Power Management Verification for SOC ICs

David Akselrod, Alex Miretsky, Sergey Golubkov, Pearl Liu, Afshin Parvaresh, Shu Wang, Feng Yan

Advanced Micro Devices

Advanced Micro Devices Inc., Markham, ON, Canada

Abstract **-This paper presents an overview of power-aware power management verification methodology employed in verification of AMD Accelerated Processor Units (APUs) and Graphics Processing Units (GPUs). The employed methodology presents an approach balancing the demand for optimal test coverage of power-management features of SoC Integrated Circuits (ICs) of growing complexity within given window of project duration. The paper analyzes challenges introduced by physical design flow and provides solutions and mitigation techniques.**

I. Introduction

Verification in general is one of the biggest challenges in design of complex multi-core SoC ICs. Rapid increase in the complexity of modern SoC systems drives the complexity of power-management logic implementing a wide spectrum of functionality geared towards lowering power consumption - power-gating (PG), Dynamic Voltage and Frequency Scaling (DVFS), retention elements and many others. Since some elements of IC related to power-management (such as re-timing buffers introduced by physical design flow) are often not present at RTL level, verification of complex power-management schemes presents one of the major challenges in modern SoC verification [1], [2] especially while design-cycles are being significantly shortened.

One of the difficulties of power-aware verification is in the lack of all the required components at the early stages of the project. For instance, feed-through buffer cells and repeater flip-flops introduced during physical design (PD) stage are instrumental for power-aware simulation but are available only at middle stages of the design cycle. Power-aware netlist used for gate-level simulations is available at rather late stages. The implemented solution is dividing power-aware verification into a number of hierarchical stages. Such approach will provide coverage sufficient to discover most of the power-related issues at quite early stages of the project.

Power management verification techniques received extensive coverage [1], [3], [4], [5], to name a few. An impact of modern power management techniques on verification was described in [1]. Verification of clock gating, voltage scaling, and power gating techniques and the challenges in power-related features design and verification were described in [4]. IEEE1801 standard (UPF) has been an integral component of specifying power-related features of SoC ICs [3], [5].

A “hybrid” tile-level gatesim approach, in which the GLS is split to simulate each individual physical “tile”, to which the physical layout of the chip is divided, at netlist level is described in [6].

II. Power management techniques

The following categories, categorized in Table 1, include most of the techniques employed for power management.

1. Frequency scaling

   Frequency scaling is employed for a number of reasons:

   a) Reducing power consumption of the entire IC or its parts via lowering the frequency of a selected number of the clock sources. Since the dynamic power consumption of CMOS circuit is proportional to its clocks frequencies, reducing individual functional clocks frequencies will lead to dynamic power saving.
   b) Adjusting clock frequencies of the IC to the characteristics of a given die, external power sources levels (e.g. voltage droop), power supply current levels, external temperature or other parameters. Since IC timing behavior depends on its power supply levels, external temperature, specific part's characteristics or a number of additional dependencies, adjusting its clock frequencies will ensure its correct functionality.

2. Clock gating

Clock gating is a specific case of the technique above, in which clock sources frequency is lowered to zero, effectively stopping (gating) the functional clocks. The power consumption will therefore be limited to leakage.

3. Voltage scaling

In this technique, the power supplies voltage levels are adjusted to the current clock frequencies, to the demand for processing speed and performance at specific time, and to the characteristics of the given die. Voltage scaling is usually done in conjunction with frequency scaling.

a. Voltage and Frequency scaling

VFS (voltage and frequency scaling), is a combination of the Frequency scaling and Voltage scaling techniques. In VFS, frequency and voltage pairs are either pre-determined, determined per each specific die during one-time calibration process or dynamically adjusted during chip operation. Dynamic Voltage and Frequency Scaling (DVFS) and Adaptive Voltage and Frequency Scaling (AVFS) are common techniques used [7]. For example, AVFS [8], [7] dynamically adjusts voltage levels and clock frequencies during chip's operation, based on a number of input parameters. AVFS transitions and operating parameters are influenced by on- and off-die sensors such as ring oscillators, or activity monitors.

4. Power states transitions

Power states transitions, depicted in Fig. *1*, are transitions between the set of power states $\{M_i,F_i,V_i\}$ i=1…N where N is a total number of different power states. $M_{i,}$, F_i and V_i are operational mode, frequency and voltage associated with the power state I, respectively. Additional conditions may differentiate between power states. Each power state may have a number of sub-states with their own characteristics and transition conditions. There is a certain overlap between the concepts of VFS and power states transitions. Power states may be more explicitly tied to different operational modes. For example, a microprocessor power states transitions could be implemented via VFS but each power state will be tied to the pre-defined operational mode of the processor. The value F_i could be zero meaning that in power state *i* the clock is gated. Similarly, value V_i could be zero meaning that in power state *i* the voltage is power-gated. An elaborate set of conditions controlled based on activity of underlying logic define various power states transitions. Power states transitions could be influenced by sensors (on-die temperature sensors, ring oscillators, or activity monitors) as well as by activity levels and current performance requirements of the underlying logic.

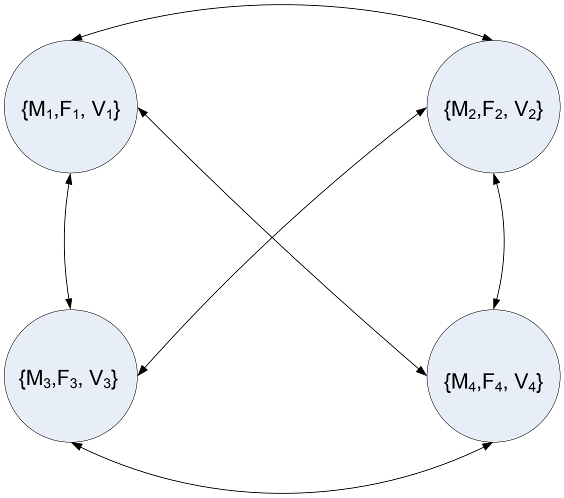

Fig. 1 Power States transitions

5. Localized power gating

The localized power gating will switch off the power supply of an area in IC. Usually, this is accomplished via local on-die power supply switch. The advantage of this approach is ability to cut off the power to an arbitrary part of the design. One of disadvantages is the area of the power switch.

6. Voltage Rail shutdown (Voltage Islands)

Voltage Island concept is used to discontinue the power supply of an area of SoC powered by dedicated power supply [9]. Usually, this is accomplished by a dedicated power domain, supply to which is controlled by the external power switch. The technique above is usually applied when the die area and the circuitry, power supply of which is interrupted, is too large to be controlled by a single on-die power switch and when using a number of an on-die power switches instead isn't beneficial from area perspective. In some cases the voltage rail shutdown could be implemented using internal switch.

7. Performance and resources control

In this type of power management technique, only a subset of a total number of functional blocks (e.g. computational units) are engaged. The rest are either power gated or their clocks are gated. This method allows to save power in those situations where the dynamically monitored performance demand is below the maximal performance capacity. Powering down or clock-gating a functional module for a period of time when the given module's functionality is not required is also employed. For example, clock-gating a functional module for a short period of time may allow another functional module, supplying this module with data on request, enter a low-power, diagnostic or maintenance mode (e.g. DRAM self-refresh operation). Performance and resources control can be implemented with internal hardware or with software control.

Power Management Techniques	**Internal Power Switch**	**Voltage changes**	**Clock gating**	**Frequency changes**	**External Power shutdown**	**Power Supply Interrupted**
Frequency scaling	No	No	No	Yes	No	No
Clock gating	No	No	Yes	Yes	No	No
VFS	No	Yes	Yes	Yes	No	No
	Yes	Yes	Yes	Yes	No	No
Power Gating	Yes	Yes	--	--	No	Yes
Voltage Island shutdown	No	Yes	--	--	Yes	Yes
	Yes	Yes	--	--	No	Yes
Power State Transitions	No	Yes	Yes	Yes	No	No
Performance control	If PG enabled for this block	No	Yes	No	No	No

Table 1 Various Power Management Techniques

III. Power-Aware Verification

A. Power-Aware Design Flow

UPF generation flow, depicted in Fig. 2, is a key element of both Power Management implementation and Power-Aware verification of SOC ICs.

Along with RTL description, functional IPs teams provide parameters which include information about power-aware components and behavior of those IPs, such as isolation, power domains, retention policies, always-on logic, among the other parameters. Those parameters are processed by SOC CAD flow which generates UPF files corresponding to supplied power parameters and RTL. The generated UPF files are used for power-aware synthesis and scan insertion flow. They are also used by Design Verification (DV) team.

During Physical Design (PD) stage, the UPF files are further updated. For example, re-timing repeaters (flip-flops or buffers) are added. To incorporate the elements added during PD stage into power-aware simulation, flip-flop repeaters and feed-through cells (re-timing buffers chains crossing entire physical tile) could be extracted from the PD netlist and incorporated into the verification environment. This is done at Back-Annotated (BA) Feed-Through Buffer (FT) and Repeater Flop (RF) RTL stage, depicted in Fig. 2. Those elements are instrumental to the fidelity of the verification since PD-inserted repeater cells with wrong power supplies would pose a serious issue to the correct IC functionality. Power-aware netlist is usually available at the final stages of the project. Detailed discussion on the merits and disadvantages of power-aware gate-level simulation (PA GLS) is beyond the scope of this work. Such verification can be done at the advanced stages of the project, when the PD netlist is free from static checks violations. The RTL-level stages will usually provide most if not all the coverage of power-related features of the SoC – power management, power gating, DVFS, retention policies, among the others. Obviously, abandoning PA GLS will shorten the design cycle where power-aware verification is already one of the "long poles".

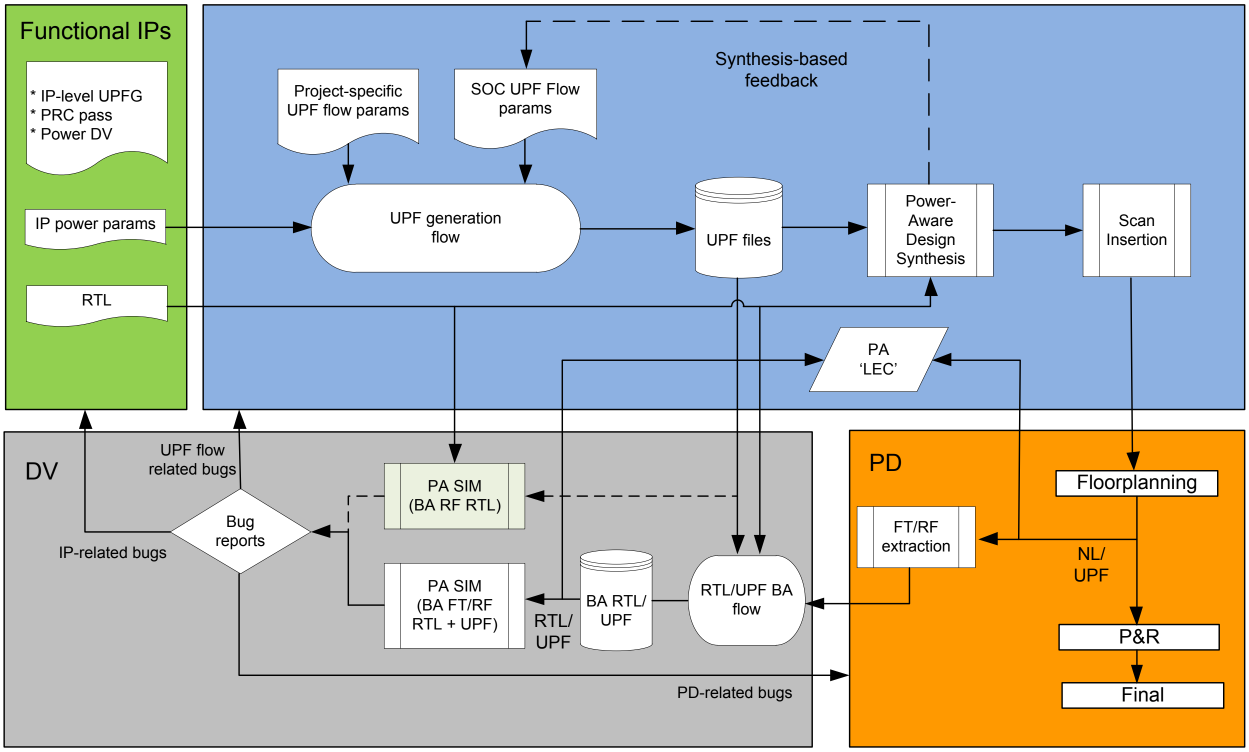

Fig. 2 Power-aware flow chart.

B. Power verification – RTL level

1. Non-power-gating verification

The non-power-gating verification category assumes that the power management of the circuitry in question is carried out without interrupting the power supply of the circuitry in question. In some of such techniques the voltage levels are regulated. However, power supply is never interrupted. Thanks to that, verification can be carried out without employing power-aware tools that emulate power interruption to parts of the design.

2. Power-aware verification

Power-aware verification category refers to the power-management techniques in which power supply to the areas in question is interrupted. For this category a power-aware simulation tools need to be employed for verification. Those tools, based on RTL and UPF description, emulate power events such as power gating. Those tools also infer design elements described outside RTL, in UPF, such as isolation ("clamp") cells.

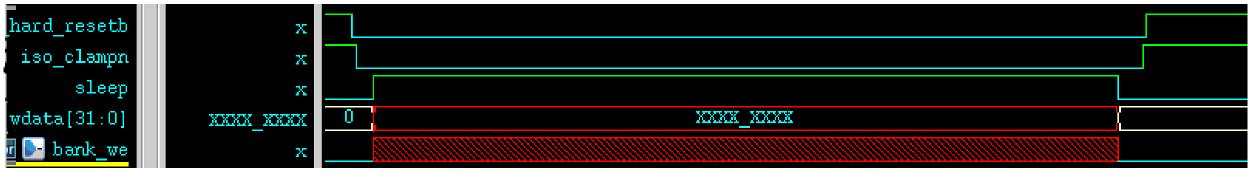

Fig. 3 Power-aware simulation of power gating event

Fig. *3* shows a waveform of power-gating simulation. The sequence of events is:

- Active-low reset signal "hard_resetb" is applied to the power-gated block.
- Active-low isolation control "iso_clampn" is asserted, activating isolation of the outside-bound signals from the voltage island area.
- Power gating control signal "sleep" is asserted. In the resulting circuit, "sleep" signal will control the power switch to the power-gated logic.

  The following is an excerpt from the UPF file describing the power switch:

```
create_power_switch SW_PD_P2 \
        -domain PD_P2 \
        -input_supply_port {vin VDD} \
        -output_supply_port {vout VDDINT_P2} \
        -control_port {NSLEEPIN1 P2_sleep} \
        -on_state {SW_ON vin {!NSLEEPIN1}} \
        -off_state {SW_OFF {!(!NSLEEPIN1)}}
map_power_switch SW_PD_P2 \
        -domain PD_P2 \
        -lib_cells {CELL_NAME}
```

- When the power is gated, we can see that the "X" is applied to the internal signals "wdata" and "bank_we". Applying "X" to powered-gated logic's signals allows spotting incorrect isolation. Outside-bound signals without proper isolation (which is also simulated by power-aware simulation tool using UPF files) will bear "X" values and will cause failure of the simulation. Once the power is restored, the simulator removes "X" from "wdata" and "bank_we".

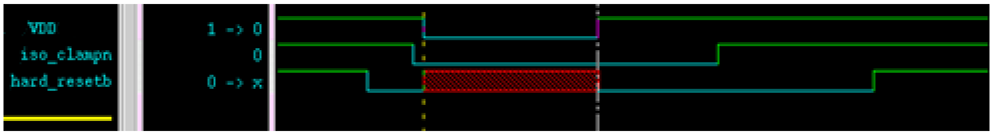

Fig. 4 Power-aware simulation of Voltage Island shutdown event

Fig. 4 shows a waveform of voltage island shutdown event simulation. The signals shown are (top to bottom): voltage island's power supply, isolation control, and the internal reset signal of the power-gated logic. The sequence of events is:

- Active-low reset signal "hard_resetb" is applied to the voltage island block prior to the power shutdown event.
- Active-low isolation control "iso_clampn" is asserted, activating isolation of the outside-bound signals from the voltage island area.
- Internal power supply VDD of the voltage island is interrupted. When the power is cut, we can see that the "X" is applied to the voltage island's internal signal "hard_resetb". Once the power is restored, the simulator removes "X" from the hard_resetb signal as seen in Fig. 4.

Additional aspect of power-aware RTL verification is modelling of memories used by the SoC. As a rule, power-related features of the memories are supplied within the open source Liberty modeling format memory library. The Liberty library would define such power features as power-gating, power gating controls, isolation of outputs coming from power-gated areas, among the other features. In this respect, memory models bear the description of power-aware features inside the model description rather than in UPF files.
The following excerpt will define internal power domain within memory which will be power gated when PG input of the memory is asserted:

```
pg_pin (VDD_PG) {
    pg_type : "internal_power";
    direction : internal;
    voltage_name : "VDD_PG";
    switch_function : "PG";
    pg_function : "VDD";
```

A number of additional techniques are applied to supplement power management verification.

- Functional assertions

Functional assertions make sure that the assumptions governing the protocols related to power events are observed during the simulations runs. One example is Power States Transition checker. During power states transitions, a set of conditions is checked. For example, during power gating events, the logic in question is first reset, then isolation ("clamps") is activated and only then the power is cut off ("gated"). The checker will also make sure that, in the case above, the output of the isolation cell is equal to one specified in the UPF file. An auto generated checker would make sure that the various conditions like those above are satisfied.

Functional assertions can also catch invalid power states transitions. When that happens, the offending power states or their transitions should be reviewed.

- Functional coverage

Functional coverage points make sure that all required power-related states and state transitions were exercised by verification testbench. One example could be Power States Table (PST) states and state transitions coverage.

C. Hybrid verification

Repeaters are introduced both during synthesis and physical design stages in order to satisfy chip's timing constraints and increase timing performance overall [10], [11], [12]. Those repeaters are implemented as either buffers or flip-flops. As a rule, flip-flop repeaters are back-annotated to the RTL view of the SoC in order to simulate the behavior of added memory elements. However, the power domain information contained in the UPF files is not back-annotated during this process. The repeater buffers introduced during synthesis and PD stages are usually not back-annotated since buffers don't change logic behavior of the RTL-level SoC.

Obviously, introduction of repeater elements at post-RTL stages presents a verification "hole" for power-aware verification since those elements were not verified at power-aware RTL verification stage.

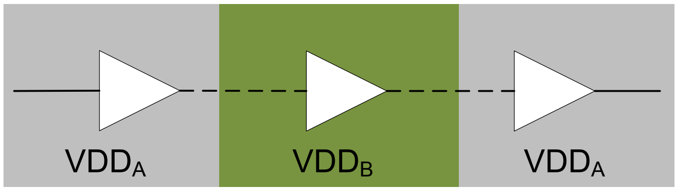

Fig. 5 Interconnect buffer repeaters allocated to different power domains

Fig. 5 depicts three interconnect re-timing buffers. Let's assume that the buffer (sometimes referred to as "feed-through") in the middle stage, connected to the power domain VDD_B, different from the power domain of the other two buffers, VDD_A. The functionality of the interconnect wire timed with the buffers will be interrupted if VDD_B power rail is power-gated.

In order to mitigate the risk involved in lack of verification of PD-added elements a new flow was proposed and implemented:

- Power-Aware Repeater/Feedthrough Back Annotation (PARBA) flow [2] provides additional safety layer simulating power-annotated repeaters and feed-through buffers, back-annotated from PD to RTL
- PARBA flow runs typically take place after PD team insert repeaters and feedthrough buffers for placement and timing purposes in the netlist
- PD-provided file containing flip-flop repeaters and feedthrough buffers information, including power domain allocation information, is used as input to CAD flow to re-group existing back-annotated repeater flip-flops based on power domains. The CAD flow also adds feedthrough buffers to the RTL. Both for flip-

flop repeaters and feedthrough buffers, UPF files are updated with their respective power domains information.

- UPF flow will use same PD-provided file with back annotated repeaters information to produce UPF containing repeater/feedthrough buffers power domains information.
- Static Power Rule Check (PRC) and power-aware Logic Equivalence Check (LEC) will ensure that synthesis and PD flow don't break the power intent of the IC
- Other static checking tools ensure that all power connections for analog IPs and macros are connected properly in the physical design all the way up through the IO pads ("bumps"), also ensuring that no power or ground pins are unconnected.

D. Power-aware Gate-level verification

In this section we don't analyze benefits of non-power-aware gate-level simulations but rather concentrate on power-aware gate-level verification. Power-aware GLS topic has been a point of extensive discussion among verification experts. Whereas one direction focuses on implementation of various techniques of GLS [13], [14]; other concentrate on justifications of inclusion of power-aware GLS in verification of ICs [15], [16], [17].

1) Issues and difficulties of Power-Aware GLS

The following are issues and difficulties generally mentioned in relation to power-aware gate-level simulations:

a) Since the complexity of ICs is constantly growing resulting in respective growth in the number of gate-level netlist components, a longer periods of time, often a number of months, are required for bring-up effort of power-aware GLS.

b) Long debug 'turnaround' time. Because of the reason mentioned above, the compilation and simulation time of the GLS in modern SoCs may become prohibitively long increasing debug cycle turnaround time.

c) Bring-up and debug effort associated with power-aware GLS consumes significant resources on interaction among many teams – IP, front-end integration, PD and DV.

d) Low coverage of power-aware GLS. Usually, because of size and complexity of the gatesim environment, only a few testcases usually used in power-aware GLS resulting in low coverage of such simulations.

2) Areas of concern and their solution

The main concern voiced regarding exclusion of power-aware GLS verification is DV coverage risk stemming from the power structures absent in RTL and added during synthesis or PD - clamps, power domains, domains crossing, among the others. Nevertheless, the following are the mitigation factors justifying exclusion of power-aware GLS:

a) Most of the power structures (especially those introduced during synthesis) are verified in power-aware RTL (clamps, power domains, domains crossing, etc.)

b) The PARBA flow described above simulates flip-flop repeaters and feed-through buffers, back-annotated from PD to RTL power-aware environment.

c) Power-aware LEC and PRC static checks guarantee that back-annotated RTL and UPF are equivalent to the floorplan netlist and floorplan UPF from PD.

d) As PD flow progresses through its stages (floorplan to place & route and to final netlist) power-aware LEC and PRC static checks are used at each step to guarantee PD netlist equivalence and compliance with Power Rules. Detailed description of static checks flow is beyond the scope of this paper.

8. Conclusions

In this paper we presented design verification techniques employed in verification of AMD Accelerated Processor Units (APUs) and Graphics Processing Units (GPUs). The employed methodology presents an approach balancing the demand for optimal test coverage of power-management features of SoC systems of growing complexity within given window of project duration. We categorized power management techniques and provided a detailed description of verification flow.

References

[1] S. Hemmady, et. al. B. Kapoor, "Impact of SoC power management techniques on verification and testing," in *Proceedings IEEE Int. Symposium on Quality Electronic Design*, 2009, pp. 692-695.

[2] Ashwini Chandrashekhara Holla, "Insight Into Power Gating Verification," in *SNUG Canada*, Ottawa, 2013.

[3] R. Mukherjee, A. Srivastava, and S. Kakkar F. Bembaron, "Low Power Verification Methodology using UPF," in *DVCON'2009*, 2009, pp. 228-233.

[4] B. Kapoor, A. Hunter, and P. Tiwari, "Power management verification experiences in Wireless SoCs," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011.

[5] C. Trummer et al., "Simulation-based verification of power aware System-on-Chip designs using UPF IEEE 1801," in *NORCHIP*, 2009, pp. 1-4.

[6] K. Chan, "Method to Partition Your Gate Simulation Debug," in *SNUG Canada*, 2015.

[7] E. Beigne and P. Vivet, "An innovative local adaptive voltage scaling architecture for on-chip variability compensation," in *New Circuits and Systems Conference (NEWCAS)*, 2011 , pp. 510-513.

[8] M. Nakai et al., "Dynamic voltage and frequency management for a low-power embedded microprocessor," *Solid-State Circuits, IEEE Journal of*, vol. 40, pp. 28-35, Jan. 2005.

[9] D.E. Lackey et al., "Managing power and performance for system-on-chip designs using Voltage Islands," in *Computer Aided Design, 2002. ICCAD 2002. IEEE/ACM International Conference on*, 2002, pp. 195-202.

[10] A. Nalamalpu and W. Burleson, "A practical approach to DSM repeater insertion: satisfying delay constraints while minimizing area and power," in *ASIC/SOC Conference*, 2001, pp. 152-156.

[11] K. Banerjee and A. Mehrotra, "A Power-Optimal Repeater Insertion Methodology for Global Interconnects in Nanometer Designs," *IEEE Transactions on Electroc Devices*, vol. 49, no. 11, p. 7, Nov 2002.

[12] Y. Peng and X. Liu, "An Efficient Low-Power Repeater-Insertion Scheme," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 25, no. 12, pp. 2726-2736, Dec. 2006.

[13] P. Daglio, D. Iezzi, D. Rimondi, C. Roma, and S. Santapa, "Building the hierarchy from a flat netlist for a fast and accurate post-layout simulation with parasitic components," in *Design, Automation and Test in Europe Conference*, 2004, pp. 336-337.

[14] D. Kim, M. Ciesielski, K. Shim, and S. Yang, "Temporal parallel simulation: A fast gate-level HDL simulation using higher level models," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011, pp. 1-6.

[15] G. Jalan. (2011, June) Gate Level Simulations : A Necessary Evil. [Online]. http://whatisverification.blogspot.ca/2011/06/gate-level-simulations-necessary-evil.html

[16] A. Chandran and R. Vincent, "Gatelevel Simulations – Continuing Value in Functional Simulation," in *DVCon India*, 2014.

[17] A. Khandelwal, A. Gaur, and D. Mahajan. (2014, March) Gate level simulations: verification flow and challenges. EDN network. [Online]. http://www.edn.com/design/integrated-circuit-design/4429282/Gate-level-simulations--verification-flow-and-challenges

[18] E. Beigne et al., "Fine-grain DVFS and AVFS techniques for complex SoC design: An overview of architectural solutions through technology nodes," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, 2015, pp. 1550-1553.