Case Study: Power-aware IP and Mixed-Signal Verification

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

ABSTRACT

Power intent verification, whose complexity increases exponentially with the number of power domains and the number of different power states those domains can assume, is further complicated by the need to integrate digital and mixed-signal IP blocks. Digital IP blocks may be complex enough to have their own advanced low power techniques implemented internally. For mixed signal designs, Vdd and Gnd supply signals are treated very differently at each side of the digital-analog boundary. Advanced low power designs are in a way always mixed signal designs, given the analog components used to implement various flavors of power regulators.

There are many possible errors that can creep into advanced low power designs – reversed polarity of switch controls, incorrect connection of level shifters, missing isolators between power domains, and buffers of global wires crossing power domains being powered from the wrong supply. Any of these errors can elude traditional functional verification but still prove fatal to the chip, causing re-spins or even expensive field recalls.

This paper first explains the design techniques deployed to reduce power, including multiple power domains with power shut-off (PSO) and multiple supply voltages (MSV). The problems of integrating a power-aware IP block into the design's overall power intent are discussed next. Macro modeling is introduced as a solution to adequately capture the block's power intent. Recent extensions to power-aware simulation and formal verification of structural power intent in mixed signal designs are explained after that. Finally, the paper concludes by describing the structural power intent errors introduced across the analog-digital boundaries, how these were found and fixed, and how it saved the customer a design re-spin.

1. INTRODUCTION

The requirement for low-power design is no longer the burden of a few specialized semiconductor companies. Today, almost all semiconductor companies must meet very stringent power requirements for their products. Some of the reasons behind this movement to low-power design are: 1) Shrinking process geometry has resulted in greater integration of circuits than before. The power that used to be dissipated by several chips is now dissipated by one chip. 2) Shrinking process geometry results in higher leakage power. We are seeing leakage power dominate dynamic power, and leakage power is continuously dissipated even when the chip is operating in idle mode. 3) The semiconductor industry has been shifting focus from enterprise computing and communication to consumer electronics, where battery life is extremely important. 4) Increasing awareness to protect the environment and reduce global warming has led to regulations to limit energy consumption.

Before this movement toward low-power design, power dissipation was not a primary concern. Once timing and area objectives were met, power reduction was almost an afterthought using techniques such as clock gating and multi-Vt cells. More advanced low-power techniques, such as multiple-supply voltage (MSV) and power shut-off (PSO), were well known but very difficult to implement. Most designers feel that implementing power shut-off will increase design complexity by 2 to 4 times. Most designs did not have very stringent power requirements. Therefore, designers did not implement these advanced low-power techniques. A few designs, notably the cellular phone chips, had very low power requirements. The design teams had to implement these advanced low-power design techniques, and they paid the price of increased design complexity.

With an entire industry shifting toward low-power design, it is no longer feasible to pay the price of increased design complexity because very few companies can afford to double or quadruple the number of designers. Certainly, no company is able to double or quadruple the design schedule. Today, low-power design techniques are not only applied to most SoC designs. They are also applied to many IP and analog/mixed-signal designs, which must be designed and verified as a stand-alone circuit as well as an integrated portion of the SoC. Much effort has been focused on automating the complex steps in design and verification of low-power SoC, but very little has been done in the area of low-power IP and analog/mixed-signal designs.

This paper will explore some of the commonly used low-power design techniques and introduce the concept of low-power structural verification. This is a powerful and efficient verification technique that identifis electrical problems in a low-power design. In order to apply structural verification, the power intent and features of IP blocks must be described. The Common Power Format (CPF) macro model [1] has been developed for this purpose. This paper will then show how macro model and structural verification can be extended to check low-power features in analog/mixed-signal designs. Finally, this paper describes the experience of applying this verification technique to an analog/mixed-signal design

2. LOW-POWER DESIGN TECHNIQUES

Power dissipation can be broadly categorized into dynamic and leakage power. As the names suggest, dynamic power is dissipated when the circuit is switching, and leakage power is dissipated constantly, regardless of switching activity, like a leaky faucet. [2]

The majority of the Low-power SoC designs today employ multiple supply voltage (MSV) and/or power shut-off (PSO). This section will explain these advanced low-power techniques and examine their complexities during design and verification.

There are many other low-power design techniques, such as Dynamic Voltage Frequency Scaling (DVFS), that are being implemented to reduce power. Other than DVFS, they will not be

covered in this paper because most of them are quite complex, and each deserves an entire paper to address. MSV and PSO will be sufficient to illustrate the verification of low-power IP and analog/mixed-signal designs.

2.1 Multiply Supply Voltage (MSV)

It is well known that the power dissipation due to switching circuits is proportional to the square of the supply voltage.

$$P_{switching} = TR \bullet F \bullet C_{load} \bullet V_{dd}^{\ 2}$$

Where TR is the toggle rate, F is the frequency, C_{load} is the capacitance loading, and V_{dd} is the supply voltage.

We can reduce switching power by reducing any of the above components, but reducing the supply voltage has the greatest effect due to the quadratic relationship. In addition to reducing switching power, reduced supply voltage will also decrease leakage power. Unfortunately, the speed of logic gates is also reduced with a decrease in supply voltage. As a result, we can only reduce the supply voltage to blocks of logic that are not timing critical (have plenty of positive slack). The logic gates that are on the critical path must be powered by a higher supply voltage to maintain performance. The net effect is that logic gates on the same chip are operating at different supply voltages, and there are voltage boundaries where logic gates with different voltages must interface.

2.1.1 Level Shifter

When logic gates of different supply voltages interface with each other, a level-shifter cell needs to be inserted between these two logic gates to ensure that the driving signal is compatible with the voltage of the receiving gate. In some cases, high-to-low voltage crossing may not require a level-shifter. In almost all cases, low-to-high voltage crossing will require a level-shifter because the driving cell may not drive the receiver input to a high enough voltage to switch off the PMOS transistor. This leads to leakage current.

A level shifter typically has two power pins – one for input and another for output. The input power pin is connected to the power supply of the driving domain, and the output power pin is connected to the power supply of the receiving domain. This ensures that the level shifter is able to receive the signal from the driving domain and drive a compatible signal to the receiving domain.

2.1.2 MSV Considerations

Some designs have different power sources and grids that operate at the same voltage. These power sources may not track each other so that one source could be operating at +10% nominal voltage while the other is operating at -10% nominal voltage. For these cases, it may be necessary to insert a level-shifter between these two voltage domains to eliminate leakage current.

MSV is one of the easier low-power techniques to implement. It involves the insertion of the level-shifter cell and the construction of the correct power grids. This requires moderate effort and minimal area overhead. The gate delay associated with a level shifter is normally not an issue since critical paths are almost never found at a voltage crossing.

While MSV is fairly straight forward to implement, it does have its share of challenges. MSV designs need to be verified to ensure correct chip operation and minimal power dissipation. Some of the common problems include missing level shifter and incorrect power/ground connection. MSV cannot be deployed if the different supply voltages are not available either from the board or through voltage regulators. One designer reported that the power savings from MSV was offset by loss of regulator efficiency at the desired voltages. Optimal partition of blocks and voltage assignment may be difficult to determine. It may require multiple trial synthesis runs to find the right balance between power and timing. However, advanced synthesis tools will help to automate this process.

MSV should be deployed only when the block has a reasonable size. When the block is too small, the power benefit does not justify the effort. Lots of small voltage domains can also create routing obstructions and placement issues.

2.2 Power Shut-off (PSO)

Very often, blocks of logic within a chip are not used for some period of time. Clock gating can be used to eliminate dynamic power, but the CMOS gates continue to dissipate leakage power. And this leakage gets worse at lower process geometry. The only solution is to switch off the power (or ground) to these logic gates when they are sitting idle. This technique is also known as MTCMOS. (This paper assumes header switches that shut off the power.)

2.2.1 Isolation

PSO is a simple idea but is considerably more complex to implement and verify than MSV. Figure 1 shows two inverters on either side of an off-to-on domain crossing.

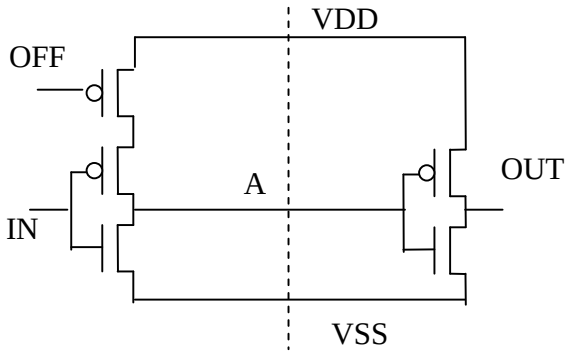

Figure 1. Off-to-On Domain Crossing

When the OFF control is asserted high, the driving inverter is switched off, and its output (net A) will be floating. This turns on both PMOS and NMOS transistors of the receiving inverter and causes short-circuit current to flow between VDD and VSS. An isolation cell must be inserted on net A to drive a high or low signal to the receiving inverter to ensure that either the PMOS or NMOS transistor is shut off.

Most of the libraries contain a wide array of isolation cells to isolate high, low, or even hold the previous value. They also have enabled level shifters that combine the isolation and level shifting functions into one cell. In the rare event that a library does not have isolation cells, combinational logic gates can be used for the isolation function. However, it is extremely important to make sure that the data input of the logic gate does not have an unprotected buffer or inverter. Otherwise, there will be an off-to-on domain crossing similar to what is shown in Figure 1.

2.2.2 State Retention

A block that is switched off can have important control or status registers that must not lose their state. These registers can be implemented with a state-retention flop to hold the state even when the primary power is shut off. The main advantage of using a state-retention flop is that the state can be restored quickly.

The main disadvantage is that state-retention flops are larger in area and consume more power during functional mode than standard flops. With this in mind, state-retention flops are used sparingly. If there are lots of data to retain through a PSO cycle, the data can be scanned out to another memory prior to powering down. Another method is to use standby mode, which will be described later.

The main concern with using state-retention flop is not using enough of them. In this case, vital data is lost during PSO, and the SoC will not function after powering up. This must be verified with low-power simulation.

2.2.3 Power Switches

Shutting off the power to a power domain can be done either internally within the chip or externally. When power is switched within the chip, power switches need to be added. They are inserted in the design during placement. However, with the intent for power switches (internal or external) specified in the power intent file, all of the frontend tools have an understanding where the power switches will be inserted. Therefore, PSO domains can be verified even in the RTL stage.

There are many types of power switches and many methods for controlling them. The main considerations for power switches are number of power switches, power ramp up and down time, dynamic IR drop, and rush current.

2.2.4 Feedthrough

Feedthrough is required when the path around a power domain is too long. Therefore, feeding a signal through a power domain is often the preferred method. This applies to both MSV and PSO designs.

In Figure 2, signal A from PD1 is routed around PD2 and requires lots of buffering. Signal B is fed through PD2 and gets to the destination with fewer buffering and less delay.

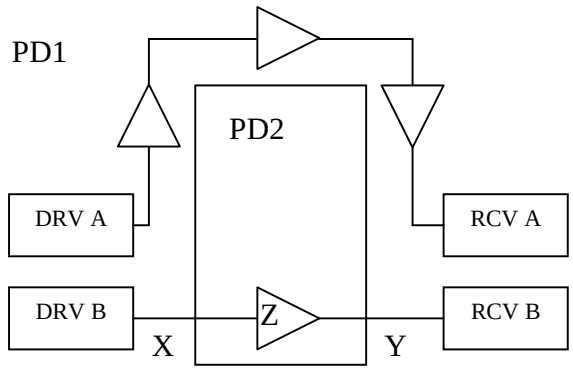

Figure 2. Feedthrough example

Signal A and all of its buffers are clearly in PD1. Signal B, however, crosses into PD2 at point X and exits back to PD1 at point Y. Does this mean that there are two domain crossings? The answer depends on how buffer Z is powered. If buffer Z is powered by PD1's power net, then there is no domain crossing. If buffer Z is powered by PD2's power net, then there are two domain crossings.

In order to be powered by PD1's power net, buffer Z must be a special always-on (AON) buffer. This buffer has a primary power pin that is only a follow pin (to connect the power grid). The primary power pin does not connect to any transistors inside buffer Z. These transistors are powered by a secondary power pin that is connected to PD1's power net.

The name "always-on buffer" is a historical term and is not always accurate. Consider a switchable PD1 and AON PD2. In this case, buffer Z is powered by PD1's power net and is off when PD1 is off. Therefore, buffer Z is not always-on. A more accurate term might be "on when necessary." However, that is too difficult to say, so a new term "global cell" has been actively used.

If buffer Z is powered by PD2's power net, then we have two domain crossings. If PD1 and PD2 are AON but operate at different voltage levels (MSV), then level shifters need to be placed at point X and Y. If PD1 is AON and PD2 is PSO, then we have an off-to-on domain crossing at point Y, and an isolation cell is needed there. If PD1 is PSO and PD2 is AON, then we have an off-to-on domain crossing at point X, and an isolation cell is needed there. If PD1 and PD2 are both independently PSO, we need isolation cells at both point X and Y.

2.2.5 Standby Mode

Standby mode is a special combination of PSO and MSV. When in standby mode, the supply voltage is lowered to a point that will allow all memory elements to retain their contents but not high enough to switch any circuits. It is typically used when contents of RAM's need to be retained or when state retention registers are too expensive in terms of area or power.

Since the standby mode voltage is not high enough to support circuit switching, isolation cells must be inserted at the inputs of the power domain to ensure that the circuits inside the standby domain will not switch.

2.2.5 PSO Considerations

Verification and implementation complexities increase drastically when PSO is used. This will be discussed in the next section. Some commonly seen problems are missing isolation cells, missing state-retention flops, and incorrect power control and sequencing. If there are multiple PSO domains operating independently, then the concept of "more on" needs to be considered in addition to AON and OFF. The tradeoff between these complexities and power savings must be analyzed carefully. The key to a successful PSO design is verifying the design early and often.

2.3 Dynamic Voltage Frequency Scaling (DVFS)

DVFS is basically dynamic MSV. The supply voltage and clock frequency are adjusted on-the-fly to meet the performance requirement of the system at any given time. When high performance is required, the supply voltage is ramped up and then the clock frequency is increased to meet the system throughput requirement. When high performance is not required, the clock frequency is decreased and then the supply voltage is reduced to save power. Sometimes, a DVFS block can be switched off for a period of time. In this case, PSO also comes into play.

DVFS is commonly used in processor design, and adjustments are made based on system demand. It provides the power savings of MSV but also provides high performance when needed. DVFS is very complex to implement and usually requires a voltage regulator.

Functional and timing verification are also extremely complex. Transition between power modes, especially when the supply voltage is changing, may need special verification. Due to high cost of implementation, careful analysis of benefit/cost tradeoff must be made.

For some designs, the power savings of DVFS outweigh the design complexity. Therefore, even some mixed-signal designs are implementing DVFS. This poses special verification challenges and requires verification methodology that is beyond the scope of this paper. [3]

3. LOW-POWER VERIFICATION

While low-power cell insertion is not a trivial task, low-power verification is an even more difficult task. Excessive leakage power due to incorrect low-power implementation is one of the main reasons for design re-spin. The number of power states, power modes, and power mode transitions increases exponentially with the number of power domains. Therefore, a relatively simple low-power design with a few power domains could easily become an order of magnitude more difficult to verify.

Traditionally, verification is synonymous with functional simulation and requires a team of verification engineers. It is only as good and complete as the testbenches written. Even the fastest simulators run several orders of magnitude slower than the design being simulated. So, runtime is another limitation. In order to reduce reliance on functional verification, many tools have been developed. For example, code coverage provides a metric for completeness of verification. Static timing analysis (STA) has virtually eliminated the need for timing-annotated simulation. Formal equivalence checking has reduced the need for gate-level regression.

For low-power designs, there are two types of verification – functional and structural. Functional verification must be power-aware in order to uncover logical errors in the design. Structure verification examines the interfaces at power domain boundaries to determine if there are any electrical problems. Finally, we shall see that formal equivalence checking also needs to be power-aware.

3.1 Low-Power Structural Verification

Many of the low-power design requirements are structural in nature. For example, every off-to-on domain crossing needs to be protected by an isolation cell, and every low-to-high voltage domain crossing needs a level shifter. Low-power structural verification has been developed to validate the existence of appropriate low-power cells at every domain crossing. Taking advantage of these structural requirements, low-power verification can be done independent of simulation pattern, which means that it runs much faster than low-power functional verification.

Low-power structural verification can also verify consistency of power intent against the low-power structure in the design. For example, if an isolation rule is not specified between off and on domains, low-power structural verification will identify that a required isolation rule is missing. This is known as power intent file quality check (QC).

There is quite a bit of overlap between low-power structural verification and functional verification in terms of errors detected. The following sections will explore some of the overlaps as well as the non-overlaps.

3.2 Low-Power Functional Verification

Low-power functional verification merges traditional logic simulation with special handling for low-power cells and states. It is able to mimic the behavior of low-power designs to ensure correct operation. In the past, these kinds of simulation required lots of manual effort, such as writing PLI routines, creating low-power models, writing special testbench force/release commands, etc. Today, power-aware simulators are able to read in the power intent file and automate all of these low-power functions.

Here are some examples of what a low-power simulator can do. The RTL does not have any isolation cells and state-retention flops instantiated. Based on the power intent file, a low-power simulator will automatically infer these low-power cells and perform their functions. When a power domain is shut off, the low-power simulator will force a logical X-state on all cells and nets. This is commonly known as corruption during PSO.

Section 2.2.1 provided a reason for needing isolation cells. Another interpretation based on low-power simulation is that isolation cells prevent the propagation of X (corrupted state) into the logic that is on. If a design is missing an isolation cell and the corresponding power domain is shut off, then X will most likely be propagated to the rest of the chip resulting in a failed simulation. When the verification engineer examines the simulation waveform, he will see that most of the signals at an X state. He will then need to debug by tracing back to the source of the X, which is a PSO domain output that was not isolated. This is a very inefficient way to debug this problem because of simulation runtime and debugging time. Low-power structural verification will identify this error without simulation. This is one example of the overlap between low-power structural and functional verification.

However, there are many low-power errors that cannot be detected structurally. For example, using an isolate-low cell instead of an isolate-high cell is a functional problem, not a structural problem. One must run a low-power functional simulation to detect the error resulting from the wrong isolation state. Low-power structural verification only recognizes an off-to-on domain crossing and the existence of an isolation cell. It does not have the ability to know the correct isolation state.

Another common error that requires low-power functional verification is missing state retention flop. This is typically a power intent file specification error. If a flop is not specified as state retention, its state will not be saved and restored during a PSO cycle. After state restore, this flop will still be in the X-state. This X-state will propagate and hopefully result in a detectable error.

These simple examples illustrate that both low-power structural and functional verification are necessary. Always run low-power structural verification first to ensure that the power intent file is correct and to catch the easy problems. Clean up the structural errors before running low-power functional verification to find the logic errors. This is the most efficient approach.

3.3 Low-Power Equivalence Checking

Most low-power designs start with RTL without isolation cells. The isolation cells are inserted by the tools based on power intent specification. During equivalence checking, the isolation cells in the netlist cannot be matched in the RTL. This causes a non-equivalence if the equivalence checker is not power-aware. To eliminate the error, some designers add pin constraints to the isolation control pin to open up the isolation cell. This works well for some designs but cannot detect all errors.

Figure 3 shows two circuits that are logically equivalent when pin constraint of 1 is applied to EN. However, when the isolation cells are enabled (EN = 0), the flop A sees 0 when flop B sees 1 at their inputs.

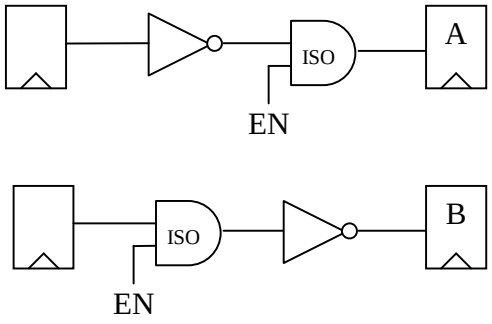

Figure 3. Low-Power Equivalence Checking 1

In Figure 4, the top circuit has a net that feeds through PD2. On the bottom, a regular buffer has been inserted in the switchable domain PD2. This creates an off-to-on domain crossing from PD2 to PD1 that needs to be isolated. Once again, when pin constraint of 1 is applied to EN, the two circuits are equivalent. But when the isolation cells are enabled (EN = 0), the feedthrough value does not get through PD2 and the isolation cell in the bottom circuit.

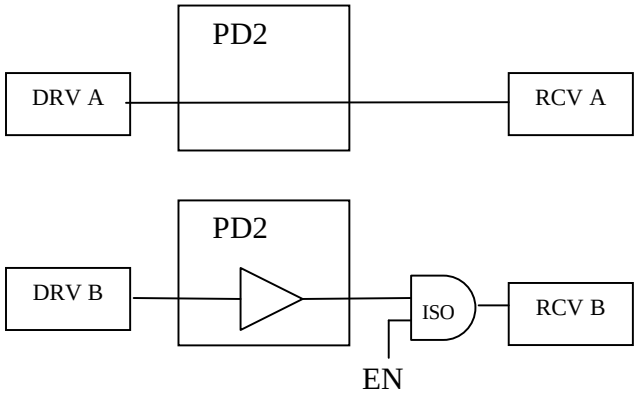

Figure 4. Low-Power Equivalence Checking 2

These two examples illustrate the importance of power-aware equivalence checking.

3.4 Closed-Loop Verification

The power intent file is the power architecture specification. It drives the implementation tools to insert, place, and connect low-power cells. Since it has a direct impact on the final design, it must be treated with the same respect as RTL. It must be verified to be correct before driving implementation and then used as the golden reference to verify the implementation. All designers should be very familiar with this concept. RTL is first simulated to verify correctness. Then it is synthesized into a logical netlist. Finally, it is used as the golden reference for equivalence checking against logical and physical netlists. If the RTL is changed at any time, the simulation, synthesis, and equivalence checking steps must be repeated.

Similarly, power intent file must be verified by low-power structural and functional verification. After driving the implementation tools, it is used as the golden reference for low-power structural, functional, and equivalence verification. This closes the verification loop.

The key message in this entire section is that power-aware verification is critical. Mistakes can be made in every step of the design flow, including the power intent file, and especially in some manual steps, such as ECO. Applying power-aware verification at every step in the design flow will ensure that mistakes are caught early to prevent schedule delay and costly re-spin.

4. LOW-POWER IP

With shrinking geometry, today's chips can fit a lot of transistors. This leads to highly integrated SoC with lots of functions, multiple cores, and multiple interfaces. High level of integration necessitates the use of IP. Examples of these IP are RAM, PLL, voltage regulator, and high-speed interfaces. Some designs even have repeated blocks that are placed and routed separately and used as an IP in the SoC.[4] Increasingly, IP blocks are also designed with advanced low-power techniques for low-power application. For example, there are RAM's with multiple power domains, built-in power switches, and isolation cells. The memory core can be shut off separately from the peripheral logic.

In order to use low-power IP effectively, it is essential to model an IP's low-power features so it can be verified with the rest of the SoC. Most IP blocks are delivered with various models, such as simulation, Liberty, and LEF. These models were conceived many years ago and are not designed to model low-power IP. Liberty has been enhanced in recent years to include some low-power attributes, but it is still not adequate. That is why CPF has the macro model concept. An IP is a black box, and its circuits inside are invisible. CPF macro model describes the power intent of the IP.

4.1 CPF Macro Model

Since low-power structural verification primarily analyzes domain crossings, we must be able to model the power domain of each IP block's input and output data pin. When the driver and receiver's data pins are incompatible, low-power structural verification will signal an error. Therefore, at a very minimum, we must be able to describe the relationship between data pins and power/ground (PG) pins. This is known as related PG pin.

For any standard cell gate, there is only one pair of PG pins. All input and output data pins are related to that one pair of PG pins. There is no need to model this. When there are two pairs of PG pins, the situation is not so simple. Figure 5 shows a simple pad cell. There are two data pins (A and PAD) and two pairs of PG pins (VDD/VSS and VDDG/VSS). (Note – A power (or ground) pin can be paired with more than one ground (or power) pin to form more than one power domain.)

Without any modeling information, it would not be possible to know whether data pin A is related to VDD/VSS pair or VDDG/VSS pair. Using CPF macro model, the power intent of this cell can be easily described in Figure 6.

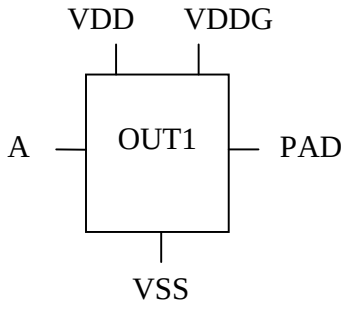

Figure 5. Simple Pad Cell Example

```
set_macro_model OUT1
  create_power_domain -name PD_core \
    -boundary_ports {A}
  create_power_domain -name PD_pad \
  -default -boundary_ports {PAD}

  update_power_domain -name PD_core \
    -primary_power_net VDD \
    -primary_ground_net VSS
  update_power_domain -name PD_pad \
    -primary_power_net VDDG \
    -primary_ground_net VSS

  create_nominal_condition -name high \
    -voltage 3.3
  create_nominal_condition -name low \
    -voltage 1.0

  create_power_mode -name MMon \
    -domain_conditions \
    {PD_core@low PD_pad@high} -default
end_macro_model
```

Figure 6. Pad Cell Macro Model

The macro model first establishes that there are two power domains associated with the cell. Each power domain is then associated with boundary ports, PG pins, and nominal voltage. Finally, the legal power modes for all power domains are specified. Any tool that reads the macro model will understand that data pin A is related to PG pins VDD/VSS and data pin PAD is related to PG pins VDDG/VSS. VDD operates 1.0V always-on, and VDDG operates at 3.3V always-on.

(Note – Liberty has since added the related_power_pin and related_ground_pin attributes to describe this relationship. However, it cannot model some of the other low-power features described later.)

Once a macro model has been defined, it needs to be instantiated in the CPF file to associate the power domains in the macro model to the power domains at the top level. Figure 7 shows instantiation and domain mapping for macro model. PD_core in the macro model is mapped to TOP. This established that TOP should also be a 1.0V domain and that data pin A is in domain TOP. Furthermore, the VDD/VSS PG pins must also be connected to the primary PG nets of domain TOP. Similar relationship applies to domain IO and PD_pad. If any of these are inconsistent, low-power structural verification will flag an error.

Figure 7 assumes that the macro model defined in Figure 6 is in the out1.cpf file.

```
set_instance ioring/inst1 –domain_mapping \
  ((PD_core TOP) {PD_pad IO})
include out1.cpf
```

Figure 7. Macro Model Instantiation

As mentioned earlier, many low-power IP blocks have built-in isolation, power switches, feedthrough ports, and analog signals that should not be connected to any digital level shifter or isolation cell. These can all be supported by CPF macro model (but not by Liberty).

Let's expand on the simple pad cell example in Figure 5 and add an isolation cell for data pin A. Figure 8 shows a block diagram of this circuit.

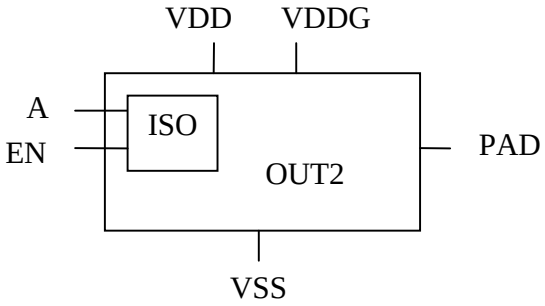

Figure 8. Built-in Isolation Example

Figure 9 shows that only a few lines of CPF code need to be added to the macro model in Figure 6 to describe this cell. The EN pin needs to be added to the boundary port list of PD_core. An isolation rule is added to specify that data pin A is protected by an isolation cell inside the macro.

```
set_macro_model OUT2
  create_power_domain -name PD_core \
    -boundary_ports {A EN}

...

  create_isolation_rule –name iso1 \
    –to PD_core -pins { A } \
    -isolation_condition {! iso_en}
end_macro_model
```

Figure 9. Built-in Isolation Macro Model

It is very important to know whether an IP block has built-in isolation. If there is no built-in isolation, then an external isolation must be inserted to protect off-to-on domain crossing. If there is a built-in isolation, there is no need to waste area and power by inserting another isolation cell. Low-power structural verification will examine the macro model and determine if there are any domain crossing violations.

Figure 10 shows a cell with only one pair of PG pins but also a built-in power switch. Because there is an internal power switch, another power domain must be defined. The macro model becomes slightly more complex because the switchable domain can be either on or off. Therefore, we have to define additional nominal voltages and power modes.

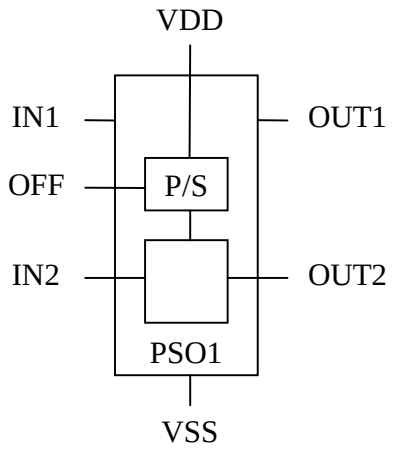

Figure 10. Built-in Power Switch Example

Figure 11 is the complete macro model for the cell shown in Figure 10.

```
set_macro_model PSO1
  create_power_domain -name PD_core –default \
    -boundary_ports {IN1 OUT1}
  create_power_domain -name PD_pso \
    -boundary_ports {IN2 OUT2} \
    -shutoff_condition {OFF} \
    -base_domain {PD_core}

  update_power_domain -name PD_core \
    -primary_power_net VDD \
    -primary_ground_net VSS

  create_nominal_condition -name on \
    -voltage 1.0
  create_nominal_condition -name off \
    -voltage 0

  create_power_mode -name MMon \
    -domain_conditions \
    {PD_core@on PD_pso@on} –default
  create_power_mode -name MMoff \
    -domain_conditions \
    {PD_core@on PD_pso@off}
end_macro_model
```

Figure 11. Built-in Power Switch Macro Model

Since the OUT2 pin does not have built-in isolation, it needs an external isolation if it is connected to a “more-on” power domain.

This section has provided some basic macro model constructs and ideas that will be referenced in a later section. Other commonly used CPF macro model commands are set_wire_feedthrough_ports and set_floating_ports, which is used to model analog pins. There is much more, but a thorough discussion of CPF macro model is beyond the scope of this paper. Please refer to the CPF Language Reference Manual [1] for a complete description.

One final comment about CPF macro model is that it can be used to represent a hierarchical block. This is similar to the Interface Logic Model (ILM) concept. When a hierarchical block is placed and routed, it can be treated as an IP block with a CPF macro model. This will reduce low-power structural verification runtime significantly.

4.2 Low-Power Simulation Model

For low-power functional verification, it would be most ideal if the existing simulation models can be re-used without modification. This is the case when using the Cadence Incisive Enterprise Simulator. It is able to perform PSO simulation with models that do not have PG pins. When using other power-aware simulators, one must code a power-aware simulation model that is based on the state of the PG pins. Figure 12 shows an example of a power-aware buffer model with PG pins.

```
module buf (A, Y, VDD, VSS);
  input A, VDD, VSS;
  output Y;
  assign Y = (VDD==1 && VSS==1) ? A : 1’bx;
endmodule
```

Figure 12. Power-Aware Simulation Model

On the surface, it appears that simulation with PG pins provide more coverage. However, the reality is that it does not provide any more coverage, if low-power structural verification is run first. This is because PG pin connection is inherently a structural issue. An incorrect PG connection will create additional domain crossing. Figure 13 shows an example.

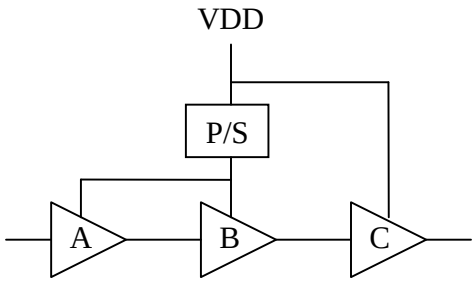

Figure 13. PG Pin Connection Error

Buffers A, B, and C should be powered by the output of the power switch. However, buffer C is incorrectly connected to the AON VDD. This creates an off-to-on domain crossing between buffer B and C. Since there is no isolation cell between buffer B and C, low-power structural verification will flag an error, without having to run timing-consuming simulation and debugging.

The problems with running power-aware simulation model are: 1) A physical netlist is not available until very late in the design cycle. 2) Gate-level simulation is notorious for being runtime intensive. Most design teams run very little if any gate-level simulation. As a result, there is very limited coverage. Low-power structural verification is the most efficient method to detect PG connection problems.

5. LOW-POWER MIXED-SIGNAL DESIGNS

For analog/mixed-signal designs, simulation is the only way to verify the design. There are commercially available tools that will enable mixed-signal simulation. They merge the digital logic simulator with some sort of SPICE analog simulator. On the analog simulation side, SPICE is slow and cannot handle large designs, such as an entire block of analog circuits. To improve on speed and capacity, analog behavioral model can be used instead of the transistor

circuits. This is faster but has questionable coverage. (At this point, we are not dealing with the accuracy of analog simulation.)

Leakage current is not easy to simulate. It takes a while for the driving node to float to a value that turns on both PMOS and NMOS transistors. The designer must be looking for this increased leakage current. Often analog behavioral models do not model leakage current. Therefore, off-to-on domain crossing may not be detectable in functional verification.

As we have seen, low-power structural verification provides significant advantages over low-power functional verification for detecting the most commonly occurring low-power errors in digital SoC designs. The same advantage can be realized with analog/mixed-signal circuits. This section documents how low-power structural verification, along with the power intent file and CPF macro model, was applied to a mixed-signal design and detected errors that were not found by low-power simulation.

5.1 Setup

This case study is based on a large SoC with several mixed-signal blocks and several large blocks of digital logic. Each mixed-signal block contains analog circuits and synthesized control logic. One of the mixed-signal blocks was taken through detailed low-power verification. This involved low-power structural verification with macro models for the analog circuits and functional verification with analog behavioral models. At the top level, full-chip low-power structural verification was performed with macro models representing the mixed-signal blocks. This step was very straightforward because we only had to code macro models for the mixed-signal blocks using steps outlined in section 4.1. It mostly consisted of identifying related PG pins and any built-in isolation. Since a top-level CPF file was already available, low-power structural verification was performed easily with Conformal Low Power (CLP).

Low-power verification at the mixed-signal block level was far more involved. The analog circuits were captured in Virtuoso schematics. From these schematics, a verilog netlist was generated. This netlist contained black boxes for each of the analog circuits and the synthesized netlist of the control logic. Behavioral models were already created for the analog circuits for functional simulation. However, macro models had to be created for structural verification. While this is a very straightforward process, the large number of analog circuits made this a very tedious process.

After generating all the models, we still needed a top-level CPF file to drive verification. For the mixed-signal block, the golden power intent is in the schematics because they have all the low-power cells instantiated and all the PG pins connected. This is very different than the digital logic world, where the CPF file is first created and then used to drive low-power cell insertion and PG connection. Therefore, CPF file creation became an exercise of reverse engineering the power architecture based on the analog schematics.

CPF file coding became a very manual and timing consuming exercise. We had to start with all of the primary PG pins and create power domain for each valid PG pin pair. Each PG pin pair had to be traced to establish the domain mapping of each macro model (analog circuit) connected to the PG pin pair. If the power net is connected to a power switch, then the output of the power switch became another power domain, and this process continued. We also had to find all the power switches, isolation cells, and level shifters to code the corresponding low-power rules. Something that took this much manual effort was not surprisingly full of mistakes.

Fortunately, CPF QC is part of Conformal Low Power. CLP flagged many inconsistencies between the CPF and the design. All of these inconsistencies were mistakes in the CPF file and were fixed.

Low-power functional verification proceeded in parallel with CPF coding and QC. Since the analog behavioral models were power aware (see section 4.2), they handled their own power shutoff and did not need corruption to be performed by the simulator. Therefore, functional verification was completed before structural verification because it only needed CPF description for the synthesized control logic.

5.2 Results

After low-power functional verification was completed, the team was moving forward with tapeout. However, we continued to work on low-power structural verification. Our persistence paid off when we found several low-power errors as shown in Figure 14.

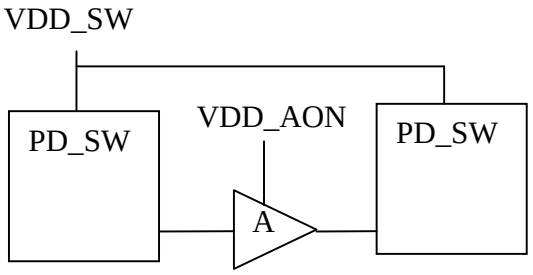

Figure 14. Low-Power Error

Buffer A was incorrectly connected to VDD_AON. Its driver and receiver are both switchable and connected to VDD_SW. When VDD_SW is shut off, there is an off-to-on domain crossing at the input of buffer A. Low-power functional verification was not able to detect this error because the X-state propagated through buffer A was not observable since the receiving block is also shut off. Low-power structural verification didn't have any trouble identifying this problem.

Section 3.2 discussed the overlap between low-power structural and functional verification. It also pointed out some errors that cannot be identified by low-power structural verification. Figure 14 illustrates an error that cannot be identified by low-power functional verification. Therefore, both structural and functional verification must be run to ensure a fully functional low-power design.

5.3 Future Work

Low-power structural verification has proven itself repeatedly on digital SoC designs. In fact, it is part of the sign-off flow for most design teams. Now, it has also proven itself on analog/mixed-signal designs. While this case study validates the concept and proves the value of low-power structural verification, extraction of the power intent from transistor schematics and creation of CPF macro models are non-trivial tasks. The time and effort required screamed loudly for automation. This will be especially true as the design evolves. One would not want to do manual power intent extraction repeatedly as the design changes.

Work is already underway to automate these tasks in order to make low-power structural verification a key part of the analog/mixed-signal verification flow. With some minimal specification of power domains for PG pin pairs and related PG pin attributes, a CPF file along with CPF macro models will be generated automatically from

the analog schematics. When the tool is ready, the full benefit of low-power structural verification can be realized for every analog/mixed-signal design.

6. CONCLUSION

Advanced low-power design techniques, such as MSV, PSO, and DVFS, achieve significant power savings but also introduce significant design, verification, and implementation complexities. This is especially true for low-power design verification. Much progress has been made in recent years to develop low-power methodology and solution for digital SoC designs and low-power IP. Some of these concepts can be applied to analog/mixed-signal designs.

In this case study, low-power structural verification based on power intent file was successfully applied to a mixed-signal SoC. At the top level, the mixed-signal blocks were treated as IP blocks and verified with macro models. In the mixed-signal block, the analog circuits were treated as IP blocks and represented as macro models. This technique uncovered several design errors that had escaped low-power functional verification. It clearly illustrates the need for both structural and functional verification in low-power digital and mixed-signal designs.

7. ACKNOWLEDGMENTS

I have been fortunate to be involved in low-power design and methodology at Cadence from the very early days. There are many serious challenges facing the low-power designer. However, it has been extremely gratifying to see the rapid adoption of our low-power solution industry-wide and to have helped many customers successfully implement their low-power designs. Therefore, my sincerest thanks go to everyone, inside and outside of Cadence, that has been involved with the Power Forward Initiative for making all of this possible.

7. REFERENCES

[1] Common Power Format Language Reference. Version 1.1. August, 2010.

[2] Neyaz Khan and William Winkeler,. 2008. Power Assertions and Coverage for improving quality of Low Power Verification and closure of Power Intent. In DVCON (San Jose, Calif., Feb 19 - 21, 2008).

[3] Neyaz Khan, Mixed Signal Verification of Dynamic Adaptiove Power Management in Low Power SoC. In DVCON (San Jose, Calif., Feb. 22 – 25, 2010).

[4] A Practical Guide to Low-Power Design - http://www.powerforward.org