cadence®

Virtuoso Studio Known Problems and Solutions

Product Version IC23.1
November 2023

Printed in the United States of America.

Cadence Design Systems, Inc. (Cadence), 2655 Seely Ave., San Jose, CA 95134, USA.

Contents

Known Problems and Solutions in Virtuoso Studio IC23.1

This Known Problems and Solutions document describes the known issues and suggests the workarounds for these issues. Each issue is identified as a Cadence Change Request (CCR) number.

Important

Only the problems and solutions that were known at the time of release are described in this document. Log on to Cadence Online Support to view an up-to-date list.

The following list includes those products where known problems and solutions have been identified in the Virtuoso Studio IC23.1 release.

- Analog Library
- Cadence Application Infrastructure
- Cadence Job Monitor
- Cadence Library Manager
- Cadence SKILL IDE
- Command-Line IP Selector
- Cadence User Interface SKILL
- Design Data Translators
- HSPICE/SPICE Interface
- Spectre Interactive Environment
- SystemVerilog Integration Environment
- VHDL In
- Virtuoso Abstract Generator
- Virtuoso ADE Assembler

- Virtuoso ADE Explorer
- Virtuoso ADE Verifier
- Virtuoso Automated Placement and Routing Solution
- Virtuoso Design Environment
- Virtuoso Design Environment SKILL
- Virtuoso Design Rule Driven Editing
- Virtuoso Electrically Aware Design Flow
- Virtuoso Fluid Guard Ring
- Virtuoso Hierarchy Editor
- Virtuoso Layout Suite EXL
- Virtuoso Layout Suite XL
- Virtuoso Layout Viewer
- Virtuoso Module Generator
- Virtuoso Multi-Patterning Technology
- Virtuoso MultiTech Framework
- Virtuoso NC Verilog
- Virtuoso Parameterized Cell
- Virtuoso Parasitic Aware Design
- Virtuoso Relative Object Design
- Virtuoso RF
- Virtuoso Schematic Editor
- Virtuoso Symbolic Placement of Devices
- Virtuoso SystemVerilog Netlister
- Virtuoso Technology Database Checker
- Virtuoso Visualization and Analysis XL
- Virtuoso Width Spacing Patterns

Known Problems in Other Virtuoso Studio Products

If a product is not listed in the Known Problems and Solutions in Virtuoso Studio IC23.1 list, it implies that there are no known problems in the IC23.1 release for that product.

Known Problems and Solutions in Analog Library

This document describes the known issues with the Analog Library and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to Analog Library

CCR 1641594: Design variables used in a bsource expression are not copied in an ADE session

Description: Design variables used in an expression specified in the Edit Object Properties form for `bsource` cannot be copied to an ADE session because bsource does not support automatic detection of design variables.

Solution: You need to manually add each design variable used in the expression specified in the Edit Object Properties form for `bsource` and set its value in the ADE session.

Known Problems and Solutions in Cadence Application Infrastructure

This document describes the known issues with Cadence® Application Infrastructure and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to the Cadence Infrastructure

What's New in Cadence Application Infrastructure

General Issues

Problem: Application hangs on lock request to a remote system

Solution: Check whether the network lock daemons `lockd` and `statd` are running on the remote system. `lockd` and `statd` must always be running on every UNIX system that runs Cadence applications or stores data. Some applications that use read and write locks use the operating system's `fcntl` function, which relies on `lockd` and `statd` daemons. (CLS Edit locks no longer use `fcntl`.) If the daemons are running, but calls to them still hang, verify that you have the latest operating system patches.

For DFII applications, you can also try the following workaround:

➤ Set the following environment variable:

```
DD_DONT_DO_OS_LOCKS
```

Problem: Locking requests to a Linux machine fail

Description: Locking requests to a Linux machine fail with the following error:

```
"No locks available"
```

While CLS Edit locks no longer use `fcntl`, some applications that use read and write locks do use the operating system's `fcntl` function. Locking requests to a Linux machine will fail if the Linux NFS server does not support locking.

Solution: Upgrade your NFS software to a version that supports locking. For DFII applications, you can also use the following workaround:

➤ Set the following environment variable:

```
DD_DONT_DO_OS_LOCKS
```

Problem: Malformed Lock-Stake file

Description: Your Cadence application does not run and you get the following error:

```
*Warning* file /usr/xyz/CDS.log Malformed Lock-Stake file.
Failed to lock log file: /usr/xyz/CDS.log
```

Solution: A "Malformed Lock-Stake File" error message indicates that the Lock-Stake file is either empty or corrupted. A Lock-Stake file can be empty if it was created when the disk was full. A Lock-Stake file can be corrupted if it was manually edited or if the application or system exited abnormally.

Resolve the cause of the problem, for example, a full disk. If it is safe to remove the lock on the file, remove the Lock-Stake file (*`filename`*`.cdslck`) manually or with `clsAdminTool`.

Known Problems and Solutions in Cadence Job Monitor

This document describes the known issues with Cadence® Job Monitor and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Job Monitor

Cadence Change Requests

CCR 72452: cdsqmgr should give warning if hostname is invalid in config

Problem: cdsqmanager does not display a warning if there is an invalid host name in the configuration file. However, it ignores any invalid hosts.

There is no workaround available.

CCR 71543: More clear message when cannot find current working dir

Problem: When the job manager cannot find the current working directory, it executes the job using a temporary directory. It also reports the following error in the email sent to the requestor of the job:

```
PS: Failed to open given output file; stdout output included in this report
PS: Failed to open given error file; stderr output included in this report
```

even though no output or error file was specified for the job.

Solution: None. The job is executed as expected. You can ignore the error message.

CCR 37670: Job Monitor does not accept 00 in min as start time

Problem: In the Run form, if you type `00` in the minute part of the *Start Time* field, when you click *OK*, you get the following error:

```
Invalid start time specified.
Correct the problem and try again.
```

Solution: To specify 00 minutes, instead of typing in the minutes, select the minute part of the field and use the scroll bar to set the minutes.

CCR 25677: Stack trace on NT when selected an empty month

Problem: In the Run form, the *Start Date* field erroneously lets you select an empty month, (by selecting the empty item after *Dec*). When this happens, a stack trace is displayed in the window in which you started the job monitor.

Solution: None. This is only a display problem. The month defaults to the current month and the job runs as expected.

CCR 6914: Job Properties cut off when description is too long

Description: If the description in the Job Properties window is too long, the complete description is not displayed.

There is no workaround available.

Known Problems and Solutions in Cadence Library Manager

This document describes the known issues with Cadence® Library Manager and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to Library Manager

Introduction to Library Path Editor

CCR 1132056: Irrelevant Error while closing the cell view property form under Design Sync environment

Description: While adding a property to a DM managed cellview that does not have the `data.dm` property file, the cellview's `data.dm` file is created using the View Property Editor dialog box.

Now, if you click the *Cancel* button to discard these changes without adding any properties to the `data.dm` file, this file gets removed but the error message is displayed in CIW.

Solution: To avoid the error message, it is recommended to create the `data.dm` file by saving any first property using the View Property Editor dialog box.

CCR 501893: Unable to create new VerilogA view

Description: If you have Design Sync loaded as your design management tool, when you try to create a new cellview of type VerilogA with the Library Manager's *File – New – Cell View* command, the view is not created.

Solution: Use one of the following workarounds:

- Create a schematic view first. Close the schematic view, then create the VerilogA view.

- Create the VerilogA view from the Virtuoso CIW's *File – New – Cellview* command instead.

Known Problems and Solutions in Cadence SKILL IDE

This document describes the known issues with Cadence® SKILL IDE and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

SKILL Integrated Development Environment

Cadence Change Requests

CCR 560629: Design-Managed file not prompted for checkout when opened in the IDE

Description: In the SKILL IDE, if you open a Design-Managed SKILL file, you are not prompted for checkout and the file opens in read-only mode.

Solution: Check out the DM SKILL file outside the SKILL IDE first, and then proceed to make changes in the IDE.

Known Problems and Solutions in Command-Line IP Selector

This document describes the known issues with Command-Line IP Selector and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Topics

Introduction to Command-Line IP Selector

CCR 1791843: CLIPS cannot hide top-level instances in the 'Hierarchy' window.

Description: The option 'Hide Instances of Cells' does not work on the top instances in the 'Hierarchy' window. It only works on the instances down in the digital hierarchy.

Workaround: Currently, a workaround is not available.

CCR 1767826: Cannot force design elaboration when an external file is changed

Description: After the first round of design elaboration, the button *Elaboration* is deactivated. But, if it detects any changes in the existing top level xrunArgs file, the button is reactivated. However, CLIPS is unable to detect the changes made in external files, and the *Elaboration* buttons remains deactivated.

Workaround: Delete the input file in the window *Use Existing xrun files*, and add the file again. This will trigger the detection of a new file and activate the button *Elaboration*.

CCR 1813090: Current CLIPS state cannot be saved in a directory if the file name is same as the current sub-directory name

Description: When you save current CLIPS state in a directory where the file name is same as the current sub-directory name, then an error is reported. It fails to save the state. The sub-directory remains opened.

Workaround: Save CLIPS current state in a file name different from the current sub-directory name.

CCR 1830338: Model parameter in cds_globals_ip_0 can not be recognized by SFE in post-run

Description: CLIPS cannot generate the model parameters from design variables in ADE state to the netlist for an IP block which has switched to config view and is associated to an ADE state. Model parameters are generated in the cds_globals_ip_x block for each IP block, but these parameters cannot be recognized as global parameters in the analog engine.

Workaround: If you have model parameters in ADE state, which are associated with an IP block in CLIPS, you have to manually modify the netlist results. Merge all cds_globals_ip_x back to cds_globals and add all parameters into the final module. Additionally, model parameters cannot be scoped, so name conflicts of parameters from different IPs need to be solved before running them through CLIPS.

Known Problems and Solutions in Cadence User Interface SKILL

This document describes the known issues with Cadence® User Interface and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to Cadence User Interface

Cadence Change Requests

CCR 930231: How to redefine a callback for a field in a form?

Description: You cannot redefine a callback for a field in a form after the form is created and displayed.

For example, you have set the callback while creating a `<hiCreateButtonBoxField>` field in the `<hiCreateAppForm>` form and want to change the callback for the `<hiCreateButtonBoxField>` field later. You can determine the associated callback i.e., `<_callback>` from the properties of the field and overwrite it as shown below:

```
hiGetCurrentForm() -> field_ViewSummary ->_callback = list("printf(\"changed
callback\")");
```

If you check the properties of the field, the callback is changed. However, if you click the button, the old callback is called.

Solution: You can work around this situation in one of the following ways:

1. Design the field callback in a way where the callback is associated with a function that can be modified.
2. Delete the field from the form and add a new, identical field with a different callback.

CCR 377616: Modal dialog focus problems with twm NoTitleFocus setting

Description: In the twm window manager, sometimes fields in modal dialog boxes do not regain focus when you click them after clicking another window. This problem occurs when `NoTitleFocus` is set in the `.twmrc` file.

Solution: Do not use the `NoTitleFocus` setting with twm.

CCR 64048: Virtuoso applications go blank if you click a hyperlink after Netscape crashes

Description: If you click a hyperlink in a hypertext field after Netscape has crashed, Virtuoso application windows and forms go blank.

Solution: Kill the Netscape process.

General Issues

Problem: Dockable window gets smaller

Description: When you repeatedly undock and dock a window in a dock area that has multiple dockable windows, the window gets slightly smaller each time you dock it. This is due to an underlying Qt problem.

Solution: Resize the window after you dock it.

Problem: Cannot define draw-through commands and pop-ups for the same mouse button

Description: Any mouse button defined as a pop-up cannot be used as a draw-through command. If you use it, the pop-up menu is not usable and disappears after you release the mouse button.

Solution: Do not define a draw-through command and pop-up for the same mouse button.

Problem: Mouse button is difficult to double-click

Description: Some users find it difficult to double-click the mouse buttons.

Solution: You can define the timeout value for the double-click and the distance for the draw-through command. You can change the multiclick time from the User Preferences form on the *Set Options* menu of the CIW. Use the following command:

```
hiSetMultiClickTime(clickTime)
```

clickTime is the timeout interval in milliseconds between the two button clicks. The default is 200 milliseconds. You cannot set the multiclick time to less than 50 milliseconds. `hiGetMultiClickTime` returns the current setting of the timeout interval for the double click.

`hiSetDrawThruDelta(`*`delta`*`)` is the distance the mouse moved from the starting click to be interpreted as a draw-through. The default is 5 pixels. You cannot set an interval of less than 3 pixels. `hiGetDrawThruDelta` returns the current setting of the draw-through delta.

Problem: The MultiRep enter function is blocked by other enter functions

Description: Commands such as `Copy` and `Move`, which use the `enterMultiRep` procedure to allow operations across windows containing different cellviews, might not work correctly in certain situations. For example, you cannot `Copy` or `Move` from window A to window B if window B already has another enter function active.

Solution: If you cancel or terminate the enter function in window B, you can finish the `Copy` or `Move`.

Problem: Enter function `done` procedure is called with a variable number of arguments

Description: If you start an enter function with the `acceptString` flag set to `t`, the argument list passed to the `done` callback of that enter function has a different number of arguments, depending on whether you supply points or a string.

Solution: The following SKILL code using the `@rest` argument list keyword is one solution.

```
    ;
    ; This procedure acts as the entry point for the example.
;

procedure( testIt()
    enterBox( ?acceptString t ?doneProc "doneCB" )
)


;
; The following callback procedure first checks if a
; string is being passed in, then it checks if
; the function was canceled or finished successfully.
```

```
;

procedure( doneCB( @rest rest )
    if( length(rest) == 1 && stringp(car(rest)) then
        printf("The user entered the string: %s" car(rest))
    else
        if( cadr(rest) == nil then
            printf("The user canceled the command.")
        else
            printf("The user entered the box:%L"caddr(rest))
        )   ))
```

Problem: Hide/Cancel on options form hides the wrong options form

Description: If more than one enter function's options form is currently being displayed, clicking the *Hide* button of the non-active enter function's options form causes the active enter function's form to disappear. Clicking *Hide* again on the non-active form causes the active form to reappear.

There is no workaround available.

Problem: Drawthrough affects next enter function's option form

Description: The options form associated with an enter function will sometimes not appear even though "Options Displayed When Commands Start" is selected in the User Preferences form (`expertMode == nil`) if a drawthrough command was executed just prior to invoking the enter function.

Solution: Press the *F3* key to make the form appear.

Problem: Form field format can affect form values

Description: When you specify a format, such as `%gpF`, SKILL interprets the `p` portion of the format to mean pico (10^{e-12}). Any character immediately following a number, if any, is taken as a modifier.

Solution: Leave a space between the format of the input and the actual number (for example, `%g pF`) or put the units in a separate field (such as a cyclic).

Problem: Toggle fields are not displayed with correct values

Description: If you place a toggle field in a form before it displays and then change using `form->togglefield->value = newValueList`, the form can show incorrect values when initially displayed.

When you create a form, a toggle field has its values stored at two levels: once at the toggle field level and once at the toggle item level. There is no way to keep these two entities the same once you create the form but have not yet displayed it. Thus, the values stored at the toggle item level override.

Solution: Set or query `form->togglefield->toggleitem->value`, rather than set the value at the field level to ensure that the form reflects the new values when it displays for the first time.

Problem: Toggle button text can be truncated if the field prompts are long

Description: On one-dimensional forms, the input area of all fields is left-justified to the end of the longest field prompt. If prompts are long, the input area of fields might be limited because of the fixed width of one-dimensional forms. In the case of toggle and radio fields, these buttons are clipped with no way of seeing the missing buttons if the space provided is inadequate.

Solution: You can split the field prompts and input area into two separate lines using standalone labels. Or you can specify the width of the field on a two-dimensional form. Toggle and radio fields can also have their buttons placed on multiple lines using the `?itemsPerRow` parameter.

Problem: Format specification and field value problem

Description: Form fields format whatever data is in the type-in field. If you supply a format string with your type-in field and that format appears in the field, you can have problems if you change the value of the field but do not remove the format string.

If a string field has a format of "%s inches," formatting an input of "10" produces "10 inches" in the field. If you then change only the "10" portion of the field to "20," but do not remove the "inches," the field shows "20 inches inches" because the "20 inches" shown in the field was taken as input and then formatted.

Solution: Remove both the data and the format string when you change the data and format string.

Problem: Entering leading zeroes in numeric fields results in octal interpretation

Description: If a number is immediately preceded by a zero, SKILL interprets this as an octal number.

Solution: Use the *Backspace* key to delete the zero before you type a new number.

Problem: Two-dimensional forms are sometimes illegible because of large font sizes

Description: If you changed your label font using `hiSetFont` to a larger font than the default label font size (`"-*-helvetica-bold-r-*-*-12-*"`), some two-dimensional forms might display illegibly. The labels in some two-dimensional form fields can overlap.

Solution: Create forms with a normal font size. To return the font to the default (so that the forms display correctly), execute the following SKILL function:

```
hiSetFont("label" "-*-helvetica-bold-r-*-*-12-*")
```

Problem: Integer field in a form displays incorrectly if the integer is large

Description: If you type a large integer, such as 10000000000000, into an integer field that has a range of positive integers only (such as `range = list(1 nil)`), you can get the following warning:

```
*WARNING* int is less than min range (1 nil) value - setting back to previous
value.
```

However, the field might not be reset correctly to the previous value, and the field value is set to a random number. The field reflects this incorrect value.

Solution: Cancel the form and redisplay it so the field is reset to its previously saved value. Or type in a new value for this field.

Use a float field if you want the field to accept larger numbers.

Problem: Return behavior from `hiDisplayForm` is unpredictable

Description: `hiDisplayForm` does not return until you dismiss the form. If multiple forms are displayed, any code after the `hiDisplayForm` of the first form displayed is not executed until all other forms are dismissed. For example:

```
procedure(form1()
        hiDisplayForm(form1)
        printf("form1 down\n")
)

procedure(form2()
        hiDisplayForm(form2)
        printf("form2 down\n")
)
```

```
form1(); form1 is displayed
form2(); form2 is displayed
<Cancel form1>
<Cancel form2>
"form1 down"
"form2 down"
```

The `printf` statement in `form1` is not executed until both forms are dismissed.

Solution: Forms “block” as they have in previous versions of the software. Do not write code that depends on this blocking behavior because you cannot control how many forms are pushed on the stack (as in the example above).

Instead, write code in the form of callbacks, which always execute when the form’s *OK*, *Apply*, or *Cancel* buttons are pressed. Also, `hiCreateAppForm` has an argument, *`?dontBlock`*, which prevents programmatic blocking for that form.

Problem: `hiDisplayForm` in `prog` leaves local variables accessible from the CIW

Description: The blocking behavior of forms can leave local variables in the procedure `prog` or `let` available to the user.

For example:

```
field = hiCreateStringField(
?name 'x
?value "x"
)
hiCreateForm( 'form
"form"
"t"
list( field )
)
        a = "outside"
        procedure( test()
           prog( ( a )
        a = "inside"
        hiDisplayForm( form )
       )
    )
    [1] load the following code.
    [2] print the value of a ("outside")
    [3] execute test() (form pops up)
    [4] print the value of a ("inside"!)
    [5] cancel the form.
    [6] print the value of a ("outside")
```

Solution: Use non-program-blocking forms by using `hiCreateAppForm` and pass `?dontBlock` as one of the arguments. Refer to the *User Interface SKILL Functions Reference* for more information about the `hiCreateAppForm ?dontBlock` argument.

Problem: A float field incorrectly displays the value input if the float value exceeds a maximum

Description: A float field displays the value incorrectly if you input a large number.

Solution: If the number entered is not in scientific notation (for example, `1e20`) or does not contain a decimal point, it is converted into an integer (which may overflow) before being converted into a float. This results in an incorrect number.

To ensure correct translation, always include a decimal point at the end of the whole number or use scientific notation.

For example, instead of

```
100000000000000000000
```

use one of the following:

```
100000000000000000000.
100000000000000000000.0
1e20
```

Problem: Form names with spaces cause problems

Description: Whenever a space character appears in a symbol of a form name, callbacks for the field and form generate error messages.

Solution: Don't include spaces in the form name.

Problem: Enlarging opus fonts corrupts most UIs

Description: Using fonts larger than the default fonts for text and labels can cause problems such as the text being obscured or not lining up correctly in forms and windows.

Solution: Either do not change the default fonts, or try to choose fonts that are approximately the same size as or smaller than the default fonts.

Problem: modifyCallback does not work correctly with all fields

Description: The `?modifyCallback` argument does not work as expected with all fields, especially with fields that take points, such as bounding box fields, point fields, and point list fields.

Solution: Do not use the `?modifyCallback` argument with any fields except string fields. Specifically, do not use the `?modifyCallback` argument with the following functions: `hiCreateBBoxField`, `hiCreateFloatField`, `hiCreateIntField`, `hiCreateListField`, `hiCreatePointField`, `hiCreatePointListField`. You can use this argument with `hiCreateStringField` and `hiCreateComboField`.

Problem: Programs started from Virtuoso have the locale set to C, which is not the expected locale

Solution: For all Virtuoso applications on all platforms, the locale is automatically set to C.

If you want to restore your original locale settings in a script or shell that is invoked from a Virtuoso application, source one of the following scripts:

For Bourne shell:
your_install_dir`/tools/dfII/samples/local/cdsOrigLocale.sh`

For C shell:
your_install_dir`/tools/dfII/samples/local/cdsOrigLocale.csh`

For example:
`source` *your_install_dir*`/tools/dfII/samples/local/cdsOrigLocale.csh`
– or –
`.` *your_install_dir*`/tools/dfII/samples/local/cdsOrigLocale.sh`

Problem: Calling `hiDisplayListBox` more than once causes errors

Description: If you call `hiDisplayListBox` more than once with the same list box symbol, without closing the previous list boxes, multiple list boxes are displayed and the list box symbol is overwritten.

As a result, only the last list box displayed can be closed. The *OK* and *Cancel* buttons on the other list boxes do not work; the boxes remain on the screen until the end of the session.

Solution: Use `hiShowListBox` instead of `hiDisplayListBox`. When you call `hiShowListBox` more than once with the same list box symbol and the list box symbol is already bound to a list box, `hiShowListBox` raises the list box to the top instead of creating a new copy.

Problem: Strokes editor might refer to applications that don't exist

Description: The setup files for the Strokes editor might refer to applications that don't exist. As a result, you might see warnings about "No application registered for prefix xxx" when loading `prependInstallPath("etc/sted/stroke.il")` or `prependInstallPath("etc/sted/defstrokes.il")`.

Solution: If you see warnings, correct or add the appropriate application prefix to `stroke.il` or `defstrokes.il`, then load these files again.

Problem: Stroke Recognition should not nest non-nestable commands

Description: When you use strokes to enter commands, commands that normally do not nest may nest instead of being cancelled. For example, if you choose Copy with the stroke C while currently in a Move command, when the Copy is finished, the Move command will again be active. In this example, if you had selected Copy via a bindkey, then the Move command would have been cancelled.

There is no workaround available.

Problem: `hiRepeat` repeats the wrong command

Description: The `hiRepeat` function repeats the last command executed, whether you execute the command from a menu or bindkey or type it into the CIW.

A common problem occurs when you use a keyboard bindkey to cancel or end an interactive command. The `cancelEnterFun` or `finishEnterFun` function is the last command executed instead of the `enter` function.

Solution: The CIW displays the command that will be repeated instead of `hiRepeat` in the mouse bindings line. Verify that this is the correct command before using `hiRepeat`.

Problem: Problems differentiating between parameters and properties

Description: It is difficult to differentiate parameters, default parameters, and properties on the Properties form.

Solution: The Properties form lists items in the following order:

1. Attributes
2. Default Parameters

3. Parameters
4. Properties (in alphabetical order)

Problem: Value listed as a possible choice can be ignored when only one possible choice is specified

Description: When you add an enumerated string property using the Add Property form, if only one possible choice is specified and this choice doesn't equal the specified value, a string property is created with the specified value. The other value listed as a possible choice is ignored.

Solution: Include the value as a possible choice.

Problem: `hiReplayFile` doesn't work when called in a replay file

Description: You cannot nest the `hiReplayFile` function in a replay file.

Solution: Run the `hiReplayFile` function interactively, then use the log file generated from the session as a replay file.

Problem: An empty window is brought up when you view a file that doesn't exist

Description: When you call `view` on a file that doesn't exist, an empty text window is created and displayed.

Solution: Close the empty window and try `view` again with another file.

Problem: The draw-through area select across windows seems to use the screen coordinates instead of the window coordinates

Description: The draw-through area select does not work across equivalent windows because the coordinates are always calculated relative to the window the draw-through was started in.

Solution: Use the `geSingleSelectBox` function, which is bound to *Shift-A*. The `geSingleSelectBox` function makes a two-points selection instead of a draw-through selection.

Problem: `hiResizeWindow`, `hiZoomAbsoluteScale` uses old window fit coordinates

Description: If `hiZoomAbsoluteScale` is called immediately after `hiResizeWindow` (on the same line of the CIW or within one function call), the Zoom will behave as if the window had not been resized.

Solution: Call `hiFlush` after calling `hiResizeWindow` and before calling `hiZoomAbsoluteScale` (or any function that relies on the size of the window).

Known Problems and Solutions in Design Data Translators

This document describes the known issues with Design Data Translators on OpenAccess and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

This document contains the following sections:

Related Topics

Overview of the Translators

Design Data Translators SKILL Reference

XStream

CCR 1177790: Perform "unlockAll" on each hierarchy of the layout does not cleanly erase color locking

Description: Removed color locked shapes may still exists in the GDS after XStream Out. In some cases, the data that has gone through mptUpdate may contain “lock override” anchors created using ICADV12.1 ISR3 or prior. The 'Lock override prohibition' option in ICADV12.1ISR4 prevents “unlockall” operation to remove the “lock override” anchors during descending and edit from top to bottom. As a result, color locking is not removed completely in the layout prior to XStream Out. You may observe undesirable color locked shapes in the gds after XStream Out.

Solution: To ensure that the color locked shapes remove properly, you have to unlock all from bottom to top hierarchy.

CCR 1175605: Variant is not generated for instance marked by ocalocking during 1-1 color mapping (Anchor in full-color for ISDA)

Description: A limitation prevents XStream Out to translate the color defined by the coloring engine into the output GDSII.

Solution: There is no workaround available.

CCR 734056: PVS LVS mismatch on using gds created from Pcell layout in IC61, OK within DFII

Description: As per the following code, users may not get the correct value of `bBox` after the master cellView is closed.

```
devId = dbOpenCellViewByType("VIP50CLZ3" "NPN" "layout")


obj = dbCreateParamInst(pcCellView devId nil 0+spacing:0 "R180" 1 list("Nb"
"string"  NumBase))


dbClose(devId)


rs_LL = car(obj~>bBox)
rs_UR = cadr(obj~>bBox)
```

`rs_LL` and `rs_UR` may not have correct values as master cellView has closed before calling `obj~>bBox`.

Solution: To get the correct value of `bBox`, users need to keep cellView open till it queries for `bBox`, as mentioned in the code below:

```
devId = dbOpenCellViewByType("VIP50CLZ3" "NPN" "layout")
obj = dbCreateParamInst(pcCellView devId nil 0+spacing:0 "R180" 1 list("Nb"
"string"  NumBase))
rs_LL = car(obj~>bBox)
rs_UR = cadr(obj~>bBox)
dbClose(devId) <----- move here in the end
```

CCR 488466: Object map is not loaded correctly during strmout

Description: When loading an object map file in the Objects tab of Stream Out GUI, the form should be populated with object name and other relevant data. However, in this release, the object name field is not displayed in the GUI.

Solution: Suggest using the strmout command with `-objectMap` command line option.

CCR 415118: XStream template file should be loaded if lib name not set

Description: Currently there is no possibility to initialize the XSTREAM IN and OUT forms skill as it was possible with PIPO (PCR 908046). This is done through automatically loading a template file.

While loading the template file, it is mandatory to include the stream file name (strmFile) and the library name (library)in the template file. In case the names are not included, first a popup window shows up with the message that the strmFile name and the library name is required and then the template file loading is aborted.

Solution: Suggest either removing the option ‘library’ or option ‘strmFile’ from the template file or have a dummy library name in the template file.

CCR 277193: Two different binaries for msgHelp exist in IC5141USR3

Description: The message help system for XSTREAM can be accessed through a binary called `msgHelp` located in the directory `$CDS/tools/dfII/bin/msgHelp`. In the release IC 6.1.0, a new binary with the same name exists in the directory `$CDS/tools/bin`.

Solution: Create an alias for the XSTREAM message help system to point to the correct `msgHelp` utility.

In the following example, an alias named `xstreamMsgHelp` is created to point to the correct `msgHelp` directory.

```
alias xstreamMsgHelp "$CDS/tools/dfII/bin/msgHelp"
```

SpiceIn

CCR 442828: SpiceIn should not try to map model names to schematic components

Description: SpiceIn parses the model path and tries to map model names found in the library sections to cell names in the reference libraries. This should not be the case since these representations doesn't exist in the DFII library and only exists in simulation models files.

Solution: There is no workaround available.

CDL Out

CCR 278800: nino cdlin postfix illegal character to subtype string

Description: The value of the `subtype` property created in the database might contain illegal characters. This might happen due to the uninitialization of the variable used to create this property by CDL In.

Solution: There is no workaround available.

LEF/DEF

CCR 83875 LEF In does not accept LEF file generated by CDBA LEF Out

Problem: If the macro pin names contain hierarchy delimiter characters, CDBA LEF Out, while writing out the macro pin information, does not escape the hierarchy delimiters.

When you create a technology library using OpenAccess LEF In from this output file, the file is not read and the translation fails.

Solution: The LEF file should be manually edited so that if pin names contain hierarchy delimiters, these should be escaped.

CCR 53381: Routesaspaths makes NETS vias written in SPECIALNETS.

Problem: If a net in the design consists of disassembled routes (paths and vias) and you perform DEF Out of that design, it does not write the geometries associated with the net.

Solution: To write all the geometries associated with the net consisting of a disassembled routes (paths and vias), first assemble paths and vias to routes.

Allegro Translator (IC23.1 Virtuoso MultiTech Framework Only)

CCR: 1777824 SIP_OA:Design with TILP's crashes while exercising DEF/ STREAM

Problem: Designs with technology independent layout Pcells (TILP), created when package layout from Allegro is imported into Virtuoso, crash during XStream or DEF translation.

Solution: XStream and DEF translators do not support translation of package design components. There is no workaround available.

CCR: 1715599 SIP_OA: RatsNest /TPoint are not translated to OA

Problem: The following components of Allegro package design cannot be imported into Virtuoso using the Allegro Import translator:

- Ratsnest intersection, which is the point where three paths intersect.
- Tpoint or a ratsnest T, which is a point in the physical layout of a net that indicates the signal path splits into multiple paths.

Solution: Allegro Import translator does not support translation of Allegro package layout components, RatsNest intersections and Tpoint.

Known Problems and Solutions in HSPICE/SPICE Interface

This document describes the known issues with HSPICE/SPICE Interface and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to HSPICE/SPICE Interface

CCR 491344 The OSS HSPICE script should not check for existence of hspice.exe

Description: The 276 interface HSPICE script still checks for the existence of the hspice.exe file in the HSPICE installdir, but this file is no longer provided by Meta.

Solution: Create a link in `<installdir>/bin/hspice.exe` to point to `<installdir>/bin/hspice`.

CCR 491338 si hardcoded to look for awk in /bin

Description: The “si” is hardcoded to look for “awk” at /bin.

Solution: Requires a link from `/usr/bin/awk` to `/bin/awk` on `hppa` and `ibmrs`.

CCR 54032 Remote HSPICE fails if key commands are aliased

Description: If the user aliases commands used by the remote HSPICE script to nonstandard UNIX, remote HSPICE fails. For example, if “rm” is an alias to “rm -i,” the script fails.

Solution: Change the script to use the full path to the command as in `/usr/bin/rm` or escape the command with “/,” e.g., /rm.

Known Problems and Solutions in Spectre Interactive Environment

This document describes the known issues with Spectre Interactive Environment and suggests workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Documents

- Spectre Interactive Environment User Guide
- Spectre Interactive Environment What's New

CCR 1979448: The Force and Release triggers are effective only in the first simulation run

Description: The Force and Release triggers specified in the Triggers and Actions table are effective only in the first simulation run after they are added. If you stop and rerun the simulation, these triggers do not work.

Solution: Before every simulation run, right-click the net in the *Netlist Hierarchy* assistant and choose *Force Net*, and enter a value to be forced. You need to do this even if the triggers are already added. In addition, you can delete the previously added triggers.

CCR 1899706: The mouse pointer is not visible while a simulation is in progress in the Spectre Interactive Environment.

Description: If you are using a VNC viewer, at times, the mouse pointer becomes invisible when a simulation is in progress in the Spectre Interactive Environment. As a result, you cannot click the buttons on the toolbar to pause or stop the simulation.

Solution: Upgrade to the latest version of VNC viewer and use a new VNC session.

Known Problems and Solutions in SystemVerilog Integration Environment

This document describes the known issues with Virtuoso® Verilog Environment for SystemVerilog Integration (SystemVerilog Integration Environment) and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introducing SystemVerilog Integration Environment

CCR 1104924: Simulation settings do not update when you switch between the NC Verilog Environment and the SystemVerilog Integration Environment

Description: The simulation settings for NC Verilog Environment do not update when you perform the following steps.

1. Start a Virtuoso session.
2. Use the SystemVerilog Integration Environment.

   Launch the environment for a design and initialize the run directory. You can also customize settings and netlist the design.
3. Exit the SystemVerilog Integration Environment.
4. Launch the NC Verilog Environment.

In this case, the NC Verilog Environment fails to netlist a design because the simulation settings, including view lists, do not update.

Similarly, when you first use the NC Verilog Environment in a session and then use the SystemVerilog Integration Environment, the simulation settings do not update.

Solution: Ensure that the flag `vlogifVicSimSimulator` is not set in your run directory and start a new Virtuoso session. Then use the SystemVerilog Integration Environment or the NC Verilog Environment.

Known Problems and Solutions in VHDL In

This document describes the known issues with VHDL In and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introducing VHDL In

CCR 483839: CDS_INST_DIR works as a reserved variable for vhdlin

Description: The CDS_INST_DIR is an environment variable that is evaluated to cds_root verilog while running VHDL Import.

Solution: Use a variable name other than CDS_INST_DIR in the cds.lib file.

CCR 483244: VHDL design having generate statement is imported as text instead of schematic

Description: VHPL APIs do not provide details of generate constructs while using the ncvhdl parser, therefore if a VHDL design has generate constructs, the design is imported as text instead of schematic.

Solution: This is a known issue and currently no solution is provided for this.

CCR 448134: VHDL In will not be able to import configuration bindings

Description: If there are VHDL configurations in VHDL design to be imported, the binding information is available in the generated schematic.

Solution: This is a known issue and currently no solution is provided for this.

Known Problems and Solutions in Virtuoso Abstract Generator

This document describes the known issues with Virtuoso Abstract Generator and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Topics

- *Virtuoso Abstract Generator User Guide*

CCR 2824651: "ExtractPurposeList" option is not supported for Power Net extraction in IC23.1

Description: The ExtractPurposeList SKILL option introduced in IC23.1 is currently supported only for signal net extraction, and not for power net extraction. This option can be used to improve the performance of signal net extraction in dense designs having a large shape count.

Solution: There is no solution or workaround available.

CCR 2429070: Unsupported options for Power net extraction in IC23.1

Description: The following options are currently not supported for power net extraction:

- ExtractDistPwr
- ExtractWidthPwr

Solution: There is no solution or workaround available.

CCR 2765133: Trim shapes are not honored for a simple GSPEC (eg M1 M1) for Power net extraction in IC23.1

Description: This is a rare design scenario. To account for performance optimization during power net extraction, the trim engine is only triggered when non-simple geometric specifications (that include some sort of geometrical manipulation operations on the metal

shapes using logical operators) are defined. Trim shapes on metal layers are automatically derived from the technology specifications. There is no change in the approach.

Workaround: As a workaround, you can specify the geometrical specifications with the required metal purposes for the impact of the trim shapes on the metal layers to be accounted for during power extraction.

CCR 2733481: Abstract Generator: Support for groupArrays

Description: Currently Abstract Generator does not process group array objects.

Solution: There is no solution or workaround available.

Known Problems and Solutions in Virtuoso ADE Assembler

This document describes the known issues with Virtuoso® ADE Assembler and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Documents

Getting Started with Virtuoso ADE Assembler

Introduction to Virtuoso Variation Option

What's New in Virtuoso ADE Explorer and Virtuoso ADE Assembler

CCR 2808717: Oppoint expressions do not evaluate when you use an extracted view in the Switch View list

Description: If you add an extracted view to the beginning of Switch View List on the Environment Options form for a test and then run a simulation for the test, the results of the operating points are not plotted. Neither the operating points are found in the save statement of the netlist.

This issue is observed with the extracted views only. Smart Views are handled as expected when added to Switch View List.

Solution: Use a config view to point to the extracted view instead of the switch view list. After this, config sweeps work and operating points will be plotted.

CCR 2794501: Bindkeys are not set when the tool name is used as `ADE Assembler`

Description: The `hiGetBindKeyPrefixList()` function returns bindkeys for both application names, `ADE Assembler` and `assembler`. However, if application name is set as `ADE Assembler` in `hiSetBindKeys()`, bindkeys are not set.

For example,

```
;;the following does not work
hiSetBindKeys( "ADE Assembler" list(
    list("Ctrl<Key>3" "print(\"Ctrl 3 working with ADE Assembler\")")
    list("Ctrl<Key>4" "print(\"Ctrl 4 working with ADE Assembler\")")
))
;;but this does
hiSetBindKeys( "assembler" list(
     list("Ctrl<Key>1" "print(\"Ctrl 1 working with assembler\")")
     list("Ctrl<Key>2" "print(\"Ctrl 2 working with assembler\")")
))
```

Solution: Use the application name as `assembler` when setting bindkeys for Virtuoso ADE Assembler.

CCR 2692214: Netlist generation fails if a component name is missing from the CDF of a VerilogA view

Description: Generally, for VerilogA views, CDF is automatically generated and the names of all the components used in that view are added to the CDF. If the CDF is missing for a VerilogA view or the names of some components are missing from the CDF, netlist generation fails for that view and the following error is reported:

```
*Error* eval: undefined function - _nlCustomPatchVerilogaDebugTrace
```

Solution: Specify the following statement in the `.cdsinit` file:

```
(procedure _nlCustomPatchVerilogaDebugTrace(@rest args) t)
```

CCR 2408738: The `spectre.envOpts netlistBBox` environment variable does not work in ADE Explorer and ADE Assembler

Description: You cannot use the `netlistBBox` variable to resize the netlist window in ADE Explorer and ADE Assembler.

Solution: Currently, there is no way to change the default window size. You need to manually resize each window.

CCR 2259524: If a Monte Carlo simulation is run with two or more tests that contain the same set of swept local variables, the Yield results view does not show results

Description: When multiple tests contain the same set of swept variables, the Yield results view does not show results for those tests. The results appear fine in the Detail and Detail-Transpose views.

Solution: Use a run plan or global variables to sweep a common set of variables for multiple tests.

CCR 1986691: When the test setup contains a pre-run script, the Report Identical Histories Before Run feature ignores any changes in the pre-run script done outside ADE Assembler

Description: If you edit the pre-run script for a test using a test editor outside ADE Assembler, the Report Identical Histories Before Run feature is unable to detect the changes, and does not report any difference in the setup.

Solution: If you have enabled reporting of identical histories, ADE Assembler displays a message reporting that the active setup is similar to the setup of an existing history. On that message, you can confirm that you still need to run the simulation. Alternatively, you can choose to disable reporting of identical histories.

CCR 1963792: Overridden specifications are ignored for fault expressions in the *Fault* results view

Description: The *Detail* and *Fault* results view show different values for the `faultexpr` outputs with overridden specifications. This is because currently, the *Fault* result view ignores the overridden specifications.

Solution: Currently, a workaround is not available.

CCR 1698716: The Plot and Save check boxes are disabled when using a config to bind an instance to a DSPF view

Description: If you are working on a design that uses a config to bind an instance to a DSPF view, you cannot toggle the *Plot* and *Save* check boxes in the *Outputs* section of ADE L or the *Outputs Setup* pane of ADE XL, Explorer, and Assembler, because the check boxes are disabled.

Solution: To enable these check boxes, set the `allowInvalidObjectSelection` environment variable, as shown below:

- To set this variable in the `.cdsenv` file, use the call:
  `auCore.selection allowInvalidObjectSelection boolean t`

- To set this variable in the .cdsinit file or CIW, use the call:
  envSetVal("auCore.selection" "allowInvalidObjectSelection"
  'boolean t)

Alternatively, set the following SKILL variable in the .cdsinit file, as shown below:
_allowInvalidObjectSelection = t

CCR 1696853: Enable or disable status of sweeps and corners are ignored if you edit the swept variable or corner in the setup and update the points table

Description: If you disable any swept point or corner on the Run Preview tab, and then add or edit the swept variable or corner in the setup, all the points are automatically enabled when you update the points table. For example, if you have a swept variable with values as 1:0.1:1.5. You open Run Preview and disable some sweep points. Next, you edit the sweep values of that variable in the Data View pane value and set its values as 1:0.1:2. If you update the points table in the Run Preview tab, all points become enabled.

Workaround: Currently, a workaround is not available.

CCR 1657981: dyn_floatdcpath violations are not visible in the Checks/Asserts view

Description: When using MMISIM 16.1, the dyn_floatdcpath violations do not appear in the Checks/Asserts view. However, these violations are visible in the XML report for the dynamic checks.

Workaround: Add the following in the *Additional Arguments* field on the *Miscellaneous* tab of the Analog Options form:

```
leaki_sim_mode=0
```

CCR 1582503: The *Quick Plot* command ignores reference waveform when plotting outputs with wave spec

Description: When you use the *Quick Plot* command to plot outputs with wave specification, it does not plot the reference waveform as done by the *Plot* command.

Solution: Use the *Plot* command to plot the outputs with wave specification.

CCR 1566735: Timing delay and bias level do not shift the reference waveform regions

Description: Currently, the timing delay value and bias level specified in the waveform compare options is not reflected by shifting the waveform in the ViVA graph window. However, the comparison algorithm does take the time delay value into account correctly. This is an incorrect representation of the graphical data shown.

Solution: Currently, a workaround is not available.

CCR 1565256: Time exclude regions are not indicated in the plotted waveforms

Description: Currently, the time specified for exclusion in the waveform compare options is not shown in the plotted waveform.

Solution: Currently, a workaround is not available.

CCR 1476640: The Virtuoso_Variation_Analysis_Op (95510) license is not checked in when the maestro view is closed after running an LDE simulation

Description: If you run a flow that checks out the Virtuoso_Variation_Analysis_Op (95510) license, such as an LDE re-simulation flow, the license is not checked in after the maestro view is closed.

Solution: Currently, a workaround is not available.

CCR 1475531: When the parasitics mode is set to `None` or `Estimated`, the ADE GXL license is checked out intermittently

Description: When the parasitics mode is set to `None` or `Estimated`, the ADE GXL license is checked out intermittently

Solution: Currently, a workaround is not available.

CCR 1408886: Wrong number of enabled asserts is shown in the Checks/Asserts tree in the Data View assistant

Description: After editing the checks and asserts in the Checks/Asserts Assistant, the *Checks/Asserts* tree in the Data View Assistant does not show the updated number of checks.

Solution: Toggle the *Cellviews* subtree in the *Checks/Asserts* tree to update the counts.

CCR 1366927: The Checks/asserts flow does not map net or instance names for the estimated or extracted parasitic views

Description: When running checks/asserts with parasitic data (extracted or estimated), instance and nets are reported in the namespace of the extracted or estimated data, but no name mapping is done from the schematic names in the checks/asserts statements or outputs.

Solution: Currently, a workaround is not available.

Known Problems and Solutions in Virtuoso ADE Explorer

This document describes the known issues with Virtuoso® ADE Explorer and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Getting Started with Virtuoso ADE Explorer

Introduction to Virtuoso Variation Option

Virtuoso ADE SKILL Reference

CCR 2710126: Netlist creation in ADE Explorer takes a long time

Description: You notice that the netlisting time for extracted views has increased since ISR29. This is due to an alias checking feature for the Virtuoso Electromagnetic Solver. This leads to a slower netlist creation.

Solution: If you do not use Virtuoso Electromagnetic Solver, then set the following shell variable in the terminal before opening Virtuoso to disable alias checking:

```
setenv NO_CDS_VEM_LVS_DRIVEN_FLOW t
```

CCR 2619866: When you are using Spectre 21.1, RTT simulations run in batch mode

Description: When you run an RTT simulation using Spectre 21.1, RTT is switched to batch mode. This is because Spectre 21.1 runs on RHEL7 that is not compatible with Virtuoso IC6.1.8 or Virtuoso IC23.1 and therefore, Spectre SKI interactive mode is not run.

Solution: Currently, there is no workaround.

CCR 2231027: Expressions or variables used to substitute time values in outputs result in errors

Description: If you specify an expression or variable name for an argument that requires a time value, the expression gives an error because the floating value returned by them is not a correct time value.

In the following example, expression `exp` fails to evaluate because an expression named `Event_Var` is used to specify time for `drplPacVolGnExpDen`:

```
Event_Var = car(sweepValues(v(\"/RFout\" ?result \"pac_sampled\")))
exp = drplPacVolGnExpDen("value(v(\"/RFin\" ?result \"pac-pac_cross\")
'\"eventtime\" Event_Var)" '(0) nil)
```

Solution: Specify the sub expression directly instead of using a variable. For the example, the `exp` expression can be written without using `Event_Var`.

```
exp = drplPacVolGnExpDen("value(v(\"/RFin\" ?result \"pac-pac_cross\")
'\"eventtime\" car(sweepValues(v(\"/RFout\" ?result \"pac_sampled\"))))" '(0) nil)
```

CCR 2113149: If the file type is not same as the extension specified in file name on the Export Results form, the extension in the file name is used

Description: When you export results using the Export Results form, if the file extension specified in the File Name field is different from the value in the File Type field, the exported results are saved in the file that is named using the file extension in the File Name field. For example, if the file name is `foo.csv`, but the file type is `.html`, the results are saved in `foo.csv` instead of `foo.html`.

Solution: Ensure that the extension used in the file name is matches the file type.

CCR 1778143: When the corners setup contains a dependent variable, the Plot command plots all points instead of the selected point

Description: When the setup contains dependent variables and those variables are used in the corners, the Plot command plots the results for all the corner points.

Consider an example setup that contains the following variables:

```
val0 = 0
vdd = 1
val1 = vdd
```

and the following corners:

```
Nominal: val0=0, vdd=1, val1=vdd
C0: val0=0, val1=vdd
C1: val0=vdd, val1=0
C2: val0=vdd, val1=vdd
```

In this scenario, when the corner variable contains a dependent variable, the netlist contains the corner parameter in the form of a dependent variable, which is further evaluated by the simulator. However, when you use the Plot command to plot a single result, ViVA XL is unable to identify the individual result value because the corner contains a mix of variables assignments (for example, `val0=0` and `val1=vdd`), and thus plots the results of all points.

Solution: Use the `VAR()` function for dependent variable assignments in the Data View assistant and the Corners Setup form. For example, `val1=VAR("vdd")`.

CCR 2022429: When you import the setup from a maestro view into a new test in ADE Explorer, the corners from ADE Assembler are also added to the setup

Description: When you create a new test in ADE Assembler, and open it in ADE Explorer to import the setup from a maestro cellview, the new test contains the corners imported from the maestro view as well the corners from the setup of ADE Assembler. Ideally, after importing the setup from a maestro cellview, the new test should not contain the extra corners taken from ADE Assembler.

Solution: Open the Corners Setup form in ADE Assembler, and disable the extra corners for the new test.

CCR 1932847: The ‘adexl.gui openSchInWin’ environment variable causing functionality loss

Description: The schematic design opens in a new window when the `adexl.gui openSchInWin` environment variable is set to `t`. This results in functionality loss, for example waveform info-balloons do not work, however, schematic operating point annotations work fine. You will notice missing toolbars that appear when the design is opened in a new tab. In addition, the ADE and schematic interactions are impacted.

Solution: Sets the environment variable to `nil` to open the schematic in a new tab.

```
envSetVal("adexl.gui" "openSchInWin" 'boolean nil)
```

Note: When this variable is set to `t`, the functionality to add operating region expressions gets disabled.

CCR 1698716: The Plot and Save check boxes are disabled when using a config to bind an instance to a DSPF view

Description: If you are working on a design that uses a config to bind an instance to a DSPF view, you cannot toggle the *Plot* and *Save* check boxes in the *Outputs* section of ADE L or the *Outputs Setup* pane of ADE XL, Explorer, and Assembler, because the check boxes are disabled.

Solution: To enable these check boxes, set the `allowInvalidObjectSelection` environment variable, as shown below:

- To set this variable in the `.cdsenv` file, use the call:
  `auCore.selection allowInvalidObjectSelection boolean t`
- To set this variable in the `.cdsinit` file or CIW, use the call:
  `envSetVal("auCore.selection" "allowInvalidObjectSelection" 'boolean t)`

Alternatively, set the following SKILL variable in the `.cdsinit` file, as shown below:
`_allowInvalidObjectSelection = t`

CCR 1696853: Enable or disable status of sweeps and corners are ignored if you edit the swept variable or corner in the setup and update the points table

Description: If you disable any swept point or corner on the Run Preview tab, and then add or edit the swept variable or corner in the setup, all the points are automatically enabled when you update the points table. For example, if you have a swept variable with values as `1:0.1:1.5`. You open Run Preview and disable some sweep points. Next, you edit the sweep values of that variable in the Data View pane value and set its values as `1:0.1:2`. If you update the points table in the Run Preview tab, all points become enabled.

Workaround: Currently, a workaround is not available.

CCR 1625376: Expression Builder is not showing tooltips for all the functions

Description: Some Calculator functions, such as `calcVal`, `delayMeasure`, `analogToDigita`, do not show the latest description in the tooltip. This is because the function catalog file (`.skeMeasuresCatalog`) has been overridden. This file contains a list of functions and a short description about each function. The Expression Builder and Calculator use this file to show function list and function description.

Workaround: You can do one of the following:

- Remove the local `.skeMeasuresCatalog` from `csf` path
- Update the local `.skeMeasuresCatalog` using same file from EAP build as base

CCR 1657981: dyn_floatdcpath violations are not visible in the Checks/ Asserts view

Description: When using `MMISIM 16.1`, the `dyn_floatdcpath` violations do not appear in the Checks/Asserts view. However, these violations are visible in the XML report for the dynamic checks.

Workaround: Add the following in the *Additional Arguments* field on the *Miscellaneous* tab of the Analog Options form:

```
leaki_sim_mode=0
```

CCR 1543451: Parameter table columns are misaligned in ADE Explorer and incremental simulation does not work

Description: When loading a saved ADE L state into ADE Explorer, in the choosing analysis form, the parameter vector table in HB analysis is not aligned correctly.

If the test is created using the ICADV12.2 FCS release, the HB Analysis forms are not correctly formatted due to a function simplifying the white space.

If the test is created using the ICADV12.2 ISR1 release, the analysis forms are well formatted, but incremental simulation fails due to extra blank lines in the state files.

Solution: Currently, a workaround is not available.

CCR 1483770: Simulation results in ADE Explorer are deleted when you try to save them with an existing name

Description: If you save a set of results in ADE Explorer and then save them again using the same name, the results are deleted.

Solution: Do not use an existing name when saving results from ADE Explorer.

CCR 1486703: Specification section of oceanxl script saved from ADE Explorer also includes other tests of the ADE Assembler setup

Description: Open ADE Assembler setup with multiple tests and specs included for outputs of various tests. Then, select a test to open ADE Explorer. In Explorer, if you now save the oceanxl script using the *Session – Save OCEAN XL Script*, it includes the specs of other tests too that are defined in the Assembler setup.

```
;====================== Specs ================================================
ocnxlPutRangeSpec( "tank_tb2" "freq_res" "505M" "495M" )
ocnxlPutRangeSpec( "tank_tb" "freq_res" "505e6" "495e6" )
```

Solution: Currently, a workaround is not available, but the script runs fine.

CCR 1484605: Panic States not available in ADE Explorer

Description: The panic states are not available in ADE Explorer.

Solution: Currently, a workaround is not available.

CCR 1473698: ADE L or XL licenses are not checked in if ADE Explorer or Assembler are launched from ADE L/XL

Description: If you are running ADE L and then open ADE Explorer, the ADE L license should be checked in and Explorer or Assembler should be checked out. If you are running ADE XL and then open Assembler, the ADE XL license should be checked in and Assembler license should be checked out, but it keeps both the licenses checked out.

Solution: Currently, a workaround is not available.

CCR 1470682: ADE Explorer reliability simulation plot all does not show the stress waveform

Description: In ADE Explorer reliability simulation, after simulation complete, the automatically generated waveform does not show the stress waveform, and when you right-click in the *Nominal* column and choose *Plot All*, only aged waveform are displayed and stress waveform is not generated.

Solution: Currently, a workaround is not available.

CCR 1468416: Saved results in ADE Explorer cannot be viewed if test name is changed

Description: If you change the name of a test in either ADE Explorer or ADE Assembler, you cannot view those saved results in ADE Explorer.

Solution: Either view the results in ADE Assembler or change the name of the test back to the previous name when you saved the result.

CCR 1465314: ADE Assembler and Explorer do not follow ADEL_UseNextLicense and ADEXL_UseNextLicense environment variables

Description: ADE L can also use an Explorer or an Assembler if the ADE L license is not available. Similarly, ADE XL can use Assembler if ADE XL license is not available.

If ADE L/XL/GXL are unavailable and ADE Explorer is available, you can set the following environment variable to not use the Explorer and Assembler licenses:

```
envSetVal("license" "ADEL_UseNextLicense" 'string "never")
```

But, still Explorer or Assembler license is checked out.

Similarly for ADE Assembler, when. ADE XL/GXL licenses are unavailable, you can set the following environment variable to not use the ADE Assembler licenses:

```
envSetVal("license" "ADEXL_UseNextLicense" 'string "never")
```

But, ADE Assembler license is checked out.

Similarly, when `ADEL_UseNextLicense` or `ADEXL_UseNextLicense` is set to "`prompt`". In this case if ADE L/XL/GXL is unavailable, it does not prompt to use Assembler/Explorer license, instead launches ADE L/XL using Assembler or Explorer license.

Solution: Currently, a workaround is not available.

CCR 1462061: session reset option is not available in Explorer

Description: The reset menu entry is not available in ADE Explorer.

Solution: Currently, a workaround is not available.

CCR 1456559: Cannot load results if the last simulation run was done in ADE Explorer read-only view

Description: Open ADE Explorer in read-only mode. Run a simulation and the close the session. Now, if you open the same Explorer view again and load the results, it does not do anything.

Solution: Currently, a workaround is not available.

CCR 1453924: No way to allow expression re-evaluation in single-point simulation run

Description: If you run a single-point simulation and then add a new expression later or modify the specs, there is no option available to re-evaluate the expression.

Solution: Currently, a workaround is not available.

CCR 1446762: Better info and error messages required for AMS run

Description: In ADE Explorer, if you click the Netlist and Run button, no messages are printed in CIW about the successful run.

Solution: You need to read the log file to get the information.

CCR 1428903: Cannot run Monte Carlo with sweeps in ADE Explorer

Description: If you try to run Monte Carlo with a sweep, the following pop-up is displayed:

```
Disable all sweeps or enable "Use Reference Point" on the monte carlo
options form.
```

Now, if you select the *Use Reference Point* option and run again, another pop-up appears:

```
a) if using reference point, choose Run->Edit Reference Point

b) if using starting setup state, edit the specified starting setup
state.
```

Neither of these options exist in ADE Explorer.

Solution: Disable the sweep and run the simulation.

CCR 1418471: Cannot load in parametric sweep files from ADE L to ADE Explorer

Description: You cannot load a parametric file in ADE Explorer. In ADE L, you can do this in the parametric analysis tool.

Solution: You need to add each variable to the Setup assistant and setup each sweep for each variable.

CCR 1415015: Provide ability to disable sweeps in ADE Explorer

Description: There is no way to disable all sweeps in ADE Explorer like it can be done using the Run Summary in ADE Assembler

Solution: Currently, a workaround is not available.

CCR 1418471: Cannot load in parametric sweep files from ADE L to ADE Explorer

Description: You cannot load a parametric file in ADE Explorer. In ADE L, you can do this in the parametric analysis tool.

Solution: You need to add each variable to the Setup assistant and setup each sweep for each variable.

Known Problems and Solutions in Virtuoso ADE Verifier

This document describes the known issues with Virtuoso® ADE Verifier and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Overview of Virtuoso ADE Verifier

CCR 2551598: Setup Library cellviews created in the current version of Virtuoso are not compatible with earlier Virtuoso releases.

Description: When you create a Setup Library cellview, `setupLib`, in Virtuoso Studio IC23.1, and open it in the Setup Library assistant from an older Virtuoso release, the cellview contents are available, but the features cannot be used. Making any changes in the opened cellview view can cause Virtuoso to exit unexpectedly.

Solution: Currently, a workaround is not available.

CCR 2551597: Sync with Reference Setup does not work as expected for parametric sets in the Setup Library assistant.

Description: Consider that you create a parametric set from in the Setup Library assistant, and the drag the sweep setup that contains this parametric set to the Global Variables section of the Data View Assistant in Virtuoso ADE Assembler. This will create a reference in the Data View assistant to all the variables in the parametric set. If you modify a variable from this set in either of the assistants, the Data View assistant displays the variable name in bold. When you right-click this highlighted variable and choose *Sync with Reference Setup*, the variable does not update with the value in the Setup Library assistant.

Solution: Currently, a workaround is not available.

CCR 2134047: Verifier ignores verification spaces when corners or global variables are deselected in ADE Assembler and run mode is 'with SPACE'.

Description: Coverage in Verifier might show less than 100% after a simulation run when you select *Local with SPACE* or *Batch with SPACE* as the run mode. This is not correct.

When you specify a verification space for an implementation, corners or variables are set up accordingly in Verifier. However, Verifier does not check if the corners or variables are globally disabled in the corresponding `maestro` cellview. If the corners or variables are globally disabled, the simulation run in Verifier run does not create the expected coverage.

Solution: Check whether corners and global variables are enabled globally in ADE Assembler when applying verification spaces. To disable corners or global variables, use the individual check boxes in the *Data View Assistant* instead of the global check box for *Corners* and *Variables*.

CCR 1726252: Variable type tolerance and range type specs are missing in Verifier

Description: If the ADE Assembler output has a tolerance specification that has expression tolerance, expression target, or both as mentioned below:

```
tol (VAR("spec_value1") * 50) calcVal("constant")
```

Verifier cannot parse the correct minimum and maximum specification when creating requirement for this kind of output and the *MinSpec* and *MaxSpec* values are left blank.

Solution: There is no workaround available.

CCR 1463089: Assertion data is not visible for Monte Carlo simulations

Description: Currently, assertion checking for Monte Carlo implementations is not supported in Verifier.

Solution: There is no workaround available.

CCR 1509719: Implementation cellviews, whose AMS simulations are set to run in interactive mode, launch SimVision

Description: When you run an implementation cellview from Verifier, whose AMS simulation is set to start in interactive mode, the Explorer or Assembler graphical user interface does not display. However, the SimVision graphical user interface displays.

Solution: Disable the interactive mode of the AMS simulation before you run the implementation cellview from Verifier.

Known Problems and Solutions in Virtuoso Automated Placement and Routing Solution

This document describes the known issues with Virtuoso® Studio Automated Placement and Routing Solution and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Topics

Virtuoso Automated Device Placement and Routing

Virtuoso Automated Standard Cell Placement and Routing

Chip Assembly Routing

CCR 2807265: Cross via QoR in customer data

Description: There is an issue with stacks of three or more vias not being properly detected or fixed.

Solution: This is a rare occurrence and will need to be manually fixed if the user needs these stacks aligned.

CCR 2789954: Which router is advised for mix mode designs

Description: Currently, none or the routers support mixed designs, which contain a mix of devices, standard cells, and small blocks in the same hierarchy.

Solution: Separate the standard cells and devices into different designs.

CCR 2532962: validLayers violations are not reported

Description: Some constraints such as `validRoutingLayers` are incorrectly applied to the design default constraint group instead of the `UserDefinedDesignDefault` constraint group. This problem can occur with any design style through the use of the *Setup* tab in the Routing assistant with user modifications to the Layer Settings for Bottom and Top Layer. Also, when the user saves the design, these constraints are captured in the disk data for that design and becomes persistent. These existing constraints can override values intended for the foundry constraint group while running the DR checking.

Workaround: If anomalies are found while running verification, use Constraint Manager to delete the constraints from the design default constraint group. The solution is being developed and will be available in a future release.

Known Problems and Solutions in Virtuoso Studio Design Environment

This document describes the known issues with Virtuoso Studio Design Environment and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Virtuoso Studio Design Environment

Virtuoso Studio Design Environment What's New

CCR 2809626: Displaying non-modal dbox "deDBox", title "Problem with cds.lib file" when creating a new cellview

Description: An error message related to `cds.lib` occurs when the user creates a new cellview and `ddUpdateLibList()` was cancelled by the `geiUpateLibListTrigger()` trigger function.

Solution: Close the cellviews specified by the `geDBox` using *File - Close Data*. Also, do not delete libraries from `cds.lib` when there are open cellviews from the specified libraries.

CCR 2725740: dbus processes keep running after the Innovus job is done with version 22.10

Description: The dbus-launch and dbus-daemon processes keep running after closing the program and need to be ended manually.

Solution: This issue occurs when the DBUS_SESSION_BUS_ADDRESS or CDS_QT_ENABLE_DBUS_AUTOLAUNCH is enabled in the environment. To resolve the issue, disable these shell environment variables.

Problem: Virtuoso exits abruptly when it is launched in the foreground and the terminal closes.

Description: When Virtuoso is launched from a terminal in the foreground, and the terminal is closed, the Virtuoso session exits without warning and the output of commands that write to STDOUT or STDERR in the parent shell are lost.

Solution: Ensure that you launch the Virtuoso session in the background. Additionally, ensure that outputs are directed to log files, which you can access later. Alternatively, you can use the Linux `screen` or `bg` commands. For information on how to start Virtuoso, see the Starting Cadence Software section of the *Virtuoso Design Environment User Guide*.

CCR 1420213: Detect and warn when XRender is missing

Description: The RENDER extension on the X server is required for proper font rendering and drawing operations. If the current display does not have the RENDER extension, this will result in poor font appearance and sub-optimal drawing performance.

Solution: You can use the following command to check if a particular display has the RENDER extension:

```
xdpyinfo [-d <DISPLAY>] | grep RENDER
```

If RENDER is present, this should print `RENDER`; if it is absent, nothing will be printed.

To resolve this situation, enable the RENDER extension in the X server configuration and restart the X server.

For more information, see the Setting Fonts using the .cdsinit File section of the *Virtuoso Design Environment User Guide*.

CCR 2259415: Graphic area of session window is destroyed under EoD

Description: Graphic and session window gets damaged in the EoD session.

Solution: This issue occurs when you use the old version of EoD. You need to use the latest version of the EoD server.

CCR 1059421: Viewfile window does not show the end of the file

Description: In a viewfile window, the vertical scrollbar does not reach the bottom of the window, which gives an appearance that more information is expected.

Solution: There is no workaround available.

CCR 944227: Bitmap and JPEG format limitation

Description: Bitmap and JPEG formats are not allowed to set a transparent background. Therefore, you cannot perform operations like exporting an image from VSE with a transparent background in the bitmap or JPEG format.

Workaround: For a transparent background. use the PNG format instead of bitmap and JPEG.

CCR 874983: Images from Copy to Clipboard, Through Export Image, Cannot Be Pasted in OpenOffice

Description: The *File – Export Image* option in VSE and VLS can successfully copy an image to X window's clipboard, but that image cannot then be reliably pasted into all versions of OpenOffice.

Solution: This issue impacts a range of applications, not just Virtuoso. We recommend that you use a later version of OpenOffice. For example, OpenOffice 3.2.0 has been tested to work as expected.

CCR 874983: Images from Copy to Clipboard, Through Export Image, Cannot Be Pasted in OpenOffice

Description: The *File – Export Image* option in VSE and VLS can successfully copy an image to X window's clipboard, but that image cannot then be reliably pasted into all versions of OpenOffice.

Solution: This issue impacts a range of applications, not just Virtuoso. We recommend that you use a later version of OpenOffice. For example, OpenOffice 3.2.0 has been tested to work as expected.

CCR 856322: CDS_LOAD_ENV and csfLookupConfig interaction results in confusing behavior

Description: Setting both `CDS_LOAD_ENV` and `csfLookupConfig` (to load `.cdsenv`) are incompatible and may result in an unintended load order for `.cdsenv` files.

Solution: Specify either `CDS_LOAD_ENV` or `csfLookupConfig` to customize `.cdsenv` loading order.

CCR 840571: Session management can cause bad exits from Virtuoso

Description: Use of session management can result in forced exits from Virtuoso that result in crashes.

Solution: Unset the environment variable `SESSION_MANAGER`:

```
csh: unsetenv SESSION_MANAGER
sh: SESSION_MANAGER=
```

CCR 608762: Torn off or floating pop-up menus are tied to the CIW for visibility and will not show up if CIW is iconified

Description: In any Virtuoso graphics editor, when you right-mouse-button (RMB) click the design canvas, you will get a context sensitive pop-up menu. This pop-up menu can be made to float, also known as “torn off”, by clicking on the "----" line at the top of the menu.

You may assume that the visibility of a floating pop-up menu will follow that of the graphics editor. In other words, if you iconify the graphics editor, the pop-up menu should be iconified along with that editor. This is not however the case, rather, the visibility of the floating pop-up menu is tied to the visibility of the CIW. The worst case scenario would occur if you have the CIW iconified. A pop-up menu made to float when CIW is iconified will not appear, and you may be confused about its correct operation.

Solution: True to ensure that the CIW is not iconified.

CCR 368282: libelf warning messages on SUSE results

Description: IC61X fails to start on a SuSE 9 based Linux OS due to different `libelf` shared library versions.

The `libelf` shared library versions on SLES 9 are different than on RHEL 2.1/3.0, and the `libXm` shared library versions on SLES 9 are different than on RHEL 2.1. An error message similar to the following appears in this situation.

```
*Error_Message:

      install path/tools.lnx86/dfII/bin/32bit/icfb.exe:
      error while loading shared libraries:libelf.so.1:
      cannot open shared object file:
      No such file or directory
```

Solution: To correct this situation, do the following.

1. Ensure the following are installed on your machine.

   `libelf.so.0` in `/usr/lib64`

   `libXm.so.3` in `/usr/X11R6/lib`

2. Check that the following links (created at installation) are present on your machine.

   install path`/tools.lnx86/lib/64bit/SuSE/libelf.so.1`
   links to
   `"/usr/lib64/libelf.so.0"`

   install path`/tools.lnx86/lib/SuSE/libXm.so.2`
   links to
   `"/usr/X11R6/lib/libXm.so.3"`

3. Add the following information to the `LD_LIBRARY_PATH` environment variable.

   install path`/tools.lnx86/lib/64bit/SuSE`
   (for running 64bit executables)

install path`/tools.lnx86/lib/SuSE`
(for running 32bit executables)

CCR 98747: Should XScale magnify irregular view names

Description: XScale and other DFII applications do not understand file system mapped directory names. Mapped library, view, and cell directory names can contain hexadecimal numbers, used to represent the special characters in the user name. For example, the original user-supplied name "`layout.placed`" is understood correctly, but the mapped file system equivalent name "`layout#2eplaced`" is not understood.

Solution: For library, cell, and view directory names, you need to enter the original user-supplied name, not the mapped file system name. This applies to graphical user interface forms, the command line in the Command Interpreter Window, and code.

CCR 98568: OA: XScale can not magnify multipart paths

Description: XScale does not work well for multipart paths (MPPs). Currently, XScale successfully scales the master path, but not subparts.

Solution: Do not use XScale on MPPs that have subparts. If you must change the size of an MPP that has subparts, you can use the `rodUnNameShape` function to convert the MPP into regular shapes, and then use XScale on the resulting shapes. However, you cannot convert the shapes back into an MPP.

1. Restart the IC61X executable. The software starts and a warning appears stating the version information is not available. This warning can be ignored – the software is now working as intended.

```
install path/lnx86/pink/tools/dfII/bin/32bit/icfb.exe:
install path/lnx86/pink/tools/lib/SuSE/libelf.so.1:
no version information available
(required by install path/lnx86/pink/tools/lib/libvirtuos_sh.so)
```

CCR 2777422: Health Monitor is listing unknown processes

Description: In the Health Monitor form, the *Process ID* field is also listing non-Virtuoso processes.

Solution: The Health Monitor form lists all the child processes of Virtuoso because they might affect the performance of Virtuoso.

For example, the libSelect process is the Library Selector (also known as the Library Browser) appears in the Process ID listing when you click *Browser* on the Virtuoso form. This application is different from the Library Manager (libManager) process.

To check all the active child processes,

➡ Run the `pstree -ahlp <virtuoso process ID>` command on Linux . To know your process ID, type `ipcGetPid()` in CIW.

All the child processes is listed as an output.

```
virtuoso,390698

  ├─cdsServIpc,391154 -c 38235 -n 4 -r 10 -k -x /lan/ssv/pegasus/
combined_flow_stamped/lnx86/2220/gcc_v9/latest/tools.lnx86/Pegasus/bin/
pegasusgui -options /tmp/ipvs.init.options.girrajk.390698.1

  │   └─pegasusgui,391155 -options /tmp/ipvs.init.options.girrajk.390698.1

  │       ├─{pegasusgui},391266

  │       …

  │       └─{pegasusgui},391399

  ├─cdsServIpc,391325 -c 38235 -n 5 -r 10 -x /grid/cic/ICADVM20.1/main/lnx86/
64/buildComplete/tools/dfII/bin/libManager -unmapped  -mpssession
virtuoso390698 -mpshost tfodcl20  -ADVM

  │   └─libManager,391327 -unmapped -mpssession virtuoso390698 -mpshost
tfodcl20 -ADVM

  │       ├─{libManager},391457

  │       …

  │       └─{libManager},391509

  ├─cdsServIpc,391326 -c 38235 -n 6 -r 10 -x /grid/cic/ICADVM20.1/main/lnx86/
64/buildComplete/tools/dfII/bin/libSelect -unmapped  -mpssession
virtuoso390698 -mpshost tfodcl20

  │   └─libSelect,391328 -unmapped -mpssession virtuoso390698 -mpshost tfodcl20

  │       ├─{libSelect},391456

  │       …
```

```
  |       └{libSelect},391508

  ├cdsServIpc,391453 -c 38235 -n 8 -r 15 -k -x … /grid/cic/ICADVM20.1/main/
lnx86/64/buildComplete/bin/runUserCmd

  |   └perfUtilExtCtrl,391454 390698 /dev/shm/perfUtil_girrajk_390698_0 /dpc/
intel_DA9109_1/P1278_CPG/EC6108/girrajk/EAD/SDR/.cadence/perf pstack strace 1
/dpc/intel_DA9109_1/P1278_CPG/EC6108/girrajk/EAD/SDR/.cadence/perf/
tfodcl20_girrajk_390698 150 gdb 0 1 1 1 /grid/cic/ICADVM20.1/main/lnx86/64/
builds/230308-948/tools.lnx86/dfII/samples/perfUtil/data/linux-x86-64 1 180 0
/home/girrajk/CDS.log 0 top girrajk 1 1 250 nil 0 nil nil 0 /tmp/perfMop_j90698
/grid/cic/ICADVM20.1/main/lnx86/64/buildComplete/bin/runUserCmd

  |       ├{perfUtilExtCtrl},391455

  |       …

  |       └{perfUtilExtCtrl},391473
```

Additional information:

- If a process ID does not match the `pstree` output, click the *Refresh* button to update the list.
- If the process to check is not a child process of Virtuoso, you can enter the process ID directly.

CCR 2828597: Issue when following subsequent links between manuals in Doc Assistant's Offline mode

Description: If you follow a cross-reference link from one product manual to another, in Doc Assistant's Offline mode, subsequent link selections within that manual will not work.

Workaround: If possible, switch to Online mode. Otherwise, if restricted to Offline mode, click Back on the page where links are not working to open the previous page. Click the product manual name in the breadcrumbs and select the required topic from the Topic of Contents.

General Issues

Problem: BadWindow (invalid Window parameter) error

Description: If you start different desktops from the same display (whether it is a local or remote display, or a VNC server), you might see the following error:

```
Qt Warning: X Error: BadWindow (invalid Window parameter) 3
Major opcode: 20 (X_GetProperty)
Resource id:  0x2000090
```

For example, if you start KDE on your VNC server, exit it, then start CDE or GNOME on the same VNC server, you might see this error.

Solution: Exit the X server or kill the VNC server and reconfigure it to use a different desktop before or during login or before starting another VNC server.

In general, if you want to change your desktop and ensure that Virtuoso applications correctly identify it, log out of your X session (if you are using VNC, kill the VNC server), and modify the startup script and properties to ensure that the correct desktop is run when the server is started. If you start the server with one desktop/window manager, then switch to another without exiting the X server or killing and restarting the VNC server, some properties for the first desktop/window manager may still remain attached to the display and cause errors in the Qt initialization logic that checks to see which window manager is running.

Known Problems and Solutions in Virtuoso Studio Design Environment SKILL

This document describes the known issues with Virtuoso® Studio Design Environment SKILL and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Graphics Editor Functions

Database Access Functions

Design Management Functions

Cadence SKILL Language

CCR 689416: Load .cdsinit goes into infinite loop

Description: Using `hi` toplevel forms in `load()` or `loadi()` starts a recursive evaluator in the load sequence.

Solution: Do not use `hi` functions that open new toplevels inside `.il` files called by `load()` or `loadi()`.

CCR 522913: deOpen SKILL function does not work correctly with config view

Description: The SKILL function `deOpen` allows opening of a config view. Currently, if you open a sub-cell in context of this configuration, it does not acquire the binding setup of the config view.

Solution: To acquire the correct configuration binding, you can open a sub-cell through the *Cadence Hierarchy Editor*, or directly from a schematic (in context of the configuration) using *Virtuoso Schematic Editor*.

Known Problems and Solutions in Virtuoso Design Rule Driven Editing

This document describes the known issues with Virtuoso® Design Rule Driven Editing and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Design Rule Driven Editing

CCR 2215672: DRD Post Edit stops checking when the number of shapes exceeds the maximum allowed

In the Post-Edit mode, when the number of shapes exceeds the maximum allowed, which is 4000, DRD stops checking and displays a message. The limit cannot be changed.

CCR 1235370: Markers created on top are not in sync with the EIP cell if you return without saving

Description: When you use *Edit In Place* to descend into a cell, *Post-Edit* mode in DRD and Virtuoso IPVS checks the area of change around the instance of the cell and places any violation markers in the parent cell even if the violation is entirely between shapes in the lower level cell. If you return to the parent cell but discard the changes you made while editing in place, the violation markers shown for the parent cell may no longer be correct.

Solution: To display the correct violations, re-check the area around the instance you descended into. If you are editing in place in order to explore what-if scenarios with no intention of keeping the changes, turn off *Post-Edit* mode to prevent the ghost markers that remain after you discard the edits.

CCR 1175254: DRD reports false centerToCenter minCutClassSpacing violations in some cases

Description: DRD reports false `centerToCenter minCutClassSpacing` violations when the technology file contains two `minCutClassSpacing` constraints with the following characteristics:

- One `minCutClassSpacing` constraint is defined for one parallel run length (PRL) range (Example: `'paraOverlap` parameter is set to `-1`, which is for cuts that have a negative PRL) and is specified with the `centerToCenter` parameter.
- Another `minCutClassSpacing` constraint is defined for another PRL range (Example: `'paraOverlap` parameter set to `0.001`, which is for cuts that have a positive PRL), but is not specified with the `centerToCenter` parameter.

This is because DRD interprets both `minCutClassSpacing` constraints as `centerToCenter` and checks for `centerToCenter` even when the PRL between the two shapes in the layout is positive, causing false `minCutClassSpacing` violations to be reported in the layout.

Solution: Do the following in your technology file:

1. For the same cut layer, add a third `minCutClassSpacing` constraint with the `'paraOverlap` parameter set to `0`. The other parameters of this newly added `minCutClassSpacing` constraint can be the same as those of the `minCutClassSpacing` constraint with `'paraOverlap` = `-1`.
2. Load and save your technology file.

CCR 1122625: DRD Post-Edit does not report violations for allowedSpacingRanges constraints defined in an AND constraint group

Description: In the technology file, if you define `allowedSpacingRanges` constraints for a layer in an AND constraint group, DRD Post-Edit does not report violations related to these constraints.

Solution: To enable DRD Post-Edit to report `allowedSpacingRanges` violations if they occur, do one of the following:

- Enable DRD Notify while editing the layout, or run DRD in Batch-Check mode (choose *Verify – Design*) on the layout after completing your edits.

- In the technology file, add a `minSpacing` constraint definition for the same layer for which you have defined the `allowedSpacingRanges` constraints. The definitions are shown in the following example:

```
( "myConstraintGroup"      nil    nil    'and
    spacings(
        ( allowedSpacingRanges "Active" 'vertical 'stepSize 0.02 (">= 0.04"))
        ( allowedSpacingRanges "Active" 'horizontal (">= 0.08") )
        ( minSpacing "Active" 0.04 )
    ) ;spacings
) ;myConstraintGroup
```

CCR 1045350: DRD does not report the required overlap value for minDirectionalOverlap violations

Description: The `minDirectionalOverlap` constraint specifies the required overlap in the given direction between two shapes on two different layers. DRD Post-Edit and DRD Batch-Check (choose *Verify – Design* in Virtuoso) modes report violations created by this constraint in the form of violation markers in the layout canvas, with corresponding entries in the Annotation Browser. However, the text descriptions of these markers do not report the required overlap value (a value of 0 is reported instead of the expected overlap value).

Solution: As a workaround, to determine the value of the required directional overlap between two overlapping shapes on two different layers, refer either to the foundry's Design Rule Manual (DRM) or to the Virtuoso Technology File associated with the process. You can use the value thus obtained to fix DRC violations by adjusting the overlap between shapes at locations where DRD reports `minDirectionalOverlap` violations.

CCR 1043998: DRD Post-Edit is not flagging allowedSpacingRanges violation in the hierarchical view

Description: DRD Notify and DRD Post-Edit do not flag `allowedSpacingRanges` constraint violations until you run the *Verify – Design* command on the shapes that are placed close to each other.

Solution: As a workaround, run the *Verify – Design* command after you edit a shape.

CCR 1043375: DRD inspection area is not covering the minClusterDistance

Description: DRD Notify or DRD Post-Edit do not flag `minClusterDistance` constraint violations until you run the *Verify – Design* command on the shapes that are placed close to each other.

Solution: As a workaround, run the *Verify – Design* command after editing a shape.

CCR 1042866: DRD Notify does not report minNeighboringShapesSpacing violations

Description: DRD supports the `minNeighboringShapesSpacing` constraint in Post-Edit and Batch-Check modes. However, it does not support this constraint in the Notify mode.

Solution: Before editing your layout, select *Post-Edit* and the corresponding *Spacing* check box in the DRD Options form. Alternatively, in the Batch Checker form (choose *Verify – Design* to open this form), select the *Spacing* check box in the *Virtuoso DRD* group box and run DRD in Batch-Check mode on your design.

CCR 1042403: DRD postedit is flagging minSideSpacing violation for the rectilinear shapes

Description: DRD post-edit is flagging `minSideSpacing` constraint violations for shapes that are not rectangular if the `shortSideToLongside` parameter is specified.

Solution: There is no workaround available.

CCR 1038130: DRD Notify and Post-Edit not reporting minSideSpacing violations

Description: The `minSideSpacing` constraint specifies the required minimum clearance (spacing) between two shapes on two different layers for the specified edge (short or long edge). Currently, Notify and Post-Edit modes do not report violations of this constraint. However, you can run DRD in Batch-Check (choose *Verify – Design*) mode to locate these violations.

Solution: As a workaround, you can run DRD in Batch-Check mode to locate `minSideSpacing` constraint violations.

CCR 1032580: Issues with DRD post-edit check for minNeighboringShapesClearance rule

Description: The `minNeighboringShapesSpacing` constraint specifies the minimum spacing between two shapes on the specified layer when neighboring shapes are present on another layer. Currently, the DRD Notify and DRD Post-Edit modes are not reporting violations of this constraint in the layout. However, you can run DRD in Batch-Check (choose *Verify – Design*) mode to locate these violations.

Solution: As a workaround, you can run DRD in Batch-Check mode to locate `minNeighboringShapesSpacing` constraint violations.

CCR 1009703: DRD does not report violations between shapes drawn on user-defined purposes
CCR 1000182: Checker does not check the spacing between Poly:drawing and Poly:dummy

Description: DRD does not report violations between shapes drawn on layers with user-defined purposes. The following example from a technology file illustrates this limitation:

```
techPurposes(
    ;( PurposeName Purpose#   Abbreviation   [Attributes] )
    ;User-Defined Purposes
    ( purp1         13          pp1 )
);

techLayerPurposePriorities(
    ;( LayerName Purpose )
    ( Metal1 purp1 )
);

( minSpacing "Metal1" 0.05 'ref "M1.SP.1" 'description "Minimum M1 Space" )
```

where, `purp1` is the user-defined purpose and `Metal1:purp1` is the LPP. A minimum spacing constraint is defined for `Metal1`.

In the layout, if you move the shapes on LPPs `Metal1:drawing` and `Metal1:purp1` closer to each other, DRD reports `minSpacing` violations between the `Metal1:drawing` shapes, but not between the `Metal1:purp1` shapes.

Solution: As a workaround, add an attribute in the technology file definition of the user-defined purpose.

In the following example, the attribute is `'sigType`:

```
techPurposes(
    ;User-Defined Purposes
    ;( PurposeName Purpose#   Abbreviation   [Attributes] )
    ( purp1         13          pp1         'sigType "signal" )
);
```

CCR 985745: Saving a technology file for the first time while Post-Edit mode and extractor are ON in Virtuoso XL might cause post-edit checking

Description: Saving a technology file for the first time in a Virtuoso session while Post-Edit mode and the extractor are enabled can cause Post-Edit checks to be performed on the areas where the extractor made reassignments. If your design contains markers, such as from a batch check run earlier, these markers may get replaced with the results of the new Post-Edit check.

Solution: Run DRD in Batch-Check mode again if any markers are overwritten.

CCR 843336: DRD performance problem — many violations, no Ctrl+c

Description: DRD is an interactive feature. Depending on the type of editing operation being performed and the number of layout objects being affected by the editing operation, DRD editing checks can become runtime or memory intensive, or both.

Solution: The default settings and behaviors listed below are provided to help increase performance:

DRD Mode	Ctrl+c Capability	Maximum Shape Limit in DRD database	Continue performing DRD checking when Maximum Shape Limit is reached
Notify and Enforce	No	1000 (dynamic) 5000 (static)	No
Post-Edit	Yes (except in a replay file)	100,000	No
Batch-Check (Verify Design)	Yes (except in a replay file)	100,000	Yes

Using Ctrl+c Capability

In DRD Post-Edit and Batch-Check modes, press `Ctrl+c` to interrupt operation. Any markers found up to that point are generated. The following message is displayed in the CIW and log file:

```
*WARNING* Checking was interrupted; results are incomplete.
```

Note: By default, `Ctrl+c` is not recognized in a replay file. To use `Ctrl+c` in a replay file, set the `drd_ctrlc_during_replay` shell environment variable. It can be set before or during a Virtuoso session.

Specifying the Shape Limit

For Enforce and Notify modes, use the following shell environment variables to redefine the default threshold:

- `DRD_MAX_DYN_SHAPES`: If not defined, the default is 1000 dynamic shapes.
- `DRD_MAX_STATIC_SHAPES`: If not defined, the default is 5000 static shapes.

For Post-Edit and Batch-Check modes, use the following shell environment variables to redefine the default threshold:

- `DRD_MAX_POSTEDIT_SHAPES`: If not defined, the default is 100,000 dynamic shapes.

Note: These environment variables are set during Virtuoso session.

Ignoring the Shape Limit

In Enforce, Notify, and Post-Edit modes, DRD stops performing a check if the limit set on the maximum number of shapes is reached. In Batch-Check mode, DRD continues to perform a check even after the limit is reached and the following message is displayed:

```
*WARNING* (LE-103865): For Verify Design checking, the maximum number
of shapes specified (100000) for creating the DRD database has been
reached.
```

If DRD continues to perform the check, the following message is displayed:

```
*WARNING* (LE-103874): DRD checking results may not be complete or
correct as performed on the incomplete DRD database.
```

If DRD stops performing the check, the following message is displayed:

```
*WARNING* (LE-103875): No DRD checking is performed on the incomplete
DRD database.
```

To force checking to continue even after the limit is reached:

- In Enforce and Notify modes, set the `drd_check_incomplete` shell environment variable.
- In Post-edit mode, set `drd_check_incomplete_postedit` the shell environment variable.

The shell environment variables can be set, modified, or unset before or during a Virtuoso session.

Note: You can use the `setShellEnvVar("envVarName=envVarValue")` and `unsetShellEnvVar("envVarName")` commands in the CIW to set and unset the shell environment variable, respectively.

CCR 221291, 140821: Stretch using stretch handle causes DRD feedback before violation occurs

Description: If you are stretching MOS devices with the stretch handles on the side opposite to the object that is in violation, warnings are displayed before the violation threshold is reached.

Solution: Using the stretch handles on the opposite side of MOS devices is not recommended when DRD is enabled.

CCR 155184: DRD enforce mode allows shorts to create with maintain connection on

Description: Pins are prevented from creating shorts when they are moved to overlap pins on different nets, but the path that is moved along with the pin is not prevented from creating a short if it overlaps a pin or path on a different net.

Solution: Do not use Enforce mode with maintain connections on.

CCR 102889: DRD has problems with commands starting in one window and ending the commands in another window

Description: When you copy objects across cellviews, post-edit markers flag violations, but other DRD editing graphics, such as arrows and halos, do not display.

Solution: Set post-edit markers to on to flag violations.

Known Problems and Solutions in Virtuoso Electrically Aware Design Flow

This document describes the known issues with the Virtuoso® Electrically Aware Design Flow and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to Virtuoso Electrically Aware Design Flow

Parasitic Aware Design in Virtuoso ADE Explorer, Virtuoso ADE Assembler, and Virtuoso Schematics L/XL

CCR 1813295: Missing or incomplete layer mappings in the process setup (.ini) file affects high precision C extraction

Description: When a conformal dielectric layer is expanded from `sub_conductor` layers that are either unmapped or mapped incorrectly in EAD process.ini file, Layout EAD fails to complete high precision C extraction, and Virtuoso may stop responding or close abruptly.

Solution: Ensure that all `conductor` and `sub_conductor` layers have correct layer mappings.

CCR 1631710: The *Automatically zoom display to selected network or parasitic* command in the EAD Browser of VSE XL does not work properly in all cases

Description: When using EAD Browser in VSE XL, the *Automatically zoom display to selected network or parasitic* command on the toolbar does not work properly for all the nets. At times, it zooms to only a part of the selected net or none of the net.

Solution: Currently, a workaround is not available. You can manually zoom to a net or use the *Zoom to Selected* command from the *View* menu of the Virtuoso Schematic Editor XL window.

CCR 1631699: When you display capacitance annotations for the top level schematic in Virtuoso Schematic Editor XL, EAD fails to determine capacitance for internal nets

Description: When you use the *Annotate All Nets Total Cap* command on the EAD Browser assistant in VSE XL to display annotations for top level schematic, EAD fails to determine capacitance for internal nets, and displays warning messages in CIW.

Solution: Currently, a workaround is not available. You can view the total capacitance value for a net in the summary table of the EAD Browser.

CCR 1557857: The commands on the EAD toolbar are active even if the setup is not loaded in the EAD Browser

Description: The commands on the EAD toolbar are active even if the setup is not loaded in the EAD Browser. However, if used, they do not have any effect.

Solution: Currently, a workaround is not available.

CCR 1409768: If the -lsf_number option is not specified with -lsf_command, the eadModelGen command is run on the local computer

Description: If you specify the `-lsf_command` option with the `eadModelGen` command to run the model generation jobs using LSF, but do not specify the `-lsf_number` option, the jobs are not sent to the LSF resource. Instead, they are run on the local computer.

Solution: Ensure that when the `-lsf_command` option is used, you also set the `-lsf_number` option to a value greater than `1`.

CCR 1123547: Resistance and Current Limit are displayed on dynamic display but not on info balloon

Description: During wire editing, a virtual node is created at the end of the newly-created wire in order to force the extractor to add a resistor that represents the wire while it is being drawn. After the wire has been created, if the wire is dangling, the virtual node is deleted, which means that no resistor is extracted for it, and no resistance or current limits are computed for it.

Solution: *For information only.*

CCR 1107646: Extracting shapes with diagonal edges at angles other than 45 degrees results in lower resistance than expected

Description: If your layout design contains shapes with diagonal edges at any angle other than 45 degrees, Layout EAD might approximate the width incorrectly resulting in lower resistance values than expected.

Solution: Use only shapes with orthogonal or 45-degree diagonal edges when designing the physical layout.

CCR 1106601: Resimulation fails when flattening the schematic that contains mutual inductors

Description: In the resimulation flow, the DUT hierarchy is flattened to stitch the layout parasitics correctly. If the DUT contains mutual inductors, these are also flattened, but the parameters pointing to the inductors are not updated. This leads to a simulation failure because the inductors cannot be found.

Solution: For the components that contain inductors and mutual inductors, copy the symbol view to a view named `noflatten` and set the following two variables in the `.cdsenv` file:

```
envSetVal("msps" "switchViewList" 'string "noflatten spectre schematic  veriloga")
envSetVal("msps" "stopViewList" 'string "noflatten spectre veriloga")
```

These variable settings inhibit the flattening of these components and the simulation runs successfully.

CCR 1105082: Point-To-Point Info Balloon is disabled when EM checking is disabled

Description: The Point-To-Point Info Balloon feature lets you visualize parasitic effects and EM violations between two user-specified points on a net, however, this feature is disabled when EM checking is disabled.

Solution: Enable EM checking from the EAD Browser assistant to use the Point-To-Point Info Balloon feature.

CCR 1072869: EAD crashes when loading a dataset stored in "psf with floats" format

Description: If you try to load a dataset where the currents were stored in "psf with floats" format, VLS EAD crashes.

Solution: Do not use "psf with floats" format in your datasets. Use the "psf" and "psfxl" formats, which are more suited for this purpose.

CCR 1039804: The EAD Browser's "Select Partial Network" feature does not support standardVias

Description: When using *Select Partial Network* mode, selecting standardVias in the canvas does not display information for the vias in the EAD Browser.

Solution: Select the wires around the vias to see information about the vias in the EAD Browser.

CCR 922947: The Point list in the EAD Results view does not contain the `ALL` option

Description: The Point list in the EAD Results View does not contain the ALL option. As a result, you cannot create a dataset that includes the results for all the data points.

Solution: Create a separate dataset for each simulation point.

Known Problems and Solutions in Virtuoso Fluid Guard Ring

This document describes the known issues with Virtuoso® Fluid Guard Ring (FGR) in Virtuoso® Layout Suite L and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

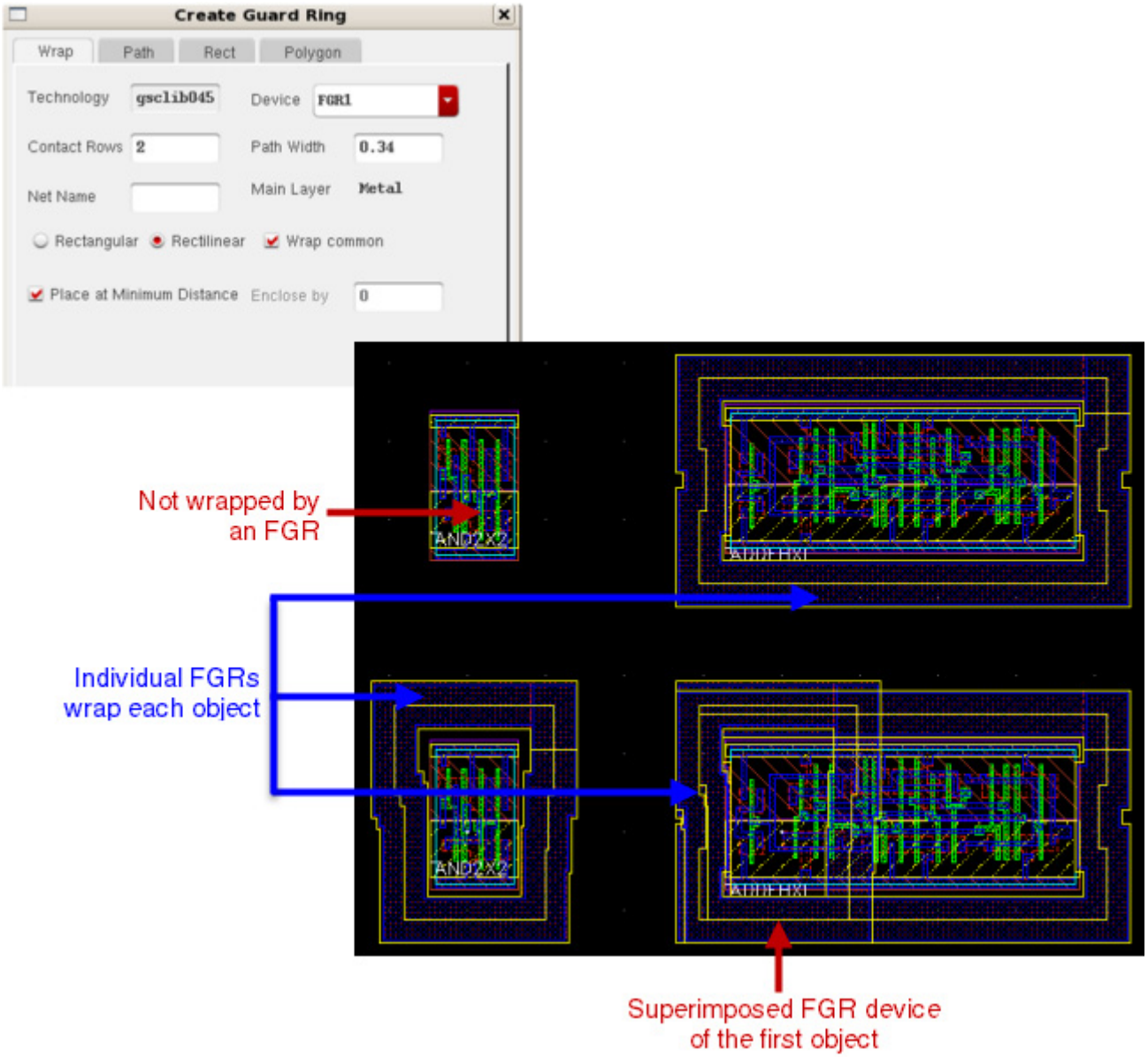
Introduction to Fluid Guard Rings

CCR 1186870: Issues seen while creating an FGR in Wrap mode with Wrap common and Rectilinear options selected

Description: The following issues have been observed when you create an FGR around a group of devices in *Wrap* mode with the *Wrap common* check box and *Rectilinear* radio button selected on the *Create Guard Ring* form:

- Individual FGR devices are generated for each object, instead of a common one to wrap all objects.

- One of the objects is not surrounded by an FGR. The corresponding FGR device gets superimposed on the FGR wrapping another object (as shown in the image below).



Solution: There is no workaround available for using wrapping objects with a rectilinear shape. If the multiple objects need to be wrapped with a single FGR, on the *Wrap* tab in the *Create Guard Ring* form, select the *Rectangle* radio button along with the *Wrap common* check box.

CCR 1179693: Unable to rename a library that contains layouts with fluid guard ring instances that use the version management feature

Description: The version management feature prevents geometry changes in the existing fluid guard ring instances (provided the instances were either created or edited using a release of Virtuoso that includes version management). When you save a layout in which fluid guard ring instances exist, this feature writes the fluid guard ring Pcell submasters to the disk. These submasters are saved in the design library of the layout in which they are instantiated. Such submasters have the cell name defined in the format `zzz_<FGR_device_name>`. For example, submasters of a fluid guard ring device, `fgr01`, have the cell name, `zzz_fgr01`.

When you rename a library, Virtuoso does not update the library information on any version-managed fluid guard ring instances in it. However, Virtuoso should ideally update this information to the renamed library.

Solution: Copy the old library to a new one, update the references, and then delete the old library.

CCR 1082970: Corner contacts correctly adhering to the DRC rules also get removed while creating and editing a Fluid Guard Ring

Description: Virtuoso automatically removes corner cuts from Fluid Guard Ring instances that are design-rule correct and need not be removed.

Solution: There is no workaround available.

CCR 13627: Convert to Polygon option to put resulting points on grid

Description: When a path is drawn at 45 degrees, and you convert the path to a polygon, the coordinates of the resulting polygon might be off grid.

Solution: You could write a SKILL program to check for (and possibly correct) off grid coordinates after paths are converted to polygons.

CCR 7231: Converting a 45-degree path to polygon results in off grid

Description: When you draw a path at 45 degrees, then the resultant path is off-grid.

Solution: There is no workaround available.

Known Problems and Solutions in Virtuoso Hierarchy Editor

This document describes the known issues with Virtuoso® Hierarchy Editor and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Virtuoso Hierarchy Editor Overview

Virtuoso Hierarchy Editor What's New

Cadence Change Requests

CCR 1171652: How do we keep case to avoid a red unbound instance in HED when ncvhdl is run?

Description: When you keep the case while using `ncvhdl`, the Hierarchy Editor (HED) may show red unbound instances whenever it tries to update the `ncvhdl` information that is imported into design libraries.

Solution: The workaround is to add the following syntax to the hdl.var file:

```
DEFINE NCVHDLOPTS -keepcase4use5x
```

Now, whenever Hierarchy Editor runs `ncvhdl`, the option to keep the case can be used.

CCR 2551940: Probe Assistant show different results depends the hierarchy

Description: When you define the constant `$default` in the view list it does not include the Top cell view which prevents probe tracing to the lower levels.

Solution: Specify the view name of the Top cell when defining `$default` in the view list to generate a complete view list including the view of the Top cell.

General Issues

Problem: Undo does not work with the Reference Verilog command

Description: When you reference a Verilog file for a cell, instance, or occurrence, you cannot undo it with the *Edit – Undo* command.

Solution: Instead of using the *Edit – Undo* command, right-click the cell, instance, or occurrence, and use the *Set Cell / Instance / Occurrence View* command to set the view to `<none>`. Alternatively, you need to click the *Undo* option twice.

Known Problems and Solutions in Virtuoso Layout Suite EXL

This document describes the known issues with Virtuoso® Layout Suite EXL and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to Virtuoso Layout Suite EXL

CCR 1990389: Constraints checker errors out with Layout XL license issue when working with Layout EXL constraints editor

Description: The PVS constraints checker issues an error with a Layout XL license issue when working with the constraints editor in Layout EXL: "Virtuoso Layout XL License is not currently checked out. This license is required to enable the OBC CV flow. Reopen design using Virtuoso Layout Suite XL or GXL license."

This is because the new Layout EXL application is not fully supported for PVS-CV and OBC-CV flows.

Solution: There is currently no workaround for this issue.

CCR 1984334: Limited options in Pegasus drop-down menu when launched through Layout EXL

Description: The *Pegasus* menu has only limited options available when invoked in Layout EXL. This is because menu options are added based on window type and `"VLS-GXL"` is no longer a valid window type in IC23.1. Pegasus has not yet been updated to add `"VLS-EXL"` as a supported window type.

Solution: There is currently no workaround for this issue.

Known Problems and Solutions in Virtuoso Layout Suite XL

This document describes the known issues with Virtuoso® Layout Suite XL and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Layout Editor Basics

Technology File Requirements for Layout XL

Connectivity Driven Editing Functions

What's New in Virtuoso Layout Suite XL

CCR 2676756: Stipple fill style disappears from layout after zoom-in

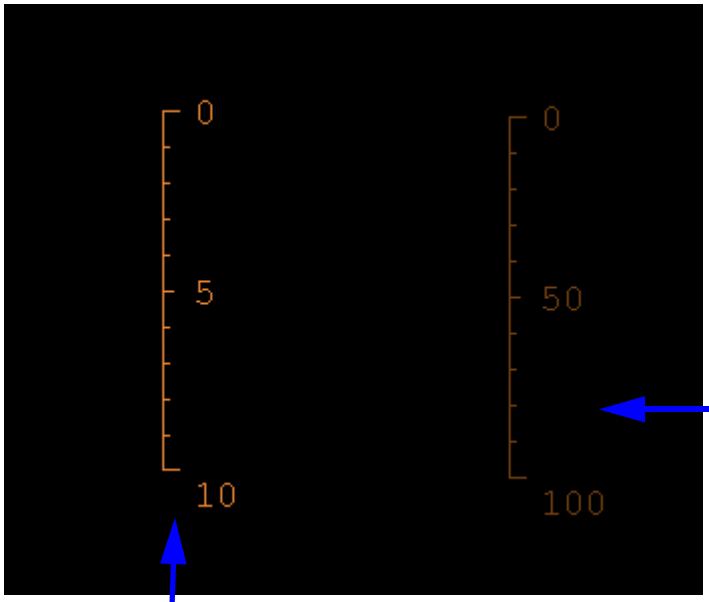
Description: When you choose *View – Zoom In*, the stipple fill style disappears from the layout after the zoom-in operation.

Solution: Install the latest drivers for graphics support.

CCR 2353478: Rulers are scaled in Edit in Concert mode by the ratio of DBUPerUUs

Description: In a package cellview, when using the Edit in Concert mode, rulers displayed in the background layout cellview get scaled up or down by the ratio of their DBUPerUU and package DBUPerUU if the rulers were created in the layout cellview.

For example, when you create a ruler of size 10 in a layout cellview, it might get displayed as size 100 in the background layout cellview in the package layout.



Ruler of size 10 created in the layout cellview displayed as size 100 in the background cellview

Ruler of size 10 created in the package cellview for comparison

Solution: There is no workaround available.

CCR 1404861: Crash observed in Property Editor for CDF list type parameter

Description: Currently the CDF parameter of list type is not supported for edit through the Property Editor. If user tries to edit CDF parameter of list type then it may result in a crash. The complete support is not in place in the current release.

Solution: There is no workaround available.

CCR 1279557: Changing via definition should take width and length specified values

Description: Currently when using the Property Editor assistant to change a via definition to a transition with different cutClass values, the old cut size is retained in the layout even if it is not valid with respect to the cutClass constraint for the layer in question.

Solution: Use the *Reset Parameters to* drop-down in the Property Editor to reset the cutClass to the default cutClass for the transition in question. Note that this resets ALL the parameters for the via definition in question.

CCR 942040: Need to preserve length in resistor

Description: If you have defined the split parameter value for a resistor as "`r R`" but Virtuoso Layout Suite XL still splits the length of the resistor across the segments, the order in which your CDF parameters are defined may be incorrect.

For example, you want to split a resistor in your design by using the split parameter, resistanceParamNames, which is defined as `r`. However, you have defined the CDF for the resistor to have a property named inductanceParamNames, `l`, which is listed before `r` in the **Edit CDF** form, as shown below:

```
s = 2

l = 4

r = 6
```

Since the parameter `l` appears before `r` in the Edit CDF form, Virtuoso Layout Suite XL uses `l` for splitting the parameter instead of using `r`. As a result, the resistor is split into two resistors of `l = 2` and `r = 6`.

To work around this issue, define the component types using the LAM file and specify the split parameter for the component type that needs to be split. For this example, the component type should be defined as `RESISTOR`, but for a different case, the component type could be defined as `INDUCTOR` or `CAPACITOR`, as appropriate.

Alternatively, to get the correct results:

- Change the order in which the parameters are defined in your CDF file. To do this, use the Edit CDF form from the *Tools* menu in the CIW.

  For the example given above, the CDF parameters should be listed as below for the resistor to be split using the `r` parameter:

  ```
  s = 2

  r = 6

  l = 4
  ```

  Here, since the `r` parameter is listed before `l`, the resistor is split as desired. As a result of the split, two resistors of `r = 3` and `l = 4` are obtained.

- Change the value of the inductanceParamNames environment variable to a value other than "1" so that Virtuoso Layout Suite XL does not incorrectly consider it as the split parameter value for the resistor.

CCR 853968: Need the Help Button ID for Object-Guides Options form

Description: When, you click the help button on the Objects-Guides Options form, the form help is not displayed correctly. Clicking the help button should take us to the correct documentation in cdnshelp.

Solution: There is no workaround available.

CCR 852661: Can select Mosaic even if Selectability is Off

Description: When you choose *View By – None*, Palette is unable to reflect the None state after the View or Sort By operations are performed.

Solution: There is no workaround available.

CCR 852639: Loading display.drf file does not update canvas despite Auto redraw ON

Description: When display.drf file is loaded manually from some location other than cwd, the layout canvas does not get updated despite Auto redraw set to ON in Palette.

Solution: To update the layout canvas view, choose *View – ReDraw/Fit-Edit*.

CCR 852254: ViewBy in Palette shows corrupted view after some View By/ SortBy operations in Palette

Description: View By in Palette shows corrupted view after some ViewBy or SortBy operations are performed in Palette.

Solution: There is no workaround available.

CCR 851265: Delete command is not enabled after loading "LSW Info File"

Description: The Delete command in the palette is not enabled even after loading *LSW Info File*.

Solution: There is no workaround available.

CCR 845311: Canvas is not synchronized with its layer palette

Description: When the two open cellviews are attached to two different technology files, the palettes are not synchronized. Also, the layout canvas is not synchronized with its palette, which is an issue.

Solution: There is no workaround available.

CCR 839265: Issues with Object and guides Lpps when changing display sets

Description: The visibility and selectability of the objects and elements in the Objects and Guides assistant are not in sync when I change the displaySet model.

Solution: There is no workaround available.

CCR 824448: Unable to close window having multi tabs after edits on routing direction on ITDB design

Description: Unable to close window having multitabs after edits on routing direction on ITDB design.

Solution: There is no workaround available.

CCR 820657: Issues with Layer Palette while working with ITDB

Description: In some cases, when using the Palette with ITDB the visibility changes are not reflected in the layout canvas.

Solution: There is no workaround available.

CCR 792878: Support backannotate to mfactored dummy symbols

Description: For PDKs that use `simM` as the schematic multiplier, the `mfactorNames` environment variable should be set to `simM` using:

```
envSetVal("layoutXL" "mfactorNames" 'string "simM")
```

This allows dummies to be generated as mfactored devices for such PDKs.

Note: Alternatively, if you require a different mfactor parameter in the schematic, you can add a CDF callback to the `mfactorNames` parameter, such as `m`. This will set the schematic mfactor parameter, for example, `simM` to `m`. The CDF callbacks will then be fired to keep the parameters in sync.

CCR 778617: After "View By None" column settings are not applied correctly

Description: The columns size setting is lost when you View by Layer and then View by None.

Solution: There is no workaround available.

CCR 694637: Layers not in the layerset remains valid when that layerset is loaded

Description: When a layerset is loaded, it is expected that all the layers in the layerset become valid and the layers become invalid. However, after loading a layerset, if the Edit Valid Layer form is opened, all the layers in *All Valid Layers* are shown as valid layers.

Solution: There is no workaround available.

CCR 640482: Rounding can only be applied to the width parameter

Description: To prevent folded devices from becoming off-grid; you can round their "width" parameter value. To do this, you must specify the appropriate rounding argument in the *Parameters* tab on the Configure Physical Hierarchy window.

Note that the window also accepts rounding values for Resistance, Capacitance and Inductance but does not support rounding for these parameters. The system does not generate any warning messages to indicate that rounding of these parameters is not supported.

Solution: This issue has been documented for your information only. There is currently no workaround available.

CCR 419455: mFactor split does not work on hierarchical design

Description: The `mfactorSplit` parameter is not handled correctly on hierarchical designs. Layout XL does not generate the correct number of mfactored instances. When the mfactor is changed, Layout XL does not succeed in binding all the instances generated.

Solution: This issue has been documented for your information only. There is currently no workaround available.

CCR 318593: Edit Properties brings up old form

Description: The *Edit Properties* form is not displayed in the new format.

Solution: There is no workaround available.

CCR 240272: No warning is issued when auto via is not placed because it is not defined in the constraint group

Description: If a valid via is not defined in the constraint group used in the current context, and you try to place a via using auto via, the via is not placed and no warning is issued. For example, if path segments intersect on layer 3 and layer 4 and no vias are defined in the current context constraint group to traverse layers 3 and 4, no vias are valid, so none are placed.

Solution: Define valid vias in the corresponding constraint group in the technology file.

CCR 8060: icfb errors if first cellview opened is abstract

Description: When the first cellview you open has the view name `abstract`, the menus in the layout editor window banner are not built correctly. The Tools menu is the only menu displayed in the editing window banner.

Solution: Select *Tools — Layout*, then *Tools — Abstracts* and the menu banner will appear correctly.

CCR 6968: Layer Selection in Edit Properties Form

Description: When the number of layers defined is large (500+), the system might not be able to display all of the layers in the layer menu on an Edit Properties form.

Solution: Specify a smaller number of valid layer-purpose pairs. Alternately, you could subdivide the full set of layer-purpose pairs into separate technology files, each of which would define a subset of the total number of valid layer-purpose pairs. You could then assign bindkeys to facilitate the quick loading of the technology files.

CCR 4393: The Layer Selection Window (LSW) does not recognize Opus*LSWGeometry setting

Description: When `Opus*LSWGeometry` is set in the `.Xdefaults` file, the Layer Selection Window displays the name *Abbrv* instead of the layer name at initialization

Solution: There is no workaround available.

CCR 4162: Underscore characters disappear in Editor Properties Form

Description: Underscore characters are not visible in text entry fields on the Edit Properties form when the text default font (- *-courier-medium-r-*-*-12-*) is used.

Solution: Do not set the default text font to (- *-courier-medium-r-*-*-12-*. Instead, set it to (-adobe-courier-bold-r-*-*-12-*), like this:

```
hiSetFont("text" "-adobe-courier-bold-r-*-*-12-*")
```

You can set the default text font by adding the statement above to your `.cdsinit` file.

CCR 3397: Hierarchical marker deletion sometimes does not work

Description: You cannot delete markers below the top level when lower levels are open in read-only mode.

Solution: Open all necessary cellviews in append mode by typing the following in the CIW:

```
foreach(cv gevMarkerCVList
        dbReopen(cv "a")
)
```

Then try to delete the markers again with the *Verify — Markers — Delete All* command.

CCR 3361: Verify – Markers – Explain gives no response for dot markers

Description: The *Verify — Markers — Explain* command does not work for dot markers.

Solution: Use *Verify — Markers — Find* to display the reason for each marker.

Known Problems and Solutions in Virtuoso Layout Viewer

This document describes the known issues with Virtuoso® Layout Viewer and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: The issues described in this document were identified in IC23.1. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

CCR 1986728: Layout Viewer: Navigator Default Open option not consistent with the behavior

Description: The *Default Open – Mode – edit* option available in the Navigator Options form lets you open a selected instance cellview in either *edit* or *read* mode. However, Layout Viewer does not support the edit operation. So, this option is not required in Layout Viewer.

Solution: *For information only.*

Known Problems and Solutions in Virtuoso Module Generator

This document describes the known issues with Virtuoso® Module Generator and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Module Generators (Modgens)

CCR 1430332: The Modgen toolbars are not displayed for designs with old local customer toolbar definitions (Virtuoso_XL.toolbars)

Description: Modgen toolbars are not displayed because of a backward compatibility issue with an old toolbar configuration (Virtuoso_XL.toolbars). This happens, for example, when a designer copies an old toolbar configuration file (from the standard shipped Virtuoso – a pre-IC6.1.6 version), modifies as per requirements, and uses it in the newer versions of Virtuoso.

Post IC6.1.6, the Toolbar Manager (CIW/Options/Toolbars or deCustomizeToolbars()) is used to customize the toolbar organization, appearance, and visibility in the different applications (such as Schematic, Layout XL, and Layout EXL). These customizations are saved in corresponding toolbar overlay files within the `.cadence` user directory.

When Virtuoso is launched, the first `<AppName>.toolbars` file (example-`Virtuoso-XL.toolbars` file) found is automatically loaded. In this case, the old toolbar configuration file is loaded. Consequently, the new toolbar configuration files (or the toolbar overlay files) fail to load.

Solution: Depending on the situation, choose one of the following solutions:

#	Situation	Solution
1	Old toolbar customizations are not required.	Delete any existing (old) toolbar custom file from the `.cadence` directory.

2	Preserve old customizations.	Use the Toolbar Manager to edit the toolbar settings and to add the required customizations. A toolbar overlay file is created that surgically modifies the standard shipped Virtuoso toolbar, allows multiple levels of overlay, and allows the final toolbar setting to be always up-to-date.

Note: It is possible to directly edit the existing toolbar overlay toolbar files (perform tasks such as copy, paste, or correct formats) as per the required customer customizations. However, it is important to maintain the format of the overlay files, avoid conflicts, and preserve transparency between overlays. As this method is prone to errors, Cadence does not recommend it.

Known Problems and Solutions in Virtuoso Multi-Patterning Technology

This document describes the known issues with Virtuoso® Multi-Patterning Technology (MPT) and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Virtuoso Multi-Patterning Technology

CCR 1546323: Cannot automatically change the color or color state of gray shapes in fully locked vias

Description: Prior to ICADV12.2 ISR1, vias were typically color locked on a per-via basis. In ICADV12.2 ISR1, the Property Editor was enhanced to enable via color locking on a per-layer basis. However, this capability requires the use of a new coloring infrastructure that can be read in ICADV12.1 and versions of ICADV12.2 prior to LA6, but not edited. To maintain compatibility with earlier releases, the Property Editor uses full via color locking whenever possible in ICADV12.2 ISR2. Via colors are locked on a per-layer basis only if you explicitly color and lock some layers, but keep at least one layer colored and unlocked. As a result, vias that have a mix of colored and locked layers, and uncolored but colorable layers, are marked as fully locked.

If a design has fully locked vias with any colorable layers that are gray (uncolored), the following situations may occur in ICADV12.2 ISR2:

- Shifting the color of a metal shape touching the via's gray shape will result in the following warning from the coloring engine:

  ```
  *WARNING* (MPT-20204): Could not shift color because the cluster contains some lock elements.
  ```

- Lock propagation from a wire that touches the via to the via's gray layer will fail:

  ```
  *WARNING* (MPT-20204): Coloring engine fails to propagate color information on some vias due to no color state precedence.
  ```

- Moving a shape locked to one color to touch an uncolored shape inside the hierarchy, which in turn touches the gray shape of a locked via, causes lock propagation to fail:

```
*WARNING* (MPT-20204): Coloring engine fails to propagate color information on
some vias due to no color state precedence.
```

You cannot create a hierarchical color lock on a gray shape in a fully locked via in a child:

```
*WARNING* (MPT-20204): Cannot create a lock on the shape because it already has
locked mask1Color.  Creating locks on top of shapes that are already locked in
the hierarchy is currently not allowed.  Ensure that the hierarchical shape is
not already locked before creating a lock on it.
```

Note: In ICADV12.2 ISR3, the Property Editor will lock only colored layers of a via. It will no longer lock a gray but colorable layer of a via which had prevented the color engine from propagating color to those layers.

Solution: Use the Property Editor to set the color lock state to unlocked for the colorable but gray layers of the via to unlocked.

CCR 1489934: Colored purposes cannot be changed during a session

Description: Only shapes on a predefined purpose in the list given by the `explicitColoredPurposes` environment variable will be colored by the color engine. Changes to the environment variable are not recognized during the current session.

Solution: Open a new session with the `explicitColoredPurposes` environment variable set to the correct list of purposes to be colored. For example, this command sets only the shapes on the `drawing` purpose to be colored by the color engine:

```
envSetVal( "mpt" "explicitColoredPurposes" 'string "drawing")
```

CCR 1284731: Mask color representation in the Multiple Patterning assistant cannot be customized

Description: The mask color representation in layout can be customized by setting the display resource file (`display.drf`). However, the custom colors are not represented in the Multiple Patterning assistant, which can only show the default mask colors.

Solution: There is no workaround available.

CCR 1278844: Outdated coloring is not detected if the design was saved prior to ICADV12.2 ISR2

Description: Outdated coloring can occur when the data is edited with the color engine off, and then is saved. It is possible that outdated coloring in a design that was saved prior to ICADV12.2 ISR2 will not be detected when the design is opened with the color engine

activated. When this happens, the Automatic UpdateColor form will not appear and coloring is left in an unknown state, which can result in unusual behavior.

Solution: Save the design using ICADV12.2 ISR2 or a newer release.

CCR 1236699: Recolor All does not always sync color on connected metal within congested routing

Description: In most cases, Recolor All will color all connected shapes on a metal layer with one color. In dense routing areas after Recolor All, overlapping shapes on the same metal layer, primarily Metal1, might not have the same color.

Solution: The workaround is to run Virtuoso IPVS to resolve odd cycle loops.

CCR 1178544: mptReconstructStitch function cannot recreate hierarchical stitches

Description: Stream In and Stream Out now support the translation of stitch objects hierarchically. The `mptReconstructStitch` function must be run after streaming in data with stitches to reconstruct the stitch shapes. However, `mptReconstructStitch` does not traverse the layout hierarchy, so hierarchical stitch reconstruction is not currently supported.

Solution: There is no workaround available.

CCR 1177235: Cannot highlight locked shapes in instances using Multiple Patterning Assistant

Description: When *Highlight Locked Color* is enabled in the Multiple Patterning Assistant, the locked shapes in instances are not highlighted. This is because the Multiple Patterning Assistant Highlight function operates only on top-level shapes.

Solution: There is no workaround available.

CCR 1070535: Select and Unselect on color locked shapes through the hierarchy can pose significant performance degradation

Description: Selecting and deselecting locked color shapes shows significant performance degradation if the circuit is comprised of more than two hierarchies. The functions will traverse down to every hierarchy regardless of whether locked color shapes are present in the hierarchy and will perform exhaustive color state validation, slowing down performance.

Solution: There is no workaround available.

CCR 1050621: *Verify — Design* is not removing loop markers for pathSegs with Stitching

Description: Stitching is used in Virtuoso to fix multi-patterning odd loop conflicts. Currently, after fixing a multi-patterning odd loop conflict with Stitching, *Verify – Design* in Virtuoso IPVS mode still reports it as an odd loop violation.

Solution: There is no workaround available.

CCR 1042773: Vias are not displayed in correct stipple dashed outline for NetClass groups

Description: Vias in a precolored NetClass group are not colored with a default stipple dashed outline for grouped shapes. This occurs because only top-level shapes in precolored nets will be shown as grouped shapes with the dashed outline.

Solution: There is no workaround available.

CCR 1004586: Circles and ellipses are not being colored when using Virtuoso MPT for interactive coloring

Description: Virtuoso MPT interactive coloring supports most layout shapes, such as rectangles, polygons, paths, and path segments. However, automatic coloring of circles and ellipses is not currently supported.

Solution: Circles and ellipses are not expected to be used in advanced node designs. Contact Cadence Customer Support if you require this feature.

CCR 993382: Hierarchy flattening command cannot be used on designs with coloring

Description: Designs with coloring from Virtuoso Multi-Patterning Technology (MPT) do not support the Hierarchy Flattening command (*Edit—Hierarchy—Flatten*) in Virtuoso. When you run hierarchy flattening on a hierarchical design with coloring, the data reaches an inconsistent state. Running subsequent editing commands can result in a warning message, hang, or crash.

Solution: To avoid this problem, do not run hierarchy flattening on designs with coloring. Instead of flattening a cell, replace the instance of the cell with the contents of the cell by doing a simple copy of all the shapes in the cell.

Known Problems and Solutions in Virtuoso Multi-Technology Solution

This document describes the known issues with Virtuoso® MultiTech Framework and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Topics

Virtuoso Multi-Technology Solution

CCR 2265010: An error is reported in CIW when you click the Select Part field for an SMD present in muratalib/smdLib

Description: Clicking the Select Part field for SMD instance reports the following error in CIW:

CDF: An error occurred when evaluating callback.

Solution: This happens because the SMD library is generated from a previous version (before ICADVM18.1 ISR10) of Virtuoso. To resolve this, recreate the SMD library using public SKILL API `vmtcsvCreateComponentCellViewsFromCsv()`.

Known Problems and Solutions in Virtuoso NC Verilog

This document describes known issues with the Virtuoso® Verilog Environment for NC-Verilog Integration (NC Verilog Environment) and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

About the NC Verilog Environment

CCR 1104924: Simulation settings do not update when you switch between the NC Verilog Environment and the SystemVerilog Integration Environment

For the description and workaround of this issue, see the *SystemVerilog Integration Environment Known Problems and Solutions*.

Known Problems and Solutions in Virtuoso Parameterized Cell

This document describes the known issues with Virtuoso® Parameterized Cell compiler and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Overview of Virtuoso Parameterized Cell

Virtuoso Parameterized Cell SKILL Reference

CCR 1761600: If PCELL parameter name is given as "cellName" then the Create Instance form cannot be opened after instantiate the PCELL

Description: The Create Instance form cannot be opened after the Pcell is instantiated and the following error message is displayed in CIW:

```
\e *Error* putprop: first arg must be either symbol, list,
defstruct or user type - nil
 or
\e *Error* dbFindOpenCellView: argument #2 should be a string
(type template = "gttg") - nil
```

This problem can occur whenever there is a conflict between the Pcell parameter name and the form field name. For example, if you specify `cellName` as an argument in the `pcDefinePCell` function, the Create Instance form does not open after the Pcell is instantiated. As a result, the field name `Cell` is replaced with `cellName` in the Create Instance form. In addition, this form is not displayed on subsequent invocation and the error message is displayed.

Solution: Exit the Virtuoso session, correct the Pcell definition, and recreate its master before invoking the `Create Instance` command.

Note: This issue can also arise with the Edit Property form.

CCR 854985: Express Pcell is not working with Calibre on linux32/64 and sol86 64bit ports

Description: Appropriate changes need to be made in third party tools scripts to accommodate the GCC 4XX paths. As a workaround, you need to set the `LD_LIBRARY_PATH` for linux 32bit as:

```
$OA_HOME/lib/linux_rhel40_gcc44x_32/opt
```

and for linux 64 bit as :-

```
$OA_HOME/lib/linux_rhel40_gcc44x_64/opt
```

CCR 694579: VLE crashes with undo command

Description: Pcell code creates instance and subMaster to have different value for the same parameter.

Solution: Pcell should not have a code that changes the value of a parameter, such as:

```
pcCellView~>parameters~>[paramName] = <value>
```

CCR 684824: Express Pcells slow in IC6.1.4EAP when cache directory is on a NFS mounted file server

Description: Express Pcell infrastructure may leave some stale scratch files in case of a crash in Virtuoso or in any third-party tool using the plugin.

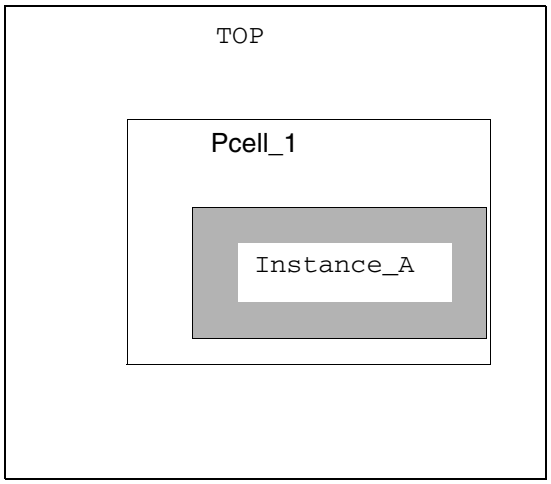
Solution: Currently, a workaround is not available. This limitation will be resolved later through CCR 741126.

CCR 491144: hierarchical copy bug

CCR 490865: incorrect library tech info after hierarchical copy

Description: When you use the Library Manager to perform an hierarchical copy of a cellview from one library to another library, and the cellview contains a Pcell instance, the system does not rebind instances contained in the Pcell to the new library name. Any instances contained in the Pcell remain bound to the original library name.

In the example below, the cellview `TOP` is in `LIB_AA`. `TOP` contains a Pcell, `Pcell_1`, which contains `Instance_A`.



After you copy `LIB_AA TOP layout` to `LIB_BB TOP layout`, Instance_A in LIB_BB is still bound to LIB_AA.

- Updates to `Instance_A in LIB_AA` also affect `Instance_A` in `LIB_BB`.
- Updates to `Instance_A` in `LIB_BB` change Instance_A in LIB_AA, but do not change Instance_A in LIB_BB.

Solution: After copying cellviews that have Pcells containing instances, you must force the system to rebind the instances by recompiling the Pcells. You can recompile all of the Pcells in your library, or just recompile the Pcells in a specific cellview, by using the script `compilePcell.il` that is provided below.

- To recompile all Pcells in a specific library, do the following:
  - In the CIW, load the following script: `compilePcell.il`
  - In the CIW, type the following:

    `pcRecompileLibPcell( t_libName )`

    For example, type

    `pcRecompileLibPcell( "LIB_B" )`
- To recompile only the Pcells in a specific cellview, do the following:
  - In the CIW, load the following script: `compilePcell.il`
  - In the CIW, type the following:

    `pcRecompilePcellUnderHierCell( t_libName t_cellName t_viewName )`

For example, type

```
pcRecompilePcellUnderHierCell( "LIB_BB" "TOP" "layout" )
```

Code for compilePcell.il Script

```
procedure( pcRecompileLibPcell(libName)
 prog((lib cv cellName viewName)
     lib = ddGetObj(libName)
     when( !lib
       warn("Library %s does not exist.\n" libName)
       return(nil)
     )
     foreach( cell lib~>cells
        foreach( view cell~>views
           cv = dbOpenCellViewByType(lib cell~>name view~>name "" "r")
           info("cv ( %s %s )\n", cv~>cellName cv~>viewName)
           cellName =  cv~>cellName
           viewName = cv~>viewName
           when( cv~>isParamCell
              dbReopen(cv "a")
              pcHIDefineParamCell(cv)
              dbSave(cv)
              ;dbWriteSkill(cv "/tmp/t.il" "w" "4.4")
              ;dbPurge(cv)
              ;load("/tmp/t.il")
           )
        )
     )
     return(t)
))


procedure( pcRecompilePcellUnderHierCell(libName cellName viewName)
 prog((cv master)
     cv = dbOpenCellViewByType(libName cellName viewName "" "r")
     ;info("cv ( %s %s )\n", cv~>cellName cv~>viewName)
     foreach( ih cv~>instHeaders
        master = ih~>master
        when( master
           when( master~>isParamCell
              dbReopen(master "a")
              pcHIDefineParamCell(master)
              info("Parameter cellview ( %s %s ) in library %s is recompiled.\n",
                   master~>cellName master~>viewName master~>libName)
              dbSave(master)
           )
           pcRecompilePcellUnderHierCell(master~>libName
                      master~>cellName master~>viewName)
        )
     )
     return(t)
))
```

Known Problems and Solutions in Virtuoso Parasitic Aware Design

This document describes the known issues with Virtuoso® Parasitic Aware Design and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Parasitic Aware Design in Virtuoso ADE Explorer, Virtuoso ADE Assembler, and Virtuoso Schematics L/XL

Virtuoso Parasitic Aware Design SKILL Functions

CCR 2874709: Parasitic Comparison is not working in IC23.1 ISR2

Description: When you choose the *Parasitics/LDE – Compare* command after running a post-layout simulation, the comparison report is not displayed and the CIW shows the following error:

```
*Error* putprop: first arg must be either symbol, list, defstruct or user type -
nil.
```

Solution: This issue has been documented for your information only. There is currently no workaround available in the IC23.1 ISR2 release. It is fixed in the IC23.1 ISR3 release.

CCR 2832012: Annotation of DC operating points to the schematic fails when simulation is run using an extracted or Smart View

Description: Consider that you run a simulation where the design is bound to an extracted or Smart View.

When you run a simulation with the design bound to the schematic, you can annotate the DC operating points, but when you bind the design to the extracted or Smart View, you cannot annotate the DC operating points on the schematic and an error is reported.

On checking the contents of each view, you observe that the schematic has resistors R0, R1 and R2, but these resistors are not present in the layout or the extracted view.

```
sch->instances~>name
("M0" "PIN1" "PIN0" "R2" "R1"
    "R0" "PIN2" "PIN3" "M1"
)

lay->instances~>name
("M1" "M0")

ev->instances~>name

("M0" "M0_7__rcx" "M0_6__rcx" "M0_5__rcx" "M0_4__rcx"
    "M0_3__rcx" "M0_2__rcx" "M0_1__rcx" "M1" "M1_7__rcx"
    "M1_6__rcx" "M1_5__rcx" "M1_4__rcx" "M1_3__rcx" "M1_2__rcx"
    "M1_1__rcx" "D16_unmatched" "C1" "C2" "C3"
    "C4" "C5" "C6" "C7" "C8"
    "Rh3" "Rh4" "Rh5"
    "Rh6" "Rh7" "Rh8" "Rh9" "Rh10"
    "Rh11" "Rh12" "Rh1" "Rh2" "Ri1"
    "Ri2" "Ri3" "Ri4" "Ri5" "Ri6"
    "Ri7" "Ri8" "Ri25" "Ri26" "Ri27"
)
```

Solution: Annotation of DC operating points on the schematic might fail if you do not have the same device in the layout as the schematic.

To annotate the DC operating points, add a small metal resistor to the layout.

CCR 2569502: The DSPF file generated after Smart View extraction does not contain the .SUBCKT statement of the blackbox cell

Description: Performing a DSPF extraction on a block that contains a blackbox cell creates a DSPF file with an additional .SUBCKT statement for the blackbox cell. However, if Smart View extraction is done on the same block, the generated DSPF file may not contain the additional .SUBCKT statement. The simulation runs even without any circuit in the blackbox cell.

Solution: The `.ccl` file used to generate the Smart View contains an option, `-suppress_empty_subckts`. If this option is set to `true`, the .SUBCKT statement is not printed. To resolve this issue and extract the Smart View successfully, comment out this command or set it to `false` in the `.ccl` file.

CCR 2460528: Simulation fails with errors during Smart View netlisting when Print Comments is enabled

Description: When running simulation with a Smart View, the simulation fails with the following error:

```
ERROR (SFE-23): The instance `//' is referencing an undefined model or subcircuit,
`Instance'. Either include the file containing the definition of `Instance', or
define `Instance' before running the simulation.
```

This is because the netlist contains comments to indicate the name mapping, as shown in the following example:

```
// Library name: Two_Stage_Opamp
// Cell name: OpAmp
// View name: schematic
// terminal mapping: AVDD         = AVDD
//           AVSS         = AVSS
//           Ibias        = Ibias
//           inm          = inm
//           inp = inp
//           out = out
subckt OpAmp AVDD AVSS Ibias inm inp out

// pmos1v Instance M4 = spectre device M4
    M4 (net10 net6 AVDD AVDD) g45p1svt w=(33u) l=700n nf=6 as=8.58p \
    ad=6.6p ps=47.12u pd=35.4u nrd=6.06061m nrs=7.87879m sa=380n \
    sb=380n sd=400n sca=4.00000 scb=0.00130 scc=0.00000 m=(1)
```

These comments have been added because the *NameMapping* check box in the *Print Comments* group box of the Environment Options form is selected.

Solution: To resolve this issue, deselect the *NameMapping* check box in the *Print Comments* group box of the Environment Options form.

CCR 1940921: An extra space added before the closing parenthesis in subcircuit port list in DSPF causes an error for simulations run using a Smart View

Description: Netlisting procedures in some process design kits (PDKs) add an extra space at the end of the port list before the closing parenthesis.

The following example of a subcircuit port list shows an extra space between `ABC` and the closing parenthesis:

```
XI1|I0|IN|R2 (I1\|OUT12\(5\) I0\|I2\|ABC ) rm1w l=1e-06 + w=2.4e-07 multi=(1)
```

For simulations run using the Smart View, this extra space results into an error, as shown below:

```
ERROR (SFE-46): ` XI1|I0|IN|R2 ': An instance of 'rm1w' can have at most 2 terminals
(but has 3).
```

Solution: To resolve this issue, use the nlIsSmartExtractedView SKILL API inside the custom netlist procedure, as shown in the example given below.

```
(nlIsSmartExtractedView ((nlGetCurrentCellView (nlGetNetlister inst)))

=> t/nil
```

Here, `inst` is an instance handle passed as an argument to the netlist procedure.

This function identifies the cellview of the specified instance as Smart View and allows updating the netlist procedure to avoid printing parenthesis surrounding the instance port connections.

CCR 1521824: Out of context probing with iterated instances in the extracted view

Description: When an extracted view is created, the drain and source pins may get swapped resulting in errors while creating a netlist using an extracted view.

Solution: Specify the permute rule in the cdf:auLvs property for MOS devices. To do so:

1. Open the Edit CDF form and select the required library and cell.
2. Click the *Simulation Information* tab.

3. Select the *By Simulator* radio button and select *auLvs* from the drop-down.
4. Set the *permuteRule* field as `(p D S)`.
5. Click *OK* to save the setting.

CCR 1130709: Parasitic Compare does not work with custom parasitic models

Description: The parasitic compare function cannot compare nets that contain custom parasitic models.

Solution: This issue has been documented for your information only. There is currently no workaround available.

CCR 1130604: Parasitic reporting does not work with custom parasitic models

Description: When using custom parasitic models in a design, the Parasitic Report Assistant does not show the values of the parasitics on the nets. Also, the values are annotated on the schematic as r=NA.

Solution: This issue has been documented for your information only. There is currently no workaround available.

CCR 726299: Term-to-term probing fails for a net between two instances of the same cell

Description: In Parasitic Reporting, when doing Terminal-to-Terminal probing on a net between two instances of the same sub-block, if you perform the terminal selection at the lower level in schematic tabs open on each of the two instances, the probing will fail and you get error messages in the CIW.

Solution: To do term-to-term probing on a net between two instances of the same sub-block, select the terminals at the higher level in which the two instance of the sub-block are instantiated.

CCR 691686: Only a complete parasitic estimate can be overridden

Description: The entire parasitic estimate will get disabled when overriding a parasitic on a member-by-member basis.

Solution: The entire parasitic estimate will be disabled when a parasitic estimate in a higher-level cellview tries to override any of its members, even though it may not override all its members. Overriding of individual members in parasitic estimates is not supported.

CCR 671880: RL estimate that is overridden by the Move to current cell command becomes a R estimate

Description: If there is an existing RL estimate set on a net in a subcell and a *Move to current cell* is run on that estimate, the overriding estimate becomes a R estimate.

Solution: Either remove the RL estimate in the subcell and recreate a new one in the current cell, or complete the R estimate override with an L estimate, after the *Move to current cell* command is run.

CCR 663420: Decoupled C estimate created with global or inherited net is flagged as out-of-context

Description: When the global or inherited net used for the definition of a decoupled C estimate does not exist in the current cellview, the net is being flagged as out-of-context.

Solution: It is currently recommended that you only use local net names for decoupled C estimate.

CCR 658718: A parasitic K estimate created between 2 RL estimates is not removed when stitching the RL from the extracted view

Description: K estimates created manually between RL estimates are not relevant when replacing the RL estimate with stitched RL. The K estimate should be removed automatically after stitching, but that is not currently happening.

Solution: The K estimate must be removed manually when stitching the respective RL. Note however, that is not essential that the K estimate be removed, as VPAD will ignore it.

CCR 525844: Net expressions with same default can conflict in estimated view

Description: A current limitation of net expressions is that multiple expressions, in the same cellview, cannot have the same global net. This can cause problems when net expressions from two different schematic views have the same default, but different property names, and are copied to the estimated view. If this occurs, a warning will be issued and the net expressions will be combined under a single name. The result of this is that they will both keep the same default, but will not be overridden separately.

Solution: Currently, a workaround is not available.

CCR 502541: Estimated parasitics report does not match Parasitic Estimates assistant

Description: Generating reports on estimated parasitics (by selecting *Parasitics – Estimated – Report Parasitics*) will give incorrect results if the estimates have been modified in the *Parasitic Estimates* assistant but the estimated view has not been rebuilt.

Solution: The reports are generated from the estimated view, not from the estimates in the *Parasitic Estimates* assistant. To obtain correct results you should rebuild the estimated view after making edits in the assistant. To ensure ~R values are correct, re-run the DC oppoint simulations and choose the new simulation results in the Setup Parasitic Estimates form.

CCR 495534: Parasitic Sweep drop-down operates on individual parameter options

Description: The *Parasitic Sweeps* option operates by modifying individual parameter settings in the *PSC* assistant. This has two consequences:

1. When parameters are disabled globally (using the *Parameters* option in the *Data View* assistant or the *Point Sweep* option in the *Run Summary* assistant), the *Parasitic Sweeps* option will also be disabled because modification of parameter options are not possible.
2. When switching the parasitic sweep mode, all device and/or parasitic parameter check boxes are updated without regard to their status when the mode was previously selected. That is, if individual parameters are disabled while in a particular mode, those settings are lost as soon as the mode is switched to another, and they will not be restored on returning to the original mode. This is contrary to the global *Parameters* and *Point*

Sweep options, which preserve the enabled/disabled state of individual parameters so that they are restored when parameters are globally re-enabled.

Solution: Currently, a workaround is not available.

CCR 464828: Parasitic browser content is read-only and disappears when made editable

Description: When opening or descending into a LCV, where the cell has been bound to a different view in an ADE Assembler test, the parasitic filter and estimate browsers may open in read-only mode. On changing the browser mode to Editable, browser content may become invisible.

Solution: Closing and re-opening the browser assistant will bring back the content.

Also, using parasitic modes to update test configs rather than rebinding them manually will help avoid the problem, since it is the bindings at the tool startup that matter (parasitic modes leave the config pointing to the schematic during startup).

CCR 276565: Crash when reporting on net that does not exist in associated extracted view

Description: If you choose *Parasitics – Report Parasitics*, then select a net from an incorrect schematic view (that is, the selected net does not exist in the associated extracted view), a crash can occur.

Solution: You should change the schematic view in the extracted view setup form (*Parasitics – Setup*) to match the top-level schematic where the selection is to be made.

Known Problems and Solutions in Virtuoso Relative Object Design

This document describes the known issues with Virtuoso® Relative Object Design (ROD) and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Relative Object Design Concepts

Virtuoso Relative Object Design SKILL Functions

CCR 916633: IC615 crashes when modifying MPPs (stretch, move origin, etc)

Description: Virtuoso closes abruptly when trying to edit multipart paths (MPPs) containing corrupt internal data.

Solution: Use the following SKILL APIs to check and/or repair faulty MPPs:

- `rodCheckMPPs`: Checks if the given list of MultiPart Paths contains corrupt internal data. If MPPs containing defective internal data are found, it fixes and/or creates markers to highlight them.
- `rodCheckAllMPPsInCellView`: Finds and opens the design specified and checks if any MPP in that design contains corrupt internal data. If MPPs containing defective internal data are found, `rodCheckAllMPPsInCellView` fixes them and creates markers on them.

For more information on `rodCheckAllMPPsInCellView` and `rodCheckMPPs` see the *Virtuoso Relative Object Design SKILL Reference.*

CCR 24535: Undo after moving aligned objects from one window to another, breaks ROD alignment

Description: *Undo* of edits involving ROD-aligned objects may alter the alignments.

Solution: Users should avoid using *Undo* on aligned ROD objects. The workaround is to realign the objects.

CCR 20965: Chopped MPP being regenerated w/ stretch

Description: This occurs during the stretch of a multipart path (MPP) if a segment that is part of a chop hole is partially selected. When a horizontal segment along the chop hole is selected and stretched downward and two horizontal segments of the chop hole line up, the chop hole refills.

Solution: You can avoid this problem by stretching before you chop. If you stretch after chopping and the chop hole is lost, you can fix it by repeating the chop either interactively with the *Chop* command or by using the `rodAddMPPChopHole` function.

CCR 18896: ROD alignments broken upon move and rotate to different cellview

Description: If a ROD object aligned to another ROD object is rotated, the aligned object should be rotated as well. Also, if the handles used in the alignments change their locations on the object due to the rotation (for example, `upperRight` becomes `upperLeft` after a 90 degree rotation), the handles stored in the alignment should be updated.

Solution: Avoid using the *Rotate* or *Move* commands to rotate ROD objects that have alignments. If objects need to be aligned and rotated, rotate them first and then align them.

CCR 18690: rodCreatePath flipping endpoints

Description: When creating a bus using `rodCreatePath` and offset subpaths, if a vertex is moved along the main path after at least one bend in next segment of the path, the endpoints on the subpaths flip upside down. This can be observed in a multipart path as well.

Solution: Specify the middle path as master path instead of the right or left subpath.

Note: Multipart path (MPP) templates are stored in the binary form of your technology library. If you want the MPP templates that you create with the Create ROD Multipart Path form to persist in your technology library, you must not load new technology data that includes the `lxRules` class using the *Replace* mode. Loading a new `lxRules` class into your current technology library using *Replace* causes the MPP template data to be lost.

CCR 7929: Stretchable Pcells do not evaluate CDF callbacks

Problem: Stretching a Pcell does not execute CDFs associated with the Pcell parameters.

Solution: Specify user-defined functions to perform the tasks formerly performed by CDFs. For information about user-defined functions, see the description of the `rodAssignHandleToParameter` function in the *Virtuoso Relative Object Design User Guide*.

CCR 7195: Create MPP doesn't allow IO type of existing term to change

Description: When you try to create a multipart path and specify a subpart as a terminal, and there is already a terminal in the cellview with the same terminal name but with a different terminal type (for example, Input/Output), the system does not create the subpart. The system displays messages in the CIW similar to these:

> w *WARNING* rodCreatePath: rodiCreateConnectivity "output": not allowed because terminal "A" already exists with I/O Type of "input" \w *WARNING* rodCreatePath: unable to put object on terminal and net - "" \w *WARNING* rodCreatePath: rodiCreateMPP: ROD internal warning. Unable to create MPP sub part

What should happen is that the system should change the type of the existing terminal to match the MPP terminal type and display a warning message saying that the type of the existing terminal has been changed.

Solution: Before creating a multipart path (MPP) with a subpart that is specified as a terminal, change the type for any existing terminals that have the same name so they match the terminal type you want to specify for the MPP subpart terminal.

CCR 6189: Crash after dbCreateInst followed by a hiUndo

Description: If you create an instance using the `dbCreateInst` function in the CIW and a ROD shape already exists with the same name in the same cellview as the instance you are creating, the system displays the following Warning in the CIW:

```
WARNING* rodiFigTriggerFunc: Creating instance named instance_name forced unname
of ROD shape_type to avoid name conflict.
*WARNING* You might want to use Undo now.
```

If you do not do an *Undo*, the system unnames the ROD object, turning it into an ordinary shape.

Also, if after doing an *Undo*, you select the object that was unnamed and query its properties, the system might crash.

Solution: Before doing the *Undo*, de-select the object that was unnamed.

CCR 4839: MPP text edits not retained when the MPP gets regenerated

Description: A text display object that is part of a ROD multipart path (MPP) can be individually selected and its properties can be changed with the property editor (for example, layer, font, size, etc.). However, when the MPP is regenerated for any reason, the text display object is recreated with its original properties. A stretch, move, or chop of an MPP causes the subparts of the MPP to be regenerated as does an undo of any of those operations.

Solution: After an MPP is regenerated, you can update the properties of MPP text display objects using the Edit Properties form.

CCR 3565: Offset subpath does not turn small inside jog correctly

Description: If the jog on a path or subpath is short, the path or subpath might turn in the wrong direction.

Solution: There is no workaround available.

Known Problems and Solutions in Virtuoso RF

This document describes the known issues with Virtuoso® RF Solution and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Topics

Introduction to Virtuoso RF Solution

CCR 2814557: Help forms are not working for many forms in Virtuoso Multi-Technology Solution

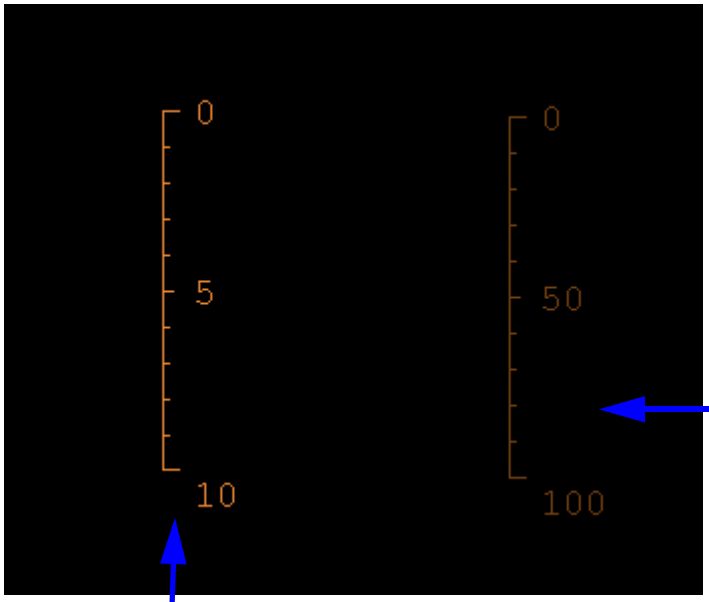
Description: Form Help buttons for tags, such as vrfCreateConnectivity, vrfSaveLayoutConnectivity, vrfSaveSchConnectivity, and vrfConfigureModuleStack, and all forms related to the Bump Management feature have a number appended at the end. The number changes randomly in run time and the associated help page never opens correctly.

Solution: Search for the desired topics in the Doc Assistant directly to see the related information.

CCR 2353478: Rulers are scaled in Edit in Concert mode by the ratio of DBUPerUUs

Description: In a package cellview, when using the Edit in Concert mode, rulers displayed in the background layout cellview get scaled up or down by the ratio of their DBUPerUU and package DBUPerUU if the rulers were created in the layout cellview.

For example, when you create a ruler of size 10 in a layout cellview, it might get displayed as size 100 in the background layout cellview in the package layout.



Solution: There is no workaround available.

CCR 2147217: Stacked designs are not supported during import or export from Allegro

Description: There is an interoperability issue between Virtuoso and Allegro for Package-on-Package designs.

Solution: This issue has been documented for your information only. There is currently no workaround available.

CCR 1917125: DieExport: Not able to perform undo once die export is completed

Description: If you open two cellviews in Virtuoso and make changes in both of them. Thereafter, saving one of the cellview would not allow the undo action for the other cellview as well. In export die feature, a new cellview is generated and saved. However due to this, the undo action stops working for the existing view. As part of exporting the die, the new cellviews

are created and saved, and therefore, the undo action does not work for the edits made in other cellviews that are open in the same virtuoso session.

Solution: This issue has been documented for your information only. There is currently no workaround available.

Known Problems and Solutions in Virtuoso Schematic Editor

This document describes the known issues with Virtuoso® Schematic Editor L and Virtuoso® Schematic Editor XL and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

This document is divided into the following sections:

- Virtuoso Schematic Editor L: Describes CCRs that impact both Virtuoso Schematic Editor L and Virtuoso Schematic Editor XL (unless otherwise stated).
- Virtuoso Schematic Editor XL: Describes CCRs that impact only Virtuoso Schematic Editor XL.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Getting Started with Virtuoso Schematic Editor

Virtuoso Schematic Editor SKILL Functions

Virtuoso Schematic Editor L

CCR 1316417: VSE loses CDF parameters entered in a secondary form raised inside Edit Object Properties

Description: When you select a component and open VSE's Edit Object Properties form, one of the component's CDF parameters shows an additional form that lets you enter more detailed parameter information. However, when you complete and save this form, VSE ignores the values you entered. The values are not applied for the selected instance and do not show when you open the secondary form.

VSE's Edit Object Properties form uses its own distinct copy of the CDF data for a selected object, rather than sharing the same data as the rest of the Virtuoso forms. This prevents other forms from impacting the VSE form data when you use both the forms simultaneously.

Solution: If you are able to edit the CDF parameters for your PDK, consider removing any `"?dontBlock t"` arguments from the forms raised by your parameter callbacks. These arguments break the link between the callback form and the main Edit Object Properties form. However, do check that your forms still work as intended. Alternatively, you can have Edit Object Properties use the same CDF data as the rest of the system, by setting this variable in your `.cdsenv`:

```
schematic editPropProtectCDF boolean nil
```

CCR 1292974: Annotations are not displayed if the value of `opParamExprList` is modified in the user CDF and you do not recreate the netlist

Description: If you modify the value of `opParamExprList` in the *Simulation Information* tab of the *Edit CDF* form for a cell, and choose the *Run* option, the annotations for the selected cell are not displayed in the schematic.

Solution: To view the annotation for the selected cell, recreate the netlist by selecting *Simulation –Netlist – Recreate* before selecting the *Simulation – Run* or *Simulation – Netlist and Run* option in the simulation window.

CCR 1245576: Modifying a waveform using SKILL affects other waveforms

Description: If you modify the existing waveform using a SKILL function, then the data of the other waveforms also gets affected.

In the following example, the `x` value of the `i0` instance can be `0`, `0.5`, `1`, and `100`. However, this also changes the `x` value of instance `i1`.

```
openResults("./CCR_TESTCASE/schematic/psf")
selectResult('dcOp)
i0 = i("/V0/PLUS" ?result 'dcOp)
i1 = i("/V1/PLUS" ?result 'dcOp)
famAddValue(i0 100 100)
```

In the following example, the value of the last `x-vector` is changed to 10.

```
wave = getData("1" ?result 'tran ?resultsDir "nand2_ring.raw")
xvec = drGetWaveformXVec(wave) ;; changes the value of the last x-vector element
to 10
drSetElem(xvec drVectorLength(xvec)-1 10)
```

However, the change in `x-vector` also reflects in other waveforms that are retrieved for transient analysis using the `nand2_ring.raw` database. For example,

```
wave2 = getData("2" ?result 'tran ?resultsDir "nand2_ring.raw")
```

Solution: The data reader optimizes memory usage for waveforms, and therefore, sharing of waveform vectors for raw data is an inherent feature.

If the waveform sharing is required, a `copy()` function can be written in SKILL to make a duplicate copy of the waveform. You can write the `copy()` function using the following SKILL functions:

- `sweepNames()`
- `sweepValues()`
- `famValue()`
- `drGetElem()`
- `drSetElem()`
- `drCreateVec()`

CCR 1235337: Want to turn off annotation balloons on schematic on startup of Virtuoso

Description: By default, the annotation balloons are not visible on the schematic. If you saved the annotation settings in the previous simulation run with annotation balloons visible on the schematic, the status of the balloon visibility is also saved in the `.cadence/dfII/annotationSetups/balloonSetup.ini` file.

Solution: If you do not want the annotation balloons to appear, perform any one of the following actions:

- Delete the `balloonSetup.ini` file before launching Virtuoso.
- The first line in the `balloonSetup.ini` file is `annBalloonVisibleField=t`. Reset the value of `annBalloonVisibleField` to `nil`.

CCR 1207859: Drawing a wire from an instance pin goes to the right rather than up as expected.

Description: For those viewing in ASCII, when you draw a wire from a pin located at the top right of an instance. In some cases, the wire leaves the right side of the pin:

```
        |
   o ---+
XXX
XXX
XXX
```

However, you might expect see a different behavior, such as

In this case, go vertically first when in Draw Mode : Route and Route and Method : Full.

```
   +---
   |
   |
   o
XXX
XXX
XXX
```

VSE routes a wire away from the edge of the instance on which the pin lies. Specifically, it's the bBox edge closest to the pin's center point.

But, a pin located exactly on a corner of the instance's bBox touches two edges, for example, top and right in the above examples. In this ambiguous case, the preferred wiring direction is left and right not top and bottom, therefore, you observe the behavior as mentioned here.

Solution: Adjust the instance bBox so that pins lie on a definite edge, instead of the corner. However, this may not be an option if you are unable to edit your device symbols.

CCR 1172191: Pinned annotation balloons do not scale with the canvas when zooming.

Description: Currently, pinned annotation balloons stay the same size when zooming in or out of the schematic.

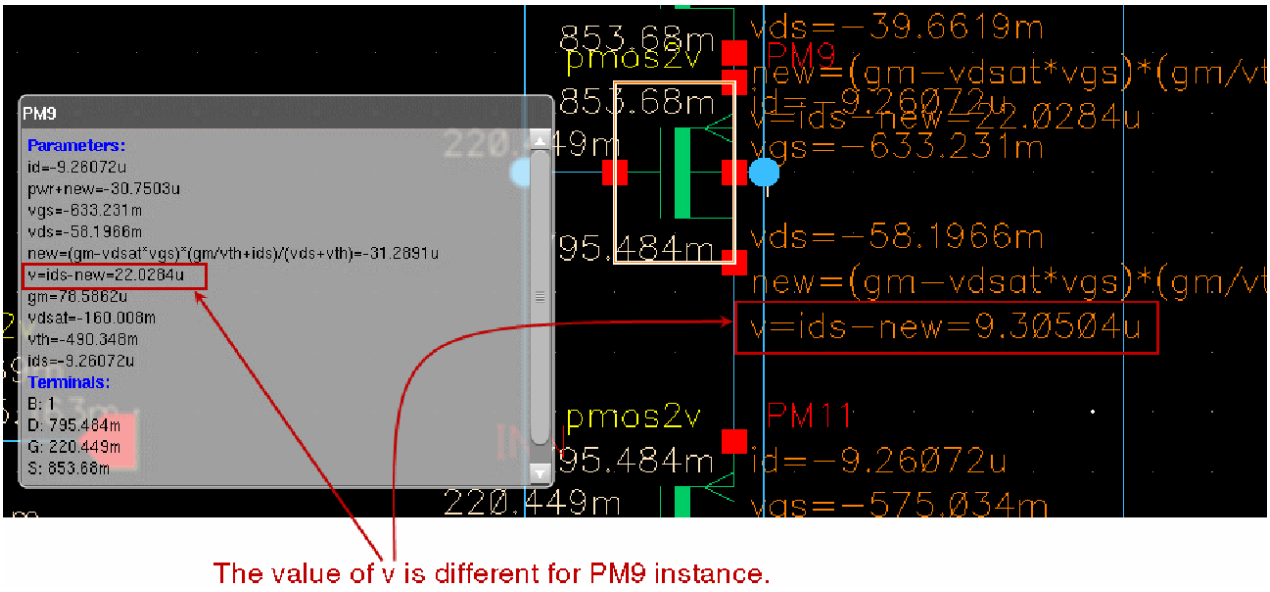
Solution: There is no workaround available.

CCR 1139057: Incorrect value of expression is displayed on canvas

Description: The value of an expression annotated on the annotation balloons is different from that on the schematic canvas. In this example the following expressions are created for Transient Operating points.

```
new=(gm-vdsat*vgs)*(gm/vth+ids)/(vds+vth)
v=ids-new
```

The values of these expressions are evaluated correctly on the annotation balloons, but not on the schematic canvas.



In the above example, for the `PM9` instance, the value of `v` is `22.0284u` on the annotation balloon, however, on the canvas, the value of `v` is `9.30504`.

Solution: The correct value of the expression is not evaluated on the schematic canvas because the variables `new` and `v` are defined as global SKILL variables. The values of expressions evaluated on the schematic canvas depend upon the order in which the Graphic Editor evaluates the label values. To avoid this issue ensure that different variable names are used for different instances, that is, `v1` and `new1` for `PM1`, `v2` and `new2` for `PM2`, and so on.

CCR 1041966: Exclude Inherited Connection Pins field of Symbol Generation Options

Description:

Lib: ether
Cell: try
View: schematic

When you create a symbol using *Create CellView from CellView*, a symbol view is generated. However, you get the following warning message while running *Check > Current Cellview..* or *Check > Cross View...* command in the symbol view.

Warning: `Port "G" in the portOrder property not found in "try symbol".`

The warning is generated because the *Create > CellView > From CellView...* command adds a *portOrder* property to the symbol view during the symbol creation. This has the pin *G* mentioned in the portOrder. Now, you should be able to take this pin name out of the *portOrder* property, but it is not recommended to do so, because during implicit netlisting, the ports are printed in the order specified by this property.

Solution: There is no workaround available.

CCR 992203: Replace function attempts to revert changes on exit

Description: In *Edit->Replace*, there is no *OK* or *Done* button to allow you to save your changes when renaming cells or instances. For example, when you fill out the *Search* and *Replace* fields and press *Apply*, VSE finds the matches, and then, *Replace All* replaces instances in the schematics. It auto-checks out the schematic.

Solution: The only way to stop canceling the checkout is to press the *Save Changes* check box on the replace form.

CCR 893019: Remastering an instance adds the additional CDF parameters of the previous master as user properties in the new instance

Description: The master of an instance in a design can have modified CDF parameters saved as user-defined properties. If you remaster such an instance using the Schematic Replace form, the new instance contains the additional CDF parameters of the previous master as user-defined properties. Such user-defined properties in the new instance can cause issues while performing tasks, such as generating the netlist.

Solution: Manually remove the additional user-defined properties that were added to the new instance.

CCR 799319: Indicator of instance is different between descended hierarchy and returned hierarchy.

Description: The *Navigator* assistant shows the instance as a block to start with as there are instances in the cellview. However, when descending into the cellview, the masters of

these instances are bound and identified as ignored instances. At this point, the HED elaborator modifies the type of instance from block to leaf to reflect that there are no valid instances in the cellview.

Solution: This behavior is by design to limit the amount of bound masters in order to provide performance enhancements in *Navigator*. If a valid, not ignored, design instance is placed in the cellview the problem is not reproducible. This is an issue only in partially completed designs.

CCR 661800: Labels in title blocks of sheet borders can not be edited with on-canvas editing

Description: Sheet border title blocks with labels can not currently be edited with on-canvas editing.

Solution: Use *Edit – Sheet Title* to access and edit the labels in title blocks.

CCR 529150: Expanding a directory in the Navigator takes an excessive amount of time for large designs

Description: It can be time consuming to expand a large design, or a design with multiple hierarchy levels, in the *Navigator*.

Solution: If the design is large, a Descend Edit operation, or the use of another assistant such as the Search assistant, may be preferable to find particular objects in a sub-level.

Note: In the *Navigator*, you have the option to cancel directory expansion using `CTRC+C`.

CCR 459242: When two instances of the same sub-cells are open in tabs, selection in the canvas is not unique to the current instance

Description: When a top cell has multiple instances of the same cell, and you want to set a constraint at the top level for design objects in the sub-cell, selection in the canvas can create an unexpected number of selections of that instance member.

The number of selected objects depends on the number of windows that are open for the sub-cell as an instance sub-cell.

Solution: To set a constraint for design objects located in sub-cell instances, the selection must be done in the *Navigator* rather than the canvas.

CCR 331210: Hierarchical selection notification missing for the first bus

Description: Due to a lack of schematic extraction, hierarchical selection can fail to notify the selection of the first of two selected buses (or ordinary nets). For example, if you:

1. Create two narrow wires in a new Schematics XL cellview.
2. Create two net names, for example: `a<0:5>` and `x<4:1>`.
3. Assign the net names to the wires.
4. Select `a<0:5>`.
5. Shift select `x<4:1>`.

Note: Notice that the *Navigator* does not show selection of the first bus.

The CIW will therefore only display notification of the second, `x<4:1>`, object selection. As a consequence of this only one bus form will be displayed.

Solution: There is no workaround available.

CCR 322217: Limitation in extraction behavior on iterated instances

Description: The extractor can produce variable results as final design connectivity, obtained after running the extractor, is not only dependent upon the design geometry (that is, the placement of instances, wires, wire labels, and so on), but also upon current connectivity.

Note: Although results may vary depending upon geometry and final design connectivity, the connectivity will always be logically correct.

Solution: There is no workaround available.

CCR 287619: Delete net does not update Navigator

Description: You must perform a *File – Check and Save* so that the Navigator will be consistent with the canvas if nets/pins are drawn, renamed, deleted, and so on.

This is necessary because of the way nets and pins are currently created in a schematic.

Solution: There is no workaround available.

CCR 256810: schHiCreateWire does not work in Edit In Place

Description: When editing in place, the create wire command, in default *Draw Mode = Route* and *Route Method = Full* mode, does not take account of the edit in place coordinates. Instead, it will use the top level coordinates and snap in incorrect locations.

Solution: There is no workaround available.

CCR 217698: Problem creating netSet property on instance with terminal as inherited connection

Description: There is a current limitation with inherited connections where iterated instances, for example `x<0:3>`, cannot be used as intended.

This is because of a limitation in the CDBA and OpenAccess 2.0 databases, in that there are no placeholders to store property information on a per-bit instance basis.

Although you can have `netSet` on an iterated instance, care must be taken as it will overwrite ALL bits identically.

Note: This impacts all DFII applications, and is not exclusive to inherited connections. The issue originates from the limitation of not being able to have `bitInstance` property values.

Solution: The workaround is to create multiple instances, for example, `x<0>`, `x<1>`, `x<2>`, and `x<3>` rather than a single iterated instance named, in this case, `x<0:3>`.

CCR 206260: Schematic route flight shorts nets

Description: There can be a problem with the schematic router where shorts are created between nets when the route flight line command is used.

Solution: As a workaround, you should perform the following:

1. Define a SKILL procedure as follows:

```
procedure( CCSrouteFlightLines()
 let((cv f_lines)
   cv = geGetEditCellView()
   f_lines = setof(x cv~>shapes x~>lpp == list("wire" "flight"))

   foreach(line f_lines
     geSelectFig(line)
     schHiRouteFlightLine()
   )
   t
  )
)
```

2. Call this newly created function, for the cellview in question, and it will perform the routing of all flight lines, one by one.

CCR 196645: Instance renumbering overflows when numeric entry >=2e10

Description: If you attempt to enter a number >=2e10 in the “Start resulting sequence with index” field, of the Renumber Instances form (first select Design – Renumber Instances), the schematic editor will encounter problems due to a numeric overflow.

This overflow causes further numeric entries in the index field to no longer be accepted, and for the schematic editor to continue to display solely on those components of the resultant overflow.

Solution: This problem occurs because the renumber instances command uses *signed*, rather than *unsigned*, integers when generating the index. **The field entry limit here is (2^31 - 1).**

A negative number can also be entered in the “Start resulting sequence index” field, and this will cause instances to be renumbered as appropriate. Note however, that instances with negative indices will not be renamed by subsequent applications of renumber instances.

Virtuoso Schematic Editor XL

CCR 2746849: Navigator to display the sets with the probed and selected objects

Description: In Virtuoso IC6.1.8 and ICADVM20.1, a workaround was provided to establish a way to see selected and probed objects in a self-contained manner in the Navigator assistant using SKILL customizations.

Solution: Starting with Virtuoso IC23.1 ISR1, probed and selected objects are displayed in the Navigator assistant by default. Cadence recommends not using the previous workaround any longer.

CCR 941270: CPF parser fails and gives the `tclsh: not found` error

Description: CPF parser fails on the Solaris machine and gives the `tclsh: not found` error.

Solution: Before starting Virtuoso, make sure that tclsh is installed in the path. It is recommended to use tclsh 8.4 or a higher version.

CCR 849686: Voltage range step values for level shifter cells are not supported in CPF import and export

Description: Currently, Virtuoso does not support step values for the `-input_voltage_range` and `-output_voltage_range` options of the `define_level_shifter_cell` CPF command.

Solution: There is no workaround available. However, you can add the step values manually in the CPF file. The issue will be resolved in a future release.

Known Problems and Solutions in Virtuoso Symbolic Placement of Devices

This document describes the known issues with Virtuoso® Symbolic Placement of Devices and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Introduction to Symbolic Placement of Devices

CCR 1554013 User-defined abutment not preserved in certain SPD commands

Description: Existing user-defined abutments are not preserved when Split Finger, Permute, Generate Chained Device, Stack, Edit Stack, and Check or Update commands are used.

Solution: To work around the issue, select the incorrectly abutted devices, unabut and reabut them using the correct user-defined abutment.

Known Problems and Solutions in Virtuoso SystemVerilog Netlister

This document describes the known issues with Virtuoso® SystemVerilog Netlister and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Topics

Introduction to Virtuoso SystemVerilog Netlister

CCR 2186855: SystemVerilog Netlister generates the xrunArgs file that causes an xrun error when used with -libmap and -v or -y.

Description: SystemVerilog Netlister generates the `lib.map` file which can be used with `xrun` and the `-libmap` command-line option. This file specifies the mapping relationship of source files and libraries. When you select the *Create binding files for xrun only* option, it creates binding files that are recognized by `xrun` and are compatible with the command-line options `-v` and `-y`. When you use the `xrunArgs` file, generated by SystemVerilog Netlister, with the `-v` or `-y` option and run `xrun`, it shows the following error:

```
xrun: *E,VCXOPT: -v/-y cannot be used with -libmap option.
```

Solution: Select both the *Create argument file* and *Create binding files for xrun only* options in the System Verilog Netlister Options form. When selected, these options create an alternate set of binding files that let you use `xrun -v` or `xrun -y` without the above error.

Known Problems and Solutions in Virtuoso Technology Database Checker

This document describes the known issues with Virtuoso® Technology Database Checker (TechDB Checker) and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Using the Technology Database Checker

CCR 1034154: When the main TechDB Checker form is closed, other TechDB Checker forms do not close automatically

Description: When the main TechDB Checker form is closed, other open TechDB Checker forms do not close automatically.

Solution: Close all open TechDB Checker forms manually.

CCR 993600: TechDB Checker does not check the orderedSpacing constraints

Description: Currently, TechDB Checker does not report the missing or found status for the `orderedSpacing` constraints. It only shows `spacing` and `spacingTable` constraints.

Solution: Check the `orderedSpacing` constraints in the technology file dump of the technology database.

Known Problems and Solutions in Virtuoso Visualization and Analysis XL

This document describes the known issues with Virtuoso® Visualization and Analysis XL and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Overview of Virtuoso Visualization and Analysis XL

Virtuoso Visualization and Analysis XL SKILL Functions

CCR 1943189: Saving histogram plot as an image is not working

Description: When you use the SKILL command to plot histograms and save the graph image, the saved image is empty.

Solution: Add a dummy command before plotting the histograms. For example:

```
winId=newWindow()
awvDisplayDate(winId)
waveString = "plot2"
dummy=histo(myWave bins 0 100) ; <== THIS WAS ADDED TO FIX MISSING HISTOGRAM
FROM SAVED WAVEFORM
histoPlot = histogram2D(myWave bins "standard" t t)
awvPlotWaveform(currentWindow() list(histoPlot) ?expr list(waveString))
currentWindow(winId)
addSubwindowTitle("Now I have the Histogram")
saveGraphImage(?window winId ?fileName "myhist" ?exactCopy t
?saveAllSubwindows t ?saveEachSubwindowSeparately nil )
```

CCR 1629658: The floatFormat argument in getAsciiWave function does not work for vcsv, csv, and matlab files

Description: When you specify the `floatFormat` argument in the `getAsciiWave` function for files of type `vcsv`, `cvs` and `matlab`, the expression evaluation fails and error messages are displayed. This argument works fine for files of other data formats.

Solution: Currently, a workaround is not available.

CCR 1310980: When the global variables have dependencies, plots may not show a continuous line

Description: If you a global variable is specified in the setup, for example, `T=temperature+10`, and temperature is swept `0:10:20`. Now, if you run a simulation, expressions may be plotted as scalars rather than continuous plots.

Solution: Specify the global variable in `VAR` syntax, for example, `T=temperature+10` should change to `T=VAR("temperature")+10`.

CCR 1046186: An extra column appears in legend while running a simple simulation in ADE.

Description: When running a single-point simulation, the trace legend displays an extra column for an arbitrary variable or parameter specified in the setup.

Solution: To hide this column, right-click the headers in the legend and the de-select the column name.

CCR 1205158: Cannot drag zeroes to graph with poles (or vice versa) but can append

Description: Zeroes cannot be dragged from Results Browser to a graph that has poles. Also, poles cannot be dragged to a graph that already has zeroes.

Solution: As a workaround, use the RMB context-sensitive menu to append poles and zeros to a graph.

CCR 1187214: When multiple vertical markers are dragged, the labels do not update correctly

Description: When multiple vertical markers are dragged, the labels do not update correctly. If there is any delta marker between these vertical markers that are being dragged, the labels for delta markers also do not update correctly.

Solution: As a workaround, drag only one vertical marker at a time.

CCR 1174994: Edge marker changes back to 20/80 threshold if moved to a short edge

Description: If an edge is short and does not cover the points required by the threshold setting of an edge marker, then the marker automatically switches to a smaller threshold, which is from 10-90 to 20-80.

Solution: Marker will skip the edge if it is short and does not cover the points required by the threshold set in the edge settings.

CCR 955927: Spec markers are not visible in the standalone mode of Virtuoso Visualization and Analysis XL

Description: When you run Virtuoso Visualization and Analysis XL in the standalone mode, spec markers are not visible.

Solution: Currently, a workaround is not available.

CCR 955915: Spec markers cannot be edited in graphs

Description: Unlike other markers, spec markers cannot be edited or moved in graphs.

Solution: Currently, a workaround is not available.

CCR 953306: Pattern of spec markers cannot be changed

Description: Currently, you cannot edit spec marker properties to change the patterns used to show pass or fail region.

Solution: Currently, a workaround is not available.

CCR 945675: If multiple outputs are plotted in a single graph, pass and fail labels of spec markers are overlapped

Description: If multiple outputs are plotted in a single graph, pass and fail labels of spec markers are overlapped. As a result, labels cannot be read properly.

Solution: You can move outputs either to different strips in the same graph or to different sub windows and make sure that only single spec marker label is displayed in one strip or subwindow.

CCR 899186: Selecting multiple horizontal and vertical markers with Ctrl and then trying to drag the markers while holding Ctrl only updates one marker

Description: When you select multiple horizontal and vertical markers by using the `Ctrl` key and the mouse pointer, and if you try to drag these markers while holding the `Ctrl` key, only one marker is updated that is dragged by the mouse pointer.

Solution: Currently, a workaround is not available.

CCR 858422: Labels are not cleared when graph is cleaned after you press x on last window or tab

Description: Labels are not cleared when graph is cleaned after you press x on last window or tab. These labels become non-movable.

Solution: Plotting a trace to this blank window allows the labels to be moved or deleted.

CCR 852379: No context menus for reference and point markers on the circular plots when you right-click

Description: No context menus are available for reference and point markers on the circular plots when you right-click

Solution: Select the marker and open the properties form to change the properties.

CCR 830951: Cannot add Point Marker in Smith chart from the marker tool box

Description: Cannot add Point Marker in Smith chart from the marker tool box

Solution: Add the point marker by using the bindkey `M`.

CCR 829280: Vertical and Horizontal Marker snapping do not work correctly in case of traces with non-monotonic x-vector

Description: For YvsY data, where the x-vector is non-monotonic, the snapping to next data point for horizontal and vertical marker do not work correctly

Solution: Currently, a workaround is not available.

CCR 820327: The order of bits do not follow the selection order when you use Shift in the analog-to-digital conversion form

Description: The selection order is from top to bottom when you press `Shift` while selecting, irrespective of the first and the last selection.

Solution: The following two workaround are available:

- The selection follows the order when you press `Ctrl` while selecting.
- For analog-to-digital, digital-to-analog forms, use the built-in alphabetical up and down sorting or sort using drag and drop.

CCR 814677: On spectral plots, cannot open trace property form by double-clicking a trace in the graph

Description: You cannot open trace property form in spectral plots by double-clicking a trace in the graph.

Solution: Do one of the following to open the trace property form:

- Select trace and press `Q`
- Double-click the trace name in trace legend
- Double-click the top of spectral bars in graph

CCR 803675: Signal or expression disappears from Eye Diagram toolbox when it is undocked from graph window

Description: Signals or expressions disappear from the Eye Diagram toolbox when they are undocked from graph window

Solution: Reselect signal and it will appear in the toolbox.

CCR 798199: limitations of undo with new window creation

Description: The following steps to create a new window cannot be undone

1. Renaming a window or tab by double clicking the tab name.

2. Moving or copying a trace to a new window. When you move a trace and click undo the from the window you used moved the trace, the trace is again displayed in this window. However, the new window remains the same.

3. Closing a window or tab by clicking the close button available on the window tab.

Solution: Currently, a workaround is not available.

CCR 794895: Snap XY value edit fields are not in synchronization with the snap and marker properties form

Description: Actual snapping criterion are in synchronization with snap and marker properties form. However, XY value edit fields are not in synchronization.

Solution: Set XY values manually.

CCR 792308: Graph Properties form does not open when you double-click a graph

Description: The *Graph Properties* form does not open when you double-click a graph.

Solution: Do one of the following steps to open the *Graph Properties* form:

- Click *Edit – Properties* (with nothing selected)
- *Graph – Properties*
- Click a blank space in the graph and press Q

- Press Shift Q any time
- Double-click a blank space in the trace legend area

CCR 780325: Virtuoso Visualization and Analysis XL creates a weird table when range and log is used

Description: Virtuoso Visualization and Analysis XL creates a weird table when range and log is used

Solution: Currently, a workaround is not available.

CCR 730406: Virtuoso Visualization and Analysis XL MDL mode is not working on Solaris86

Description: Virtuoso Visualization and Analysis XL MDL mode is not working on Solaris86.

Solution: Use Virtuoso Visualization and Analysis XL in the SKILL mode.

CCR 690563: phaseNoise does not distinguish between in- and out-of-context result directories

Description: The *phaseNoise* function does not include the results directory argument. It only creates an expression, such as `phaseNoise(1 "pss-fd.pss" ?result "pnoise-pnoise")`, which does not distinguish between in- and out-of-context result directories. This is confusing for the users who are not aware of in-context result directory.

Solution: Before plotting the new phaseNoise, right-click the new results directory in the Results Browser and choose *Set Context*. Note that the *In Context Results DB* field in the Calculator reflects this change. Now if you plot phaseNoise, it will be from the new results directory.

CCR 1767396: Template not found error when using a Calculator function

Description: When a Calculator function is removed from the Function Panel of the Calculator, but still added as a favorite function, the following error is displayed when you try to use the function:

```
No template file found for: b1f
```

Workaround: Remove the function from the favorite functions list by right-clicking the function name and choosing *Delete*.

CCR 1237103: Cannot delete expressions from Expression Editor that include a blank name

Description: Consider a scenario where you export an expression from the Expression Editor to a file and edit the file to rename the expression to a blank name, for example, from `E0` to `""`. Now, when you reload this file in Calculator, the expression is displayed with a blank name and it cannot be deleted.

Solution: As a workaround, double click the *Name* field in the Expression Editor, specify a name for the expression and then delete the expression.

CCR 826109: For reverse DC sweeps or hysteresis sweeps, the clip Calculator function does not work correctly

Description: For reverse DC sweeps or hysteresis sweeps, the `clip` calculator function does not work correctly.

Solution: Currently, a workaround is not available.

CCR 845432: Signals pane in Results Browser does not refresh when data gets changed during simulation

Description: Signals pane in Results Browser does not refresh when data gets changed during the simulation. An orange icon appears which signifies that data has changed and you again have to click results directory and browse to get the updated results.

Solution: Currently, a workaround is not available.

CCR 504581: Virtuoso Visualization and Analysis XL windows that are not bound to ADE session are not saved with the ADE state

Description: ADE maintains a list of graph windows that are created for a session. When an ADE state is saved, only the windows that are opened in the current session are saved.

The graph windows that are created through ADE, such as autoplot, direct plot are included in the session graph window list. However, the graph windows created from Results Browser

or Calculator are not part of the session graph window list and hence are not saved as part of the ADE state.

Solution: Currently, a workaround is not available.

Known Problems and Solutions in Virtuoso Width Spacing Patterns

This document describes the known issues with Virtuoso® Width Spacing Patterns and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

Virtuoso Width Spacing Patterns

CCR 2171967: Unable to open the Track Pattern Task Assistant

Description: Unable to open the Track Pattern Task Assistant by clicking the button on the Track Pattern Assistant toolbar.

Solution: As a workaround, access the Cadence Help pages for Track Pattern Assistant by clicking on the title bar of the assistant.

CCR 1482939: Repeat mode for a WSPPDef shows Stepped but the allowedRepeatMode for the active pattern is flippedOnly

Description: In the Track Pattern assistant Design Settings form, the Repeat Mode for a WSSPDef is Stepped but the allowedRepeatMode for the active WSP is set to flippedOnly in the technology file. For backward compatibility, the default repeat mode for both the global grid and regions is set to Stepped, regardless of the allowedRepeatMode setting for the active WSP. This is needed because some designs might not have the default repeat mode set or an existing pattern from an older release has been changed to flippedOnly. When the allowedRepeatMode for the active WSP is set to flippedOnly in the technology file and the Repeat Mode is Stepped, the Repeat Mode drop-down menu in the Track Pattern assistant Design Settings form will show Stepped with an asterisk (*).

Solution: As a workaround, choose an appropriate Repeat Mode from the drop-down menu for the WSSPDef in the Track Pattern assistant Design Settings form.

Known Problems and Solutions in Virtuoso Space-based Router

There are no known issues in Virtuoso Space-based Router in the IC23.1 base release.

Related Topics

- *Virtuoso Space-based Router User Guide*
- *Virtuoso Layout Suite SKILL Reference*
- *Virtuoso Technology Data ASCII Files Reference*
- *Virtuoso Width Spacing Patterns User Guide*

Known Problems and Solutions in Cadence Library Path Editor

There are no known issues in Cadence Library Path Editor in the IC23.1 base release.

Related Topics

- *Introduction to Library Path Editor*

Known Problems and Solutions in Connectivity-to-Schematic

There are no known issues in Connectivity-to-Schematic in the IC23.1 base release.

Related Topics

- *Connectivity-to-Schematic User Guide*

Known Problems and Solutions in Virtuoso Unified Custom Constraints

There are no known issues in Virtuoso Unified Custom Constraints in the base IC23.1 release.

Related Topics

- *Virtuoso Unified Custom Constraints Configuration Guide*
- *Virtuoso Unified Custom Constraints Getting Started Guide*
- *Virtuoso Unified Custom Constraints User Guide*
- *Virtuoso Unified Custom Constraints SKILL Reference*

Known Problems and Solutions in Virtuoso Design Intent

There are no known issues in Virtuoso Design Intent in the IC23.1 base and subsequent release.

Related Topics

- *Virtuoso Design Intent User Guide*
- *Virtuoso Schematic Editor User Guide*
- *Virtuoso Layout Suite XL User Guide*

Known Problems and Solutions in Virtuoso Studio Software Licensing and Configuration

There are no known issues in Virtuoso Software Licensing and Configuration in the IC23.1 base release.

Related Topics

- Configuring the Virtuoso Studio

Known Problems and Solutions in Cadence Integrator's Toolkit Database

There are no known issues in Cadence Integrator's Toolkit Database in the IC23.1 base release.

Related Topics

- Introduction to CDBA

Known Problems and Solutions in Open Simulation System

There are no known issues in Open Simulation System in the IC23.1 base release.

Related Topics

- *Open Simulation System Reference*

Known Problems and Solutions in Virtuoso Technology Data

There are no known issues in Virtuoso Technology Data in the IC23.1 base release.

Related Topics

- *Virtuoso Technology Data User Guide*
- *Virtuoso Technology Data Constraints Reference*
- *Virtuoso Technology Data ASCII Files Reference*
- *Virtuoso Technology Data SKILL Reference*

Known Problems and Solutions in Virtuoso Text Editor

There are no known issues in Virtuoso Text Editor in the IC23.1 base release.

Related Topics

- *Virtuoso Text Editor User Guide*

Known Problems and Solutions in Virtuoso Import Tools

There are no known issues in Virtuoso Import Tools in the IC23.1 base release.

Related Topics

- *Virtuoso Import Tools User Guide*

Known Problems and Solutions in Virtuoso Interactive and Assisted Routing

There are no known issues in Virtuoso Interactive and Assisted Routing in the IC23.1 base release.

Related Topics

- *Virtuoso Interactive and Assisted Routing User Guide*

Known Problems and Solutions in Virtuoso VHDL Toolbox

There are no known issues in Virtuoso VHDL Toolbox in the IC23.1 base release.

Related Topics

- *Virtuoso VHDL Toolbox User Guide*

Known Problems and Solutions in Cadence SKILL Language

There are no known issues in Cadence SKILL Language in IC23.1 base release.

Related Topics

- *Cadence SKILL Language User Guide*
- *Cadence SKILL Language Reference*
- *Cadence SKILL Development Reference*
- *Cadence Interprocess Communication SKILL Reference*
- *Cadence SKILL++ Object System Reference*

Known Problems and Solutions in Virtuoso Floorplanner

There are no known issues in Virtuoso Floorplanner in the IC23.1 base release.

Related Topics

- *Virtuoso Floorplanner User Guide*

Known Problems and Solutions in Virtuoso Concurrent Layout

This document describes the known issues with Virtuoso® Concurrent Layout and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Note: Unless otherwise stated, the issues described in this document were identified in IC23.1 or an earlier release. For a list of the issues that were fixed in this release, check the README file at downloads.cadence.com.

Related Topics

- Introduction to Concurrent Layout Editing
- Virtuoso Concurrent Layout Functions
- What's New in Virtuoso Concurrent Layout

Known Problems and Solutions in Simulation Environment

There are no known issues in Simulation Environment in the IC23.1 base release.

Related Topics

- *Simulation Environment Help*

Known Problems and Solutions in Virtuoso Photonics Solution

There are no known issues in Virtuoso Photonics Solution in this release.

Related Topics

- Introduction to the EPDA Framework
- What's New in Virtuoso Photonics Solution

Known Problems and Solutions in Virtuoso Simulation Driven Interactive Routing

There are no known issues in Virtuoso Simulation Driven Interactive Routing in the IC23.1 base release.

Related Topics

- *Virtuoso Simulation Driven Interactive Routing User Guide*

Known Problems and Solutions in Virtuoso Design Planner

There are no known issues in Virtuoso Design Planner in the IC23.1 base release.

Related Topics

- *Virtuoso Design Planner User Guide*

Known Problems and Solutions in Virtuoso Custom Digital Placer

There are no known issues in Virtuoso Custom Digital Placer in the IC23.1 base release.

Related Topics

- *Virtuoso Custom Digital Placer User Guide*

Known Problems and Solutions in Verilog In

There are no known issues in Verilog In in the IC23.1 base release.

Related Topics

- *Verilog In for Virtuoso Design Environment User Guide and Reference*

Known Problems and Solutions in Virtuoso Power Manager

There are no known issues in the Virtuoso® Power Manager in the IC23.1 ISR15 release.

Related Topics

- *Virtuoso Power Manager*

Known Problems and Solutions in Voltus-XFi Custom Power Integrity Solution XL

There are no known issues in Voltus-XFi Custom Power Integrity Solution XL in IC23.1.

This topic describes the known issues with Voltus-XFi Custom Power Integrity Solution XL and suggests the workarounds for these issues. Each issue is identified by a Cadence Change Request (CCR) number.

Related Topics

- *Voltus-XFi Custom Power Integrity Solution XL User Guide*

Known Problems and Solutions in Voltus-Fi Custom Power Integrity Solution XL

There are no known issues in Voltus-Fi Custom Power Integrity Solution XL in IC23.1.

Related Topics

- *Voltus-Fi Custom Power Integrity Solution XL User Guide*