32bit MCU Full Chip Verification using AMS Verification Flow (AMSVF)

[1] Freescale Semiconductor, Inc.
[2] NSD, Cadence Design Systems, Inc.

Kiger Zhang[1], Vincent Li[2], Dave.Zbasnik[1], Daniel Liu[2], Stefan Wuensche[2],
Prabal Bhattacharya[2], Shekar Chetput[2] and Martin O'Leary[2]

Session # 1.9

Presented at

cadence designer network

cdn LIVE™

Silicon Valley 2007

Abstract:

The design and verification of a 32bit MCU SoC design requires a powerful and efficient mixed-signal verification tool. This paper introduces the full chip mixed-signal verification flow using Cadence Virtuoso AMS Verification Flow on the design which contained RTL Verilog modules, analog SPICE netlists, and Verilog-A behavior models. After introducing the design project, the paper discusses why the command line AMSVF verification flow was chosen as preferred verification solution, and how 4 design problems could be identified using this verification solution. The paper concludes with the introduction of new AMSVF features which improve the efficiency of full chip verification.

Keywords: Mixed Signal, AMS Designer, verification, AMSVF

I. Background

The DragonFire0 family derivatives are general purpose microcontrollers based on the ColdFire Version 2 processor and manufactured in the HiP7a 0.13um process. In 1994, the innovative ColdFire Microprocessor Family was added to Freescale's Legacy 68K Family tree. This new variable-length RISC 68K Family architecture delivers the aggressive price/performance required by the cost-sensitive embedded market. In striving to meet the needs of the market with this innovative architecture, Freescale evaluated and optimized many high-level source codes from the 68K embedded system customers. Like most RISC processors, the majority of ColdFire processor instructions execute in a single cycle. The variable-length RISC ColdFire architecture gives customers greater flexibility to lower memory and system costs. Because instructions can be 16-, 32- or 48 bits long, code is packed tighter in memory resulting in better code density than traditional 32- and 64-bit RISC machines. The more efficient use of on-chip memory reduces the bus bandwidth and the memory required externally, which results in lower system cost.

Small and inexpensive, the static ColdFire core lowers system cost because it is completely synthesizable and easily integrated with memories, system modules, and peripherals. Because of its portable nature, the ColdFire core is easily targeted to different process technologies, making it attractive as a product for third-party licensing. Freescale is currently developing strategic alliances with other companies.

The DragonFire0 family has the following main features:

- Version 2 ColdFire® Core with EMAC
- Up to 159 Dhrystone 2.1 MIPS @ 166.67 MHz
- 8 Kbytes configurable cache (instruction only, data only, or split instruction/data)
- 128 Kbytes internal SRAM
- Support for booting from SPI-compatible flash, EEPROM, and FRAM devices
- 16 channel DMA controller
- 16- or 32-bit SDR/DDR controller
- USB 2.0 On-the-Go controller
- Liquid crystal display controller with support up to 800 × 600 pixels
- ADC and touchscreen controller
- FlexCAN module
- DMA supported serial peripheral interface (DSPI)
- I2C bus interface
- Synchronous serial interface (SSI)
- Two programmable interrupt controllers (PIT)

The project that will be discussed in this paper is to simulate and verify the whole DragonFire0 SoC design. In this design, a module of “Touch-Panel controller” is created, which can support 4/5/7/8 wire resistance touch panel, and hardware pen-up detection as well.

II. Architecture of the SoC design

In this design, the analog hard core and the digital portion were designed in different site. At the beginning, the analog and digital portions were simulated by SPICE simulator and Verilog simulator separately. At the integration phase, the whole chip must be simulated to verify the functionality.

Fig 1 shows the block diagram of the Touch Panel module. The “VerilogA model” emulates a 4/8 or 5/7 wires touch panel. The testbench sends it a random number to control resistor ratio to indicate the touch position. The “Analog Hard Core” implements S/H, comparator and touch panel bias network function. While the “Verilog RTL code” implements the whole logic function, such as state machine, SAR controller, FIFO controller, etc.

The design poses a challenge to the verification tool, since VerilogA model, SPICE netlist, and Verilog RTL code are used, and the simulator must meet the following all criterias:

a. support mixed language mode (VerilogA, SPICE, and Verilog),
b. freely switch between SPICE netlist and its RTL stub model,
c. re-use the existed stimuli pattern,
d. easy integration into Freescale’s design environment.

AMS Verification Flow (AMSVF), the solution provided by Cadence, was employed, which not only meets above needs well, but also provides powerful verification features.

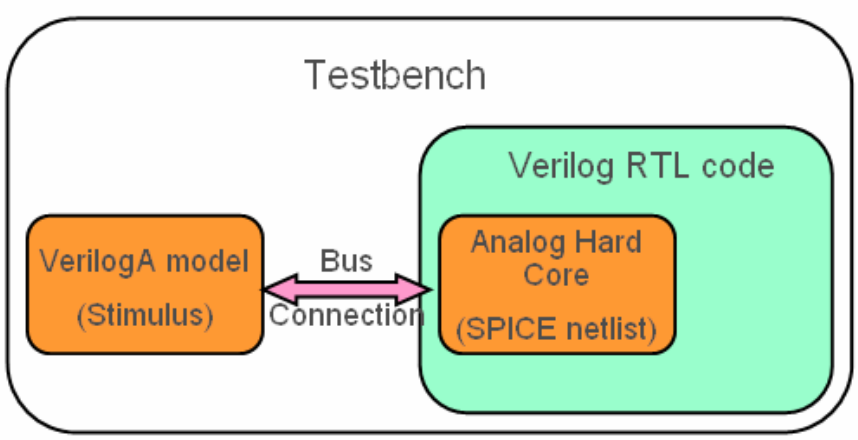

Fig 1 The block diagram of the design

III. AMS Designer and AMS Verification Flow (AMSVF)

As a new generation simulator, AMS Designer Simulator is a single executable mixed signal simulator based on the proven technology of Virtuoso Spectre and Ultrasim Simulator and the Incisive Unified Simulator engine. It provides 2 analog solvers – Spectre and Ultrasim, and supports almost all languages and SPICE netlist format. Ultrasim solver is preferred for huge full chip design because of the high performance, SPICE-like accuracy and virtually unlimited capacity.

Although AMS Designer provides a user friendly GUI in the Cadence DFII flow, most designers prefer the command line use model for full-chip verification. The reason is not only the powerful and convenient batch run capability, but also the fact that the design itself is available in Verilog, VerilogA, and SPICE text format, but not in form of schematics.

Figure 2 shows the relationship between AMSVF and AMS Designer Virtuoso Environment Flow. For command line mode and verification application based on text files, AMSVF is recommended solution provided by Cadence.

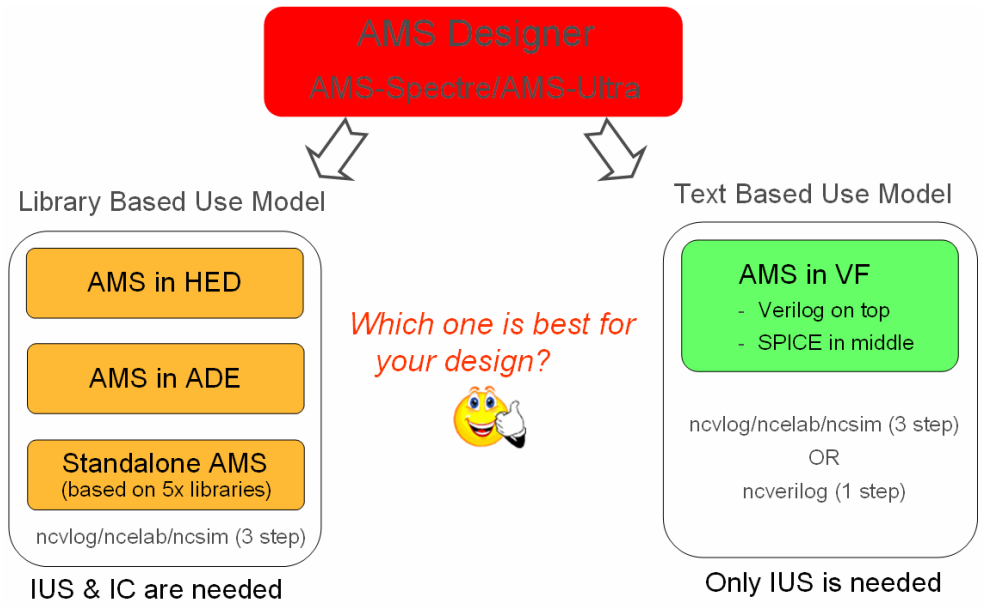

Figure 2 AMS Designer Virtuoso Environment Flow and AMSVF

IV. Use models in AMSVF

AMSVF supports two use models, ncverilog and three-step. Ncverilog is favorite in digital-centric designs. Three-step invokes a parser called ncvlog and an elaborator called ncelab to build the circuit, and then invokes the ncsim simulator to simulate the circuit.

In order to re-use existed testbench pattern and integrate the simulator into Freescale's own tools environment, the AMSVF single step command line mode was adopted.

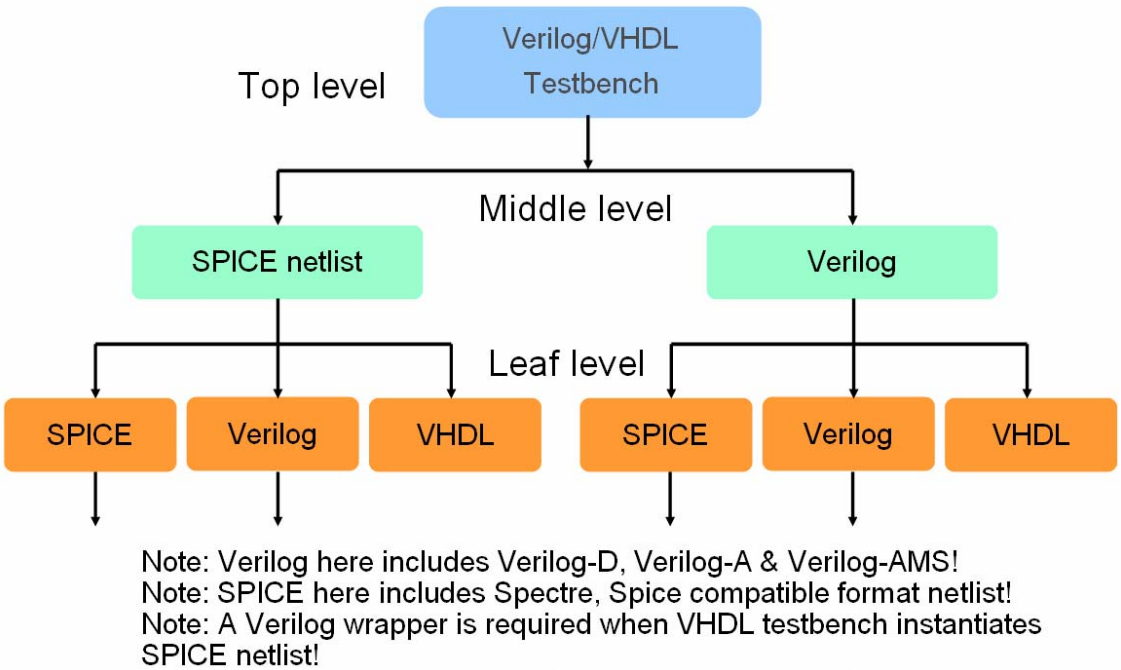

Figure 3 AMSVF's use models

Currently, AMSVF can provide the perfect support for digital testbench, see the figure 3. This situation is very common in SoC design because the digital system behavior module is created at the system

level design stage at the beginning. Another reason is digital stimulus or testbench is very easy to be implemented and very powerful. The top Verilog/VHDL level can instantiate the SPICE subckt except that a Verilog wrapper is required for VHDL testbench. The scenario is named as “Verilog on top”. Apart from it, the middle level SPICE subckt also can instantiate the Verilog/VHDL modules at leaf level, and this use model is named as “Sandwich” or “SPICE in middle”. In addition, AMSVF also supports the multiple “Sandwich” use model in some complicated designs, like “Verilog – SPICE – Verilog – SPICE – Verilog”.

The support of the two design structures makes it possible for the user to do full chip verification application. For example, for a pure digital system design, in order to get the accuracy result, some big Verilog blocks can be replaced by SPICE netlist, even with parasitic parameters for much more accurate simulation result.

V. Identifying design problems using AMSVF

The setup of AMSVF is very easy and quick. Only one week was spent to integrate the AMSVF into Freescale’s design flow, all the pattern used by the digital designer can be re-used in the mixed-signal simulation, after switching the RTL stub model to SPICE netlist. The “Verilog on top” structure is adopted to implement the full-chip verification in this project.

During the simulation and debug, several design problems and potential issues were found and fixed successfully. One design problem was found at the first run: Timing mismatch between digital and analog boundary. Separating pure digital and analog verification makes it impossible to catch such problem. In fact this signal is a pen down detection signal. It comes out from the panel; go through an inverter in the analog hard core portion, then go to the digital portion. So from the analog designer view, the timing of this signal is NOT critical. While from the digital designer view, they assume the signal comes from the analog portion is golden. Most importantly, in the method of pure digital verification, it has no way to use the touch-panel model in the pure digital testbench for the more accurate simulation.

In our mixed-signal full-chip simulation, the VerilogA model can accurately emulate the touch-panel’s behavior. Running the simulation, we found out that the detection signal always asserted even if the pen was up. With careful analysis, the issue was identified to be the panel’s capacitance being too large. When the pen is up, it needs longer time (about 1ms) to charge the panel to high level. While in the digital portion, only 4 bits are used to count the waiting time and the maximum charging time can only reach about 500us. Therefore this timing mismatch resulted in the wrong function.

The second design problem was a functional error in the analog portion. During the design process, the spec was changed and the analog portion was extended from 8 to 16 channels. The problem was found using AMSVF using a VerilogA mode to generate the comprehensive stimuli. A SPICE simulation would have also found the problem, but with the penalty of a much longer simulation time, which will significantly tighten the design schedule.

Spec misunderstanding by the digital and analog designer caused the third design problem. Multi-site design may cause this kind of design problem, due to time difference and communication issue. This kind of design problem would not be found easily without the AMSVF’s help.

The last design problem, in fact, can’t be called real design problem, it is a potential risk due to a pull-up device in the analog portion. But it is really dangerous. It doesn’t occur in common case. In the mixed-signal simulation, it can be efficiently checked out by AMSVF.

VI. Key and advanced features

AMSVF provides many useful and important features, some of them are used to reduce the manual effort to build up the testcase, and the others can significantly speed up the verification performance.

1). Automatic Bus Connection

Bus signals are commonly used in HDL language, but SPICE doesn't have a similar concept or definition. Prior to the Verilog to SPICE bus connection feature, Verilog vector bus connected to SPICE subcircuits had to be broken down into scalar ports and the nets had to be passed by order. For example,

```
module verilog;
wire [0:5] v;
analog_top xana_top ( v[0], v[1], v[2], v[3], v[4], v[5] );
endmodule

.subckt analog_top p<0> p<1> p<2> p<3> p<4> p<5>
...
.ends
```

The bus signal v[0:5] had to be split for correctly setting up the connection between Verilog instance and SPICE subckt. Often, this split process requires time-consuming manual editing, especially for bus vectors with large number of bits. With the Verilog to SPICE bus connection feature, the user can easily define the instantiation as:

```
analog_top xana_top ( v );
```

User can also set bus delimiter in SPICE netlist to tell AMSVF which sign is used for bus signal. By default the signs of <> and [ ] are treated as the bus delimiter. Even empty bus delimiter can be specified as well.

2). Port Mapping File

The port mapping file is another way allowing the user to specify how verilog ports (including buses) are mapped to SPICE. It has better flexibility and bigger case coverage. The most important is no need for the user to modify their designs. For example, the user may instantiate subcircuit analog_top in module top as:

```
module top (ext_clk, pll_clk);
input ext_clk, pll_clk;

wire [0:1] itune;

analog_top xana_top( .IN2(pll_clk), .itune(itune), .in1(ext_clk) );
...
endmodule
```

Subcircuit analog_top might look like this:

```
.subckt analog_top IN1 itune[1] itune[0] in2
```

```
...
.ends analog_top
```

In this example, the Verilog instantiation line has the mixed case for port names - IN2 and in1 and the index's order is also opposite for itune bus signal. They are not matched with the SPICE definition. Then the port mapping file feature has to be used to meet this kind of application. Add the option like "-portmap_file analog_top.pb" to sourcefile_opts property in prop.cfg file and the port mapping file contains the following:

```
IN1                     :       in1             dir=inout
{ itune[1], itune[0] }  :       itune[0:1]      dir=inout
in2                     :       IN2             dir=inout
```

3). Automatic Insertion of Connect Modules (AICM) and Block-based Discipline Resolution (BDR)

AMS Designer is well known for its strict compliance with Verilog-AMS discipline resolution process. It allows users to rely on the tool to identify inter-domain connectivity and insert the connect modules automatically. AICM algorithm determines the discipline of nets connected through undeclared ports and where to insert connect modules (CM) between analog and digital domains. Except the two common CM – electrical to logic and logic to electrical, AICM also supports bidirectional CM, which is convenient for the users who don't need to explicitly specify the port direction.

However, for the purpose of verifying big sized designs, the users need performance improvement in all aspects of the product, especially the elaboration area. BDR allows the designers to use their design knowledge of various analog and digital blocks to control the search space for discipline resolution process to work on. Moreover, the "-disres none" option even can skip the discipline resolution process to shorten the elaboration runtime if the AMSVF case has clear boundary between digital and analog or all nets are explicitly declared. Therefore, this new feature can significantly improve the performance and efficiency during the elaboration phase in AMSVF.

In addition, BDR is very useful for multiple power supply design. For example, the circuit uses 5V, 3.3V and 1.8V power supplies within the single design. When digital modules connect analog blocks with different power supply, the digital signals will have different digital disciplines, such as logic5V, logic33V and logic18V, compared with single power supply design with only one digital discipline – logic. Therefore, by this feature, the user can assign different digital disciplines to digital blocks in the design, which can tell the simulator how to correctly insert CMs.

4). Stub View

Stub view support provides a way to remove a schematic or Verilog-AMS block from a simulation. Making this substitution can help you speed up the simulation by removing blocks that do not affect the simulation result or determine if a block contributes significantly to a simulation slow down.

For example, the circuit includes 3 channels with different stimulus, and each channel doesn't affect other's result, this feature can be used to "stub" one or two channels of them and simulation speed can be accelerated.

5). New single executable command -- irun

From the coming AMSD release of IUS6.11, irun will complete replace ncverilog as the new single-step simulator. irun will fully back compatible with ncverilog. The biggest differentiator of irun is the

irun command accepts input files from several different simulation languages (such as Verilog, Verilog-AMS, SystemVerilog, Specman e, VHDL, and so on) and compiles them using appropriate compilers based on their file extensions. For example:

```
irun -amsfastspice -mess -propspath prop.cfg -analogcontrol top.scs top.v middle.vams sub.vhd
```

irun is smart enough and takes:

top.v as a Verilog file and compiles it using the Verilog parser ncvlog;
middle.vams as a Verilog-AMS file and compiles it using ncvlog -ams; and
sub.vhd as a VHDL file and compiles it using the VHDL parser ncvhdl.

The above single irun command is equivalent to the following several commands (from the three-step method for compilation, elaboration, and simulation):

```
ncvlog -mess top.v
ncvlog -ams -mess middle.vams
ncvhdl -mess sub.vhd
ncelab -amsfastspice work.top:module -propspath prop.cfg
ncsim -analogcontrol top.scs work.top:module
```

From this example, irun is much simpler than the conventional 3 steps approach. Moreover, the 3 steps commands are very easy to convert to irun command since it can support the same dash options. Most important, the new flow supports all specific features of AMSVF.

6). UPS solver

In general, a big number of RC's on power/ground net, for example, the post-layout simulation, it will dramatically slow down the performance and eat up the memory. For this kind of simulation, UPS method for IR drop analysis in AMSVF is provided to speed up the simulation while retaining the accuracy. The simulator detects the power nets first, and then separates the whole design into power network partition and signal net partition. The power networks partition is solved by UPS solver while signal net partition by Ultrasim solver. Figure 4 shows an overview of the power network simulation methodology recommended by Cadence.

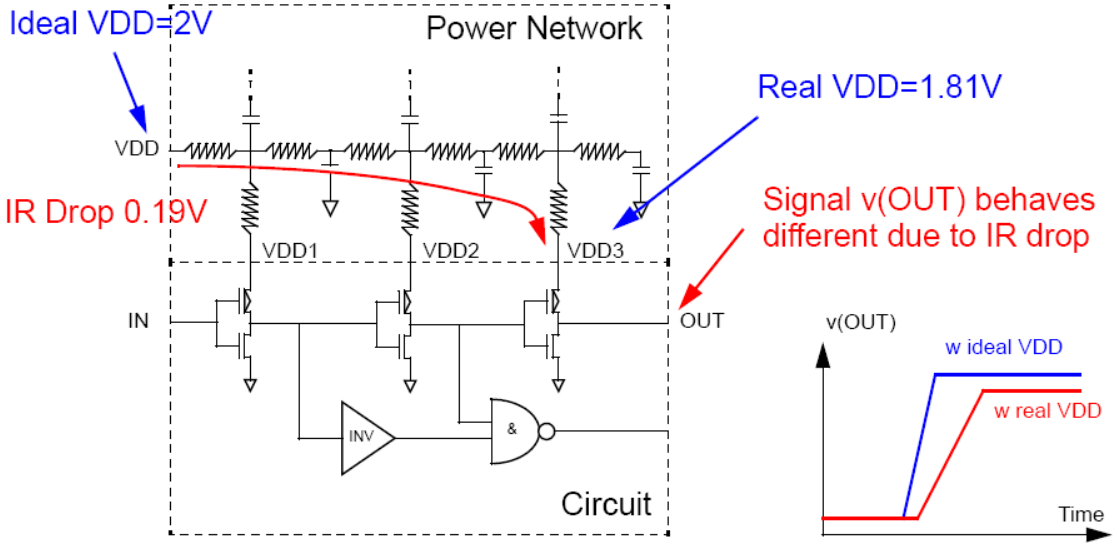

Figure 4 Overview of power network simulation methodology

See the following table for performance comparison between conventional transient simulation and IR Drop analysis with UPS solver and the simulation result for Case2 (Red line comes from UPS solver).

Table 1 performance comparison of transient and UPS solver

	Transient	UPS solver	Speedup	Accuracy error
Case1	3:22:20	26:41	7.5X	0.0401%
Case2	55:35	16:54	3X	0.0787%

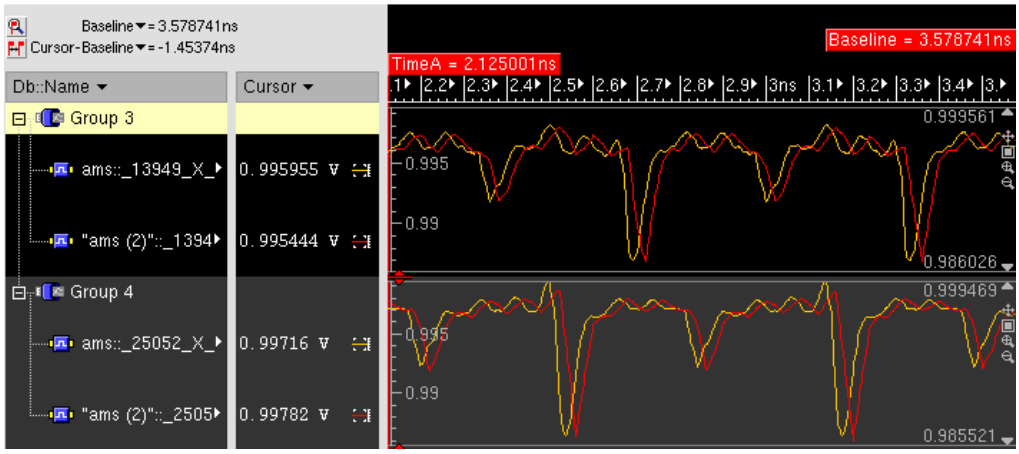

Figure 5 Simulation result comparison between tran and UPS

7). Fast Envelope Analysis

Generally, envelope simulation is introduced to simulate the modulated circuits to overcome the difficulties with conventional transient simulation where the small time steps are needed to accommodate the high frequency carrier and long time durations are required to cover the low frequency baseband signals. These types of circuits often appear in RF circuits such as transmitters and receivers.

The fast envelope simulation feature in AMSVF provides an efficient technique for analog/mixed signal circuit simulation and design. Any analog or RF portion of the circuit consisting of modulated signals can be simulated by fast envelope simulation while other portions of the circuit are simulated by digital solver or conventional transient simulation. All the simulations including digital and analog are synchronized simultaneously at each time step, which takes into account the couplings between each simulation and assures the solution accuracy. Fast Envelope analysis can skip many points in a clock cycle to reduce the large number of time steps and high computational costs.

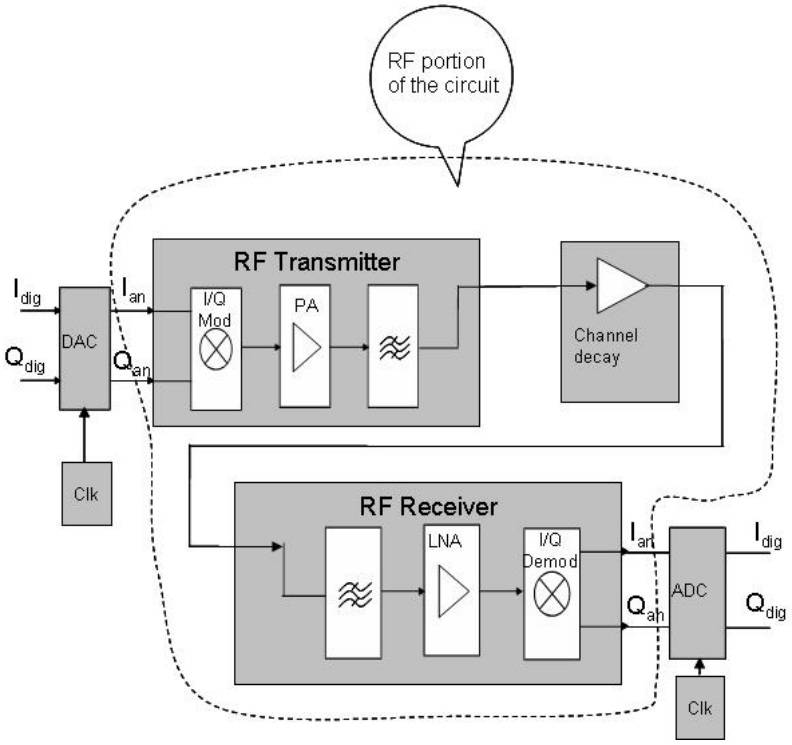

Figure 6 A complete RFIC circuit and ADC/DAC behavior module

Take the complete RFIC circuit in figure 6 as an example, which includes transmitter, receiver and ADC/DAC Verilog-AMS modules. The performance of fast envelope analysis can be improved 7X with very small accuracy loss compared with transient. Figure 8 shows the simulation waveform comparison, the last one from fast envelope has skipped many cycles.

Table 2 performance comparison of transient and fast envelope analysis

	Transient	Fast envelope	Speedup	Accuracy error
RFIC	1:52:44	15:52	7X	1.98%

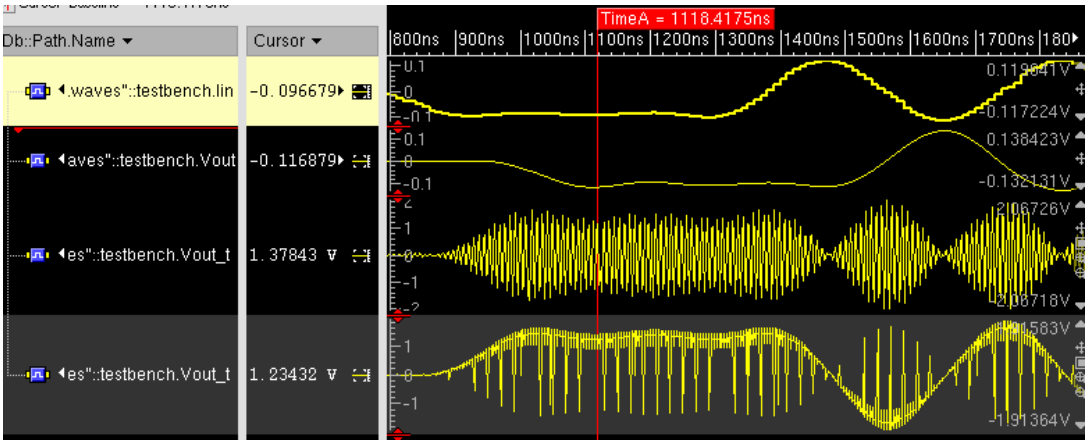

Figure 7 Simulation waveforms for transient and fast envelope analysis

VII. Summary

AMSVF has been proved to be an effective and powerful tool for full chip verification of complex mixed signal IC's, which efficiently helped to identify the design problems and potential failures in Freescale's DragonFire0 project. It has provided not only flexible use models, but also many advanced and powerful features. The solution will significantly reduce design cycles and realize first silicon success!

VIII. References

[1] Cadence® Verilog®-AMS Language Reference, Cadence Design Systems, Inc.

[2] Virtuoso® AMS Simulator User Guide, Cadence Design Systems, Inc.

[3] Virtuoso® UltraSim Simulator User Guide, Cadence Design Systems, Inc.

[4] Qian Cai, Dan Feng, Bruce McGaughy, Daniel Liu, Rendong Lin, Fangyi Rao, Vincent Li, and Yuchun Deng, "Integrated Analog Mixed-Signal Simulation for RFIC Design Flow", *Proc. CTC* 2006.

[5] Vincent Li, Tao Li, Daniel Liu, Lei Song, Stefan Wuensche, Prabal Bhattacharya, Shekar Chetput, Martin O'Leary and Angela Ruan, "Verification Flow in AMS Designer", CDNLive 2006, Beijing