

Research Article

Open SystemC Simulator with Support for Power Gating Design

**George Sobral Silveira,¹ Alisson V. Brito,²
Helder F. de A. Oliveira,³ and Elmar U. K. Melcher³**

¹ Department of Electrical Engineering, Federal University of Campina Grande (UFCG), 58429-140 Campina Grande, PB, Brazil

² Department of Informatics, Federal University of Paraiba (UFPB), 58051-900 Joao Pessoa, PB, Brazil

³ Department of System and Computer, Federal University of Campina Grande (UFCG), 58429-140 Campina Grande, PB, Brazil

Correspondence should be addressed to Alisson V. Brito, alissonbrito@dce.ufpb.br

Received 17 June 2012; Revised 19 November 2012; Accepted 10 December 2012

Academic Editor: Oliver Sander

Copyright © 2012 George Sobral Silveira et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Power gating is one of the most efficient power consumption reduction techniques. However, when applied in several different parts of a complex design, functional verification becomes a challenge. Lately, the verification process of this technique has been executed in a Register-Transfer Level (RTL) abstraction, based on the Common Power Format (CPF) and the Unified Power Format (UPF). The purpose of this paper is to present an OSCI SystemC simulator with support to the power gating design. This simulator is an alternative to assist the functional verification accomplishment of systems modeled in RTL. The possibility of controlling the retention and isolation of power gated functional block (PGFB) is presented in this work, turning the simulations more stable and accurate. Two case studies are presented to demonstrate the new features of that simulator.

1. Introduction

Due to the new requirements that the consumer market has been imposing, the semiconductors industry suffered modifications in its manufacturing process. These evolutions introduced great challenges to the design with even more complex chips and high density of transistors in a tiny silicon area, leading to an inevitable increase of power and consequently heat dissipation in the chips [1]. Ahead of this fact, the industry and academic research centers are searching for new techniques to ease the power density problem and enable the development of low power ICs.

Techniques for reducing the power consumption can be applied during the development of an IC from the design and specification system to the layout stage [1]. Among the main techniques that can be highlighted are clock gating, multi-V_{th}, power gating, voltage islands, logic restructuring, and dynamic voltage and frequency scaling (DVFS). These techniques can be combined together and require additional features such as power management controllers, cell isolation power domains, and/or retention registers for logical values [2–4].

It is common to use Transaction-Level Modeling (TLM) and Register-Transfer Level (RTL) to accomplish the functional verification of complex system on chip (SoC) [1]. Functional verification is a processes used in order to demonstrate that the objectives of the design are preserved after its implementation [5]. In accordance with the state of the art, the power gating verification process has been executed at the Register-Transfer Level (RTL) [6–9], primarily, based on Common Power Format (CPF) and Unified Power Format (UPF) [10–15]. The purpose of this work is to demonstrate a SystemC simulator, open source, with support to functional verification of designs containing the principles of the power gating technique implemented in SystemC RTL. Power gating consists of powering down internal modules of the SoC. The analysis of verification methodologies for low power design is not the focus of this work, but an overview of the methodology used in the case studies will be demonstrated.

In the examples presented in this work, the VeriSC methodology was used [16]. It is based on SystemC language and features to propose a verification flow that does not begin by Design Under Verification (DUV) implementation.

Instead, the testbench and reference model are implemented before the DUV. The SystemC simulator is independent of VeriSC and of all other verification methodologies. In this simulator version, SystemC TLM is only used to implement the testbench and DUV is implemented using SystemC RTL.

The SystemC simulator was presented in [17]; in the occasion, some new features were added to the simulator allowing to switch any module on and off during simulation. In the present work, the possibility of controlling the retention and isolation of power gated functional block (PGFB) was added, turning the simulations to be more stable and more accurate in relation to the modeled hardware designs. This turned the modified SystemC simulator into an effective open-source tool to realize functional verification of low power designs.

The remainder of this work is organized as follows. Section 2 presents the current technologies for the functional verification of low power design; Section 3 shows a power gating overview; Section 4 shows the new functions added to the SystemC kernel; Section 5 shows two SystemC-LP simulator applications; Section 6 presents the results and analysis; and Section 7 presents the final considerations.

2. Technologies for the Functional Verification of Low Power Design

Several power saving techniques can be applied during the development of a SoC, from conception of the system's architectural design until prototyping. It is consensus that the techniques for low power SoC development, applied in initial phases of the conception flow, especially in the system design and architecture phase, possess a bigger impact factor in relation to techniques applied in the implementation phase [1].

The most efficient techniques for power savings are shutting down power or reducing the voltage level of some regions of the device, known as power domains [1]. In the first design generation, power-aware SoCs had few power domains, but recent designs have more than 20 domains, resulting in numerous power modes [18]. This leads to an exponential growth in the number of power-up and power-down transitions to be verified before silicon prototyping. Problems related to power can be extremely critical, resulting in the need to modify the SoC. Figure 1 shows a diagram expressing the impact ratio of the decisions, where 80% of the power savings are reached due to decisions taken before implementation [19].

Currently, five technologies are applied to the verification accomplishment: Power Definition Markup Language (PDML) specification, power-aware simulation, verification of the powered structure, relationships power assertions, and formal analysis of the logic control of power. These technologies are essential components of an effective verification aiming at high reliability of low power design techniques application [18], as follows.

- (i) PDML provides a way to specify the architecture of the powered design independently of the description

in RTL. The PDML specification includes the powered connectivity, shutdown behavior control, and interaction between the different power domains.

- (ii) Power-awareness simulation includes power management features, in which the RTL simulator reads and interprets the PDML, where the behavior in powerup and powerdown can be modeled.
- (iii) The verification of the powered structures can be performed at the beginning of the architectural power modeling.
- (iv) Some control functions and relationships with time specified in PDML can be automatically transformed into assertions using a standard format, such as the assertions in SystemVerilog or a proprietary specification language.
- (v) Formal analysis can be used to find all faults related to the assertions, where in simulation a given set of tests does not exercise all important behaviors related to power.

Currently, low power techniques are commonly applied at RTL level due to available technologies. The advantage of RTL verification is that there exists in the market a vast verification tool chain possibility from RTL to final graphic database system (GDSII). Therefore, once a given design is verified at RTL level, subsequent design stages only need to incrementally cover the additional low-power issues that are specific to that particular design stage.

During implementation of RTL-to-GDSII flow, most projects of integrated circuits have successfully incorporated power management using techniques such as clock gating, power gating, multi-VDD. The implementation of these techniques are not fully automated and the evaluation of the tradeoff between them is not easy. Some of these power analysis and optimization are assisted by Electronic Design Automation (EDA) tools [20, 21].

3. Overview of Power Gating

The designers have several ways to manage power, some are easily implemented and others are complex with respect to the operation frequency or area. Power-gating strategy is based on adding mechanisms to turn off blocks within the SoCs that are not being used; the act of turning blocks off and on is performed in appropriate time to achieve power saving while minimizing performance impact [1].

When the event of turning off happens, power saving is not instantaneous due to internal capacitances and the nature of technology that is not ideal for power gating. The process of turning a block on requires some time that cannot be ignored by the system designer [1]. Figure 2 shows an example of the activity of a block with power gating implemented.

Differently of a block that is always active, a power-gate block is powered by a power-switching network that will supply VDD or VSS. The Complementary Metal-Oxide-Semiconductor (CMOS) switches are distributed within or

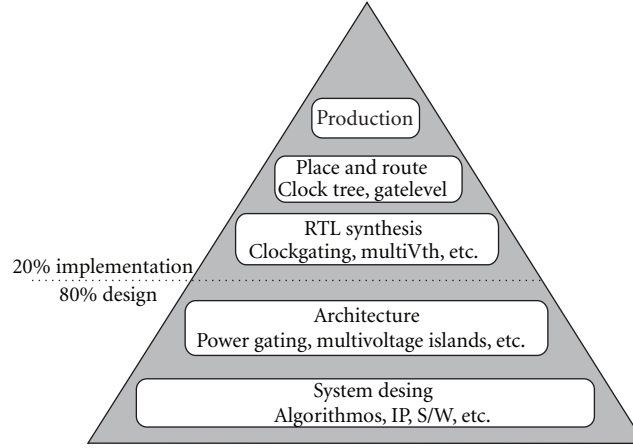


FIGURE 1: Impact factor of power savings [19].

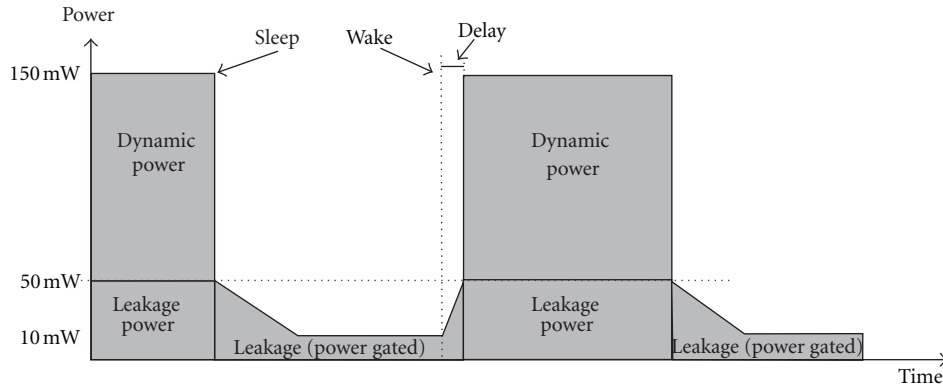


FIGURE 2: Profile with power gating, adapted [1].

around the block. The control of such switches is done by a power gating controller.

3.1. Signal Isolation. Each power domain represents a part of the chip's physical area, and even being independently switchable, these areas remain physically connected while they are turned on or off. Therefore, a logical isolation between the domains is necessary in order to avoid fluctuating signals and spread of undesired signals when the domain is off. The isolation is accomplished by logic gates which fix the input and/or output values when the power domain is turned off. In Figure 3, the isolation is represented by the block "isol."

3.2. State Retention. With the Power Gating Functional Block (PGFB) turned off, internal information about the state is lost. This may be inconvenience in certain applications. After the reactivation and restart of the PGFB activity, it should start from the initial state, reset, which can cause a significant consumption of energy and time. The retention can be accomplished in two ways: partial or total. There are several methods for saving and restoring the internal state of a PGFB, based on software, scan chains, and registers. Regardless the method, the goal is that the strategy adopted

for retention of the state is fast and efficient, providing the PGFB with a method to quickly restart the full operation after activation [1, 22, 23].

4. SystemC-LP Simulator

The support to power gating design with SystemC was based on creating functions that assist the power gating technique in semantic description. These functions are based on an approach developed to simulate partial and dynamic reconfiguration [24, 25] that originated the first simulator prototype [17]. This is a bottom-up approach adding functions to activate and deactivate modules by the programmer, simulating the run-time reconfiguration.

Two new special functions were implemented to turn modules on and off during simulation named *sc_lp_turn_on* and *sc_lp_turn_off*, respectively. These functions were written modifying the SystemC kernel source code. Table 1 presents the functions signatures in *sc_simcontext.h* SystemC kernel file. The routine can be called by user code on regular simulations.

The simulator kernel was rewritten using a Hash Map that stores modules attributes and the turn-on delay, always present when a module is reactivated on chip. The module is

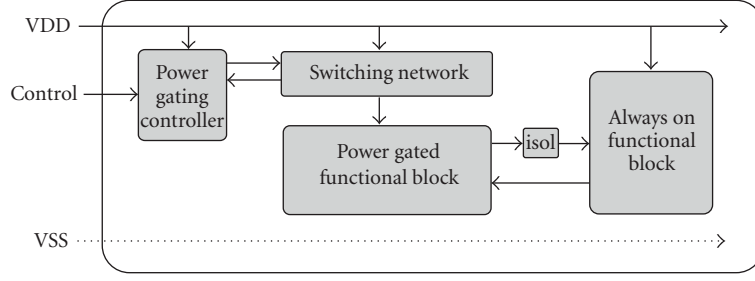


FIGURE 3: Block diagram of a SoC with power gating, adapted [1].

TABLE 1: Functions declarations.

```
void sc_lp_turn_on (const char* module_name, sc_time wakedelay);
void sc_lp_turn_off (const char* module_name);
```

TABLE 2: Functions signagtures of the PGLIB.

```
extern void pglib_int (const char* name, void* var, int value);
extern void pglib_int_set (const char* name);
extern void pglib_sc_lv (const char* name, void* var, sc_lv<64>
value);
extern void pglib_sc_lv_set (const char* name);
```

described in SystemC RTL using SC_METHOD. Each Hash Map element represents a design module and is composed of a data structure containing two variables (bool and sc_time). The boolean variable is responsible for identifying whether the module is on or not, and the variable sc_time is responsible for storing the delay necessary to reactivate the module. The elements are accessed using a key, which is the name of the module.

As presented in Section 3, the retention can be partial or total. It is desirable for the signal isolation cell that its output may change to one of the following states: (i) the output keeps the value before deactivation; (ii) the output is equal to one of the logic states “0,” “1,” “Z” or “X.” Thus, these features of signal retention and isolation were incorporated to the simulator.

Considering the signal retention, the default configuration stores the variable values, performing a total retention. However, for the output isolation the default is to keep the last value before deactivation. A library named Power Gating-Library (PGLIB) was developed to provide both, partial retention and output isolation, which must be added to the project source code to be used. These functions were not available in the first version of the simulator, shown [17]. The signatures of these functions are presented in Table 2.

In Table 2, two data types are used, Integer (*int*) and SystemC Logic Vector (*sc_lv*). For each data type, two functions were created. The first one is to declare the variables to be accessed (e.g., *pglib_int* and *pglib_sc_lv*) and the second one to set the values of those variables (e.g., *pglib_int_set* and *pglib_sc_lv_set*). The pointers to the variables and their values are stored in two Hash Maps, which use the name of the variables as keys to access their values.

The calculation of the power consumption is not the focus of this work and, therefore, was not implemented in the simulator.

5. Application of the SystemC-LP Simulator

To demonstrate the SystemC-Low Power (SC-LP) simulator, the functional verification of designs of [17, 26] is presented in this section.

5.1. Design 1. It consists of a circuit that serially receive words of 4 bits in the BCD format and convert to 7 segments format; after encoding, the data are available in the output [17]. This design has 450 code line and was implemented in RTL using SystemC version 2.2.0 with the features for power gating design. Figure 4 shows the design block diagram.

Two versions of the design were implemented in RTL using SystemC version 2.2.0, one with and another without the features for low power design.

- (i) *Design Version 1 (DV1.1)*, without low power design: The implementation was done in a simple way. The Serial2Parallel and BCD2Segment7 converters were implemented, the output of Serial2Parallel converter is connected directly to the BCD2Segment7 converter. The operation of the circuit is as follows: while the Serial2Parallel converter parallelize the bits that are read, the BCD2Segment7 converter encodes the values provided by the Serial2Parallel converter; after reading a complete word of 4 bits by Serial2Parallel converter, this module provides the result in the output.
- (ii) *Design Version 2 (DV1.2)*, with low power design: The implementation is similar to the preceding, differing in the addition of power gating controller (PGC) functions. The operation of this version differs from the previous in the following point: while the Serial2Parallel converter reads the word of 4 bits, the BCD2Segment7 module is turned off with its state retained. After reading the complete word BCD2Segment7 module is turned on, which from this point waits a time of 150 ns corresponding to the delay between wake up and activity; after this time the module executes its function and on the next step provides the value in the output. With the assistance of PGLIB, there is no retention values of the internal

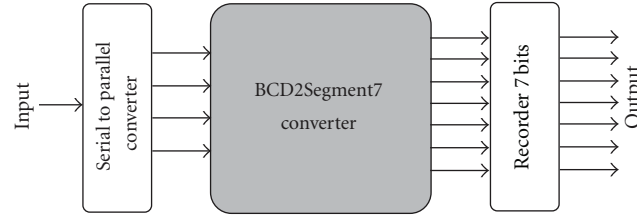


FIGURE 4: Design block diagram.

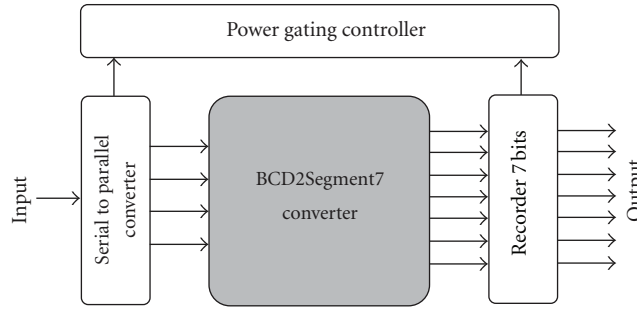


FIGURE 5: Design block diagram with PGC.

variables BCD2Segment7 module. When the module is disabled, all variables receive the value zero. The module output is isolated and its value is retained. Figure 5 shows the design block diagram with PGC.

5.2. Design 2. It consists of MPEG-4 decoder [26]; the IP core is a Simple Profile Level 0 movie decoder with approximately 21,000 line code and was implemented in a silicon chip, with 22.7 mm² at a 0.35 μm CMOS 4 ML technology with a 25 MHz working frequency. The block schematic of the MPEG-4 decoder is described in Figure 6.

The decoder circuit is divided into 10 modules, grouped into 4 functional groups, comprising bitstream demultiplexing, motion compensation, texture decoding, and image composition. The power gating technique was applied to the Inverse Discrete Cosine Transform (IDCT) submodule. This submodule was chosen because it presents inactivity time periods during its operation. The block schematic of the MPEG-4 decoder with PGC is described in Figure 7.

The PGC submodule has been added, described in SystemC RTL, with the function of calling the new SystemC-LP. The PGC receives information from Inverse Quantization (IQ) and Summing (SUM) submodules using wires. The PGC operates in the following way: it activates the IDCT whenever the IQ has information to be sent to the IDCT and the SUM is able to receive information from the IDCT. Otherwise, the later remains off. Using PGLIB, when the IDCT module is disabled only some configuration values of IDCT module are stored, the other internal variables receive the value zero. The module output is isolated and its value is retained.

6. Functional Verification

The VeriSC methodology supports projects with hierarchy concept; therefore, a project can be divided into parts to be implemented and verified [16]. The methodology consists of a verification flow, which is not initiated by the implementation of DUV. In this flow, the testbench implementation and the reference model precede the DUV. To allow the implementation of testbench before the DUV, the methodology has a mechanism to simulate the presence of DUV elements with their own testbench, without needing additional code generation. The testbench structure is described in Figure 8. The Reference Model, Source, and Check are described in ESL level, using SystemC TLM.

The source block is responsible for generating the stimuli. They are sent to DUV and Reference Model. The check block is responsible for checking whether the responses are the same. The TDriver converts transactions into signals, while the TMonitor converts signals into transactions. The reference model calculates all expected outputs of the system based on inputs that are used during functional verification. Ideally, separate engineers' teams must implement the DUV and the reference model. The goal is to decrease the probability of erroneous interpretations of the system specification propagate to DUV and verification environment.

BVE-Cover library was chosen to accomplish the functional verification with coverage of the design. Several simulations were performed with different versions of SystemC simulator and design.

- (1) SC + DV1.1: At this stage the original SystemC version 2.2.0 and the first design implementation, were used.

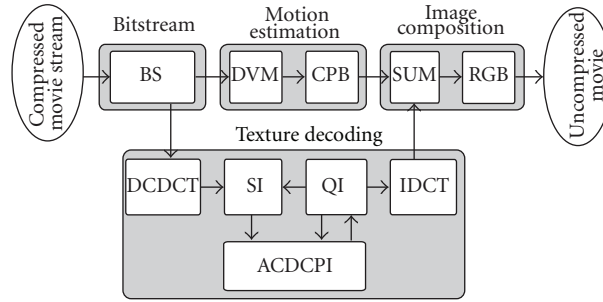


FIGURE 6: MPEG-4 decoder schematic.

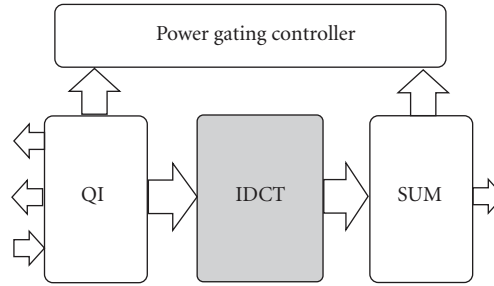


FIGURE 7: IDCT and PGC submodule connections.

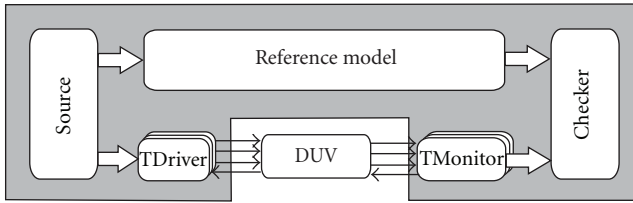


FIGURE 8: Testbench structure.

- (2) SC-LP + DV1.1: At this stage the SystemC-LP and the first design implementation, without power gating design, were used.
- (3) SC-LP + DV1.2: At this stage the SystemC-LP and the second design implementation containing the power gating design were used.
- (4) SC + DV2.1: At this stage the original SystemC version 2.2.0 and MPEG-4 decoder were used.
- (5) SC-LP + DV2.1: At this used the SystemC-LP and MPEG-4 decoder were used.
- (6) SC-LP + DV2.2: At this stage the SystemC-LP and MPEG-4 decoder containing Power Gating technique were used.

7. Results and Analysis

With the simulations results, it is possible to see the semantics of power gating design and that new features added to the simulator do not interfere with the other functions. The principles of power gating can be verified in Figure 9, which presents the waveform of simulations DV1.1.

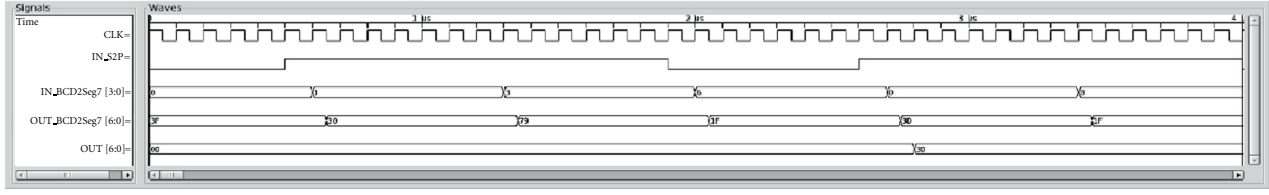
Figure 9(a) shows the changes of state of the BCD2Segment7 module while the Serial2Parallel module parallels the words.

In Figure 9(b), it is possible to see that while the parallels occur, the word of the BCD2Segment7 module maintains constant output state due to the isolation and retention that occurs prior to shut down. After the word reading by the BCD2Segment7 module plus the time delay, it turns back to activity and executes its functions to provide the values in the output.

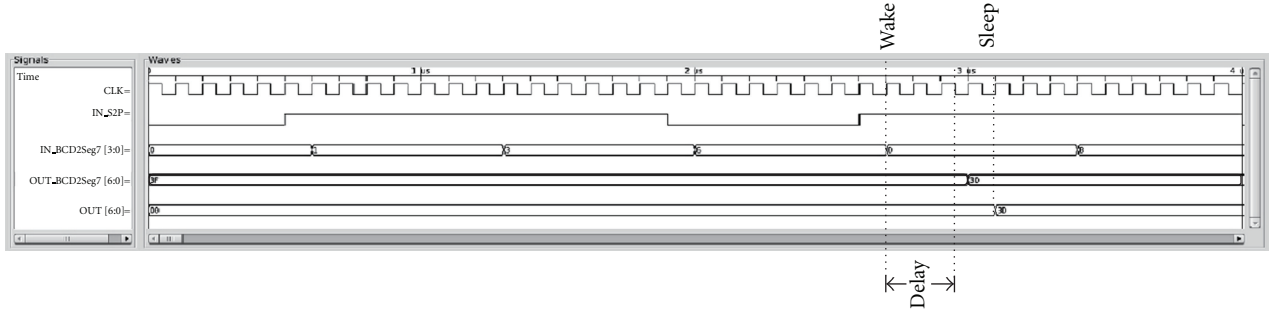
Relevant information which can be extracted from the log is the $10\mu s$ time that the submodule BCD2segment7 remains activated, which corresponds to 37% of the total time. For the IDCT submodule, it only remained activated for 41% of the total time. This value can be used as a parameter for estimating the power savings using the power gating technique.

About the simulator performance, satisfactory results were achieved. Table 3 shows values with the averages of the simulations times. It can be seen that the design simulations, using the SC-LP simulator presented a decrease of approximately 7,33% in simulation time, which shows that, even with the changes in the kernel simulator, this has performance similar with low and high complexity design.

The simulations of power gating design DV2.2 presented a decrease of approximately 6,12% in comparison with the original SystemC simulator using version 2.2.0. Comparing the simulation 5 with 6, it is possible to observe that the simulation 6 was faster than 5. It happens because in simulation 6 the IDCT module is not executed when it is deactivated. Based on that, it is possible to say that the power gating simulator can have an equivalent performance to the original SystemC simulator using RTL, depending on how the features are applied.



(a) Without power gate design



(b) With power gate design

FIGURE 9: Waveforms.

TABLE 3: Simulators performance.

Number	Simulator	Design	Time (s)
4	SC	DV2.1	466,403
5	SC-LP	DV2.1	503,315
6	SC-LP	DV2.2	496,851

8. Final Considerations

The research academic centers and industry have expended a great effort into finding alternatives to saving power in chips. This work shows an open-source simulator with technical support of power gating design. The SC-LP is an alternative tool for functional verification of the power gating design technique modeled in RTL. This presented a satisfactory performance in relation to SystemC simulator version 2.2.0 during the accomplishing power gating design simulations. In a preliminary analysis, although the main modifications in SystemC simulator version 2.3, the benefits presented in this work are still possible to be used, as the SystemC has support to block just models that use SC.THREAD, while our solution work mainly with SC.METHOD, which is the basis for RTL models.

Acknowledgment

This work is supported by “Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)” by a postgraduate research scholarship.

References

- [1] M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual: For System-on-Chip Design*, Springer, New York, NY, USA, 2007.
- [2] S.-H. Chen and J. Y. Lin, “Experiences of low power design implementation and verification,” in *Proceedings of the Asia and South Pacific Design Automation Conference (ASP-DAC '08)*, pp. 742–747, Seoul, South Korea, March 2008.
- [3] S. Jadcherla, J. Bergeron, Y. Inoue, and D. Flynn, *Verification Methodology Manual for Low Power*, Synopsys, 2009.
- [4] S. Henzler, *Power Management of Digital Circuits in Deep Sub-Micron CMOS Technologies*, Springer, 2007.
- [5] J. Bergeron, *Writing Testbenches Using System Verilog*, Springer, Boston, Mass, USA, 2006.
- [6] C. Eisner, A. Nahir, and K. Yorav, “Functional verification of power gated designs by compositional reasoning,” *Formal Methods in System Design*, vol. 35, no. 1, pp. 40–55, 2009.
- [7] V. Viswanath, S. Vasudevan, and J. A. Abraham, “Dedicated rewriting: automatic verification of low power transformations in RTL,” in *Proceedings of the 22nd International Conference on VLSI Design (VLSI '09)*, pp. 77–82, New Delhi, India, January 2009.
- [8] J. Kim, S. Kim, and K. H. Baek, “A low power SOC architecture for the V2.0+EDR bluetooth using a unified verification platform,” *IEICE Transactions on Information and Systems*, vol. E93-D, no. 9, pp. 2500–2508, 2010.
- [9] M. A. Pasha, S. Derrien, and O. Sentieys, “System level synthesis for ultra low-power wireless sensor nodes,” in *Proceedings of the 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD '10)*, pp. 493–500, Lille, France, September 2010.
- [10] T.-W. Hsieh, P. C. Hsiao, C. Y. Liao et al., “Energy-effective design & implementation of an embedded VLIW DSP,” in *Proceedings of the International SoC Design Conference (ISOC '08)*, pp. I252–I255, Busan, South Korea, November 2008.
- [11] N. Joseph, S. Sabarinath, and K. Sankarapandiammal, “FPGA based implementation of high performance architectural level low power 32-bit RISC core,” in *Proceedings of the International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom '09)*, pp. 53–57, Kottayam, India, October 2009.

- [12] S.-H. Chen and J. Y. Lin, "Implementation and verification practices of DVFS and power gating," in *Proceedings of the International Symposium on VLSI Design, Automation and Test (VLSI-DAT '09)*, pp. 19–22, Hsinchu, Taiwan, April 2009.
- [13] G. Gammie, A. Wang, H. Mair et al., "Smart reflex power and performance management technologies for 90 nm, 65 nm, and 45 nm mobile application processors," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 144–159, 2010.
- [14] M. Kuwahara, "Design and verification methods of Toshiba's wireless LAN baseband SoC," in *Proceedings of the 15th Asia and South Pacific Design Automation Conference (ASP-DAC '10)*, pp. 457–463, Taipei, Taiwan, January 2010.
- [15] C. Trummer, C. M. Kirchsteiger, C. Steger, R. Weiß, M. Pistauer, and D. Dalton, "Automated simulation-based verification of power requirements for systems-on-chips," in *Proceedings of the 13th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS '10)*, pp. 8–11, Vienna, Austria, April 2010.
- [16] K. R. G. da Silva, E. U. K. Melcher, I. Maia, and H. D. N. Cunha, "A methodology aimed at better integration of functional verification and RTL design," *Design Automation for Embedded Systems*, vol. 10, no. 4, pp. 285–298, 2005.
- [17] G. S. Silveira, A. V. Brito, and E. U. K. Melcher, "Functional verification of power gate design in systemc RTL," in *Proceedings of the 22nd Symposium on Integrated Circuits and Systems Design (SBCCI '09)*, Natal, Brazil, September 2009.
- [18] T. Anderson, "Power Mode Technologies Verify Today's SoCs," *EETimes*, 2008, <http://www.eetimes.com/showArticle.jhtml?articleID=206900466>.
- [19] A. Mathur, *Sequential Transformations for Low Power and CPF*, 2008.
- [20] Mentor Graphic's, "Mentor Graphics Low Power Solutions," Mentor Graphic's, 2012, <http://www.mentor.com/solutions/low-power/functional-verification>.
- [21] Cadence, "Cadence—Low-Power Solution," Cadence, 2012, <http://www.cadence.com/solutions/lp/pages/default.aspx>.
- [22] D.-C. Juan, Y. T. Chen, M. C. Lee, and S. C. Chang, "An efficient wake-up strategy considering spurious glitches phenomenon for power gating designs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 2, pp. 246–255, 2010.
- [23] M.-C. Lee, Y. T. Chen, Y. T. Cheng, and S. C. Chang, "An efficient wakeup scheduling considering resource constraint for sensor-based power gating designs," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD '09)*, pp. 457–460, San Jose, Calif, USA, November 2009.
- [24] A. V. Brito, M. Kühnle, M. Hübner, J. Becker, and E. U. K. Melcher, "Modelling and simulation of dynamic and partially reconfigurable systems using SystemC," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI: Emerging VLSI Technologies and Architectures (ISVLSI '07)*, pp. 35–40, Porto Alegre, Brazil, March 2007.
- [25] A. V. De Brito, E. U. K. Melcher, and W. Rosas, "An open-source tool for simulation of partially reconfigurable systems using SystemC," in *Proceedings of the IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI '06)*, pp. 434–435, Karlsruhe, Germany, March 2006.
- [26] A. Rocha, P. Lira, Y. Ju, E. Barros, and E. Melcher, "Silicon validated ip cores designed by the brazil-ip network," in *Proceedings of the IP Based SoC Design Conference and Exhibition (IP/SOC '06)*, pp. 1–5, 2006.

