AMS(Analog Mixed Signal - simulator) integration in ADE(AnalogDesignEnvironment)

Pranav Bhushan, CIC Mixed-Signal

1 Introduction

Since years there was a requirement for both analog and digital designers to have a common platform where they can use a set of tools tuned to facilitate the development of mixed-signal designs under a common cockpit known as Analog Design Environment (ADE) or commonly called as Artist.

Most of the Analog designers are familiar and friendly using ADE since years and since they were familiar with the tool so they wanted the AMS simulator to be integrated under the same hood called as AMS in ADE. This way the integration could meet the fancy of the analog designer's graphical environment use coupled with the digital designer's usage to work on synthesis and transformation of their hardware description language (HDL) descriptions into actual circuitry using ADE. The AMS in ADE flow brings these two approaches together, so a designer who prefers to work with text always has up-to-date HDL representations of schematic designs. This dual text and graphical approach is available throughout the flow for both analog and digital designs. For example, to follow an all text approach you might use a text editor to code an HDL module and use the ncvlog, ncelab, and ncsim commands to prepare and simulate the design. To follow a graphical approach, you might use the Virtuoso Schematic Composer editor to create components, use the Hierarchy Editor to define a configuration, and use SimVision to simulate and debug the design interactively. In both cases, you use the SimVision Waveform Viewer to display the simulation results.

The objective of this presentation is for analog and mixed-signal design audience who work with big A/small D and small A/big D designs. For the first time Cadence AMS simulator is integrated into Analog Design Environment (ADE) and is recently released. This functionality was desired by customers since long.

2 AMS in ADE integration

The AMS in ADE integration is a hybrid integration which is based on top of integrating the capabilities provided by classical ADE, AMS Designer (Eden Netlister) and AMS simulator all encompassed into one block. Most of the advantages of the classical ADE stuff like analysis setup, save-states, design variables, outputs selection, batch mode support, advanced analysis support etc are supported in AMS in ADE integration. However, the fast and accurate capability of Eden cell based netlister, the new connect rules setup, cds_globals generation, dual analog solver - Spectre & UltraSim support, dual simulation data reading: PSF & SST2 is added into ADE as new features.

Some of the common features of AMS in ADE integration:

- Next generation, high speed, high performance simulator integration
- Targeted for both Big-A/small-d & Big-D/small-a audience
- Integration keeps the major advantages of HED based use model
- Mixed Signal support - Analog & Digital simulator support
- Mixed Language Support - Verilog AMS & VHDL AMS
- Dual Analog Solver support - Spectre(accurate) & UltraSim(fastSpice)
- Classical ADE support: analysis, save states, save/plot, cross-probing etc.
- Mixed Mode support - Accurate Table model/new front end (SFE) support
- Integration of waveform viewing tools - Wavescan, AWD & SimVision waves.
- Incremental Compilation, Elaboration and run support.
- Post-processing support for dual data format - PSF & SST2
- Mixed-signal Partitioner, Operating Point Back annotation, cross-probing.
- Batch mode scripting simulation support - OCEAN.
- ADE Advance Analysis support - Parameteric Analysis, Distributed Processing, Corners Tool & ADE Optimizer.

3 Technical aspects:

3.1 What was missing?

- There was no direct way for ADE users to use AMS simulator with in dfII, other than to switch over to Eden - AMS Environment which most of the analog designers are not familiar. Since ADE has been market leader since years so this was a long cherised dream for customers as well as PEs to have AMS simulator integrated into ADE.
- The ADE mixed-signal (verimix) solution provided help for analog & digital designs but it had various limitations, performance bottlenecks and lack of full mixed-signal language support (Verilog AMS & VHDL AMS not supported).
- The ADE mixed-signal (verimix) solution was based on top of Spectre simulator and lacked the support of dual-analog solver - Spectre (accurate) and UltraSim(fast and efficient memory utilization) which is now extracted from AMS (ncsim) simulator.

3.2 How is it improved?

Following was done:

- **AMS_IF integration via OASIS Direct technology:**
  The robust and scalable architecture of OASIS technology was used to quickly integrate AMS simulator in ADE. The work done was special in sense that we inherited the customization code/SKILL++ classes etc from Spectre & UltraSim classes. Due to limitations of SKILL++ to do multiple inheritance we inherited the dual class info/code in AMS_IF via one single-inheritance and other via dummy inheritance approach. This way we could inherit the functionality of both Spectre_IF and UltraSim_IF integrations in AMS interface.
- **Integration of Eden Cell Based Netlister:**
  The classical ADE netlisters are view based netlisters, but one of the challenge for AMS_IF was to integrate and use AMS Direct (Eden) cell based netlister in ADE. In

order to do so we used Eden netlister PIs and provided them with hooks to so that fast cell based netlister is integrated into ADE. We also provided with capability for concatenated netlist & control file display. This also involved adding cds_globals and connect-rules support in ADE.

- **Dual Solver support:**
  The dual solver support was provided by providing an option called ‘Solver’ which was capable of exploiting the dual mode of AMS simulator wrt Spectre and UltraSim solvers as shown in Figure 1.The analog control file was added to contain the statements supported by ncsim for both the solvers. The digital save and probe statements were supported via a TCL file.

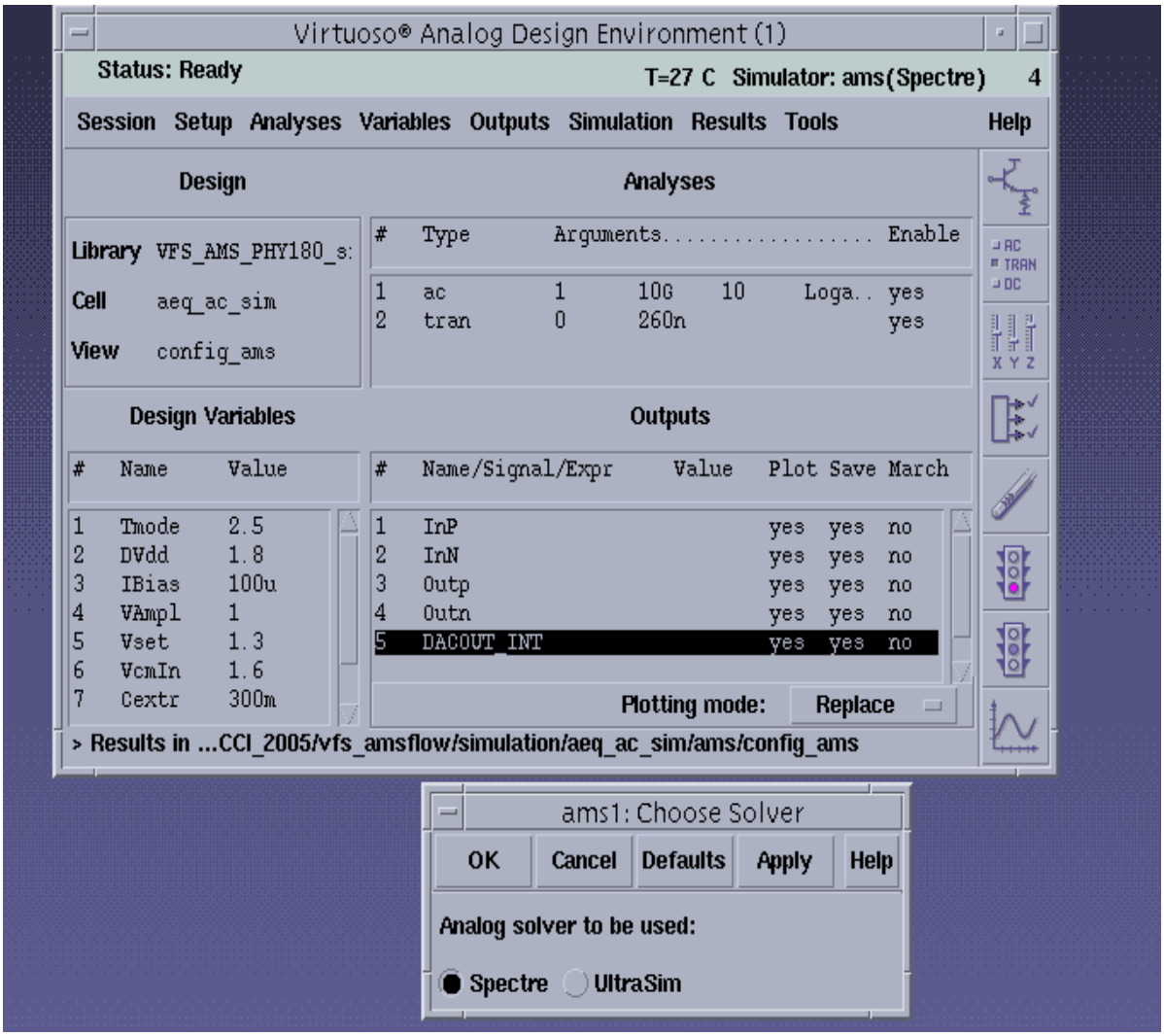

Figure 1: AMS in ADE with dual solver support

- **Display Partition Support:**
  The elaboration was done and partitioner information was generated to highlight the analog, digital and mixed-signal (analog/mixed & digital/mixed) components in design.

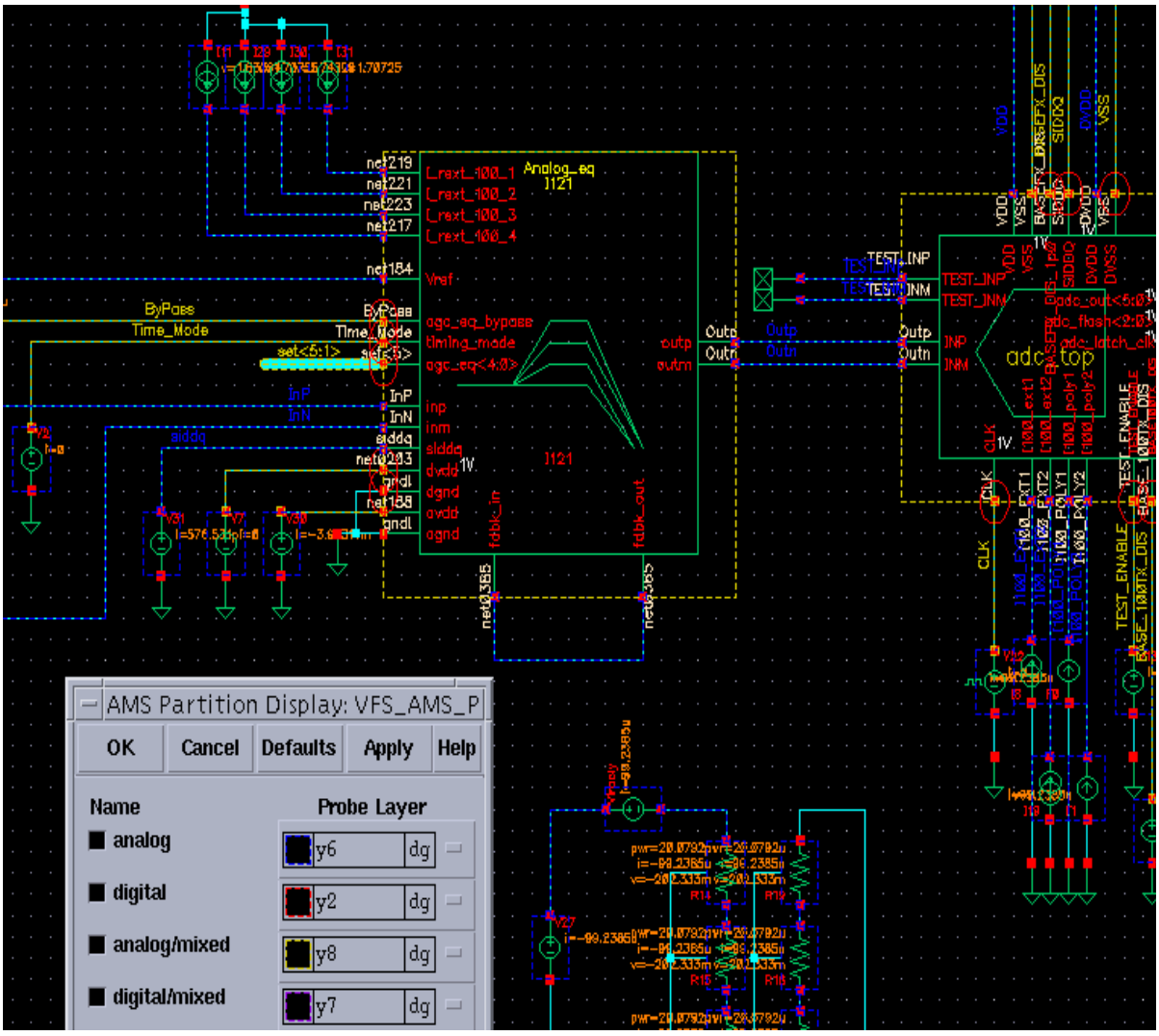

Figure 2: Display Partition Functionality in AMS in ADE (analog in blue, digital in orange, analog/mixed in yellow, digital/mixed in purple)

- **Advance functionality support:**
  ADE advance features like operating point back annotation, label display, cross-probing etc. were added for AMS in ADE. Advance tool support like Parameteric Analysis, Corners Analysis, Distributed Processing & Optimizer support were also provided.

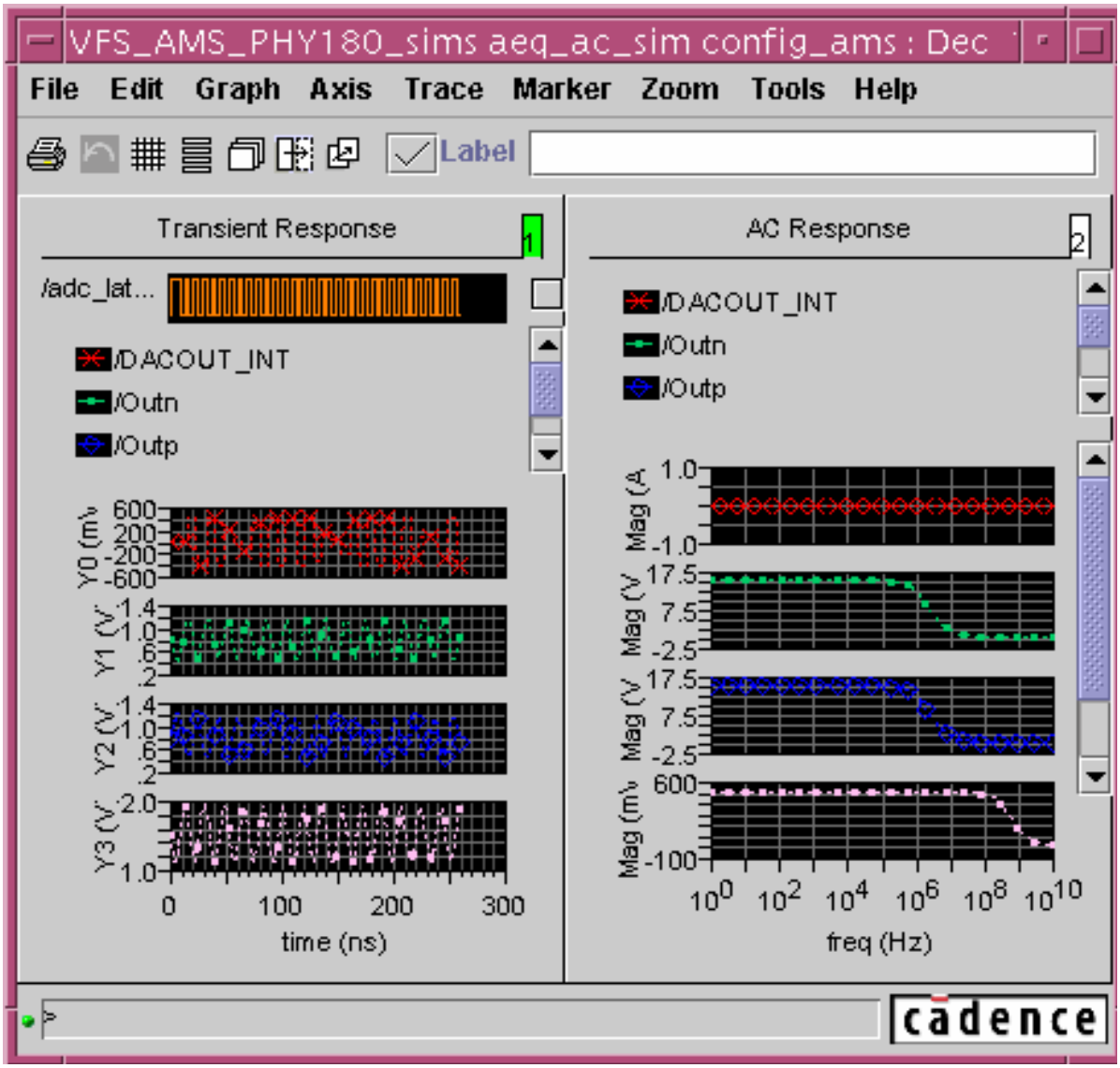

Figure 3: Wavescan Window showing auto-plotting of data

4 Demo Description

- **Design Contents and description.**

The Demo database is a schematic testbench as made up of a programmable, differential, antialiasing filter (AEQ) filtering the front end of a differential, 6bit ADC, then the differential signal is reconstructed by a 6bit DAC to verify proper circuit operation. The design consists of a behavioral model run with about 400 transistors and 400 passive components. It first runs simulation with Analog-Spectre solver and later uses the FastSpice capabilities of UltraSim solver, by flipping the config views to use 4000 transistors and about 3000 passive devices. In the flow we also switch the config view to analog-extracted-rc view adding another 5000 passive components to the configuration. In all around 4000 transistors and 8000 passive components are added and AMS-UltraSim's powerful fast simulation with little loss of accuracy for a quick verification is demonstrated.

- **Steps**
  - Load Design: Schematic AEQ testbench
  - Invoke ADE, setup ‘ams’ simulator, setup ‘AMS-Spectre’ as solver, load ADE state (ac, tran, models, outputs to be saved/plotted, connect modules etc stuff).
  - Create Netlist, Display Netlist and analog control and TCL files.
  - Run Simulation (including compilation, elaboration, simulation of design)
  - Show Display Partition interactive results, OPBA etc stuff.
  - Show interfaces with Wavescan, AWD, SimVision waves, SimVision debugger etc.
  - Flip the solver to ‘AMS-UltraSim’. Re-run Simulation
  - Switch some configs to verilogD and veriloga views, run fastSpice solver.
  - Show incremental compilation/elaboration etc stuff.
  - Show interfaces with Wavescan, AWD, SimVision waves, SimVision debugger etc.
  - Change config in design to use analog-extracted-rc view, show layout, simulate design with AMS-UltraSim to show its fast speed.