cadence®

Welcome to AMS in ADE

Product Version IC23.1
June 2023

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

Contents

Welcome to AMS in ADE

Welcome to the integration of the Virtuoso® AMS simulator and the Virtuoso® Analog Design Environment (ADE).

This document highlights the following

The integration of the AMS simulator and ADE creates a design environment with the look and feel expected by analog and mixed-signal designers who already use ADE. When using this integration, you can access designs using the same tools you currently use for pure analog and mixed signal designs.

Features of AMS in ADE

The integration of the AMS simulator and ADE has the following features:

- State files from other simulators

  When you use the AMS simulator with ADE, you can load and use any state file that ADE has saved, regardless of the simulator ADE was running when the state file was saved. Other simulators include Virtuoso® Spectre, Virtuoso® APS, and Virtuoso® XPS-MS.

- Available analyses

  Transient, AC, DC, and noise analyses are available when you use the AMS simulator and Spectre or APS solver with ADE. (When you use the AMS simulator with XPS-MS solver, only transient analysis is available.) You can save the DC operating point.

- Outputs

  You can use the ADE *Output* options to *Save All Signals* or to *Select Specific Signals*. You cannot save AC currents.

- Single Button *Netlist and Run*

Netlisted files must be compiled and elaborated before they can be simulated. When you press *Netlist and Run*, a sequence of tools, including the simulator, is invoked.

- Schematics and layout views are translated to Verilog-AMS structural netlists
- Netlists are compiled
- Elaboration runs
- Simulation runs

Each tool produces a separate log file. When any tool fails, the CIW displays a failure message. Look in the log file of the tool for descriptive error messages. The `xrun.log` file utilizes the error message and linked file hyperlinking feature enabling a quicker debug process.

- Full support for ADE display tools

  The integration of the AMS simulator and ADE makes the full set of ADE tools available. In particular, you can examine waveforms with the Virtuoso Visualization and Analysis XL display tools and take advantage of their superior performance for large mixed-signal designs as well as their analog-centric capabilities. You can backannotate DC and transient operating point information to the schematic. You can use ADE data access features such as the Calculator and Results Browser, and the ADE Direct Plot form.

- Full support for Incisive Simvision Debug tools

  The integration of the AMS simulator and ADE makes the full set of Simvision tools available. In particular, you can examine waveforms with the Incisive Simvision display tools and take advantage of their superior capability and performance for large mixed-signal designs as well as Simvision digital-centric debug capabilities. You can run your simulation in Simvison Interactive mode for real-time simulation results and text-based model debugging, such as the Calculator and Results Browser, and the ADE Direct Plot form.

- Support for running the simulator interactively

  This feature launches SimVision and provides you the ability to cross-probe from the schematic.

- OCEAN

  Ocean provides full support for the AMS simulator including several new commands such as `connectRules()`.

- Distributed simulation

  Use network mode for distributed AMS simulations.

- Visual display of signal and cellview domains

  In the schematic window, you can highlight analog nets and digital nets in different colors, with indicators showing the mixed-signal domain and location of automatically inserted connect modules. Visualization of a mixed-signal design can help you tune the design for desired characteristics.

- Graphical support for viewing and manipulating global signals

  You can see the global signals that are extracted from the schematic design after netlisting or add new global signals that are found only in text views.

- Graphical support for disciplines and connect rules

  With the Select Connect Rules form, you can create new disciplines, customize connect rules, and select the connect rules to be used during simulation.

- Support for specifying block-based discipline resolution (BDR)

  You can suggest a discrete discipline to be used for a net, terminal, instance terminal, instance, cell, or library, thereby influencing the connect modules that are used during simulation.

- Graphical support for IE card setup

  The new IE Card Setup GUI enables you to automate the process of creating custom discrete disciplines to enable connect rule/connect module creation and placement specifically improving the usability of AMS multi-supply design configurations. The new IE card setup GUI provides a single place where you can configure your connect rules and connect modules, specify the scope for the built-in ie cards, and set or change the values of important ie card parameters, such as `vsup`, `tr`, `tf`, `rout`, and so on through an intuitive and easy-to-use GUI selection.

- AMS Unified Netlister (AMS UNL)

  AMS UNL provides a comprehensive netlisting and binding framework for the AMS Designer Virtuoso Use Model (AVUM) flow. It enables you to seamlessly migrate from the OSSN-based xrun netlisting flow. It also provides enhanced benefits for the existing OSSN users. AMS UNL natively parses SystemVerilog, VHDL-AMS, and VerilogAMS text-based models. It utilizes the xrun external text referencing capabilities that allow better text integration and seamless migration with xrun digital-based flows. VHDL and VHDL-AMS designs are fully supported in the AMS UNL flow.

- Xcelium xrun integration

  The “one step” xrun digital simulator is available in AMS in ADE. The xrun use model is similar to ncverilog and works seamlessly with the -y/-v/-f options. This approach is more

consistent with the digital use model allowing design information to be more conveniently shared between digital and analog teams.

- AMS/Matlab

AMS/Matlab is officially supported. A Pcell coupler module has been added to analogLib and a utility is available which allows a user to create fixed coupler cells. There are three possible flows that a user can choose from:

  - An ADE integration where the user can launch Matlab from ADE.
  - The use model of starting Matlab and ADE separately is still supported.
  - The user can start Matlab and then can start AMS (via a run script).

For ADE Users: Differences to Expect When Using AMS in ADE

Some significant differences introduced with the AMS integration and ADE are the following.

- AMS Unified Netlister flow (AMS UNL) provides seamless migration from the OSSN-based netlisting flow. AMS UNL will become the single supported flow in ADE once all legacy migrations have taken place by customers. The plan is to deprecate the OSSN flow once this is complete
- Both the *Netlist* command and the *Netlist and Run* command can potentially call several tools to
  - Compile updated text views
  - Netlist updated schematic cellviews
  - As necessary, call xrun to compile, elaborate, and simulate

  Messages in the CIW indicate which tool is running. Each tool writes its own log file.
- The *NCBrowse* utility helps pinpoint issues in the logfiles created during compilation, elaboration and simulation. Use *Simulati*on – *Output Log – LogFile Utility* to run the *NCBrowse* utility.
- By default, AMS in ADE does not save outputs. In the ADE Simulation window, do one of the following
  - Select *Outputs – Save All* to save all output signals.
  - Select *Outputs – To Be Saved – Select On Design* and select specific output signals.

For AMS Users: Differences to Expect When Using AMS in ADE

- ADE state files are used.
  - The ADE default mechanism is used. The `.cdsenv` file provides all ADE defaults.
  - The `ams.env` file is not used. Any defaults you set using the AMS Designer mechanism are not used.

    The `ams.env` file or its variables are not supported for AMS UNL.
- To specify an `hdl.var` file to be used, select *Simulation – Options – Compiler* to open the Compiler Options window. Then, in the *hdl.var file* field, specify the file you want to use. If you do not explicitly specify an `hdl.var` file in this way, then the `hdl.var` file will not be used for your ADE session.

Where to Get More Information

For more information, see the following documents

- Select *Session – FAQ* in the Simulation window to display a list of solutions to frequently asked questions.
- The *Virtuoso Analog Design Environment User Guide*
- The *Virtuoso AMS Simulator User Guide*
- The *Cadence Hierarchy Editor User Guide*
- The *Cadence Application Infrastructure User Guide*