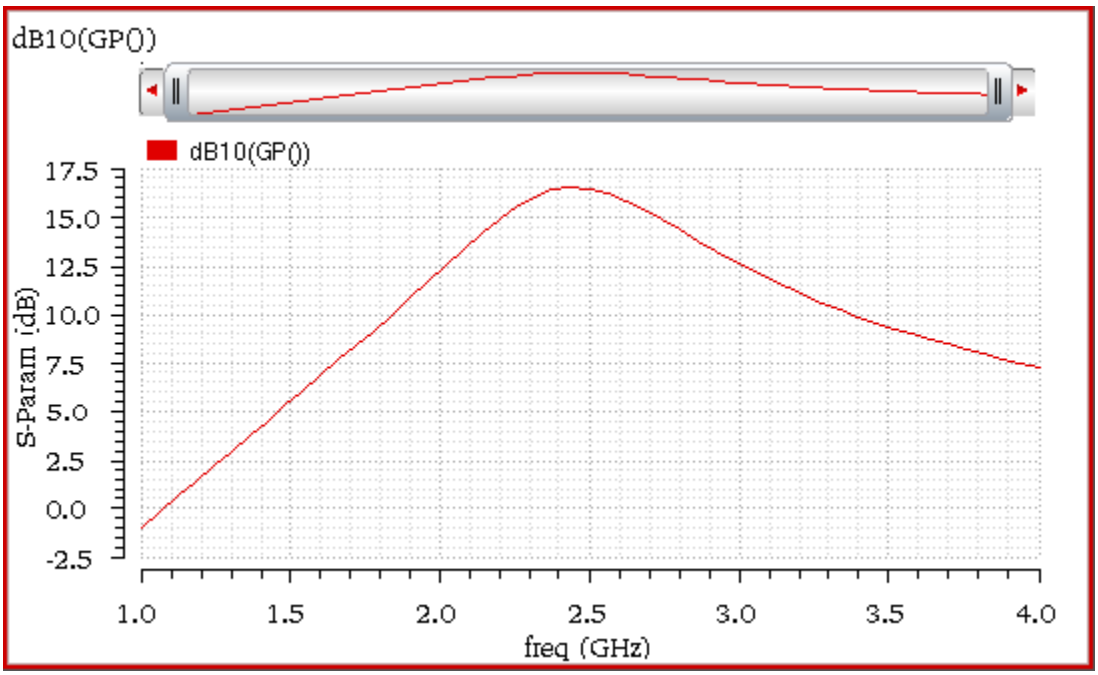
```
GP()
```

Now, convert this signal into dB by applying the `dB10` function on the above expression. The new expression in the Buffer is as follows:

```
dB10(GP())
```


© 2023 Cadence Design Systems, Inc.

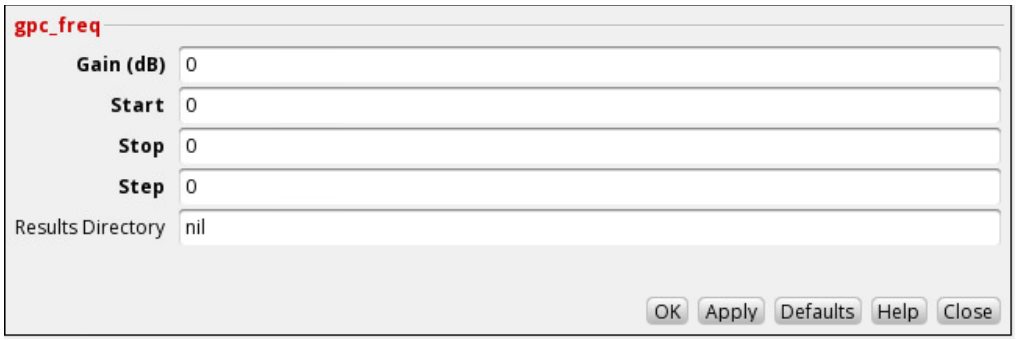
Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

gpc_freq

Returns the operating power gain circles where the gain is fixed and frequency is swept.



This function includes the following arguments:

- *Gain (dB)*—Gain value in dB.
- *Start*—Starting frequency.
- *Stop*—End frequency.
- *Step*—Frequency step size to be used.
- *Results Directory*—(Optional) Results directory path.

Note: The default value is `0` for all the fields.

Example

This example applies the `gpc_freq` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

The following values for the function arguments has been specified in this example:

- *Gain (dB)*—`16`
- *Start*—`2.4G`
- *Stop*—`2.5G`
- *Step*—`10M`

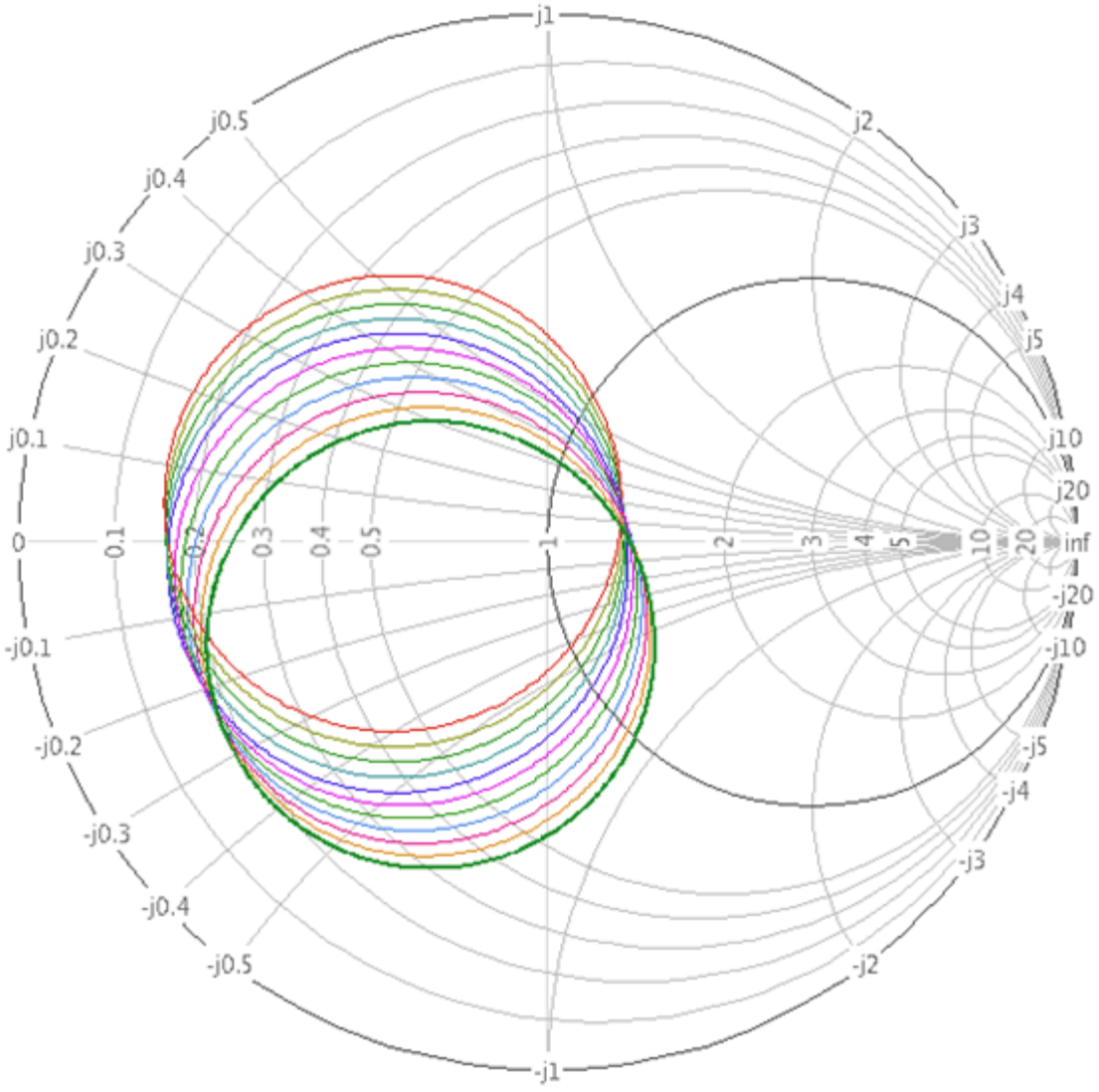
The corresponding expression created in the Buffer is as follows:

```
gpc_freq(16 2.4G 2.5G 10M)
```


© 2023 Cadence Design Systems, Inc.

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

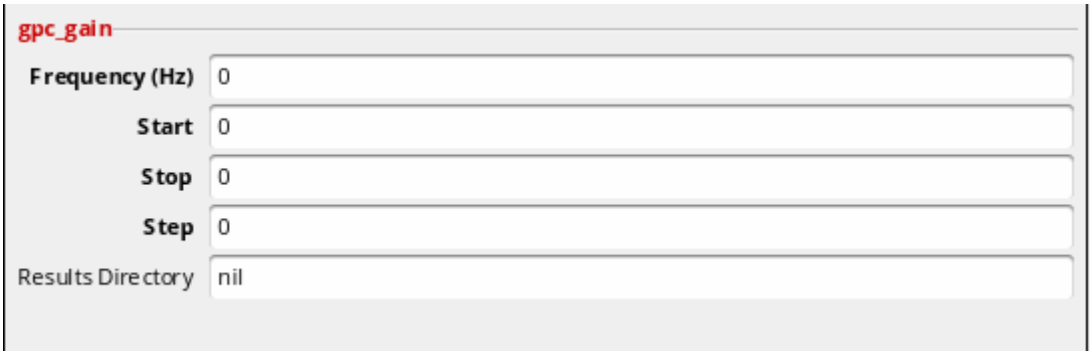
Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

gpc_gain

Returns the operating power gain circles where the frequency is fixed and gain is swept.



This function includes the following arguments:

- *Frequency (Hz)*—Frequency in Hertz.
- *Start*—Starting gain value.
- *Stop*—End gain value.
- *Step*—Gain step size to be used.
- *Results Directory*—(Optional) Results directory path.

Note: The default value is `0` for all the fields.

Example

This example applies the `gpc_gain` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

The following values for the function arguments has been specified in this example:

- *Frequency (Hz)*—`2.4G`
- *Start*—`14`
- *Stop*—`18`
- *Step*—`0.5`

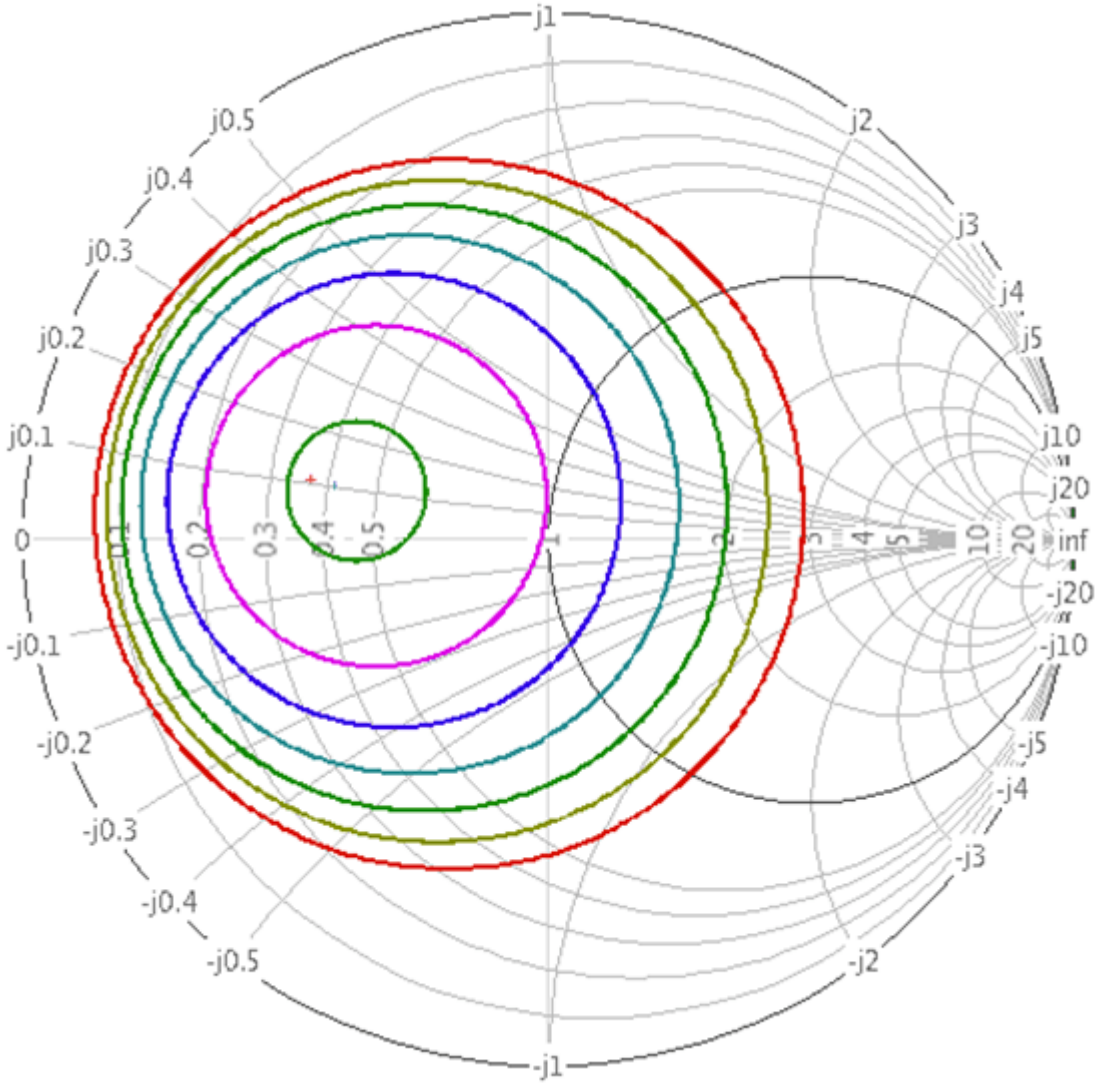
The corresponding expression created in the Buffer is as follows:

```
gpc_gain(2.4G 14 18 0.5)
```


© 2023 Cadence Design Systems, Inc.

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

GT

Returns the transducer gain.

Transducer power gain, `GT`, is defined as the ratio between the power delivered to the load and the power available from the source.

$$G_T = \frac{1-|\Gamma_S|^2}{|1-S_{11}\Gamma_S|^2}|S_{21}|^2\frac{1-|\Gamma_L|^2}{|1-S_{22}\Gamma_L|^2}$$

When source and load have impedance `Z0`, the equations are as follows:

$$\Gamma_L = \Gamma_S = 0$$

$$G_T = |S_{21}|^2$$

Example

This example applies the `gt` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `gt` function in the Function Panel, the following expression is displayed in the Buffer:

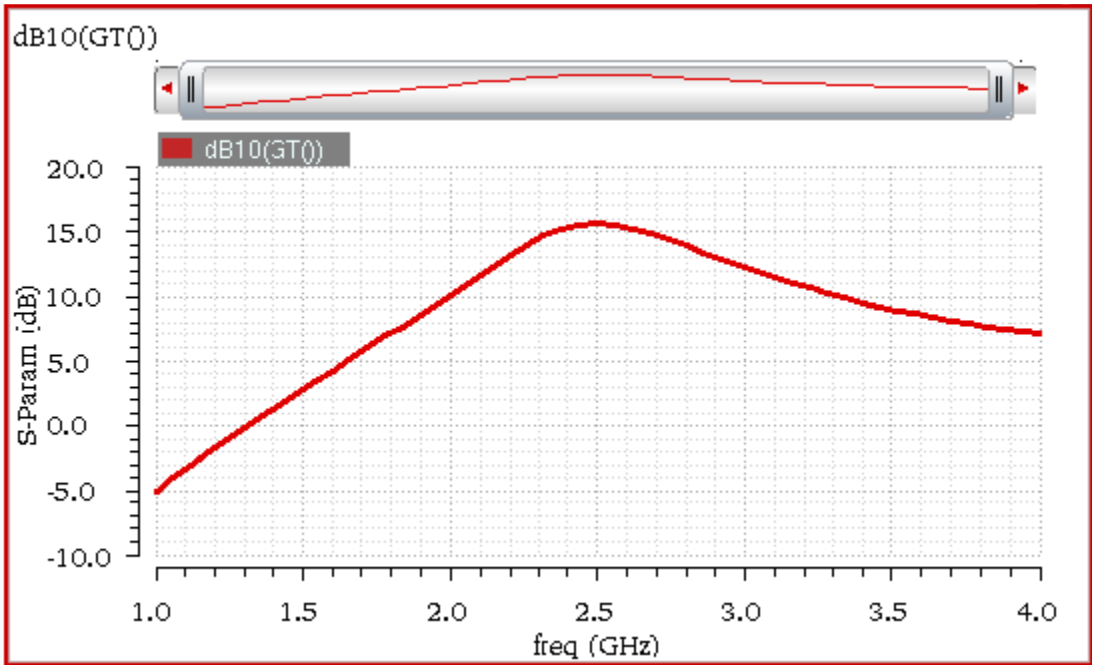
```
GT()
```

Now, convert this signal into dB by applying the `dB10` function on the above expression. The new expression in the Buffer is as follows:

```
dB10(GT())
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

Gumx

Returns the maximum unilateral power gain for a two port.

Maximum unilateral transducer power gain, `Gumx`, is the transducer power gain when `S12` is zero, and the source and load impedances conjugate are matching.

```
Gumx = |S21|^2/((1-|S11|^2)*(1-|S22|^2))
```

Example

This example applies the `gumx` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `gmux` function in the Function Panel, the following expression is displayed in the Buffer:

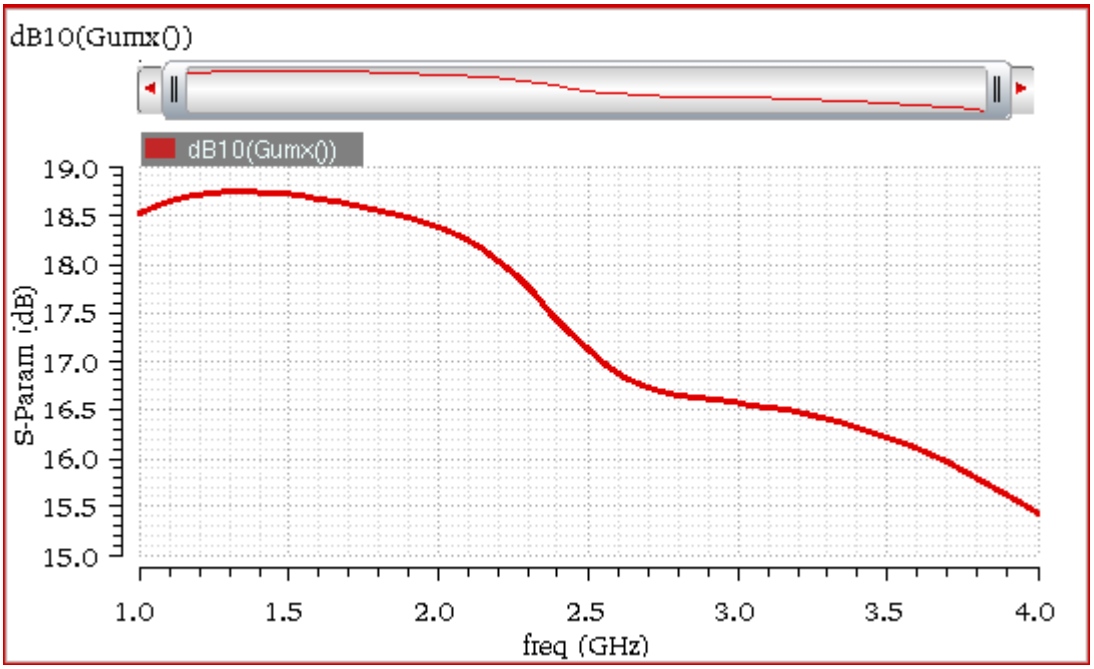
```
Gumx()
```

Now, convert this signal into dB by applying the `dB10` function on the above expression. The new expression in the Buffer is as follows:

```
dB10(Gumx())
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

Kf

Returns the Rollet stability factor.

The Rollet stability factor calculates circuit stability by using the following equation:

$$Kf = \frac{1 - \left|S_{11}\right|^2 - \left|S_{22}\right|^2 + |\Delta|^2}{2\left|S_{12}S_{21}\right|}$$

Here,

$$\Delta = S_{11}S_{22} - S_{12}S_{21}$$

When `Kf>1` and `B1f>0`, the circuit becomes unconditionally stable.

Example

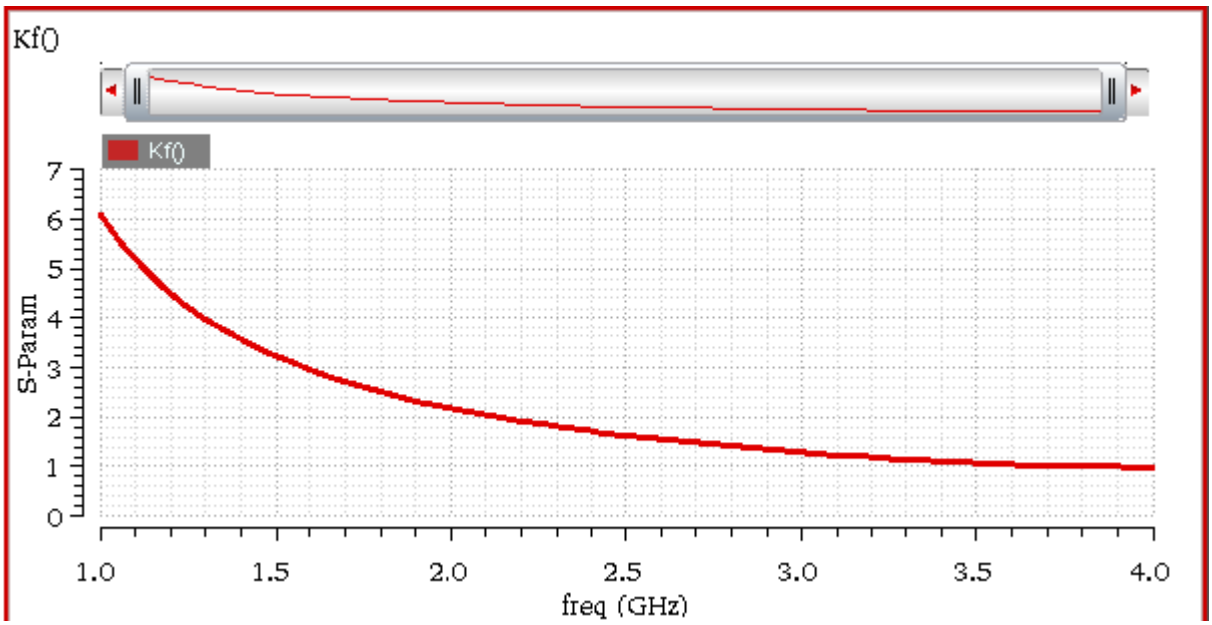
This example applies the `Kf` function on Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `Kf` function in the Function Panel, the following expression is displayed in the Buffer:

```
Kf()
```


© 2023 Cadence Design Systems, Inc.

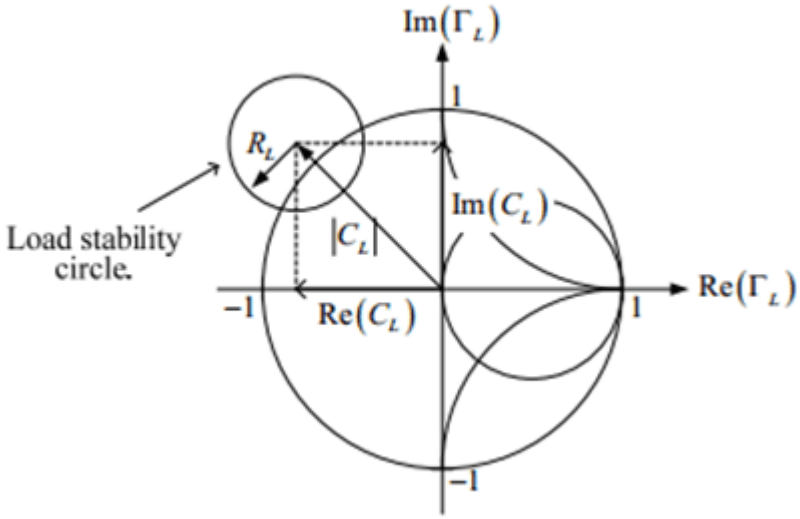
When you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

loadStability

Returns the load stability circles. Stability circles define the boundary between stable and potentially unstable Γ_L or Γ_S. For a conditionally stable circuit, the load stability on an extended Smith chart is displayed in the figure below:



This function includes the following arguments:

- *Start (Hz)*—Start of the frequency range.
- *Stop (Hz)*—End of the frequency range.
- *Step*—Frequency step size to be used.
- *Results Directory*—Results directory path.

© 2023 Cadence Design Systems, Inc.

Note: The default value is `0` for all the fields.

Example

This example applies the `loadStability` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

The following values for the function arguments has been specified in this example:

- *Start (Hz)*—`2G`
- *Stop (Hz)*—`3G`
- *Step*—`0.2G`

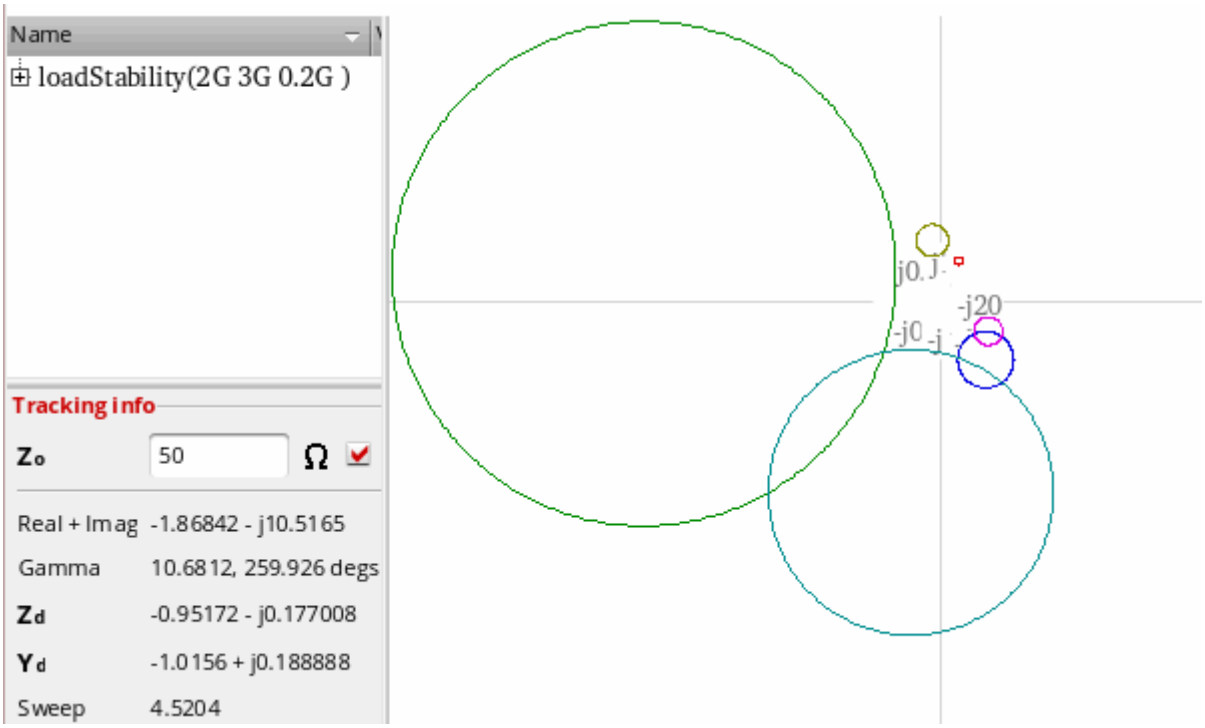
The corresponding expression created in the Buffer is as follows:

```
loadStability( sp(1 1 ?result "sp" ?resultsDir "./psf") sp(1 2
?result "sp" ?resultsDir "./psf") sp(2 1 ?result "sp" ?resultsDir "./
psf") sp(2 2 ?result "sp" ?resultsDir "./psf") '(2G 3G 0.2G) )
```

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

© 2023 Cadence Design Systems, Inc.

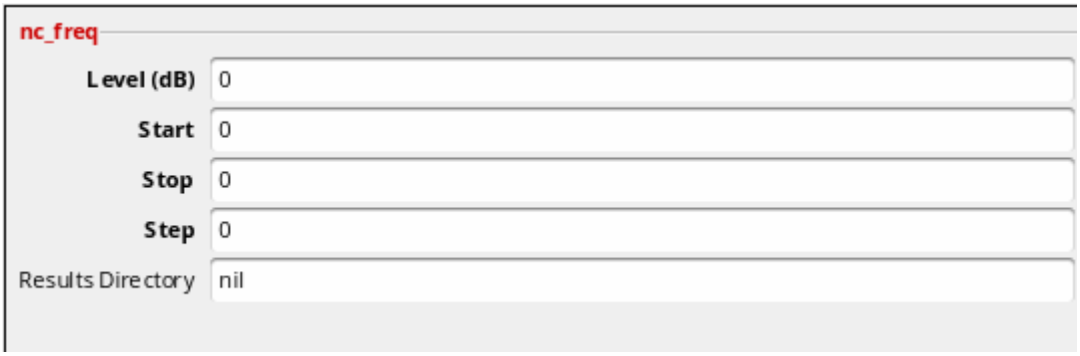
Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

nc_freq

Returns noise circles with fixed gain.



This function includes the following arguments:

- Level (dB)—Noise level.
- Start—Starting frequency
- Stop—End frequency
- Step—Frequency step size
- *Results Directory*—(Optional) Results directory path.

Note: The default value is `0` for all the fields.

Example

This example applies the `nc_freq` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

The following values for the function arguments has been specified in this example:

- *Level (dB)*—`2`
- *Start (Hz)*—`2G`
- *Stop (Hz)*—`3G`
- *Step*—`0.2G`

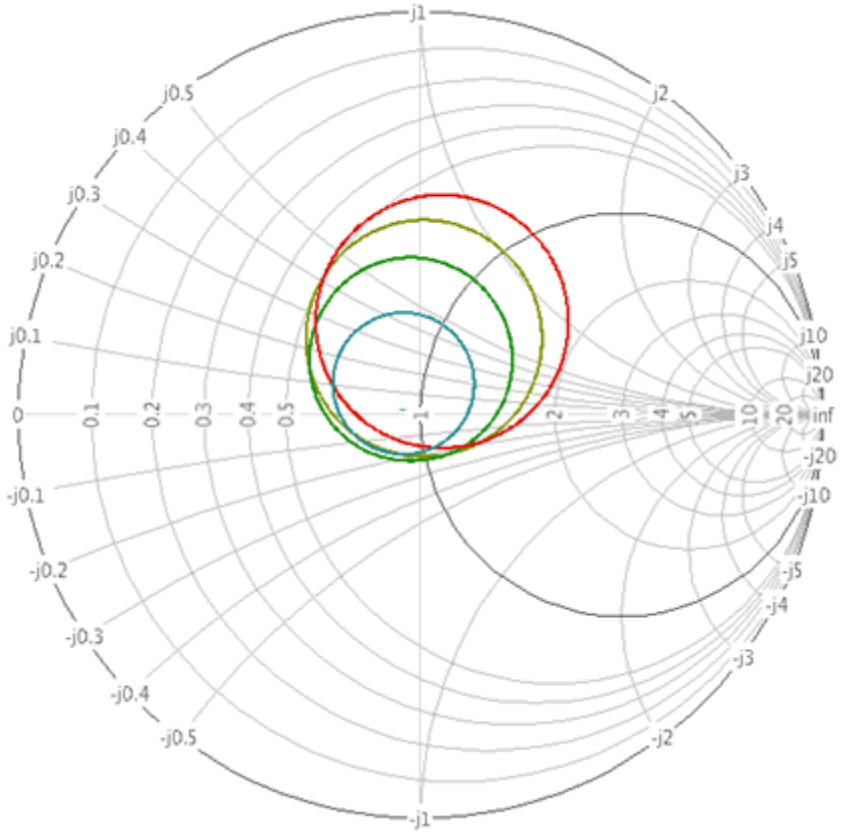
The corresponding expression created in the Buffer is as follows:

```
nc_freq(2 2G 3G 0.2G)
```


© 2023 Cadence Design Systems, Inc.

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

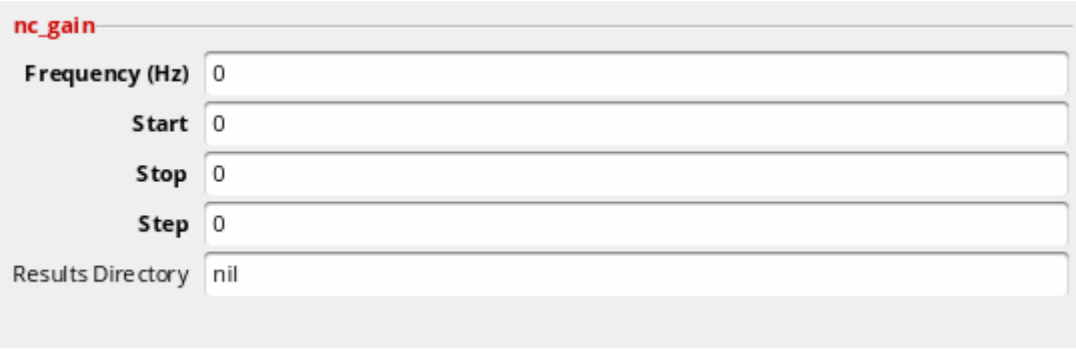
Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

nc_gain

Returns noise circles with fixed frequency.



This function includes the following arguments:

- *Frequency (Hz)*—Fixed frequency.
- *Start*—Starting noise level.
- *Stop*—End noise level.
- *Step*—Sweeping noise level step size
- *Results Directory*—(Optional) Results directory path.

Note: The default value is `0` for all the fields.

Example

This example applies the `nc_gain` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

The following values for the function arguments has been specified in this example:

- *Frequency (Hz)*—`2.4G`
- *Start (Hz)*—`1.5`
- *Stop (Hz)*—`2.4`
- *Step*—`0.2`

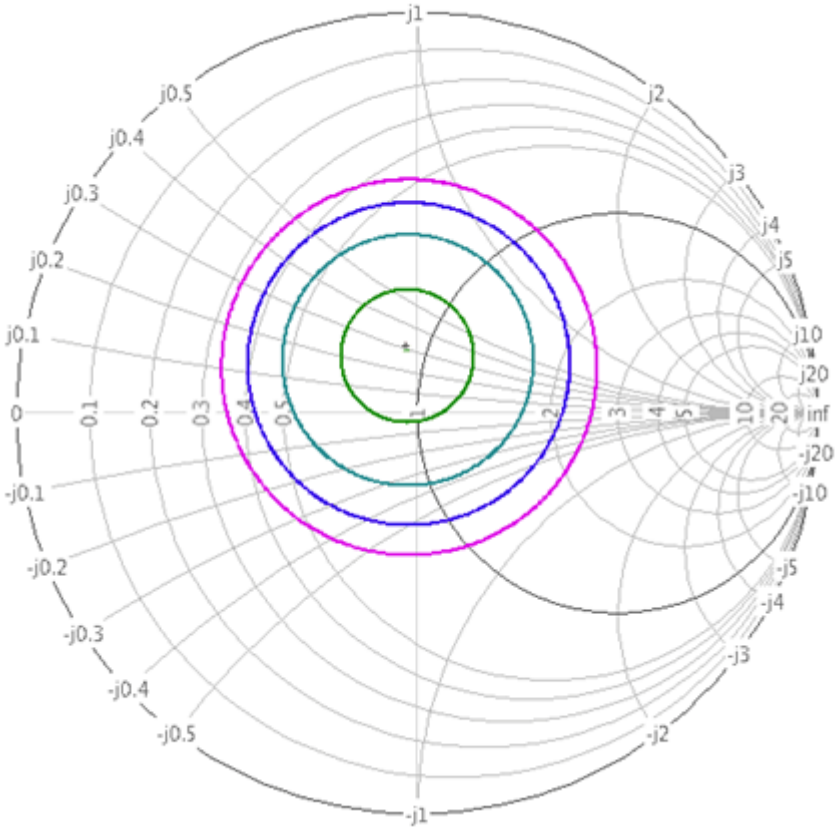
The corresponding expression created in the Buffer is as follows:

```
nc_gain(2.4G 1.5 2.4 0.2)
```


© 2023 Cadence Design Systems, Inc.

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

NF

Returns the noise figure.

```
NF=dB10(F)
```

Here, `NF` is the noise figure and `F` is the noise factor. Noise figure of a two-port is given by the ratio of total noise power, `PNO`, and the available power gain, `GA`, from the two-port multiplied with the input power `PNi`.

$$F = \frac{P_1 / P_{Ni}}{P_2 / P_{No}} = \frac{SNR_i}{SNR_0}$$

Example

This example applies the `nf` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design.

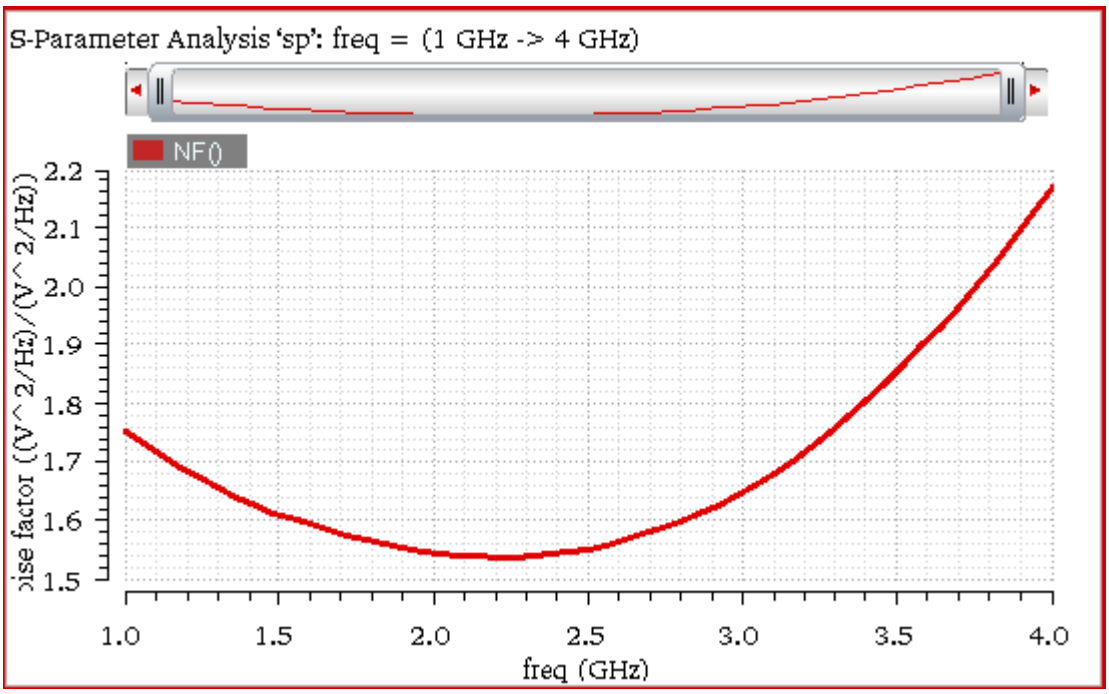
For detailed information, see How to Run Examples?.

When you select the `nf` function in the Function Panel, the following expression is displayed in the Buffer:

```
NF()
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

NFmin

Returns the minimum noise figure.

$$nF_{min} = dB10(F_{min})$$

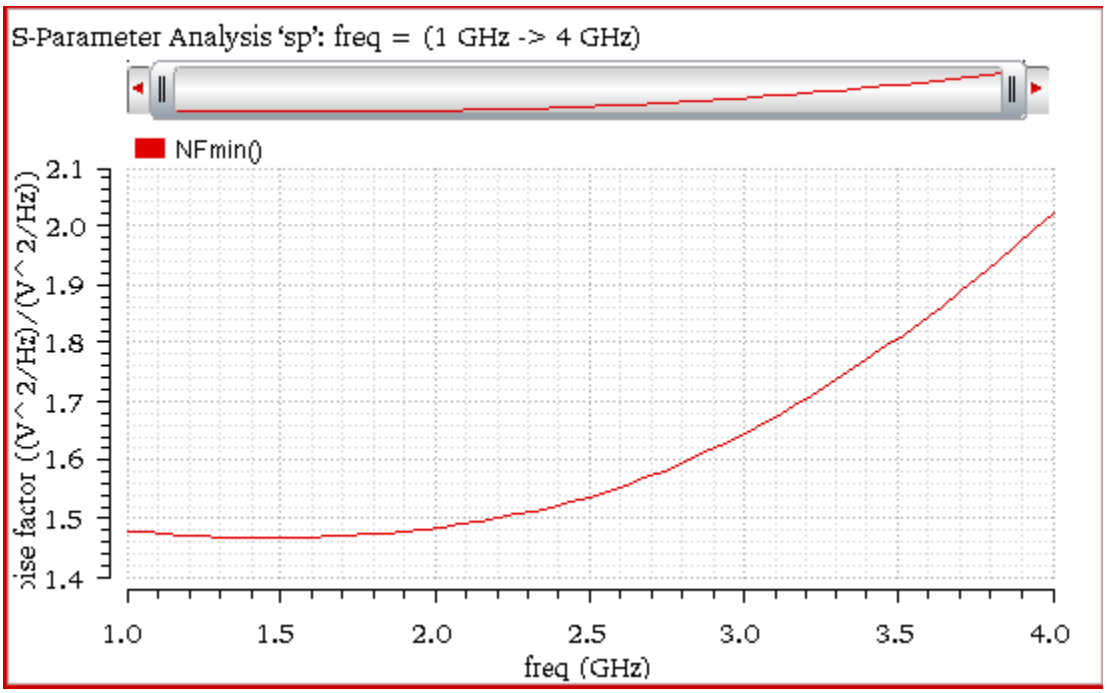
Example

This example applies the `nfmin` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `nfmin` function in the Function Panel, the following expression is displayed in the Buffer:

```
NFmin()
```

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

rn

Returns the normalized equivalent noise resistance as a function of frequency. This function is available only for a 2-port network from running an S-Parameter analysis.

$$r_n = \frac{R_n}{Z_0}$$

Here, `Rn` is the equivalent noise resistance and `Z0` is the source impedance. For more details about Rn function, see Rn.

Example

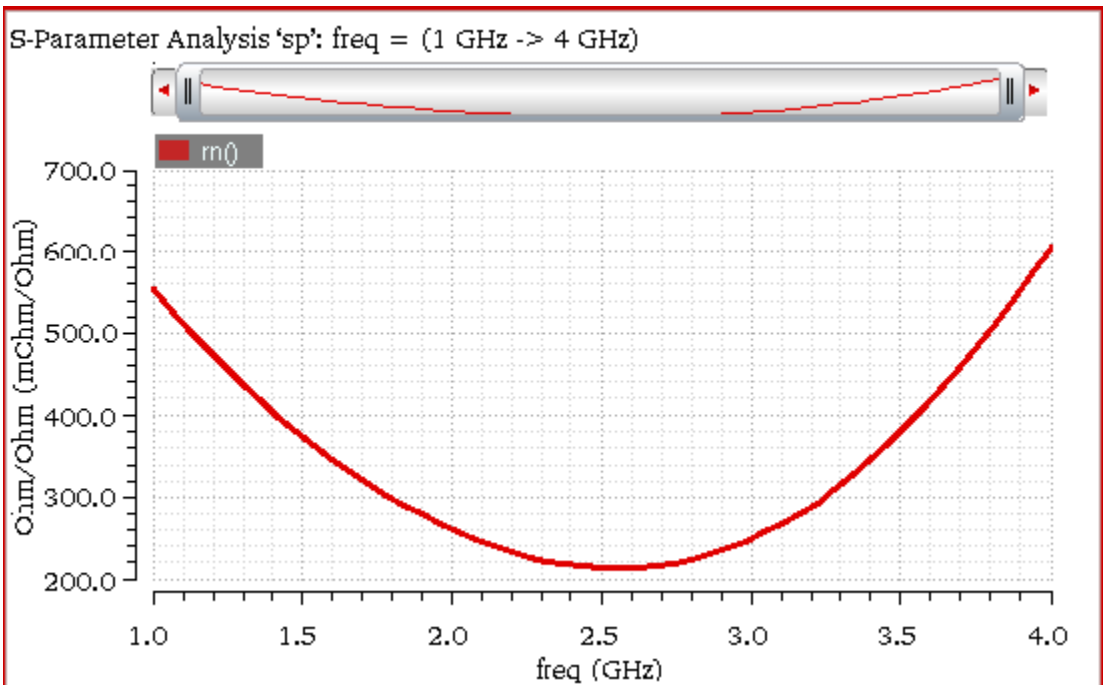
This example applies the `rn` function on the S-Parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `rn` function in the Function Panel, the following expression is displayed in the Buffer:

```
rn()
```


© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

s11

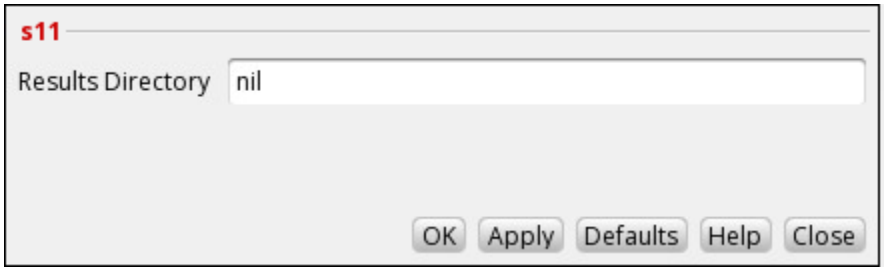
Returns s11 of a two-port network. S-parameters describe the response of an N-port network to voltage signals at each port. The first number in the subscript refers to the responding port, while the second number refers to the incident port. Therefore, S11 means the response at port1 due to a signal at port 1.

Consider a two-port network that is characterized by the scattering parameter matrix as below:

$$\begin{bmatrix} b_S \\ b_L \end{bmatrix} = \begin{bmatrix} S_{11} S_{12} \\ S_{21} S_{22} \end{bmatrix} \begin{bmatrix} a_S \\ a_L \end{bmatrix}$$

where `bS` and `bL` are the reflected waves from the input and output of the network, `aS` and `aL` are the incident waves to the input and output of the network.

$$S_{11} = \frac{b_S}{a_S} | a_L = 0$$



Example

This example applies the `s11` function on the S-parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `s11` function in the Function Panel, the following expression is displayed in the Buffer:

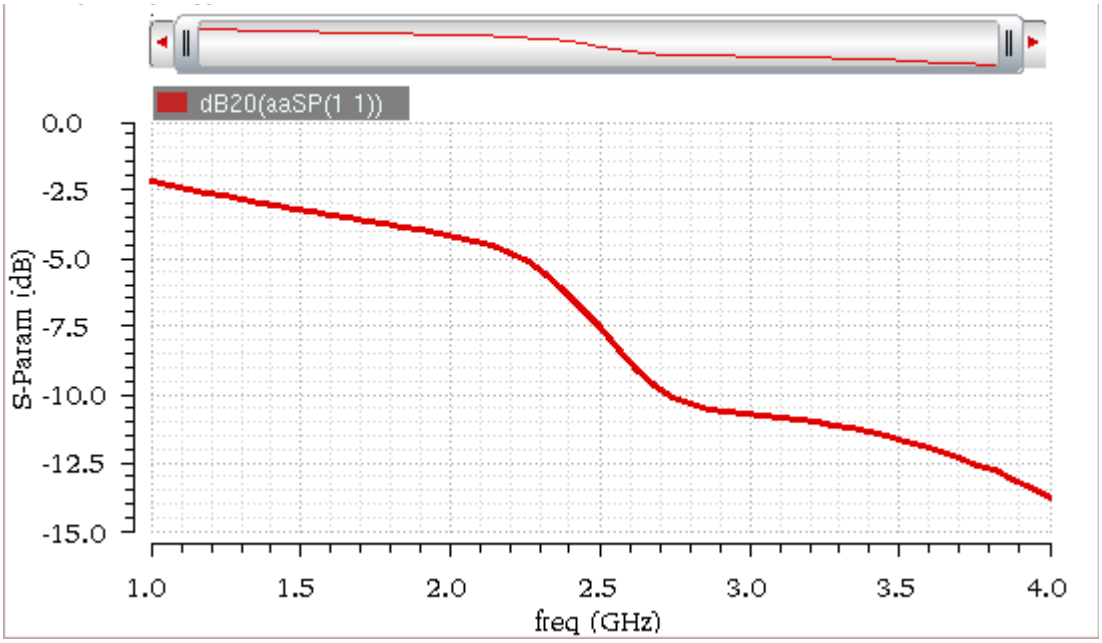
```
S11()
```

Now, convert this signal into dB by applying the `dB20` function on the above expression. The new expression in the Buffer is as follows:

© 2023 Cadence Design Systems, Inc.

```
dB20(aaSP(1 1))
```

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

s12

Returns s12 of a two-port network. S-parameters describe the response of an N-port network to voltage signals at each port. The first number in the subscript refers to the responding port, while the second number refers to the incident port. Therefore, S12 means the response at port1 due to a signal at port 2.

Consider a two-port network that is characterized by the scattering parameter matrix as below:

$$\begin{bmatrix} b_S \\ b_L \end{bmatrix} = \begin{bmatrix} S_{11} S_{12} \\ S_{21} S_{22} \end{bmatrix} \begin{bmatrix} a_S \\ a_L \end{bmatrix}$$

where `bS` and `bL` are the reflected waves from the input and output of the network, `aS` and `aL` are the incident waves to the input and output of the network.

$$S_{12} = \frac{b_S}{a_L} | a_S = 0$$

Example

This example applies the `s12` function on the S-parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `s12` function in the Function Panel, the following expression is displayed in the Buffer:

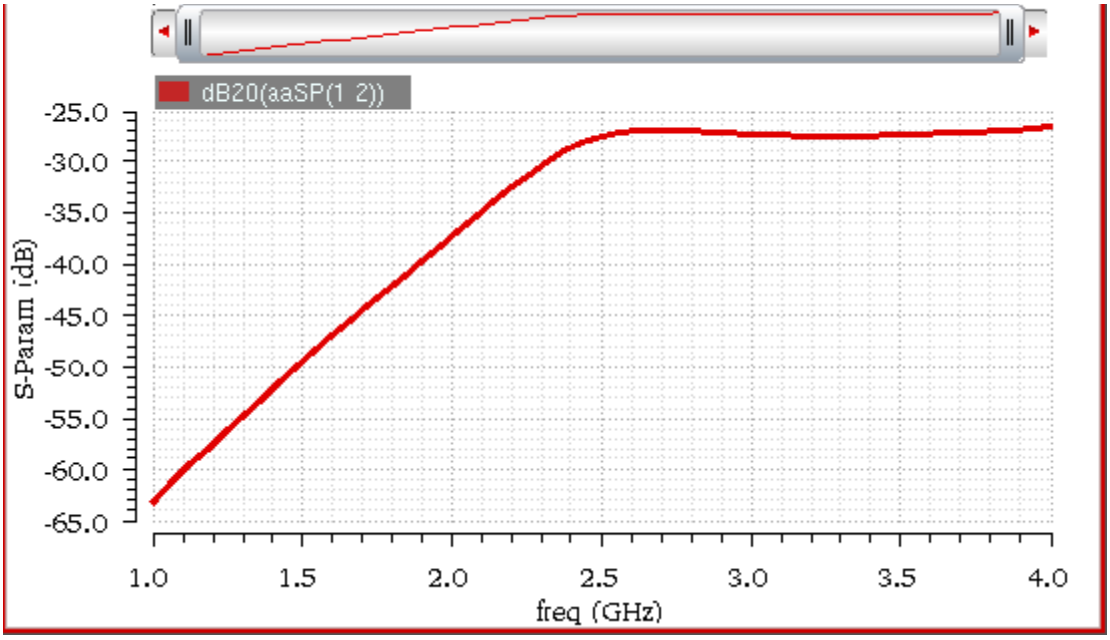
```
S12()
```

Now, convert this signal into dB by applying the `dB20` function on the above expression. The new expression in the Buffer is as follows:

© 2023 Cadence Design Systems, Inc.

```
dB20(aaSP(1 2))
```

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

s21

Returns s21 of a two-port network. S-parameters describe the response of an N-port network to voltage signals at each port. The first number in the subscript refers to the responding port, while the second number refers to the incident port. Therefore, S21 means the response at port 2 due to a signal at port 1.

Consider a two-port network that is characterized by the scattering parameter matrix as below:

$$\begin{bmatrix} b_S \\ b_L \end{bmatrix} = \begin{bmatrix} S_{11} S_{12} \\ S_{21} S_{22} \end{bmatrix} \begin{bmatrix} a_S \\ a_L \end{bmatrix}$$

where `bS` and `bL` are the reflected waves from the input and output of the network, `aS` and `aL` are the incident waves to the input and output of the network.

$$S_{21} = \frac{b_L}{a_S} | a_L = 0$$

Example

This example applies the `s21` function on the S-parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `s21` function in the Function Panel, the following expression is displayed in the Buffer:

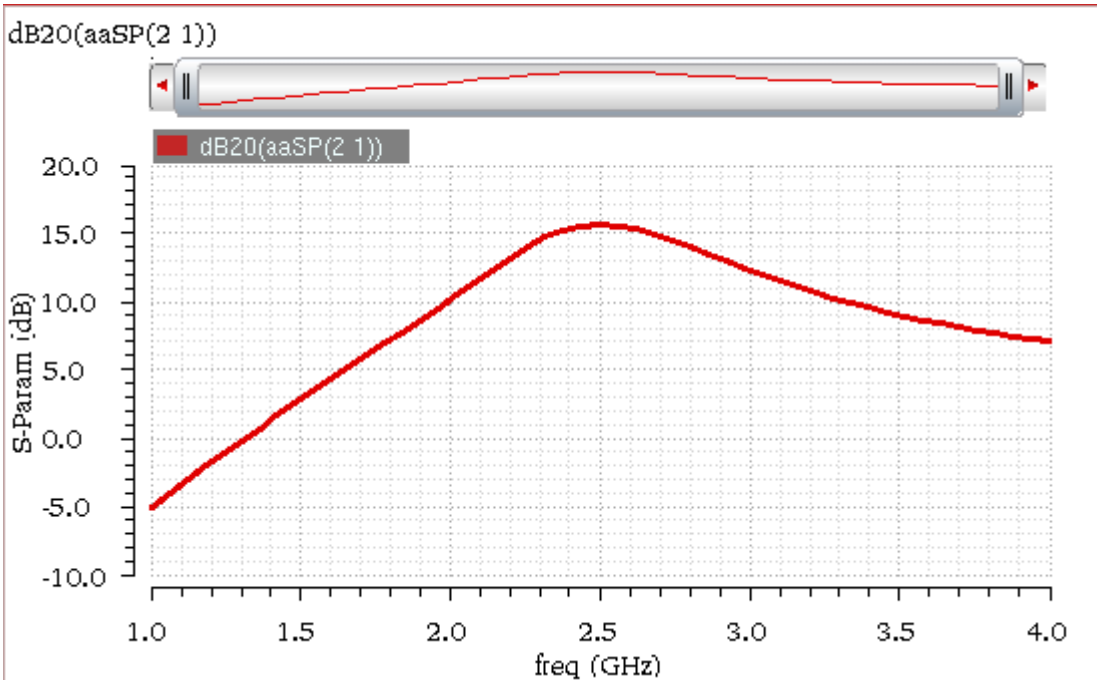
```
S21()
```

Now, convert this signal into dB by applying the `dB20` function on the above expression. The new expression in the Buffer is as follows:

© 2023 Cadence Design Systems, Inc.

```
dB20(aaSP(2 1))
```

Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

s22

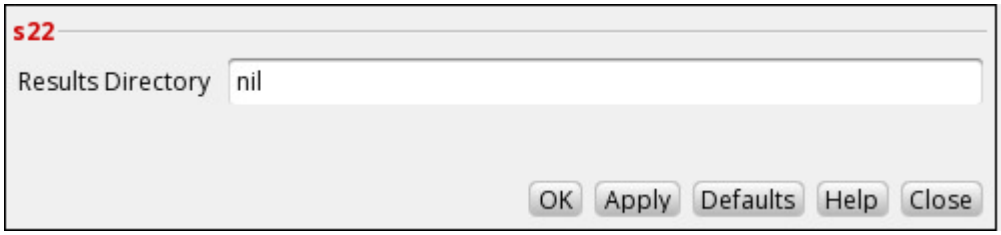
Returns s22 of a two-port network. S-parameters describe the response of an N-port network to voltage signals at each port. The first number in the subscript refers to the responding port, while the second number refers to the incident port. Therefore, S22 means the response at port 2 due to a signal at port 2.

Consider a two-port network that is characterized by the scattering parameter matrix as below:

$$\begin{bmatrix} b_S \\ b_L \end{bmatrix} = \begin{bmatrix} S_{11} S_{12} \\ S_{21} S_{22} \end{bmatrix} \begin{bmatrix} a_S \\ a_L \end{bmatrix}$$

where `bS` and `bL` are the reflected waves from the input and output of the network, `aS` and `aL` are the incident waves to the input and output of the network.

$$S_{22} = \frac{b_L}{a_L} | a_S = 0$$



Example

This example applies the `s22` function on the S-parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

When you select the `s22` function in the Function Panel, the following expression is displayed in the Buffer:

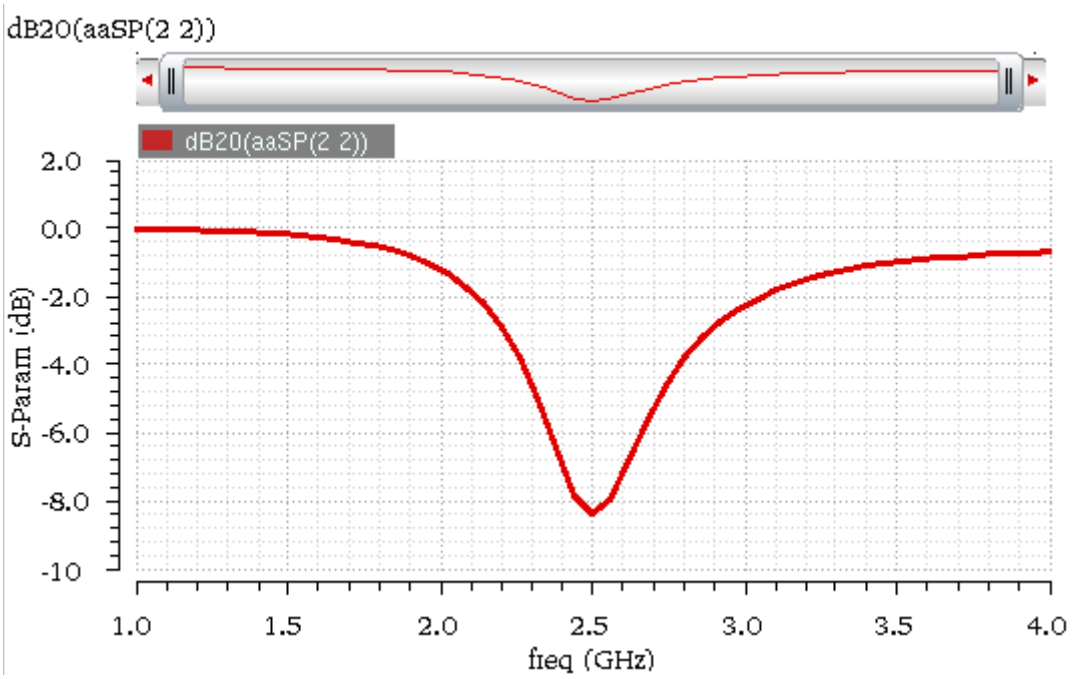
```
S22()
```

Now, convert this signal into dB by applying the `dB20` function on the above expression. The new expression in the Buffer is as follows:

```
dB20(aaSP(2 2))
```


© 2023 Cadence Design Systems, Inc.

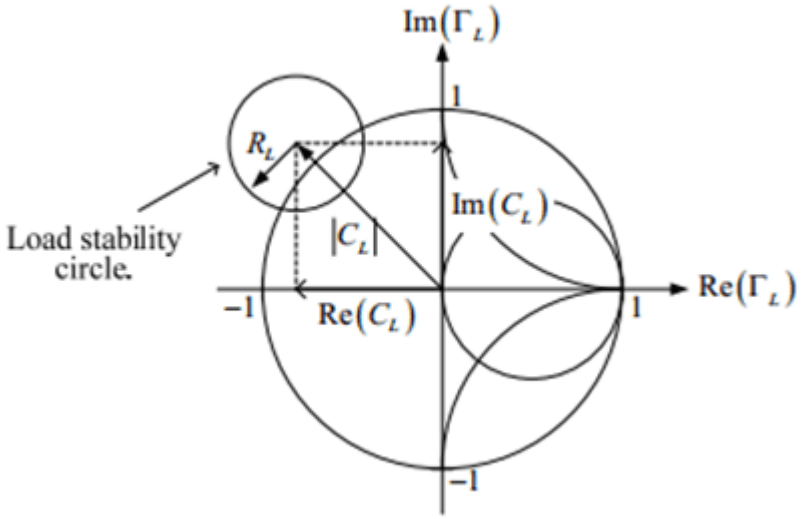
Now, when you evaluate this expression, the following output waveform appears in the graph window:



© 2023 Cadence Design Systems, Inc.

sourceStability

Returns the source stability circles. Stability circles define the boundary between stable and potentially unstable Γ_L or Γ_S. For a conditionally stable circuit, the load stability on an extended Smith chart is displayed in the figure below:



This function includes the following arguments:

- *Start (Hz)*—Start of the frequency range.
- *Stop (Hz)*—End of the frequency range.
- *Step*—Frequency step size to be used.

© 2023 Cadence Design Systems, Inc.

- *Results Directory*—(Optional) Results directory path.

Note: The default value is `0` for all the fields.

Example

This example applies the `sourceStability` function on the S-parameter analysis of a Low Noise Amplifier (LNA) design. For detailed information, see How to Run Examples?.

The following values for the function arguments has been specified in this example:

- *Start (Hz)*—`2G`
- *Stop (Hz)*—`3G`
- *Step*—`0.2G`

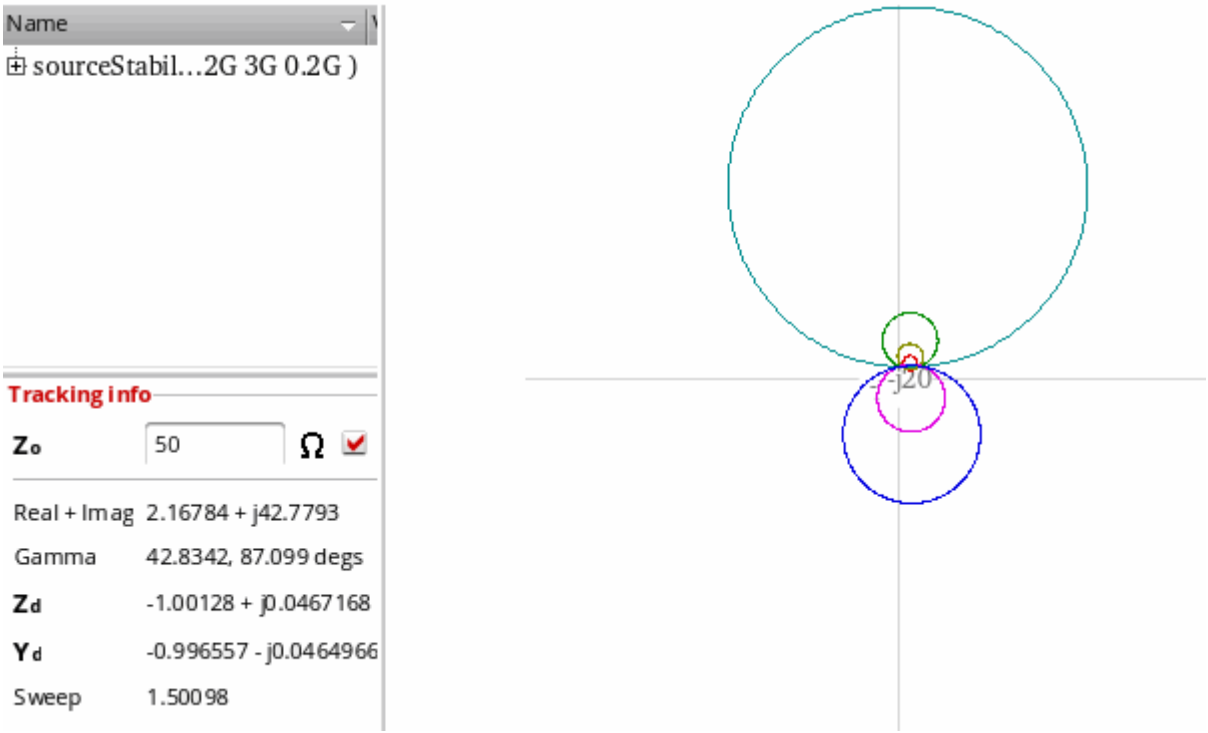
The corresponding expression created in the Buffer is as follows:

```
sourceStability( sp(1 1 ?result "sp" ?resultsDir "./psf") sp(1 2
?result "sp" ?resultsDir "./psf") sp(2 1 ?result "sp" ?resultsDir "./
psf") sp(2 2 ?result "sp" ?resultsDir "./psf") '(2G 3G 0.2G) )
```

Ensure that the `Impedance` graph type is selected in the Selection toolbar.

© 2023 Cadence Design Systems, Inc.

Now, when you evaluate this expression, the following output waveform is displayed in the graph window:



© 2023 Cadence Design Systems, Inc.

Spectre RF Functions

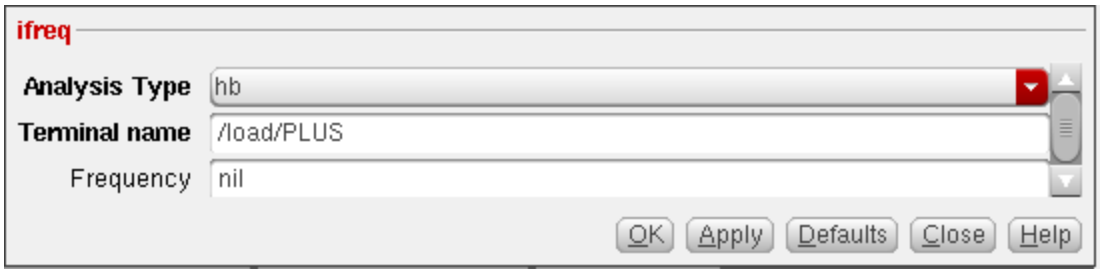
This section describes the following Spectre RF functions available in the Function Panel of Calculator:

- ifreq
- ih
- itime
- pir
- pmNoise
- pn
- pvi
- pvr
- rfGetMinDampFactor
- spm
- totalNoise
- vfreq
- vh
- vtime
- ypm
- ypm

© 2023 Cadence Design Systems, Inc.

ifreq

Returns the current of the terminal at a specified frequency or at all frequencies in the frequency domain.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, `qpss`, `pac`, `hbac`, `qpac`, and `ac`.
  Default value: `hb`
- *Terminal name*—Specify a terminal name on the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the terminal name directly from the Schematic window. The selected terminal name appears in this field.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the terminal name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

    The selected signal appears in the *Terminal name* field.
- *Frequency*—Specify the frequency for which you want to plot the results. It is an optional field.
  Valid values: Any integer or floating point number.
  Default value: `nil`.

Example

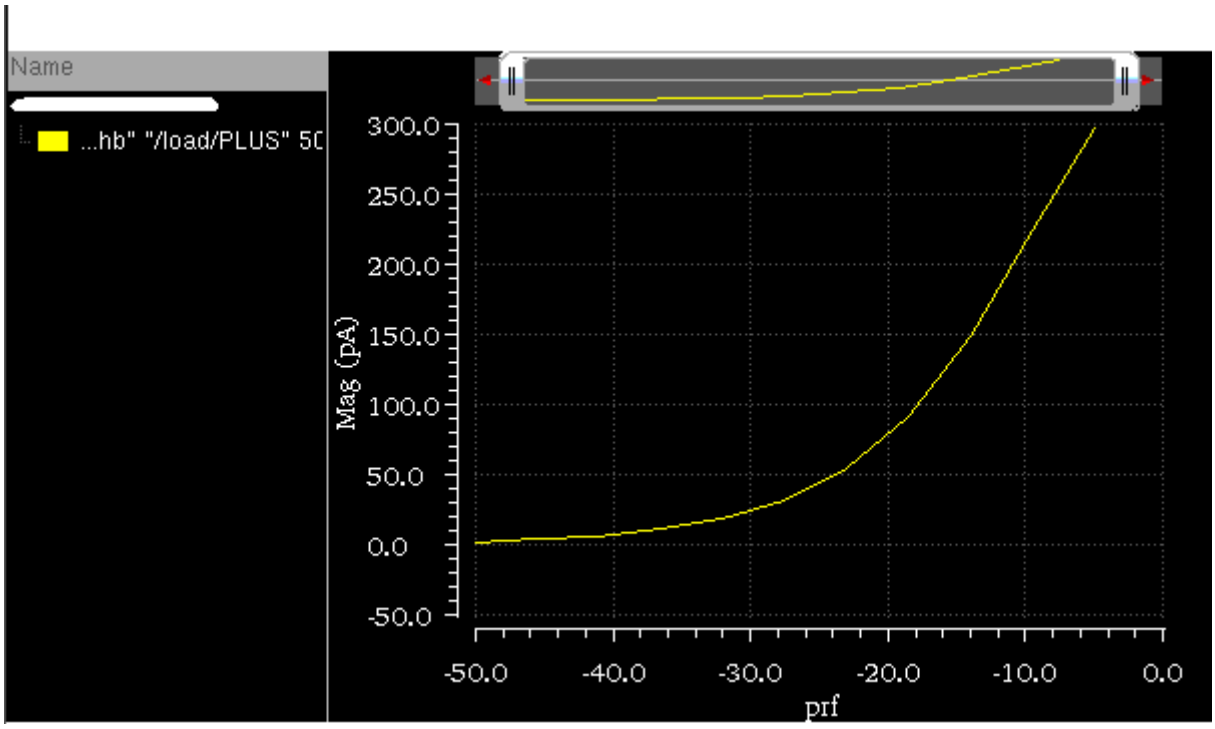
This example shows the output plot generated if you build the following expression in the Buffer:

```
ifreq("hb" "/load/PLUS" 50 )
```


© 2023 Cadence Design Systems, Inc.

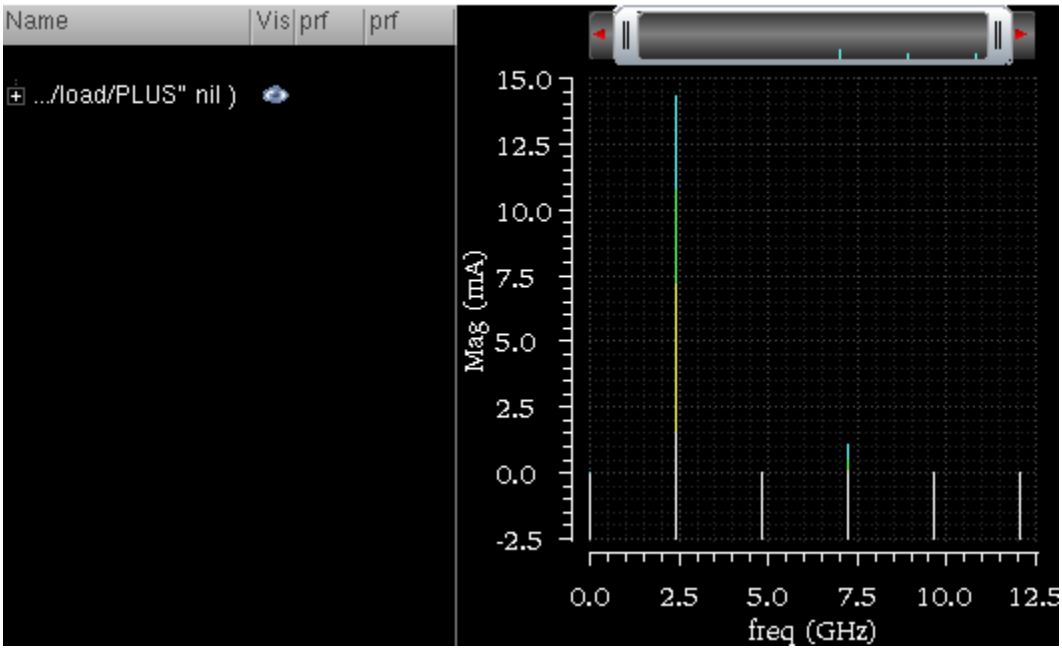
Here,

- *Analysis Type* is `hb`
- *Terminal name* is `/load/PLUS`
- *Frequency* is `50`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Frequency*=`nil`, the current on all the frequency points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the current at a specified frequency or at all frequency points:

```
i("/load/PLUS" ?result "hb_fd")
```

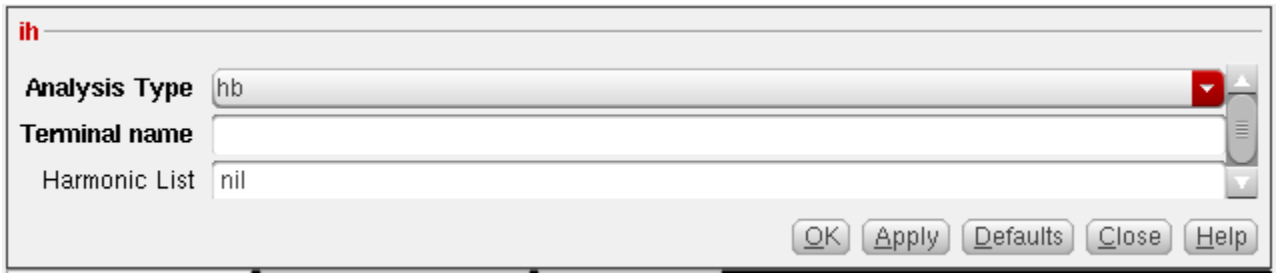
However, the similar calculation is now performed with the *ifreq* function and the following expression is created in the Buffer:

```
ifreq("hb" "/load/PLUS" nil )
```


© 2023 Cadence Design Systems, Inc.

ih

Returns the current of the terminal for a specified harmonic or for the harmonic list in the frequency domain.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, `qpss`, `pac`, `hbac`, and `qpac`.
  Default value: `hb`
- *Terminal name*— Specify the terminal name on the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the terminal name directly from the Schematic window. The selected terminal name appears in this field.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the terminal name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

    The selected signal appears in the *Terminal name* field.
- *Harmonic List*—Specify the harmonics for which you want to plot the results. It is an optional field. For analyses, such as `hb`, `pss`, `pac`, and `hbac`, you can add either single harmonic value or available list of harmonic values in this field.
  Valid values: Any integer or a list from the available list of harmonics. You can find the available harmonics by using the `harmonicList` function.
  `harmonicList( [?resultsDir t_resultsDir] [?result S_resultName]`
  Default value: `nil`.

© 2023 Cadence Design Systems, Inc.

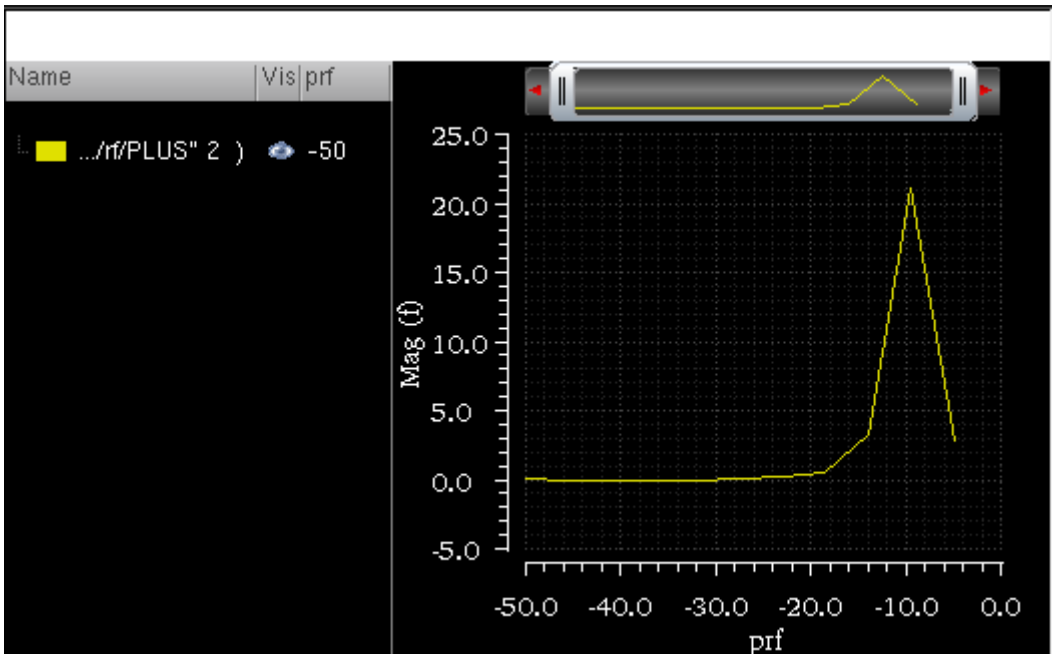
Example

This example shows the output plot generated if you build the following expression in the Buffer:

```
ih("hb" "/rf/PLUS" 2 )
```

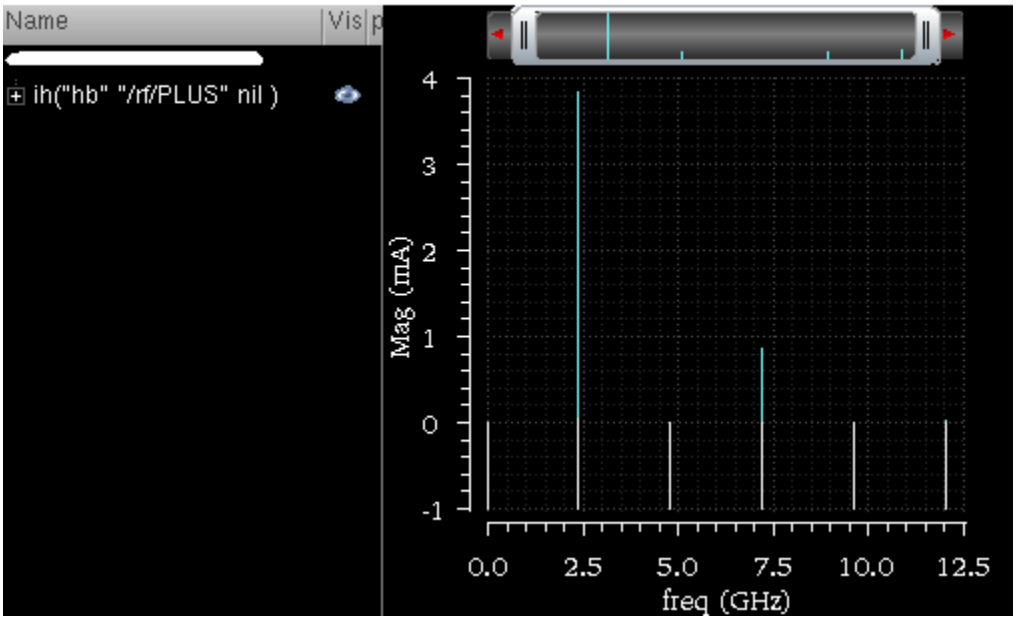
Here,

- *Analysis Type* is `hb`
- *Terminal name* is `/rf/PLUS`
- *Harmonic List* is `2`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Harmonic List*=`nil`, the current on all the harmonic points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the current at a specified harmonic or at all harmonic points:

```
i("/load/PLUS" ?result "hb_fd")
```

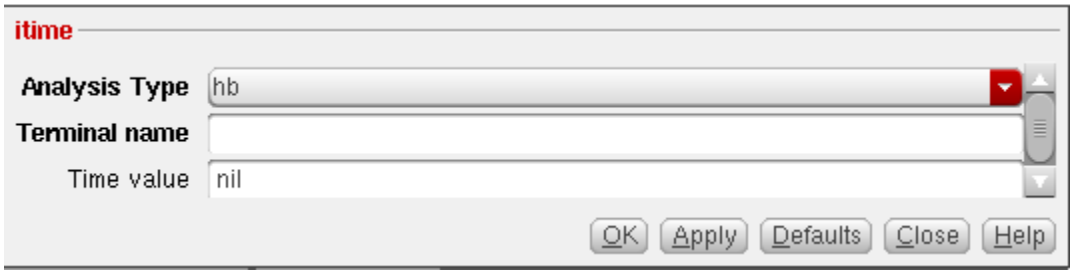
However, the similar calculation is now performed with the *ih* function and the following expression is created in the Buffer:

```
ih("hb" "/load/PLUS" nil )
```


© 2023 Cadence Design Systems, Inc.

itime

Returns the current of the terminal at a specified time point or at all time points in the time domain.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, and `tran`.
  Default value: `hb`
- *Terminal name*— Specify the terminal name on the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the terminal name directly from the Schematic window. The selected terminal name appears in this field.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the terminal name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

    The selected signal appears in the *Terminal name* field.
- *Time value*—Specify the time points for which you want to plot the results. If you specify a time point in this field, the result of the specified time is returned. Otherwise, result on all the time points are returned. It is an optional field.
  Valid values: Any integer or floating point number.
  Default value: `nil`.

Example

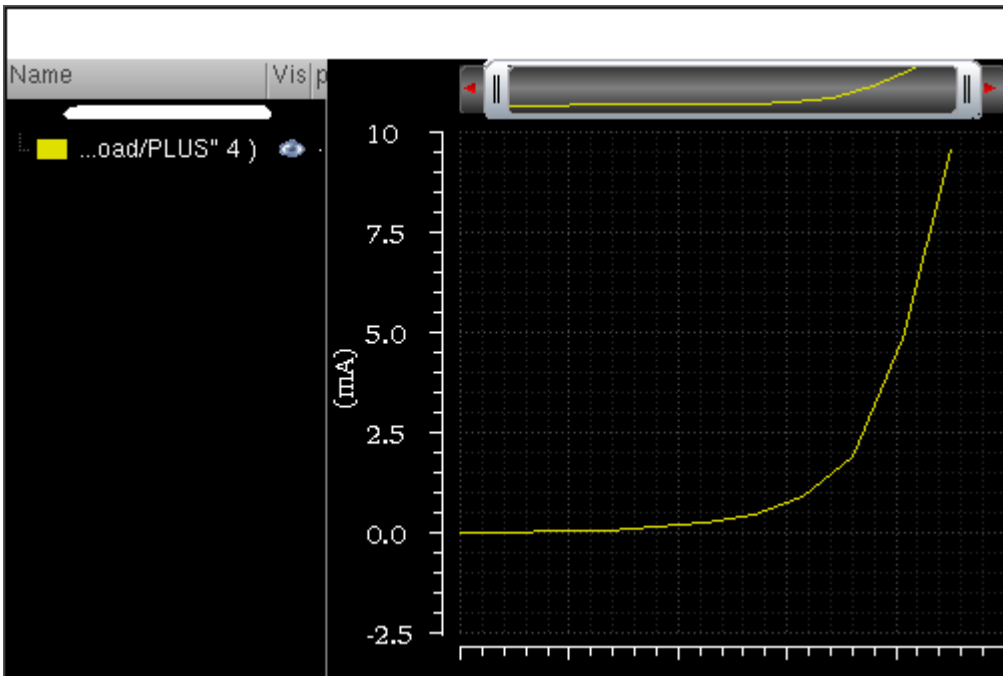
This example shows the output plot generated if you build the following expression in the Buffer:

```
itime("hb" "/load/PLUS" 4 )
```


© 2023 Cadence Design Systems, Inc.

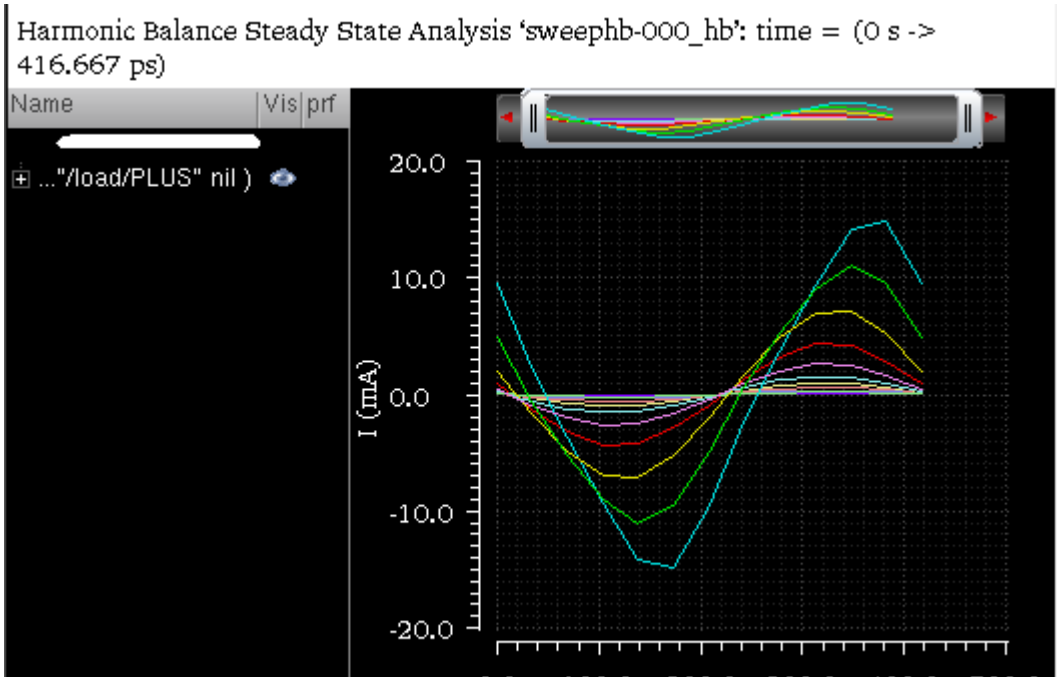
Here,

- *Analysis Type* is `hb`
- *Terminal name* is `/load/PLUS`
- *Time value* is `4`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Time value*=`nil`, the current on all the time points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the current at a time point or at all time points:

```
i("/load/PLUS" ?result "hb_td")
```

However, the similar calculation is now performed with the *itime* function and the following expression is created in the Buffer:

```
itime("hb" "/load/PLUS" nil )
```


© 2023 Cadence Design Systems, Inc.

pir

Returns the spectral power from the current at two terminals and resistance for a specified harmonic list.

This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, `qpss`, `pac`, `hbac` and `qpac`.
  Default value: `hb`
- *Branch name 1*— Specify the first branch name in the schematic or signal name from the Results Browser.
- *Branch name 2*—Specify the second branch name in the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the branch name directly from the Schematic window. The selected branch name appears in the branch name fields.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the branch name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

  The selected signal appears in the *Branch name* field.
- *Resistance*—Specify the resistance value. This field can contain any integer or floating point value. By default, it is set to blank.
- *Harmonic List*—Specify the harmonics for which you want to plot the results. It is an optional field. For analyses, such as `hb`, `pss`, `pac`, and `hbac`, you can add either single harmonic value or available list of harmonic values in this field.

© 2023 Cadence Design Systems, Inc.

Valid values: Any integer or a list from the available list of harmonics. You can find the available harmonics by using the `harmonicList` function.
`harmonicList( [?resultsDir t_resultsDir] [?result S_resultName]`
Default value: `nil`.

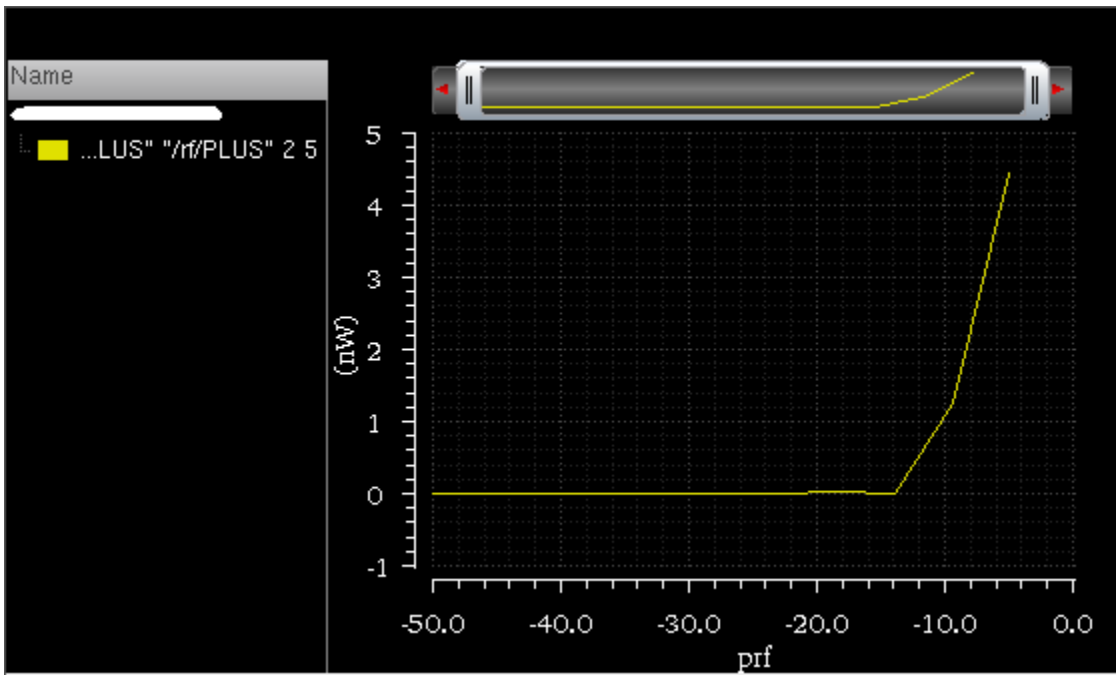
Example

This example shows the output plot generated if you build the following expression in the Buffer:

```
pir("hb" "/V1/PLUS" "/rf/PLUS" 2 5 )
```

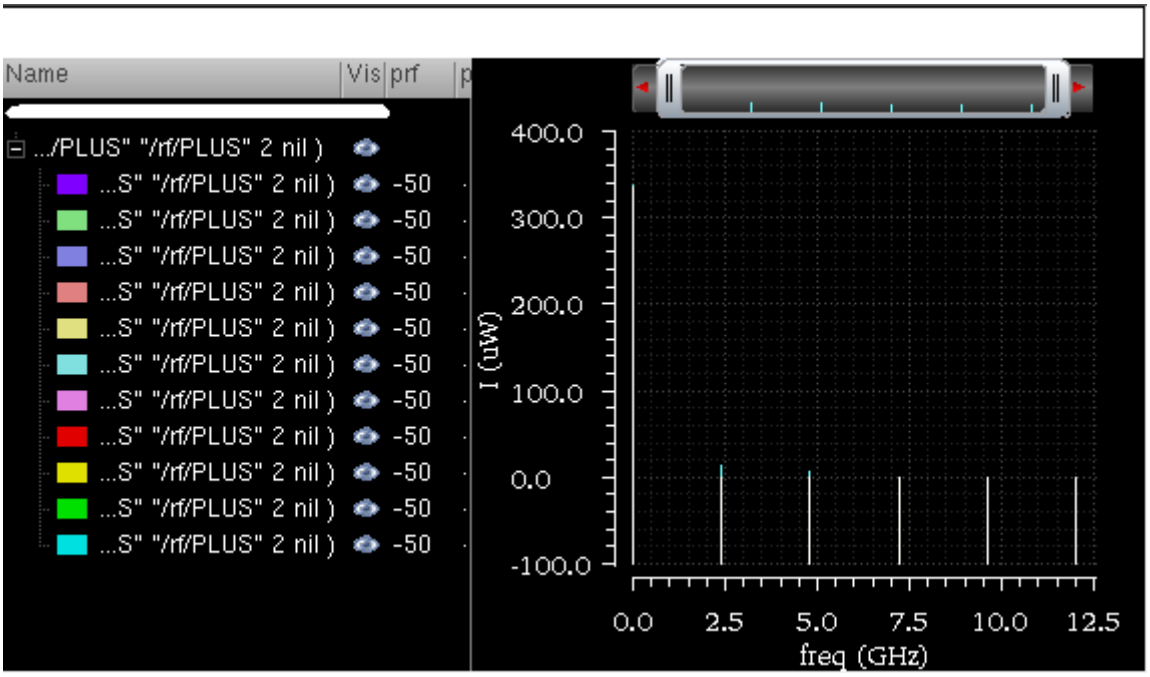
Here,

- *Analysis Type* is `hb`
- *Branch name 1* is `/V1/PLUS`
- *Branch name 2* is `/rf/PLUS`
- *Resistance* is `2`
- *Harmonic List* is `5`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Harmonic List*=`nil`, the spectral power on all the harmonic points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the spectral power from current and resistance for a specified harmonic list:

```
spectralPower(((i("/rf/PLUS" ?result "hb_fd") - i("/load/PLUS"
?result "hb_fd")) * 50) (i("/rf/PLUS" ?result "hb_fd") - i("/load/
PLUS" ?result "hb_fd")))
```

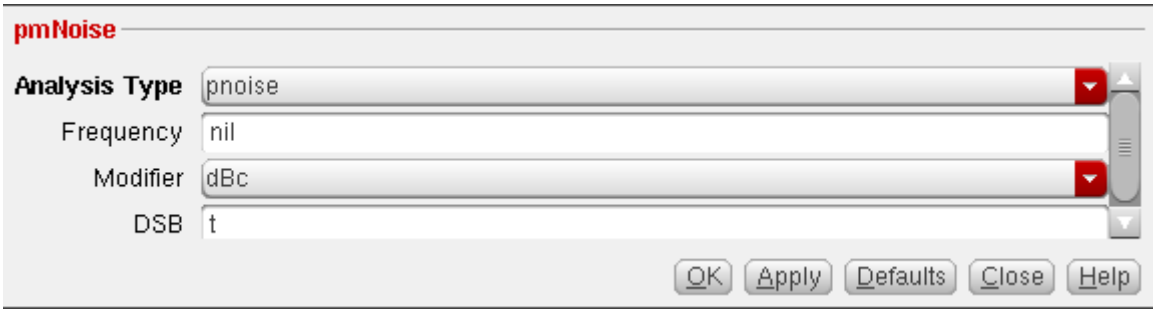
However, the similar calculation is now performed with the *pir* function and the following expression is created in the Buffer:

```
pir("hb" "/rf/PLUS" "/load/PLUS" 50 nil )
```


© 2023 Cadence Design Systems, Inc.

pmNoise

Returns the modulated phase noise for a specified frequency or for the entire spectrum.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `pnoise` and `hbnoise`.
  Default value: `pnoise`
- *Frequency*—Specify the frequency for which you want to calculate the phase noise. By default, this field is set to `nil`, which means the modulated phase noise at all frequency points are calculated.
- *Modifier*—Select the value of the modifier to be used.
  Valid values: `dBc`, `normalized`, `Power`, `Magnitude`, and `dBV`
  Default value: `dBc`
- *DSB*—Specify whether you want to include the double side band.
  Valid values: `t` and `nil`
  Default value: `t`

Example

This example shows the output generated if you build the following expression in the Buffer:

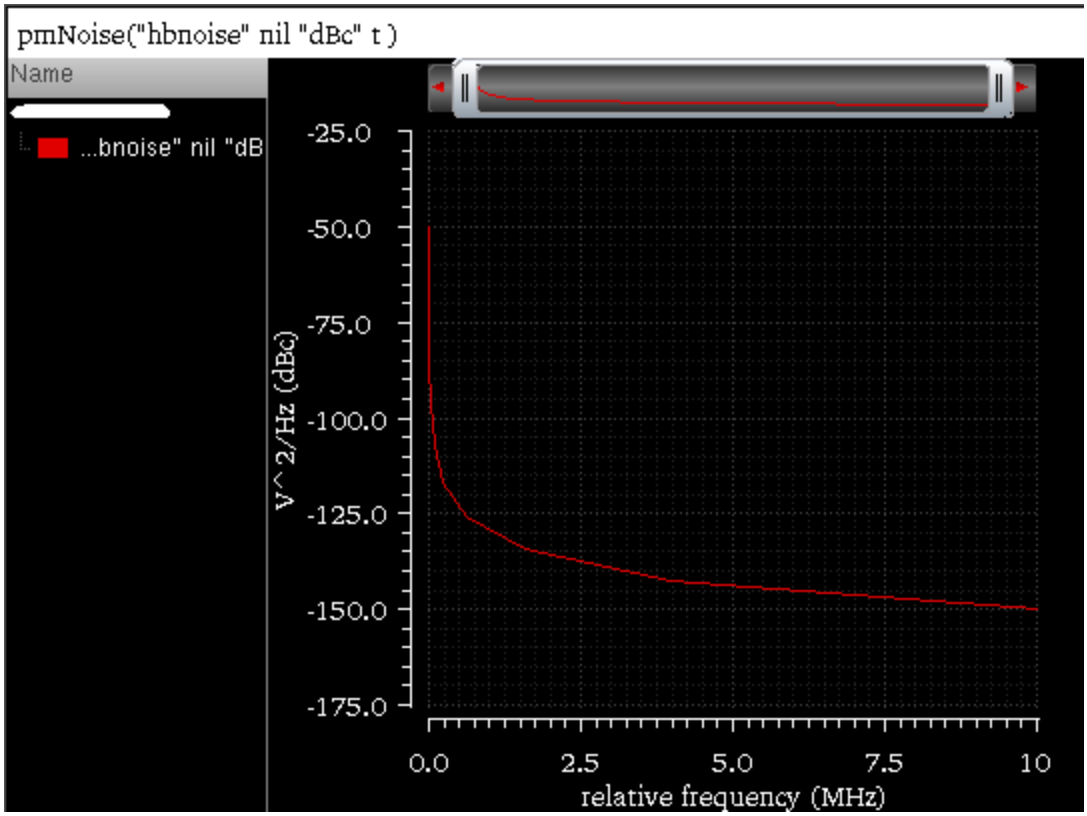
```
pmNoise("hbnoise" 50 "dBc" t )
```

- *Analysis Type* is `hbnoise`
- *Frequency* is `50`
- *Modifier* is `dBc`
- *DSB* is `t`

© 2023 Cadence Design Systems, Inc.

The output generated in the Buffer is `-50.0`, which is a scalar value.

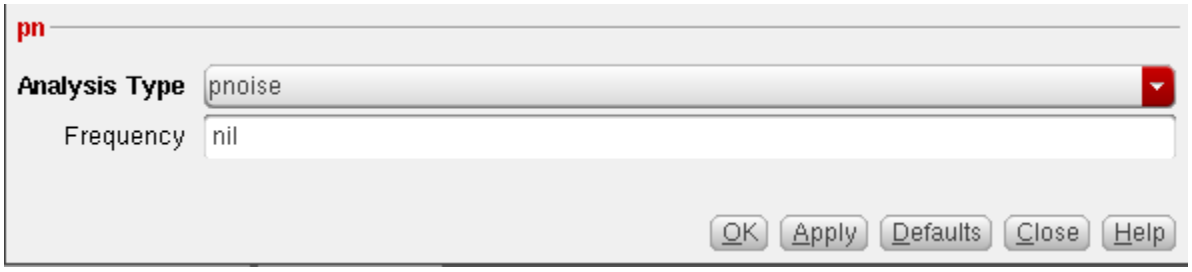
Now, if you specify *Frequency*=`nil`, phase noise on all frequency points are returned as shown in the figure below:



© 2023 Cadence Design Systems, Inc.

pn

Returns the phase noise at a specified frequency or at all frequency points.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list.
  Valid values: `pnoise`, `hbnoise`, and `qpnoise`.
  Default value: `pnoise`
- *Frequency*—Specify the frequency for which you want to calculate the phase noise. By default, this field is set to `nil`, which means the frequency at all points are calculated.

Example

This example shows the output generated if you build the following expression in the Buffer:

```
pn("hbnoise" 50 )
```

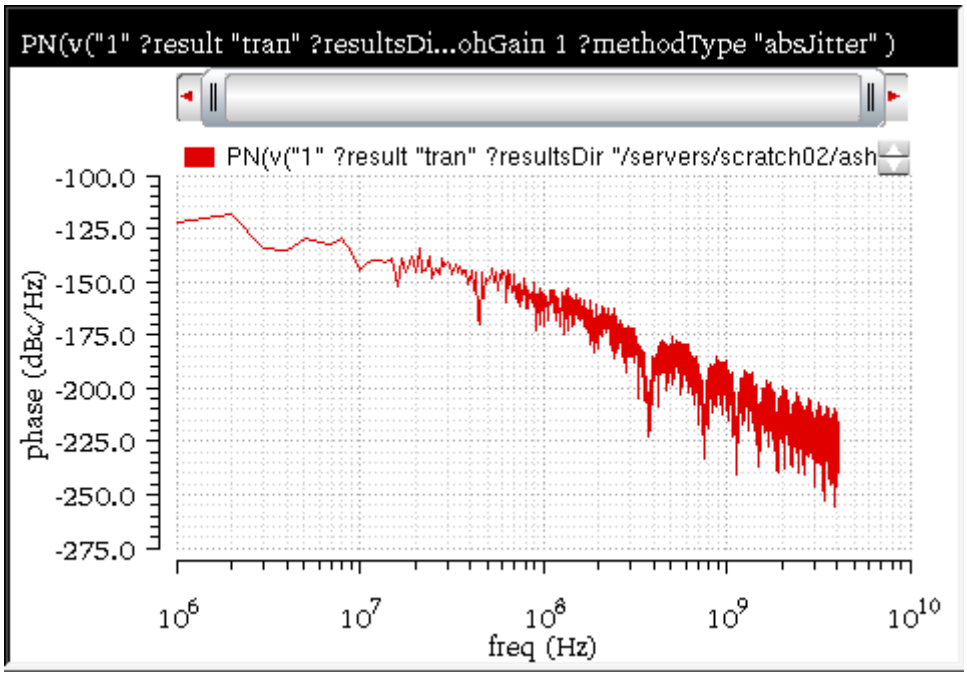
- *Analysis Type* is `hbnoise`
- *Frequency* is `50`

The output generated in the Buffer is `-53.01`, which is a scalar value.

Now, if you specify *Frequency*=`nil`, the phase noise on all frequency points are returned as shown in the figure below:

© 2023 Cadence Design Systems, Inc.

```
pn("hbnoise" nil )
```



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the phase noise at a specified frequency or at all frequency points:

```
phaseNoise(3 "hb_fd" ?result "hbnoise")
```

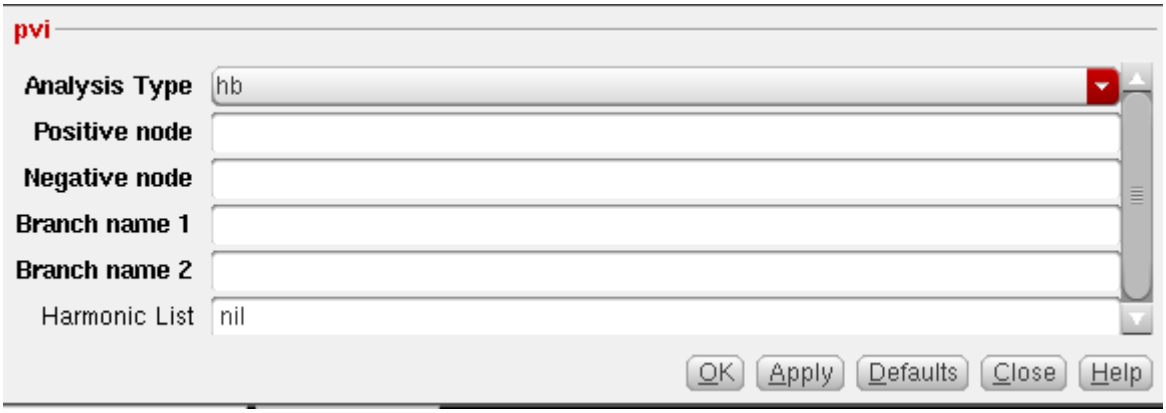
However, the similar calculation is now performed with the *pn* function and the following expression is created in the Buffer:

```
pn('hbnoise)
```


© 2023 Cadence Design Systems, Inc.

pvi

Returns the spectral power voltage on the positive and negative nodes and the current at two terminals for a specified harmonic list or for all harmonics.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, `qpss`, `pac`, `hbac` and `qpac`.
  Default value: `hb`
- *Positive node*—Specify the positive node or net. This field can also contain an explicit voltage value.
- *Negative node*—Specify the negative node or net. This field can also contain an explicit voltage value.
- *Branch name 1*— Specify the first branch name in the schematic or signal name from the Results Browser.
- *Branch name 2*—Specify the second branch name in the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the node and branch name directly from the Schematic window. The selected node or branch name appears in the node and branch name fields.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the node and branch name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

  The selected signal appears in the node and branch name fields.

© 2023 Cadence Design Systems, Inc.

- *Harmonic List*—Specify the harmonics for which you want to plot the results. It is an optional field. For analyses, such as `hb`, `pss`, `pac`, and `hbac`, you can add either single harmonic value or available list of harmonic values in this field.
  Valid values: Any integer or a list from the available list of harmonics. You can find the available harmonics by using the `harmonicList` function.
  `harmonicList( [?resultsDir t_resultsDir] [?result S_resultName]`
  Default value: `nil`

Example 1

The following example shows the output plot generated if you build the following expression in the Buffer:

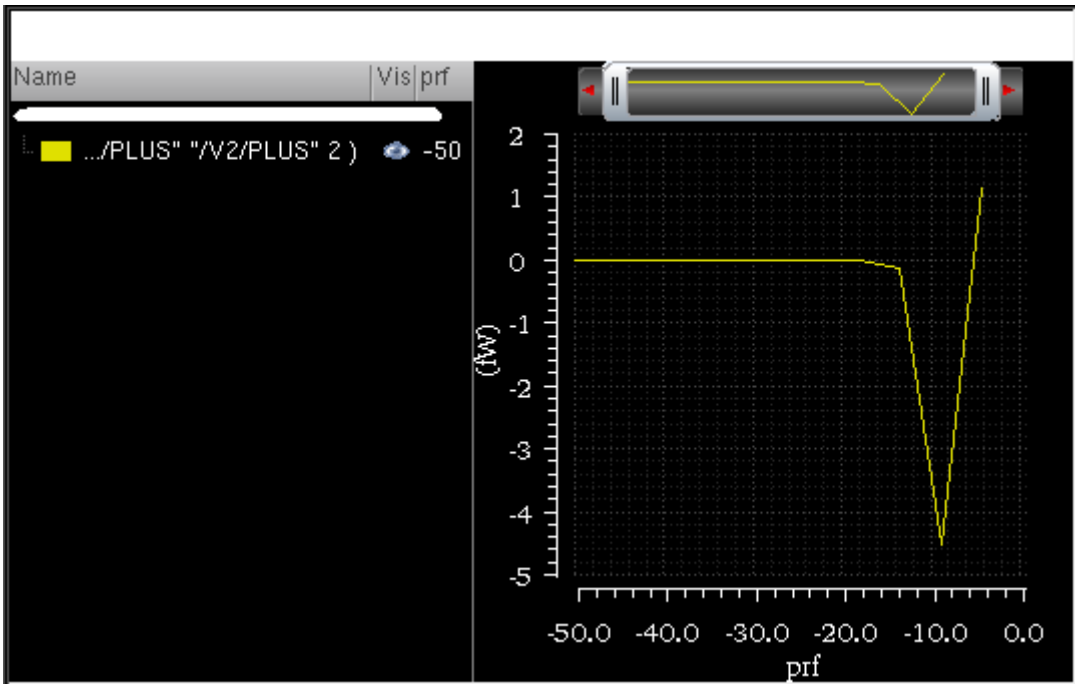
```
pvi("hb" "/RFin" "/RFout" "/V1/PLUS" "/V2/PLUS" 2)
```

Here,

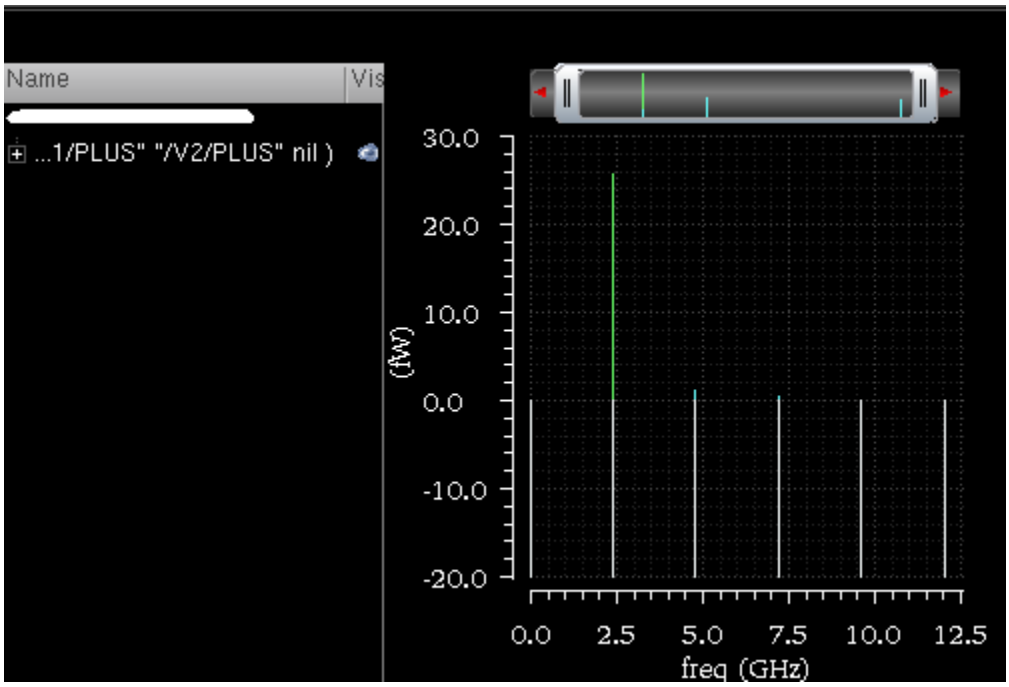
- *Analysis Type* is `hb`
- *Positive node* is `/RFin`
- *Negative node* is `/RFout`
- *Branch name* 1 `/V1/PLUS`
- *Branch name 2* `/V2/PLUS`

© 2023 Cadence Design Systems, Inc.

- *Harmonic List* is `2`



If you build the above expression with *Harmonic List*=`nil`, the spectral power on all the harmonic points are returned (as shown in the figure below).



© 2023 Cadence Design Systems, Inc.

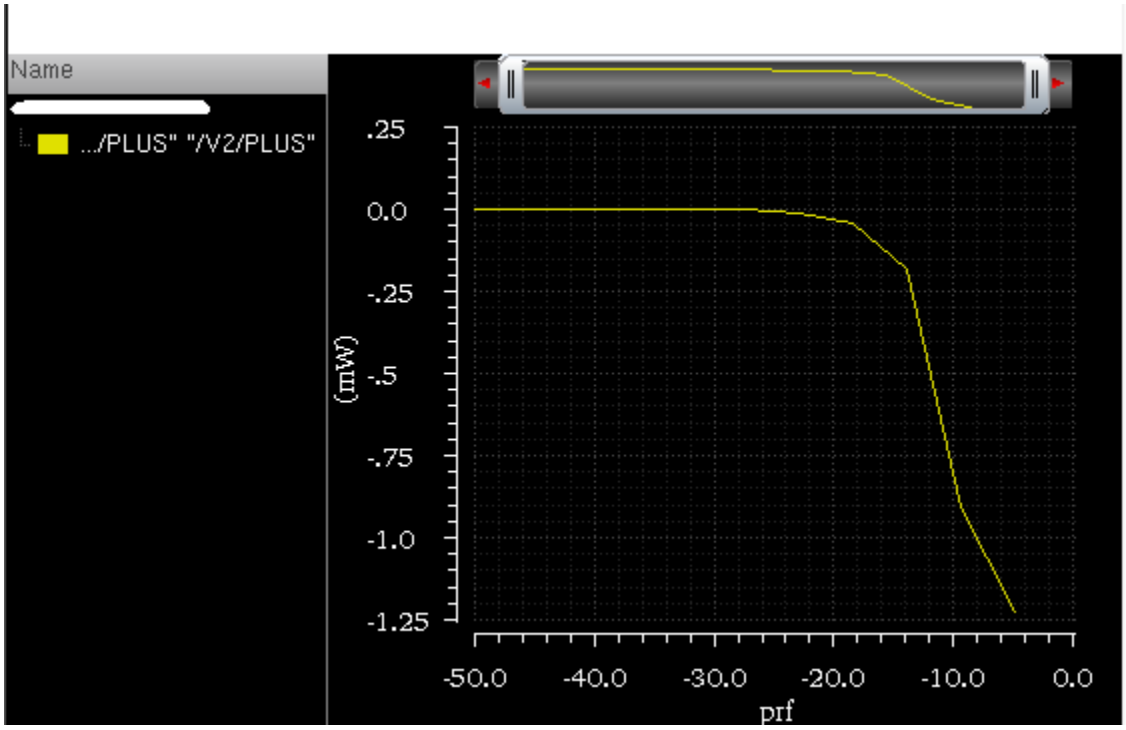
Example 2

The following example shows the output plot generated if you build the following expression in the Buffer:

```
pvi("hb" 2 3 "/V1/PLUS" "/V2/PLUS" 2 )
```

Here,

- *Analysis Type* is `hb`
- *Positive node* is `2`
- *Negative node* is `3`
- *Branch name* 1 `/V1/PLUS`
- *Branch name 2* `/V2/PLUS`
- *Harmonic List* is `2`



© 2023 Cadence Design Systems, Inc.

Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the spectral power from voltage and current at a specified harmonic list or at all harmonics:

```
spectralPower(i("/load/PLUS" ?result "hb_fd") v("/RFout" ?result
"hb_fd"))
```

However, the similar calculation is now performed with the *pvi* function and the following expression is created in the Buffer:

```
pvi("hb" "/RFout" 0 "/load/PLUS" 0 nil )
```


© 2023 Cadence Design Systems, Inc.

pvr

Returns the spectral power for a specified harmonic list or for all harmonics with resistor and voltage on the positive and negative nodes.

This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, `qpss`, `pac`, `hbac` and `qpac`.
  Default field: `hb`
- *Positive node*—Specify the positive node or net. This field can also contain an explicit voltage value.
- *Negative node*—Specify the negative node or net. This field can contain an explicit voltage value.

  When you open Calculator from ADE, you can select the node name directly from the Schematic window. The selected node name appears in the node and branch name fields.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the node name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

  The selected signal appears in the node name fields.
- *Resistance*—Specify the resistance value. This field can contain any integer or floating point value. By default, it is set to blank.

© 2023 Cadence Design Systems, Inc.

- *Harmonic List*—Specify the harmonics for which you want to plot the results. It is an optional field. For analyses, such as `hb`, `pss`, `pac`, and `hbac`, you can add either single harmonic value or available list of harmonic values in this field.
  Valid values: Any integer or a list from the available list of harmonics. You can find the available harmonics by using the `harmonicList` function.
  `harmonicList( [?resultsDir t_resultsDir] [?result S_resultName]`
  Default value: `nil`

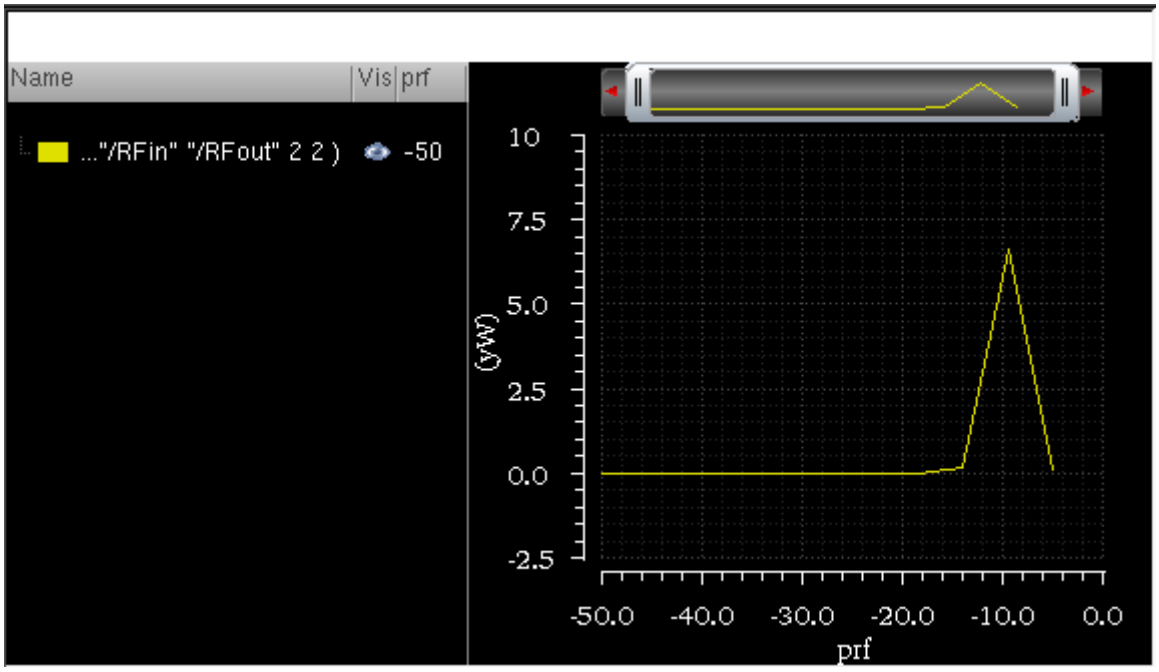
Example

The following example shows the output plot generated if you build the following expression in the Buffer:

```
pvr("hb" "/RFin" "/RFout" 2 2 )
```

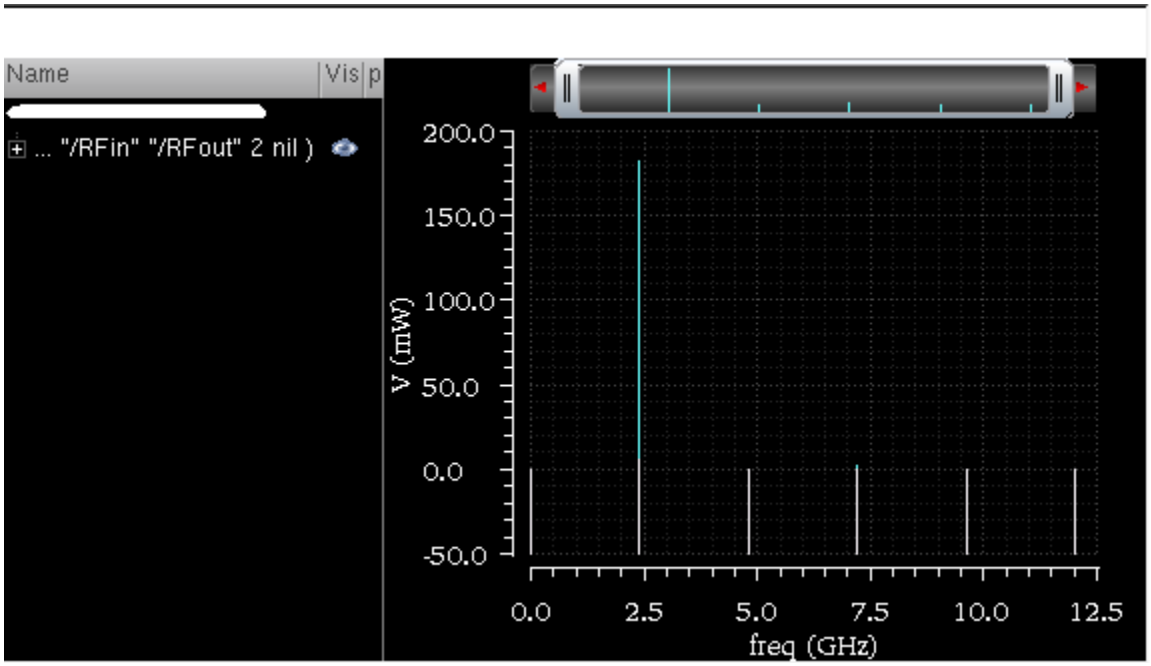
Here,

- *Analysis Type* is `hb`
- *Positive node* is `/RFin`
- *Negative node* is `/RFout`
- *Resistance* is `2`
- *Harmonic List* is `2`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Harmonic List*=`nil`, the spectral power on all the harmonic points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the spectral power at a specified harmonic or at all harmonics with resistor and voltage on the positive and negative nodes.

```
spectralPower(((v("/RFin" ?result "hb_fd") - v("/RFout" ?result
"hb_fd")) / 50) (v("/RFin" ?result "hb_fd") - v("/RFout" ?result
"hb_fd")))
```

However, the similar calculation is now performed with the *pvr* function and the following expression is created in the Buffer:

```
pvr("hb" "/RFin" "/RFout" 50 nil )
```


© 2023 Cadence Design Systems, Inc.

rfGetMinDampFactor

Returns the lowest damping ratio for the loops identified in the Loop Finder (LF) analysis.

Example

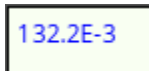
Open the results from the Loop Finder (LF) analysis and ensure that the *In Context Results DB* field in Calculator contains the path of the LF results directory.

When you click the `rfGetMinDampFactor` function available in the *Spectre RF Functions* category in the Function Panel, the following expression is created in the Buffer:

```
rfGetMinDampFactor()
```

This function does not include any argument.

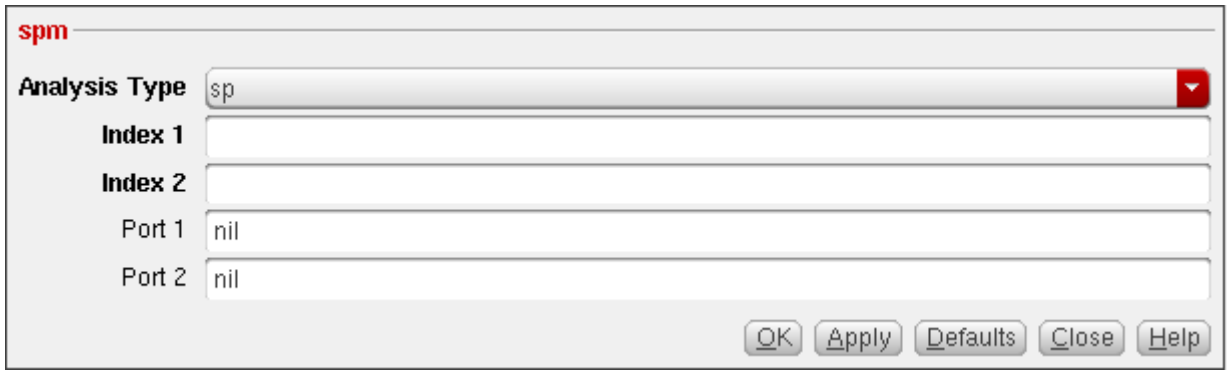
When you evaluate this expression, the lowest damping ratio for the loops given in the selected results directory is calculated and displayed in the Buffer, as shown in the figure below:



© 2023 Cadence Design Systems, Inc.

spm

Returns the waveform for s-parameters.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list.
  Valid values: `sp`, `psp`, `qpsp`, `hbsp` and `lssp`
  Default value: `sp`
- *Index 1*—Specify the port index for sp simulation. By default, this field is set to blank.
  Valid values: Available port index, such as `1`, `2`.
- *Index 2*—Specify the port index for two-port sp simulation. By default, this field is set to blank.
  Valid values: Available port index, such as `1`, `2`.
- *Port 1*—Specify the port instance. The port instance can be specified only for the differential s-parameter analysis and not applicable for `psp`, `qpsp` and `hbsp` analyses.
  Valid values: Predefined values “`c`” and “`d`” for Spectre simulator.

  Note: Do not use the `VAR` expression to specify port numbers because it returns an SRR object instead of an integer, which is expected by `spm`.
- *Port 2*—Specify the port instance. The port instance can be specified only for the differential s-parameter analysis and not applicable for `psp`, `qpsp` and `hbsp` analyses.
  Valid values: Predefined values “`c`” and “`d`” for Spectre simulator.

  Note: Do not use the `VAR` expression to specify port numbers because it returns an SRR object instead of an integer, which is expected by `spm`.

© 2023 Cadence Design Systems, Inc.

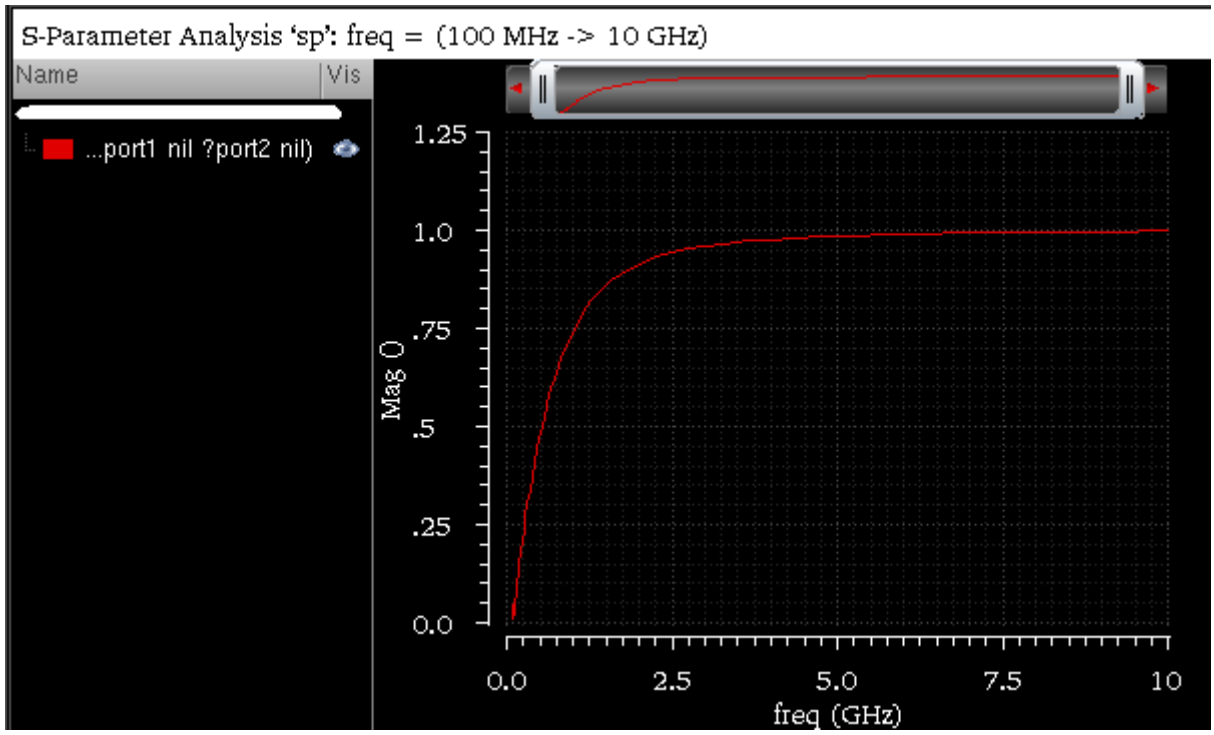
Example 1:

This example shows the output plot generated when you build the following expression in the Buffer:

```
spm("sp" 1 1 ?port1 nil ?port2 nil)
```

Here,

- *Analysis Name* is `sp`
- *Index 1* is `1`
- *Index 2* is `1`
- *Port 1* is `nil`
- *Port 2* is `nil`



Example 2

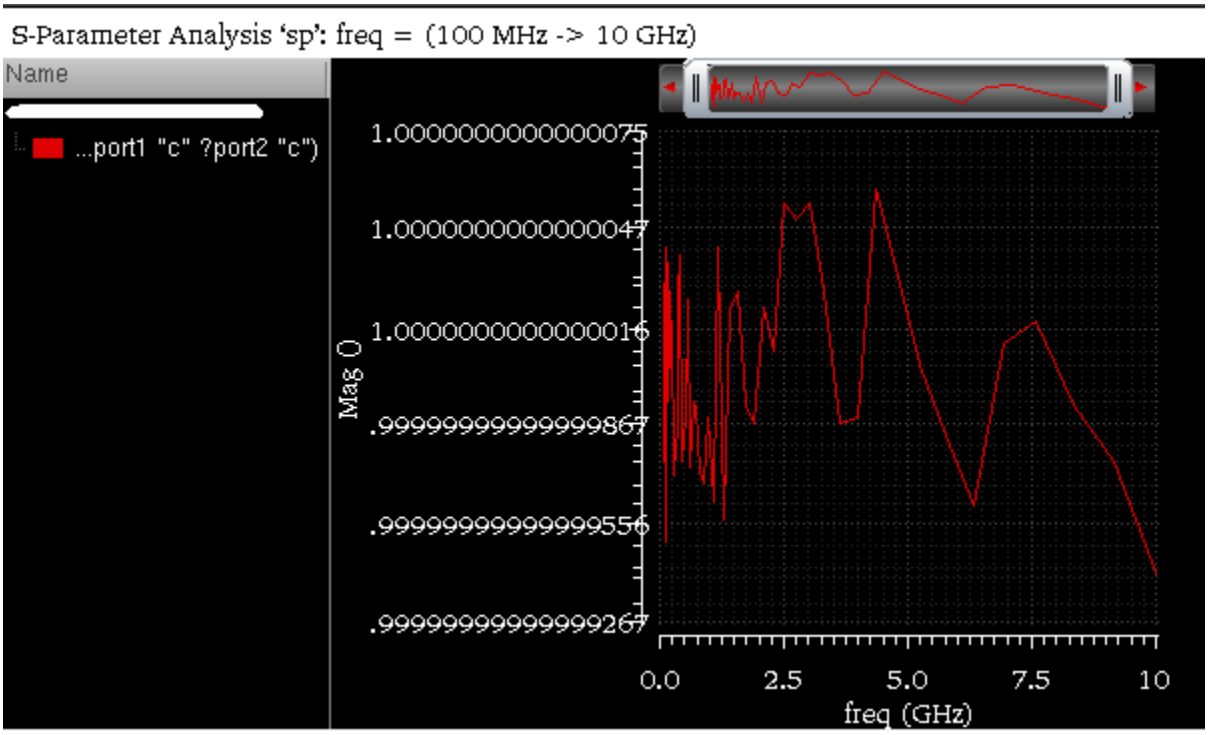
This example shows the output plot generated when you build the following expression in the Buffer for differential s-parameter analysis:

```
spm("sp" 1 1 ?port1 "c" ?port2 "c")
```


© 2023 Cadence Design Systems, Inc.

Here,

- *Analysis Name* is `sp`
- *Index 1* is `1`
- *Index 2* is `1`
- *Port 1* is "`c`"
- *Port 2* is "`c`"



Additional Information

The following expression was created in the Buffer in the previous releases when you plotted the waveform for s-parameters:

```
sp( 1 1 ?result \"sp\" ?port1 nil ?port2 nil)
```

However, the similar plot is now generated with the *spm* function and the following expression is created in the Buffer:

```
spm( \"sp\" 1 1 ?port1 nil ?port2 nil)
```


© 2023 Cadence Design Systems, Inc.

totalNoise

Calculates the integrated noise power over a specified bandwidth.

$$totalNoise = \int_{f1}^{f2} V_{out}^2(f)df$$

where $V_{out}^2(f)$ is the noise voltage spectral density in V^2/Hz computed by the `noise`, `pnoise`, `qpnoise` or `hbnoise` analyses.

Spectre keeps a track of the contribution of individual noise sources to the output noise. You can also optionally subtract the specified device instance contributions from the `totalNoise` calculation by using the formula given below.

$$totalNoise = \int_{f1}^{f2} \left[V_{out}^2(f) - \sum_d N_d^2(f) \right] df$$

$N_d(f)$ is the contribution of a specified device to the output noise.

This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `noise`, `pnoise`, `qpnoise`, and `hbnoise`.
  Default value: `noise`
- *Start Frequency*—Specify the start frequency.
- *Stop Frequency*—Specify the stop frequency.
- *Exclude Instances*—Specify a list of instances or instance names. The noise contributed by the instances specified in this field is ignored while calculating the total noise. This is an optional field.

© 2023 Cadence Design Systems, Inc.

Example

This example shows the output generated when you build the following expression in the Buffer:

```
totalNoise("hbnoise" 1k 100k nil )
```

Here,

- *Analysis Name* is `hbnoise`
- *Start Frequency* is `1KHz`
- *Stop Frequency* `100KHz`
- *Exclude Instances* is `nil`, which means that the noise contribution of every instance in the circuit is included in the output

The output generated in the Buffer is `479.7E-9`, which is a scalar value.

Example 2

This example shows the output generated when you build the following expression in the Buffer:

```
totalNoise("noise" 1G 2G list("/R0", "/R1"))
```

Here,

- *Analysis Name* is `noise`
- *Start Frequency* is `1G`
- *Stop Frequency* is `2G`
- *Exclude Instances* are `list("/R0", "/R1")`

In this example, the noise contributed by resistor schematic instances `/R0` and `/R1` is excluded from the `totalNoise` calculation.

Example 3

The following example shows the total noise calculated for the `noise`, `hbnoise`, and `pnoise` analyses with various start and end frequency values. The *Exclude Instances* argument is specified for some expressions that means noise from the given instances is excluded.

© 2023 Cadence Design Systems, Inc.

The results database used or this example is: `../totalNoise/psf`

Details	Value
getData("V_cntl" ?result 'hb_osctp)	5.581
totalNoise("noise" 0 1000 list("/C5" "/R1"))	1.173a
totalNoise("noise" 0 1000000)	1.706f
totalNoise("hbnoise" 0 1000000)	35.47Z
totalNoise("hbnoise" 0 1000000 list("/R1" "/C3"))	35.14Z
totalNoise("pnoise" 0 1000)	165.8f
totalNoise("pnoise" 0 1000000 list("/R1" "/R2"))	165.5p

© 2023 Cadence Design Systems, Inc.

vfreq

Returns the voltage of net at a specified frequency or at all frequencies in the frequency domain.

This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, `qpss`, `pac`, `hbac`, `qpac`, and `ac`.
  Default value: `hb`
- *Net name*— Specify the net name from the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the net name directly from the Schematic window. The selected net name appears in this field.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the net name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

    The selected signal appears in the *Net name* field.
- *Frequency*—Specify the frequency for which you want to plot the results. It is an optional field.
  Valid values: Any integer or floating point number.
  Default Value: `nil`

Example

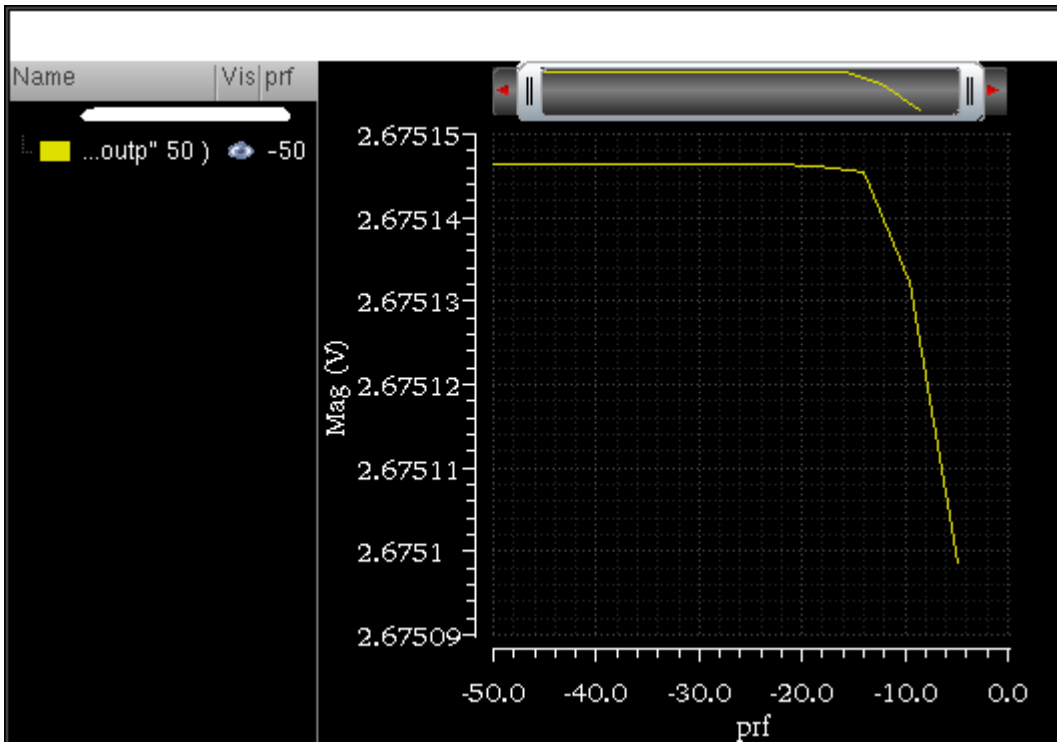
This example shows the output plot generated if you build the following expression in the Buffer:

```
vfreq("hb" "/outp" 50 )
```


© 2023 Cadence Design Systems, Inc.

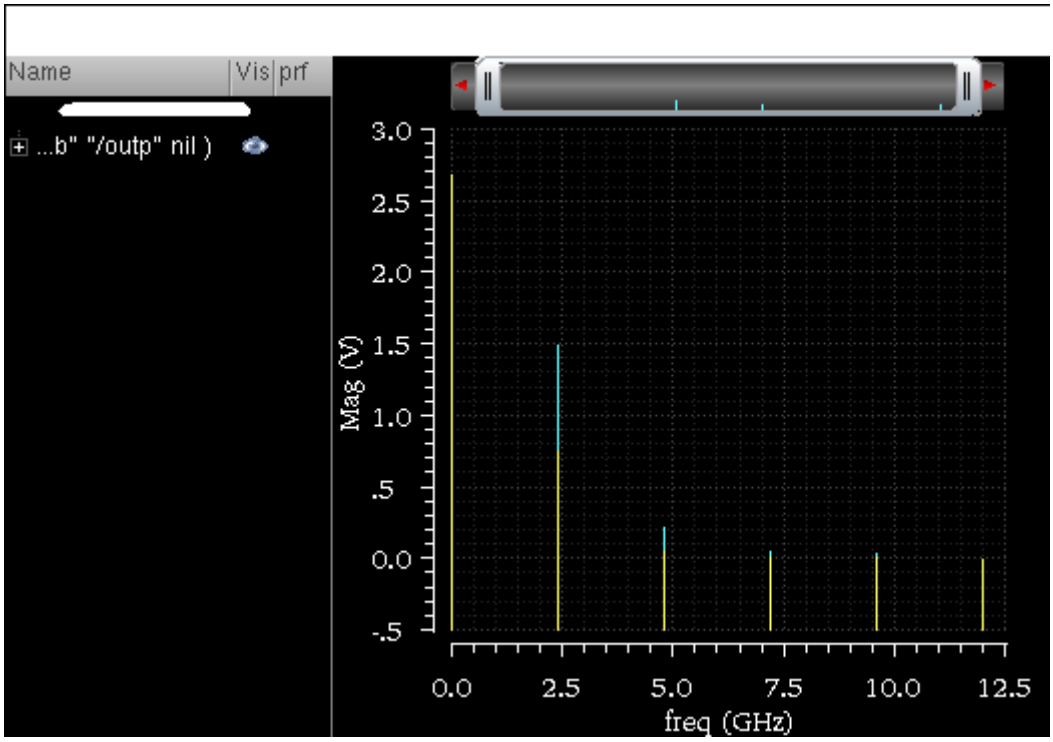
Here,

- *Analysis Name* is `hb`
- *Net name* is `/outp`
- *Frequency* is `50`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Frequency*=`nil`, the voltage on all the frequency points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated voltage of net at a specified frequency or at all frequencies in the frequency domain:

```
v("/RFout" ?result "hbac")
```

However, the similar calculation is now performed with the *vfreq* function and the following expression is created in the Buffer:

```
vfreq('hbac "/RFout")
```


© 2023 Cadence Design Systems, Inc.

vh

Returns the voltage on net at a specified harmonic or at all harmonics in the frequency domain.

This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, `qpss`, `pac`, `hbac`, and `qpac`.
  Default value: `hb`
- *Net name*— Specify the net name from the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the net name directly from the Schematic window. The selected net name appears in this field.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the net name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

    The selected signal appears in the *Net name* field.
- *Harmonic List*—Specify the harmonics for which you want to plot the results. It is an optional field. For analyses, such as `hb`, `pss`, `pac`, and `hbac`, you can add either single harmonic value or available list of harmonic values in this field.
  Valid values: Any integer or a list from the available list of harmonics. You can find the available harmonics by using the `harmonicList` function.
  `harmonicList( [?resultsDir t_resultsDir] [?result S_resultName]`
  Default value: `nil`

© 2023 Cadence Design Systems, Inc.

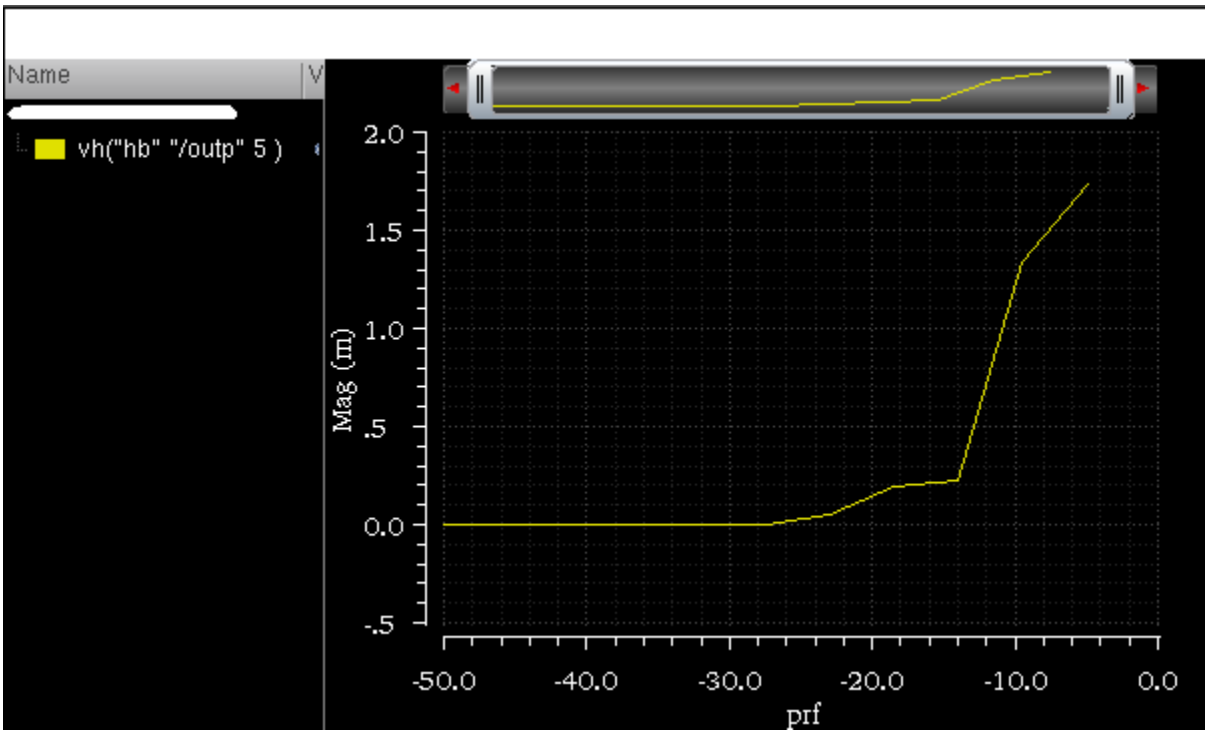
Example

This example shows the output plot generated if you build the following expression in the Buffer:

```
vh("hb" "/outp" 5 )
```

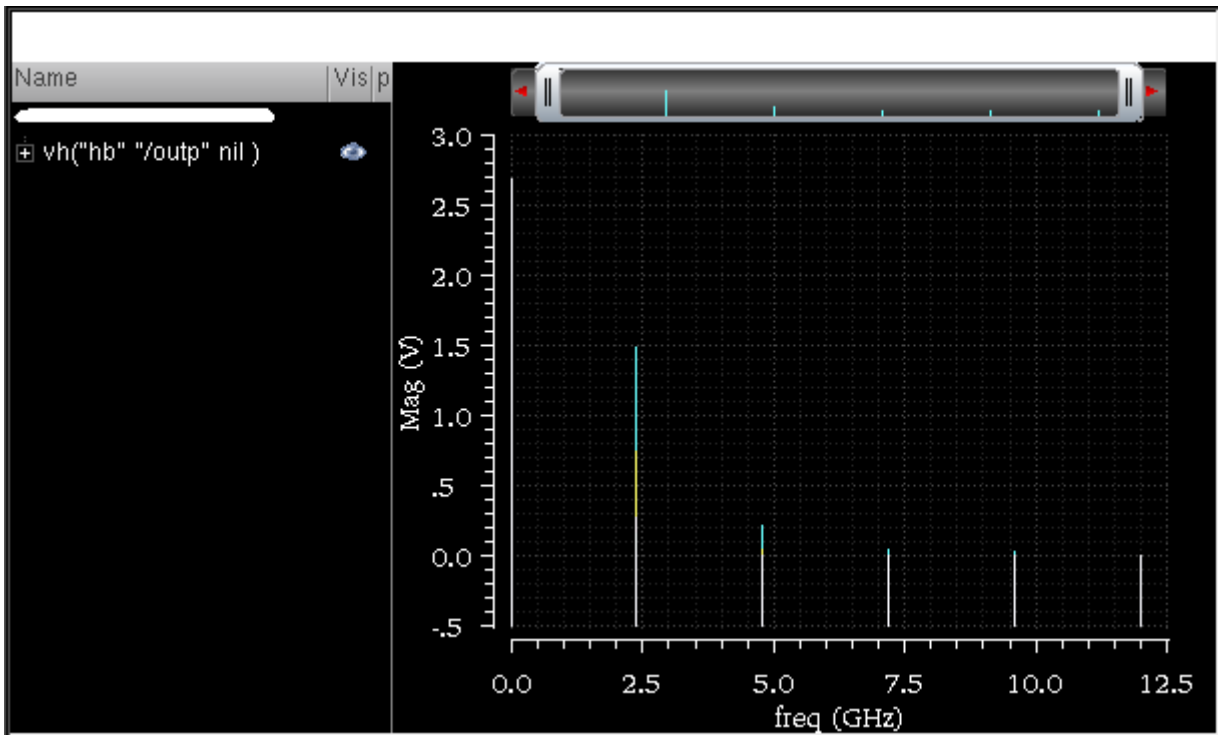
Here,

- *Analysis Name* is `hb`
- *Net name* is `/outp`
- *Harmonic List* is `5`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Harmonic List*=`nil`, the voltage on all the harmonic points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the voltage on net at a specified harmonic or at all harmonics in the frequency domain:

```
v("/RFout" ?result "hb_fd")
```

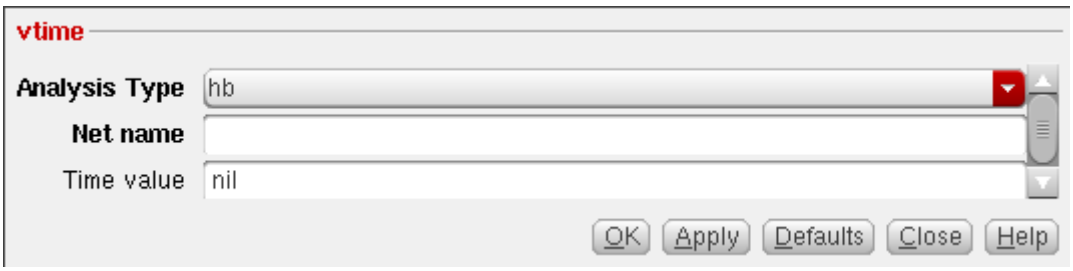
However, the similar calculation is now performed with the *vh* function and the following expression is created in the Buffer:

```
vh('hb "/RFout")
```


© 2023 Cadence Design Systems, Inc.

vtime

Returns the voltage of net at a specified time point or at all time points in the time domain.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list. The available analyses are `hb`, `pss`, and `tran`.
  Default value: `hb`
- *Net name*— Specify the net name from the schematic or signal name from the Results Browser.

  When you open Calculator from ADE, you can select the terminal name directly from the Schematic window. The selected terminal name appears in this field.

  If you are working in the standalone mode and building expressions by using the Results Browser, you can get the terminal name from the Results Browser:

  - Select the *Wave* radio button on the Selection toolbar.
  - Click a current signal in the Results Browser.

    The selected signal appears in the *Terminal name* field.
- *Time value*—Specify the time points for which you want to plot the results. If you specify a time point in this field, the result of the specified time is returned. Otherwise, It is an optional field.
  Valid values: Any integer or floating point number.
  Default value: `nil`.

Example

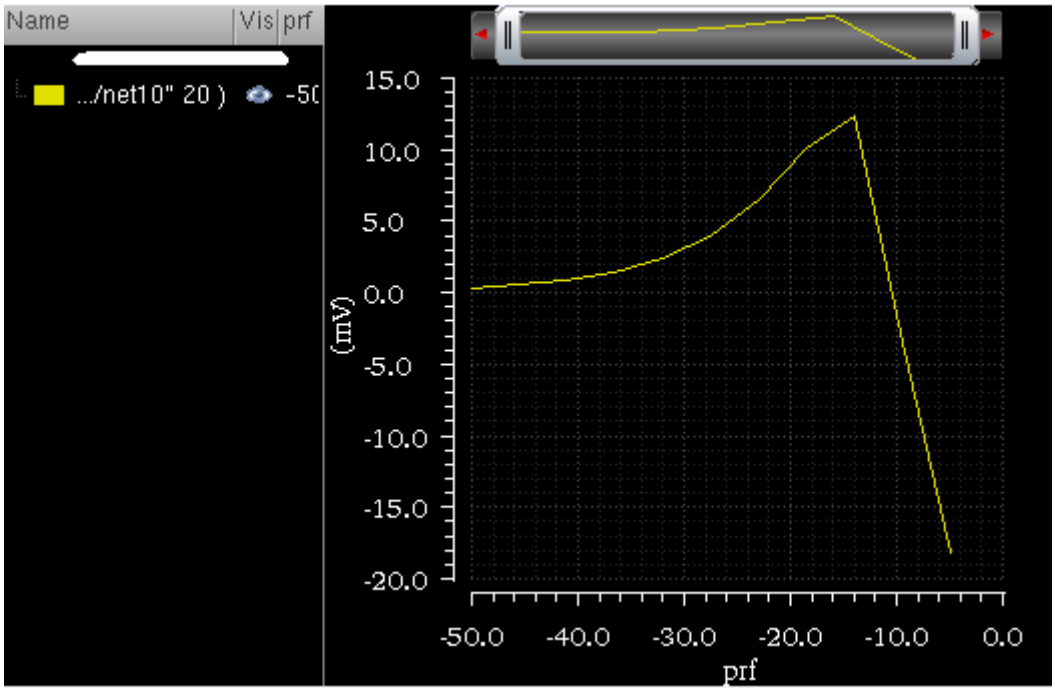
This example shows the output plot generated if you build the following expression in the Buffer:

```
vtime("hb" "/net10" 20)
```


© 2023 Cadence Design Systems, Inc.

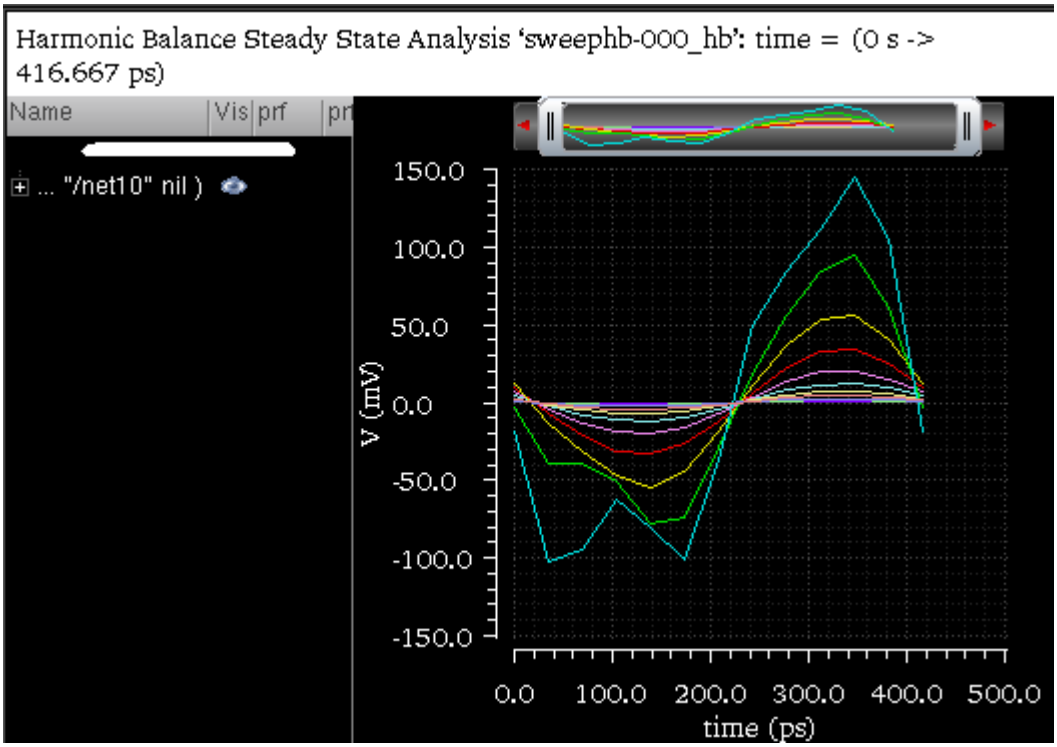
Here,

- *Analysis Type* is `hb`
- *Net name* is `/net10`
- *Time value* is `20`



© 2023 Cadence Design Systems, Inc.

If you build the above expression with *Time value*=`nil`, the voltage on all the time points are returned (as shown in the figure below).



Additional Information

The following expression was created in the Buffer in the previous releases when you calculated the voltage of net at a specified time point or at all time points in the time domain:

```
v("/RFin" ?result "hb_td")
```

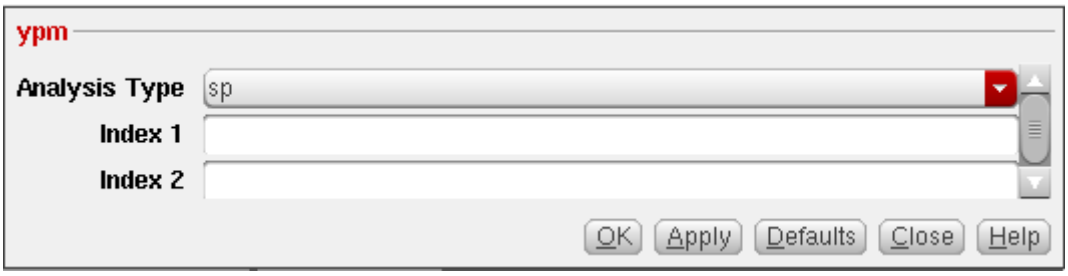
However, the similar calculation is now performed with the *vtime* function and the following expression is created in the Buffer:

```
vtime("hb" "/RFin" nil )
```


© 2023 Cadence Design Systems, Inc.

ypm

Returns the waveform for the y-parameter.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list.
  Valid values: `sp`, `psp`, `qpsp`, and `hbsp`
  Default value: `sp`
- *Index 1*—Specify the port index for sp simulation. By default, this field is set to blank.
  Valid values: Available port index, such as `1`, `2`
- *Index 2*—Specify the port index for sp simulation. By default, this field is set to blank.
  Valid values: Available port index, such as `1`, `2`

Example

This example shows the output plot generated when you build the following expression in the Buffer:

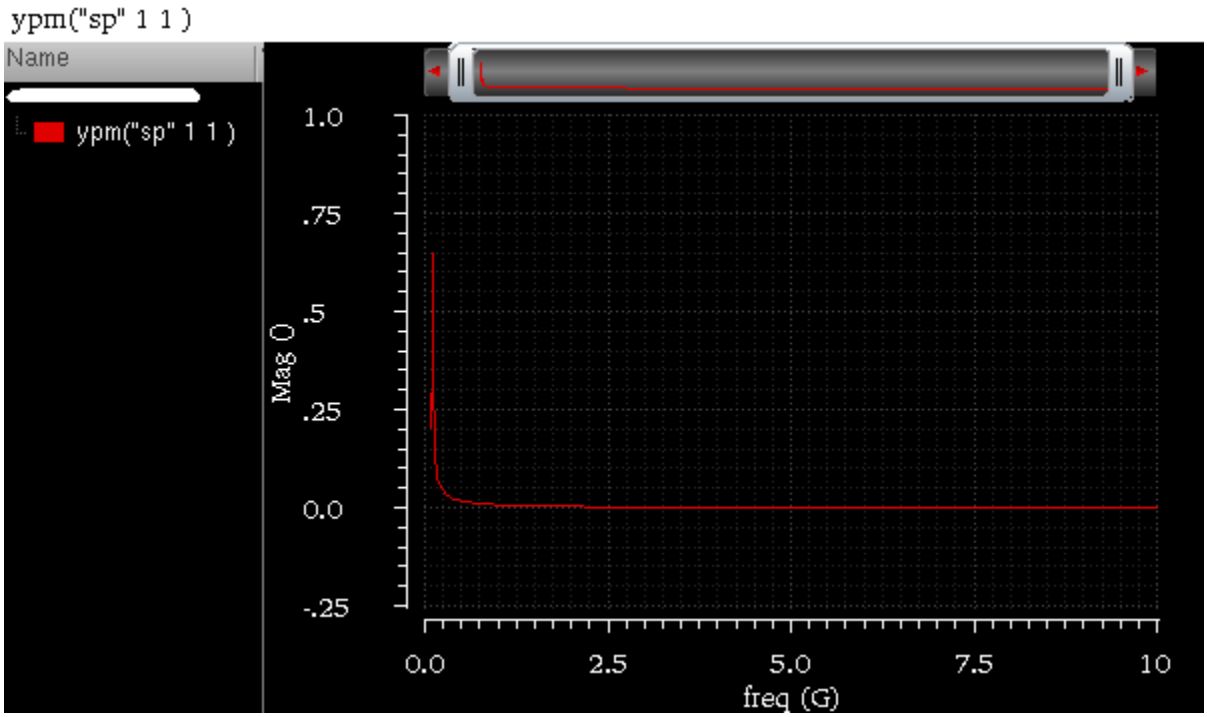
```
ypm("sp" 1 1)
```

Here,

- *Analysis Name* is `sp`
- *Index 1* is `1`

© 2023 Cadence Design Systems, Inc.

- *Index 2* is `1`



Additional Information

The following expression was created in the Buffer in the previous releases when you plotted the waveform for y-parameters:

```
yp(1 1 ?result \"sp\")
```

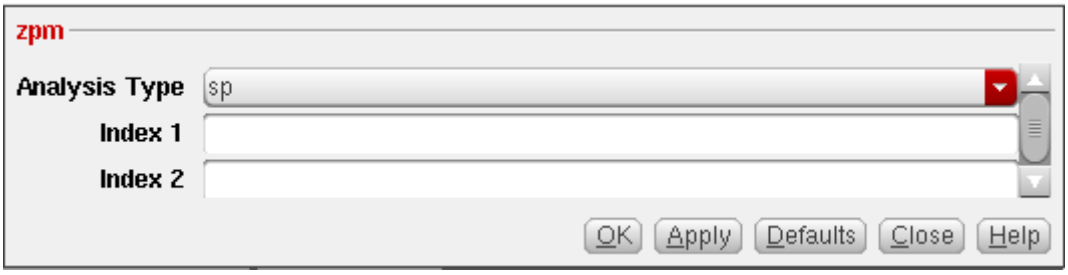
However, the similar plot is now generated with the *ypm* function and the following expression is created in the Buffer:

```
ypm('sp 1 1)
```


© 2023 Cadence Design Systems, Inc.

zpm

Returns the waveform for the z-parameter.



This function includes the following fields:

- *Analysis Type*—Select the analysis type from the drop-down list.
  Valid values: `sp`, `psp`, `qpsp`, and `hbsp`
  Default value: `sp`
- *Index 1*—Specify the port index for sp simulation. By default, this field is set to blank.
  Valid values: Available port index, such as `1`, `2`
- *Index 2*—Specify the port index for sp simulation. By default, this field is set to blank.
  Valid values: Available port index, such as `1`, `2`

Example

This example shows the output plot generated when you build the following expression in the Buffer:

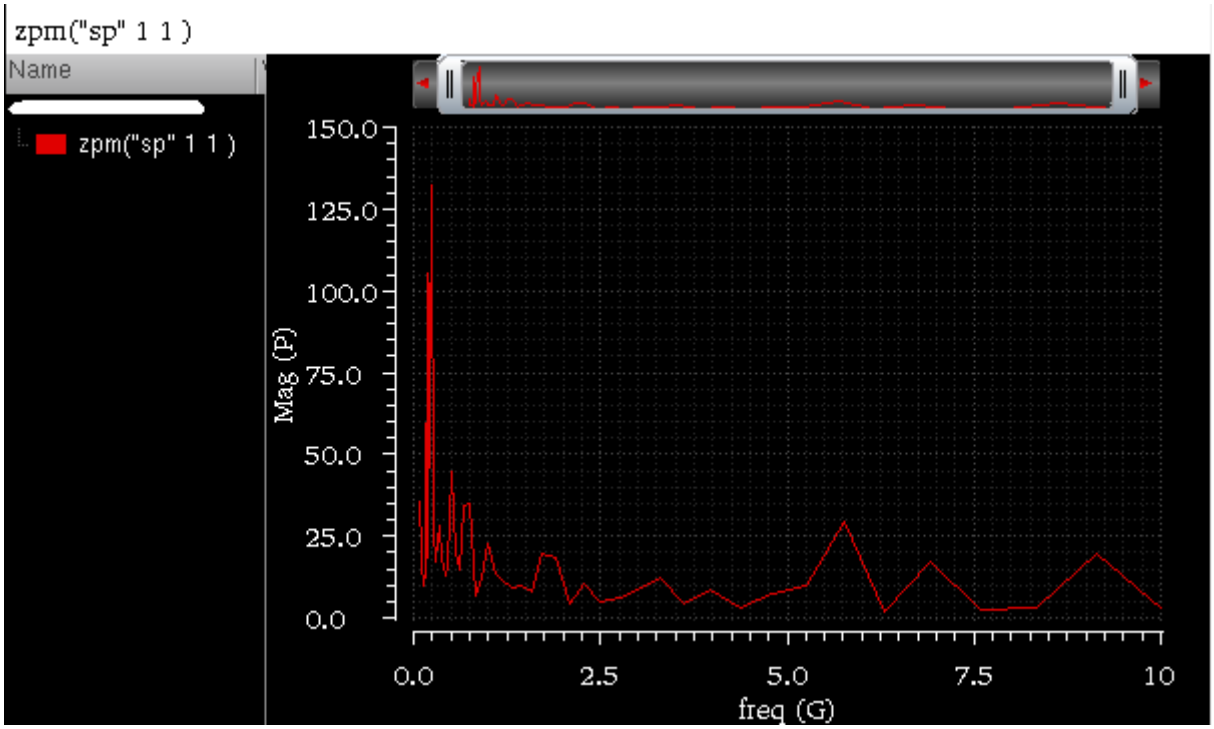
```
zpm("sp" 1 1)
```

Here,

- *Analysis Name* is `sp`
- *Index 1* is `1`

© 2023 Cadence Design Systems, Inc.

- *Index 2* is `1`



Additional Information

The following expression was created in the Buffer in the previous releases when you plotted the waveform for z-parameters:

```
zp(1 1 ?result \"sp\")
```

However, the similar plot is now generated with the *zpm* function and the following expression is created in the Buffer:

```
zpm('sp 1 1)
```

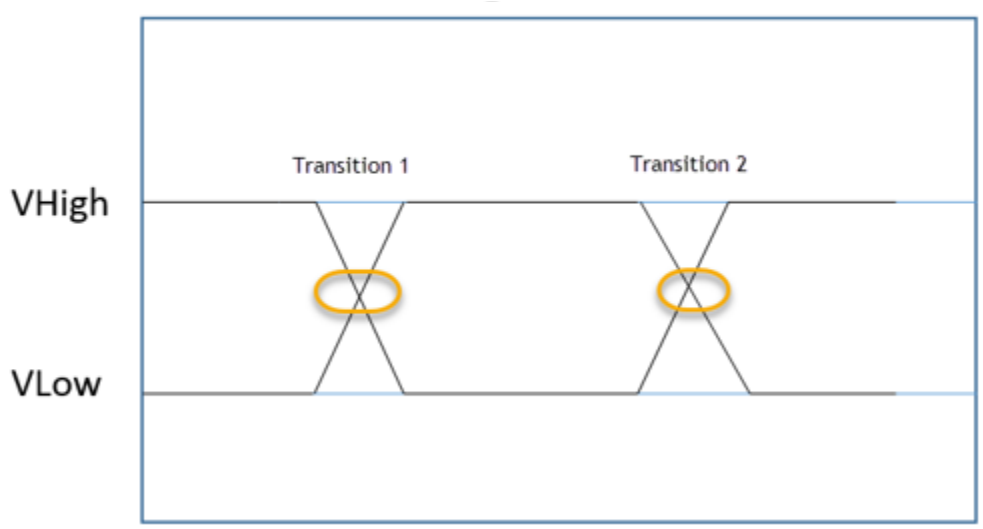

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

C

More About Eye Diagrams

An eye diagram represents the signal split into sections and then overlaid on top of each other. This gives a good indication of the quality of the signals in high speed digital transmissions. The figure below shows an ideal eye diagram.



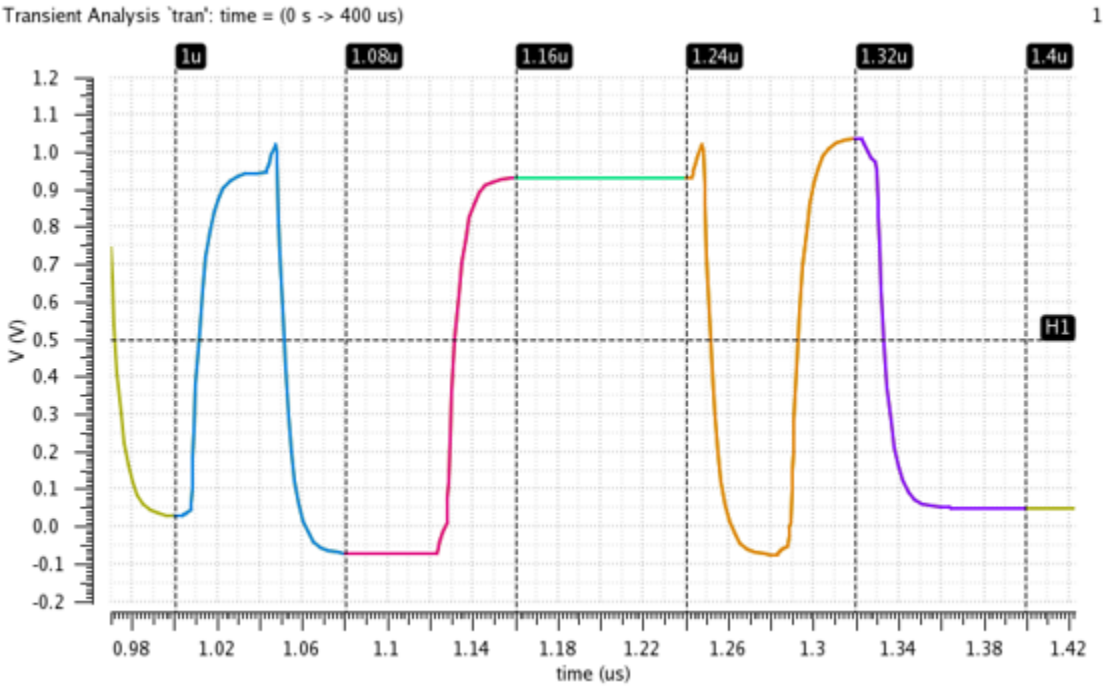
However, signals are affected by noise, process, voltage, temperature, so the signals do not lie on top of each other when they are overlaid.

Eye diagrams are formed by overlaying segments of the input signal over an eye period. A new segment is started for every trigger event. For a regular eye diagram, the trigger period is equal to the eye period.

© 2023 Cadence Design Systems, Inc.

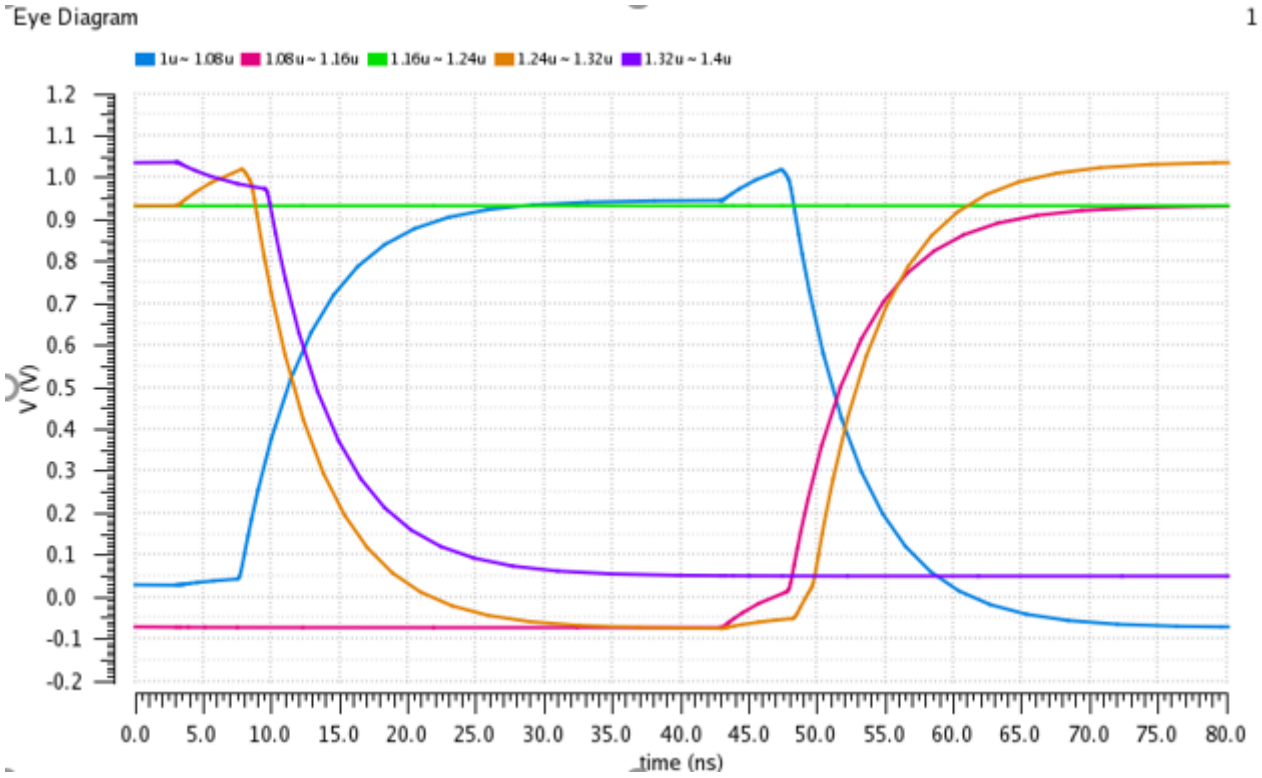
NRZ Eye Measurements

Consider creating an eye diagram using the following signal from time 1 us to 1.4 us. The signal has a unit interval (UI) of 40 ns.



© 2023 Cadence Design Systems, Inc.

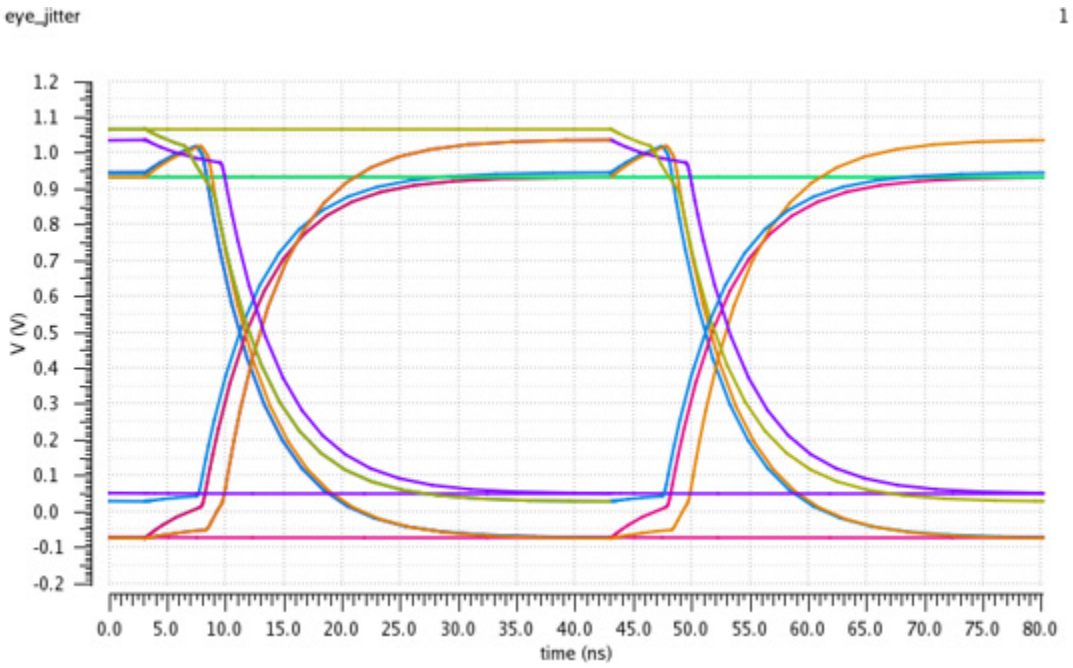
To create a closed eye diagram, the eye period is set to 2 UI or 80 ns. The signal is periodically divided into segments of length 80ns and then overlaid to create the eye diagram.



When internal triggering is used, the trigger period determines how often a new segment is created. When a trigger period of 1 unit interval and an eye period of 2 unit intervals, a more

© 2023 Cadence Design Systems, Inc.

detailed eye diagram can be obtained such that each cross appears in both transition areas. The following eye diagram is created with a trigger period of 40 ns and an eye period of 80 ns.



The metrics that can be calculated are based on performing statistical analysis of the eye diagram. The signal being analyzed can be thought of as a sequence of 1's and 0's, with transitions between the two logic levels. As a result, it is of interest to analyze the distribution of the timing of the transitions between 0 and 1 levels, and also to analyze the distribution of the signal levels of the parts of the eye diagram representing 0 and 1 values. There are various fields on the form that need to be completed to perform the analysis. In addition to the descriptions in the table below, the explanation of the measurements performed will clarify the meaning of the fields.

Field	Description
Threshold	The Y-axis level (such as voltage) which represents the switching threshold between level 0 and level 1 of the signal - typically half the signal range. This is used to compute statistical information about the times (relative to the beginning of the eye diagram) at which the signal crosses the threshold level

© 2023 Cadence Design Systems, Inc.

Field	Description
Level 0 X-range	The region of time within the eye diagram period, which together with the level 0 Y-range is used to identify a set of y values to analyze. The x-range and y-range specify a box on the eye diagram containing the y-values which will be statistically analyzed. By analyzing the distribution of the y values within this region, information on the mean and standard deviation of the parts of the signal corresponding to a digital 0 (or low) can be obtained. This may be specified as a percentage range of the x-axis, or a specific time range (if xval is chosen).
Level 0 Y-range	The region of the eye diagram's y-axis which together with the level 0 x-range is used to identify a set of y-values to analyze. The x-range and y-range specify a box on the eye diagram containing the y-values which will be statistically analyzed. By analyzing the distribution of the y-values within this region, information on the mean and standard deviation of the parts of the signal corresponding to a digital 0 (or low) can be obtained. This can be specified as a percentage of the y range of the data, or as a range of absolute values.
Level 1 X-range	Similar to Level 0 x-range, except that the region specified by this together with level 1 y-range is used to analyze the digital 1 (or high) levels in the signal. Typically Level 1 and Level 0 x ranges are the same.
No of Bins	This is the number of bins used to form the horizontal (threshold crossing times) and vertical (amplitude variation) histograms.

© 2023 Cadence Design Systems, Inc.

Field	Description
Sampling Interval	If left blank, the raw data within the level 1 and level 0 regions are used to analyze the amplitude variation of the signal. This means that there is some sensitivity to the spacing between the data points in the signal (caused by variable time steps in the simulator); if there is some clustering of points in a portion of the curve, it may skew the distribution. This field allows you to specify a regular sampling interval, which will mean that the analysis is done by breaking the interval into even time points, thus removing the effect of the simulator's variable time step. If used, this should be small enough not to miss significant local variation of the signal
Plot BER Curve	Plots the BER/bathtub curve of the jitter

Assumptions

The following assumption have been made while calculating the advanced option values:

- The eye diagram period should be two unit intervals.
- The trigger period is one unit interval or signal triggered.

Measurements to be Computed

There are two groups of measurements that the Advanced Analysis computes — based on horizontal histograms and based on vertical histograms. In the computations below, 3sigma is used for the extent of the data, as this represents 99.7% of the values in a normal distribution.

Level 0 Mean is the mean of the y-values within the Level 0 Region.

Level 0 Stddev is the standard deviation of the y-values within the Level 0 Region.

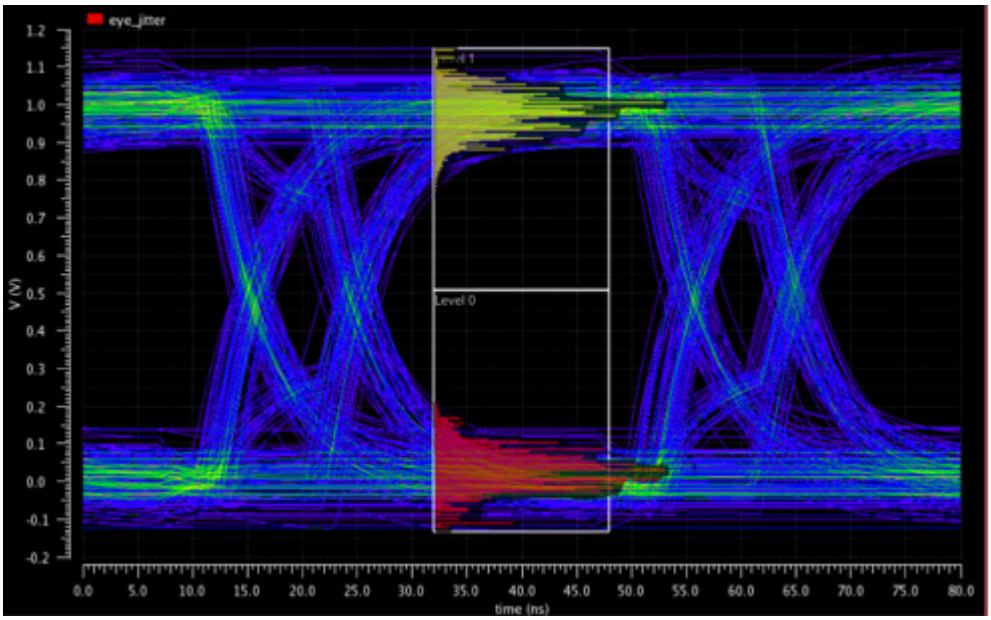
Level 1 Mean is the mean of the y-values within the Level 1 Region.

Level 1 Stddev is the standard deviation of the y-values within the Level 1 Region.

These values are shown on a real eye diagram with many signals transitioning. The risetime, falltime, and jitter of each signal impacts the plot, which can be obtained from interference, PVT, and some jitter is just random. Notice that the transition areas are highlighted in the

© 2023 Cadence Design Systems, Inc.

figure below. The level 0 (in red) and level 1 (in yellow) annotation shows the voltage high and lows areas. The histograms that represent the signals in level 1 and level 0 areas have been superimposed.



The following measurements have been calculated. Each measurement will be explained in the steps below:

Outputs

Measurement	Value
eyeDiagram(v("jitter" ?result "tran-tran") 0.5u 4...	
Threshold crossing Average (crossing time)	39.29n (s)
Threshold crossing stddev	20n (s)
Level 0 mean	14.96m (V)
Level 0 stddev	53.63m (V)
Level 1 mean	993.9m (V)
Level 1 stddev	53.08m (V)
Eye Amplitude	978.9m (V)
Eye Height	658.8m (V)
Eye Width	34.11n (s)
Eye S/N	9.174
Eye Rise Time	6.9n (s)
Eye Fall Time	6.816n (s)
Random Jitter (left)	983.5p (s)
Random Jitter (right)	1.001n (s)
Deterministic Jitter	0 (s)

© 2023 Cadence Design Systems, Inc.

Eye Amplitude

Eye Amplitude is the mean to the mean amplitude of the eye, is calculated as follows:

Eye Amplitude = Level1 Mean - Level 0 Mean

= 993.9m - 14.96m

= 978.9mV

Signal to Noise Ratio

The signal to noise of the eye, is calculated as follows:

S/N = (Level1 Mean - Level 0 Mean) / (Level 1 stddev + Level 0 stddev)

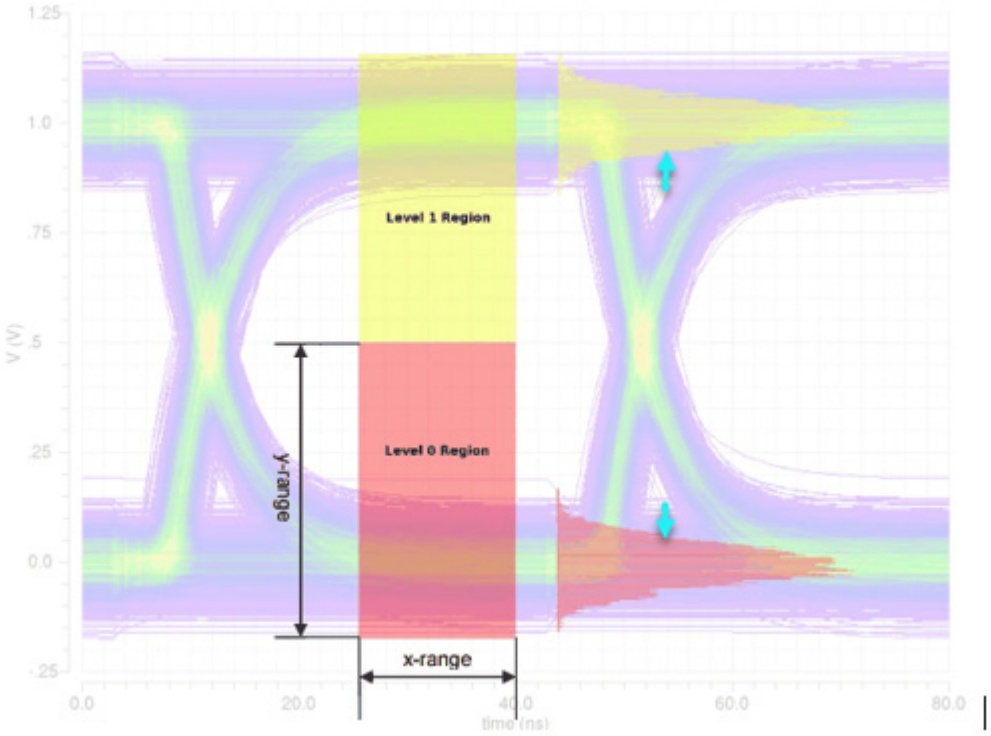
= (993.9m -14.96m) / (53.08m+53.63m)

= 9.174

© 2023 Cadence Design Systems, Inc.

Eye Height

Eye Height is also known as the vertical eye width. Consider the vertical histograms. There are two transition areas on the eye diagram, as shown on the figure above. Then, if you take the vertical histograms and see them flipped over the eye.



Eye Height is calculated with the following formula:

eyeHeight = (MeanLevel1-3*stddevLevel1) - (MeanLevel0+3*stddevLevel0)

= (993.9m-3*53.08m) - (14.96m-3*53.63m)

= 658.8mV

The level 0 and level 1 regions are those specified by the X and Y ranges in the Eye Diagram assistant. All the Y-points in the waveform within these regions are used to perform the statistical analysis, unless they have been sampled first according to the Sampling Interval field. In the plotted vertical histograms in the above diagram, X-axis represents the amplitude variation.

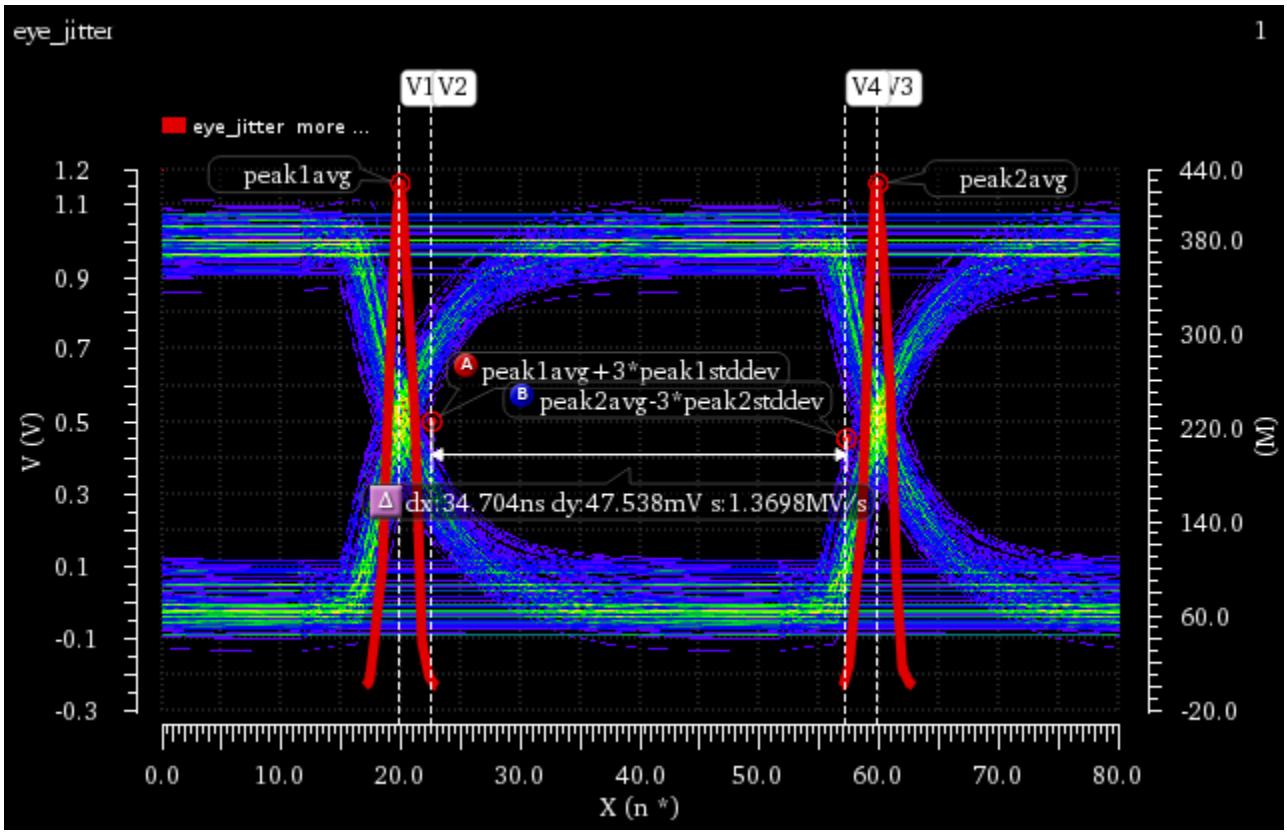
© 2023 Cadence Design Systems, Inc.

Eye Width

The eye width is calculated using the peaks of the horizontal histograms. The average and standard deviations of these peaks are used to calculate the eye width.

eyeWidth = (peak2avg - (3*peak2stddev)) - (peak1avg + (3*peak1stddev))

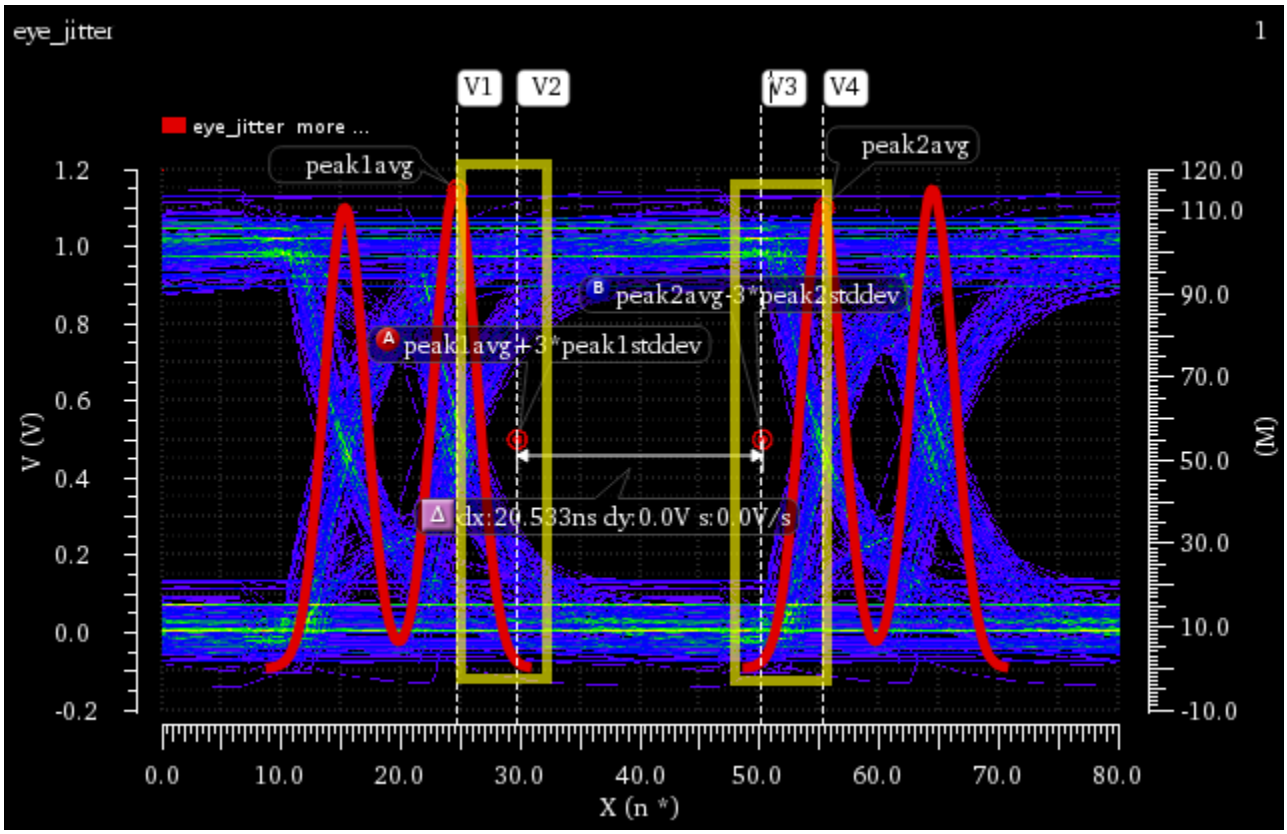
When no deterministic jitter is present, the distribution of the crossings forms a single normal distribution. The probability of a point being within 3*stddev of the mean is 99.7%. The eye width is calculated using the points shown in the markers below.



When deterministic jitter is detected, the crossings on each side are split between the two bell curves instead of a single one. In this case, the mean and standard deviations of the bell curves closest to the center from each side is used to calculate the eye width. The mean and

© 2023 Cadence Design Systems, Inc.

standard deviation are calculated from the points, shown by markers within the rectangles in the figure below.



Eye Rise Time

Two thresholds are created at the 20% and 80% points between the level 0 mean and level 1 mean. At each of these two thresholds, a horizontal histogram is computed (an analysis of the times of the crossing points of these two thresholds), and the resulting rise time is the difference in the mean crossing time at each of these two thresholds.

The calculations for the threshold crossing assume that only a single transition is used; therefore, one additional metric, eye risetime, is computed. In the eye risetime, two thresholds are created at 20% and 80% points between level 0 mean and level 1 mean. At each of these two thresholds, a horizontal histogram is computed (that indicates an analysis of the times of the crossing points of these two thresholds), and the resulting risetime is the difference in the mean crossing time at each of these two thresholds.

Right-click the Level0Mean Measurement and choose *Send to ADE*. This example uses the CIW to calculate the rise and fall times.

© 2023 Cadence Design Systems, Inc.

```
waveform=eyeDiagram(v("jitter" ?result "tran-tran") 0.5u 40.00u 80n ?autoCenter t)
Level0Mean=eyeMeasurement(eyeDiagram(v("jitter" ?result "tran-tran") 5e-07 4e-05
8e-08 ?autoCenter t) 5e-07 4e-05 8e-08 0.5 nil t 40.0 0.0 t 60.0 50.0 t 40.0 50.0
t 60.0 100.0 100.0 "level0Mean")

EyeAmp=eyeMeasurement(eyeDiagram(v("jitter" ?result "tran-tran") 5e-07 4e-05 8e-08
?autoCenter t) 5e-07 4e-05 8e-08 0.5 nil t 40.0 0.0 t 60.0 50.0 t 40.0 50.0 t 60.0
100.0 100.0 "amplitude")

start=0.5u

end=40u

period=80n

riseFallTimePercentLow=20

riseFallTimePercentHigh = 80

RisingLow=awvEyeCross(waveform start end period Level0Mean +
EyeAmp*riseFallTimePercentLow/100) "rising")

RisingHigh=awvEyeCross(waveform start end period Level0Mean +
(EyeAmp*riseFallTimePercentHigh/100) "rising")

riseTime=average(RisingHigh) - average(RisingLow)

=6.8997ns

=6.9ns
```

Eye Fall Time

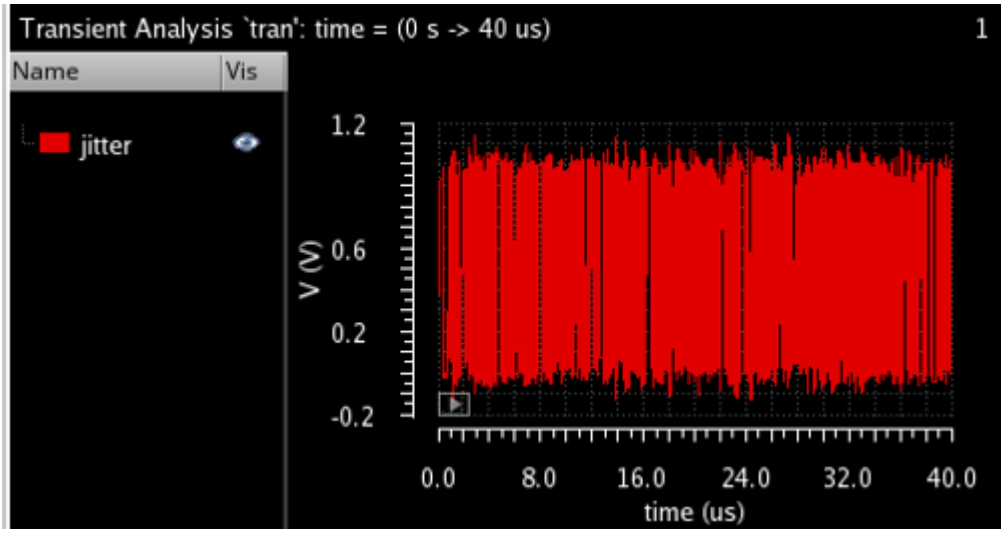
Similarly for fallTime:

```
FallingLow=awvEyeCross(waveform start end period Level0Mean +
(EyeAmp*riseFallTimePercentLow/100) "falling") FallingHigh=awvEyeCross(waveform
start end period Level0Mean + (EyeAmp*riseFallTimePercentHigh/100) "falling")

fallTime=average(FallingLow)-average(FallingHigh)

=6.816254ns

=6.816ns
```


© 2023 Cadence Design Systems, Inc.

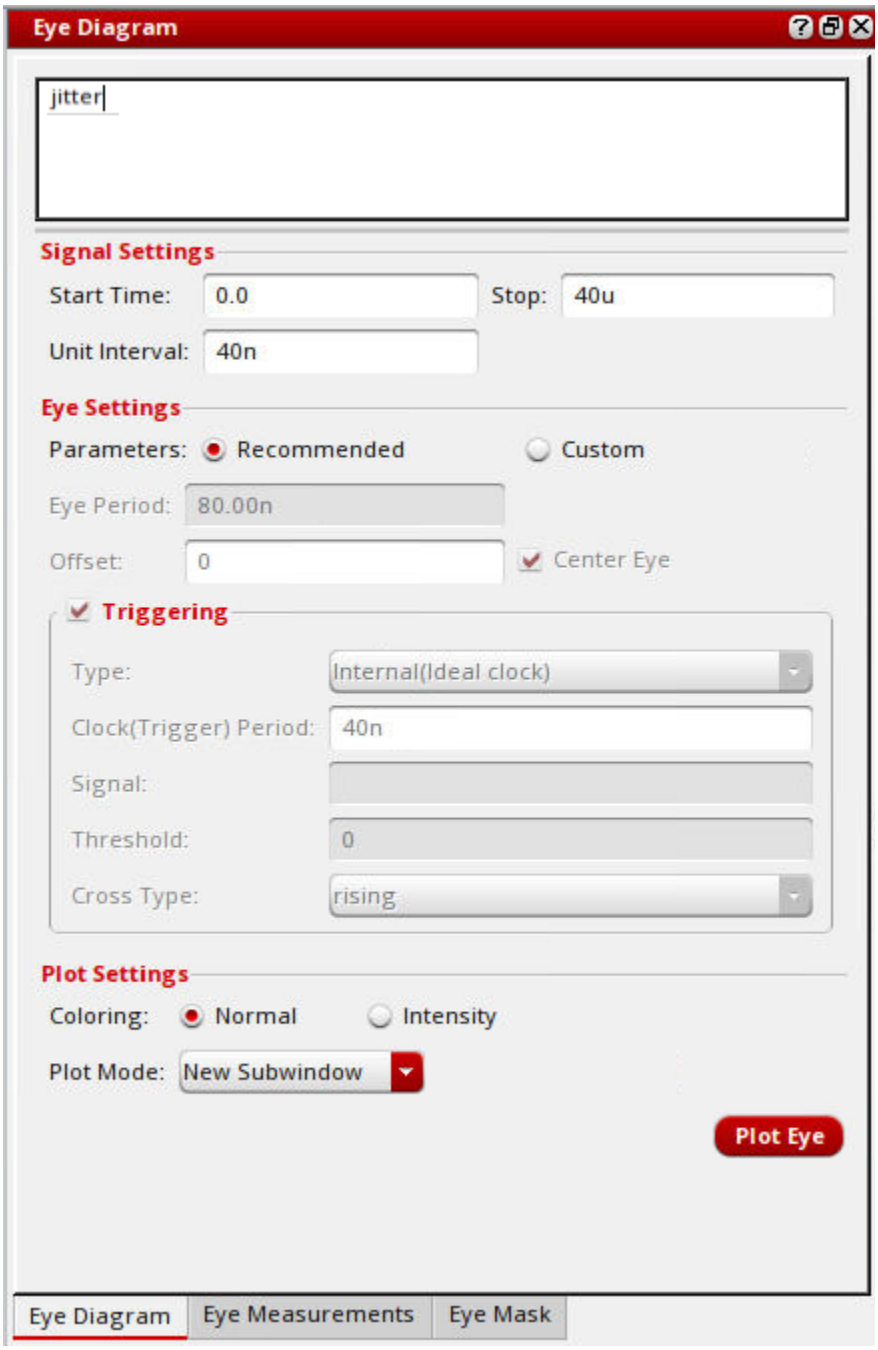
Example

Consider the following example to understand the random and deterministic jitter.



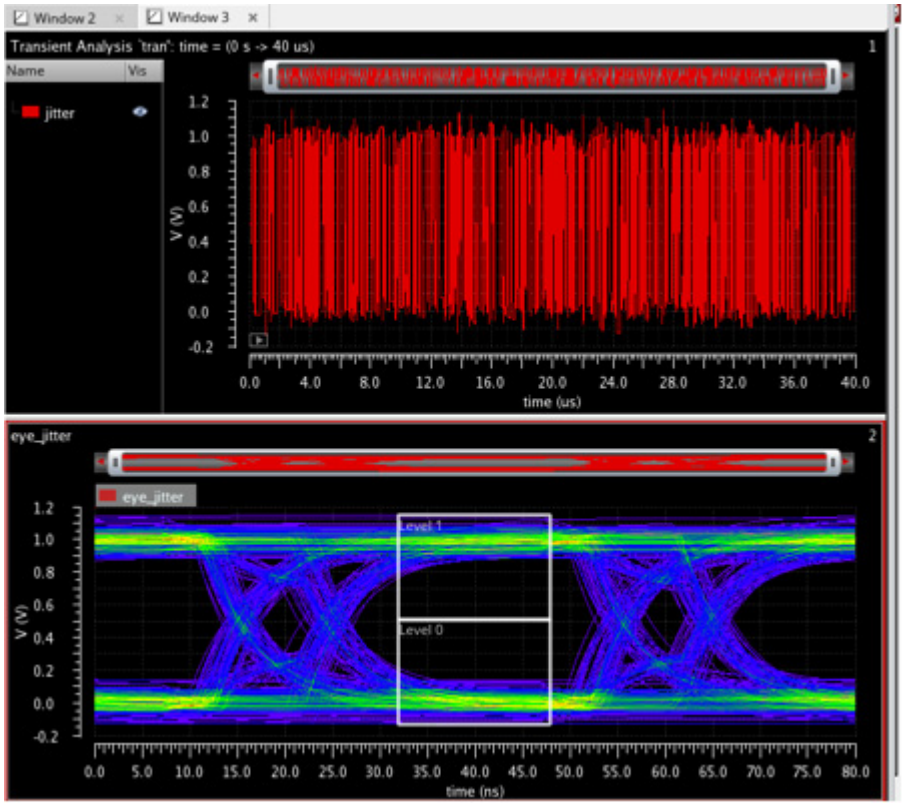
© 2023 Cadence Design Systems, Inc.

Specify the following values in the Eye Diagram assistant:



© 2023 Cadence Design Systems, Inc.

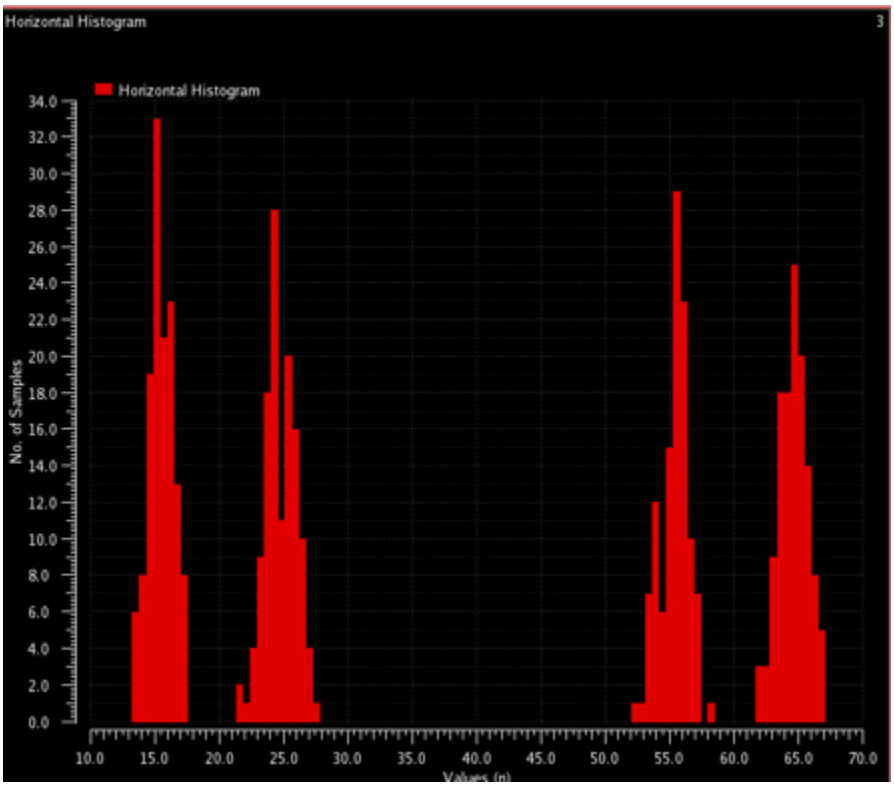
Now, plot the eye diagram. The following plots appear.



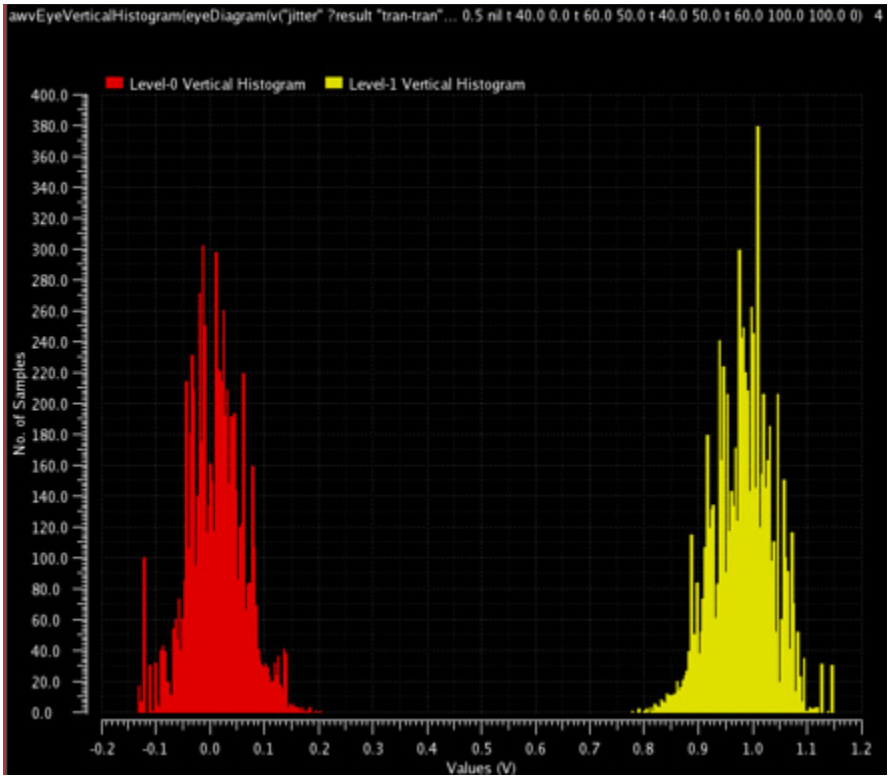
Level 1 and Level 0 histogram areas can be annotated:

© 2023 Cadence Design Systems, Inc.

Plot the histograms and measurements:

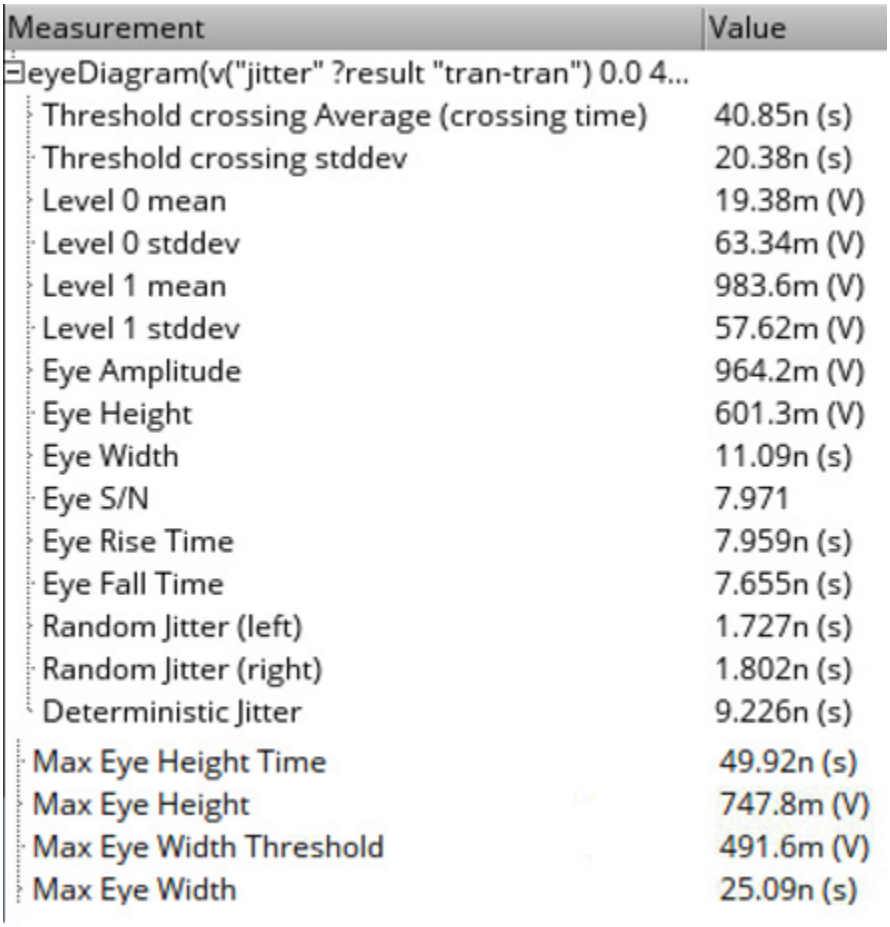


© 2023 Cadence Design Systems, Inc.



© 2023 Cadence Design Systems, Inc.

The random jitter and deterministic jitter values are shown in the Measurement section, as shown in the figure below:



Measurement	Value
eyeDiagram(v("jitter" ?result "tran-tran") 0.0 4...	
Threshold crossing Average (crossing time)	40.85n (s)
Threshold crossing stddev	20.38n (s)
Level 0 mean	19.38m (V)
Level 0 stddev	63.34m (V)
Level 1 mean	983.6m (V)
Level 1 stddev	57.62m (V)
Eye Amplitude	964.2m (V)
Eye Height	601.3m (V)
Eye Width	11.09n (s)
Eye S/N	7.971
Eye Rise Time	7.959n (s)
Eye Fall Time	7.655n (s)
Random Jitter (left)	1.727n (s)
Random Jitter (right)	1.802n (s)
Deterministic Jitter	9.226n (s)
Max Eye Height Time	49.92n (s)
Max Eye Height	747.8m (V)
Max Eye Width Threshold	491.6m (V)
Max Eye Width	25.09n (s)

Threshold crossing Average and stddev

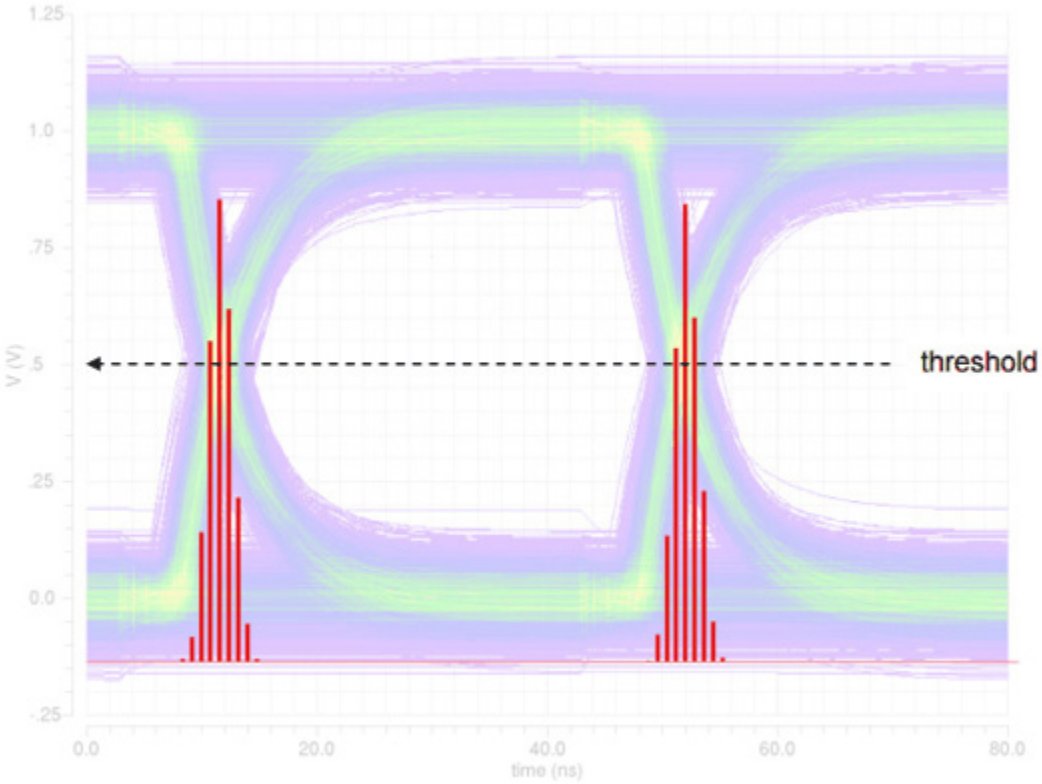
Threshold crossing stddev

Measures the jitter of the signal. It is accurately computed only when there is a single transition region in the eye diagram because it is analyzed over the entire period, and therefore, it gives the overall standard deviation of both transition regions (as shown in the figure below).

Threshold crossing average

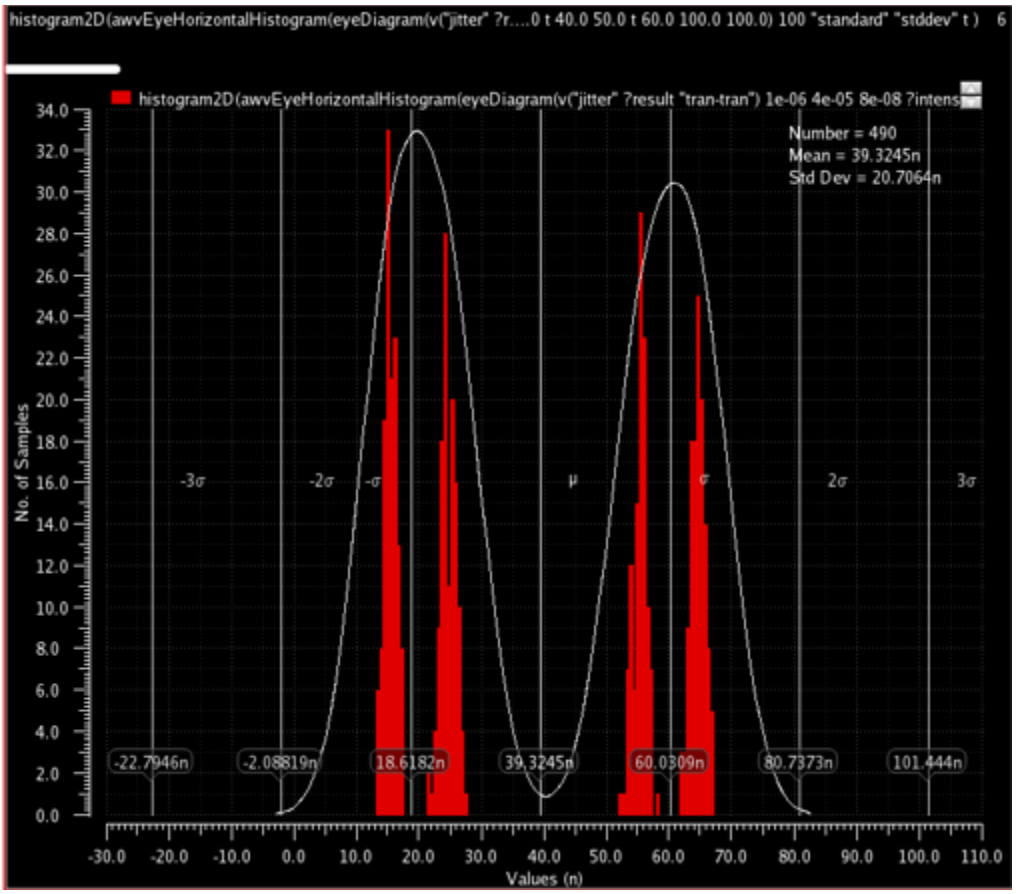
© 2023 Cadence Design Systems, Inc.

Similar to the threshold crossing stddev, it is computed over the entire period.



© 2023 Cadence Design Systems, Inc.

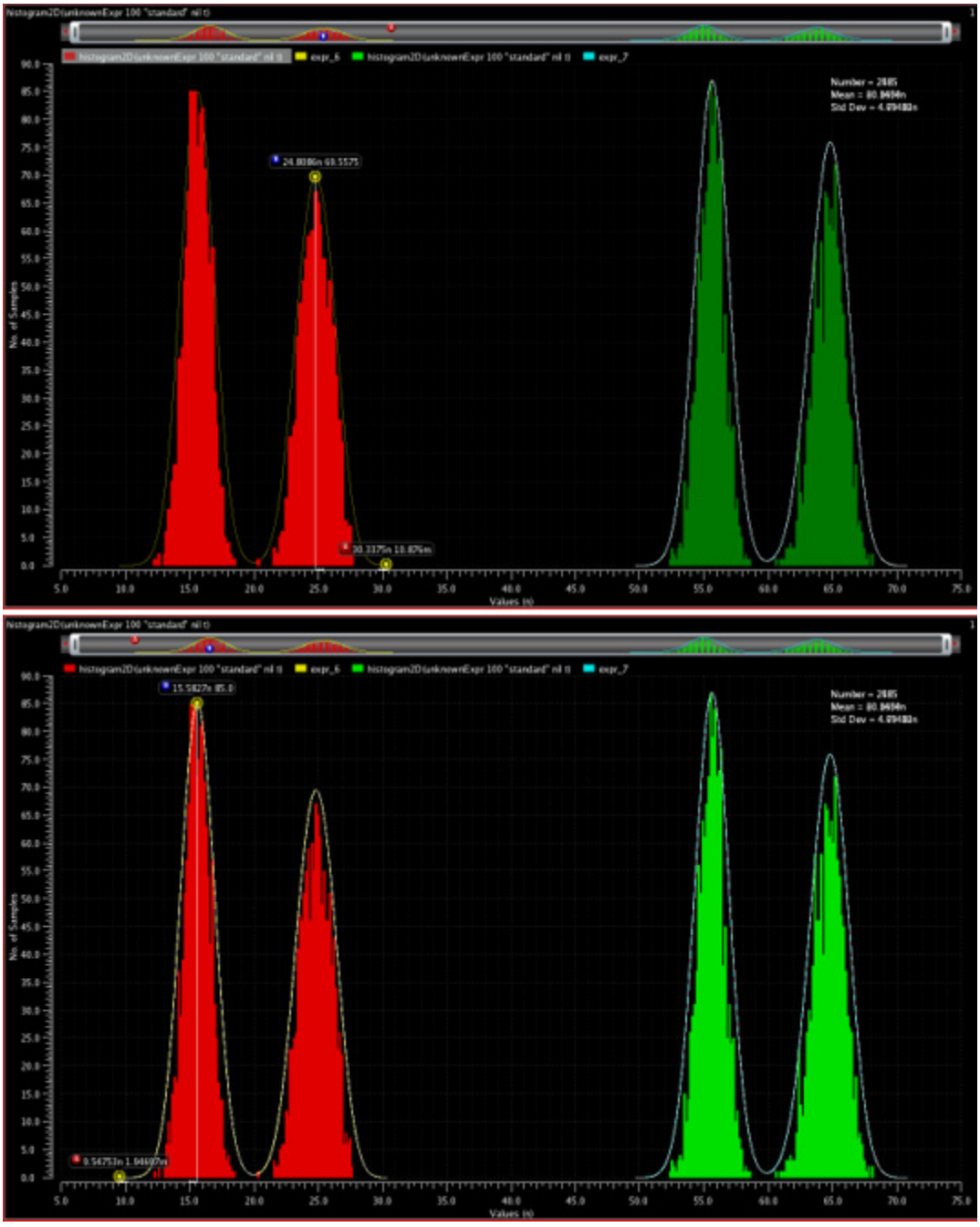
The Threshold crossing Average and stddev values are measured by using the `histogram2D` function on the horizontal histograms. The Mean and Std Dev values annotated on the plot are the same as those in the Measurements.



© 2023 Cadence Design Systems, Inc.

Deterministic and Random Jitter

The following histograms are created for each side and density estimation is applied to fit each histogram to a distribution J(x), as shown in the figure below:



© 2023 Cadence Design Systems, Inc.

If deterministic jitter is present, each side's distribution is expected to have two peaks, and the deterministic jitter is the time difference of the peaks.

Random jitter is the standard deviation of the tails of the distribution.

Deterministic and random jitter are calculated using the histograms of the crossing points. For random jitter (left), the crossing times on the left half of the eye diagram (assuming 2 UI and centered) are sampled into a histogram (the red one). Kernel density estimation is then used to approximate a smooth distribution for the histogram. The histograms are similar to the horizontal histogram but isolated so in order to do the curve fitting. The deterministic jitter is the distance between the peaks (B markers on top and bottom picture). The random jitter is the standard deviation of the distribution. It is calculated as:

$$\sqrt{\sum_i p_i (x_i - \mu)^2}$$

where u is the mean (B marker), x is the x-values from the A marker to B marker (tail of the distribution) and p is the corresponding height. This is repeated for the other tail (A to B in the figure below). The average stddev of the two tails is the output value of Random Jitter (left). The same is repeated for the green histogram to calculate Random Jitter (right). The output Deterministic Jitter is the average of the deterministic jitter of red and green.

BER curves are calculated as follows:

Left BER plot—Integrate left distribution from x to infinity.

$$BER_L(x) = \int_x^{\infty} J_L(x')\, dx'$$

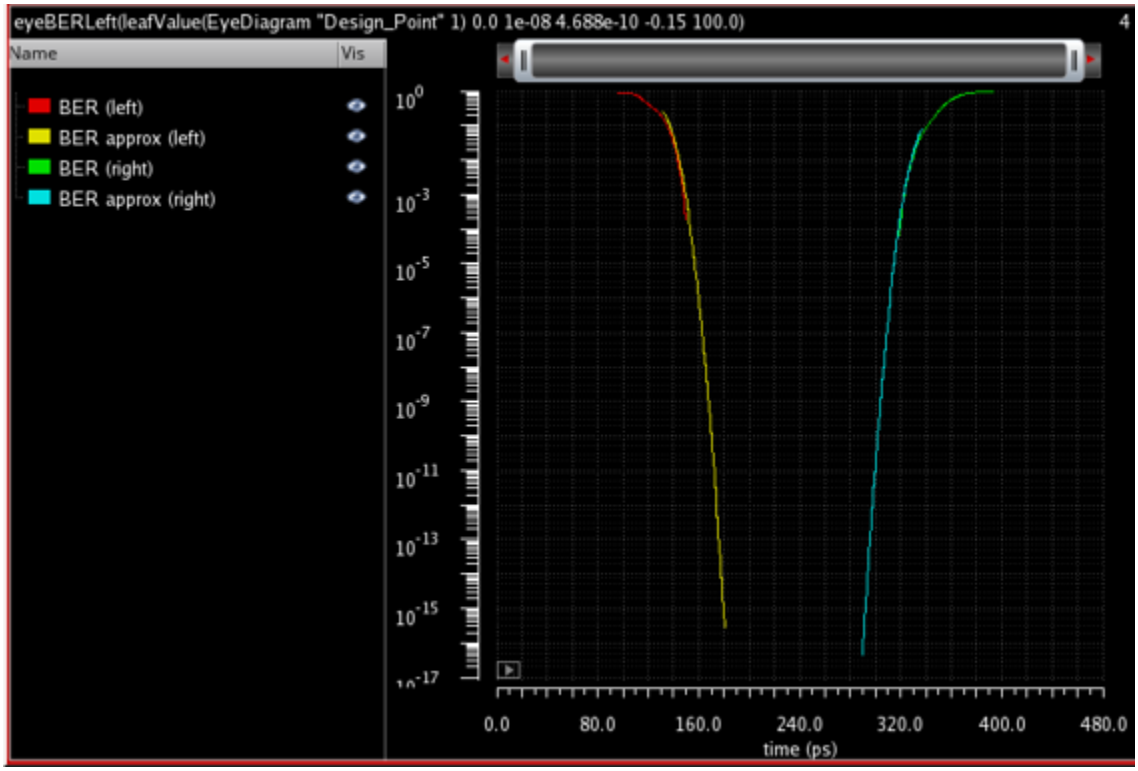
The mean and standard deviation of the right-side tail of the left distribution is used to estimate beyond the range of the histogram.

Right BER plot—Integrate right distribution -infinity to x.

$$BER_R(x) = \int_{-\infty}^{x} J_R(x)\, dx$$

© 2023 Cadence Design Systems, Inc.

The mean and standard deviation of the left-side tail of the right distribution is used to estimate beyond the range of the histogram.



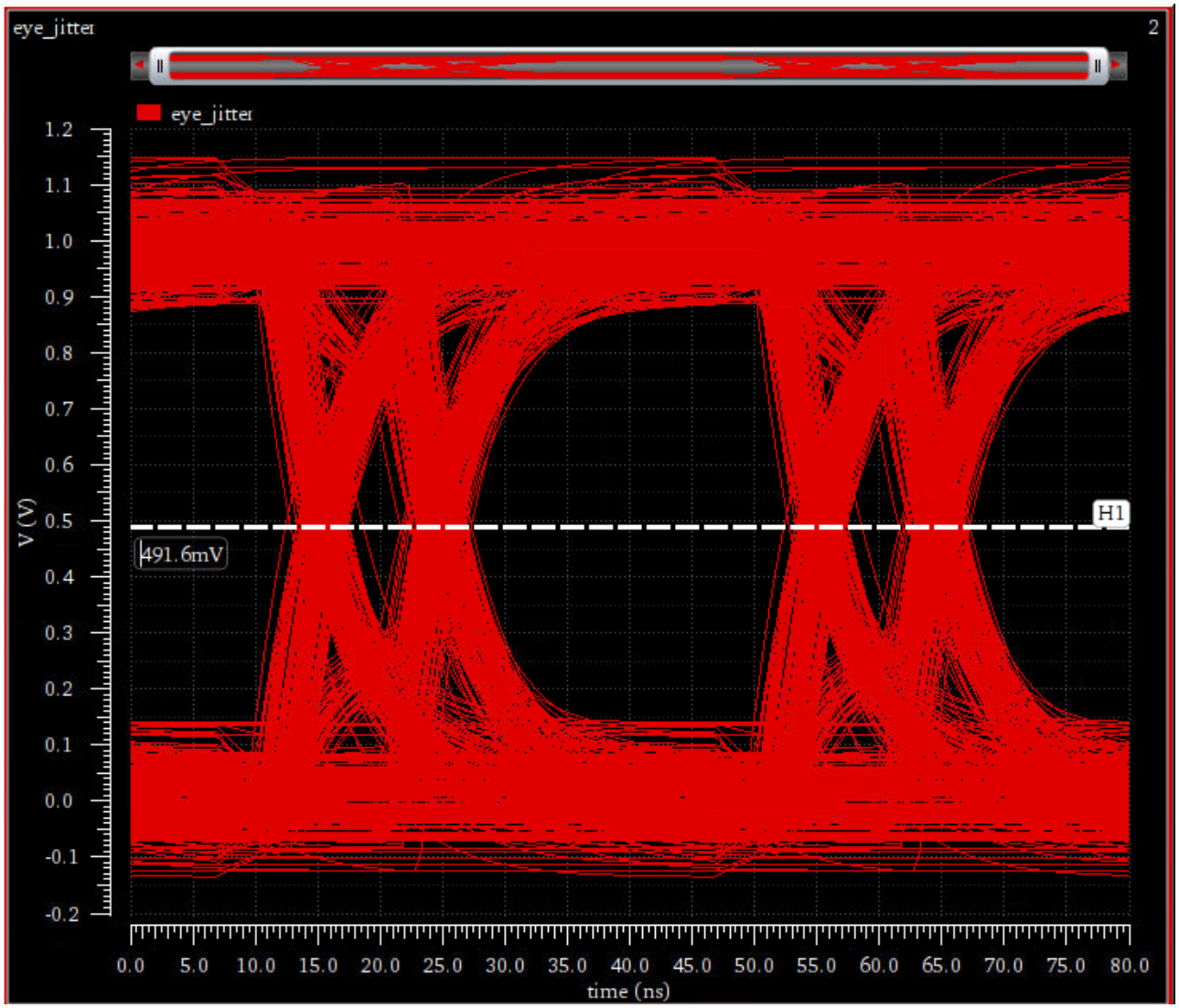
Additional Measurements

- `Max Eye Height and Max Eye Height Time`

  Max Eye Height calculates the largest vertical opening between 25% to 75% of the eye period above and below the threshold. The Eye Height Time is the time where the

© 2023 Cadence Design Systems, Inc.

maximum eye height occurs. For example, the Maximum Eye Width is given as `25.09ns` at a threshold of `491.6mV` in the eye diagram below.

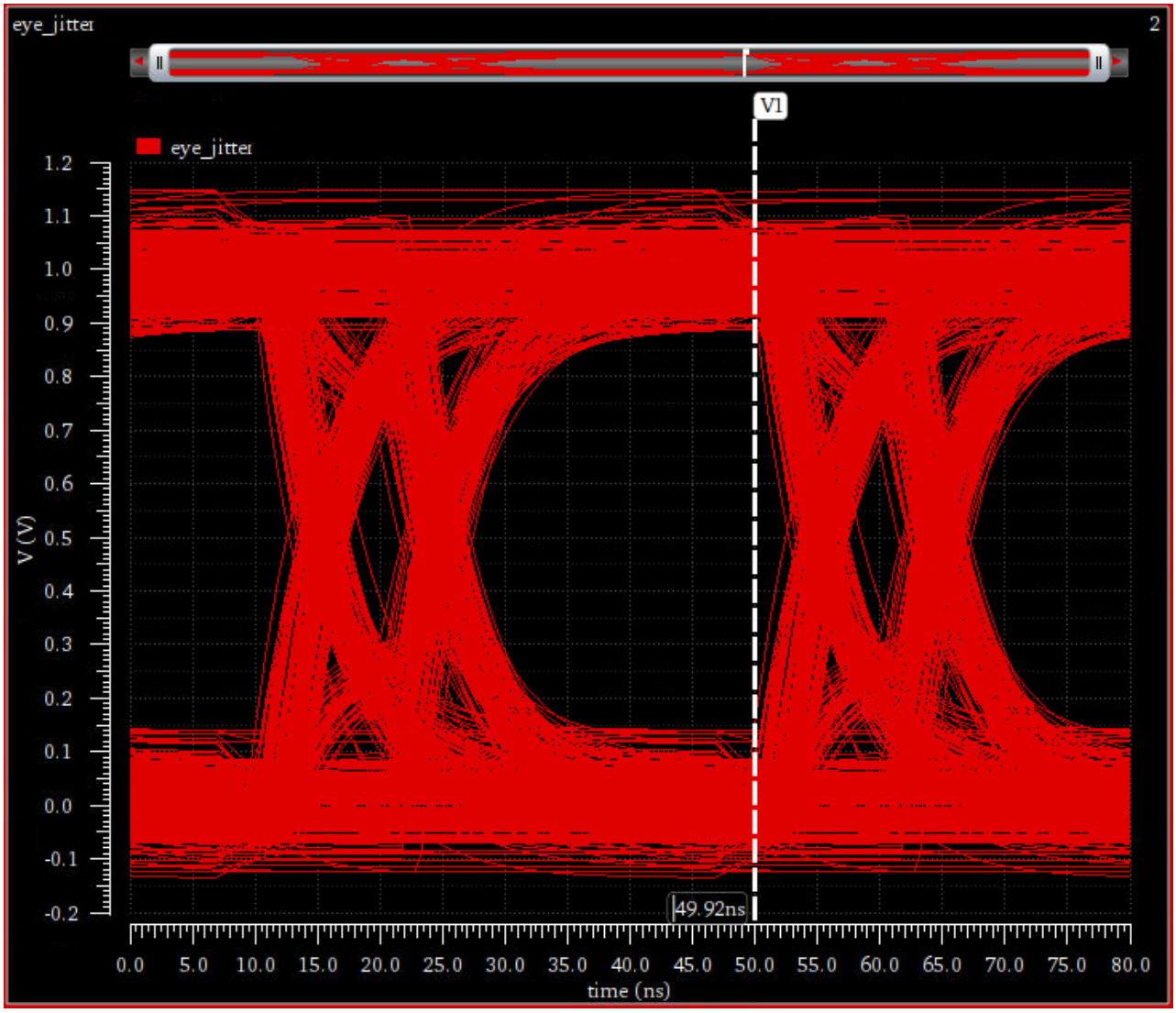


- `Max Eye Width and Max Eye Width Threshold`

  Max Eye Width finds the largest horizontal opening between the points of the maximum eye height. Max Eye Width Threshold is the Y-value where the maximum eye width

© 2023 Cadence Design Systems, Inc.

occurs. For example, the Maximum Eye Height at `747.8mV` is shown as `49.92ns` in the eye diagram below.



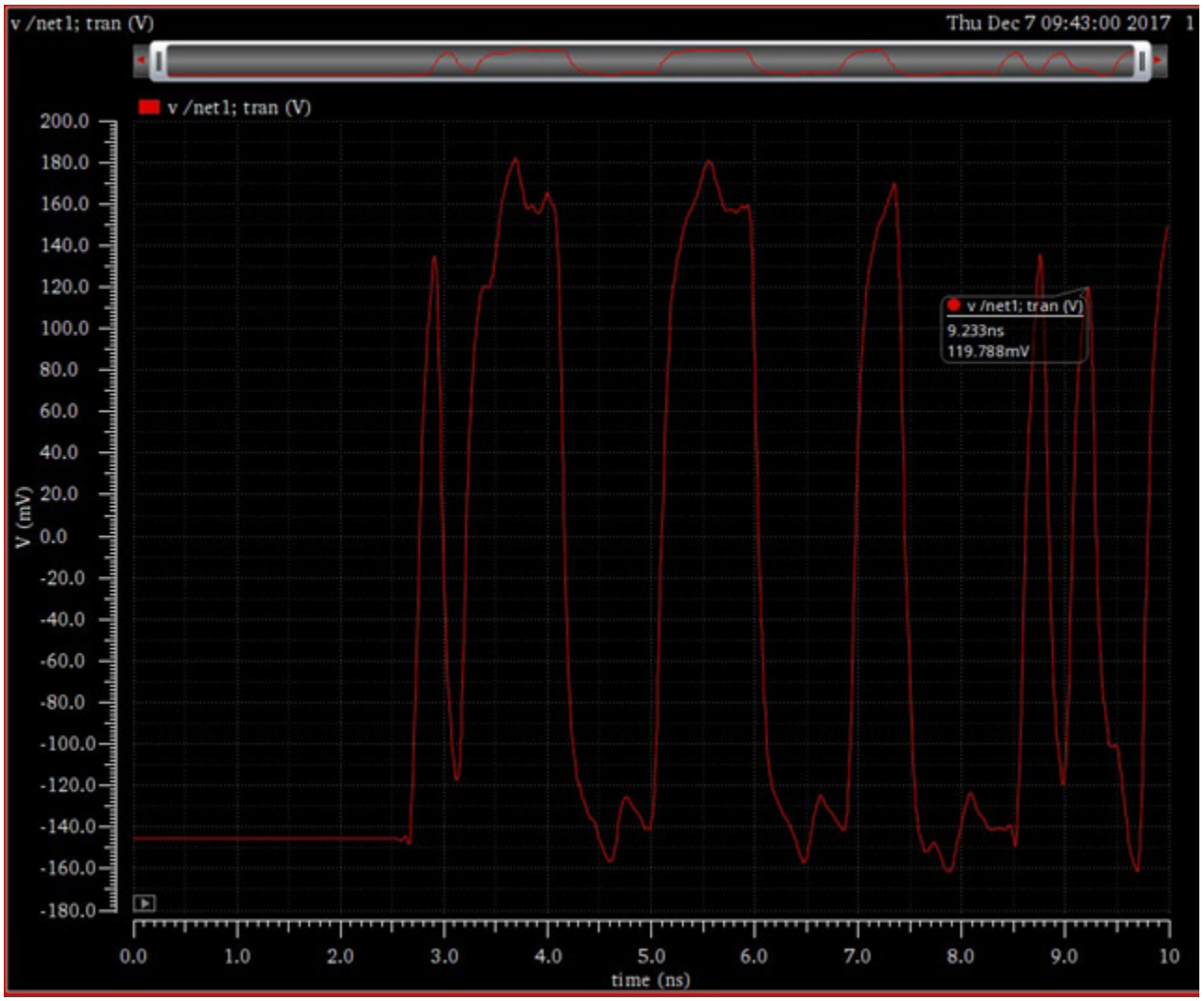
The following measurements are displayed in the *Outputs* section:

Max Eye Height Time	49.92n (s)
Max Eye Height	747.8m (V)
Max Eye Width Threshold	491.6m (V)
Max Eye Width	25.09n (s)

© 2023 Cadence Design Systems, Inc.

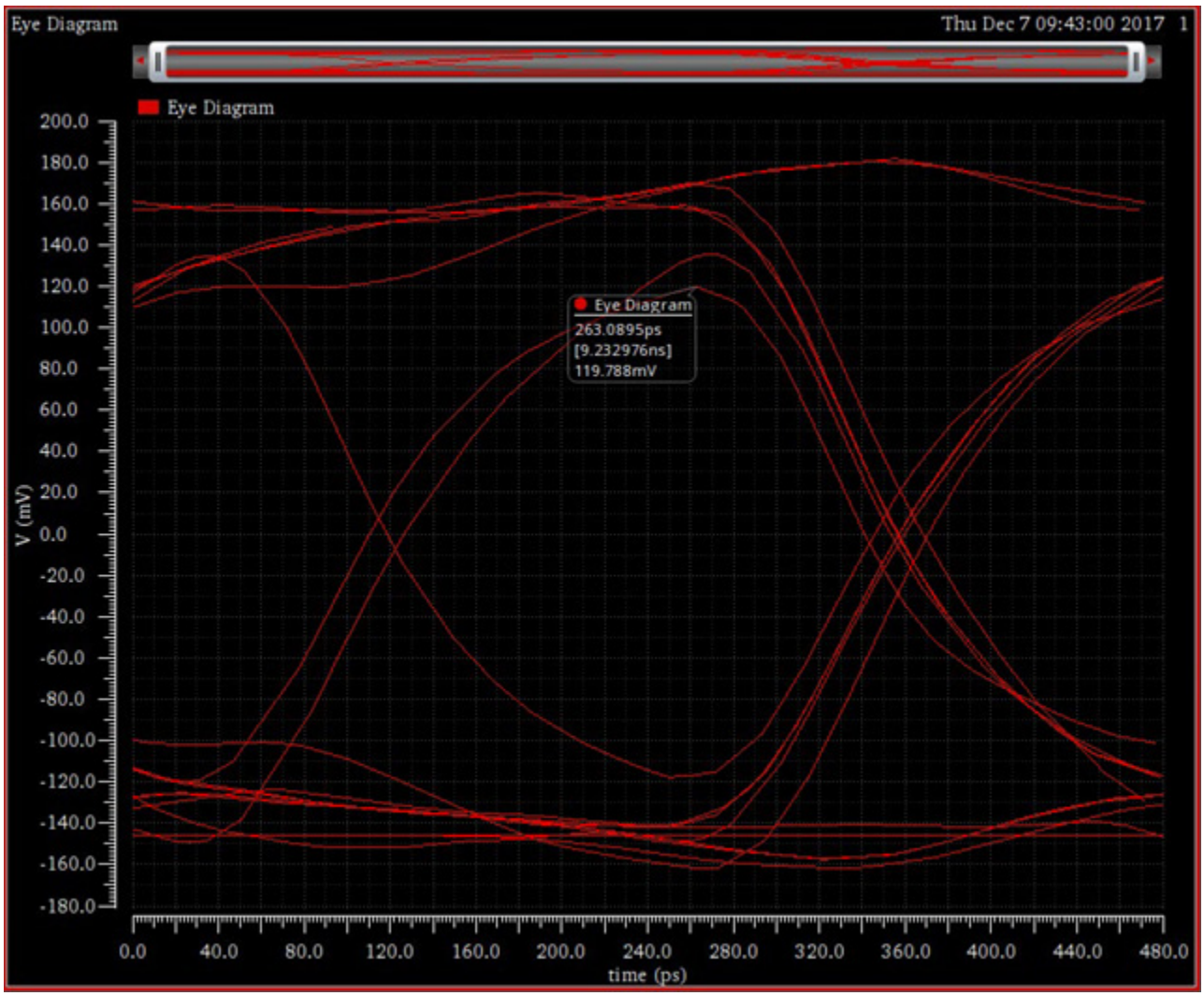
Displaying the Original Time in Marker Labels

The time value for the original waveform before it gets folded into an eye diagram is also displayed in the marker label on the eye diagram. For example, consider the waveform for the `v/net1` signal shown in the figure below. The label displays the time value on the x-axis as `9.233ns` when the voltage on Y-axis is set to `119.788mV`.



© 2023 Cadence Design Systems, Inc.

Now, if you create the eye diagram for this signal, the label at the same voltage (`119.788mv`) displays `263.0895ps[9.232976ns]119.788mv`, where `9.2329ns` is the original time before the eye diagram is generated.

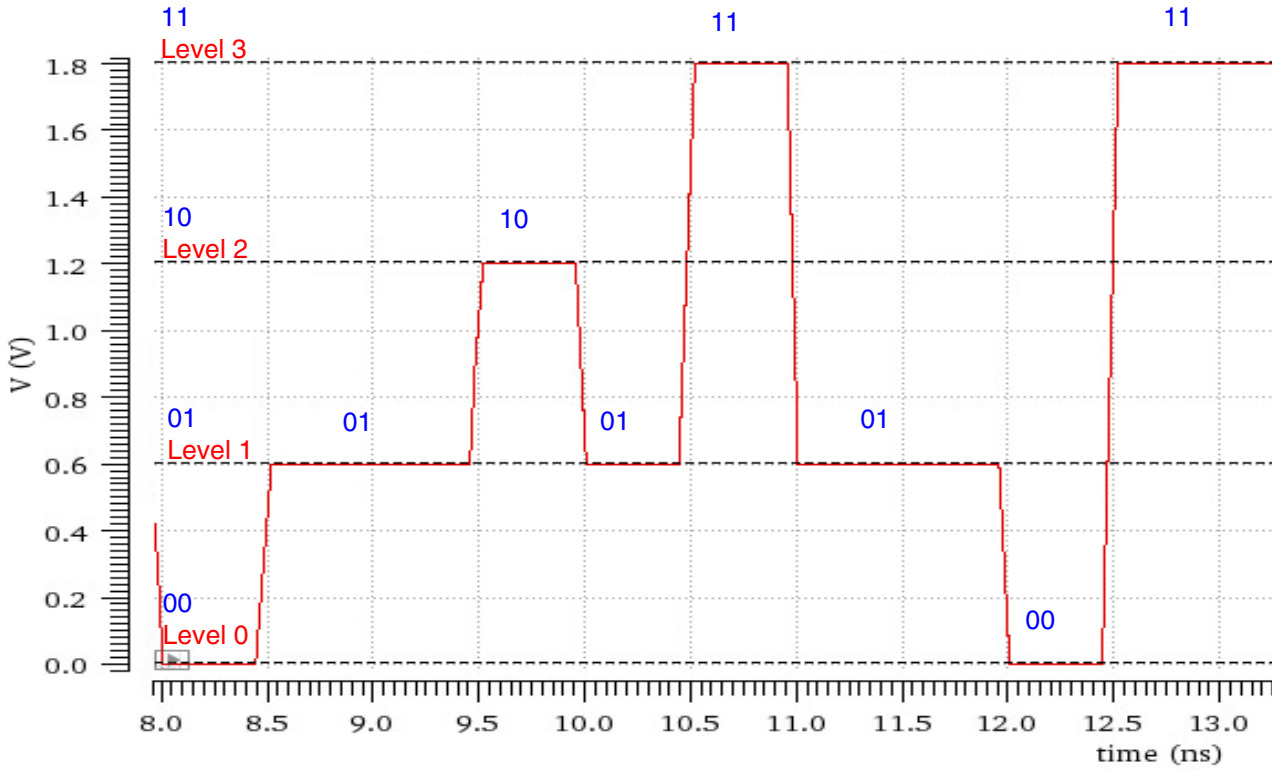


© 2023 Cadence Design Systems, Inc.

PAM4 Eye Measurements

The Pulse-Amplitude Modulation 4-Level (PAM4) is a modulation scheme that uses four voltage levels to represent four combinations of two-bit logic: 00, 01, 10, 11.

For each voltage level, you can measure the mean and standard deviations.

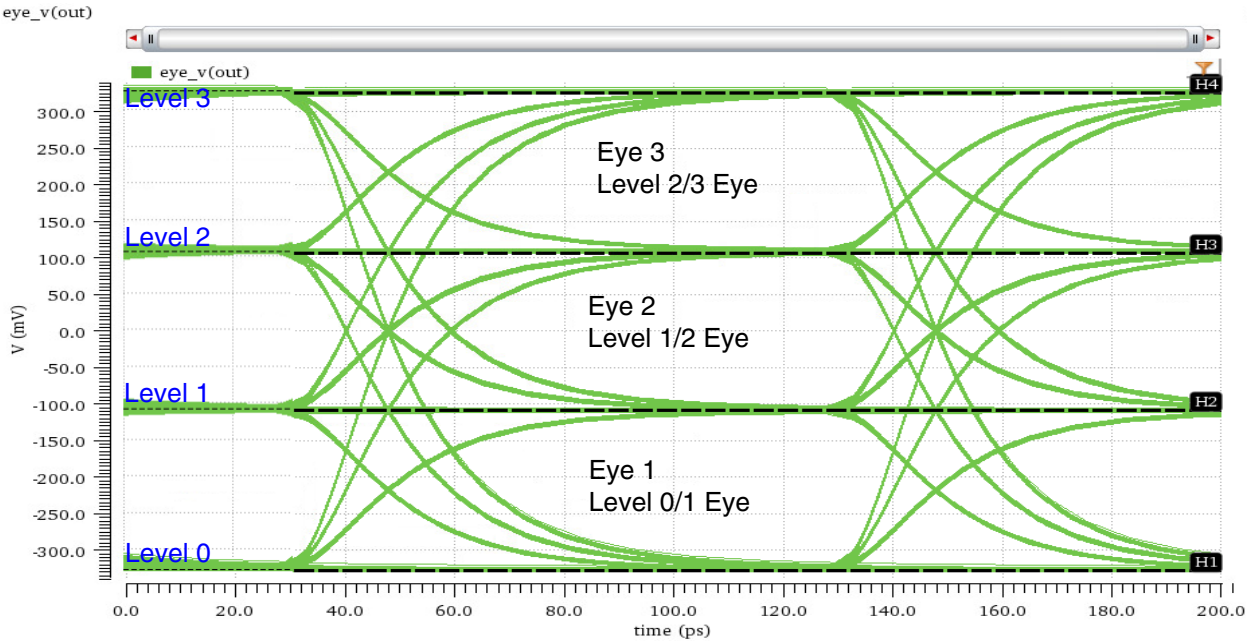


These four voltage levels result into three eyes and 12 transitions:

- Eye 1 (Level 0/1 Eye): Eye formed between level 0 and level 1
- Eye 2 (Level 1/2 Eye): Eye formed between level 1 and level 2

© 2023 Cadence Design Systems, Inc.

- Eye 3 (Level 2/3 Eye): Eye formed between level 2 and level 3



For each transition, you can calculate the rise/fall time:

- **0-1 Transition**: Time taken to rise from 20% to 80% of the difference between level 0 and level 1
- **0-2 Transition**: Time taken to rise from 20% to 80% of the difference between level 0 and level 2
- **0-3 Transition**: Time taken to rise from 20% to 80% of the difference between level 0 and level 3
- **1-0 Transition**: Time taken to fall from 80% to 20% of the difference between level 1 and level 0
- **1-2 Transition**: Time taken to rise from 20% to 80% of the difference between level 1 and level 2
- **1-3 Transition**: Time taken to rise from 20% to 80% of the difference between level 1 and level 3
- **2-0 Transition**: Time taken to fall from 80% to 20% of the difference between level 2 and level 0
- **2-1 Transition**: Time taken to fall from 80% to 20% of the difference between level 2 and level 1

© 2023 Cadence Design Systems, Inc.

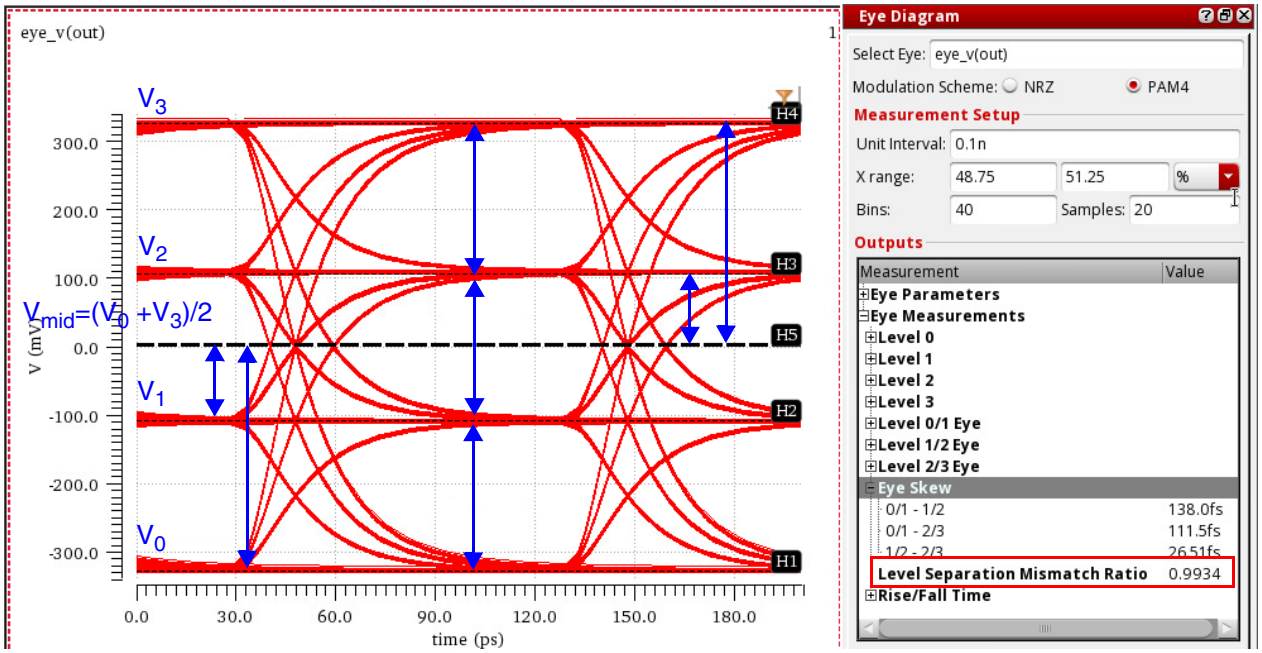
- **2-3 Transition**: Time taken to rise from 20% to 80% of the difference between level 2 and level 3
- **3-0 Transition**: Time taken to fall from 80% to 20% of the difference between level 3 and level 0
- **3-1 Transition**: Time taken to fall from 80% to 20% of the difference between level 3 and level 1
- **3-2 Transition**: Time taken to fall from 80% to 20% of the difference between level 3 and level 2

Eye Linearity

Eye linearity depends on the following two factors:

- **Eye Compression**: Eye compression is also referred to as *Level Separation Mismatch Ratio* (RLM or R_{LM}). R_{LM} combines the spacing of the four levels into a ratio that ranges from `0-1`. The R_{LM} is used to compare the eye heights of the three eyes formed between four voltage levels. If the eye heights of the three eyes are equal or nearly equal, the value of R_{LM} is `1` or nearly equal to `1`.

  In the following figure, all the three eyes are nearly equal in heights. Therefore, the R_{LM} is `0.9934`, which is nearly equal to `1`.



© 2023 Cadence Design Systems, Inc.

The four voltage levels are: V_0, V_1, V_2, and V_3. The mid-range level is V_{mid}. R_{LM} can be calculated as follows:

$$V_{mid} = \frac{V_0 + V_3}{2}$$

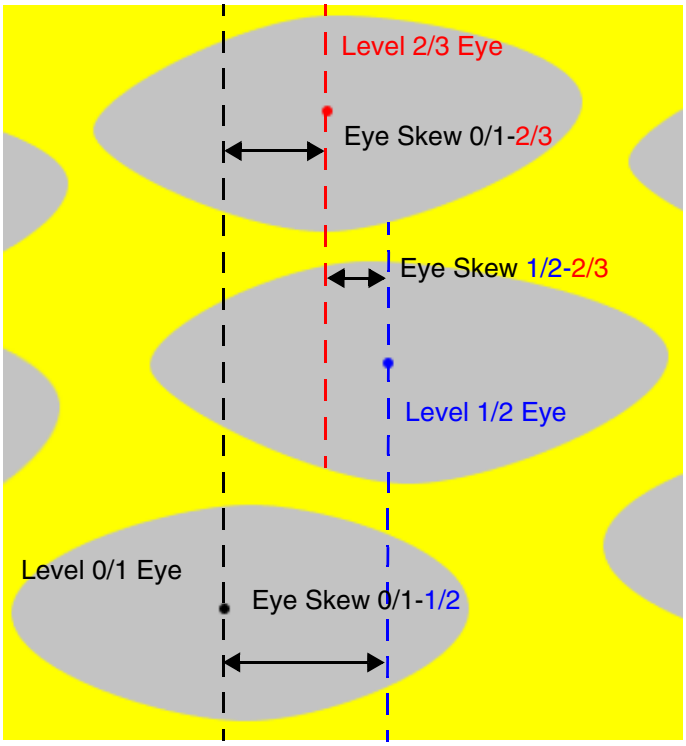
$$ES1 = \frac{V_1 - V_{mid}}{V_0 - V_{mid}}$$

$$ES2 = \frac{V_2 - V_{mid}}{V_3 - V_{mid}}$$

$$R_{LM} = min((3 \times ES1), (3 \times ES2), (2 - 3 \times ES1), (2 - 3 \times ES2))$$

© 2023 Cadence Design Systems, Inc.

- **Eye Skew**: Eye skew is the time difference between the centers of the eyes.
  - **0/1-1/2**: The time difference between the centers of the level 0/1 and level 1/2 eyes.
  - **01/-2/3**: The time difference between the centers of the level 0/1 and level 2/3 eyes.
  - **1/2-2/3**: The time difference between the centers of the level 1/2 and level 2/3 eyes.



NRZ vs PAM3 vs PAM4 Eye Measurements

The following table describes the difference between NRZ and PAM4 eye measurements.

	NRZ	PAM3	PAM4
Voltage levels	2 (0 and 1)	3 (00, 01, 10)	4 (00, 01, 10, 11)

© 2023 Cadence Design Systems, Inc.

	NRZ	PAM3	PAM4
Number of Eyes	A single eye is formed between level 0 and level 1.	Two eyes are formed between the following levels: ■ Level 0 and level 1 ■ Level 1 and level 2	Three eyes are formed between the following levels: ■ Level 0 and level 1 ■ Level 1 and level 2 ■ Level 2 and level 3
Number of Transitions	Two transitions: ■ 0-1 Transition ■ 1-0 Transition	Six transition: ■ 0-1 Transition ■ 0-2 Transition ■ 1-0 Transition ■ 1-2 Transition ■ 2-0 Transition ■ 2-1 Transition	12 transitions ■ 0-1 Transition ■ 0-2 Transition ■ 0-3 Transition ■ 1-0 Transition ■ 1-2 Transition ■ 1-3 Transition ■ 2-0 Transition ■ 2-1 Transition ■ 2-3 Transition ■ 3-0 Transition ■ 3-1 Transition ■ 3-2 Transition

© 2023 Cadence Design Systems, Inc.

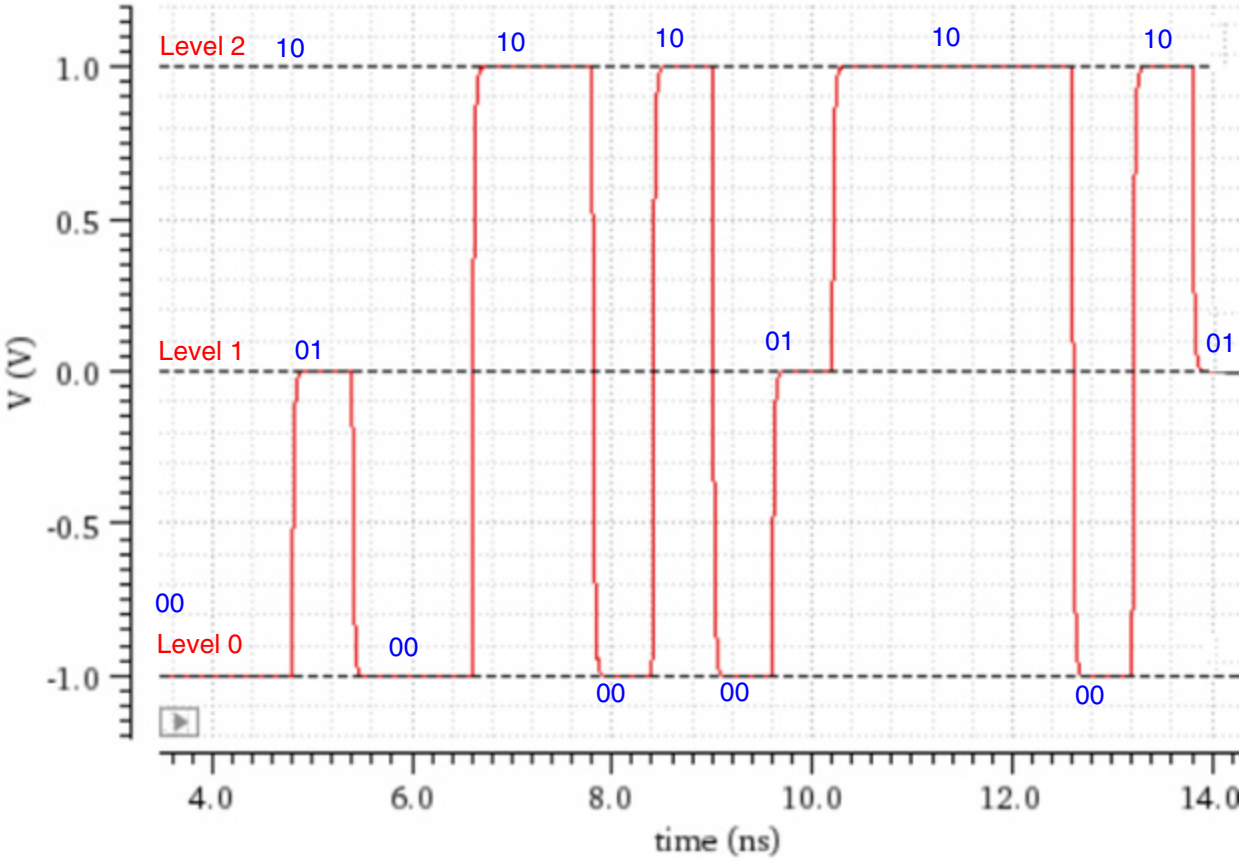
PAM3 Eye Diagram

Pulse-Amplitude Modulation 3-Level (PAM3) is a digital signaling technique (modulation scheme) that uses three voltage levels to represent three combinations of two-bit logic: 00, 01 and 10.

It offers a good balance of power efficiency, robustness, and complexity. It is well-suited for a variety of applications, such as high-speed data communication systems and optical networks.

It is more power efficient and robust to noise and distortion than non-return-to-zero (NRZ), but it is more susceptible to inter-symbol interference than Pulse-Amplitude Modulation 4-Level (PAM4).

The following figure shows three different voltages levels that PAM3 uses to represent data.

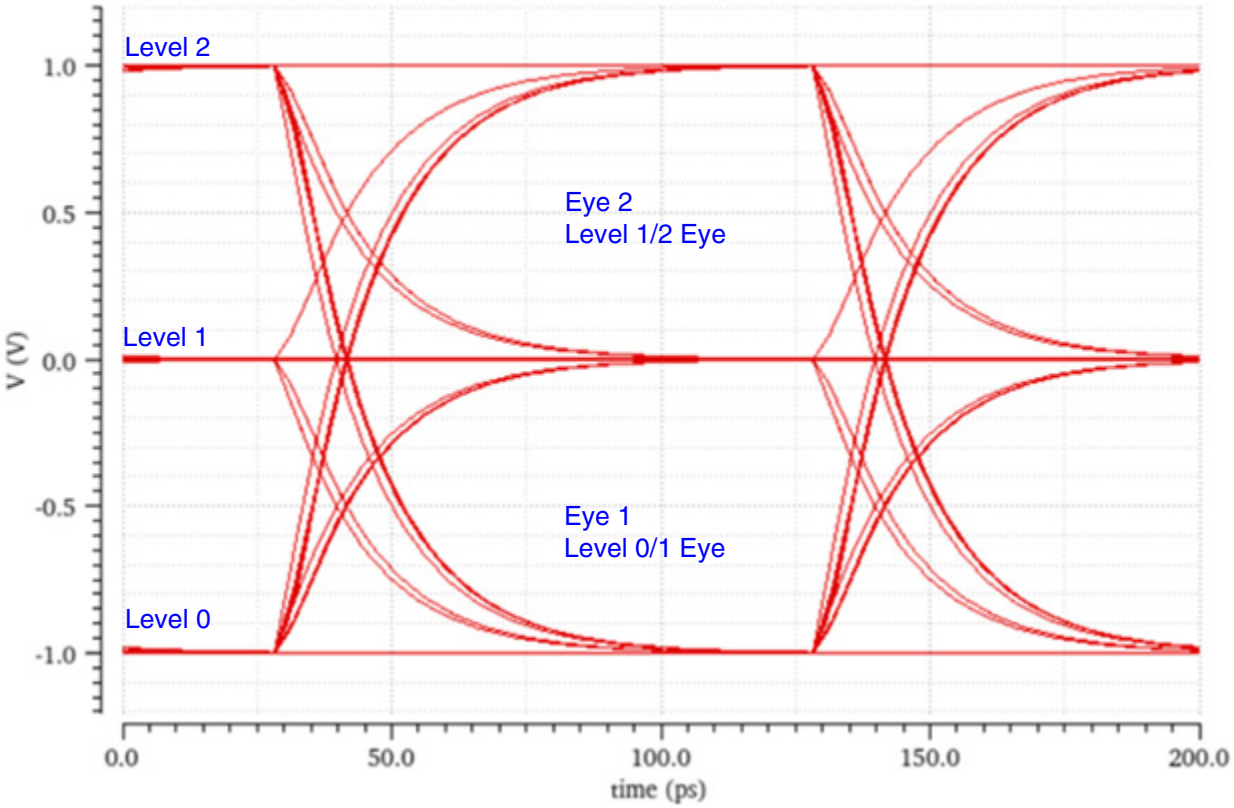


These three voltage levels result into two eyes and 6 transitions:

- Eye 1 (Level 0/1 Eye): Eye formed between level 0 and level 1

© 2023 Cadence Design Systems, Inc.

- Eye 2 (Level 1/2 Eye): Eye formed between level 1 and level 2



Related Topics

Eye Measurements for Different Levels

Eye Parameters and Eye Measurements

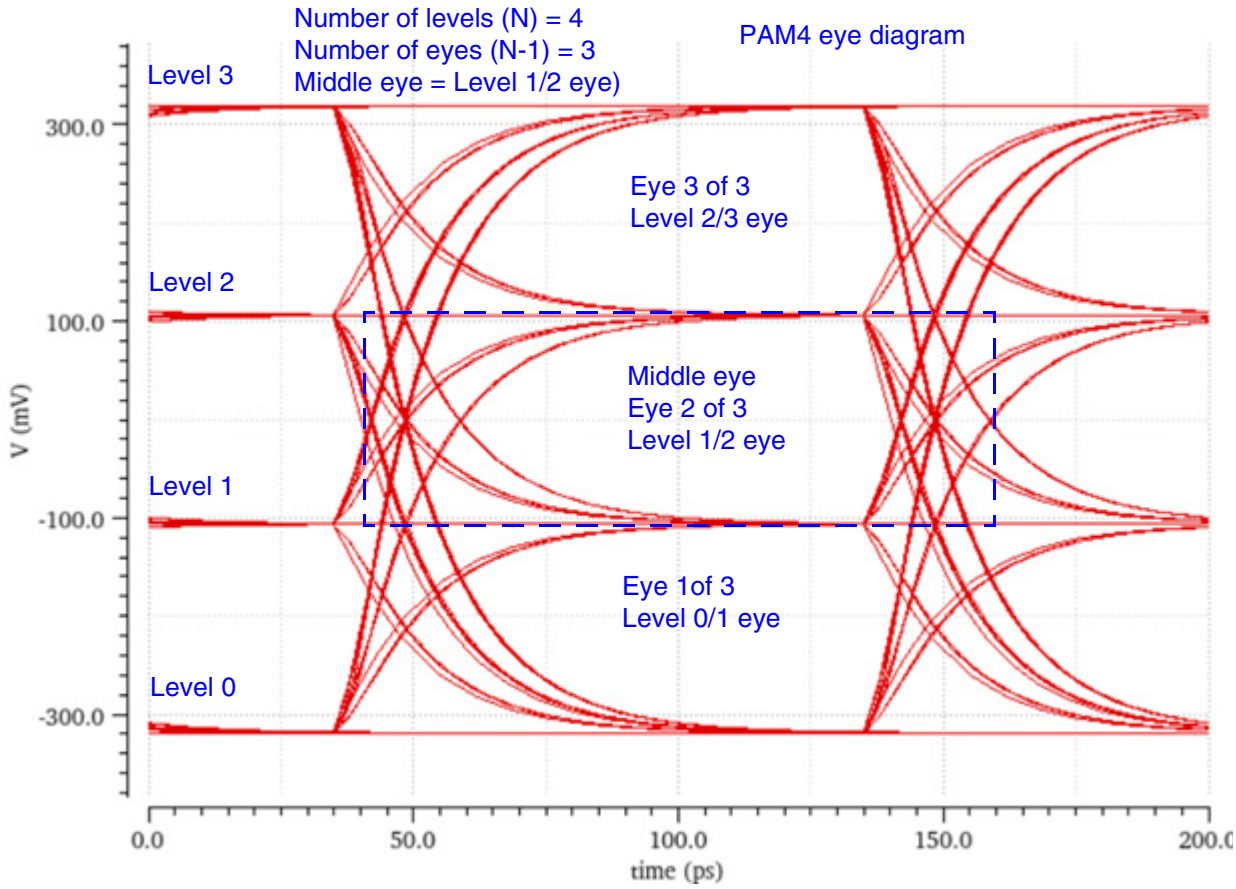
© 2023 Cadence Design Systems, Inc.

Eye Measurements for Different Levels

Eye measurements, such as eye height, eye width, and eye amplitude are based on T_{mid}, which is the midpoint of the maximum eye width of the middle eye.

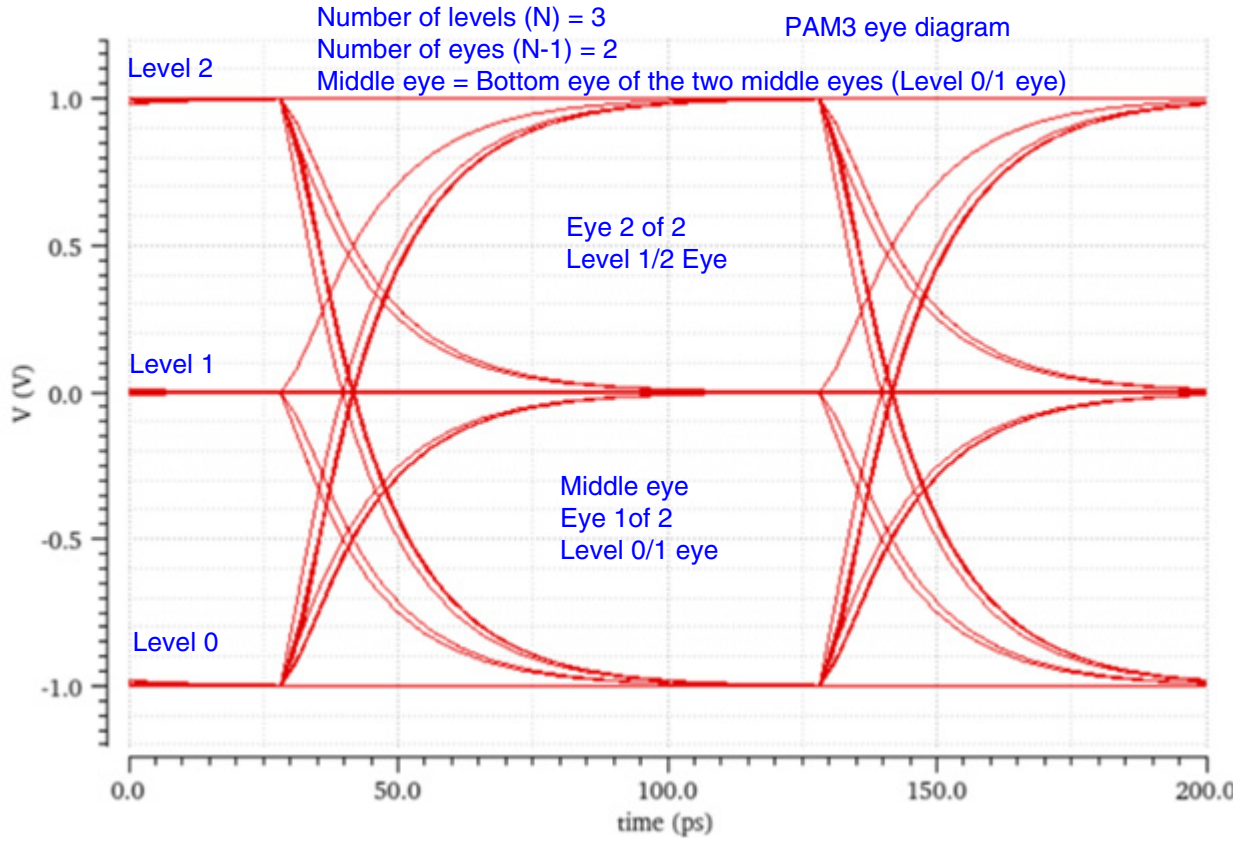
The selection of the middle eye depends on the number of levels (N) in the eye diagram. The number of eyes formed is one less than N. For example, the number of levels (N) in PAM4 is 4, and the number of eyes is 3 (4-1). The number of levels (N) in PAM3 is 3, and the number of eyes formed is 2 (3-1).

- If N is an even number, the number of eyes formed (N-1) is odd. Therefore, the second eye is the middle eye.



© 2023 Cadence Design Systems, Inc.

- If `N` is an odd number, the number of eyes formed, (`N-1`), is even. Therefore, the bottom-most of the two middle eyes is considered as the middle eye.

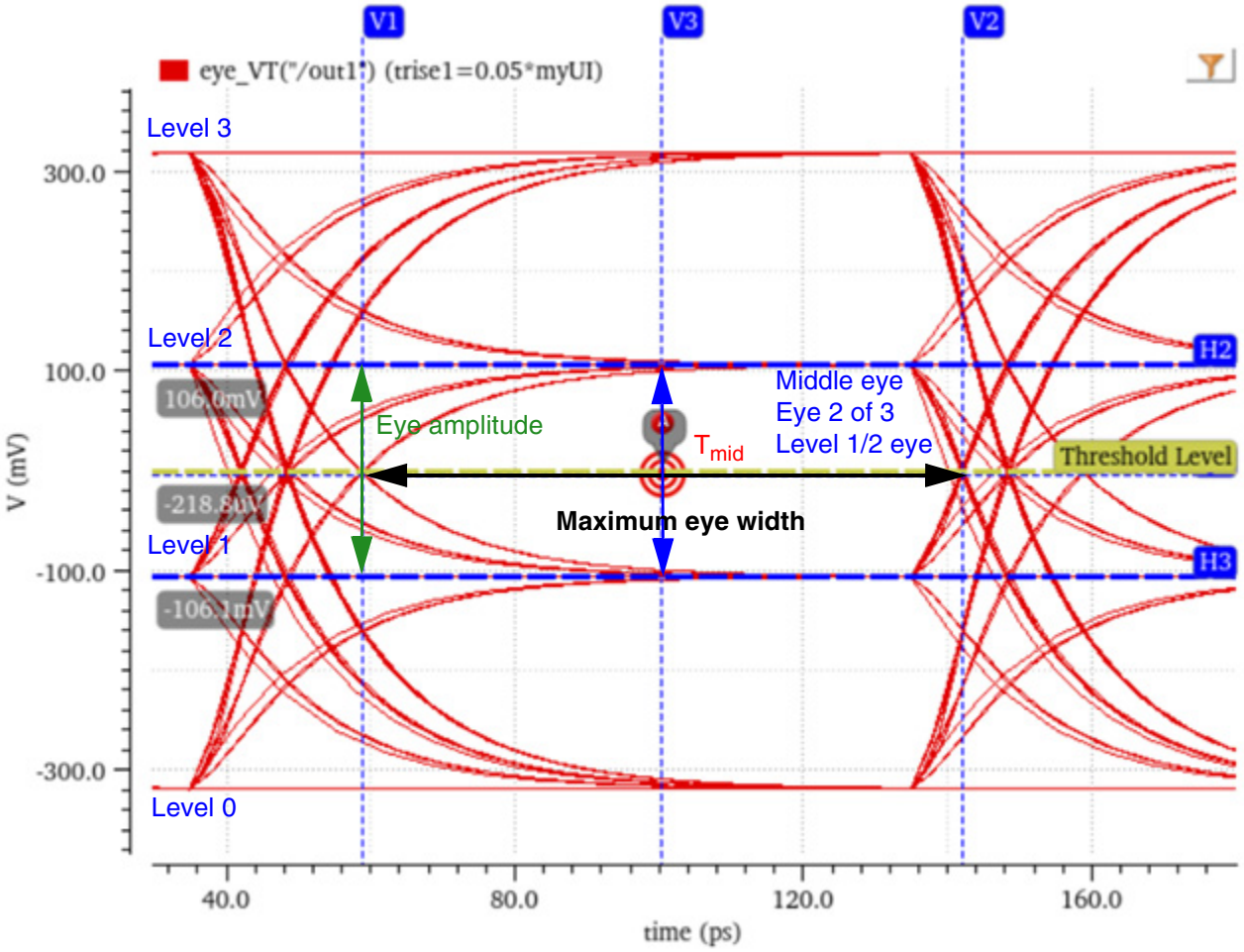


After identifying the middle eye, its maximum eye width is calculated, and the following calculations are performed:

- The midpoint of the maximum eye width is T_{mid}.
- The eye height at T_{mid} is the T_{mid} height.
- The midpoint of the T_{mid} height is the threshold.

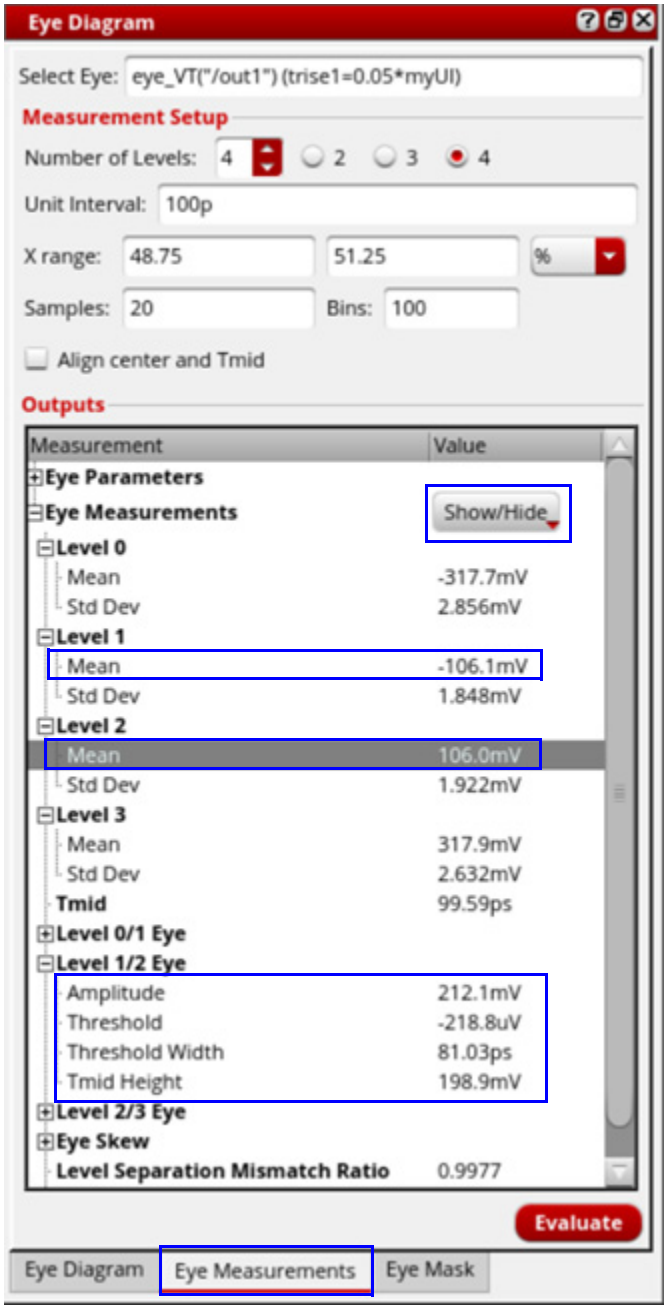
© 2023 Cadence Design Systems, Inc.

- Eye amplitude is the distance between the mean of the two levels that contain the eye.



© 2023 Cadence Design Systems, Inc.

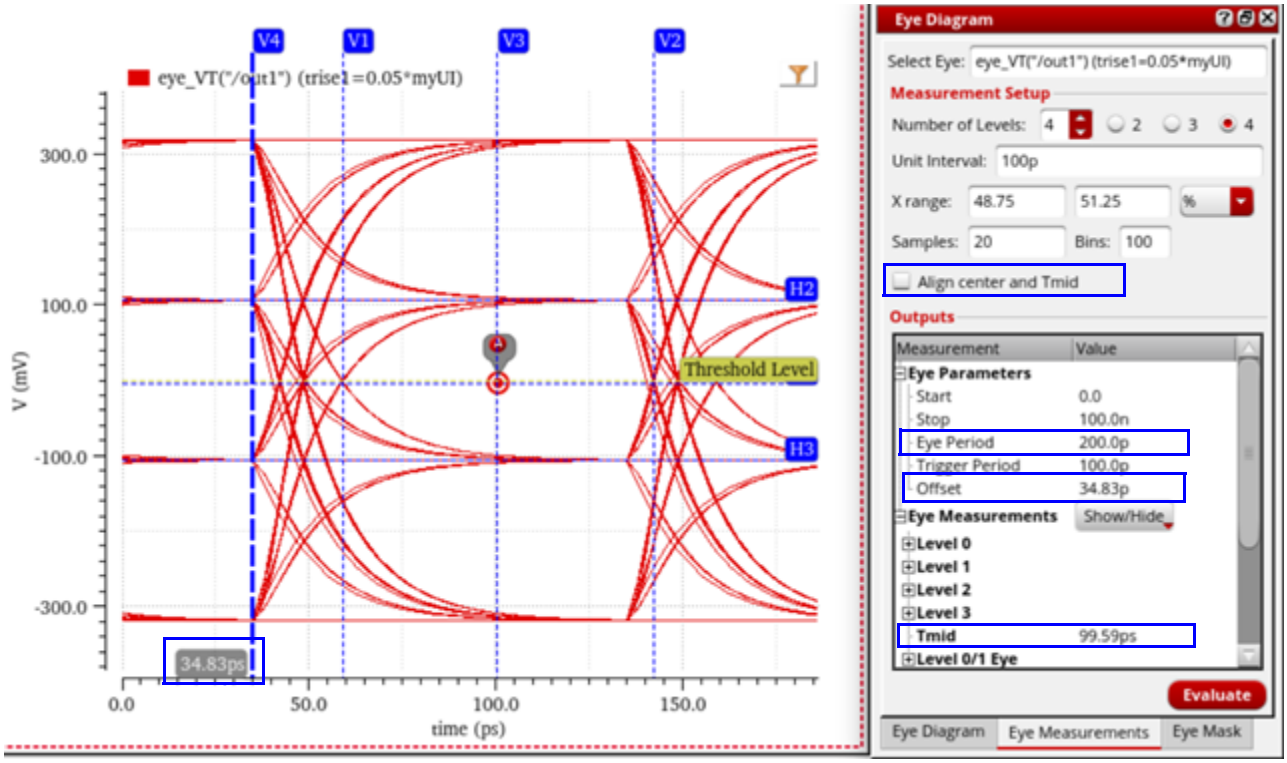
These calculations can be verified using the *Eye Measurements* tab of the Eye Diagram assistant.



In addition to the default eye measurements, you can display the following eye measurements by clicking the *Show/Hide* drop-down arrow and then selecting the corresponding check marks: *Max Width Threshold*, *Max Width*, *Max Height Time*, and *Max Height*.

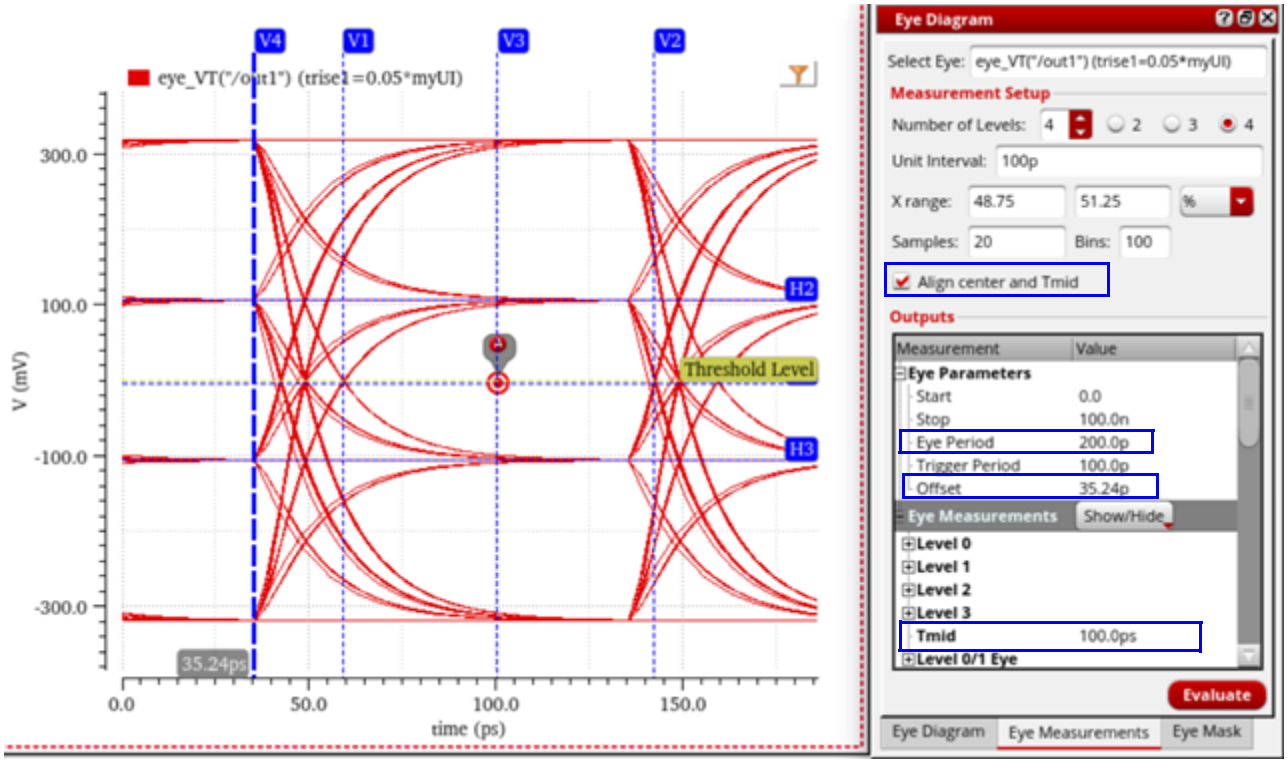
© 2023 Cadence Design Systems, Inc.

In the following eye diagram, observe that the offset of the eye is at `34.83ps`, T_{mid} is at `99.59ps`, and the center of the eye (`eye period/2`) is at `100ps` (`200/2`), which indicates that T_{mid} and the center of the eye are not aligned together.



© 2023 Cadence Design Systems, Inc.

You can align T_{mid} with the center of the eye by selecting the *Align center and Tmid* check box and then clicking *Evaluate*. This adjusts the offset of the eye so that T_{mid} becomes the midpoint of the eye period.



You can also edit the offset point of the eye diagram using the *Offset* field in the Eye Trace Properties form. To open the Eye Trace Properties form, right-click an eye diagram trace and choose *Trace Properties*.

Related Topics

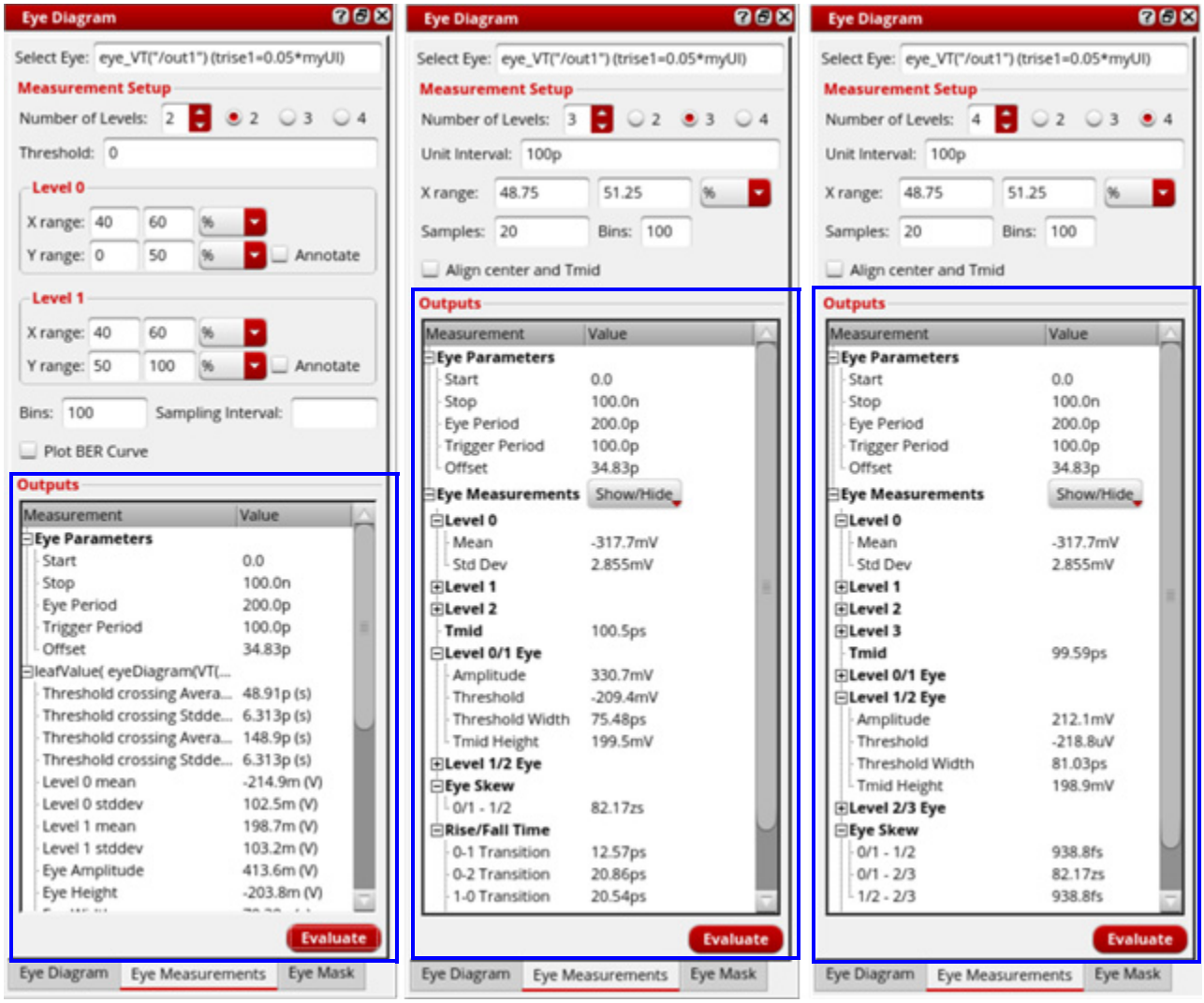
Eye Parameters and Eye Measurements

© 2023 Cadence Design Systems, Inc.

Eye Parameters and Eye Measurements

Eye parameters and eye measurements are displayed in the *Outputs* section of the *Eye Measurements* tab of the Eye Diagram assistant.

NRZ PAM3 PAM4



Depending on the modulation scheme, an eye diagram can have different voltage levels. For example, PAM4 has four voltage levels.

For each voltage level, you can calculate the mean and standard deviation:

- **Level 0 Mean**: Mean of the voltage values within level 0
- **Level 0 Std Dev**: Standard deviation of voltage values within level 0

© 2023 Cadence Design Systems, Inc.

- **Level 1 Mean**: Mean of the voltage values within level 1
- **Level 1 Std Dev**: Standard deviation of voltage values within level 1
- **Level 2 Mean**: Mean of the voltage values within level 2
- **Level 2 Std Dev**: Standard deviation of voltage values within level 2

`Tmid` is the midpoint of the maximum eye width of the middle eye. To know more about how the middle eye is selected, see Eye Measurements for Different Levels.

For each eye, you can calculate the following eye measurements:

- **Amplitude**: Eye amplitude is the distance between the mean points of the two levels that contain the eye.
  - **Amplitude** $_{\text{(Level 0/1 Eye)}}$ = Distance between the mean points of the level 0 and level 1
  - **Amplitude** $_{\text{(Level 1/2 Eye)}}$ = Distance between the mean points of the level 1 and level 2
  - **Amplitude** $_{\text{(Level 2/3 Eye)}}$ = Distance between the mean points of the level 2 and level 3
- **Threshold**: The midpoint of the `Tmid` height is the threshold, where `Tmid` height is the eye height at `Tmid`
- **Threshold Width**: Eye width at the threshold
- **T_{mid} Height**: `Tmid` height is the eye height at `Tmid`
- **Max Width Threshold**: Voltage level at which the eye width is maximum
- **Max Width**: Maximum width of the eye
- **Max Height Time**: Time (X-axis value) at which eye attains the maximum height
- **Max Height**: Maximum height of the eye

Related Topics

Eye Parameters and Eye Measurements

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

D

Constants

This chapter lists constants and their definitions for the SKILL mode.

Table D-1 Constants in the SKILL Mode

Constant	Definition
Boltzmann	1.380622e-23
charge	1.6021917e-19
degPerRad	57.2957795130823
epp0	8.854e-12
pi	3.14159265358979323846
sqrt2	1.41421356237309504880
twoPi	6.28318530717958647688

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

E

Defining New SKILL Functions

You can define a function and add it to the *SKILL User Defined Functions* category in the calculator by following these steps:

1. Define the form that prompts for user-defined arguments to the function.
2. Define the syntax of the function in the callback procedure.
3. Register the function.

Defining a Form

The following example shows how to define an input form for a function that takes three arguments. The first argument is the buffer expression. The other two arguments are the boundaries of the range of the expression on which you want to operate.

```
procedure( CreateMyForm()
   let( ( fieldList a b )
    a = ahiCreateStringField(
        ?name 'from
        ?prompt "From"
        ?value ""
    )
    b = ahiCreateStringField(
        ?name 'to
        ?prompt "To"
        ?value ""
    )
    fieldList = list(
        list( a 5:0 120:25 40 )
        list( b 160:0 110:25 30 )
    )
    calCreateSpecialFunctionsForm( 'MyForm
        fieldList )))
```

In this example, the From and To fields are string fields created in a two-dimensional form specification for fieldList. The form is created by the call to `calCreateSpecialFunctionsForm`. This function creates and registers the form with the specified form symbol, MyForm.

© 2023 Cadence Design Systems, Inc.

Defining a Callback Procedure

You define a callback procedure that is called from the entry on the Calculator User Defined Functions category. Since this example uses a form to prompt for additional information required by the special function, the callback procedure is

```
procedure( MySpecialFunctionCB()
  calCreateSpecialFunction(
        ?formSym 'MyForm
        ?formInitProc 'CreateMyForm
        ?formTitle "Test"
        ?formCallback "calSpecialFunctionInput( 'test
        '(from to) )"
  )
)
```

In this procedure, a call is made to `calCreateSpecialFunction`, which creates and displays the form and then builds the expression in the buffer with the specified form fields.

Using Stack Registers in the Procedure

You can use the special symbol `'STACK` in the list of form fields to get expressions from the stack.

For example, if you want to insert a stack element between the From and To arguments in the special function expression, you can specify the callback line as follows:

```
?formCallback "calSpecialFunctionInput('test '(from STACK to))"
```

If your special function does not require a form to prompt for , you can define your callback as follows:

```
procedure( MySpecialFunctionCB()
    calSpecialFunctionInput( 'test nil )
)
```

Registering the Function

You register the function and callback with the `calRegisterSpecialFunction`:

```
calRegisterSpecialFunction(
    list( "test" 'MySpecialFunctionCB )
)
```

The next time you open the calculator, the functions you defined appear in the User Defined Functions category.

© 2023 Cadence Design Systems, Inc.

Defining a Custom Function

Custom functions need to be supported for both single and multi-dimensional waveform (parametric) data.

A custom function example is shown below:

```
procedure(myFunction(wf)
  cond(
    (drIsWaveform(wf)
      ;; your custom function code
    )
    (famIsFamily(wf)
      famMap('myFunction wf)
    )
    (t
      error("myFunction cannot handle argument %L\n" wf)
    )
)
```


© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

F

Working With Function Templates

This chapter describes the function templates and how you can use them to create new functions in Virtuoso Visualization and Analysis XL Calculator with supporting examples.

- Function Templates
  - Function Template Search Paths
  - Template Catalog Summary File
- Creating a template file
- Working with Template File
- Examples
  - Example 1: Sample template with single argument: average.ocn
  - Example 2: Sample template with multiple arguments: delay.ocn
  - Example 3: Signature described by a format statement: compression.ocn
  - Example 4: Creating your own template
- Advanced Features to provide GUI Hints

© 2023 Cadence Design Systems, Inc.

Function Templates

Function templates are a mechanism to facilitate easier construction and addition of new functions in the Calculator. They are described in a prescribed format in function template file.

Virtuoso Visualization and Analysis XL uses these template files to perform the following tasks:

- Builds lists of function names, separated into categories and displays it in the GUI panel.
- Generates an expression using rules present in the template file.

Function Template Search Paths

Virtuoso Visualization and Analysis XL searches for function templates using csfsearchpath. The priority of searching a UDF GUI template will be:

1. `<CSF_SEARCH_PATHS>/measures`
2. `<CDS_INST_DIR>/tools/dfII/etc/tools/viva/functionTemplates`

Virtuoso Visualization and Analysis XL function templates, shipped by Cadence, are stored at the following location:

`<CDS_INST_DIR>/tools/dfII/etc/tools/viva/functionTemplates`

Template Catalog Summary File

The template catalog summary file `.skeMeaseruesCatalog` lists the categories and the templates that exists in each category. Virtuoso Visualization and Analysis XL template catalog file is shipped in the following directory.

`<CDS_ROOT>/tools/dfII/etc/tools/viva/functionTemplates/.skeMeasuresCatalog`

Different categories are specified in-between the lines `skeBeginCategory` and `skeEndCategory`. The below example defines Math, Modifier, RF Functions, Special Function, and Trigonometric as the categories.

```
skeBeginCategory
Math
Modifier
RF Functions
Special Functions
Trigonometric
```


© 2023 Cadence Design Systems, Inc.

```
skeEndCategory
```

You can specify different functions within a category in between the lines:

```
skeBeginMeasures <Category Name> & skeEndCategory.
```

For example, function “Fourier Evaluation” of category “Special Functions can be defined as follows:

```
skeBeginMeasures.Special Functions
fourEval;"Fourier Evaluation"
skeEndMeasures
```


© 2023 Cadence Design Systems, Inc.

Creating a template file

To add a new function, write a template file for your function and override `.skeMeasuresCatalog` file.

To create a template file, perform the following steps:

1. Choose one of the two directories listed below where you want to store the new functions:
   - `<CSF_SEARCH_PATHS>/measures`
   - `<CDS_INST_DIR>/tools/dfII/local/tools/wavescan/measures`
2. After creating a directory named "`measures`" which is accessible to the complete project team, copy `<CDS_ROOT>/tools/dfII/etc/tools/wavescan/measures/.skeMeasuresCatalog` to `CSF_SEARCH_PATHS/measures/.skeMeasuresCatalog`.
3. Open `CSF_SEARCH_PATHS/measures/.skeMeasuresCatalog` for editing.
4. Add an entry of the new function in between `skeBeginMeasures &` `skeEndMeasures` tags of the category where the new function is required to belong.

   For example to add a function named 'testfun' in math category, add testfun; in between `skeBeginMeasures Math` & nearest `skeEndMeasures` line.

   Note: Current implementation doesn't merge the entries specified in separate `.skeMeasuresCatalog` files.

© 2023 Cadence Design Systems, Inc.

Working with Template File

Each Virtuoso Visualization and Analysis XL function template file describes a Calculator function. The template provides information such as:

- The function name
- The function's categories
- Input parameters and default values
- Gui building tips such as adjacent row hints and parameter dependencies
- Tool tip information (currently ignored).
- Rules to build the function expression

The template is divided into following sections:

Header

Header describes the general function information such as the name, display name, description, and category list.

- **function name:** This is the name used to construct the expression
- **name:** This is the display name used as the dialog label in the function panel.
- **Category list:** Each template can belong to multiple categories. For example, the riseTime template could belong to "Special Functions" and "transient". Currently we do not define categories according to analysis type. The current categories are:
  - Math
  - Modifier
  - RF Functions
  - Special Functions
  - Trigonometric

The Analysis Section

The analysis section describes how to generate the expression, the signal, and parameter arguments.

© 2023 Cadence Design Systems, Inc.

- **args:** Describe how to build the expression
- **signals:** Describe each signal argument:
  - prompt
  - tool tip
  - params. describe each parameter
  - prompt
  - tool tip
  - type
  - default value
  - required

© 2023 Cadence Design Systems, Inc.

Examples

Example 1: Sample template with single argument: average.ocn

Location:

```
<CDS_INST_DIR>/tools/dfII/etc/tools/wavescan/measures/average.ocn
```

```
 1;;; GUI builder information ;;;;;;;;;;;;;;;;;;;;;;;;;;
 2 ocnmRegGUIBuilder(
 3   '(nil
 4     function average
 5     name average
 6     description "Compute average of a waveform over its entire range."
 7     category ("Special Functions")
 8     analysis (nil
 9                general (nil
10                     args (signal)
11                     signals (nil
12                              signal (nil
13                                      prompt "Signal"
14                                      tooltip "signal to average"
15                                      )
16                     )
17                     inputrange t
18                     )
19     )
20     outputs (result)
21    )
22 )
```

~

- The function name used to build the expression is defined by line 4 (function average).
- Line 5 is the name used in the dialog label if a function panel is displayed. In this case the function signature takes a single argument (signal) and no function panel is required.

© 2023 Cadence Design Systems, Inc.

- Line 6 is a simple description that will be used at some point for bubble help. (This feature is not yet implemented).
- Line 7 is a set of category names. Typically each function belongs to a single category. Optionally a function can belong to multiple categories.
- Line 8 begins the analysis section.
- Line 10 is the args statement which provides the rule used by Virtuoso Visualization and Analysis XL to generate the function expression. The args statement is a list of ordered signal and parameter names and in our example contains a single item: signal. Therefore the signature looks something like: `average(VT("/net10"))`, it takes a single signal name

  The order in the args statement defines the order the parameter values will be named in the expression that is put into the buffer. This may or may not be the same order that is displayed in the function panel. Some functions require a more complex mechanism to describe the signature. The template mechanism provides a format statement that will be described in a later example.
- Line 11 starts the signal section. A function signature might contain multiple signal parameters (ex: the delay function).
- The single signal description begins at Line 12. Line 13 is the signal prompt used to name the signal parameter in the Function Panel. Line 14 is the signal tool tip. This is currently not used.

Example 2: Sample template with multiple arguments: delay.ocn

Location:

(<CDS_INST_DIR>/tools/dfII/etc/tools/wavescan/measures/delay.ocn)

```
 2 ;;;;;;;;;;;;;;;;;;;;;;;;; GUI builder information
;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 3 ocnmRegGUIBuilder(
 4   '(nil
 5     function delay
 6     name delay
 7     description "delay "
 8     category ("Special Functions")
 9     analysis (nil
10               tran  (nil
```


© 2023 Cadence Design Systems, Inc.

```
11                  args (signal1 threshold1 edge1 type1 signal2 threshold2
edge2 type2 numberOfOccurences)
12                   signals (nil
13                            signal1 (nil
14                                    prompt "Signal1"
15                                    tooltip "signal to measure"
16                                    )
17                            signal2 (nil
18                                    prompt "Signal2"
19                                    tooltip "signal to measure"
20                                    )
21                   )
22                   params (nil
23                           threshold1 (nil
24                                     prompt "Threshold Value 1"
25                                     tooltip "Threshold Value 1"
26                                     default 2.5
27                                     type float
28                                     min 0)
…
     periodicity1(nil
     prompt "Periodicity 1”
     tooltip "Periodicity 1"
     default 1
     type float )
 …
65                           periodicity2 (nil
66                                     prompt "Periodicity 2"
67                                     tooltip "Periodicity 2"
68                                     default 1
69                                     type float
70                            )
71                            numberOfOccurences (nil
```


© 2023 Cadence Design Systems, Inc.

```
 72                                      prompt "Number of occurrences"
 73                                      tooltip "Occurrence choice"
 74                                      default single
 75                                    type ((single (0 0 nil nil)) (multiple
(" %s  %s  t  %s " periodicity1 periodicity2 sweepName)))
 76                                )
 77                               sweepName (nil
 78                                      prompt "Plot/print vs."
79                                tooltip "Independent variable to plot against."
 80                                      required nil
 81                                      default trigger
 82                                      type (trigger target cycle)
 83                                      min 0)
 84                     )
```

The `delay` template is similar to `average`. Some differences:

- Line 11: The args description contains multiple parameters so a Function Panel will be built to describe this function.

  The signals section contains two signals named signal1 and signal2. The prompts (lines 14 and 18) will name the respective fields in the panel.

  This template contains parameters beginning at Line 22.

- Line 77: the parameter sweepName. Line 82 names the type: a simple list of cyclic choices. The default value (line 81) names the default choice to be initially displayed.

- Line 71: a much more complex example of a cyclic type parameter `(numberOfOccurances)`.

  The cyclic type (Line 75) is a list of two choices: single and multiple:

  ```
  type ((single (0 0 nil nil)) (multiple (" %s  %s  t  %s " periodicity1
  periodicity2 sweepName)))
  ```

  Each choice has an associated value that describes what must be added to the expression signature.

  If you select “single”, then the value of numberOfOccurances put into the expression string is literally “0 0 nil nil”.

© 2023 Cadence Design Systems, Inc.

If you select “multiple”, the choice value is described by a formatted string and will look like: “<periodicity1> <periodicty2> t <sweepName>”. If the default values are used, then the choice resolves to: 1 1 t “trigger”.

Example 3: Signature described by a format statement: compression.ocn

Location:

(<CDS_INST_DIR>/tools/dfII/etc/tools/wavescan/measures/compress.ocn)

```
 9     analysis (nil
10               general  (nil
11                    ;; args (signal ...)
12            format ("compression(dB20(harmonic(%s, %s)), ?x %s, ?compress
%s)" signal harmNumber xpoint compressiondb)
```

The compression function is one of those examples where the expression does not conform to a simple rule of function name plus name/value pairs for the parameters. We have to use the format statement (line 12 above).

The format uses %s to substitute in the values of the named parameters (signal, harmNumber, xpoint, and compressiondb).

© 2023 Cadence Design Systems, Inc.

Example 4: Creating your own template

This example creates a template for calculator function named ‘trap’ that exists in a new category named ‘MyProject’.

Tip

To build a new template, it is advised to pick an existing similar template and modify it accordingly.

1. Define the function signature:
   - Function name
   - Signal parameters
   - Additional parameters

   For example to create a new function named trap with signature:

   ```
   trap(<signal> <from> <to>)
   ```

   The template signal section will contain a single signal named Signal. The parameter section will contain two parameters named From and To, both of type float.

2. Go to the source directory for template files:

   ```
   <CDS_INST_DIR>/tools/dfII/etc/tools/wavescan/measures
   ```

3. Find an existing similar template so that you do not have to start from scratch. For example, if the new function trap was similar to existing function clip:

   ```
   % cp clip.ocn trap.ocn
   % chmod +w trap.ocn
   ```

4. Define the function information in the header section. Replace the information indicated in boldface below. Redefine the category name as MyProject.

   ```
   ocnmRegGUIBuilder(
   '(nil
   function trap
   name trap
   description "short tool tip description for trap function”
   category ("MyProject")
   ```

5. Define the args (or format) statement so we have a rule for building the expression.

   For the trap example, no change is required in the args statement that comes from the clip template.

© 2023 Cadence Design Systems, Inc.

```
args (signal From To)
```

6. Define the signal parameter in the signal section.

   For the trap example reuse the clip example. Change the tool tip as required.

```
signals (nil
        signal1 (nil
                prompt "Signal"
                tooltip "signal to trap "
                )
```

7. Define the parameters

   For the trap example, reuse the parameter definitions from the clip template. Change the tool tip values as required

```
params (nil
            From (nil
                        prompt "From"
                        tooltip "Trap Start"
                        default 0
                        type float
                        min 0)
             To (nil
                        prompt "To"
                        tooltip "End Trap Range"
                        default 0
                        type float
                        min 0)
        )
```

8. Install the measures file into the template directory.

   Copy trap.ocn to `<USER_HOME>/measures` or `CSF_SEARCH_PATHS/measures`.

   For more information, refer Function Template Search Paths.

9. Add entry in catalog file by doing following tasks:

   - Install your new project catalog file.

© 2023 Cadence Design Systems, Inc.

```
% cp  <CDS_INST_DIR>/tools/dfII/etc/tools/wavescan/measures/
.skeMeasuresCatalog CSF_SEARCH_PATHS/measures.
```

For more information, refer Template Catalog Summary File.

- Edit your new project catalog file to add the new category and function name:
  - Add category name at line #2 after skeBeginCategory keyword
  - Add new category section before skeBeginMeasures.Math.

  Your `.skeMeasuresCatalog` file will look like as below:

```
skeBeginCategory
MyProject
Math
Modifier
RF Functions
Special Functions
Trigonometric
skeEndCategory
skeBeginMeasures.MyProject
trap;trap
skeEndMeasures
skeBeginMeasures.Math
exp;exp
dB20;dB20
```

Now run Virtuoso Visualization and Analysis XL and open calculator, you should be able to see trap function added in `MyProject` function category.

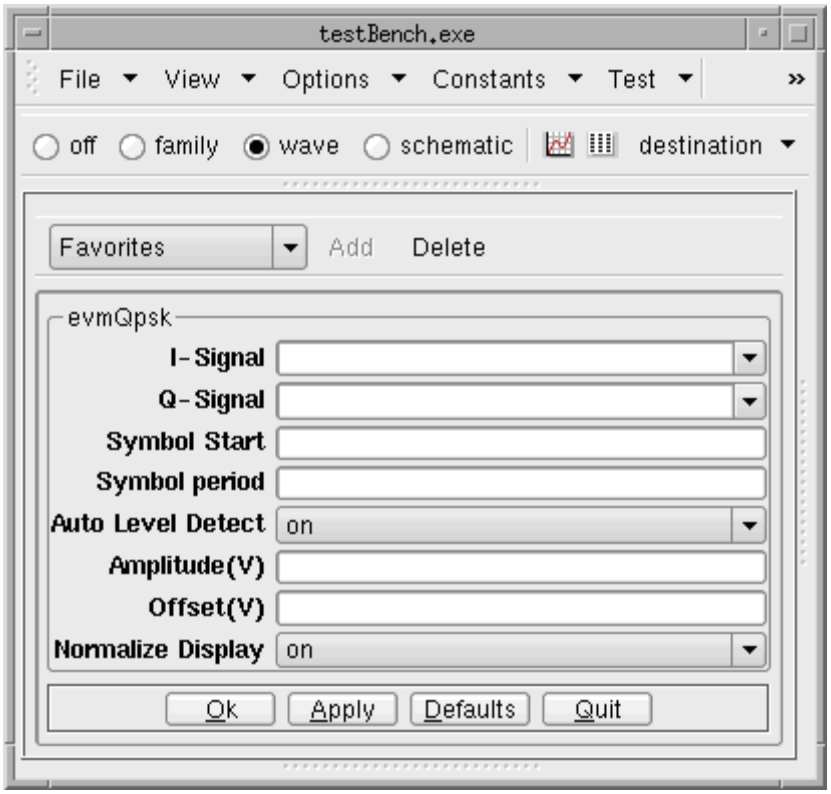
© 2023 Cadence Design Systems, Inc.

Advanced Features to provide GUI Hints

The template format supports the following advanced features:

- Sometimes it makes sense for two or more parameters to share the same row. An optional parameter property `gridRowHint <num>` directs Virtuoso Visualization and Analysis XL to layout all parameters with the same `gridRowHint value` on the same row.
- Sometimes a parameter controls whether one or more other parameters are enabled or disabled. An optional parameter property `guiEnableHint <target list>` controls whether other parameters are enabled.

Example: evm.ocn template (evmQpsk)



The above example is the GUI form for the original evmQpsk 6.1.0 template. The template contains no `gridRowHints` or `guiEnableHints`.

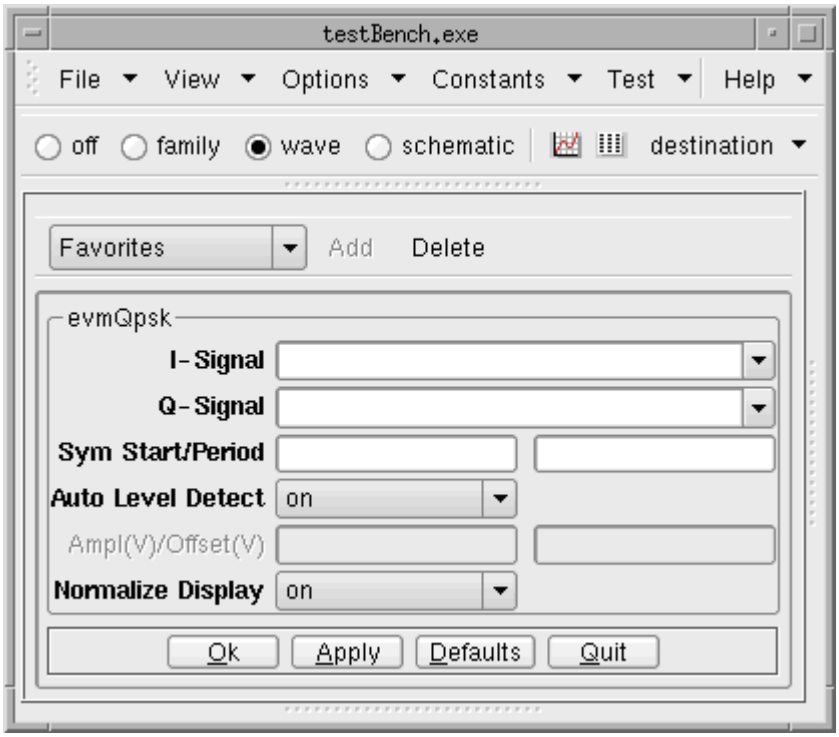
The next example shows how the two hint properties are used to:

- Place the Symbol Start and Symbol period parameters on the same row.
- Place the Amplitude and Offset parameters on the same row.

© 2023 Cadence Design Systems, Inc.

- The form has three columns instead of two to make the form more compact.
- The Auto Level Detect parameter will enable/disable the Amplitude and Offset parameters.

Example: evmQpsk with gridRowHint and gridEnableHints:



The following is a snippet of the evmQpsk template with modifications is italized:

```
delay (nil
    prompt "Sym Start/Period"
    tooltip "Symbol Start"
    type float
    required t
    guiRowHint 1
    min 0)
sampling (nil
    prompt ""
    tooltip "Symbol period"
    type float
    required t
    guiRowHint 1
    min 0)
```


© 2023 Cadence Design Systems, Inc.

Notes on template changes:

- The guiRowHint properties for delay and sampling have the same value (guiRowHint 1), so the prompt/value pairs for both parameters are put on the same row.
- Column 1 contains the prompt for the delay value. Column 3 is supposed to contain the prompt for the sampling value; however, I wanted to simplify the layout and have three columns instead of four. Look at the prompt value for the delay parameter: it describes BOTH the delay and sampling value fields. The prompt for the sampling parameter is set to an empty string (“”).

The following is a snippet of the evmQpsk template with modifications is italized:

```
Autoleveldetect(nil
    prompt "Auto Level Detect"
    tooltip "Auto Level Detect"
    type ( ("on" (t, nil, nil)) ("off" ("nil, %s, %s" voltage offset)))
    guiRowHint 2
    guiEnableHint ( (voltage off) (offset off) )
)
```

The autoleveldetect parameter is an enumerated type with values “on” and “off”. When it is “on”, there is no need to specify two other parameters: voltage and offset because those values will be automatically calculated.

If you want the GUI fields representing these other parameters to be disabled when autoleveldetect is “on”. The guiEnableHint target list gives the name(s) of the other parameters to enable (voltage and offset). Each target is specified as a list: the first name is the target name, followed by the set of autoleveldetect values that will turn that target on.

Examine the first target list for autoleveldetect’s guiEnableHint:

```
guiEnableHint( (voltage off) (offset off))
```

This hint tells the GUI that when the autoleveldetect enumerated value is “off”, the gui for the voltage parameter can be enabled. For any other autoenablehint value, the gui for the voltage parameter will be disabled.

© 2023 Cadence Design Systems, Inc.

© 2023 Cadence Design Systems, Inc.

Index

Numerics

1/x 1445

A

acos 1411
acosh 1417
asin 1423
asinh 1429
atan 1435
atanh 1441
avg 1019

B

bindkeys 565
bus
    creating 477
    expanding 491

C

calculator
    buffer 621
    constants 645
    display results in a table 653
    expressions
        building, algebraic mode 639
        building, RPN mode 638
    functions
        1/x 1445
        acos 1411
        acosh 1417
        asin 1423
        asinh 1429
        atan 1435
        atanh 1441
        avg 1019
        clip (SKILL) 1035
        compression 1046
        compressionVRI 1049
        conj 1379
        convolve 1052
        cos 1408
        cross 1055
        db10 1382
        db20 1385
        dbm 1062
        delay 1064
        deriv 1074
        dft 1077
        dftbb 1083
        exp 1454
        eyeDiagram 1103
        eyeHeightAtXY 1107, 1119
        flip 1147
        harmonicFreq 1185
        iinteg 1195
        int 1457
        integ 1201
        ipnVRI 1209
        ln 1460
        log10 1463
        mag 1394
        overshoot 1224
        ph 1397
        pow 1249
        psd 1255
        psdbb 1260
        pzbode 1269
        pzfilter 1273
        real 1391
        risetime 1278
        rms 1283
        sample 1298
        sin 1420
        sinh 1426
        spectralPower 1317
        stddev 1325
        tan 1432
        tanh 1438
        x**2 1469
        xmax 1365
        xmin 1368
        xval 1371
        y**x 1472
        ymax 1374
        ymin 1376
graph expressions, evaluating 225
memories

© 2023 Cadence Design Systems, Inc.

deleting 633, 652
saving 649
opening 604
selecting
signals 636
traces 637
signals, selecting 636
stack 620
var 673
cdsenv file, variables
calculator
height 987
rpnMode 973
width 986
xLocation 984
yLocation 985
graph frame
height 766
width 765
Results Browser
dataDirHome 969
historyLength 967
plotstyle 968
clip 1035
compression 1046
compressionVRI 1049
conj 1379
convolve 1052
cos 1408
cross 1055

D

db10 1382
db20 1385
dbm 1062
delay 1064
deleting objects 191
delta marker
placing 524
deriv 1074
dft 1077
dftbb 1083

E

editing
graph attributes 202
markers 520
exp 1454
eyeDiagram 1103
eyeHeightAtXY 1107, 1119

F

flip 1147

G

Graph Window
opening 136
graphs
attributes, editing 202
colors 190
customizing 189
objects 191
panning 192
zooming 193

H

harmonicFreq 1185

I

iinteg 1195
int 1457
integ 1201

L

ln 1460
log10 1463

M

mag 1394
markers
adding 495
delta marker
placing 524
editing 520
moving 520
moving

© 2023 Cadence Design Systems, Inc.

markers 520

N

newlink status 634

O

objects
deleting 191
overshoot 1224

P

panning graphs 192
ph 1397
pow 1249
psd 1255
psdbb 1260
pzbode 1269
pzfilter 1273

R

real 1391
risetime 1278
rms 1283

S

sample 1298
saving
session 31
session
saving 31
sin 1420
sinh 1426
spectralPower 1317
stddev 1325
supported data formats 26

T

tan 1432
tanh 1438
traces
analog representation of a digital trace 467
selecting 388
symbols, display 395

V

Virtuoso Visualization and Analysis tool components 23

X

x**2 1469
xmax 1365
xmin 1368
xval 1371

Y

y**x 1472
ymax 1374
ymin 1376

Z

zooming graphs 193

© 2023 Cadence Design Systems, Inc.

Virtuoso Visualization and Analysis XL User Guide

© 2023 Cadence Design Systems, Inc.