Analog Behavioral Modeling and Mixed-Mode Simulation with SABER and Verilog

A description is given of specific analog behavioral modeling and mixed-mode simulation techniques using SABER and Verilog. Full-channel simulations have been carried out on a class I partial response maximum likelihood (PRML) read/write channel chip. Complex analog circuits such as an adaptive feed-forward equalizer, an automatic gain control block, and a phase-locked loop are modeled in detail with the SABER MAST mixed-signal behavioral modeling language. A simulation speedup of two orders of magnitude has been achieved compared to SPICE.

by Ben B. Sheng, Hugh S.C. Wallace, and James S. Ignowski

For more than two decades, the analog IC design community has been relying on variations of the original Berkeley SPICE, introduced in the 1970s, as the simulation tool for verifying and fine-tuning analog designs. Over the years, many enhancements have been put into these different flavors of SPICE, while increasingly more powerful computers have been used for running these circuit simulations. However, SPICE remains a low-level circuit simulator. It produces accurate results, but is inherently slow. Today's analog and mixed-mode designs are becoming increasingly complex. Functional simulations for larger mixed-signal designs are impractical with SPICE. Meanwhile, as the pressure increases for low-cost, high-integration ASICs ("systems on a chip"), many analog functions are being integrated into largely digital chips. The need for new simulation methodologies is becoming more urgent.

In recent years, benefits from using analog and mixed-mode behavioral modeling languages have received increased recognition. The basic approach is to use a SPICE-like continuous-time simulator, which provides good accuracy in simulations, together with a fast digital simulator to give orders of magnitude faster digital circuit simulations. The modeling language is flexible so that designers can model analog subsystems in different levels of abstraction. The modeling language gives designers control over the trade-off between simulation speed and accuracy.

This paper presents some of the bottom-up modeling techniques and simulation approaches that have been adopted during the process of modeling and simulating the read-write channel chip for an HP DDS-3 DAT drive.

Analog Behavioral Modeling

The idea of behavioral modeling is not new to analog designers. Macro models have been widely used by SPICE users. The newer-generation mixed-mode circuit simulators, such as SABER by Analogy, Inc. and SPECTRE by Cadence Design Systems, Inc., have greatly enhanced designers' ability to model analog and mixed-mode circuits and systems by providing a flexible behavioral modeling language. With this modeling language, a designer can behaviorally describe an analog or mixed-mode device or subsystem at whatever level of abstraction is appropriate for a given simulation accuracy-versus-speed trade-off. One can use this modeling language to write BSIM models for MOS transistors and use these BSIM models to achieve simulation results that are as accurate as those from SPICE simulations. The same modeling language can be used to describe an analog-to-digital converter (ADC) behaviorally, without having to refer to any of its internal circuit elements.

Several modeling approaches are discussed in this section. Based on the scopes of these different approaches and their simulation speed-versus-accuracy trade-offs, they can be categorized as either high-level, medium-level, or low-level modeling. In the following subsections, specific examples are given for high-level and medium-level modeling. Although low-level modeling is a very important part of analog modeling and simulation, the techniques used to do low-level modeling are very similar to those used in higher-level modeling. The only difference is that these techniques are used to model much smaller devices, such as MOSFETs, diodes, and bipolar junction transistors. For brevity, discussion of low-level modeling is omitted.

High-Level Modeling

High-level modeling refers to behavioral models that describe large analog and mixed-mode subsystems in a high level of abstraction. This approach provides the fastest simulation speed but the least detail in the circuits that are modeled.

An ADC can be modeled with a clock input signal that triggers each conversion, an analog input signal, an analog reference signal, and digital outputs. In addition to this basic structure, some realistic behavior can be included in the model. For example, the model can include characteristics such as differential nonlinearity, integral nonlinearity, and metastability characteristics. A behavioral 3-bit ADC model written in the SABER MAST modeling language is shown in Fig. 1.

As can be seen in this example, high-level modeling can be used in describing analog and mixed-mode subsystems, with some detail included. This particular approach is suitable for functional simulations of large systems. A large number of functional simulations can be carried out quickly, but circuit details are often omitted. This type of high-level modeling can speed up simulations by at least three orders of magnitude compared to SPICE, at the cost of not being able to simulate the fine details in the circuits.

For complex mixed-signal designs, chip-level connectivity verification is often a problem, since neither traditional analog simulators such as SPICE nor pure digital simulators such as Verilog can, for example, check voltage reference levels or common-mode levels of differential signals. One of the most important benefits of using high-level models is the ability to verify top-level circuit connectivity when the final chip is composed. One specific example is given in the ADC model of Fig. 1. Line 21 checks that the reference voltage to the ADC is not connected incorrectly (negative reference).

Another key point is the ability to do *analog assertion*. Traditional graphical analog postprocessors work well if there are a manageable number of signals and time windows to look at—in other words, when dealing with simulation results of a relatively small circuit. For system-level simulations, in which multiple complex analog blocks interact with each other and hundreds of signals are changing constantly, it becomes very difficult to track all the important signals to make sure that all the circuits are operating within their specified parameters. The analog modeling language allows designers to put specific assertions in the models to monitor the analog signals and give warning messages or even abort simulations when the model is operating outside of a set of prespecified parameters. An example of such an assertion can be seen in line 22 of Fig. 1, where the input signal is compared to a prespecified maximum level.

The ADC model not only evaluates signals in the analog domain, but also schedules digital events. As mentioned earlier, the newer-generation simulators not only provide SPICE-like analog simulation engines, but also have built-in digital simulators, which give them much improved performance in simulating mixed-mode systems. Built-in digital simulators are orders of magnitude faster in simulating digital circuits than a SPICE-like simulator. We will discuss mixed-mode simulations in more detail later.

Medium-Level Modeling

Medium-level modeling refers to behavioral modeling of smaller building blocks, such as an operational amplifier (op amp), a multiplier, an integrator, or a comparator. These circuits typically contain a few dozen transistors. Models for these circuits can have more physical characteristics, which track SPICE simulation results.

For modeling the read/write channel chip, we used medium-level models most extensively. Three different techniques were used to develop the models, based on the different circuit structures.

The first approach is to create generic building blocks with flexible parameters that can be used to customize these generic models when they are used in different applications. A good example for this approach is the model for an op amp. Characteristics of a generic op amp include dc gain, pole location (if a second pole location is important, one can simply connect two one-pole models in series), zero location, input impedance, input capacitive load, output impedance, output capacitance, output common-mode level, slew rate, output swing, and nonlinear distortion. More comprehensive characteristics can be added to this list when needed. Fig. 2 shows an example of such a generic op amp model. Some of the parameters listed above are omitted for brevity.

This type of generic circuit model can be made flexible enough to satisfy a variety of modeling and simulation needs. Additional characterization can be included, when appropriate, at relatively low simulation speed cost.

In some cases, when the system performance is highly sensitive to the modeling accuracy of a small block, a curve-fitting approach can be used. This approach is suitable for small circuits with few input ports. Outputs of a circuit can be empirically fit to carefully chosen mathematical functions. These models are fully customized for the given circuits. They can be made very accurate in representing a few key characteristics of some small blocks, but they are not reusable for modeling other circuits, and can become very complex when more input/output signals are added to the equations. Designers who take this approach must be careful to produce a model that covers the entire input signal range to prevent the models from producing erroneous simulation results.

The third approach taken in modeling circuits in the read/write channel chip is to use the first-order MOSFET equations to approximate circuit behaviors. A good example is a MOSFET switch. The on-resistance of a MOSFET switch can be approximated with the first-order equations accurately enough for most applications. The switches are effectively modeled as nonlinear resistors. If a more accurate model is needed, the curve-fitting method can be used.

```
template adc clk in ref b2 b1 b0 = tau, td, dnl, inl, vmax
{
  # Capacitive input load of 1 pF
  c.input_load in 0 = 1p
  # Compute a random integral nonlinearity
  number random_inl = (2*rand()-1)*inl
  when(event_on(clk, clk_last)) {
    # Looking for rising edges in clk.
    if (clk == l4_1 & clk_last == l4_0) {
      schedule_event(time, b2, l4_x)
      schedule_event(time, b1, l4_x)
      schedule_event(time, b0, l4_x)
      # Tell the analog simulator to step on the clock edge to get accurate
      # analog signal value
      schedule_next_time(time)
      # Sample input voltage (with some adjustment for correct zeroing)
      # and reference voltage
      vref = v(ref)
      vin = v(in) + vref/8
      # Error checking
      if (vref < 0) error (”The voltage reference to the ADC is negative”)
      if(abs(vin) > vmax) error (”The ADC input signal is out of range”)
      # Determine the sign bit
      if (vin < 0) d2 = 0
      else d2 =1
      # Add random differential nonlinearity
      vin = abs(vin) + (2*rand()-1)*dnl
      # Clipping the output
      if (vin > vref) vin = vref
      # Add random integral nonlinearity
      vin = vin + random_inl*(vref - abs(vin))*abs(vin)*4/vref/vref
      if (vin < vref/2) d1 = 0
      else {
        d1 = 1
        vin = vin - vref/2
      }
      if (vin < vref/4) d0 = 0
      else {
        d0 = 1
        vin = vin - vref/4
      }
      # Compute resolution time. If td+t_resolve > 1-clock-cycle, then the ADC is
      # in a metastable state (a conversion error occurs)
      t_resolve = tau*ln(2/(abs(vin)+1u))
      if (d2 == 1) schedule_event(time+td+t_resolve, b2, l4_1)
      else schedule_event(time+td+t_resolve, b2, l4_0)
      if (d1 == 1) schedule_event(time+td+t_resolve, b1, l4_1)
      else schedule_event(time+td+t_resolve, b1, l4_0)
      if (d0 == 1) schedule_event(time+td+t_resolve, b0, l4_1)
      else schedule_event(time+td+t_resolve, b0, l4_0)
    }
  }
}
```

Fig. 1. *A behavioral 3-bit ADC model written in the SABER MAST modeling language.*

```
template opamp inp inm outp outm vcm vdd gnd pd = ci, gm, gm3, vswing, ro, co, rz, imax
{
  c.cip inp 0 = ci
  c.cim inm 0 = ci
  values {
    # Define the block’s inputs
    vint = v(inp,inm)
    vinp = v(inp)
    vinm = v(inm)
    if (v(inm) > v(inp)) {
      vinp = v(inm)
      vinm = v(inp)
    }
    # Gain and 3rd order nonlinear distortion
    idiff = gm*vint + gm3*vint**3
    # Output swing, the output clips at ”vswing”
    iclip_up = (vswing - v(outp,outm))/ro/1k
    iclip_dn = (vswing - v(outm,outp))/ro/1k
    if (v(outp,outm) > vswing) {
      iclip_up = iclip_up*10meg
    }
    else if (v(outp,outm) < -vswing) {
      iclip_dn = iclip_dn*10meg
    }
    idiff = idiff + iclip_up - iclip_dn;
    # Slew rate
    if (idiff > imax) idiff = imax
    if (pd ~= l4_0) idiff = 0
  }
  equations {
    # A pole and a zero
    icapp = d_by_dt((v(outp) - icapp*rz)*co)
    icapm = d_by_dt((v(outm) - icapm*rz)*co)
    i(outp) -= ioutp
    ioutp: ioutp = idiff - v(outp,vcm)/ro - icapp
    i(outm) -= ioutm
    ioutm: ioutm = -idiff - v(outm,vcm)/ro - icapm
  }
}
```

Fig. 2. *An example of a generic op amp model.*

Mixed-Mode Simulations

SPICE is an integration-based simulator. Although it gives good simulation accuracy, it is inherently slow, since it has to set up and solve a nonlinear system matrix for the entire circuit at every integration time point. The integration time step is determined by the largest signal change in the circuit, regardless of whether it is a digital signal or an analog signal. It is impractical to do system-level simulation with such an algorithm when the system has more than a few thousand transistors.

To run system-level simulations on a complex mixed-signal design and get reasonable accuracy, it is necessary to use an event-driven digital simulator to simulate the digital portion of the design, together with an integration-based analog simulator to simulate the analog circuits. The newer-generation mixed-mode simulators provide designers with both digital and analog simulation engines in one simulator. Continuous-time analog signals are computed by the analog simulator, while the digital simulator evaluates the event queue. As illustrated by the ADC example, designers can specify the interactions between the digital and analog domains. These interactions are resolved with the relaxation method.[1] The computational overhead for resolving these interactions is minimal, since the interactions are by nature discrete events. In addition, the built-in digital simulators make it easy for the mixed-mode simulators to cosimulate with widely used external digital simulators, such as Cadence’s Verilog-XL. Since the mixed-mode simulators already create digital event queues, they can share their event queues with an external digital simulator’s event queue, and vice versa. The ability to cosimulate with well-established digital simulators makes it possible to use all of the available digital libraries, which are often built based on careful characterizations and can provide accurate simulation results.

Fig. 3 shows the functional block diagram of the read/write channel chip. The shaded blocks indicate analog functions that were modeled with a behavioral analog modeling language. The rest of the chip (approximately 40%) is digital, including all of the system timing and control functions, a maximum-likelihood sequence detector (MLSD or Viterbi detector), and much of the phase-locked loop. These digital circuits were designed with Verilog HDL and synthesized with Synopsys using HP CMOS34 (1.0-μm CMOS) standard cells. For the final system-level simulations, the digital circuits were simulated with Verilog and behavioral models of the analog circuits were simulated with SABER. Special interface models were built to establish connections between the two simulators.

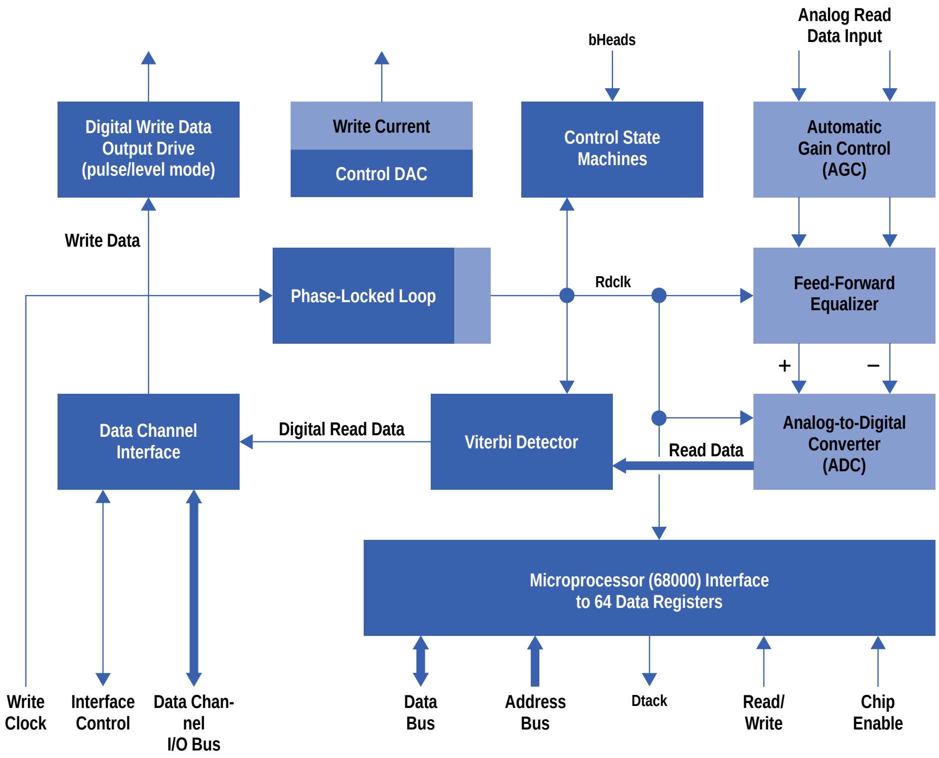

***Fig. 3.** Functional block diagram of the read/write channel chip. The shaded blocks indicate analog functions that were modeled with a behavioral analog modeling language.*

System-Level Simulation Example: Read/Write IC

The read/write channel chip is a mixed-signal IC with high analog circuit content. Approximately 60% of the core area is analog in nature. Because of the complexity and size of the analog circuits, many typical design considerations became more challenging. Simulating the interfaces and interactions between analog blocks as well as between the analog blocks and their digital control blocks was critical for a successful design. Translation of system-level specifications into meaningful circuit block design specifications was difficult, and probably most important of all, managing the system-wide budget of circuit nonideal behaviors was a formidable task. Ultimately our IC design team, working closely with the system design team from HP's Computer Peripherals Bristol Division, developed a simulation strategy of linking and correlating the simulation results of several tools to provide a closed loop of circuit and system verification.

In the initial step of the design process, the customer specified circuit performance by building a system model using C and Mathematica that was partitioned in a fashion consistent with the circuit partition. Much of the read/write channel chip's analog signal path was based on an analog PRML (partial response maximum likelihood) read channel chip that was developed for disk drive applications, so there was some previous work that was leveraged in architecting the read/write channel chip. As functional descriptions of circuit operation were put into the system model, specific limits and design goals could be identified. This information drove the design of the analog circuit blocks, which were then simulated in SPICE. The detailed results of the individual block simulations were incorporated back into the system model and real-world, nonideal circuit behaviors, such as offsets, nonlinearity, and PVT (process, voltage, and temperature) dependencies, were introduced into the system model. This increased level of detail in the system model led to very clear and well-defined specifications for the analog circuit blocks. This linking of system performance to circuit block performance allowed us to determine overall channel performance and define criteria for deciding when the design had acceptable margin and was ready for tape release.

In parallel with the system simulations, high-level simulations using SABER and Verilog were being run with analog block models. The high-level simulations performed functional and timing checks of the digital control blocks and their interfaces to the behavioral representations of the analog blocks. In this way, the operation of analog circuitry under the control of complex state machines was verified. The results of high-level simulations of the analog portion of the system were compared to the results of the Mathematica system model for verification. Analysis and debugging of the analog behavioral simulation results were valuable for modeling and for confirming observations from the test and characterization of first silicon.

Critical Analysis

The read/write channel chip was our first major chip design for which high-level system design and verification were extremely critical to the chip's performance. Many previous mixed-signal designs depended on just the individual functional blocks' meeting the specifications to ensure that the final chip would perform to the system specifications. For the read/write channel chip, the system performance and trade-offs could only be evaluated after much of the actual circuit design was completed. The system performance of the read/write channel chip was extremely sensitive to a number of nonidealities in the signal path, so multiple iterations were carried out between careful circuit characterizations and system performance evaluations.

After the project was finished, and with an increasing need to define formally a methodology for new mixed-mode signal processing chips, the strengths and weaknesses of the read/write channel chip methodology were evaluated.

The high-level SABER models verified that the digital control circuitry functioned correctly with the analog blocks. This contribution cannot be overstated. The number of signals together with the complex interaction between the digital and analog blocks cannot be checked adequately in any other manner. The behavioral modeling of the analog blocks also discovered problems within the analog circuitry. While many of the main related analog blocks were simulated in SPICE together, there were others that were not, because of size and speed limitations in SPICE. The netlist extraction for the top-level SABER simulations was automatically generated from the chip transistor schematic, so the high-level simulations gave good confidence that the chip was correctly connected.

The main weakness in the process was that one engineer was responsible for developing all the analog behavioral models. This was because of manpower constraints in this project and the new introduction of the SABER tools. The investment in the learning curve for the new tools could not be absorbed by this project. The process that was followed to develop more than 100 behavioral models was examination of the schematics and SPICE plots that were available, and communication with the block designers. This worked fairly well in most cases, but there were some instances of incorrect behavioral modeling. The main problem was that human interpretation was needed to create the behavioral models. Characterization tests to compare the circuits with the models could have helped, but with so many blocks, some being continuously modified, a full set of characterization tests was not practical with the given amount of time and resources.

Enhanced Methodology

An enhanced methodology based on the one described above has been developed. This new methodology is intended to eliminate the weaknesses described in the read/write channel chip mixed-mode methodology by leveraging as much as possible from the behavioral synthesis methodology used in digital VLSI designs. The key is to remove as much human interpretation as possible and use the test vector set as a common link from customer definition to final chip simulation. Since not all the specifications in a mixed-mode design can be encoded with traditional digital test vectors, many of the analog specifications can only be evaluated by specific analog tests. The key to these tests being successful is to develop them from the customer's written specifications or the circuit design engineers' knowledge when the behavioral model is developed. The intention is that the block designer will develop the behavioral model. By taking this approach and developing tests that will run on both the real schematic and its behavioral model, a set of characterization tests can be exercised on the behavioral model. By starting a block design with a behavioral model based on written specifications and a top-level customer model, a more specification-driven evaluation of each circuit block can be used to determine whether the circuit will perform its desired function. This makes it possible to delay the exact circuit implementation until the full requirements of the block are understood. It could, for example, make the difference in deciding whether a block should be designed in the analog or the digital domain.

The approach of having the block designers develop the behavioral models still requires human interpretation and complete understanding of the circuit. Although this approach is an improvement of the methodology, the human interpretation is still a weakness. A number of simulators now gaining wide acceptance in VLSI digital design allow the whole chip to be converted into a FET-level simulation. The simulation speed is much faster (10×) than a SPICE simulation, with apparently improved convergence. The improvement in speed is obtained by running many parts of the circuit linearly. This approach compromises accuracy somewhat. The key to this new way to simulate mixed-mode systems is to define which blocks need to be simulated in a SPICE-like mode to ensure the required accuracy. This whole-chip simulation can only be performed when a complete FET schematic is available, so it is seen as a final verification check using the main mixed-mode simulation vector set.

Acknowledgments

The authors would like to thank Derek Knee, Kalwant Singh, and Mike Gilsdorf for reviewing this paper and giving valuable feedback.

Reference

1. *The Cutting Edge*, Analogy Inc., March 1996.

▶ Go to Next Article

▶ Go to Journal Home Page