New Challenges in Verification of Mixed-Signal IP and SoC Design

Luke Lang
Cadence Design Systems, Inc.
2655 Seely Ave.
San Jose, CA 95134
1-408-576-3640
lukelang@cadence.com

Christina Chu
Cadence Design Systems, Inc.
2655 Seely Ave.
San Jose, CA 95134
1-408-895-3230
cchris@cadence.com

ABSTRACT

With the increasing demand in mobile and industry controller applications, a SoC design has more and more mixed-signal contents with the usage of some advanced power management techniques, such as power gating, dynamic voltage and frequency scaling etc. Traditional mixed-signal verification methodology relies on circuit simulation at different abstract levels. At the SoC level, mixed-signal functional simulation is the most commonly used technique to ensure the functional correctness of a mixed-signal design. However, will the functional correctness verified by simulation guarantee fully functioning silicon? Some other short comings of the simulation approach are strong dependency of the functionality coverage on the test vectors and the very long simulation time due to mixed-signal content.

The paper starts with an example showing that a mixed-signal block or IP with advanced power management techniques imposes unique challenges to SoC level verification. Such a design may pass SoC level mixed-signal functional simulation but fail on silicon due to electrical failures, even when low power functionality is included during simulation. The paper then proposes to use the static verification methodology, which is widely used for digital low power designs, on mixed-signal IP and SoC and demonstrates how it will help to catch the electrical failures, which cannot be detected by using the functional simulation.

Such a methodology requires the proper modeling of a mixed-signal IP with power management features. Traditionally, a mixed-signal IP is considered as a block box at SoC integration level. As a result, the designer will face the challenges of how they can ensure the IP is properly integrated into the SoC. Typically, designers can choose to create a cell model using Liberty to enable some basic checks. However, with advanced power management features, the IP may have complex interface logic, such as isolation and clamp diodes, to ensure low power functionality. Such low power logic imposes additional constraints on how this IP should be integrated at SoC level. In addition, power management introduces additional complexity in the functional modes of the IP, which must be checked against top level modes to ensure the IP is used properly. As a result a more complex macro cell model is needed for such complex low power functionality of a mixed-signal IP. In this paper, the author will introduce such a model and demonstrate a prototype of macro model generation of a mixed-signal IP with power management features from the conventional schematic driven analog and mixed-signal design environment. This automation can also automatically identify the power and ground connection, special low power cells such as isolation and level shifter cells and is able to automatically create power intent abstracts. Such a model can then be used by a formal power structure verification tool to check the power connectivity and functionality at SoC level.

The paper is then concluded with several real design examples where SoC level design failures can be caught using this new methodology.

1. INTRODUCTION

Today, most, if not all, SoC designs contain both digital and analog circuits. There are several reasons for the disappearance of purely digital SoC. 1) Rapidly shrinking process geometry has resulted in unprecedented integration of circuits and functions onto a single SoC. This naturally includes integration of analog and digital circuits. 2) Mobile communication SoC's are inherently analog in dealing with RF signals. 3) High-speed serial data interfaces require analog circuits for clock generation. 4) Advanced low-power techniques require voltage regulators and possibly PLL's to adjust supply voltage and clock frequency. 5) Micro controllers interface with the external world through analog circuits, such as sensors, actuators, etc.

Since digital SoC implementation tools do not deal with analog circuits, these analog/mixed-signal circuits have traditionally been treated as IP blocks and appear as a black-box to the digital tools. The digital SoC is verified with models representing the analog/mixed-signal circuits. The analog/mixed-signal circuits are verified standalone, usually through SPICE simulation. This design approach has been in use for several decades. One simple and common example is the embedding of memories in a digital chip.

With increasing integration, the interaction between analog and digital has never been more complex. Advances in analog/mixed signal simulation and analog behavioral modeling have been well documented.[1] For the most part, they have adequately addressed the verification accuracy and performance issues. But the rapid adoption of advanced low-power design techniques for mixed-signal design has introduced new challenges.

A design that is functionally correct may not work in silicon due to incorrect low-power implementation. For example, a missing isolation cell will lead to excessive leakage current. Yet an extra isolation cell may block propagation of data. These are just some of the basic considerations of implementing power gating. With some of the more complex low-power techniques, such as dynamic voltage and frequency scaling (DVFS) and adaptive voltage scaling (AVS), the number of failure mechanisms grows rapidly. This makes the already difficult mixed-signal verification task even more challenging.

One solution is to leverage the low-power structural verification technique that has been commonly used for digital low-power design flow and apply it to mixed-signal low-power designs.[2] Structural verification relies on electrical properties of driving and receiving circuits to determine compatibility. It does not require functional simulation and therefore reduces the demand on testbench coverage for mixed-signal verification. However, the traditional IP or black box abstraction using Liberty and simulation model breaks down because these models only contain functional, timing, and power dissipation information. They are missing the low-power intent of the mixed-signal IP. We shall see how this gap is filled by Common Power Format (CPF) macro models and CPF-enabled Analog Mixed-signal Simulator (CPF-AMS).

This paper will first review some basic concepts of power intent. These concepts are applied to model a mixed-signal IP block. Using some sample designs, this paper will describe the automation to generate the CPF design model and the CPF macro model automatically from a circuit schematic. Both of these files are needed for low-power mixed-signal dynamic simulation and structural verification.

2. POWER INTENT OVERVIEW

The power-intent file describes all the power information of a design, including the operating voltage and conditions of various blocks in the design. When interfacing blocks operate at different voltage, the need for low-power circuits, such as isolation cells and level shifters, is also part of the power intent file. Once the intent is described, the designer will not have to labor through the tedious tasks of inserting and verifying these low-power circuits. These tasks can be automated by EDA tools. Using one power-intent file for all EDA tools in a design flow will ensure consistency in verification and implementation of the low-power design.

Currently, there are two industry open standard formats for power intent: Si2 Common Power Format (CPF)[3] and Accellera Unified Power Format (UPF)[4]. (The second version of the UPF standard is IEEE 1801.[5]) Both CPF and UPF are being used for low-power digital SoC. This paper is based on the CPF standard primarily because only CPF has the macro model construct. We shall see why this is important in later sections.

This section introduces some basic power intents used in this paper. For a complete description of the CPF power intent, please refer to the CPF Language Reference.[3]

2.1 Power Domains

A power domain is a group of design components that are connected to the same primary power and ground nets. If the primary power/ground nets can be shut off through internal power or ground switch cells or external mechanisms, then this power domain is called a switchable power domain. For a switchable power domain, the control signal that shuts off the switch cell is known as the shutoff condition in the CPF file.

Another key concept is boundary port. It specifies the expected power domain of the input and output ports. This determines the electrical compatibility of the internal circuits relative to the I/O ports.

Figure 1 shows a simple power architecture consisting of two power domains. PD2 consists of block u2 and is always-on. Its primary power and ground nets are VDDG and VSS. PD1 is a switchable power domain and consists of block u1. Its primary power and ground nets are VDD and VSS.

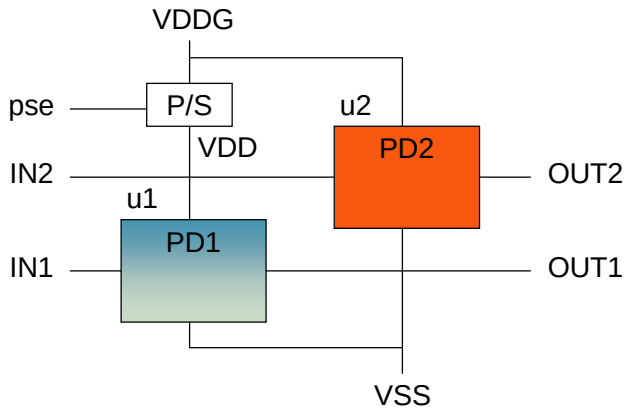

Figure 1. Sample Power Architecture

Figure 2 lists the corresponding CPF commands to define the power domains in Figure 1. PD1 is defined as block u1 with boundary ports of IN1 and OUT1. Its base domain is PD2. This means that it must have the same voltage as PD2 and will be off if PD2 is switched off. The pse signal controls the power switch for PD1. PD2 is defined as the default power domain with the –default option. This means that all design instances and boundary ports belong to this power domain unless they have been specified as belonging to another power domain. In this case, block u2 and ports IN2 and OUT2 do not belong to PD1. Therefore, they are in the default PD2 domain.

```
create_power_domain -name PD2 -default

create_power_domain -name PD1 \
  -instances u1 -base_domains PD2 \
  -boundary_ports {IN1 OUT1} \
  -shutoff_condition {pse} \

update_power_domain -name PD2 \
  -primary_power_net VDDG \
  -primary_ground_net VSS


update_power_domain -name PD1 \
  -primary_power_net VDD \
  -primary_ground_net VSS
```

Figure 2. Sample CPF for Power Domain

2.2 Nominal Conditions and Power Modes

The CPF commands listed in Figure 2 specify that PD1 can be switched off. There is no specification of the voltage of PD2 as well as whether PD2 can be switched off externally. In CPF, nominal conditions and power modes are needed to specify this additional information.

In general, each power domain can operate at one or more voltage level. Each operating voltage is a nominal condition. Each valid combination of nominal conditions for all power domains is a power mode.

Figure 3 lists the CPF commands to specify three nominal conditions (1.08V, 0.9V and 0V) and three power modes. PD1 and PD2 operate

at the same voltage since PD1 is derived from PD2, except when PD1 is switched off.

```
create_nominal_condition -name high -voltage
1.08
create_nominal_condition -name low -voltage 0.9
create_nominal_condition -name off -voltage 0

create_power_mode -name PM_HIGH –default \
  -domain_conditions {PD1@high PD2@high}
create_power_mode -name PM_LOW \
  -domain_conditions {PD1@low PD2@low}
create_power_mode -name PM_OFF \
  -domain_conditions {PD1@off PD2@low}
```

Figure 3. Sample CPF for Nominal Conditions and Power Modes

From the power modes, we can tell that PD2 is always on. Also, the design requires multiple supply voltage.

2.3 Isolation Rules

Figure 1 shows that PD1 drives a signal to PD2. Since PD1 can be switched off while PD2 is on, this signal must be isolated prior to driving into PD2. This requirement is specified as an isolation rule in Figure 4.

```
create_isolation_rule -name ISO1 \
  -from PD1 –to PD2 –isolation_output low \
  -isolation_condition iso_en

update_isolation_rule –name ISO1 \
  -location to
```

Figure 4. Sample CPF for Isolation Rule

This isolation rule specifies PD1 as the source domain and PD2 as the destination domain. The isolation is activated when the iso_en control signal is high. During that time, the data from PD1 to PD2 must be driven low, as specified by "-isolation_output low". Finally, the isolation cell should be instantiated in the "to" domain (PD2).

3. MODELING IP BLOCKS

As mentioned previously, using a black box abstraction, such as a Liberty model, to represent mixed-signal IP blocks have been used for decades. CPF macro model was introduced to provide the much needed power intent of library cells and IP blocks that cannot be fully described in Liberty.

3.1 Extracting Related Power Pin

CPF boundary port associates each data pin to the power and ground pins of the IP block. This not only indicates the expected voltage of the pin. It also indicates the state of the each pin. When a power net is switched off, all of its related data pins are also switched off. Specifically, the related output data pins are undriven or floating. EDA tools must identify these floating pins and enable isolation for those pins.

One of the challenges of developing such model in Liberty is determining the related power pins and annotating this attribute. This is a mostly manual process involving communication from the IP designer to the model coder. Worse yet, this is not a process that can be easily verified. Therefore, the desire for automation has been voiced by many designers.

Recently, Cadence Virtuoso Schematic Editor's Power Intent Export Assistant (VSE PIEA) implemented extraction of CPF macro model. This has not only met the needs of designers looking for automatic CPF macro model generation. It also allows extraction of related power pins from the CPF macro model boundary ports list.

The extraction is based on a fairly simple algorithm and is illustrated in Figure 5.

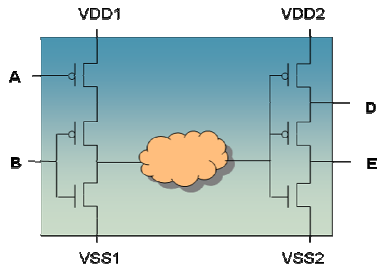

Figure 5. Fully Deterministic Boundary Ports

For each I/O pin, trace the circuit to power and ground pins to determine related power pin. In Figure 5, it is very obvious that pins A and B are related to VDD1/VSS1, and pins D and E are related to VDD2/VSS2.

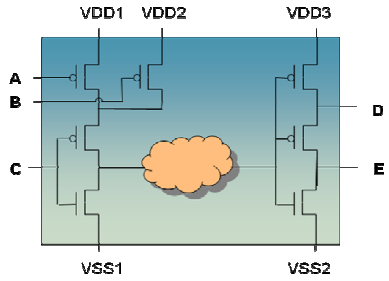

Figure 6. Partially Deterministic Boundary Ports

However, the circuit in Figure 6 is not so simple. Tracing to power and ground pins works very well for pins A, B, D, and E. But it is not clear if pin C should be traced to VDD1 or VDD2. The designer is prompted for input whenever the tool is not able to determine related power pins automatically.

3.2 CPF Macro Model

A CPF macro model uses standard CPF commands to describe the power intent of IP blocks, such as RAM, ROM, PLL, etc. The main difference when compared to a CPF design model is that a CPF macro model primarily addresses the power intent of the macro ports. However, when a CPF macro model is used with low-power simulation, it needs to specify the *internal instances of the simulation model* so that the simulator knows what should be corrupted during power shutoff.

Although the Liberty model has some low-power attributes, the CPF macro model is still required for two main reasons: 1) The CPF macro model contains key features, such as power modes, that allow more complete description and exhaustive verification of the IP block. 2) In many cases, the timing and power dissipation models in Liberty are not needed. All that is required is the simpler CPF macro model.

Low-power mixed-signal structural verification is a good example to demonstrate the benefits of CPF macro model. In this situation, the only concern is to verify the proper electrical interface of analog circuits with their surrounding analog and digital circuits. This can be done by using boundary port information to check voltage compatibility. Timing and power dissipation are not needed. As a result, the CPF macro model is much more concise. With automation from PIEA, CPF macro models have the additional advantage of being generated automatically.

```
set_cpf_version 1.1
set_hierarchy_separator /

set_macro_model macro_decoder

  create_nominal_condition -name ON -state on \
       -voltage 2.5

  create_power_domain -name PD_2p5V \
       -boundary_ports { wl_255 wl_254… wl_0 …
}
  update_power_domain -name PD_2p5V \
       -primary_power_net vdd \
       -primary_ground_net vss

  create_power_mode -name mode0 -default \
       -domain_conditions { PD_2p5V@ON }

end_macro_model
```

Figure 7. Sample CPF Macro Model

Figure 7 shows a sample CPF macro model generated by PIEA. It has boundary ports that are related to vdd and vss pins. This is sufficient to establish the power domain of all input and output macro ports. During low-power structural verification, Conformal Low Power (CLP)[6] will check for electrical compatibility of the driving and receiving power domains. One thing to keep in mind is that PIEA does not understand functional behavior of the macro. Therefore, it can only generate one power mode with the assumption that all power domains are on. If any of the power domains can be switched off or operate at different nominal conditions, this information must be manually added to the CPF macro model.

4. LOW-POWER MIXED-SIGNAL STRUCTURAL VERIFICATION

Low-power mixed-signal structural verification has been mentioned previously in this paper and discussed at length in [2]. One key benefit is that structural verification does not depend on test patterns or simulation coverage. There is also significant runtime advantage over low-power simulation.

Figure 8 illustrates an incompatible power domain crossing. The driver operates at 1.0 V and can be switched off. The receiver is expecting a 1.2 V signal. The receiver circuit will have excessive leakage regardless of whether the driver is on or off. This is one of the many failure mechanisms that CLP detects. By adding an enabled level shifter cell in Figure 9, the receiver gets the 1.2 V signal that it needs and is also protected when the driver is shut off.

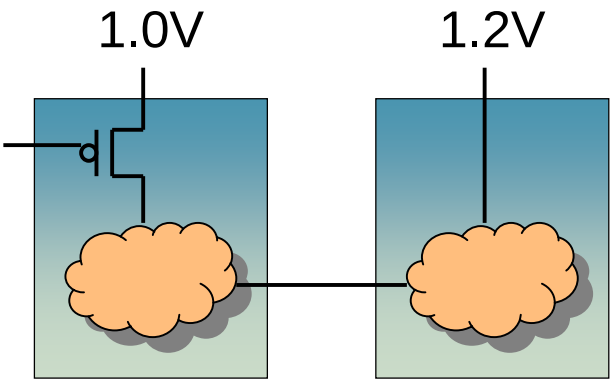

Figure 8. Incompatible Domain Crossing

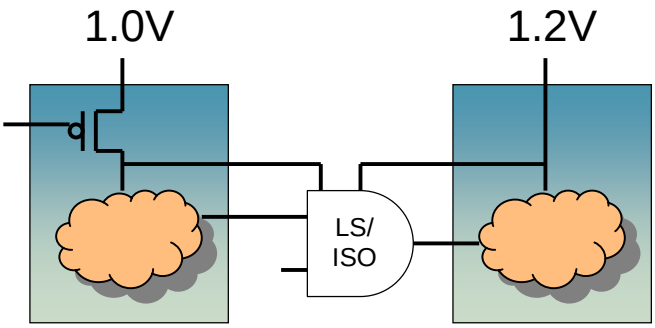

Figure 9. Compatible Domain Crossing

4.1 Automatic Extraction of CPF Design Model

In order to perform low-power structural verification, CLP must be directed by a CPF design model to associate power domains to power and ground pins, specify isolation or level shifting at domain crossings, etc. By visual inspection of Figure 9, we can see three power domains: 1.0 V always on, 1.0 V switchable, and 1.2 V always on. We also see an isolation rule, a level shifter rule, and a power switch rule. These can be manually coded to form the CPF design model.

PIEA can generate this CPF design model from the Virtuoso Schematic Editor environment. The user can register low-power cells such as level shifter, isolation cell and power switches by importing technology CPF file or entering this information through GUI interface. PIEA extracts the power domain information by associating power/ground pair to the corresponding design components. It will also create all of the rules associated with the low-power cells. Once the power structure of the design is extracted, the user can review and make any modification through GUI interfaces. Structural verification can be performed from the Virtuoso Schematic Editor (VSE) by directly invoking Conformal Low Power.

4.2 Detected Failure Mechanisms

Application of this verification method to real designs has uncovered many design errors.

Figure 10 shows an inverter that is powered by a 3V power but driven by 2V signal from an analog circuit. PIEA generated a CPF macro model for the analog circuit with a 2V boundary port. CLP

detected a 2V to 3V domain crossing without a level shifter cell. It issues an error to signal this problem.

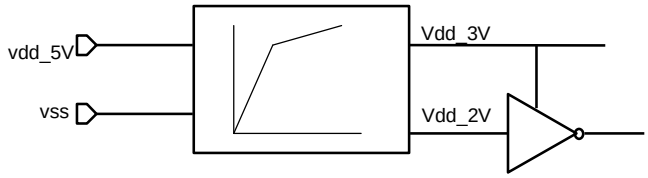

Figure 10. Missing Level Shifter

Figure 11 shows a level shifter's output power pin incorrectly connected to a 2V supply (vdd_2V) instead to the output of the LDO (vout_1p2V). CLP detected incompatible voltage levels between the level shifter output (vo_2V) and inverter input, which is expected to be driven by a 1.2V signal.

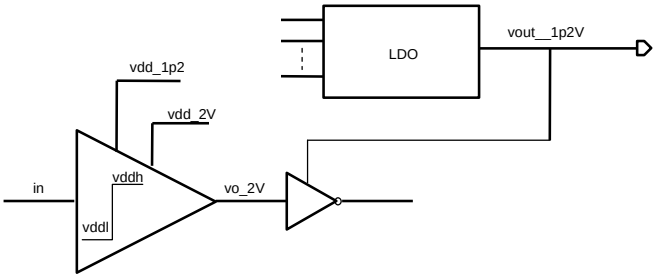

Figure 11. Incorrect Power Connection

5. LOW-POWER MIXED-SIGNAL SIMULATION

While low-power structural verification can detect many design errors that will result in electrical failures, it is not able to detect functional errors. Functional errors will result in incorrect logical behavior in digital logic or incorrect voltage/current in analog circuits. Some examples of functional errors are incorrect isolation output level, isolation enabled at the wrong time, and circuits shutting off at the wrong time. These problems can only be detected by simulation.

A non-power aware simulation can ensure correct functionality when all power supplies are turned on. This is always run before the low-power modes are enabled. In low-power modes, the simulator needs to perform special tasks to support power shut off and low-power cell insertion. For low-power mixed-signal designs, more consideration must be given to how analog circuits and signals need to behave in low-power modes.

This section introduces some of the basic concepts of power shutoff simulation for digital logic, analog transistor circuit, and analog behavioral model.

5.1 Digital Logic Shutoff

Under normal operation, digital signals have simulation values of 1 or 0. When a power domain is shut off, the simulator forces a value of X onto digital signals and registers. After powering up, the registers will continue to hold the X value until they are cleared by reset or loaded with a new value. In some cases, some registers need to retain their states before power shutoff and restore their states after power on. The simulator must be able to support this state retention operation. Also, a low-power digital simulator will often need to perform the isolation function between domain crossings, if the isolation cells are not instantiated in the RTL code but the isolation rules are specified in the power-intent file. This is similar in concept to the synthesis tool inserting isolation cells at domain crossings. The main difference is that the simulator does not write out a new design with the inserted isolation cells.

5.2 Analog Circuit or Behavioral Model Shutoff

Low-power mixed-signal simulation can to be run with the analog side either as transistor circuits or behavioral model and the digital side as RTL code or gate-level netlist. This flexibility allows the designer to achieve optimum trade-off between performance and accuracy. With the CPF file, power intent is specified at a higher power-domain level of abstraction. There is no need to connect power and ground nets or configure voltage sources. When a power domain is switched off, the intention is to switch off everything, analog or digital. Based on the power domain definition, the CPF-AMS simulator automatically drives the correct supply voltage to the analog blocks.

Figure 12 shows an example CPF file that defines a top-level switchable power domain (PD2), which is switched off when the macro_pse control signal is low. The analog circuit is described as a CPF macro model (macro_decoder) and instantiated with the set_instance command. We have seen this macro model in Figure 7. It contains a PD_2p5V domain, which is mapped to the top-level PD2. This means that PD_2p5V and PD2 will share the same power/ground connection and power up and down together. The user attribute option specifies the power/ground pins of the analog macros that CPF-AMS needs to drive. When PD2 is on (power mode mode_on), 2.5V is applied to the power pin. When PD2 is off (power mode mode_off), 0.0V is applied. These voltage values and conditions are all specified in the CPF file.

```
create_power_domain -name PD2 \
   -shutoff_condition !macro_pse \
   -base_domains PD_top \

update_power_domain -name PD2 \
   -user_attributes \
  {amscpf_power_supply {macro_inst.vdd}
   amscpf_power_ground {macro_inst.gnd}}

set_instance macro_inst –model macro_decoder \
   –domain_mapping {PD_2p5V PD2}

create_nominal_condition -name ON -state on \
       -voltage 2.5
create_nominal_condition -name OFF -state on \
       -voltage 0.0

create_power_mode -name mode_on -default \
       -domain_conditions { PD2@ON }
create_power_mode -name mode_off \
       -domain_conditions { PD2@OFF }

......
```

Figure 12. Power and Ground Connection for Analog Macro Model

5.3 Analog/Digital Interface

In the AMS simulator, the interface between analog and digital domains are translated by connect modules. They basically translate between digital 1/0 values and analog voltages. For CPF-AMS, the Connect Modules have been enhanced to be power aware. This means that they understand the power-down state and translate to/from the corresponding analog power-down voltage.

The power-aware Connect Modules include isolation clamping capability, which is defined in CPF. The isolation clamping is applied on the digital side of the crossing. On the analog side, signals can be clamped to a pre-defined value when the analog block is shut off.

5.4 Simulation Example

Figure 13 shows a sample low-power mixed-signal simulation waveform. The circuit generates a saw tooth waveform. The red waveform is the analog output, which is a combination of digital-to-analog converter output (driven by a counter) and an analog noise generator. In the first two cycles, both the digital counter and the analog noise generator are on. As a result, we have a noisy saw tooth. In the next two cycles, the analog noise generator is turned off and clamped to the value specified by the isolation rule in the CPF. This results in a clean saw tooth. In the next two cycles, the digital counter is turned off with outputs held (as specified by a hold-type isolation rule), and the analog noise generator is turned on. This results in a noisy flat line.

6. CONCLUSION

Low-power design requirements introduce significant verification complexity. Much has been done in the digital design space to address this verification complexity and minimize impact to overall design schedule. Some of the key components of this solution are power intent file, low-power structural verification, and state corruption for low-power simulation. These components must be migrated to the analog/mixed-signal design space. To facilitate such migration, automation has even been extended to extraction of power intent from circuit schematic.

Application of this low-power verification solution to mixed-signal designs have shown excellent results and identified many design errors prior to tapeout. By applying the CPF macro model and analog behavioral model abstractions, the overall runtime of low-power mixed-signal verification is also reduced. This will lead to much needed improvement in quality and efficiency for mixed-signal designs.

7. ACKNOWLEDGMENTS

We would like to acknowledge the assistance and contribution by Qingyu Lin, Amit Chopra, and Pei-Der Tseng.

8. REFERENCES

[1] Neyaz Khan, Mixed Signal Verification of Dynamic Adaptive Power Management in Low Power SoC. In DVCON (San Jose, Calif., Feb. 22 – 25, 2010).
[2] Luke Lang, Case Study: Power-aware IP and Mixed-Signal Verification. In DVCON (San Jose, Calif., Feb 28 – March 3, 2010).
[3] Si2 Common Power Format Language Reference. Version 1.1. August, 2010.
[4] Accellera Unified Power Format, Version 1.0, February 22, 2007.
[5] IEEE Standard 1801-2009, March, 27, 2009.
[6] Cadence Encounter Conformal Low Power User Guide, October, 2011.
[7] Qingyu Lin, Mixed Signal Simulation Workshop, Low Power Simulation Extension, Jan. 2012.

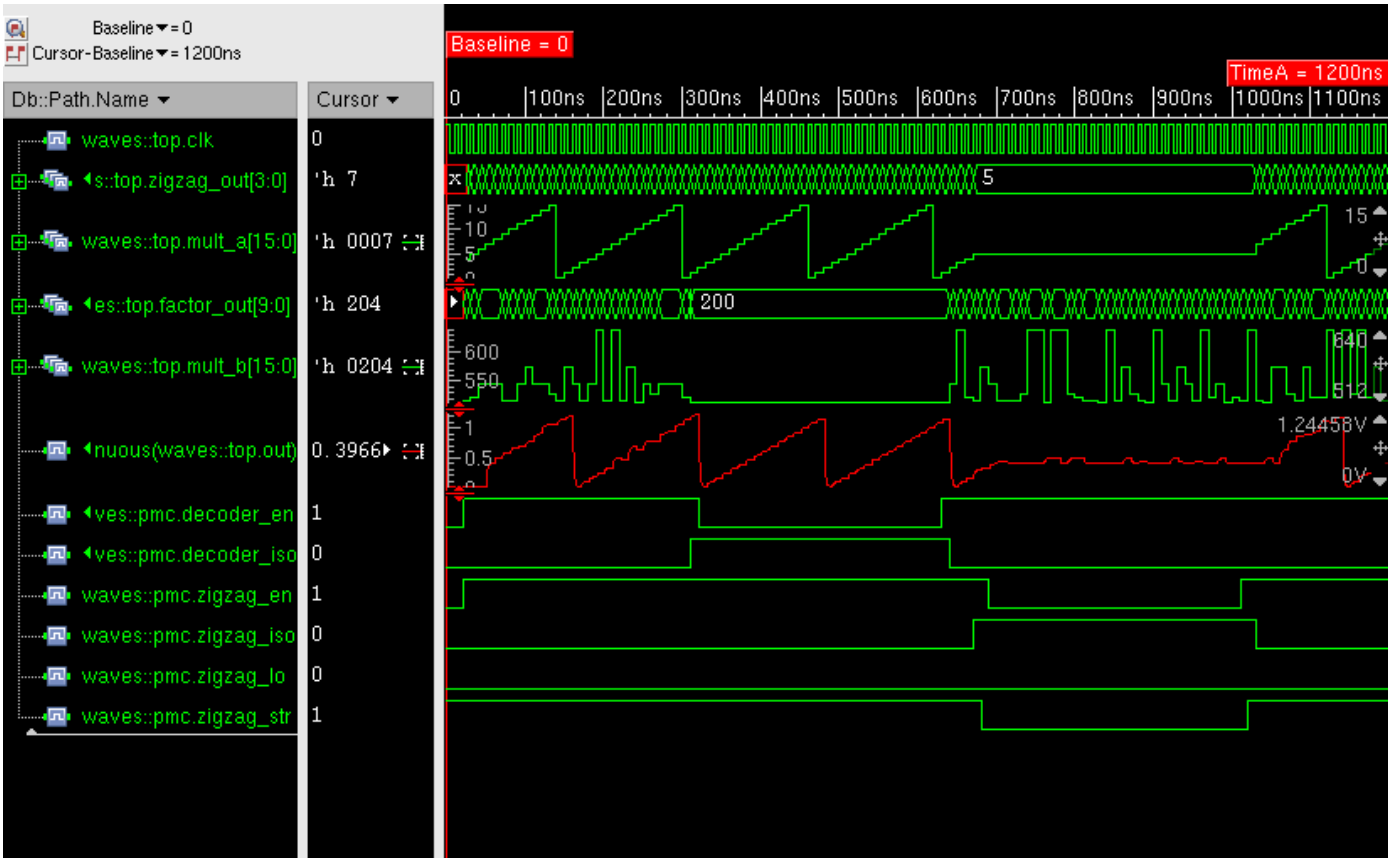

Figure 13. Sample Low-Power Mixed-Signal Simulation Waveform