Low Power Verification with UPF: Principle and Practice

Jianfeng Liu, Mi-Sook Hong, Bong Hyun Lee
JungYun Choi, HyoSig Won, Kyu-Myung Choi
Samsung Electronics, Co. Ltd
San #24, Nongseo-Dong, Giheung-Gu
Yongin-City, Gyeonggi-Do, Korea 446-711
Email: jf.liu@samsung.com

Harsha Vardhan, Aditya Kher
Synopsys Inc.
RMZ Infinity, Tower A, Municipal No. 3,
Old Madras Road, Banniganahali, Bangalore, India
Telephone: 91-80-40188336
Email: {Harsha.Vardhan, Aditya.Kher}@synopsys.com

ABSTRACT

With the widespread adoption of advanced low power design and implementation techniques in SoC designs, the role of low power verification has been more critical than ever. Advanced low power design techniques, such as power gating, state retention, multi-VDD etc, require significant revisions of the verification methodologies, library infrastructure, advanced CAD tool support and serious engineering efforts to tackle the huge complexity in both implementation and verification.

In this paper, the principle of low power verification is explained in detail. Based on the low power intent in the Unified Power Format (UPF), this paper emphasizes the interaction of the low power attribute in design and implementation codes, and EDA tools in low power verification. The paper highlights an extensive checklist for conducting successful low power verification with UPF, including checks for library, UPF power intent, low power static verification and dynamic verification.

This paper will share our experience of low power verification with the Synopsys® MV-Tools for a complex low power design, with about 30 power domains, complex power-state machines and retention schemes. Besides the common problems addressed in the multi-voltage low power verification flows, some special problems in verifying the testcase design will also be addressed, such as the legacy RTL codes where protection cells has been inserted, multi-rail cells and the complex power states in the design. The paper emphasizes the low power verification challenges faced for complex low power designs, and illustrates how the MV-Tools help to solve these challenges.

We conclude with how the UPF based voltage aware static and dynamic verification methodology for low power designs can help save a lot of verification effort, silicon debug time and ensure first pass silicon success.

Categories and Subject Descriptors

B.7.2 **[Design Aids]:** Verification

General Terms

Experimentation, Verification

Keywords

Low Power, Verification, UPF.

1. INTRODUCTION

Power consumption is one of the most important design metrics in current Silicon-On-Chip (SoC) designs. The importance of the low power design not only comes from the objective of extending the battery life of mobile devices, but also is motivated by package cost, electricity bills, circuit reliability and environment issues.

To meet the budget of low power metric in SoC design, it is common that one SoC design employs a couple of complex low power design techniques, from traditional clock gating to advanced power gating and multi-VDD design techniques, from the device level up to architecture and system level [1]. The application of these complex low power techniques not only increase implementation complexity, but also creates significant challenges for verification [2].

In this paper, the principle and practice of low power verification with UPF is documented. Section 2 describes the verification challenges from advanced low power design techniques. Section 3 explains the principles of low power verification with UPF. Section 4 shares our experience of conducting low power verification with UPF using Synopsys MV-Tools. Section 5 concludes the paper.

2. THE VERIFICATION PERSPECTIVE OF LOW POWER DESIGN

Low power design techniques can be divided into two categories from the verification perspective, based on whether the technique involves any voltage control. Traditional low power design techniques such as clock gating, multi-threshold logic and other logic or cell optimization techniques have no influence on the power network of the design, while advanced low power design techniques, such as multi-VDD, power gating, voltage scaling, VDD-standby etc, highly influence the power network of design. Although the influence of the traditional low power design techniques on verification is trivial, advanced low power design techniques which involve voltage control, introduce significant verification challenges.

First, the power gating design dramatically increases the verification complexity. The verification complexity is significantly increased by the addition of new problems of verifying isolation strategy, power gating control strategy, state retention implementation, AON signal buffers, etc. Moreover, the problem of verifying the power gating control unit is huge when the number of power gating domains is big, due to the exponential increase of the possible power states and state transitions.

Second, multi-VDD and voltage scaling designs require signal resolution at voltage domain interfaces to take the supply voltage values of related driven supply nets into account.

Third, verifying voltage control related low power design techniques demands the power intent to be in a separate specification, separated from the golden design intent.

Finally, the seamless low power verification flow requires significant updates of the whole verification infrastructure, such as adding the related low power attributes in the library, enhancing CAD tools aware of the influence of voltage control on logic simulation, and making seamless integration flow with the original verification environment.

3. THE PRINCIPLE OF LOW POWER VERIFICATION WITH UPF

Low power verification with UPF, i.e. the process of verifying whether the low power intent is defined and implemented correctly, involves several steps. UPF flow compatibility check of library is the first step in the verification flow. After finishing UPF file, the first step to start is UPF quality check, to make sure that the UPF power intent is defined correctly and consistently, with respect to the golden design RTL codes or netlist. After the UPF power intent is qualified, static verification is the best choice to make sure that the design is implemented in a way matching the UPF power intent, and there is no architecture error. Finally, dynamic verification must be performed to check all the bugs which are not covered by the previous steps, such as errors related with power control sequences, power states, state transitions etc.

3.1 Library Check

Low power verification flow with UPF needs strong support from library attributes related to power/ground supply, and the usage attribute of the low power control related pins. Liberty standard 2007.12 [3] defines a complete set of library attributes, which are enough for low power verification flow. However, in practice, the library providers only add parts of the liberty attributes defined in the liberty standards. Therefore, it is important to check whether necessary library attributes have been added to the library infrastructure before starting the low power verification, or else, it might take a lot of time to debug the design but finally find that the source of a problem comes from a missing or wrong attribute in library.

The first mandatory attribute in the library is the pg_pin attribute. Table 1 lists the mandatory attributes [5].

Table 1 PG Pin Requirements in Liberty Files

Library Level Attribute	Cell Level Attribute	Pin Level Attribute
voltage_map	pg_pin	pg_type voltage_name related_power_pin input_signal_level_low input_signal_level_high output_signal_level_low output_signal_level_high always_on

In addition to the PG pin information, other library attributes are required for multi-voltage cells, such as isolation cells, level shifters, retention cells, switch cells and always-on cells, to identify the type of the cell, the usage of the pin and their multi-voltage behavior. These mandatory attributes are defined in Table 2.

Some important notes for library check are:

1. Although strictly speaking, power_down_function attributes are not needed for all the multi-voltage cells in the static verification and dynamic verification flow, it is recommended to add this attribute for all the output pins of the multi-voltage cell libraries for the low power equivalence check flow.
2. The “std_cell_main_rail” attribute is needed for one of the primary_power type PG pins of the level shifter cells.
3. The “direction” attribute is mandatory for the pg-pin of power switch cells. The value of the “direction” attribute is output for the virtual supply net, and input for the reset of the PG pins
4. The liberty attributes of related_power_pin and related_ground_pin will be used for port/pin based partition in UPF, which is required for verifying multi-rail cells and hard-macros where each pin could operate on a different voltage.

Table 2 Liberty Attributes to Model Multi-Voltage Cells

Cell Type	Cell Level Attributes	Pin Level Attribute
Isolation	is_isolation_cell	isolation_cell_enable_pin isolation_cell_data_pin
Level Shifter	is_level_shifter level_shifter_type input_voltage_range output_voltage_range	level_shifter_enable_pin level_shifter_data_pin input_voltage_range output_voltage_range input_signal_level
Retention	retention_cell	retention_pin power_gating_pin nextstate_type
Switch	switch_cell_type dc_current related_switch_pin related_pg_pin related_internal_pg_pin	switch_function pg_function switch_pin always_on power_down_function
Always-on	always_on	always_on

3.2 UPF Qualification

As an executable power specification, the UPF power intent file needs strict qualification process before usage, to avoid a time consuming revision in the later implementation and verification flow.

Besides common UPF command syntax checks [4], the mandatory checks to qualify a UPF file, in our opinion, are:

1. All the states listed in the add_port_state command must be used at least once in power state table.
2. Isolation policy must be the mandatory and sufficient condition of the power state table.
3. In most cases, level shifter policy can be automatically inferred by the CAD tools from the power state table, therefore, level shifter policy can be omitted. However, if level shifter policy is defined, it must conform to the power state table definition.
4. The UPF object name should not overlap with the design object name in the same scope
5. All reference to design objects, i.e. instance names, module name, signal hierarchy and names, must conform to the design database.
6. All references to library objects, such as PG pins, multi-voltage cell name etc., must conform to the object names and object attributes in the library.

Besides the above general checks, there are some checks specific to certain design styles. For example, instead of one chain of power switches, two chains of power switch cells are used in mother-daughter type power switch design flow. In this case, the resolution type of the virtual supply definition must be defined as the parallel type.

Although one UPF file could be perfectly correct according to the syntax and design database related check, the lack of certain syntax support in some EDA tools makes the UPF unusable. Because the UPF file is supposed to be used in both in the implementation and verification flow, it is important to make a UPF command and option support table, which contains the commands and options which are supported by all the EDA tools needed in the design flow of a projects. The UPF file qualification process must take this check into account.

3.3 Low Power Static Verification

The aim of the low power static verification is to check for architecture errors related to low power design, and violations with respect to the library attribute usage and the UPF specification.

One of major requirements in a power gating design is that a spatial crossing must be in an electrically safe state at all time [2]. Corrupted signals from a power down domain must be protected by isolation cells if the corrupted signals drive some active logic at some power states. Table 3 shows the verification check list for isolation cells.

Table 3 Isolation Cell Check List

No	Description
ISO_1	Missing isolation cell
ISO_2	Redundant isolation cell
ISO_3	Normal isolation cell locate in OFF block
ISO_4	Always-on isolation cells are located in the ON block
ISO_5	Isolation cell data input pin is driven by an always-on constant.
ISO_6	Isolation cell output is floating.
ISO_7	Isolation control signal has wrong polarity
ISO_8	Isolation control signal is tied to a constant
ISO_9	Isolation control signal corrupted when the destination domain is ON
ISO_10	Isolation cell type mismatch with UPF specification
ISO_11	Isolation cell name mismatch with UPF specification

Except ISO_4, the checks shown in Table 3 are, we believe, severe errors which require mandatory revision of the design. ISO_4 is not mandatory to fix, but recommended because of implementation cost of always-on isolation cells.

To protect the spatial crossing in multi-VDD design, level shifters are needed. The check points for level shifter cells are:

1. Missing level shifter cell from low voltage to high voltage
2. Missing level shifter cell from high voltage to low voltage
3. Incorrect level shifter cell type
4. Input Voltage is different between the UPF specification and library attribute
5. Output Voltage is different between the UPF specification and library attribute
6. Level shifter location mismatch between the design and the library attribute.

Note that the high to low type level shifters are needed mainly for accurate timing analysis purpose, therefore, this check is recommended to be an optional check.

When both isolation and level shifter cells are specified to protect the same wire and their location specification is the same power domain, an enable level shifter should be used instead of one isolation cell and one level shifters. It is recommended that this cell combination should be checked, which should only happen in the above scenario and only in this scenario. Naturally, enable level shifter inherit most checks from both isolation cells and level shifters, which will not be listed here to avoid overlap.

To enable fast state restoration after power up, retention cells are used for some or all registers in power gating domains. The checks for retention cell are listed as the following:

1. Redundant retention cells other than the retention rule defined in UPF specifications
2. Mismatch between retention cell name used in the design and the UPF specification
3. Retention cell control (save/restore) signal should come from always-on logic
4. Retention cell control signal (save/restore) is tied to a constant

Always-on check is one of the most critical checks for power gating design. A bug related with always on can make serious troubles for the chip operation. Although doubles of specific scenarios can be made for always on checks, two general rules can be summarized as follows:

1. Signals with always on requirement should not be corrupted unless the destination logic is powered off.
2. Always on cells should be used only when necessary, and its input(s) should not be corrupted when the always on cell itself is still power on.

In the above principles, signals with always on requirement can be divided into two categories: one is the cell pin defined with always on attribute in library; the other category includes the signals which should be treated as always on in the design, such as clock, reset signal etc. Note that, by default, all the pins of an always-on cell have always-on attribute.

Power switches can be grouped to cut off supply nets to a power domain. Checks on the power switches are summarized as follows:

1. Incorrect power switch type (header/footer)
2. Power switch name mismatch with UPF specification
3. Power switch control signal should come from always on path
4. Power switch control signal should use always on buffers when a buffer is needed between power switch cells
5. Power switch control signal is connected incorrectly with respect to UPF specification

Beside the above logic rule checks, physical rule check is mandatory to ensure that each cell is connected to the correct power and ground supply net. The multi-VDD and power gating techniques complicate this issue significantly by increasing the number of the possible power supply net and making the difference of real and virtual supply nets. Special attention must be paid to the PG connectivity of the following cells:

1. Standard cells in a power gating domain: to check if real supply net and virtual supply net are connected correctly
2. Multi-Voltage cells, especially always on isolation cells, always on buffer/inverter, level shifters, retention cells etc.
3. The PG connection of power switch cells should respect the UPF specification and library pin “direction” attribute
4. Multi-rail cells and Hard-Macros.

The checkpoints listed above are targeted to cover as much as possible the bug scenarios that appear in the general power gating and multi-VDD designs. However, real designs and specific design flows could introduce additional bugs which are not covered here.

3.4 Low Power Dynamic Verification

Although static verification techniques are powerful and exhaustive, the bugs which could be found by low power static verification are limited in scope, because power management is essentially a dynamic process.

Simply speaking, the goal of dynamic verification is to find any bugs uncovered by the low power static verification step. To achieve a successful low power dynamic verification, both a voltage aware simulator and a detailed low power verification plan are mandatory.

A voltage aware simulator resolves the logic value with the consideration of voltage influence. Essentially, a voltage aware simulator provides the following benefits:

1. When a block is powered down, all the outputs of the combinational elements and sequential elements in this powered down block should be forced to be ‘X’ value before power up. The X-injection mechanism enables us to catch bugs in the un-protected power domain crossovers. In the waveform, X value will appear in the un-protected crossover. In simulation, abnormal behavior of the logics driven by the un-protected crossover will be observed soon after the driving power domain is power down.
2. The simulator implements retention functions according to the control signals specified in the power intent specification.
3. Logic resolution should take the accompanied voltage value into consideration for a multi-VDD design

To uncover a bug in simulation, verification planning, i.e., defining the possible scenario and test vectors to exercise the bug, is the most important step. The first part in the verification plan is to check whether the power aware simulation behavior works as specified by UPF and power control signal values. For example:

1. The values of the logic in a power gating domain must be checked whether they are corrupted as expected when the power domain sleep enable signal is asserted.
2. The process of power up of a power gating domain must be checked carefully after the sleep enable signal is de-asserted, i.e. whether the power down starts at a proper power state after power up. Most power gating domain uses Power-On-Reset (POR) techniques, therefore, registers should be reset immediately after the sleep enable signal is de-asserted.
3. The isolation output clamping value must be checked against UPF specification for each isolation policy, especially when more than one isolation policies are defined for a power domain, and there are some power domain interfaces exempt from the isolation policy.
4. The retention behavior could be very complex, and vary significantly with cell types and library vendors. The save and restore behavior of some retention libraries depends on clock and reset signals. A detailed study of the retention waveform must be conducted to check if the retention model works as expected with respect to the waveform provided by the library vendor.
5. All the always-on signals, multi-voltage cell control signals and wakeup signals must be checked in detail to avoid any unexpected corruptions. Although the always-on checks can be done by the low power static verification step, the frequent design update and ECO could easily introduce this kind of errors, given that low power static verification is not complete or not reported due to mistakes by static verification engineers.
6. Memory and Hard-Macro power gating behavior must be paid special attention when they are put in a power gating domain. Hard-macro simulation models could contain some behavior codes, such as initial statement, which could have unexpected behavior in power aware simulation.
7. The behavior of power gating components which are in the form of the encrypted codes should be checked. The code encryption could incur problems for X-injection process in power aware simulation, depending on in what form of encryption the codes are encrypted.
8. The behavior of power gating components in VDHL language must be checked, because the power aware simulator may have limited mix-language support for power aware simulation.

The second part of the verification plan should cover the possible low power control sequence bugs as follows:

1. The polarity of control signals after system boot-up: In most SoC designs, all the power gating blocks are turned on after system boot-up is finished, i.e. the SoC starts with the all-on state. However, there are indeed some SoC designs require some or all of the power gating modules to be at OFF state after the system is boot-up. The related sleep enable signals, isolation enable signals and other power gating control related signals must be checked just after the system boot-up process is finished to make sure that the system boot-up correctly.
2. The sequence order and polarity of the sleep enable signal(s) and isolation control signal(s) for each power gating domain must be checked. Simply speaking, isolation enable signals have to be asserted earlier than sleep enable signal is asserted, and has to be de-asserted later than sleep enable signal is de-asserted. In real SoC designs, multiple sleep enable signals (eg. mother-daughter type power switch control signals) and multiple isolation enable signals (eg. for high clamp value and low clamp value) will make the scenario complicated.
3. The sequence order and polarity of retention related control signals must be checked. The sequence relationship between save/restore signal, the isolation control signal and sleep enable signal should be checked first. Figure 1 shows an example waveform of dual control signal retention cell [2]. Beside this, the condition of save and restore edge must be checked with respect to the guidelines in the library specification. Note that although reset signal is not shown in Figure 1, reset signal should be kept inactive at the save and restore edge. Furthermore, some specific design flows put further constraints on save and restore condition than that specified in library.

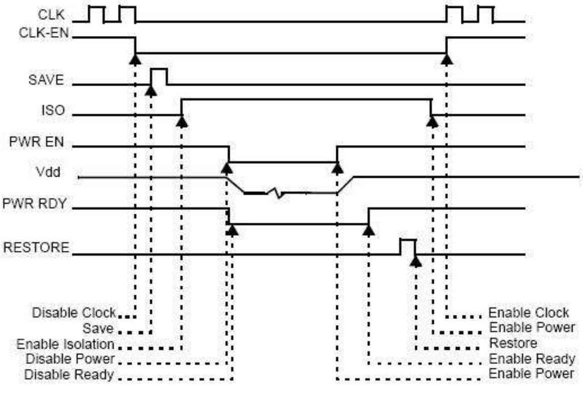


Figure 1. Dual Signal Retention

4. The relationship between power gating control signals and low power state control signals of IP must be checked. Some IPs and bus protocols require that the IP must be in a specific state before going to power down state.
5. The relationship between clock gating and power gating may need to be checked according to design guide. Some design flows require clock gating signal to be asserted during the power down period, to reduce the clock tree power. But this is not mandatory because there is no functional error when the clock is not gated for power down scenario.

6. The relationship between clock gating control and standby control signal must be checked.

Besides the above sequence checks for each power gating domain, sequence checks must be done for control signals between different power gating domains, with respect to specific SoC design flow document.

The final part of the verification plan should cover the power state and state transitions. First, the design must be verified to work as expected in all the power states and for all the possible legal transitions between power states. Second, all the registers related to power control must be verified to be in a proper state with respect to the power state and power state transitions. Therefore, the verification plan should not only cover the start and end states of state transitions, but also the internal state requirements (eg. power control FSM) for state transitions.

Given the regularity of the power control sequence, assertion based verification should be used in conjunction with planning based verification methodology, to provide a predictable and measurable verification flow. Essentially, all the power control sequence and power state transition related checks can be covered by assertions. Beside that, all the always-on related checks and power up states check can be done by assertions.

4. THE PRACTICE OF LOW POWER VERIFICATION WITH UPF

4.1 Test Case Introduction

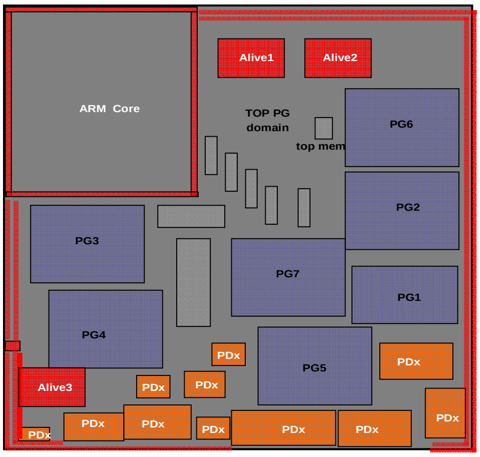


Figure 2 Power domains in the design

The design testcase has very stringent power requirement under different operation modes to extend the battery operation life. For the design used in the evaluation, the design size is about 40 million gates, with several 10s of IPs. It employs multi-VDD, power gating [MTCMOS headers and footers], complex state retention and Dynamic Voltage and Frequency Scaling (DVFS) based power management techniques to meet aggressive power budgets. As shown in Figure 2, there are 30 different power domains. Among these 30 power domains, 8 are power gating domains controlled by software registers, 1 hard-macro IP has power gating scheme in the IP, and several memory components are power gating memories.

One notable low power architecture of the evaluated design is that not only some IPs are power gated, but also the top design module is power gated. Most power gating domains for IPs are controlled independently by software registers. When the top power domain is power gated, all the other power gating domains are power gated except one special power domain. Some power domains including top power domain have state retention capability that can be controlled by software registers.

Significant verification challenges are posed by such a complex low power design

1. Power intent in UPF: capturing the power intent for a design with 30 power domains is a very challenging task. With UPF specification running to over 2000 lines, the first verification challenge is to check for consistency and correctness of the specification itself.
2. Legacy design codes: The design uses legacy code for power gating blocks, in which isolation cells have been inserted. This puts a significant deviation from the standard UPF Flow.
3. Multi-rail cells: A multi-rail cell has more than one power supply. As these cells can not be partitioned to a specific power domain, special care must be taken for these cells.
4. Power state space: The number of possible power state numbers is huge for a 30 power domain design. In our design, the number of legal power states numbers more than 4000. Such a large number of power states make the power state coverage a big challenge.
5. Custom retention: The design use both clock dependent and clock independent type retention cells in the design. Furthermore, the design flow requires complex restore behavioral models, which depends on relationships among restore control, clock and reset signals.

4.2 Low Power Verification Flow with MV-Tools

Low power verification should be performed in three stages, that is, RTL stage, pre-layout netlist stage and Post-layout netlist stage. As shown in Figure 3, at each stage, MV-Tools create a multi-voltage database (MVDB) with the inputs of design database (eg. Verilog codes), UPF file and library, then the MVDB can be used by MVRC for low power static verification and MVSIM for low power dynamic verification [5]. The MVDB generation process comprises of two steps: multi-voltage compilation with MVCMP, and multi-voltage elaboration with MVDBGEN.

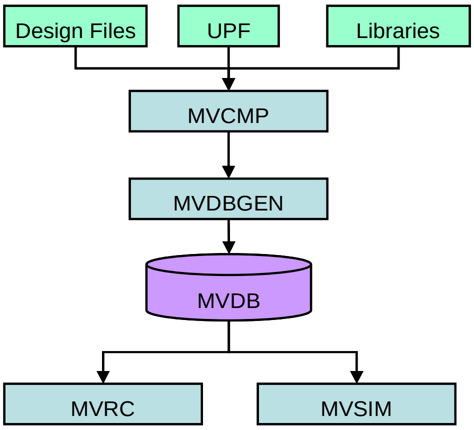


Figure 3 MVDB generation

Although simple at both pre-layout and post-layout netlist stages, preparing design file at RTL stage for MVDB generation flow requires some efforts for a large SoC design, due to variability of IP providers:

1. Separate Verilog codes from VHDL code
2. Bypass compilation file list: some hard-macro models having behavior codes may make multi-voltage compilation difficult.

Some encrypted modules (depending on the way of compilation) may also need to bypass the multi-voltage compilation step.

To ease the effort of the preparing the bypass compilation file list, MV-Tools provides two environment variables. The file list can be specified with wildcard support in one environment variable, or directory names can be specified where all the files in the directory will bypass the multi-voltage compilation step.

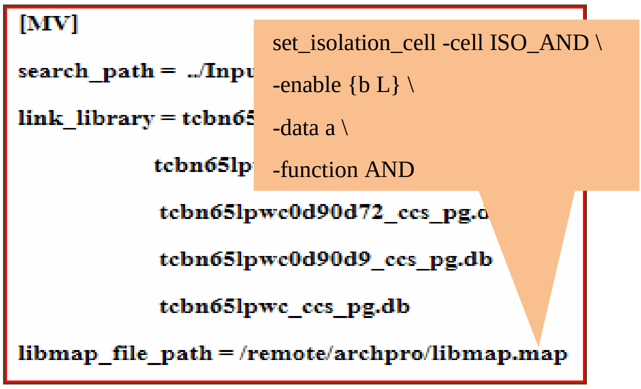
4.3 Low Power Static Verification

MVRC® was used for static rule check on design and UPF. Specifically, tool version 2009.06 was used [5].

4.3.1 MVRC Flow Setup

MVRC can be seamlessly integrated in all the three stages in the standard UPF flow, including RTL stage, pre-layout gate netlist, and post-layout netlist. MVRC provides a set of TCL commands to perform different kind of static checks for each stage in the UPF flow. A Makefile template is provided with the tool release for easy customization and setup. The MVRC tool setup flow is easy and flexible according to our usage experience.

Generally, MVRC recognizes special cells in the design based on the cell and pin level attributes of their liberty .db descriptions. If liberty does not have correct attributes then a cell mapping file can be used to override or append attributes and description to .db information, which is called LIBMAP flow. As shown in Figure 4, the initialization file (archpro.ini) was used to direct MV tools, about the correct location of such cell mapping file.



```
[MV]
search_path = ../Inpu
link_library = tcbn65
               tcbn65lp
                tcbn65lpwc0d90d72_ccs_pg.d
                tcbn65lpwc0d90d9_ccs_pg.db
                tcbn65lpwc_ccs_pg.db
libmap_file_path = /remote/archpro/libmap.map
```

```
set_isolation_cell -cell ISO_AND \
-enable {b L} \
-data a \
-function AND
```

Figure 4 MVRC LIBMAP flow

The LIBMAP flow is very helpful when a library does not have all the necessary attributes listed in Section 3.1, and significant delay is expected to add the missing attributes due to the strict verification procedure in library infrastructure team.

4.3.2 MVRC Check Summary

MVRC offers a rich variety of checks that allowed comprehensive verification of both UPF quality and the design database. The checks could be divided in two main categories – *critique* flow and *intent* flow. In *critique* flow, the power intent infrastructure (power domain definitions, isolation & level shifter strategies etc.) is checked against power state table as a reference. These checks therefore could validate UPF power intent alone. The *intent* flow on the other hand, verifies the design structure against complete UPF.

Structural checks verify protection logic against the Power State Table in UPF. These checks cover scenarios such as missing or redundant cells, incorrect type of cell, incorrect location (i.e. power domain), incorrect isolation polarity and isolation-enable rails and incorrect type of level shifting used(by verifying the “standard main rail” attribute). For retention registers, MVRC verified the reachability of the retention control specified in the UPF against the actual signal in the design.

Architecture checks verify that isolation control and power switch control signals are generated from the correct domain. They also perform island-ordering checks on important signals such as Clock, Scan, Reset and other rails that are required to remain always ON.

Once a PG netlist and the UPF file generated by P&R tools, such as ICC, become available, MVRC was used to perform power-ground connectivity checks based on power partitions described in UPF. It could further check PG connectivity of level shifters, isolation cells and power switches and further validate UPF completely against the netlist.

4.3.3 User Experience

The overall user experience for MVRC was very positive. The Quality of Results increased confidence in the design and power intent. The compile and run time performance was also satisfactory.

MVRC shows excellent quality to qualify a UPF file. MVRC could find almost all the critical bugs listed in Section 3.2. Some bugs found by MVRC in the UPF qualification for the testcase design are:

- Missing isolation policy
- Redundant isolation policy
- Power port state not defined, but used in power state table
- Missing level shifter policy
- Signal/instance name defined in UPF, but does not exist in the design database

One notable point is that MVRC does not strictly check the UPF support table of the Synopsys Eclypse platform. When a UPF file contains some command options which are not supported by Design Compiler (DC), MVRC does not create a warning message.

MVRC discovered many significant bugs in the testcase design, and shows the quality of sign-off tools. Some of bugs found by MVRC are listed here:

- Not Reachable of retention controls: the control signals for some retention registers in one particular power domain found to be un-connected in the netlist.
- For some instances of dual rail isolation cells and always ON cells, the back-up and primary power rails were shorted.
- Missing ‘no-isolation’ policies on the output of ‘ON/OFF’ power domain.
- Island Order checks – Normal buffers were found to be inserted (instead of always ON cells) for isolation enable signals and wakeup signals, which was illustrated in Figure 5.

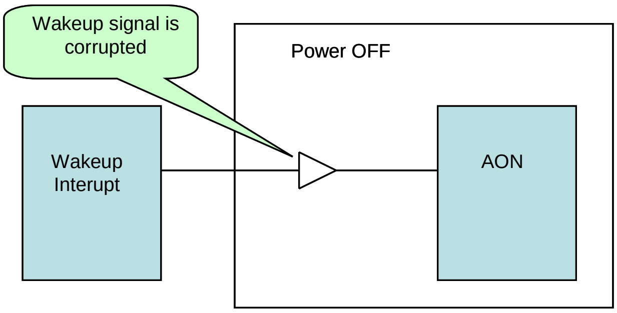


Figure 5. Corruption of Wakeup signal

The results of rule checks are output in the form of plain-text, tabulated reports. Although the reports themselves may be clear enough to understand, traversing between a given error/warning in the text report and the design or UPF is tedious and inconvenient. Furthermore, for a large design with complex power intent, the length of a single text file was too long, making the usability more cumbersome.

Based on our feedback, Synopsys R&D has enhanced the reporting structure in next release (version 2009.12 beta release) of the tool. In particular, GUI-based tabbed browsing of reports and hyper-linking between a given error in the report and the source of error in the netlist alleviate the usability issues to great extent.

4.4 MVSIM

4.4.1 The Legacy Simulation Flow

For verification of dynamic power, the legacy test environments in Samsung used in-house X-injection tools. This tool along with non-standard power management schemes had several limitations:

1. Voltage as binary value: Treating voltage as "real" variable that could ramp-up or ramp-down was lacking in the homegrown tools. This prevented engineers from accurately verifying the switching activity during voltage ramps.
2. Non-scalable methodology: The design was very large. The power architecture was also complex. The legacy methods could not be scaled to handle this combination of scope and complexity.
3. Power-on Reset: Verifying power-on reset was important criteria of the low power verification setup. The legacy method could not test this scenario.
4. Unplanned or Unsafe state transitions: The design went through several power and logic states. Therefore, ability to detect unsafe or unplanned states was another important condition that the legacy tools could not meet.

The limitations described above prompted the adoption of a more sophisticated, standard and scalable low power verification methodology.

4.4.2 MVSIM Flow Setup

The design was verified at block level using power-aware test benches. There were three flavors of such verification environments maintained throughout the verification effort:

1. OpenVera + NC-Verilog (3-step)
2. Specman 'e' + NC-Verilog (3-step)
3. OpenVera + VCS

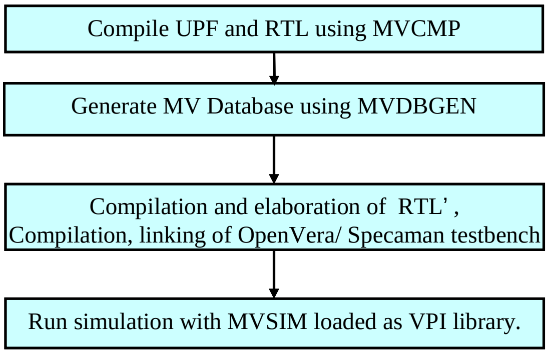

Compile UPF and RTL using MVCMP
Generate MV Database using MVDBGEN
Compilation and elaboration of RTL' ,
Compilation, linking of OpenVera/ Specaman testbench
Run simulation with MVSIM loaded as VPI library.

Figure 6. MVSIM Integration with HVL Based Testbench

The MVSIM integration flow is illustrated in Figure 6. MVSIM allowed a seamless and easy integration with all the existing functional verification environments of the testcase design, regardless of testbench language and the choice of simulators.

According to our experience, the general flow to setup a low power simulation is as follows:

1. Check that the original simulation works correctly: although this looks straightforward, it is very important, especially for an on-going project.
2. Elaborate multi-voltage database MVDB, but run simulations without low power simulation options (called TRANSPARENT mode in MVSIM) on MVDB to check if the low power elaboration process is perfect
3. Run low power simulation in PROTECTED mode or ACCURATE MODE, depending on the design database. If design database is golden RTL codes without multi-voltage cells, low power simulation should be run in the PROTECTED mode. The ACCURATE MODE is mainly used for gate-level simulations, where multi-voltage cells have been inserted.
4. In cases of any abnormal behavior happen, check low power related information in the log file and the power control signals and signal corruptions in the waveform to locate the source of problem. Revise the design files or UPF specification to correct the problems.

4.4.3 Addressing the Challenges in the Test Case Design with MVSIM

The verification challenges of the testcase design (listed in Section 4.1) are addressed by MVSIM as follows:

1. Protection Gates at RTL level: As the testcase design has protection gates that are already placed at RTL stage, the challenge was to accurately verify them with respect to specification in UPF. This is a non-standard UPF flow because some IPs have inserted protection cells and some don't. The designers wanted to run simulations by virtually simulating protection cells only where they are not placed. For this case, the PROTECTED mode of MVSIM simulation helped. In PROTECTED mode, MVSIM would associate UPF protection policies only for the crossovers for which protection gates are NOT present at RTL stage. It respects already existing protection gates.
2. Handling of Multi-rail Macros: For multi rail macros, explicit connect_supply_net commands are written in UPF to route the power network to the macro ports. MVDBGEN recognizes any cell with more than one pg_pin as a multi rail macro. Once MVDBGEN marks a cell as multi rail macro, MVSIM does not corrupt its internals. Multi voltage semantics such as shutdown corruption are applied only to the logical ports that are related to power ports where voltage events take place.
3. Hierarchical Power State Tables (PST): Hierarchical PST and wild cards in supply port states helped to make the legal state specification of such a complex chip compact and readable.
4. Continuous assignment is the power down domain: another particular helpful feature was to mark certain signal paths as always-on at RTL. Such signals, even if passing through the shutdown domain, would not be corrupted by MVSIM, which is particularly useful to model the always on control signals, such as resets, isolation enables, save, and restore signals and clocks when they traverse through different power domains.
5. Custom Retention: Tool was enhanced to attribute different types of retention policies to different sets of registers. Enhancements were made to apply clock dependent and clock independent retention schemes used in the testcase design.

4.4.4 User Experience

The power aware simulation behavior provided by MVSIM can help engineers to easily detect many problems related with the power gating, multi-VDD design [6,7,8,9]. The design issues caught by MVSIM for the power gating design can be categorized into following groups:

1. Power gating control sequence
2. Signal Corruption in Power Gating domains
3. Protection cell related

One of the important design specifications in our power gating design is the power on reset, i.e, sequential elements can be reset when the reset signal is actively asserted at edge of the power supply is given. Therefore, it is very important that the reset is asserted before sleep signal is de-asserted. Figure 7 shows a reset sequence error: reset is asserted after the deassertion of the sleep enable signal, where the register output Q has a corrupted value "x" for the period after the power supply is given back and before the reset signal is asserted.

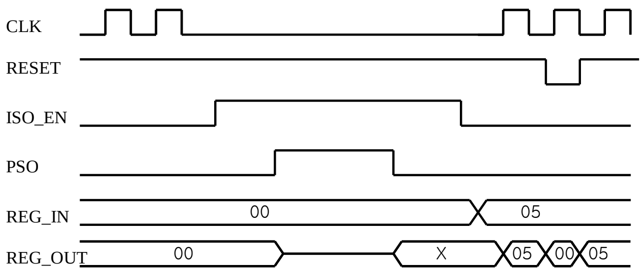


Figure 7. Power On Reset Assertion Sequence

With custom retention of type clock low retention, there is an extra dependency on clock polarity for a successful restoration. Figure 8 illustrates that the restoration operation was not successful as the clock was gated to wrong polarity [1] when the save_restore signal was de-asserted for restoration. The clock should have been gated to '0' at save_restore negative edge for successful operation of the retention flop.

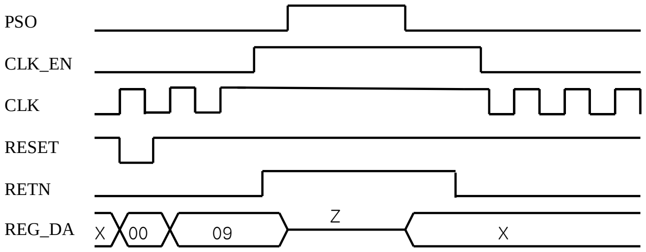


Figure 8. Failed Control Sequence for Custom Retention

Another issue with the power gating design is the signal corruption in the power gating domain. One common error is that some always on signals are corrupted because some normal buffers or multiplexers with switched supply net are used when routing in the power gating domain. These errors are usually serious design bugs, which could cause functional errors at the power down mode. With the power aware simulators like MVSIM, these bugs can be identified easily by the X-injection mechanism.

MVSIM corruption mechanism again helps us to find isolation cells related bugs, such as incorrect isolation enable polarity, incorrect protection cell type etc. Besides catching the isolation enable polarity problem with waveform, MVSIM generates warning message in the log files as follows:

```
[MVSIM] WARNING 5514: Output of Isolation cell
tb.dut.n148 for Isolation policy gprs_iso_in
is '0'. Expected value = '1' at time = 256 ps.
```

MVSIM also has the capability of detecting the case of the missing level shifter, and the input of signal runs out the input_voltage_range of level shifter, which is very helpful for verifying the multi-VDD and voltage scaling design. Figure 9 shows how the level shifter output was corrupted when the input is the out of range specified in the liberty attribute input_voltage_range of the level shifter.

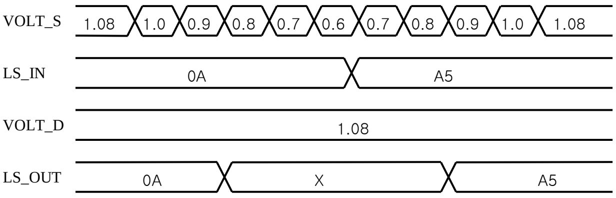


Figure 9. Level Shifter Corruption when Voltage Change

The desired features missing in MVSIM are verification planning and sequence assertion automation. Verification planning of a design with large PST is a complex task to which we could find no satisfactory solution. An automated way of capturing the planning intent (that adequately describes high level power state transitions), and verifying that the entire intended verification plan was indeed covered will be a great value addition to the tool and low power verification methodology. Further, automated generation of power control sequence assertion will greatly improve the verification confidence and debugging capability.

5. CONCLUSIONS

In this paper, both the principle and the practice of the low power verification with UPF have been studied in detail.

The paper highlights an extensive checklist for successful low power verification with UPF, including checks for the library attribute, UPF power intent, low power static verification and dynamic verification. The paper shares our experiences of low power UPF verification using MV-Tools. The procedure to setup the verification flow and how the MV-Tools help uncover low power bugs are explained.

In conclusion, UPF based low power static and dynamic verification methodology helps save verification efforts, reduce silicon debug time and ensure first pass silicon success.

6. REFERENCES

[1] M. Keating, D. Flynn, R. Aitken, A. Gibbons, K. Shi, Low Power Methodology Manual – For System-On-Chip Design, Springer, July 2007

[2] S. Jadcherla, J. Bergeron, Y. Inoue, D. Flynn, Verification Methodology Manual For Low Power, Synopsys, Feb. 2009.

[3] Liberty User Guide and Reference Manual, Version 2007.12, http://www.opensourceliberty.org/

[4] Unified Power Format 1.0, Feb 2007, http://www.unifiedpowerformat.com

[5] Synopsys Low Power Verification Tools Suite User Guide, Version 2009.06, Jun 2009

[6] J. Liu, H. Vardhan, A. Kher, J.Y Choi, et al., Multi-Voltage Verification for High End Mobile Processor Design, SNUG India, 2009

[7] G. Kumar S, et al, Verifying Complex Low Power Integrated Graphics Chip: A Methodology Using MVSIM and MVRC, SNUG India, 2009

[8] S. Ram, P. A. Krishna, P. Tiwari, Challenges of Multi-Voltage Verification on a complex low-power design, SNUG San Jose, 2008

[9] P. Bhairi, et al, "Addressing the challenges in full chip power aware functional verification with MVSIM, SNUG India, 2008