Behavioral Modeling in Industrial IC Design Experiences and Observations

Ira Miller
WMSG Tempe Design Center
Frewscale Semiconductor
2100 E. Elliot Rd.
Tempe, Arizona 85284
480-413-4607
ira.miller@freescale.com

Ana Ferreira-Noullet
WMSG Toulouse Design Center
Freescale Semiconductor
31023 Toulouse
Cedex – FRANCE BP1029
++(33)5 61 19 9304
ana.ferreira.noullet@freescale.com

ABSTRACT

Large systems on a chip (SoC), means more functions, gates, software, engineering, cost and IP (1). This increase in the digital content, coupled with the mixing of analog functionality on a single chip, and the rush to market requirements, increases the breath of complexity for the hardware and software requirements to obtain "first pass success" development cycles. For very large SoCs, expenses are rising to such levels that potentially limit participation to a few large companies in the future.

IP is an enabler to smaller development centers allowing them to partner with other smaller development centers to share expenses and expertise, to participate in market segments that would be prohibitive for each company alone. Technology to support digital only "systems on a chip (SoC)" IP is common among digital tools, but fall short for analog IP. This paper presents some of the observations for analog IP in mixed signal systems development and verification.

Keywords

Analog Modeling, AMS, VerilogA, VerilogAMS, VHDL-AMS.

1. Introduction

1.1 Modeling

As a reminder, modeling is preliminary work or construction that serves as a plan from which a final product can be made. Such work can be used to test and perfect final product results. We must keep this in mind and not model for the sake of modeling. The total process must be a means to an end, and carefully thought out.

The levels of model abstraction used in the verification of the final product, provide a means to speed the development and verification at various levels of abstraction in the development and verification of a product.

1.2 Modeling for Integrated Circuit Design

Modeling "at the transistor level" in the integrated circuit industry has roots in the very basic building elements used to construct and verify the references and signal processing structures in a system. These basic structures are the primitives found in the popular Spice simulator for integrated circuit design. Since conception, the models have evolved to increased accuracy, but improvements in speed of simulation have been small without going to higher levels of abstraction. Spice simulation increases exponentially (as the device count increase past a certain number of devices) limiting the practicality of Spice simulation for large systems.

The solution for speeding the development and verification of digital functions was the development of models to represent the behavior of the digital functions, at a higher level of abstraction than transistor level models. The behavioral models are calibrated to agree with the transistor behavior with respect to delays, rise and fall times. The establish of the methodology allowed a proliferation of supporting tools enabling the development of large digital SoCs available today.

To further the advancement of mixed signal chip development, a methodology to model analog functions was required. Because analog functions are continuous in time, a viable solution was and is not as easy to develop as it was for digital systems.

2. Analog Behavioral Modeling

Early in the history of analog behavioral modeling, a product called Saber was developed to ease the burden of simulation time. The tool was effective to model smaller systems, such as switching power supplies, where the tool could encapsulate the whole design. A drawback of the tools was that it did not interface well with the digital simulators and supporting verification tools. But is was a good enabler to the development and evolution of analog behavioral modeling.

In the beginning of the development of a standard for analog behavioral modeling, Cadence donated its IP solution to higher level analog behavioral simulation. The VerilogA Language Manual (LRM) that resulted from the donated simulation tool, and the work of a standards body, established a standard that simulation tool developers can reference to build tools that can simulate models from different entities if coded to the standard.

A VerilogAMS LRM is currently in development to add features for mixed simulation, such as extensions to address connect modules to provide interface between analog and digital behavior.

In addition to the VerilogA/AMS simulator development, several Spice simulation providers offer behavioral modeling extensions to extend Spice into the realm of the mixed signal design space. Although the tools are effective for smaller (big-A/little-d) systems such as switching power supply design, they lack the capability to simulate and verify a large SoC system.

3. Intellectual Property

Development centers that wish cohabitate with other development centers to bring products to market must be able to share work among themselves, without additional expense and time. The collection of this information is referred to as Intellectual Propriety (IP).

The cataloging and documentation of the IP should allow easy and clear implementation of the information in a consistent and reusable format. At the beginning of sharing information is a good description (specification) of what the joint project. The goal is to have an executable specification that is developed, verified and well understood, to avoid specification reinterpretation and modification of silicon, at the silicon verification phase of the project.

A methodology must be in place to sort out and certify good IP and control the format of the modules so that they will work consistently as expected. Such details as a well thought out, meaningful and expandable naming convention with revision control. The location of application specific IP should be locatable in the database via a search engine. An automated maintenance routine should be used to keep the IP database up to date with the latest information and archive old models and restrict outdated model proliferation.

Simulation and measurement routines (test benches) should be a part of the IP to enable consistent and repeatable characterization and verification of the circuit level portion of the data. The test benches should provide adequate stress testing under the many conditions expected for the use of the model, and provide adequate expansion hooks to the test bench to expand the test coverage for application specific realms.

3.1 Reuse

IP requires that reuse of information to describe circuit behavior. At the definition of a product a top level simulation of the specification can optimize partitioning between the analog and digital functions in mixed signal domains, optimize performance and reduce development cycle times.

With the ever increase in SoC complexity and desire for short development times, designing every product from scratch is becoming less desirable and a reuse methodology is now a requirement.

Models in a library should be developed consistent with a reuse methodology and address application specific applications with a minimum number of models and types. Model behavior can be modified at instantiation to reduce the size of the database.

The methodology in place should be well documented and understood so that there is a well understood commonality.

The levels of analog models can be thought of as implemented for the MOS transistor model. Level 1, Level 2, Level 3, BSIM, and so forth. When a Level 1 model is used the user should have a basic understanding of the limitations and capability of the function.

In order to provide some granularity in behavioral models the models should have levels of abstraction for tradeoffs between speed and accuracy. The various levels should consist of simple models for top level simulation with increased degrees of accuracy as the simulations move down to the device models.

Modeling accuracy and variation can also be derived from functional model parameters tied to process parameters to include process variations, temperature variation, and voltage variation and can be inclusive to derive system variation and yield estimation.

4. Modeling Methodology

VerilogAMS is an enabler for IP and reuse. It allows the description of analog behavior, it is transistor level circuit friendly and is an extension of Verilog. Hopefully tools will to support VerilogAMS as they have for Verilog.

The VerilogAMS standard suggest that models follow a description level from 0 to N, where N describes the functionality of the block.

A 0 level is a <<stub>> model containing either an empty description or a simple one with all voltage and current set to zero values.A level 1 is block dependent and contains basic parameters and functions. A level 1 operational amplifier for

example, could contain parameters to describe the gain-bandwidth Product, slew rate, offset or other effects.

A higher level 2 we would add more complex performance parameters such as noise, process or temperature effects.

Each classification level for a module must be clearly defined with respect to the accuracy and speed of execution, and how the levels interact with other levels.

5. Development Tools

VerilogAMS allows a top-down mixed signal design methodology to be adopted, something that many analog designers are not accustomed to. The goal is to develop the system at a high level of abstraction before converting analog block to transistor level implementation. The development and verification of the design is a combination of simulation using a mix of behavioral models and transistor models. Behavioral and transistor level models can be swapped in and out of the design as the design progressed to a final simulation traceability to silicon.

The use of behavioral languages is strongly suggested. Using mixed level simulation tools it is now possible using the same test-bench to setup these simulations. The test bench methodology to drive the SoC should have traceability to lab and final test methodology if possible.

Many of the tools in a mixed signal design flow are provided by the three major CAD vendors: Cadence, Mentor and Synopsis. Some exploration tools are also used outside the flow, and are provided by smaller specialty tool suppliers.

The Table 1 presents the three EDA Vendors and the tools available for a top down design mixed signal flow.

Table 1- EDA vendors and Tools

Vendor	Cadence	Mentor	Synopsys
Behavioral Language based	Verilog-AMS	VHDL-AMS Verilog-AMS	VHDL-AMS Verilog-AMS
AMS Simulator Tool	AMS-Designer	ADV-AMS	Discovery-AMS
Suported Description levels	Verilog-D Verilog-A Verilog-AMS Transistor level Gate level	Verilog-D Verilog-A Verilog-AMS Transistor level Gate level	Verilog-D Verilog-A Verilog-AMS Transistor level Gate level
FAST-SPICE Tool	ULTRASIM	MACH-TA	NANOSIM
AMS+Fast Spice Solution	AMS-ULTRA	ADV-AMS	Discovery-AMS

5.1 Mixed Signal Design Flow

A mixed signal flow must be organized in order to support a top-down and a bottom-up design methodology as shown in the Figure 1:

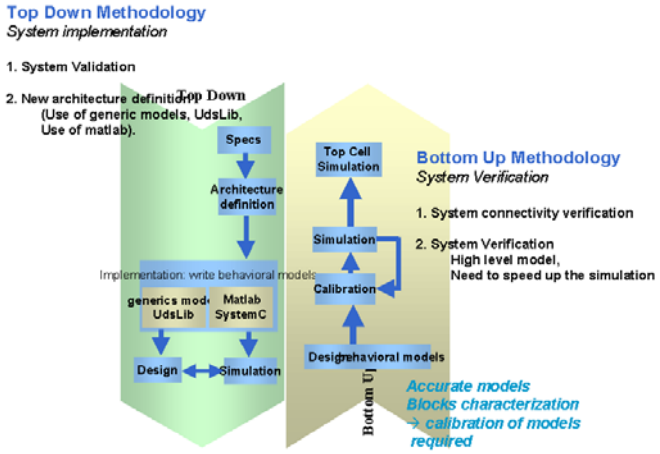

Figure 1: Top-down/bottom-up methodology used in a mixed signal design flow.

The top-down design flow starts from a top-cell block as described in a project specification book. This description consists of the definition of main blocks of definition in terms of pin-outs.

Each main block is modeled in a behavioral language as VerilogAMS (or VHDL-AMS), and top-level simulations are performed using the tools such as presented in the table 1. The goal is to verify global connectivity and resolve any inter-blocks conflicts. Each block can take on a variety of levels to optimize the speed and accuracy of verification.

The first level consists of creating a <<stub>> model with all signals set to null values. The <<stub>> model allows a check of general connectivity. The block can be set to a <<turn-off>> mode in order to keep test bench consistency when switching levels. Different levels of accuracy are used as simulations progress toward full chip verification, with each block eventually verified at the transistor level with parasitic parameters extracted from layout.

The bottom up flow is used to develop and verify transistor level development methodology and is used to simulate the SoC at the transistor level using Fast-Spice tools.

6. Parameterizable Model Example

The model , written in a behavioral language such as Verilog-AMS or VHDL-AMS, must be written in a reusable way and allow modification if required.

Parameterized VerilogAMS models provides this capability. The characteristic of the model are coded in a <<programmable>> way. For example, a simple ADC can take on different characteristics such as conversion time, gain error or off-set error at the instantiation, by passing in specific parameters to select the behavior.

The following ADC illustrates a parameterizable model:

```
//------------------------------------------------------------------------
// Description : 16bits A to D Converter taking in count
//               linear (offset and gain) and non-linear (DNL)
//               errors,
//       Based on two 8 bits ADC (half-flash architecture).
//------------------------------------------------------------------------
`define NB 16                   // adc Bit Number.

`define NS 65535   // adc State Number = pow(2,`NB)-1.
`define NT 255                  // Table Number corresponding
//to a `NB/2 bits ADC  = pow(2,`NB/2)-1.
`include "constants.vams"
`include "discipline.vams"
module adc_16bits (vin, start, vref, Bout, ack, dvdd, dvee, avdd, avee);
  inout vin;          // Analog input voltage port.
  input start;        // Digital input port signalizing the start of the
//conversion (positive edge).
  input vref;         // Reference voltage (FSR).
  output [`NB-1:0] Bout;      // Digital output data port (binary
//word).
  inout ack;          // Digital output port signalizing the end of
//the conversion(positive edge).
  inout dvdd;         // Digital vdd / dvee
  inout dvee;         // Digital ground
  inout avdd;         // Analog vdd / avee
  inout avee;         // Analog ground

  electrical vin;
  electrical start;
  electrical vref;
  electrical [`NB-1:0] Bout;
  electrical ack;
  electrical dvdd;
  electrical dvee;
  electrical avdd;
  electrical avee;

  branch (vin, avee) brin;      // input resistor branch.
  branch (vin, avee) bcin;      // input capacitor branch.
  branch (Bout, dvee) brBout;         // vector branches are
always numbered [0,n]
  parameter integer SYNCHRONE  = 1;         // Boolean
selection to take in count or not the start synchonization (positive
edge).
  parameter real   TCONV   = 1n;      // Conversion time.
  parameter real   TRISE   = 1p;     // Digital output voltage Rise
Time (ack port).
  parameter real   RIN    = 1G;       // Input resistance
(between vin and avee ports).
  parameter real   CIN    = 0.0;      // Input capacitance
(between vin and avee ports).
  parameter real   ISUPPLY = 0.0;     // DC current
consumption from vcc to vee supply pins.
-------------------------------------------------------------------------
```

Another advantage of parameterized models is that the behavioral model can be calibrated to the transistor implantation without having to rewrite the model. The calibration method consists of fitting the model to the transistor level simulation (or lab data) results via the parameter passed in at instantiation. Tools are now appearing to automate calibration. The Cadence tool APTIVIA can be used to fit models to transistor level simulation results.

7. Conclusion

IP, behavioral modeling, and reuse; with the introduction of computer aided support tools, will enable the development and verification of mixed SoC chips, to evolve to new levels of functionality. Carefully thought out methodologies and implementations are required to efficiently manage the IP generation and databases for efficient cost effective solutions.

8. References

(1). Roberts, Bill, about “Compound complexity—and how to manage it”. *Electronic Business* (Sept. 2004).