VLSI Design

Advanced VLSI Architecture Design for Emerging Digital Systems

Guest Editors: Yu-Cheng Fan, Qiaoyan Yu, Thomas Schumann, Ying-Ren Chien, and Chih-Cheng Lu

Advanced VLSI Architecture Design for Emerging Digital Systems

VLSI Design

Advanced VLSI Architecture Design for Emerging Digital Systems

Guest Editors: Yu-Cheng Fan, Qiaoyan Yu,
Thomas Schumann, Ying-Ren Chien, and Chih-Cheng Lu

Editorial Board

Contents

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 746132, 2 pages
http://dx.doi.org/10.1155/2014/746132

Editorial

Advanced VLSI Architecture Design for Emerging Digital Systems

Yu-Cheng Fan,[1] Qiaoyan Yu,[2] Thomas Schumann,[3] Ying-Ren Chien,[4] and Chih-Cheng Lu[5]

[1]*Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan*
[2]*Department of Electrical and Computer Engineering, University of New Hampshire, Durham, NH 03824, USA*
[3]*Department of Electrical Engineering and Information Technology, Hochschule Darmstadt-University of Applied Sciences, Birkenweg 8, 64295 Darmstadt, Germany*
[4]*Department of Electrical Engineering, National Ilan University, Yilan 260, Taiwan*
[5]*Division for Biomedical & Industrial IC Technology, Industrial Technology Research Institute, Hsinchu 310, Taiwan*

Correspondence should be addressed to Yu-Cheng Fan; skystar@ntut.edu.tw

Received 29 September 2014; Accepted 29 September 2014; Published 22 December 2014

With physical feature sizes in VLSI designs decreasing rapidly, existing efficient architecture designs need be reexamined. Advanced VLSI architecture designs are required to further reduce power consumption, compress chip area, and speed up operating frequency for high performance integrated circuits. With time-to-market pressure and rising mask costs in the semiconductor industry, engineering change order (ECO) design methodology plays a main role in advanced chip design. Digital systems such as communication and multimedia applications demand for advanced VLSI architecture design methodologies so that low power consumption, small area overhead, high speed, and low cost can be achieved.

This special issue is dedicated to aspects of VLSI architecture design and their applications. Special interest focuses on emerging digital systems. This special issue contains eight papers that focus on the power minimization design, efficient hardware Trojan detection, low-area Wallace multiplier, gate-level circuit reliability analysis, low power and high speed arithmetic circuits, power effective fractional-N-PLL frequency synthesizer, power-saving architecture for network on chip, and ECO design.

In the paper entitled "*On-chip power minimization using serialization-widening with frequent value encoding*," the authors address the problem of the high-power consumption of the on-chip data buses by exploring a new framework for memory data bus. In particular, serialization-widening (SW) of data bus with frequent value encoding (FVE) is proposed to minimize the power consumption of the on-chip cache data bus.

In the paper entitled "*Efficient hardware trojan detection with differential cascade voltage switch logic*," the authors present to exploit the inherent feature of differential cascade voltage switch logic (DCVSL) to detect hardware trojans (HTs) at runtime. By examining special power characteristics of DCVSL systems upon HT insertion, the authors can detect HTs, even if the HT size is small. Simulation results show that the method achieves up to 100% HT detection rate. The evaluation on ISCAS benchmark circuits shows that the scheme obtains a HT detection rate in the range of 66% to 98%.

In the paper entitled "*Low-Area Wallace multiplier*," the authors propose a reduced-area Wallace multiplier without compromising on the speed of the original Wallace multiplier. The proposed designs are synthesized using Synopsys Design Compiler in 90 nm process technology and achieve the lowest area cost as compared to other tree-based multipliers. The speed of the proposed and reference multipliers is almost the same.

In the paper entitled "*Gate-level circuit reliability analysis: a survey*," the authors provide an overview of some typical methods for reliability analysis with special focus on gate-level circuits that are either large or small, with or without

reconvergent fan-outs. It is intended to help the readers gain an insight into the reliability issues and their complexity as well as optional solutions. Understanding the reliability analysis is also a first step towards advanced circuit designs for improved reliability in the future research.

In the paper entitled "*Performance analysis of modified drain gating techniques for low power and high speed arithmetic circuits*," the authors present several high performance and low power techniques for CMOS circuits. In these design methodologies, drain gating technique and its variations are modified by adding an additional NMOS sleep transistor at the output node. This method helps to achieve faster discharge and thereby provides higher speed. Intensive simulations are performed using Cadence Virtuoso in a 45 nm standard CMOS technology at room temperature with supply voltage of 1.2 V. Comparative analysis of the present circuits with standard CMOS circuits shows smaller propagation delay and lesser power consumption.

In the paper entitled "*Optimization of fractional-N-PLL frequency synthesizer for power effective design*," the authors design a low power fractional-N phase-locked loop (FNPLL) frequency synthesizer for industrial application, which is based on VLSI. The design of FNPLL has been optimized using different VLSI techniques to acquire significant performance in terms of speed with relatively less power consumption.

In the paper entitled "*Design of smart power-saving architecture for network on chip*," the authors present a novel architecture, namely, smart power-saving (SPS), for low power consumption and low area in virtual channels of NoC. Comparing with related works, the new proposed method reduces with 37.31%, 45.79%, and 19.26% on power consumption and reduces with 49.4%, 25.5%, and 14.4% on area, respectively.

Finally, in the paper entitled "*Engineering change orders design using multiple variables linear programming for VLSI design*," the authors present an engineering change orders (ECO) design using multiple variable linear programming for VLSI design. The authors adopt linear programming technique to plan and balance the spare cells and target cells to meet the new specification according to logic transformation. The proposed method solves the related problem of resource for ECO problems and provides a hardware-efficient solution.

Acknowledgments

Finally, the Guest Editors would like to thank all the authors who sent their valuable contributions and all the reviewers for their valuable comments.

Yu-Cheng Fan
Qiaoyan Yu
Thomas Schumann
Ying-Ren Chien
Chih-Cheng Lu

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 698041, 5 pages
http://dx.doi.org/10.1155/2014/698041

Research Article

Engineering Change Orders Design Using Multiple Variables Linear Programming for VLSI Design

Yu-Cheng Fan, Chih-Kang Lin, Shih-Ying Chou, Chun-Hung Wang, Shu-Hsien Wu, and Hung-Kuan Liu

Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

Correspondence should be addressed to Yu-Cheng Fan; skystar@ntut.edu.tw

Received 7 June 2014; Accepted 18 July 2014; Published 24 August 2014

Academic Editor: Chih-Cheng Lu

An engineering change orders design using multiple variable linear programming for VLSI design is presented in this paper. This approach addresses the main issues of resource between spare cells and target cells. We adopt linear programming technique to plan and balance the spare cells and target cells to meet the new specification according to logic transformation. The proposed method solves the related problem of resource for ECO problems and provides a well solution. The scheme shows new concept to manage the spare cells to meet possible target cells for ECO research.

1. Introduction

Engineering change orders (ECO) are important technologies used for changes in integrated circuit (IC) layout and compensate for design problems. Traditionally, when chip shows errors, it often requires new photomasks for all layers. However, photomasks of deep-submicron semiconductor fabrication process are very expensive. In order to save money, ECO technology modifies only a few of the metal layers (metal-mask ECO) to reduce the cost of photomasks for all layers [1].

To perform the ECO, IC designers adopt sprinkling many unused logic gates during IC design flow. When chip is manufactured and shows design errors, IC designers modify the gate-level net-list using the presprinkling unused logic gates. At the same time, the designers track and verify the modification to check formal equivalence after ECO process. The designers must guarantee the revised design matching the revised specification.

How to achieve ECO efficiently? There are some literatures that address this problem and provide related solution. In literature [2], Tan and Jiang describe a typical metal-only ECO flow with four steps that include placement and spare cell distribution, logic difference extraction, metal-only ECO synthesis, and ECO routing [2]. Kuo et al. insert spare cells with constant insertion for engineering change and describe an iterative method to determine feasible mapping solutions for an EC problem [3]. Besides, in order to perform ECO efficiently, literature [4–9] adopt minimal change EC equations automatically. Brand proposed incremental synthesis method [4]. Huang presented a hybrid tool for automatic logic rectification [5]. Lin et al. addressed logic synthesis techniques for engineering change problems [6]. Shinsha et al. performed incremental logic synthesis through gate logic structure identification [7]. Swamy et al. achieved minimal logic resynthesis for engineering change [8]. Watanabe and Brayton presented another kind of incremental synthesis technique for engineering changes [9]. However, few researchers discuss the resource between spare cells and target cells. Therefore, in order to solve the problems, we adopt linear programming technique to plan and balance the spare cells and target cells in this paper. The proposed scheme meets the new specification according to logic transformation and overcomes the related problems of resource for ECO research.

This paper is organized as follows. In Section 2, we address typical ECO design flow. In Section 3, logic transformation is discussed. In Section 4, multiple variables linear

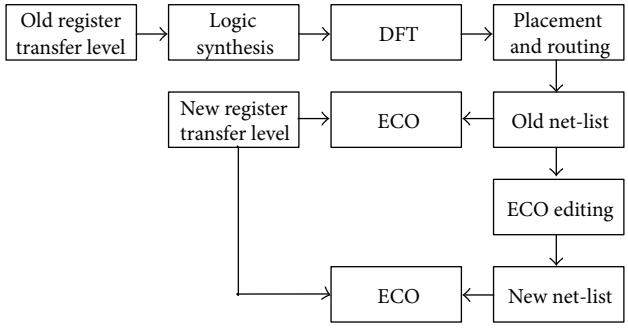


Figure 1: A typical ECO design flow.

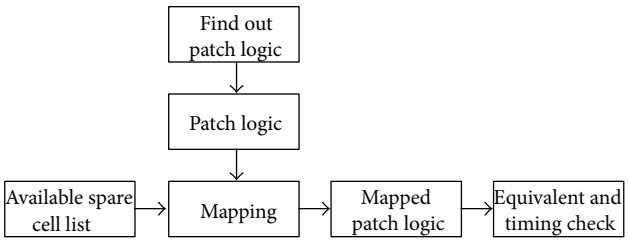


Figure 2: Two-phase ECO design flow.

programming for VLSI design is presented. In Section 5, we discuss the advantage and disadvantage of the related works. Finally, we conclude this paper in Section 6.

2. Typical ECO Design Flow

Before describing the proposed method, we address a typical manual ECO design flow in Figure 1. IC designers perform the change in register transfer level and verify fixed code matching the new specification at first. Then, old net-list is scanned to search the possible fix points. After the possible fix points are searched, IC designers modify the net-list and check the functionally equivalent between new net-list and new register transfer level [10–14].

Next, we describe two-phase ECO design flow in Figure 2. To patch the logics of the modified circuit, we prepare available spare cell list. According to logic function, the modified circuit is mapped to specified logics. After patching logic, equivalent check and timing check are performed to make sure that the new function met the new specification.

However, there are some important problems that appear during patching logic. Are there enough spare cells and types to satisfy the consumption of patch logic? How to estimate the quantity and logic types of ECO procedure? In order to solve this problem, we proposed an engineering change orders design using multiple variables linear programming for VLSI design in this paper.

3. Logic Transformation

Before discussing the engineering change orders design using multiple variables linear programming, we addressed ECO logic transformation. Figure 3 describes an ECO problem with an equation out = $(A + (BC)')'$. Figure 3(b) lists the available spare cells. According to the list, we discover the available spare cells are not enough. In order to solve the problem, we adopt another mapping solution with an equation out = $(A'BC)$ instead of the original equation in Figure 3(c). It requires one AND and one INV gate. The mapping solution in Figure 3(c) requires gates fewer than the available spare cells and is constructed with the available spare cells.

However, most of spare cells only provide basic logical functions that include AND, OR, NOT, NAND, and NOR. Half Adder (HA), Full Adder (FA), And-Or-Inverter (AOI), and Or-And-Inverter (OAI) can provide complex logical functions. We can adopt these logical cells to perform ECO function. For example, AOI22 can be implemented by two NAND and one AND cells in Figure 4. According to the existing resources of spare cells, we can resynthesize the changed function lists.

4. Multiple Variables Linear Programming for VLSI Design

Although logic transformation skill makes the ECO technology come true, a chip often does not own enough spare cells to modify the function to meet a new specification. How to allocate limited resource? We should estimate quantity of spare cells and logic transformation rule to perform optimal engineering charge orders.

In Figure 5, it describes the engineering change orders design using multiple variables linear programming for VLSI design and relation of logic transformation. "Logic A" is one kind of spare cells that can be transformed into "Logic a" or "Logic b." Similarly, "Logic B" can be transformed into "Logic a," "Logic b," or "Logic c." "Logic C" performs ECO function instead of "Logic c" or "Logic d." Besides, Logic D is transformed into "Logic c" or "Logic d." Equivalently, "Logic E" is transformed to "Logic d" or "Logic e" to achieve ECO function.

We assume X_1, X_2, X_3, X_4, and X_5 are the number of spare cells, Logic A, Logic B, Logic C, Logic D, and Logic E. Let Y_1, Y_2, Y_3, Y_4, and Y_5 be the desired number of target cells, Logic a, Logic b, Logic c, Logic d, and Logic e.

Besides, Aa is the number of spare cells (Logic A) to be transformed into Logic a and Ab is the number of spare cells (Logic A) to be transformed into Logic b. Similarly, Ba, Bb, and Bc are the number of spare cells (Logic B) to be transformed into Logic a, Logic b, and Logic c. In a similar way, Cc and Cd are the number of spare cells (Logic C) to be transformed into Logic c and Logic d, Dc and Dd are the number of spare cells (Logic D) to be transformed into Logic c and Logic d, and Ed and Ee are the number of spare cells (Logic E) to be transformed into Logic d and Logic e.

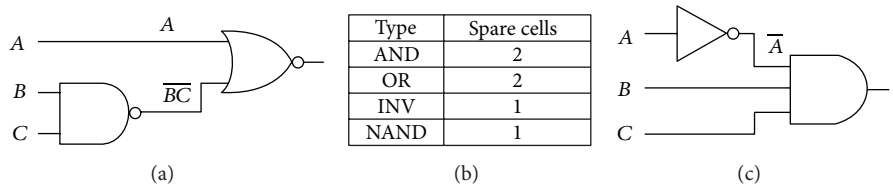


Figure 3: Example of an ECO problem. (a) EC equation: output = $(A + (BC)')'$. (b) Spare cells. (c) Mapping: output = $(A'BC)$.

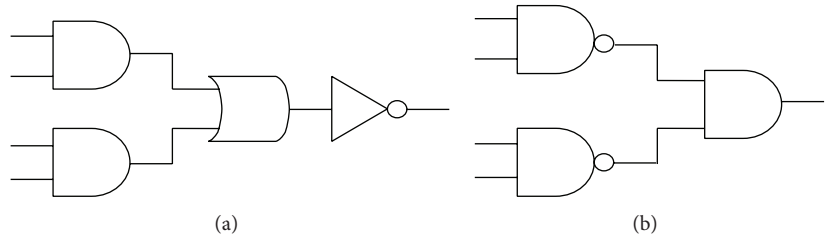


Figure 4: AOI22 can be implemented by two NAND and one AND cells.

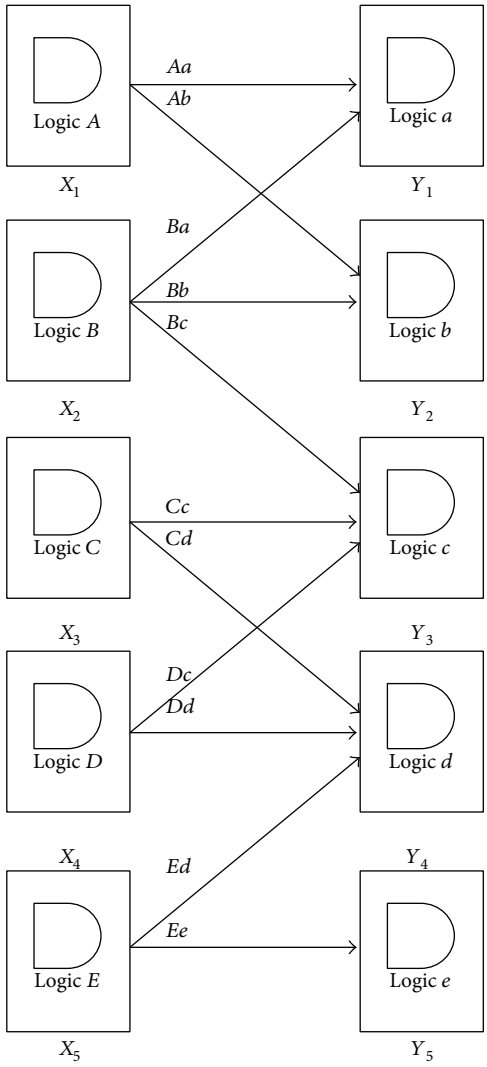


Figure 5: ECO design using multiple variables linear programming for VLSI design and relation of logic transformation.

Therefore, the restriction rule of the number of spare cells and transformed target cells in Figure 5 is written as follows:

$$\begin{aligned} X_1 &= Aa + Ab; \\ X_2 &= Ba + Bb + Bc; \\ X_3 &= Cc + Cd; \\ X_4 &= Dc + Dd; \\ X_5 &= Ed + Ee. \end{aligned} \tag{1}$$

Besides, the restriction rule of the engineering change orders design using multiple variables linear programming in Figure 5 is written as follows:

$$Aa + Ba \geqq Y_1; \tag{2}$$

$$Ab + Bb \geqq Y_2; \tag{3}$$

$$Bc + Cc + Dc \geqq Y_3; \tag{4}$$

$$Cd + Dd + Ed \geqq Y_4; \tag{5}$$

$$Ed + Ee \geqq Y_5. \tag{6}$$

However, spare cells are not often enough; designer should balance the spare cell allocation to meet all requirements of desirable cells.

We assume one case when $Bb \leqq Y_2$. In order to provide enough spare cells, we should increase the number of Ab to achieve $Ab + Bb \geqq Y_2$.

Similarly, when $Cc + Dc \leqq Y_3$, we should increase Bc number to meet $Bc + Cc + Dc \geqq Y_3$.

Therefore, we define another restriction rule of the engineering change orders design which is written as follows:

$$\begin{aligned} Aa &= X_1 - Ab; \\ Ba &= X_2 - Bb - Bc. \end{aligned} \tag{7}$$

Table 1: ECO methods comparison.

Method	Traditional ECO	Proposed method
Cell resource prediction	Constraint based	Multiple variables linear programming
Predictive precision of patching logic number	Normal precision	High precision
Balance between spare cells and target cells	Low balance	High balance
Restriction rule	Not define	Define
Resource optimization	Not define	Define
Solution boundary	Not define	Define

According to formulas (2) and (7), we can balance the number of Ab, Bb, and Bc to achieve the target number Y_1. Consider the following:

$$X_1 - Ab + X_2 - Bb - Bc \geqq Y_1. \tag{8}$$

In a similar way, we define the restriction rule of the engineering change orders design which is written as follows:

$$\begin{aligned} Bc &= X_2 - Ba - Bb; \\ Cc &= X_3 - Cd; \\ Dc &= X_4 - Dd. \end{aligned} \tag{9}$$

According to formulas (4) and (9), we can balance the number of Bc, Cc, and Dc to achieve the target number Y_3. Consider

$$X_2 - Ba - Bb + X_3 - Cd + X_4 - Dd \geqq Y_3. \tag{10}$$

We model the engineering change orders problems using multiple variables linear programming. According to the functions, we can understand the engineering change orders relation between supply and requirement. Then, designer can estimate and perform ECO using spare cell efficiently.

5. Discussion

In this Section, we discuss the advantage and disadvantage of the related works. Table 1 shows ECO method comparison. The proposed approach designs a multiple variable linear programming ECO for VLSI design. Our method can predict cell resource accurately using multiple variable linear programming techniques. Traditional ECO is not to predict it well. Besides, our scheme provides a high accurate prediction of patching logic number to balance between spare cells and target cells. It is hard for traditional ECO method to do these. Moreover, we define restriction rule, resource optimization, and solution boundary of ECO problem to increase the efficiency of the proposed ECO method and provide a well solution.

6. Conclusion

In this paper, we proposed an engineering change orders design using multiple variables linear programming for VLSI design. The paper discusses typical ECO design flow, logic transformation, and multiple variables linear programming for VLSI design. The presented scheme estimates the resource of spare cells and provides a well solution of ECO problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Science Council of Taiwan under Grant nos. NSC 101-2221-E-027-135-MY2 and 102-2622-E-027-008-CC3. The authors gratefully acknowledge the Chip Implementation Center (CIC), for supplying the technology models used in IC design.

References

[1] J. A. Roy and I. L. Markov, "ECO-system: embracing the change in placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 12, pp. 2173–2185, 2007.

[2] C. Tan and I. H. Jiang, "Recent research development in metal-only ECO," in *Proceedings of the 54th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS '11)*, pp. 1–4, August 2011.

[3] Y. M. Kuo, Y. T. Chang, S. C. Chang, and M. Marek-Sadowska, "Spare cells with constant insertion for engineering change," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 3, pp. 456–460, 2009.

[4] D. Brand, A. Drumm, S. Kundu, and P. Narain, "Incremental synthesis," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 14–18, 1994.

[5] S. Huang, K. Chen, and K. Cheng, "AutoFix: A hybrid tool for automatic logic rectification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 9, pp. 1376–1384, 1999.

[6] C. Lin, K. Chen, and M. Marek-Sadowska, "Logic synthesis for engineering change," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 2-3, pp. 282–292, 1999.

[7] T. Shinsha, T. Kubo, Y. Sakataya, J. Koshishita, and K. Ishihara, "Incremental logic synthesis through gate logic structure identification," in *Proceedings of the IEEE/ACM Conference on Design Automation*, pp. 391–397, Jun 1986.

[8] G. Swamy, S. Rajamani, C. Lennard, and R. K. Brayton, "Minimal logic re-synthesis for engineering change," in *Proceedings of*

the IEEE International Symposium on Circuits and Systems, pp. 1596–1599, 1997.

[9] Y. Watanabe and R. K. Brayton, "Incremental synthesis for engineering changes," in *Proceedings of the IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '91)*, pp. 40–43, Cambridge, Mass, USA, October 1991.

[10] J. Wang, *Finding the Minimal Logic Difference for Functional ECO*, Taiwan Cadence Design Systems, 2012.

[11] Y. C. Fan and H. W. Tsao, "Watermarking for intellectual property protection," *IEE Electronics Letters*, vol. 39, no. 18, pp. 1316–1318, 2003.

[12] Y. Fan and H. Tsao, "Boundary scan test scheme for IP core identification via watermarking," *IEICE Transactions on Information and Systems*, vol. E88-D, no. 7, pp. 1397–1400, 2005.

[13] Y. Fan, "Testing-based watermarking techniques for intellectual -property identification in SOC design," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 3, pp. 467–479, 2008.

[14] Y. Fan and Y. Chiang, "Discrete wavelet transform on color picture interpolation of digital still camera," *VLSI Design*, vol. 2013, Article ID 738057, 9 pages, 2013.

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 531653, 10 pages
http://dx.doi.org/10.1155/2014/531653

Research Article

Design of Smart Power-Saving Architecture for Network on Chip

Trong-Yen Lee and Chi-Han Huang

Department of Electronic Engineering, National Taipei University of Technology, Taipei 10608, Taiwan

Correspondence should be addressed to Trong-Yen Lee; tylee@ntut.edu.tw

Received 5 June 2014; Accepted 27 June 2014; Published 6 August 2014

Academic Editor: Yu-Cheng Fan

In network-on-chip (NoC), the data transferring by virtual channels can avoid the issue of data loss and deadlock. Many virtual channels on one input or output port in router are included. However, the router includes five I/O ports, and then the power issue is very important in virtual channels. In this paper, a novel architecture, namely, Smart Power-Saving (SPS), for low power consumption and low area in virtual channels of NoC is proposed. The SPS architecture can accord different environmental factors to dynamically save power and optimization area in NoC. Comparison with related works, the new proposed method reduces 37.31%, 45.79%, and 19.26% on power consumption and reduces 49.4%, 25.5% and 14.4% on area, respectively.

1. Introduction

In recent years, the 3-dimensional IC and TSV (Through-Silicon Via) technology are proposed to solve area issues. The 3-dimensional IC of Intel Ivy Bridge processor and the 16-core multicore architecture can be implemented in 22 nm [1]. Therefore, the multicore and heterogeneous systems are popular research in SoC (system-on-chip). These architectures require high throughput and performance to transfer data in a multicore SoC. Therefore, the NoC (network-on-chip) can be proposed to solve this requirement, but it derived new problems such as power consumption and area [2, 3].

The NoC architecture [1] consists of processing element (PE), network interface (NI), router, and topology which is shown in Figure 1. The PEs transfer information to NI, the NI packages the information into flits then passes to routers. The routers have difference corner router (CR), edge router (ER), and router (R); the CR, ER and R has three, four, and five I/O ports to access information then each port includes n virtual channels. Router includes transmission channel, routing computation (RC), virtual channel arbiter (VA), switch arbiter (SA), and crossbar (XBAR). The flits includes header, body, and tail; the header flit has PE priority, source address, destination address, and so forth. The RC uses header flit and routing algorithms to find transmission path. VA uses two stages arbitration to select most high priority packet transmission and then will sign transmission channel. SA uses two stages arbitration and will select most body flits into XBAR to transmit. The VA will be working when the packet is arrival. The SA operation when the flit is arrival. The tail flit represents last flit, and then the router will unregister transmission channel. The router topology includes mesh, star, and fat tree [4, 5].

Yoon et al. [6] analysis of virtual channels (VCs) can avoid routing and protocol deadlock and improve the routing performance when the packet traffic is congested. The VCs can solve packet switch hard issue but it leads the power and area and so forth issue in NoC.

Nicopoulos et al. [2] proposed IntelliBuffer architecture to solve PV (process variation) to reduce the power consumption in layer 1 [7]. It differs from the conventional architecture in two fundamental ways. First, these slots use clock-gating to reduce the power consumption when slots are empty. In order to avoid data loss transmission, one of slots clock keeps to access data in each I/O port. Second, the router creates a leakage classification register (LCR) table; then the write and read pointer always accesses the lowest power consumption slots from the LCR table.

Taassori et al. [3] proposed an adaptive data compression technology to reduce the number of packet bits in layer 3 [7]. It reduces of the number of transmissions. Therefore, it can improve power consumption of router. Palma et al. [8] use T-Bus-Invert technology to reduce the hamming distance transition activity rate to improve the power consumption.

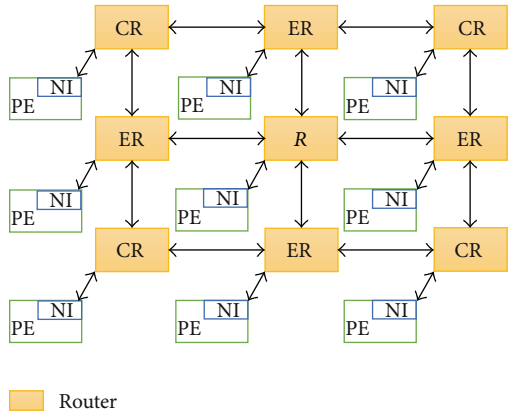


Figure 1: NoC architecture.

Jafarzadeh et al. [9] use end-to-end data coding technology to minimize switching activity rate and routing path to improve NI power consumption.

Lee et al. [10] proposed buffer clock-gating architecture and used clock-gating to reduce the transmit power consumption when slots are empty and full. Ezz-Eldin et al. [11] proposed an adaptive virtual channel with two sections in layer 1 [7]. First, the work used hierarchical multiplexing tree for Virtual Channels (VCs) to reduce area. Second, it uses clock-gating to reduce power consumption. Rosa et al. [12] proposed dynamic frequency scaling in PE for NoC. It considers the communication and loading rate to control the router frequency to reduce the power consumption.

Huaxi et al. [13] proposed fat tree-based optical NoC; this architecture includes topology, placement, layout, and protocol. This paper proposed low power and cost router optical turnaround router to improve the power consumption. Gu et al. [14] proposed Cygnus router to optimize the router algorithms to reduce the power consumption. Swaminathan et al. [15] create two FIFOs in NI. Use two FIFO dynamic configuration data access to improve throughput and power consumption.

In the next section we analyse the power consumption under the difference VCs access. Section 3 we introduce the topology and router packet architecture, we addition the SPS in router to save power. In Section 4 we present SPS with router design. Section 5 contains experimental results and Section 6 concludes this paper.

2. Power Issue with Virtual Channels

The multicore architecture and big data communication are more popular in next generation. Traditional communication technologies cannot meet a large amount of traffic on multicore and heterogeneous chip. The NoC can solve this issue. It uses network transmission method to make the difference core communication at same time. The NoC can solve the communication issue but the big data access enhances the power consumption.

The router composed of the arbitration and transmission unit [16] is illustrated in Figure 2. The arbitration unit selects the highest priority packet sent to next router. The arbitration unit includes routing computation (RC), VC arbiter (VA), and switch arbiter (SA). The RC is the calculation of routing paths and priorities. The VA contains a number of two-stage arbitrations to select packet and sign up VCs. First stage selects the local highest priority packet from input VCs to crossbar and signs up VCs. Second stage selects the global highest priority packet from input crossbar to output VCs and signs up VCs. The SA also contains a number of two-stage arbitrations to select flits for transmission. First stage selects the local highest priority flits from input VCs to crossbar. Second stage selects the global highest priority flits from input crossbar to output VCs. The VA executed prepacket and the SA executed preflits.

The router with transmission unit is illustrated in Figure 3. In this unit, it includes nVCs to access large packet from input physical channel to output physical channel. A power consumption calculation to VCs is shown in (1). The variable of n represents the number of access packets or flits in VCs. The variable of f represents access frequency in VCs. The variable of c represents capacitance and v represents voltage in VCs. Nicopoulos et al. [2] and Katabami et al. [17] proposed clock-gating to solve this issue.

In this paper, we proposed a dynamic control of each virtual channel clock in different transmission environments. Whether packet transfer is complete, the SPS can effectively reduce the power consumption and does not affect the transmission performance. Consider

$$P_{n\mathrm{VCs}} = \sum_{n=1}^{\infty} 1^n \times (f \times c \times v)^2. \quad (1)$$

3. Router and Topology with SPS

3.1. Relation of Topology and Router. The relation of topology and router is illustrated in Figure 4. The router uses different transmission mode with topologies. For example, the mesh uses the X-Y routing to transmit. The X-Y routing flow chart for 2×2 meshes is illustrated in Figure 5, when the MSB of destination router address (R_{dm}) is equal to the MSB of current router address (R_{cm}) and if the LSB of router

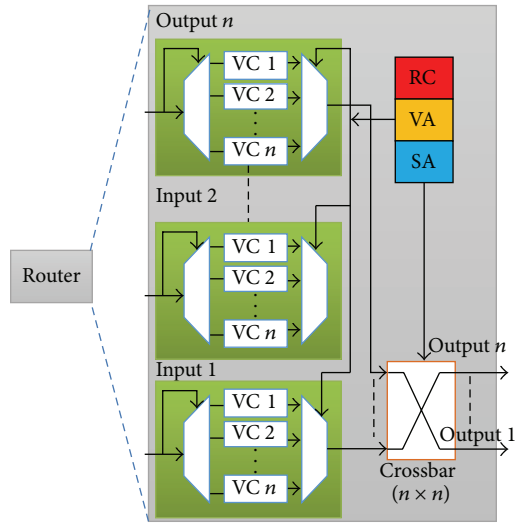


Figure 2: Router architecture with NoC.

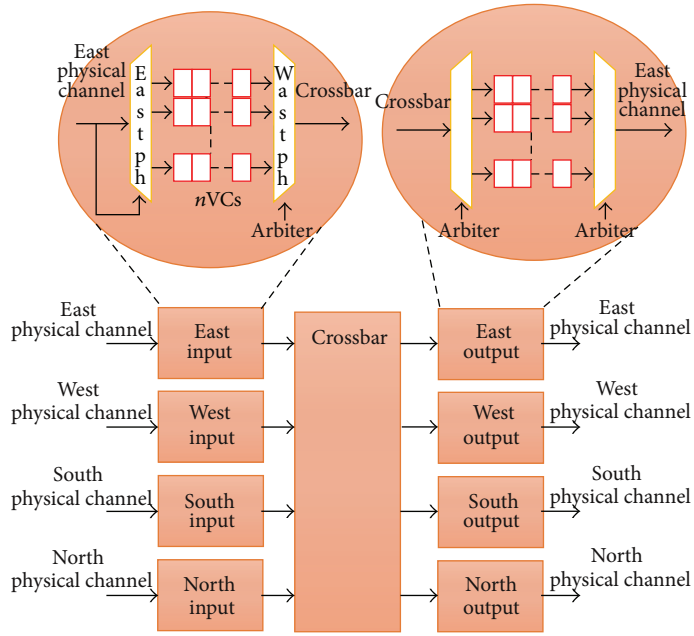


Figure 3: Transmission unit.

addresses (R_{dl} and R_{cl}) is equal then it means the flits arrival. Otherwise, the X-Y routing algorithm includes two-stage flows. In stage one, the flits are sent until that the R_{dm} equals of R_{cm} on the x-axis routers. In stage two, the flits are sent to the destination by y-axis routers. The virtual channel will be initialed under packet transmit on two routers, which procedure is shown on Algorithm 1.

The control method of arbiter architecture uses different transmission mode to design. The VC arbiter and switch bar are by the topology and priority to design the routing computation unit. Algorithm 2 constructs VC two stages arbitration of prepackets. Stage 1 decided high priority packet into crossbar from local VCs (input VCs) of each packet at lines 3 to 4 and lines 8 to 10. Stage 2 decided most important packet to transmission from global VCs (output VCs) of each packet at lines 5 to 6 and lines 11 to 13.

Sign up Algorithm
Input: R_{roth} and E_{mp}.
(1) while (flits arrival) do
(2) if (R_{rothf2} is header and adx is free channel)
(3) {sign up the channel and select the channel to output}
(4) else if (R_{rothf2} is body and $adx = R_{roths2}$)
(5) {select the channel to output}
(6) else if (R_{rothf2} is tail and $adx = R_{roths2}$)
(7) {clear the channel and select the channel to output;}
(8) else
(9) {read back flit to virtual channel}
(10) end while

Algorithm 1: Channel sign up algorithm.

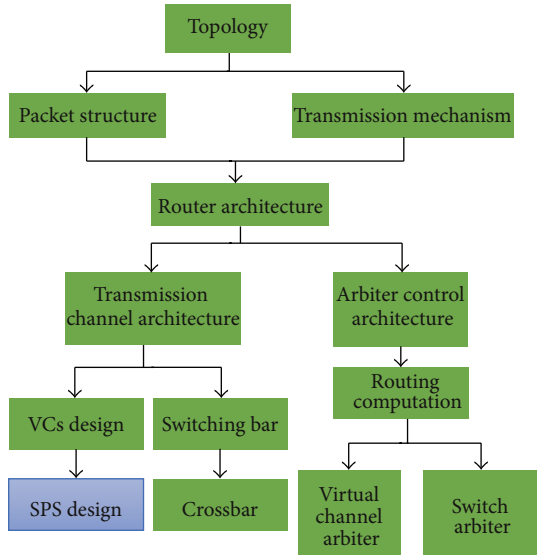


Figure 4: Topology and router relation with SPS.

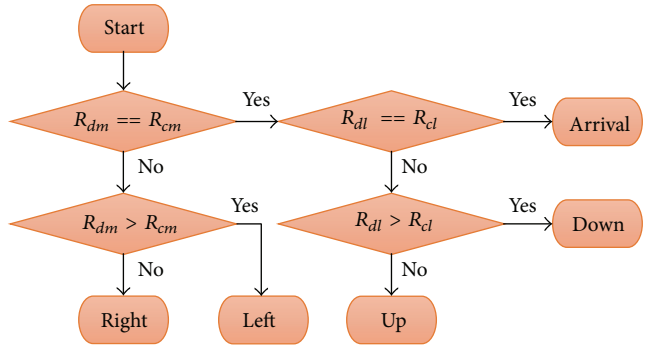


Figure 5: X-Y routing flow chart.

Virtual channel arbitration
Input: header flits
/∗Control signal enable∗/
(1) while (header flits) do
(2) use lottery arbitration to select local and global highest priority flits
(3) if (local)
(4) {V_{ai} = local input virtual channel address}
(5) if (global)
(6) {V_{ao} = global input virtual channel address}
(7) end while
/∗Channel switch∗/
(8) Case V_{ai}
(9) {C_{ri1} = local packet of V_{ai}}
(10) end case
(11) Case V_{ao}
(12) {R_{it} = global packet of V_{ao}}
(13) end case

Algorithm 2: VC arbitration algorithm.

Switch arbitration
Input: body and tail flits
/∗Control signal enable∗/
(1) while (body or tail flits) do
(2) use channel sign up register to select local and global highest priority flits
(3) if (local)
(4) {S_{ai} = local input virtual channel address}
(5) if (global)
(6) {S_{ao} = global input virtual channel address}
(7) end while
/∗Channel switch∗/
(8) Case S_{ai}
(9) {C_{ri2} = local packet of S_{ai}}
(10) end case
(11) Case S_{ao}
(12) {R_{ot} = global packet of S_{ao}}
(13) end case

Algorithm 3: Switch arbitration algorithm.

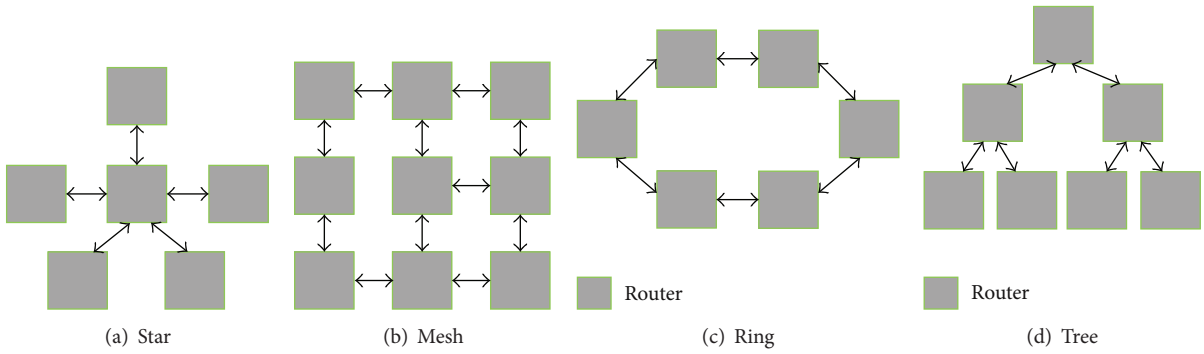


Figure 6: Router connection topology architecture.

Algorithm 3 constructs VC two stages arbitration of preflits. Stage 1 decided high priority flit into crossbar from local VCs (input VCs) of each flit at lines 3 to 4 and lines 8 to 10. Stage 2 decided most important flit to transmit from global VCs (output VCs) of each flits at lines 5 to 6 and lines 11 to 13.

The router includes four directions to connect other routers and one local physical channel to connect PE in transmission channel architecture. There have been nVCs of each physical channel without local physical channel. The switch bar support for transmission the most important packet to output channel. The SPS controls each VCs power consumption when the channel status changes. The SPS architecture is introduced in next section.

3.2. Topology Architecture. The topology is definition of the packet transmission path between router and link. The router connection topology architecture is shown in Figure 6; they include star, mesh, ring, and tree topologies. The RC algorithms depend on topology architecture in arbitration unit. The VA and SA algorithms depend on packet priority in arbitration unit. In this paper, the topology is the 2×2 mesh, the RC algorithm is X-Y routing, and the VA and SA algorithms are lottery [18].

The router that connects with PE is shown in Figure 7; so that the PE and router access information, use the network interface (NI). It handles the information between router and PE. The NI includes two level designs [19] as shown in Figure 8. It contains three modules to meet the specifications of the different layers. The shell module needs to meet IP specification. The kernel module needs to meet the NoC topology specification.

3.3. Flits with Router Architecture. The flit specification with router is shown in Figure 9; the flit type of 2-bit 00 represents the one packet; this flit type does not sign up VCs. The 2-bit 01 represents the *header* flit which includes routing information and address; this flit type always is determined in sign up channel. The 2-bit 10 represents the *body* flit which includes transmission information; this flit payload records the segment packet. The 2-bit 11 represent the *tail* as last transmission information; this flit not only records the last segment packet but also cleans the VCs.

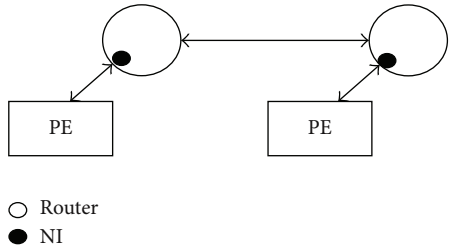


Figure 7: Router connection with PE.

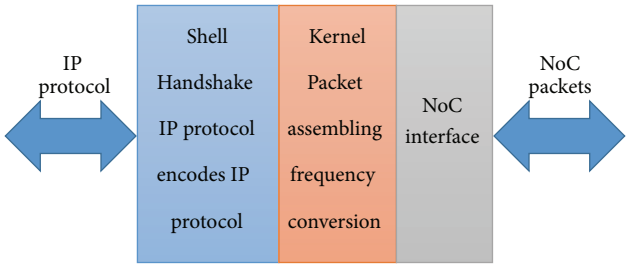


Figure 8: NI breakdown into Shell, Kernel, and interface.

Flit type (00)	Source address	Destinations address	Payload
Flit type (01)	Source address	Destinations address	Routing information
Flit type (10)	Payload		
Flit type (11)	Payload		

Figure 9: Flits type of router.

4. SPS with Router Design

The VC that contains many slots to access data led to extra power consumption. In this paper, we propose SPS architecture to reduce the power consumption.

4.1. Router with SPS Architecture. The proposed router with SPS architecture is illustrated in Figure 10. The physical channel (PC) is used to connect other routers and access information. The input VCs (IVC) is used to store information from PCs. It always is designed by FIFO or other sequential logic. The arbiter decides the flits priority to control input switch logic (ISL) and output switch logic (OSL) to transmit flits. It includes RC, VA, and SA. The crossbar (CR) connects IVC to OVC, the switch signal form arbiter. The output VCs (OVC) store information from CR. The proposed SPS uses the transmission channel status to dynamic control IVC and OVC clock in essential operating.

The VCs with SPS architecture are illustrated in Figure 11. It controls system clock into I/O VC to reduce power consumption. In this architecture, the VC contains 0 to $i-1$ slots to access data.

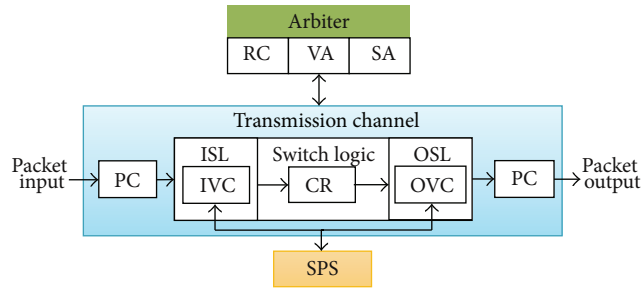


Figure 10: Router with SPS architecture.

4.2. Design of SPS Control Timimg. The VCs access timing diagrams of SPS architecture are illustrated in Figure 12. The Clock Block A indicates that the VCs have no information to transmit. The Clock Block B indicates that the VCs are writing information. The Clock Block C indicates that the data in VCs are waiting to transmit. Our analysis for unused clock-gating architecture is shown in (2). The slots access information of power consumption is denoted by P_a. The slot content full and empty of power consumption are denoted by P_f and P_e, respectively. The P_s is power consumption except for P_f, P_e, and P_a. The unused clock-gating architecture does not control clock for sequential logic in nVCs. Therefore, the logic will generate power consumption in high transmission structure.

The clocking gating consumes power in Clock Block B and Clock Block C. Our analysis for clock-gating architecture is shown in (3). The P_{g1} is power consumption of empty gating. The clock-gating architecture does not control clock when VCs is full stage. The VCs always store flits to wait for transmission.

The SPS consumes power in Clock Block B. Our analysis for SPS architecture is shown in (4). The P_{g2} is power consumption of SPS. It saves the power consumption of empty and full gating for nVCs. Consider

$$P_{r1} = P_a + P_f + P_e + P_s, \tag{2}$$

$$P_{r2} = P_a + P_f + P_s + P_{g1}, \tag{3}$$

$$P_{r3} = P_a + P_s + P_{g2}. \tag{4}$$

4.3. Design of SPS. The proposed SPS uses the VCs status to dynamic control clock of each VC. The CFSM of SPS with VCs is illustrated in Figure 13; it contains two CFSM in this architecture.

The first CFSM includes initial, empty, full, and waiting status. *Initial status*: when the VC is reset, the structure is into the initial status until the flit arrive. *Empty status*: when the user resets the VCs or the flits transport to next storage unit, the structure is into this status. *Full status*: the store flit in VC is full. *Waiting status*: When the user resest the VCs or the store flit is complete.

The VCs with SPS algorithm is illustrated in Algorithm 4. In line 3, the VCs will initialize the VCs count and flags. The VCs will access flits to change VCs count when channel packet or arbiter signal arrive at line 4 to 9. When the VCs

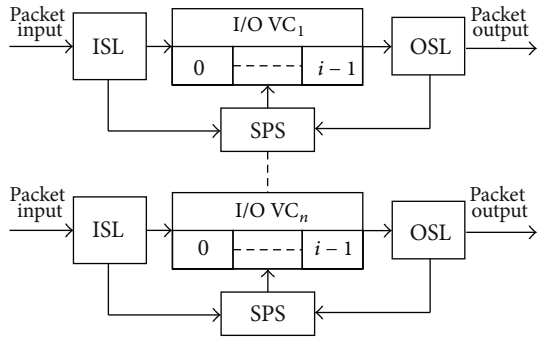


Figure 11: VCs with SPS architecture.

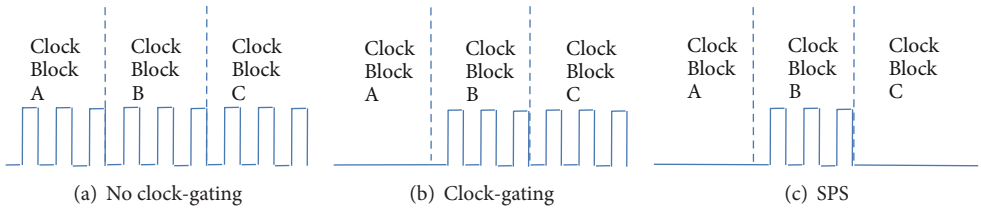


Figure 12: VCs power with clock diagram.

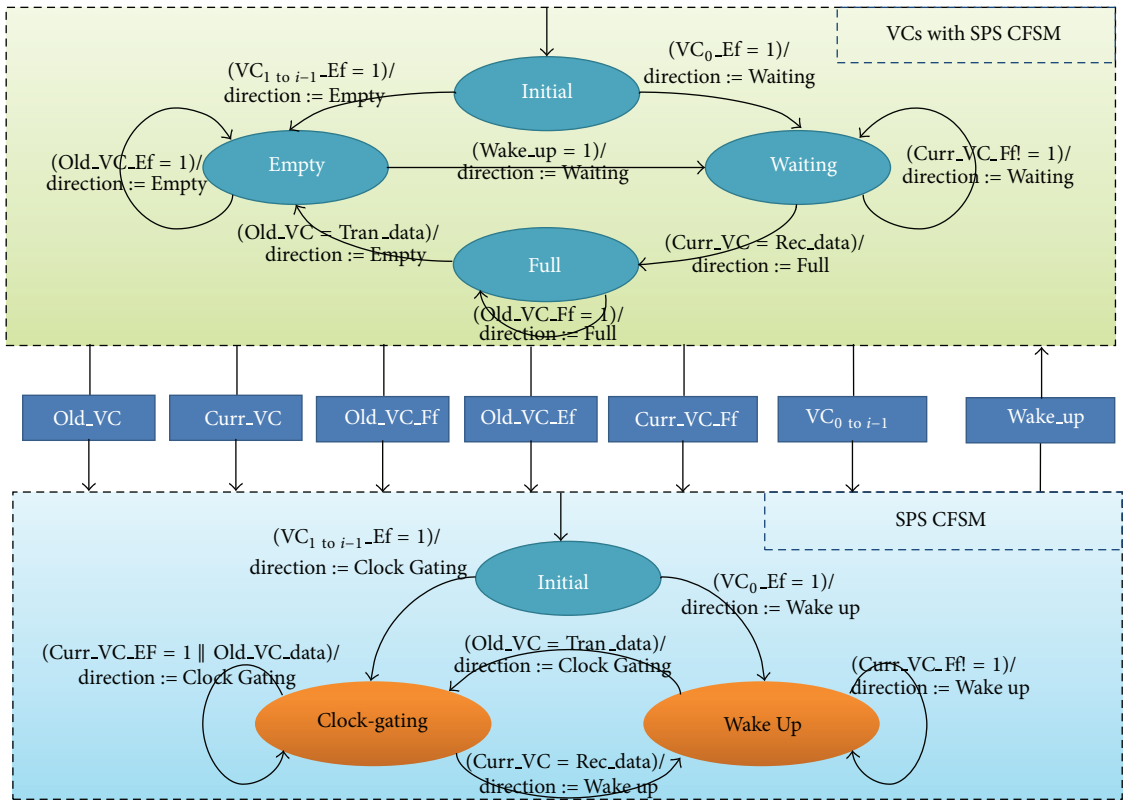


Figure 13: CFSM of SPS with VCs.

count can be changed, then the VCs flag will be changed at line 10 to 17.

The second CFSM includes initial, clock-gating, and wake up status. *Initial status*: this principle is the first CFSM of initial state. *Clock-gating*: when the VC changes to full or empty, then SPS will disable this VC clock and change to this status. *Wake up*: when the VC want to store flit, one VC will wake up.

The SPS algorithm is illustrated in Algorithm 5. In line 3, the SPS will initialize VCs clock and access status from VCs

VCs with SPS Algorithm
Input: VCs clock, channel packet, arbiter signal and reset.
Output: channel packet, channel status
(1) VC_{count} is integer and range is $1 \le VC_{count} \le n$
(2) VC_{flag} includes full flag and empty flag
(3) initial VC_{count} and VC_{flag}
(4) while (channel packet or arbiter signal be arrival) do
(5) if (channel packet be arrival and full flag != 1)
(6) {$VC_{count} = VC_{count} + 1$ and packet store in VCs}
(7) if (arbiter signal be arrival and empty flag != 1)
(8) {$VC_{count} = VC_{count} - 1$ and packet be read from VCs}
(9) end while
(10) while (VC_{count} be change) do
(11) if ($VC_{count} = n$)
(12) {assign full flag to 1}
(13) else if ($VC_{count} = 1$)
(14) {assign empty flag to 1}
(15) else
(16) {assign full flag and empty flag to 0}
(17) end while

Algorithm 4: VCs with SPS algorithm.

SPS Algorithm
Input: system clock, channel packet, arbiter signal and reset.
Output: VCs clock
(1) VC_{group} is VCs group of 4 direction port
(2) VC_{flag} includes full flag and empty flag
(3) Initial VCs clock and access VCs count and stage flag
(4) follow LCR to arrangement all slots priority;
(5) VC_{clki} is VCs clock of each VC_{group} //where $1 \le i \le n$
(6) Example VC_{group} = East port
(7) initial $VC_{clki} = 0$; //where $1 \le i \le n$
(8) while (virtual channel be write) do
(9) if (VC_{flag} = empty)
(10) {VC_{clki} = system clock}
(11) If (VC_{flag} = full flag)
(12) {$VC_{clki} = 0$ and VC_{clki+1} = system clock}
(13) end while
(14) while (virtual channel be read) do
(15) if (empty flag = 1)
(16) {$VC_{clki} = 0$}
(17) end while

Algorithm 5: SPS algorithm.

with VC flags. The slots priority from LCR [2] and each VCs clock can be initialized at lines 4, 5, and 7. The SPS controls VCs clock to reduce the VCs power consumption when VCs is accessed and flags changed at lines 8 to 17.

5. Experimental Results

In this section, we proposed autotesting architect for router with SPS. This architect includes four modules of autotesting. The first module is test-vector generator (TVG); the FSM is illustrated in Figure 14; the Idle status is waiting for the requirement of start testing, when the requirement arrives, TVG then will change status from idle to generator. When the requirement is cancelled, the status be changed from generator to idle. The generator status will generate test-vector and compare-vector; this is illustrated in Figure 15; we use *c* language to generate lottery arbitration [18] in test-vector at control step 1. We use HDL to design the conventional router to generate the compare-vector and the input pattern from the test-vector at control step 2. When the compare-vector and test-vector functions are complete then the status will be changed from generator to vector output (VO) at control step 3. The VO status will transform test-vector and compare-vector to Xilinx memory IP files, through memory to control data output to test and compare only one clock.

Router with SPS Algorithm
Input: system clock, start, Lottery Input.
Output: test-start, Implement-results
(1) If start testing
(2) {test-start = 1; pass VD}
(3) While (read test data from and start bit set-up to one) do
(4) Lottery Input = Test-vector
(5) Implement-results = Test-vector use Router with SPS to transmission;
(6) Test-vector address = Test-vector address + 1;
(7) If (test finish or start = 0)
(8) {test-start = 0}
(9) End while

Algorithm 6: Router with SPS testing algorithm.

The second module is vector database (VD); the control flow graph is illustrated in Figure 16; the module writes VO status vector in memory. The database includes two vectors to test and analyze the proposed circuit. The lottery database is provided test packet for router with SPS. The compare database is provided analysis for router with SPS.

The third module is router with SPS; we use VD to propose the test-vector to implement this module. The testing algorithm is illustrated in Algorithm 6, when the start signal set up to one from I/O, then the module starts to test and pass this signal to VD at lines 1 to 2. When testing is started, the input signal will be read from VD, shown at lines 3 and 4 in Algorithm 6. The read test-vector delay time is one clock from VD to router with SPS. The router with SPS uses VD test-vector to compute at line 6. When this pattern computation is finish, the next pattern will be read from VD at line 6. When the test pattern computation is finished or start signal is cancelled, test-start set up and stop testing at lines 7 and 8.

The final module, verification module, is illustrated in Figure 17; we verify the function in this module. The function verification is comparing of compare-vector and implement-results from VD and router with SPS. If the pattern is error, then verification result returns error signal.

The hardware experimental environment uses Xilinx FPGA xc5vlx50t-1ff1136 to verify SPS architecture. The software experimental environment uses Xilinx ISE 12.3 and the

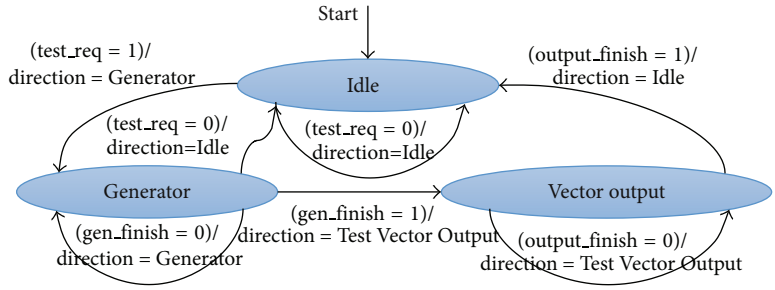


Figure 14: Test-vector generator (TVG) module FSM.

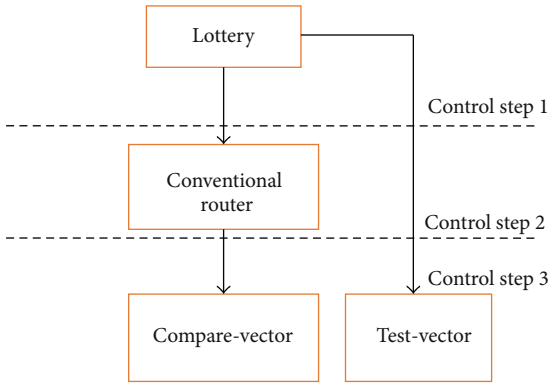


Figure 15: Generator status control and data flow graphs.

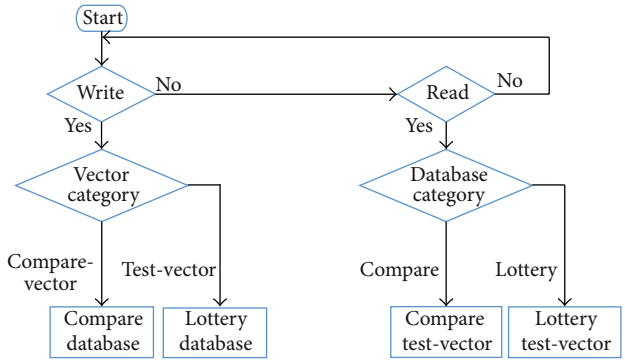


Figure 16: Vector database (VD) control flow graph.

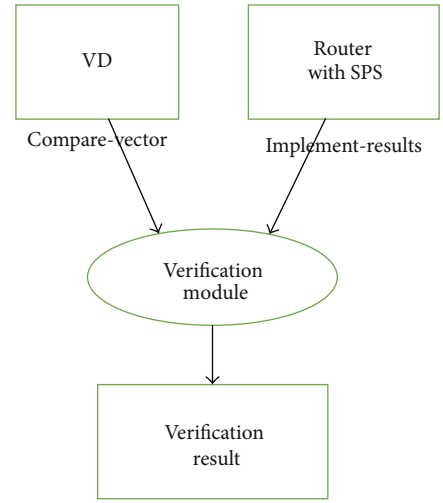


Figure 17: Verification module data flow graphs.

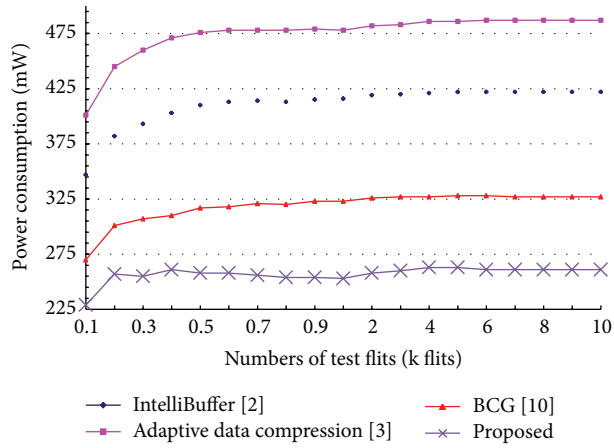


Figure 18: Power consumption distribution.

analysis tools use Modelsim 6.6, Xilinx Chipscope ILA, and Xpower 12.3, which are supported by Xilinx. The test experimental environment uses 2×2 mesh and X-Y routing; the PC have 4 VCs to access flits. The power consumption distribution is illustrated in Figure 18; the number of test packets is from 100 to 10000. The packet format is flit and packet length is 18 bits.

Comparing related works, as shown in Table 1, IntelliBuffer [2], adaptive data compression [3], and buffer clock-gating [10], the proposed method reduces 37.31%, 45.79%, and 19.26% on power consumption, respectively, and reduces 49.4%, 25.5% and 14.4% on area, respectively.

6. Conclusions

The *Smart Power-Saving (SPS)* architecture for network-on-chip was presented. A clock control circuit and SPS algorithm are demonstrated to reduce the power consumption on the NoC architecture. From experimental results, the proposed

Table 1: Comparison of power consumption and area.

Methods	Constraints			
	Power consumption (mW)	Area (number of slices)	Improved power	Improved area
IntelliBuffer [2]	410.42	1551	37.31%	49.4%
Adaptive data compression [3]	474.53	1054	45.79%	25.5%
Buffer clock-gating [10]	318.63	917	19.26%	14.4%
Newly proposed	257.05	785		

SPS architecture is more efficient to reduce the power consumption than IntelliBuffer [1], adaptive data compression [3], and buffer clock-gating [10] in the NoC architecture.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the Ministry of Science and Technology of the Republic of China, Taiwan, for partially supporting this research.

References

[1] D. James, "Intel Ivy Bridge unveiled—the first commercial tri-gate, high-k, metal-gate CPU," in *Proceedings of the Custom Integrated Circuits Conference (CICC '12)*, pp. 9–12, September 2012.

[2] C. Nicopoulos, S. Srinivasan, A. Yanamandra et al., "On the effects of process variation in network-on-chip architectures," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 3, pp. 240–254, 2010.

[3] M. Taassori, M. Taassori, and M. Mossavi, "Adaptive data compression in NoC architectures for power optimization," *International Review on Computers and Software*, vol. 5, no. 5, pp. 540–547, 2010.

[4] D. Bertozzi and L. Benini, "Xpipes: a network-on-chip architecture for gigascale systems-on-chip," *IEEE Circuits and Systems Magazine*, vol. 4, no. 2, pp. 18–31, 2004.

[5] S. J. Lee, K. Lee, and H. J. Yoo, "Analysis and implementation of practical, cost-effective networks on chips," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 422–433, 2005.

[6] Y. J. Yoon, N. Concer, M. Petracca, and L. Carloni, "Virtual channels versus multiple physical networks: a comparative analysis," in *Proceedings of the 47th ACM/IEEE Design Automation Conference (DAC '10)*, pp. 162–165, June 2010.

[7] L. Benini and G. de Micheli, "Networks on chips: a new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, pp. 70–78, 2002.

[8] J. C. S. Palma, L. S. Indrusiak, F. G. Moraes, R. Reis, and M. Glesner, "Reducing the power consumption in networks-on-chip through data coding schemes," in *Proceedings of the 14th IEEE International Conference on Electronics, Circuits and Systems (ICECS '07)*, pp. 1007–1010, December 2007.

[9] N. Jafarzadeh, M. Palesi, A. Khademzadeh, and A. Afzali-Kusha, "Data Encoding Techniques for Reducing Energy Consumption in Network-on-Chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 675–685, 2014.

[10] T. Y. Lee, C. H. Huang, and X. S. Lin, "Design of buffer clock-gating architecture for network-on-chip," in *Proceedings of the 22th VLSI Design/CAD Symposium*, pp. 2–5, August 2011.

[11] R. Ezz-Eldin, M. A. El-Moursy, and A. M. Refaat, "Low leakage power NoC switch using AVC," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '12)*, pp. 2549–2552, Seoul, Republic of Korea, May 2012.

[12] T. R. da Rosa, V. Larrea, N. Calazans, and F. G. Moraes, "Power consumption reduction in MPSoCs through DFS," in *Proceedings of the 25th Symposium on Integrated Circuits and Systems Design (SBCCI '12)*, pp. 1–6, 2012.

[13] G. Huaxi, X. Jiang, and Z. Wei, "A low-power fat tree-based optical network-on-chip for multiprocessor system-on-chip," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE '09)*, pp. 3–8, April 2009.

[14] H. Gu, K. H. Mo, J. Xu, and W. Zhang, "A low-power low-cost optical router for optical networks-on-chip in multiprocessor systems-on-chip," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI '09)*, pp. 19–24, Tampa, Fla, USA, May 2009.

[15] K. Swaminathan, G. Lakshminarayanan, F. Lang, M. Fahmi, and S. B. Ko, "Design of a low power network interface for Network on chip," in *Proceedings of the 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE '13)*, pp. 1–4, May 2013.

[16] R. Mullins, A. West, and S. Moore, "Low-latency virtual-channel routers for on-chip networks," in *Proceedings of the 31st Annual International Symposium on Computer Architecture (ISCA '04)*, pp. 188–197, 2004.

[17] H. Katabami, H. Saito, and T. Yoneda, "Design of a GALS-NoC using soft-cores on FPGAs," in *Proceeding of the Embedded Multicore Socs (MCSoC '13)*, pp. 26–28, September 2013.

[18] J. Wang, Y. Li, Q. Peng, and T. Tan, "A dynamic priority arbiter for network-on-chip," in *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES '09)*, pp. 253–256, July 2009.

[19] S. Saponara, L. Fanucci, and M. Coppola, "Design and coverage-driven verification of a novel network-interface IP macrocell for network-on-chip interconnects," *Journal of Microprocessors and Microsystems*, vol. 35, no. 6, pp. 579–592, 2011.

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 406416, 7 pages
http://dx.doi.org/10.1155/2014/406416

Research Article

Optimization of Fractional-N-PLL Frequency Synthesizer for Power Effective Design

Sahar Arshad,[1] Muhammad Ismail,[1] Usman Ahmad,[2] Anees ul Husnain,[3] and Qaiser Ijaz[3]

[1] *Department of Electronic Engineering, University College of Engineering and Technology, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan*
[2] *Scholar Teacher Research Alliance for Problem Solving (STRAPS), Bahawalpur 63100, Pakistan*
[3] *Department of Computer System Engineering, University College of Engineering and Technology, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan*

Correspondence should be addressed to Qaiser Ijaz; qaiser.ijaz@iub.edu.pk

Received 10 May 2014; Accepted 7 June 2014; Published 23 July 2014

Academic Editor: Yu-Cheng Fan

We are going to design and simulate low power fractional-N phase-locked loop (FNPLL) frequency synthesizer for industrial application, which is based on VLSI. The design of FNPLL has been optimized using different VLSI techniques to acquire significant performance in terms of speed with relatively less power consumption. One of the major contributions in optimization is contributed by the loop filter as it limits the switching time between cycles. Sigma-delta modulator attenuates the noise generated by the loop filter. This paper presents the implementation details and simulation results of all the blocks of optimized design.

1. Introduction

For many manufacturers and product developers, it is a good idea to reduce power consumption in electronic products. It is also an important idea to gain competitive advantage in an increasingly power hungry world. Low power consumption gives many benefits to designers and to users; for example, the main advantage is that it reduces stringent cooling requirements and it results in inexpensive and more compact products [1]. The rapid rise in power requirements has promoted governments and industry to increase energy efficiency and design low power components. The majority of frequency synthesis techniques fall into two categories: either direct frequency synthesis or indirect frequency synthesis [2]. To achieve fine frequency steps, the direct frequency synthesis technique is used because it is based on using digital techniques. To generate multiples (integer or noninteger) of a reference frequency, indirect frequency synthesis is used because it is based on a phase-locked loop (PLL). Here, the latter technique is used because we are going to implement PLL. It is used to generate a signal whose phase is related to the phase of the input signal and this signal is called an output signal of the PLL. The input signal is called the "reference" signal. In a feedback loop, the oscillator is controlled by the output signal from the phase detector [3, 4]. The circuit compares the phase of a signal obtained from its output oscillator with the phase of the input signal to keep the phases matched by adjusting the frequency of its oscillator. A phase locked loop (PLL) architecture has two types, a Fractional-N PLL (FNPLL) and an integer-N PLL [5]. For a given frequency resolution, the latter has high reference frequency than the former, and, hence, the loop bandwidth which is limited to 10% of the reference frequency can be set larger in the FNPLL than in the integer-N-PLL. Therefore, the latter architecture is used for faster locking. This speed advantage of the FNPLL, however, comes at the price of increased design complexity [6]. This is because the fractional-N operation in steady state requires fractional spur reduction circuits whose quantization noise folds into the PLL spectrum via loop nonlinearities, demanding more significant design efforts to minimize the loop nonlinearities.

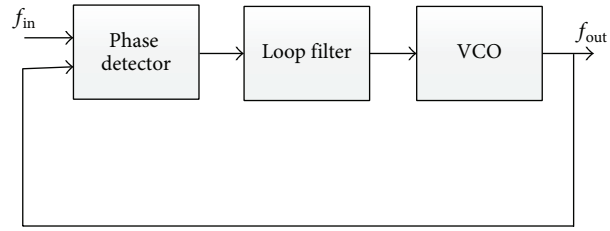


Figure 1: PLL system representation.

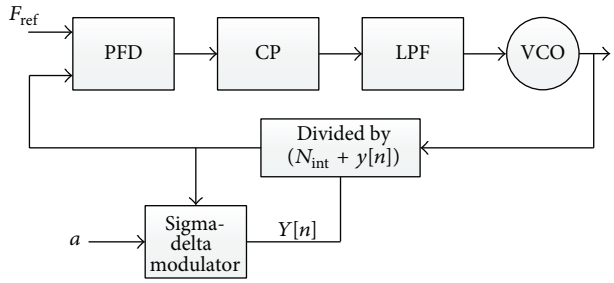


Figure 2: Sigma-delta FNPLL arrangement.

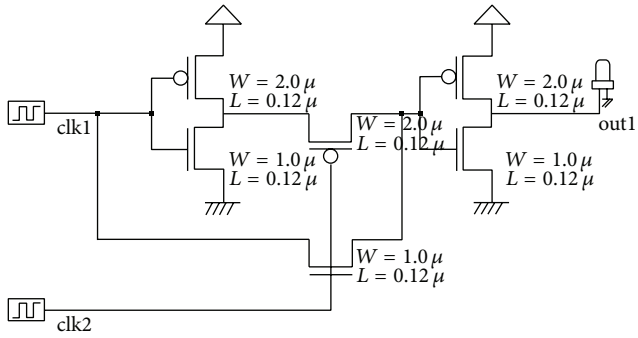


Figure 3: Schematic of phase detector (PD).

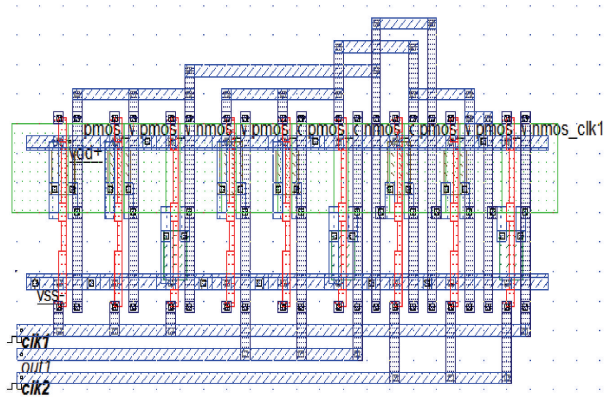


Figure 4: Layout of PD.

On the contrary, in the absence of fractional spurs, integer-N-PLLs involve less design complexity. Here, FNPLL is required. The expression of output frequency of the FNPLL is

$$\text{Freq}_{\text{FNPLL}} = (M \cdot n) * \text{Freq}_{\text{Ref}}. \quad (1)$$

In this equation, M is an integer, and n is the fractional part. To obtain the desired fractional division ratio dual modulus is used [7]. Using the sigma-delta modulation technique, we can remove the fractional spurs. This technique generates a random integer number. The average of these random numbers will result in the desired ratio. A phase detector,

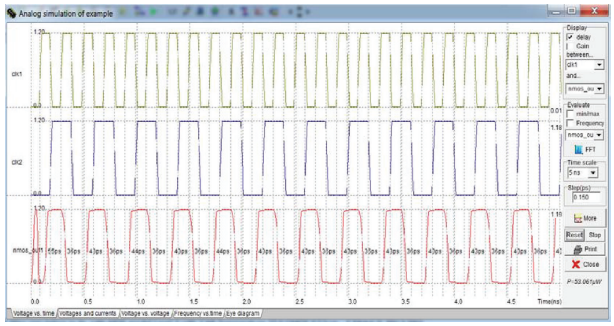


Figure 5: V versus T.

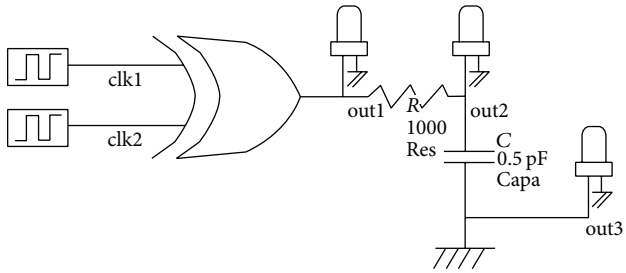


Figure 6: CMOS circuit of PD with LF.

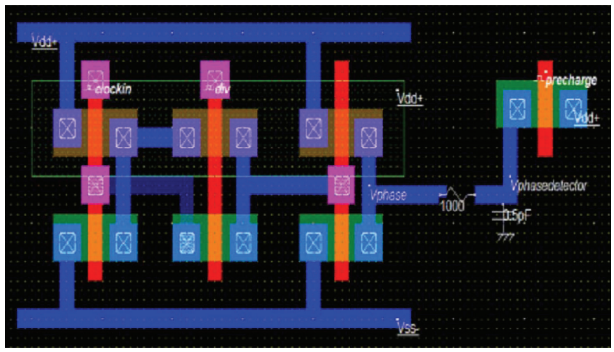


Figure 7: Layout of PD with LF.

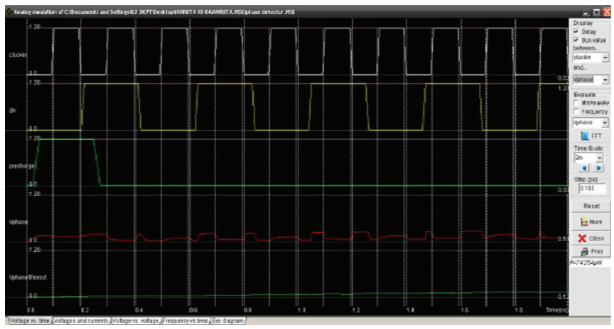


Figure 8: V versus T.

a loop filter, and a voltage controlled oscillator (VCO) are the main parts of phase-locked loop, as shown in Figure 1.

The important part of the phase-locked loop (PLL) is phase detector. It is also called a phase comparator, logic circuit, frequency mixer, or an analog multiplier that generates a voltage signal and this voltage signal shows the phase difference. Three units are coupled as a feedback system as shown in Figure 1. The periodic output signal is generated by the oscillator. The applications of PLL are versatile; for example, it can generate different stable frequencies or it can obtain a signal from noisy signals. A complete phase-locked loop block can be obtained from single integrated circuit.

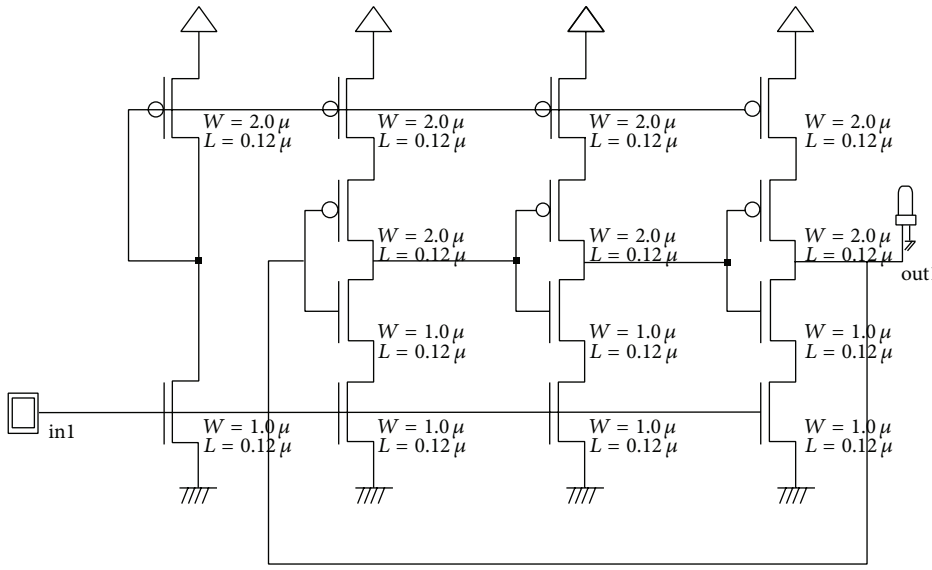


Figure 9: VCO schematic.

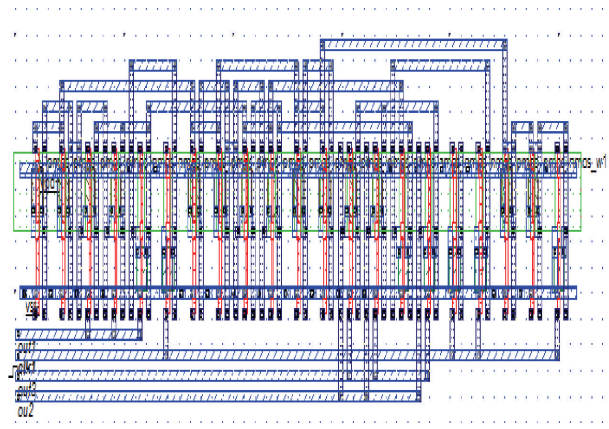


Figure 10: VCO layout.

This technique is used in advanced electronic products which have different output frequencies from some Hz to many Giga Hz [8]. To get low power consumption, high speed, and stability, we decide to design phase-locked loop of architecture fractional-n using 0.12 micrometer CMOS/VLSI design. As the demand of PLL is growing day by day in the field of communications, low leakage transistors will be used for maintaining low power but for this we have to make a little compromise on frequency.

The structure of FNPLL is depicted in Figure 2. We can control characteristics of PLL by using low pass filter, for example, transients response and bandwidth. The basic and essential functional unit of PLL is VCO. VCO is used for clock generation [9]. For synthesizing aspired frequencies, we use PLL with arbitrary frequency division (+N) method. This proposed technique has the ability to give fast settling time, reduce phase noise, and also reduce the effect of spurious frequencies when compared with existing FNPLL techniques.

2. PLL Design Using 0.12 Micrometer

2.1. Phase Detector. The first block has two inputs, the reference input and the feedback. It compares frequencies of input and produces an output using phase difference of inputs. To represent this block XOR gates are used. The gate produces a square wave when one-fourth of period shift of 90 degrees takes place at clock input, whereas output is different for all other angles. We apply output of the XOR gates to LPF which results in analog voltage, proportional to phase difference.

Figure 3 depicts a CMOS circuit of phase detector, Figure 4 describes layout, and Figure 5 represents the output waveform.

2.2. Loop Filter. To get pure DC voltage along with rectifiers filters, the electronic circuits are also used. The second block of PLL is loop filter and it has two distinct functions. First, maintains stability, that is defined by describing the

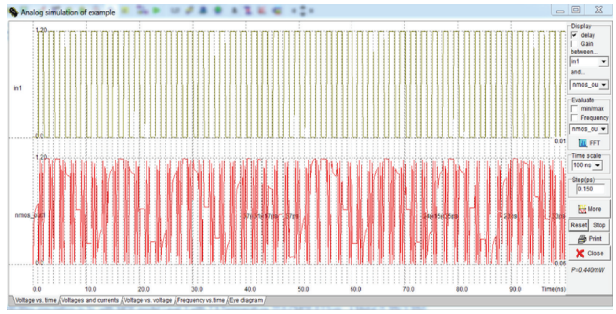


Figure 11: V versus T.

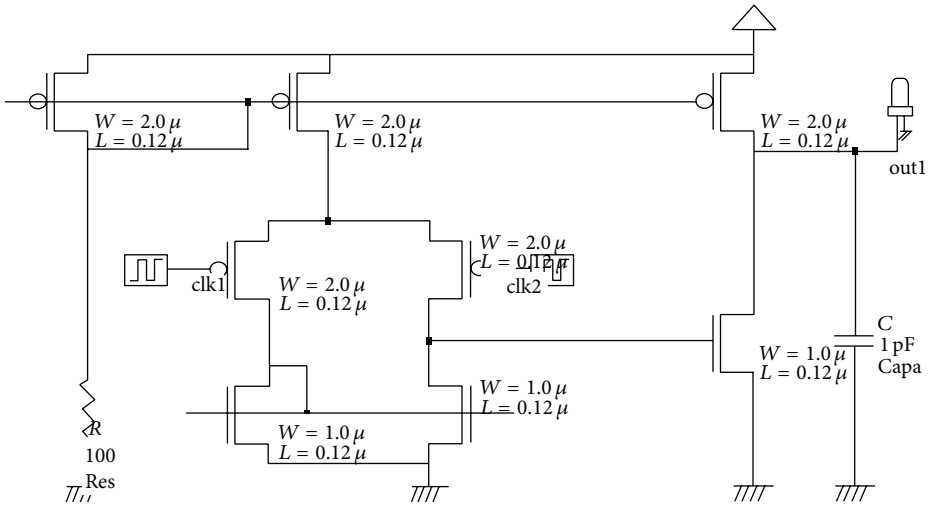


Figure 12: CMOS circuit of comparator.

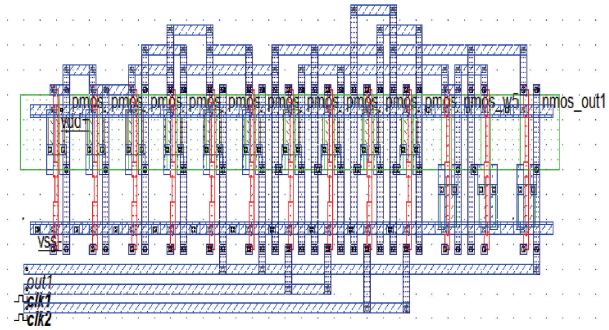


Figure 13: Layout of comparator.

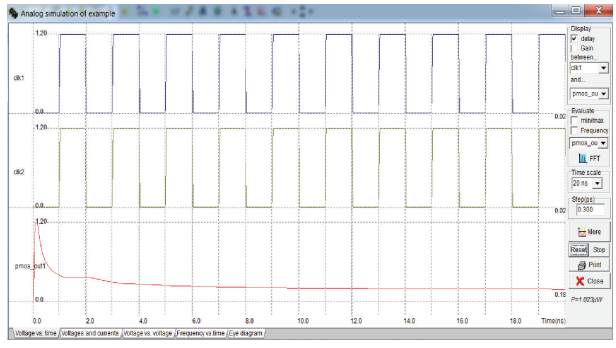


Figure 14: V versus T.

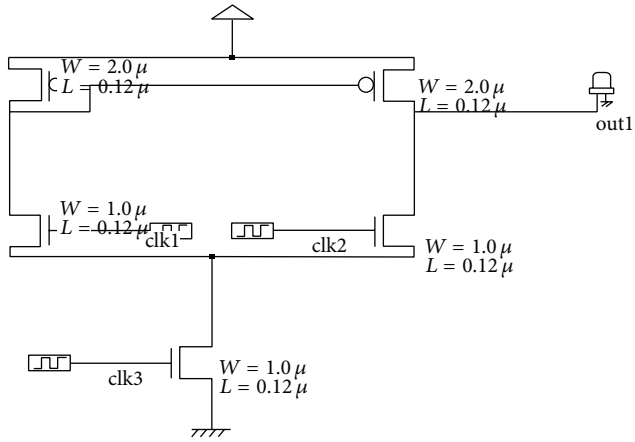


Figure 15: CMOS schematic of OTA.

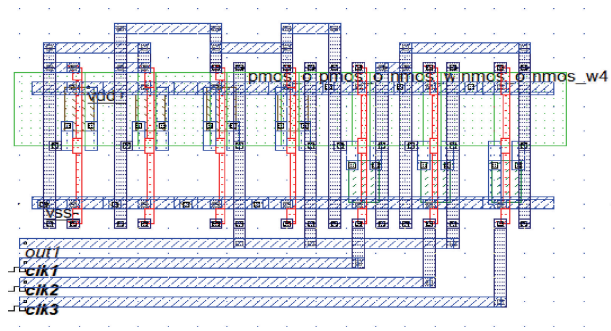


Figure 16: OTA layout.

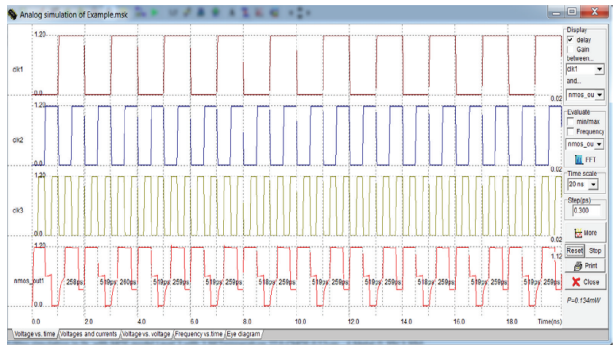


Figure 17: V versus T.

loop dynamics. This explains the response of the loop to uncertainties. The 2nd function is applied to the VCO control input which appears at the phase detector output. This frequency produces FM sidebands and modulates the VCO [10]. Other features of the PLL, for example, bandwidth, transient response, lock range, and capture range, can be controlled by LPF. The LPF is used to attenuate this energy, but it can also reject band. The low pass filter can be obtained by using a capacitor of large value and the capacitor is charged and discharged with the help of the switch resistance R_{on}. By the help of R_{on}.C delay a low pass filter can be created. Figure 6 depicts a CMOS schematic of phase detector with loop filter, Figure 7 shows layout, and Figure 8 shows output waveform.

2.3. Voltage Controlled Oscillator. As VCO is a source of varying output signal so the frequency of the output signal is regulated over a DC voltage range. The output signal can be a square wave or a triangular wave form. The oscillation frequency is controlled by the value of input voltage [11]. Figure 9 shows a CMOS circuit of VCO, Figure 10 shows layout, and Figure 11 shows output waveform.

2.4. Sigma-Delta Modulator. Sigma-delta modulation technique is used to convert high definition signals to low definition signals in digital domain. We designed sigma-delta modulator using 0.12 micrometer feature size and then the layout was obtained. The input is the aspired fractional number (n) and the output is the sum of quantization noise

and a DC part [12, 13]. By the use of integer divider quantization noise was generated. Figures 12 and 15 show the CMOS circuit; Figures 13 and 16 show the layout of comparator and operational transconductance amplifier. Figures 14 and 17 show the output waveforms.

3. Conclusion

Power usage and heat dissipation are one of the biggest challenges of VLSI industry today. In order to design the low power consuming component, without making significant change in performance, the design of FNPLL frequency synthesizer was implemented and simulated. The optimized design was implemented to 0.12 micrometer technology. Using CMOS logic, the schematics were designed and verified functionally and then prefabrication layout was sketched. The simulation curves of the layouts reflected reduction in power consumption, for the optimized design.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] R. Jacob Baker, *CMOS Circuit Design, Layout and Simulation*, IEEE Press, John Wiley & Sons, 3rd edition, 2010.

[2] A. Anil and R. K. Sharma, "A high efficiency charge pump for low voltage devices," *International Journal of VLSI Design & Communication Systems*, vol. 3, no. 3, 2012.

[3] U. L. Rohde, *Digital PLL Frequency Synthesis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.

[4] B. K. Mishra, S. Save, and S. Patil, "Design and analysis of second and third order PLL at 450 MHz," *International Journal of VLSI Design & Communication Systems*, vol. 2, no. 1, 2011.

[5] N. Weste and D. Harris, *CMOS VLSI Design—A Circuits and Systems Perspective*, Pearson Education, 3rd edition, 2005.

[6] U. A. Belorkar and S. A. Ladhake, "Design of low power phase lock loop using 45 nm VLSI technology," *International Journal of VLSI Design & Communication Systems*, vol. 1, no. 2, 2010.

[7] T. A. D. Riley, M. A. Copeland, and T. A. Kwasniewski, "Delta-Sigma modulation in fractional-n frequency synthesis," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 5, pp. 553–559, 1993.

[8] M. H. Perrott, "Fractional-N Frequency Synthesizer Design Using The PLL Design Assistant and CppSim Programs," July 2008.

[9] S. Franssila, *Introduction to Microfabrication*, John Wiley & Sons, 2004.

[10] K. Woo, Y. Liu, E. Nam, and D. Ham, "Fast-lock hybrid PLL combining fractional-N and integer-N modes of differing bandwidths," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 379–389, 2008.

[11] N. Fatahi and H. Nabovati, "Design of low noise fractional-N frequency synthesizer using sigma-delta modulation technique," in *Proceedings of the 27th International Conference on Microelectronics (MIEL '10)*, pp. 369–372, IEEE, May 2010.

[12] S. Borkar, "Obeying Moore's law beyond 0.18 micron," in *Proceedings of the 13th Annual IEEE International ASIC/SOC Conference*, pp. 26–31, September 2000.

[13] R. K. Krishnamurthy, A. Alvandpour, V. De, and S. Borkar, "High-performance and low-power challenges for sub-70 nm microprocessor circuits," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 125–128, May 2002.

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 380362, 5 pages
http://dx.doi.org/10.1155/2014/380362

Research Article

Performance Analysis of Modified Drain Gating Techniques for Low Power and High Speed Arithmetic Circuits

Shikha Panwar, Mayuresh Piske, and Aatreya Vivek Madgula

School of Electronics Engineering (SENSE), VIT University, Vandalur-Kelambakkam Road, Chennai 600127, India

Correspondence should be addressed to Shikha Panwar; shikha.panwar24@gmail.com

Received 2 May 2014; Accepted 27 June 2014; Published 15 July 2014

Academic Editor: Yu-Cheng Fan

This paper presents several high performance and low power techniques for CMOS circuits. In these design methodologies, drain gating technique and its variations are modified by adding an additional NMOS sleep transistor at the output node which helps in faster discharge and thereby providing higher speed. In order to achieve high performance, the proposed design techniques trade power for performance in the delay critical sections of the circuit. Intensive simulations are performed using Cadence Virtuoso in a 45 nm standard CMOS technology at room temperature with supply voltage of 1.2 V. Comparative analysis of the present circuits with standard CMOS circuits shows smaller propagation delay and lesser power consumption.

1. Introduction

As we move on to finer MOSFET technologies, transistor delay has decreased remarkably which helped in achieving higher performance in CMOS VLSI processors. With technology scaling, it is required to reduce the threshold and power supply voltages. As square of power supply voltage is directly proportional to dynamic power dissipation, to achieve less consumption of power, supply voltage has to be reduced. Static power and dynamic power are two main components of total power dissipation. Static power consumption is calculated in the form of leakage current through each device. Substantial increase has been observed in subthreshold leakage current with scaling of threshold voltage [1]. Subthreshold current I_{ST} is given by [1]

$$I_{ST} = \mu_0 \mathrm{Cox} \left(\frac{W}{L}\right) (m-1) \left(V_T\right)^2 \times e^{(V_g - V_{th})/mV_T} \times \left(1 - e^{-V_{DS}/V_T}\right), \quad (1)$$

where

$$m = 1 + \frac{\mathrm{Cdm}}{\mathrm{Cox}}, \quad (2)$$

where thermal voltage, $V_T = KT/q$, μ_0 is the mobility, V_g is the gate voltage, V_{th} is the threshold voltage, V_{DS} is termed as drain to source voltage, and m is the body effect coefficient. Cdm and Cox are the depletion layer and gate oxide capacitances, respectively.

To counteract the excessive leakage in CMOS circuit, many architectural techniques have been proposed over the years. Power gating [2] and stacking effect [3] are two well-known techniques for reducing leakage power dissipation. Power gating normally makes use of sleep transistors that are connected either between the power supply and the pull-up network (PUN) or between the pull-down network (PDN) and ground. Sleep transistors are switched on when the circuit is evaluating and they are switched off in standby mode to conserve the leakage power in the logic circuit. Multi-threshold-CMOS (MTCMOS) [4] technique is also an effective way to achieve considerable decline in leakage power consumption. In MTCMOS technique, high V_{th} sleep transistors are added in the circuit whereas PUN and PDN use low V_{th} devices. In dual threshold circuits [5], low V_{th} devices are used in the delay critical sections and high V_{th} devices are used to reduce the leakage current in the circuitry.

Stacking of transistor in series reduces the subthreshold leakage current when one transistor is in the off state. Stacking effect is used in sleepy stack technique [6] and force stack technique [7]. Sleepy stack technique provides better results than forced stack technique. In forced stack, an extra sleep

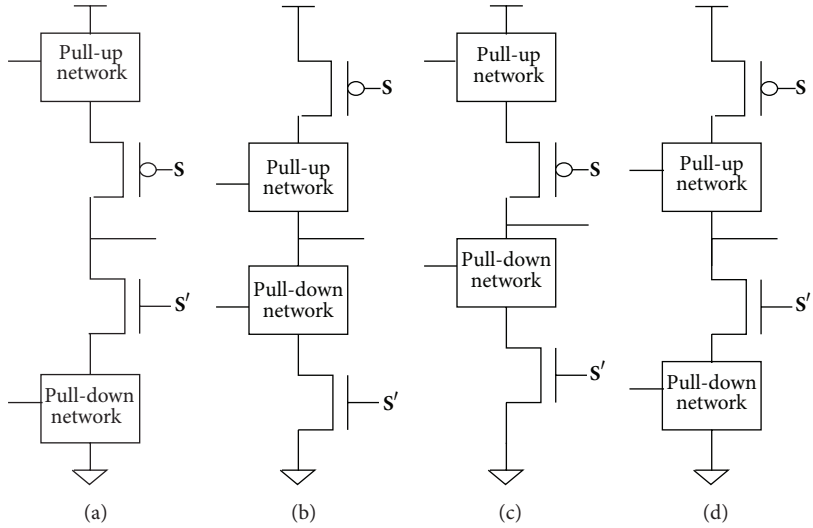


Figure 1: (a) Drain gating, (b) power gating, (c) drain-header and power-footer gating (DHPF), and (d) drain-footer and power-header gating (DFPH).

transistor is inserted for each input of the gate both in PUN and in PDN resulting in higher delay and area. In sleepy stack, an additional sleep transistor is connected in parallel with the transistor stack. This reduces the leakage current but at the same time delay in the circuit is increased.

LECTOR [8] and GALEOR [9] are also two leakage tolerant techniques. LECTOR makes use of two leakage control transistors (LCTs) that are connected between the PUN and PDN. In the same time GALEOR technique makes use of gated leakage transistors (GLTs). Both LCTs and GLTs reduce leakage by increasing the resistance between supply voltage and ground.

Another efficient technique to counter the leakage current problem is drain gating and its variation [10], explained in detail in Section 2. The modified circuits are proposed in Section 3. Simulation results taking NAND gate, 1-bit full adder, and 8-bit RCA (Ripple carry adder) as test bench circuits are enumerated in Section 4 and Section 5 provides the final conclusion.

2. Drain Gating Technique and Its Variant Circuits

In drain gating technique [10] shown in Figure 1(a), two sleep transistors are added between the PUN and PDN. PMOS transistor with sleep input **(S)** is connected between PUN and output node, whereas NMOS transistor with sleep input **(S′)** is inserted between the output node and PDN. When the circuit is in evaluation mode, the NMOS and PMOS sleep transistors are turned on resulting in low resistance conducting path. When the circuit is in standby, both transistors are switched off to reduce the standby power. Other variant circuits of drain gating are, namely, power gating, drain-header and power-footer gating (DHPF), and drain-footer and power-header gating (DFPH). In power gating technique, PMOS sleep transistor with input **(S)** is added between the power supply and the PUN, whereas NMOS sleep transistor with input **(S′)** is added between the PDN and ground as shown in Figure 1(b). The two mixed techniques DHPF and DFPH are shown in Figures 1(c) and 1(d), respectively. As the name suggests, in DHPF, a PMOS sleep switch is inserted between PUN and output node and an NMOS sleep switch is inserted between the PDN and ground rail. DFPH consists of an NMOS sleep switch between output node and PDN and a PMOS sleep switch between the power supply and the PUN. Comparative results in Section 4 indicate that power gating technique is the best leakage tolerant technique whereas drain gating technique has the least delay among the previously proposed circuits.

3. The Proposed High Speed Circuit Techniques

The proposed circuits are aimed at reducing the propagation delay incurred by drain gating technique and its variations. Four different circuit techniques, namely *high speed drain gating (HS-drain gating)*, *HS-power gating*, *HS-DHPF*, and *HS-DFPH* as shown in Figures 2(a), 2(b), 2(c), and 2(d) respectively, are proposed in this section. In *HS-drain gating* technique an additional sleep transistor with sleep input **(S)** is connected at the output node parallel to the NMOS sleep transistor **(S′)** and PDN. During the active mode, when the logic circuit evaluates the circuits output, the added NMOS sleep transistor **(S)** provides an additional discharging path in the circuit. This added transistor helps in speedy evaluation, hence providing higher speed. In a similar fashion, an additional NMOS sleep transistor with sleep input **(S)** is added to power gating, DHPF, and DFPH circuits.

The proposed cicuits have been verified by taking NAND gate, 1-bit full adder, and 8-bit RCA as test bench circuits. Experimental results in Section 4 prove that the modified HS-drain gating technique has the the least delay among

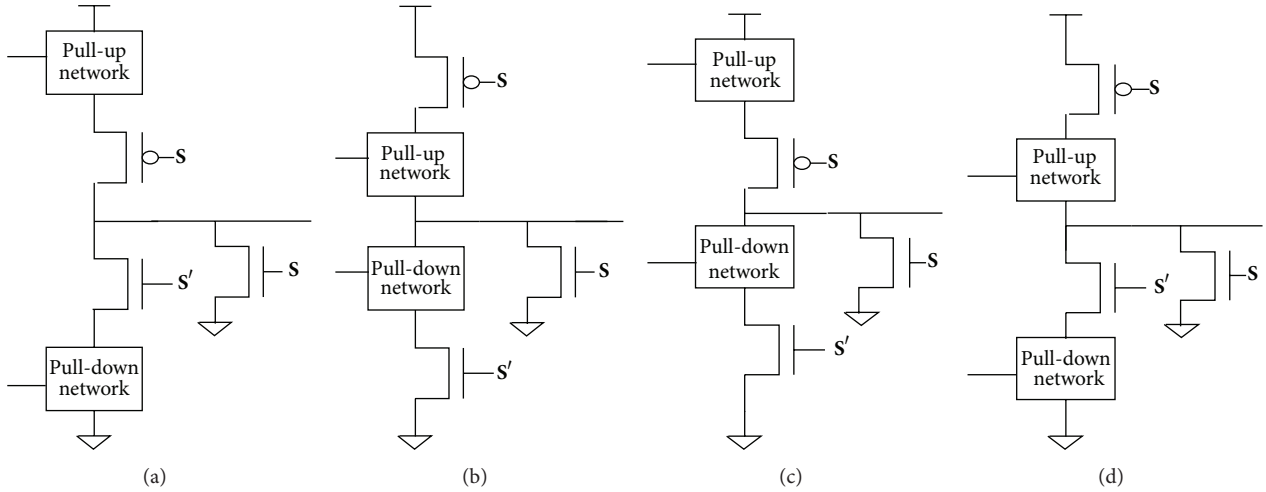


Figure 2: (a) HS-drain gating, (b) HS-power gating, (c) HS-DHPF, and (d) HS-DFPH.

Table 1: Power and delay values of NAND gate, FA, and 8-bit RCA using various techniques.

Circuit techniques	NAND gate		FA		8-bit RCA	
	Power (nW)	Delay (ps)	Power (nW)	Delay (ps)	Power (uW)	Delay (ps)
Standard CMOS	22.32	45*e*3	2.1*e*3	30*e*3	52.2	23.7*e*3
Drain gating	12.63	25*e*3	393	15*e*3	7.53	8.85*e*3
Power gating	8.73	205*e*3	238	150*e*3	2.39	20.5*e*3
DHPF	11.08	80*e*3	340	25*e*3	3.02	12*e*3
DFPH	8.71	175*e*3	245	150*e*3	4.02	15.5*e*3
HS-drain gating	18.42	2.22	250	15.08	6.97	10.3
HS-power gating	16.57	11.76	246	49.9	2.13	21.4
HS-DHPF	16.70	6.3	248	35.97	4.15	11.9
HS-DFPH	16.64	11.71	247	44.87	2.92	17

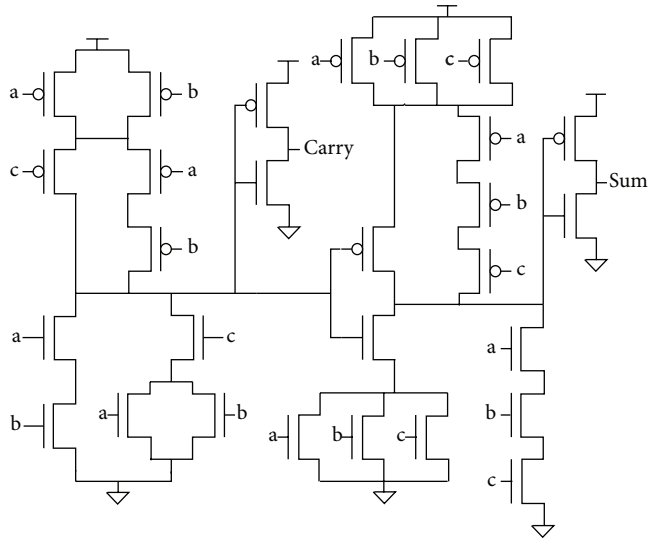


Figure 3: 1-bit CMOS full adder.

the existing and proposed architectural techniques as shown in Figure 5. Also HS-power gating technique has the lowest power as compared to standard CMOS circuit and the newly proposed circuits as shown in Table 1. The ratio of PMOS to NMOS size is set to be equal to 2.

Two-input NAND gate using HS-drain gating operates in two modes, namely, *sleep or standby mode* and *active mode*. When the circuit is in active mode, sleep input **(S)** is in low state, and output node gets charged to power supply voltage. Both NMOS and PMOS sleep transistors connected between PUN and PDN are turned on and output is evaluated. For example, if we provide input to the PUN as 0(XX) where XX stands for input vectors (00, 01, 10, 11), output will be high for the first three cases and low for the fourth case for the NAND gate. Sleep signal should be provided in the form of alternate high and low signals. When sleep signal **(S = 1)**, both PMOS and NMOS sleep transistors between PUN and PDN network turn off and additional NMOS sleep transistor is turned on, discharging the output node to ground thereby resulting in higher performance. A trade-off is achieved between power and delay so as to maintain high speed in the proposed circuits.

4. Simulations and Results

Two-input NAND gate, 1-bit full adder, and 8-bit RCA are implemented using the proposed high speed architectural techniques. The circuit diagrams for 1-bit full adder and 8-bit RCA are shown in Figures 3 and 4, respectively. Each stage in 8-bit RCA consists of a 1-bit full adder (FA). Each FA circuit consists of 28 transistors. In RCA, carry is propagated from one stage to another and final carry is obtained as C_8 shown in Figure 4.

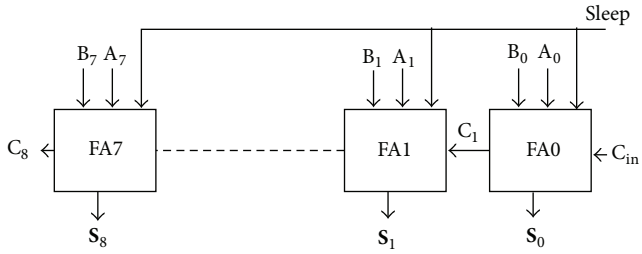


Figure 4: 8-bit RCA.

The total power consumption and propagation delay of various existing and proposed techniques for NAND gate, FA, and 8-bit RCA are compared in Table 1. HS-drain gating technique has the least delay. HS-power gating, HS-DFPH, and HS-DHPF suffer from 50%, 39%, and 13% propagation delay with respect to HS-drain gating technique. Standard drain gating and its variants circuit techniques suffer from 99% propagation delay in comparison with HS-drain gating technique. Circuits employing HS-power gating technique have very low power consumption. Power savings of nearly 85% are achieved in arithmetic architectures employing HS-power gating technique. HS-drain gating technique has the least power saving among the proposed circuits. HS-DHPF and HS-DFPH techniques optimize the power and delay in CMOS arithmetic circuits.

The corner analysis for the drain gating design and its variants is plotted along with that of the modified high speed counterparts. Figure 5 shows the temperature versus the propagation delay graph for 8-bit RCA using the existing techniques and the proposed techniques.

Similarly Figure 6 shows the plot of process corners versus the propagation delay of 8-bit RCA using the existing techniques and the proposed techniques.

On observing the comparative graph shown in Figures 5 and 6, we can infer that the designs made using the modified high speed drain gating technique and its corresponding variants have substantial reduction in the propagation delay when compared to the designs made using the CMOS, drain gating technique, and its variants.

5. Conclusions

In this paper, we have tabulated the total power consumption and the propagation delay for certain circuits using the existing low power and performance enhancing techniques and the newly proposed ones. Also we have made a comparative study of these techniques for the parameters like temperature, process corners, and propagation delay. Simulation results show that the proposed circuits work effectively even at extreme temperature and at different transistor configurations.

From the above mentioned experimental data, we can observe that, by implementing the high speed modified designs for the drain gating technique and its variants, we are able to enhance the performance of the design at lower power

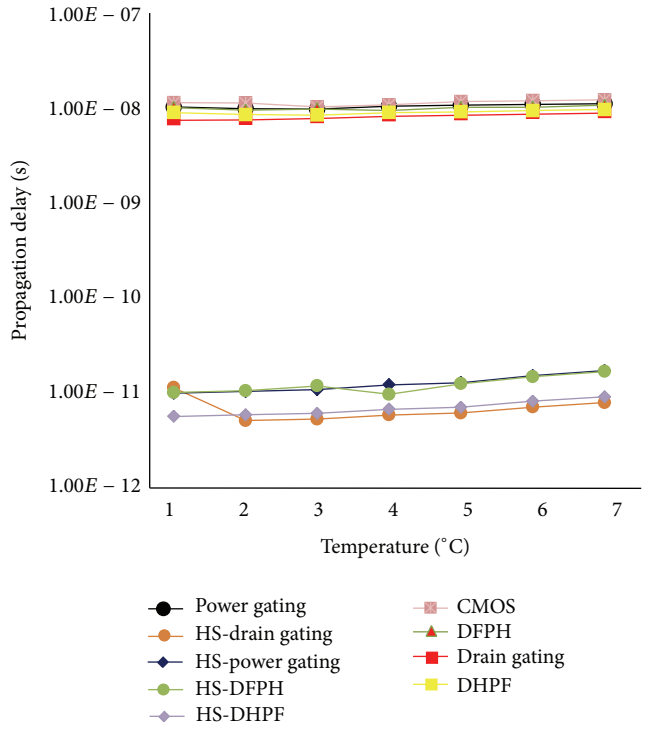


Figure 5: Temperature versus the propagation delay for the existing and the proposed techniques.

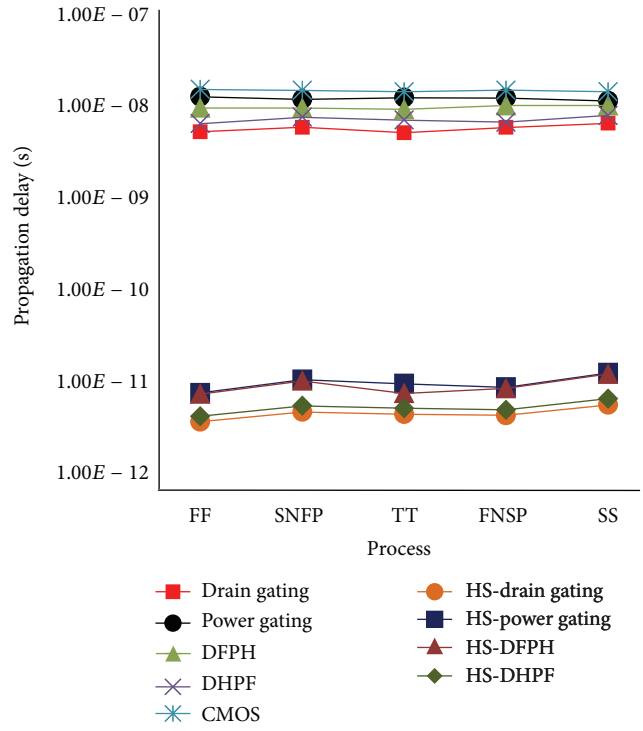


Figure 6: Process versus the propagation delay for 8-bit RCA using the existing and the proposed techniques.

consumption. Power consumption savings as observed in 8-bit RCA and 1-bit full adder are 95% and 88%, respectively, whereas propagation delay has been reduced by almost 99% in both RCA and full adder circuit.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, 2003.

[2] M. Powell, S. Yang, B. Falsafi, K. Roy, and T. N. Vijaykumar, "Gated-Vdd: a circuit technique to reduce leakage in deep-submicron cache memories," in *Proceedings of the IEEE Symposium on Low Power Electronics and Design (ISLPED '00)*, pp. 90–95, July 2000.

[3] M. Johnson, D. Somasekhar, L. Y. Chiou, and K. Roy, "Leakage control with efficient use of transistor stacks in single threshold CMOS," *IEEE Transactions on VLSI Systems*, vol. 10, no. 1, pp. 1–5, 2002.

[4] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu, and J. Yamada, "1-V power supply high-speed digital circuit technology with multithreshold-voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, 1995.

[5] L. Wei, Z. Chen, M. C. Johnson, K. Roy, Y. Ye, and V. K. De, "Design and optimization of dual-threshold circuits for low-voltage low-power applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 1, pp. 16–24, 1999.

[6] J. C. Park and V. J. Mooney III, "Sleepy stack leakage reduction," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 11, pp. 1250–1263, 2006.

[7] S. Narendra, V. De, D. Antoniadis, A. Chandrakasan, and S. Borkar, "Scaling of stack effect and its application for leakage reduction," in *Proceedings of the International Symposium on Low Electronics and Design (ISLPED '01)*, pp. 195–200, Huntington Beach, Calif, USA, August 2001.

[8] N. Hanchate and N. Ranganathan, "LECTOR: a technique for leakage reduction in CMOS dircuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 2, pp. 196–205, 2004.

[9] S. Katrue and D. Kudithipudi, "GALEOR: leakage reduction for CMOS circuits," in *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems (ICECS '08)*, pp. 574–577, September 2008.

[10] J. W. Chun and C. Y. R. Chen, "A novel leakage power reduction technique for CMOS circuit design," in *Proceedings of the International SoC Design Conference (ISOCC '10)*, pp. 119–122, November 2010.

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 529392, 12 pages
http://dx.doi.org/10.1155/2014/529392

Review Article

Gate-Level Circuit Reliability Analysis: A Survey

Ran Xiao and Chunhong Chen

Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada N9B 3P4

Correspondence should be addressed to Chunhong Chen; cchen@uwindsor.ca

Received 25 April 2014; Accepted 7 June 2014; Published 10 July 2014

Academic Editor: Yu-Cheng Fan

Circuit reliability has become a growing concern in today's nanoelectronics, which motivates strong research interest over the years in reliability analysis and reliability-oriented circuit design. While quite a few approaches for circuit reliability analysis have been reported, there is a lack of comparative studies on their pros and cons in terms of both accuracy and efficiency. This paper provides an overview of some typical methods for reliability analysis with focus on gate-level circuits, large or small, with or without reconvergent fanouts. It is intended to help the readers gain an insight into the reliability issues, and their complexity as well as optional solutions. Understanding the reliability analysis is also a first step towards advanced circuit designs for improved reliability in the future research.

1. Introduction

As CMOS technology keeps scaling down to their fundamental physical limits, electronic circuits have become less reliable than ever before [1]. The reason is manifold. First of all, the higher integration density and lower voltage/current thresholds have increased the likelihood of soft errors [2, 3]. Secondly, process variations due to random dopant fluctuation or manufacturing defects have negative impacts on circuit performance and may cause circuits to malfunction [1]. These physical-level defects would statistically lead to probabilistic device characteristics. Also, some emerging nanoscale electronic components (such as single electron devices) have demonstrated their nondeterministic characteristics due to uncertainty inherent in their operation under high temperature and external random noise [4, 5]. This may further degrade the reliability of future nanoelectronic circuits. Thus, circuit reliability has been a growing concern in today's micro- and nanoelectronics, leading to the increasing research interest in reliability analysis and reliability-oriented circuit design.

For any reliability-aware architecture design, it is indispensable to estimate the reliability of application circuits both accurately and efficiently. However, analyzing the reliability (or the error propagation) for logic circuits could be computationally expensive in general (see Section 1.3 for details). Some approaches have been reported in literature, which tackle the problem either analytically or numerically (by simulation). The contribution of this paper is to provide an extensive overview and comparative study on typical reliability estimation methods with our simulation results and/or results reported in literature.

We first review the key concepts in reliability analysis and its role in circuit design and then describe and evaluate several existing mainstream approaches for reliability analysis by looking at their accuracy, efficiency, and flexibility. Examples and simulation results are also given in order to show their advantages and disadvantages. Finally, we provide some useful suggestions on how to choose an appropriate reliability analysis method under different circumstances, along with some remarks on possible future work.

1.1. Signal Probability and Reliability. The probability of a logic signal s is by default defined as the probability of the signal being logic "1" and is expressed as $P_s = \Pr\{s = \text{"1"}\}$. The reliability of the probabilistic signal s is defined as the probability that its value is correct (i.e., it is equal to its error-free value) and is expressed as $r_s = \Pr\{s = \text{its error-free value}\}$. In gate-level design, the output signal of a gate may become unreliable due to its unreliable inputs and/or errors of gate itself. If we use the classical *von*

Neumann model [6] for gate errors, any gate can be associated independently with an error probability ε_i. In other words, the gate is modeled as a binary symmetric channel that generates a bit flip (from $0 \to 1$ or $1 \to 0$) by mistake at its output (known as *von Neumann* error [6]) symmetrically with the same probability. Thus, each gate i in the circuit has an independent gate reliability $r_i = 1 - \varepsilon_i$, which is assumed to be localized and statistically stable. Also, it is reasonable to assume that the error probability for any gate falls within $[0, 0.5]$ (or $r_i \in [0.5, 1]$).

The reliability for a combinational logic circuit (denoted by R_C) is defined as the probability of the correct functioning at its outputs (i.e., the joint signal reliability of all primary outputs). This reliability can be generally expressed as a function of gate reliabilities in the circuit (denoted by $\mathbf{r} = \{r_1, r_2, \ldots, r_{N_g}\}$, where N_g is the number of gates), as well as signal probabilities of all primary inputs(denoted by $\mathbf{P_{in}} = \{P_{\text{in}1}, P_{\text{in}2}, \ldots, P_{\text{in}N_{\text{in}}}\}$, where N_{in} is the number of primary inputs), that is,

$$R_C = \Pr\{\text{all outputs are correct}\} = f(\mathbf{r}, \mathbf{P_{in}}), \quad (1)$$

where the function f depends on the topology of the circuit under consideration. Note that the primary inputs are assumed to be fully reliable ($r_s = 1$ if s is a primary input). Under a particular case where all primary input probabilities are a constant (say 0.5), R_C turns out to be a function of $\mathbf{r}$ only.

It is worth noting that gate errors may come from either external noises (thermal noise, crosstalk, or radiation) [3] or inherent device stochastic behaviors [4]. In literature, the term "soft error" is used to emphasize the temporariness of the errors due to random external noises (e.g., glitches). In this paper, however, a more general term of *von Neumann* gate error model is used instead, as the probabilistic feature of gates is expected to exist widely and independently throughout the circuit. This differs from single-event upsets due to soft errors, where external noises are usually correlated temporally and spatially. In other words, our focus is the error propagation in combinational networks, where the gate-level logic masking is considered. For instance, some logic errors may not affect (or propagate to) final outputs if they occur in a nonsensitized portion of the circuit. Identifying these nonsensitized gates would be critical for reliability estimation and improvement.

1.2. Role of Reliability Analysis. In order to guide the IC design for reliable logic operations, it is required to develop tools that can accurately and efficiently evaluate circuit reliability, which is also a first step towards reliability improvement. However, reliability analysis is a nontrivial task due to the large size of IC circuits as well as the complexity of signal correlation and probability/reliability propagation within the circuit (as will become clear later in this paper). On the other hand, circuit reliability can be generally improved by increasing the gate reliabilities. This can be done by using redundant components. Classic redundancy techniques such as TMR [5] or NAND-multiplexing [7] achieve this by systematically replicating logic gates (other than sizing up the transistors) at the cost of increased area and power dissipation. One of the key issues in this context is to select the most critical (in terms of reliability and cost) components (or logic gates) in the circuit and improve the circuit reliability by increasing the robustness of only a few gates. In order to detect these critical gates, multiple cycles of reliability analysis are usually conducted for the whole circuit. In a more general term, accurate and efficient reliability analysis can provide a guideline for future reliability-oriented architecture design.

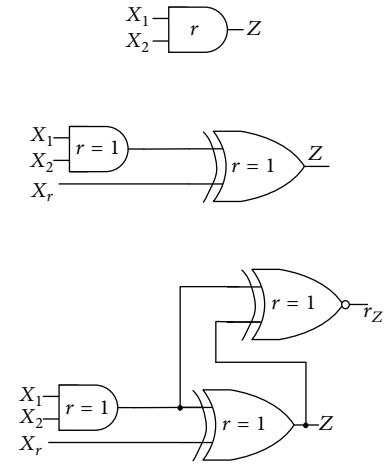


Figure 1: An AND gate and its equivalent circuit.

1.3. Complexity of Gate-Level Reliability Analysis. It is understood that the problem of determining whether the signal probability at a given node is nonzero is equivalent to the Boolean satisfiability (SAT) problem [8], a problem of determining whether there exists an interpretation that satisfies a given Boolean formula. A Boolean formula is called satisfiable if the variables of this given formula can be assigned in such a way as to make the formula evaluate to TRUE (3). The SAT has been proved to be an NP-complete problem (see [9]). The problem of computing all signal probabilities in a circuit can be formulated as a random satisfiability problem, which is to determine the probability that a random assignment of variables will satisfy a given Boolean formula [9]. The random satisfiability problem lies in a class of problems, called #P-complete, which is conjectured to be even harder than NP-complete. In the following, we show that the reliability evaluation problem is equivalent to the signal probability calculation problem and thus prove that it is also a #P-complete problem.

Let us consider a two-input AND gate ($Z = X_1 X_2$) which has the gate reliability r, as shown in Figure 1. We first add an extra XOR gate at the output, as well as an extra input X_r, with an assumption that both the XOR gate and original AND gate are error-free. The signal probability of this extra input is equal to the original gate error rate ε (i.e., $P_{X_r} = \Pr\{X_r = \text{"1"}\} = 1 - r$). This ensures that the output Z of this extra XOR gate is equivalent to the original output of the AND gate.

For a combinational logic circuit, we first duplicate the whole circuit. In the original circuit, we make each gate error-free in order to compute the correct value at primary

outputs. For the duplicated one, we extract the reliability of each gate using the aforementioned method (as a result, all gates are also error-free in the duplicated circuit and the gates' number is doubled). Then, we add 2-input XNOR gates for each pair of corresponding primary outputs in the original and duplicated circuits. Thus, the output reliability can be expressed as the signal probability at the output of the XNOR gates. By doing so (i.e., duplicating the circuit and extracting gate reliabilities), we see that the reliability estimation of original circuit is equivalent to the problem of computing the signal probabilities of the transformed circuit.

For a combinational logic circuit with N_{in} primary inputs, N_{out} primary outputs, and N_g logic gates, the problem of evaluating the signal reliability of all primary outputs and their joint reliability (i.e., the overall circuit reliability R_C) can be solved by exhaustively calculating all $2^{(N_{\text{in}}+N_g)}$ scenarios. In each scenario, the expected (correct) output and actual output values need to be calculated with the complexity of $O(N_g)$. The total complexity is then $O(N_g \cdot 2^{N_{\text{in}}+N_g})$. As circuits become very large, it would be difficult or even impossible to perform the exact analysis of the reliability due to the exponential complexity. Usually, some tradeoff has to be made between the accuracy and efficiency for reliability analysis.

In order to tackle this issue, a number of different approaches have been reported in literature, including probabilistic transfer matrix (PTM) method [10–12], Bayesian networks (BN) [13–15], Markov random field (MRF) [16–20], Monte Carlo (MC) simulation, testing-based method [3], stochastic computation model (SCM) [2, 21], probabilistic gate model (PGM) [22–25], observability-based analysis [26], Boolean difference-based error calculator (BDEC), and correlation coefficient method- (CCM-) based approaches [8, 26–28]. In the following, we overview some of these approaches and analyze their pros and cons in terms of accuracy, efficiency, and flexibility with simulation results.

2. Probabilistic Transfer Matrix (PTM) Method

An accurate analytical model for reliability analysis problem is based on the probabilistic transfer matrices (PTMs), which compute the circuit output reliability for all input patterns [10, 11]. This computational framework begins with the definition of a probability matrix which is used to represent the probability of a logic gate's output for each input pattern. For instance, the probability matrix representation for a two-input NAND logic gate is shown in Figure 2, where each column of the matrix $\mathbf{M}_g$ represents the probability of the gate output Z being "0" or "1" for all different input patterns (i.e., X_1X_2 = "00," "01," "10," and "11"). For example, the element $\mathbf{M}_{11} = \Pr\{Z = 0 \mid X_1X_2 = 00\} = 1 - r$, where r is the gate reliability. In general, the probability matrix for an n-input 1-output gate is a $2^n \times 2$ matrix.

For a circuit, all gate probability matrices shall be combined together to construct the PTM of the whole circuit. More specifically, the serial and parallel connections of gates correspond to a matrix product and tensor product [10], respectively. The fanout behavior is represented by explicit fanout gates, where a 1-input m-output fanout gate is simply mimicked by a 1-input m-output buffer gate. A fault-free circuit has an ideal transfer matrix (ITM), where the correct value of the output occurs with the probability of 1. This means that, in each row of the PTM, there is single "1" for the correct output value and there are "0"s for other output combinations. The circuit reliability (i.e., the probability of outputs being correct) is evaluated by comparing its PTM and ITM.

The process of combining gate probability matrices implicitly takes into account the signal dependency between gates by considering the underlying joint and conditional probabilities within the circuit. As a result, the calculation of the circuit PTM is exact. However, the limited scalability is often a price that has to be paid for this computational framework to capture complex circuit behaviors. Consider a combinational logic circuit with N_{in} primary inputs, N_{out} primary outputs, and N_g logic gates. The circuit PTM is a matrix with $2^{N_{\text{in}}}$ rows and $2^{N_{\text{out}}}$ columns (i.e., $2^{N_{\text{in}}} \times 2^{N_{\text{out}}}$), which contains the transition probability from all input combinations toward all output combinations. In other words, its space complexity is $O(2^{N_{\text{in}}+N_{\text{out}}})$. This exponential space requirement is the main bottleneck of PTM approach. Particularly, for a computer with 2 GB memory, the maximum size of the circuit that can be handled is limited to 16 input/output signals. By utilizing some advanced computation methods (such as algebraic decision diagrams (ADDs) and encoding [10, 11]), the signal width may be extended up to ~50, where the signal width is defined as the largest number of signals at any level in the circuit. Unfortunately, this limit is still computationally unacceptable in the real world for large-scale benchmark circuits (e.g., C2670 which has 157 inputs and 64 outputs). Nonetheless, for small circuits, the PTM is a very good analytical method, as it provides exact results within a reasonable runtime and shows the probabilistic behavior of unreliable logic gates.

Also, this approach can serve as the foundation of many other heuristic approaches by providing other important information such as signal probabilities and observability, with the capability of analyzing the effect of electrical masking on error mitigation as well. For instance, in [10], the observability of a gate g is defined as the ratio of the error probability of the whole circuit and the error probability ε_i of this gate, that is, $(1 - R_C(\varepsilon_i))/\varepsilon$, where $R_C(\varepsilon_i)$ is the circuit reliability when the only unreliable gate is ith gate (with all other gates being error-free). Clearly, the gate with highest observability can be regarded as the most susceptible, meaning that it will impact (or decrease) the circuit reliability the most. It should be noted that this only represents the simplest case where only single gate failure is considered. In most real cases, however, the gate observabilities may not be independent, and thus the joint observabilities usually need to be considered instead.

The detailed algorithm with the PTM is summarized as follows.

Step 1. Levelize the circuit; compute PTMs of each logic component in each level denoted by $\mathbf{M}^j_{\text{Lv}_i}$.

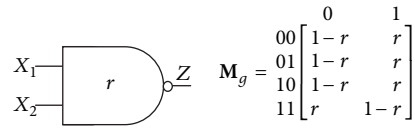


Figure 2: (a) A 2-input NAND gate and (b) its probability matrix $\mathbf{M}_g$ (according to [10]).

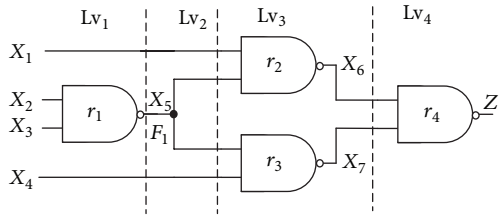


Figure 3: The example circuit schematic (a portion of C17 benchmark circuit).

Step 2. Within one level, the PTMs of each logic components (gates, wires, and fanout nodes) are tensored together to form the PTM of the current level; that is, $\mathbf{M}_{\mathrm{Lv}_i} = \mathbf{M}^1_{\mathrm{Lv}_i} \otimes \mathbf{M}^2_{\mathrm{Lv}_i} \cdots$;

Step 3. The PTMs of all levels are then multiplied together to get the circuit PTM; that is, $\mathbf{M} = \prod_i \mathbf{M}_{\mathrm{Lv}_i}$.

Step 4. Calculate the ideal transfer matrix $\mathbf{J}$ using the truth table of the logic function (error-free signal probabilities $p(i)$ for input patterns are evaluated with the computation complexity of $O(N_g \cdot 2^{N_{\mathrm{in}}})$).

Step 5. The circuit reliability is given by [11]:

$$R_C = \sum_{i,j}^{J(i,j)=1} M(i,j)\, p(i). \tag{2}$$

We take a simple circuit as an example to illustrate the analysis process of PTM approach. The circuit schematic is shown in Figure 3, where the circuit has 4 levels, and the fanout F_1 reconverges at gate number 4, generating the dependency between signal X_6 and X_7. Since there are four inputs (X_1, X_2, X_3, and X_4) and one single output Z, the circuit PTM would be a 16×2 matrix $\mathbf{M}$ which stores the probability of occurrence of all input-output vector pairs. The $\mathbf{M}$ is constructed by combining PTMs of all levels (using matrix product due to serial connection in this case), while the PTM of each level is calculated by combining PTMs of each logic components within the current level (using tensor product due to their parallel connection). More specifically, we have (based on [10])

$$\begin{aligned}\mathbf{M} &= \left(I \otimes \mathrm{NAND}_1 \otimes I\right)_{16\times 8} \cdot \left(I \otimes F \otimes I\right)_{8\times 16} \\ &\quad \cdot \left(\mathrm{NAND}_2 \otimes \mathrm{NAND}_3\right)_{16\times 4} \cdot \left(\mathrm{NAND}_4\right)_{4\times 2} \\ \mathbf{I} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad F = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ \mathrm{NAND}_i &= \begin{bmatrix} 1-r_i & r_i \\ 1-r_i & r_i \\ 1-r_i & r_i \\ r_i & 1-r_i \end{bmatrix},\end{aligned} \tag{3}$$

where the matrix $\mathbf{I}$ refers to a 2×2 identity PTM, and each parenthesized term in (3) corresponds to a specific circuit level. Assuming the gate reliabilities are $r_1 = r_2 = r_3 = r_4 = 0.95$ and the probability of all input signals is equally 0.5, the circuit PTM and ideal transfer matrix are found using the above algorithm as follows:

$$\mathbf{M} = \begin{matrix} 0.8622 & 0.1378 \\ 0.1312 & 0.8688 \\ 0.8622 & 0.1378 \\ 0.1312 & 0.8688 \\ 0.8622 & 0.1378 \\ 0.1312 & 0.8688 \\ 0.8622 & 0.1378 \\ 0.8238 & 0.1762 \\ 0.1312 & 0.8688 \\ 0.0928 & 0.9073 \\ 0.1312 & 0.8688 \\ 0.0928 & 0.9073 \\ 0.1312 & 0.8688 \\ 0.0928 & 0.9073 \\ 0.8238 & 0.1762 \\ 0.8217 & 0.1783 \end{matrix}, \qquad \mathbf{J} = \begin{matrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{matrix}. \tag{4}$$

It can be seen from $\mathbf{M}$ that the output reliability depends on input patterns. The lowest and highest values for the output reliability are 0.8217 and 0.9073, which occur when the input vector $(X_1X_2X_3X_4) = (1111)$ and (1001, 1011, and 1101), respectively. The circuit reliability is found to be $R_C = 0.8658$ with the runtime of 0.2798 s.

The PTM algorithm has been implemented on some small circuits. The simulation results show that its performance is fairly good for circuits with less than 20 gates. If the circuit

size increases to ~40, both runtime and memory cost will grow dramatically, making the PTM method computationally expensive. In order to handle large-scale circuits, a variant PTM method was proposed in [11], where the input vector sampling is used. The simulation results show that this does improve efficiency with reduced memory cost, while the accuracy remains to be seen.

In summary, the PTM method has two major limitations. First, the signal width of the circuit that can be analyzed is very limited. This is due to the fact that its space complexity grows exponentially with the number of inputs and outputs, leading to prohibitively massive matrix storage and manipulation overhead for large-scale circuits. Secondly, the circuit structure needs to be preprocessed (such as circuit levelization and identification of the fanout nodes and wire pairs) prior to the algorithm implementation. Also, the PTM assumes all signals are correlated, which makes the method less efficient for circuits with no or a few reconvergent fanouts.

3. Monte Carlo (MC) Simulation

MC is a widely known simulation-based approach, where experimental data are collected to characterize the behavior of a circuit by randomly sampling its activity [2]. It is usually used when an analytical approach is unavailable or difficult to implement. The obvious drawbacks of this approach lie in the fact that numerous pseudorandom numbers need to be generated, and a large number of simulation runs must be executed to reach a stable result. This makes the reliability analysis for large circuits a very time-consuming process. As a stochastic computation framework, the MC method makes the result gradually converge to its exact value as more simulation runs are performed. In the process of achieving relatively stable results, certain statistical parameters (such as standard deviation σ and/or coefficient of variance (CV) which is defined as the ratio of the standard deviation and the mean, i.e., σ/μ) are usually used as the stopping criteria. In [2], CV = 0.001 is used to represent an acceptable level of accuracy, and the number of simulation runs required is given by

$$N_{\mathrm{MC}} = \frac{1 - R_C}{R_C} \cdot \frac{1}{\mathrm{CV}^2} \approx 10^6 \cdot \left(\frac{1}{R_C} - 1 \right), \tag{5}$$

where R_C is again the circuit reliability. Since the circuit reliability usually decreases with the circuit size (N_g), the N_{MC} will increase with the circuit size for a given accuracy (measured by CV). Assuming that the R_C ranges from 0.1 to 0.9, the number of MC runs will vary around $10^5 \sim 10^7$. It should be mentioned that (5) only gives an approximated range of N_{MC}, and its actual value is usually determined experimentally for real circuits. Let us take the circuit of Figure 3 again as an example. From (5), the required N_{MC} is $\sim 1.55 \times 10^5$ if $R_C = 0.8658$. Figure 4 shows the relative error at R_C against N_{MC}. It can be seen from the figure that after $\sim 10^4$ runs, the result becomes relatively stable around its final value. However, a small random fluctuation is inevitable. Even after $\sim 10^5$ simulation runs, the relative error of the MC result is a nonzero value ($5.636e-04$), indicating a low convergent rate with the MC. This is a common feature for stochastic computations.

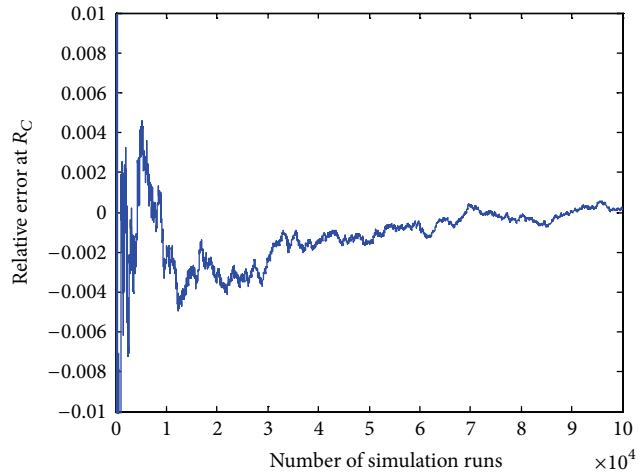


FIGURE 4: The relative error of circuit reliability R_C of Figure 3 versus the number of MC simulation runs N_{MC}.

4. Stochastic Computation Model (SCM)

Unlike the MC method which uses Bernoullisequences for simulation, the SCM approach takes non-Bernoulli sequences [2, 21]. In a non-Bernoulli sequence, for a given probability p and a sequence length N, the number of "1"s to be generated is fixed and given by $N \cdot p$, and only the positions of the "1"s are determined by a random permutation of binary bits. Therefore, in SCM approach, less pseudorandom numbers are generated for the same length of simulation, compared to MC simulation where pseudorandom numbers are independently generated for each gate or input to mimic the behavior of probabilistic circuits [2].

Consider a circuit with N_{in}, N_{out}, N_g, $\mathbf{P_{in}}$, and ε (refer to the previous sections for definitions of these variables). If we use a sequence length of N, the total required number of random numbers is given by $(N_{\mathrm{in}} + N_g) \cdot N$ in MC simulation. In contrast, for the SCM approach with the same sequence length, only $N\varepsilon$ pseudorandom numbers need to be generated (for the positions of "1"s) for a gate with error rate ε. Therefore, the total number of random numbers is reduced to $(N_{\mathrm{in}} \cdot p_{\mathrm{in}} + N_g \cdot \varepsilon) \cdot N$. Since the gate error rate ε is usually a small value which can be viewed as a scale factor, the total required random number is significantly reduced. In other words, for a specific level of accuracy, the non-Bernoulli sequence requires a smaller sequence length than the Bernoulli sequence does. However, how to efficiently determine the required minimum sequence length for the SCM is still an open question. In [2], an empirical function (rather than an analytical expression) was used for this purpose.

Again, we took the example circuit of Figure 3 and used the same sequence length with MC (i.e., $N_{\mathrm{SCM}} = N_{\mathrm{MC}} = 10^5$) with gate error rate $\varepsilon = 0.001$. The SCM and MC

Table 1: Runtime comparison of MC and SCM on benchmark circuits ($\varepsilon = 0.01$).

Circuit	Size	MC (10^6 runs) Runtime (s)	SCM (10^6 runs) Runtime (s)	
			$\varepsilon = 0.01$	$\varepsilon = 0.1$
c432	160	183	31	38
c499	202	203	37	45
c880	383	373	63	77
c1355	546	472	92	111
c1908	880	842	183	215
c2670	1193	1151	265	311
c3540	1669	1616	409	505
c5315	2406	2548	786	961
c7552	3512	3732	1325	1495

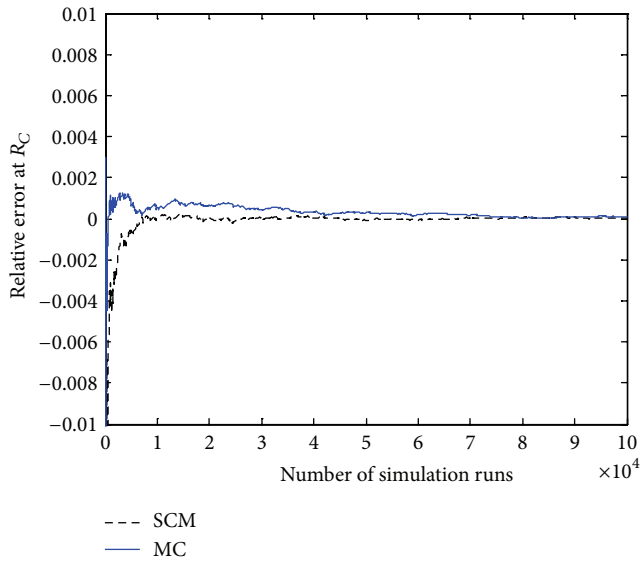


Figure 5: The relative error of circuit reliability R_C versus the number of simulation runs N_{MC}/N_{SCM}.

simulation results are compared in Figure 5, where both have a similar convergence rate. However, the runtimes with SCM and MC are $T_{SCM} = 0.0745$ s and $T_{MC} = 1.2528$ s, respectively, indicating that the SCM method is more efficient than the MC. This efficiency improvement is mainly due to less random numbers that are generated in the SCM simulation.

We also implemented both SCM and MC approaches in *Matlab* with the same sequence length of 10^6 (gate error rate $\varepsilon = 0.01$) and tested their performance on ISCAS'85 benchmark circuits. The results are shown in Table 1, where the runtime with the SCM is around 1/6~1/3 of that with the MC. One of the disadvantages of SCM is the difficulty in determining its simulation sequence length N_{SCM}. Also, its runtime is proportional to gate error rate ε as well as input probabilities. If ε is relatively large (say 0.2), the runtime improvement of SCM over MC would be marginal (only scaled by a constant).

5. Probabilistic Gate Model (PGM)

The PGM is another reliability analysis method which is based on the probabilistic models of unreliable logic gates [22–25]. In the simple version of PGM, the input signals of each gate in the circuit are assumed to be independent. Under this assumption, the output probability of each gate can be easily calculated using the information of input signal probabilities and gate error rate. For instance, consider a 2-input NAND gate with input probabilities of X_1 and X_2 and gate error rate of ε. Its output signal probability can be expressed as (after [24])

$$\begin{aligned} Z &= \Pr(\text{“1”} \mid \text{gate faulty}) \cdot \Pr(\text{gate faulty}) \\ &\quad + \Pr(\text{“1”} \mid \text{gate not faulty}) \cdot \Pr(\text{gate not faulty}) \\ &= (1-\varepsilon) + (2\varepsilon - 1) X_1 X_2. \end{aligned} \tag{6}$$

This output probability Z can be used recursively as the input information at next level of gates. One of the main features with PGM is that the circuit reliability is analyzed by exhaustively evaluating each input combination and output. For any given input combination, the error-free output value Z_{ef} is calculated, and then the output signal probability P_O is evaluated using the PGM of all gates in the circuit. Depending on the error-free output value, the output reliability R for this specific input combination is given by [24]

$$R = \begin{cases} P_O, & Z_{ef} = 1, \\ 1 - P_O, & Z_{ef} = 0. \end{cases} \tag{7}$$

Finally, the overall output reliability is the weighted sum of all conditional output reliabilities over all possible input combinations, where the weight is the probability of a specific input combination.

Intuitively, the operation process of PGM is similar to PTM in the sense that both of them consider all input combinations in a forward topological order. An obvious disadvantage with the PGM approach is that it is almost impossible to exhaustively enumerate all input combinations when the number of inputs increases (say to 30 and above). Therefore, a certain sampling technique is often necessary for large circuits. The input patterns sampling becomes another source of errors, in addition to the inaccuracy caused by signal independence assumption in constructing gate PGMs (it should be pointed out that while signal correlations due to fanouts originating from the primary inputs are eliminated by assigning the deterministic values (either "0" or "1") to all primary inputs, those caused by other reconvergent fanouts nodes are not).

In order to eliminate all signal correlations, an accurate PGM algorithm was proposed in [24] where deterministic values are assigned explicitly to all reconvergent fanout nodes within the circuit. More specifically, for each fanout, the original circuit is transformed to two auxiliary circuits [24], one with the fanout node being set to logic value "0" and the other to "1." In each of these two circuits, the output probability is computed by using conditional probabilities for the given value at the fanout. This procedure is executed iteratively until

Table 2: Simulation results for simple PGM in comparison with MC.

Circuit	Size	Simple PGM approach (10^3 samples)			Monte Carlo (10^6 runs)
		Average error (%)	Max error (%)	Runtime (s)	Runtime (s)
C432	160	0.54	1.59	4.66	183
C499	202	0.1	0.31	4.92	203
C880	383	0.61	2.83	9.17	373
C1355	546	1.26	1.66	12.21	472
C1908	880	0.39	0.85	18.69	842
C2670	1193	2.43	16.61	25.72	1151
C3540	1669	0.077	2.27	39.46	1616
C5315	2307	10.88	43.16	61.42	2548
C7552	3512	2.68	13.68	75.19	3732
Average	—	**2.18**	**9.22**	—	—

Table 3: Comparison of simple PGM and accurate PGM.

Circuit	Size	Simple PGM approach (10^3 samples)		Accurate PGM approach (10^3 samples) [24]
		Average error (%)	Runtime (s)	Runtime (s)
Cu	43	1.37	0.0277	0.10
z4ml	45	0.94	0.0039	0.05
x2	38	0.52	0.0275	0.22
Mux	50	0.52	0.282	0.10

all fanouts have been processed. If all input combinations are simulated, this procedure will lead to exact results for any circuits. However, for a circuit with N_f reconvergent fanouts, a total of 2^{N_f} auxiliary circuits are required and analyzed. Therefore, the computation complexity becomes $O(N_g \cdot 2^{N_{\text{in}}+N_f})$ [24]. However, in many real circuits, the number of reconvergent fanouts N_f is comparable to the number of gates (N_g). Thus, the complexity of the above accurate PGM algorithm is still an exponential function of the circuit size, making it infeasible in general for large circuits.

In an effort to improve the efficiency of the accurate PGM method, a modular PGM approach was also introduced in [24]. It is based on the observation that many large circuits contain a limited number of simple logic components that are used repeatedly. With this in mind, circuits can be decomposed into several modules whose reliabilities are calculated using the accurate PGM method. The circuit output reliability is then evaluated by combining these modules along the path from primary inputs. Unfortunately, the input sampling is still needed in this case for large-scale circuits.

For the example circuit of Figure 3 with 4 input signals, a total of 16 input combinations need to be considered. We plot the conditional output reliability for each input combination in Figure 6, which shows that the output reliability varies within a relatively small range (no more than ±10%) for different input combinations. In other words, the input vector sampling can be implemented effectively with small errors. The overall output reliability is given by a weighted sum over all input combinations and is found to be $R_C = 0.8701$ (with the runtime of $T_{\text{PGM}} = 0.0093$ s), compared to the accurate value of 0.8658 given by PTM (i.e., the relative error is as low as ~0.5%).

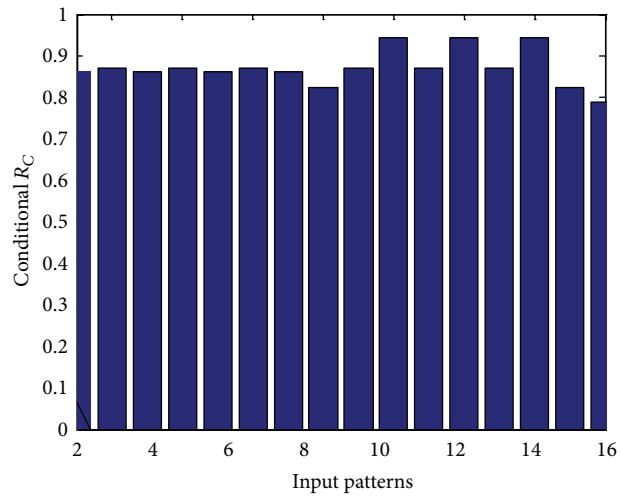


Figure 6: The conditional output reliability of Figure 3 for different input combinations (x-axis labels 1~16 indicate 16 input patterns from 0000~1111).

In order to see the performance of different PGM algorithms on large circuits, we implemented the simple PGM algorithm in *Matlab* and tested it on ISCAS'85 benchmarks. The results are shown in Table 2. We also compare the simple PGM with both accurate and modular PGM methods in Tables 3 and 4, where the simulation results for both accurate and modular PGM methods are taken from [24].

It can be seen from these tables that the simple PGM algorithm can provide highly accurate results if the circuits (such as C432 and C1355) have no or few reconvergent fanouts and/or if the fanouts originate from the primary inputs.

Table 4: Simulation results for modular PGM in comparison with MC.

Circuit	Size	Modular PGM approach (10^3 samples) [24]		Monte-Carlo (10^6 runs)
		Average error (%)	Runtime (s)	Runtime (s)
C432	160	9.21	0.25	183
C499	202	0.11	2.15	203
C1355	546	4.2	2.98	472
C2670	1193	0.43	4.26	1151

For those circuits with significant fanouts (such as C2670 and C5315), the average (or maximum) errors for the simple PGM can increase significantly (in particular, the maximum error is up to 43% for C5315, as shown in Table 2). From Table 3, the accurate PGM need longer runtimes than the simple PGM for small circuits. Results in Table 4 confirm that the modular PGM is very efficient while the accuracy may not always be good enough for some circuits (with an average error of 9% for C432).

In summary, for all the above three different versions of PGM, the input sampling is inevitable for improved efficiency if the number of primary inputs N_{in} is large (~30). This is mainly where the analysis errors come in. Thus, it can be concluded that they represent a good model only for circuits with a small number of primary inputs, where no input sampling is required. For the circuits without reconvergent fanouts, the input sampling in the PGM approach is unnecessary, because both signal probability and output reliability in this case can be computed within $O(N_g)$ time (see [29] for details).

6. Observability-Based Reliability Analysis

Another reliability analysis method was presented in [26], which is based on the observation that an error at the output of any gate is the cumulative effect of a local error component attributed to the error probability of the gate, and a propagated error component was attributed to the failure of gates in its transitive fan-in cone. In [26], the observability of a gate (or its output signal) is the conditional circuit error probability given the single error at current gate. The value of this observability can be simply defined as $o_i = (1 - R_C(\varepsilon_i = 1))$, where $R_C(\varepsilon_i = 1)$ is the circuit reliability given a single error with the current gate, and can be calculated using Boolean differences [29], symbolic techniques (such as BDDs), or simulation method. It can be expected that the gate observabilities are highly related to the input probabilities.

For a single-fault case (i.e., only one gate in the circuit is erroneous), the circuit reliability (assuming a single primary output) can be simply calculated by considering each fault case individually. Assume that the error rate and observability of the ith gate are ε_i and o_i, respectively. If gate i is erroneous while the other gates are fault-free, the output reliability simply is equal to o_i. Thus, the overall reliability can be easily calculated by

$$R_C = \sum_{\text{all } i} \left(\varepsilon_i \cdot o_i \cdot \prod_{j \neq i} \left(1 - \varepsilon_j\right) \right), \tag{8}$$

which is exact for the single-fault case.

If a multiple-error case is considered, the complexity of computing the reliability will grow exponentially with N_g. In order to improve the efficiency in this case, the following two assumptions are used in [26]: (a) the impacts of gate failures on the primary output are decoupled, which implies that the output is erroneous if an odd number of gates are simultaneously observable and (b) the observabilities of all gates are independent. As a result, the simultaneous observability of multiple gates is simply the product of their individual observabilities.

We took the example circuit of Figure 3 for illustration. First, let us assume all four gates ($G_1 \sim G_4$) in the circuit are erroneous with the probabilities of ε_1, ε_2, ε_3, and ε_4, respectively (other cases can be analyzed similarly). Based on the above assumption (a), we only need to consider the cases where an odd number (1 or 3) of gates is simultaneously observable. This means that when an even number (0, 2, or 4) of gates is observable, the output signal Z will has correct value as gate errors are logically masked by one another. Secondly, under the assumption (b), the probability of only one gate being observable is given by $\sum_i (o_i \cdot \prod_{j \neq i} (1 - o_j))$ (the probability of three gates being simultaneously observable can be calculated similarly). Based on these assumptions, a closed-form expression for the circuit reliability of the circuit (assuming a single primary output) can be written generally as a function of error probabilities and observabilities of all gates [26]; that is,

$$R_C = \frac{1}{2} \left(1 + \prod_i \left(1 - 2\varepsilon_i o_i\right) \right), \tag{9}$$

which can be computed efficiently if all gate observabilities are known (however, this analysis is only suitable for small circuits or large ones with small values of gate error probabilities, which will be clear later). The gate observability can be determined using the PTM method. For instance, the observability of gate G_1 in Figure 3 is calculated as the output reliability by setting $r_1 = 0$ and $r_2 = r_3 = r_4 = 1$. The results are $[o_1, o_2, o_3, o_4] = [0.25, 0.375, 0.375, 0]$. We calculate the circuit reliability R_C using the above expression and plot the results against the accurate values given by the PTM in Figure 7(a) for different values of gate reliability. The relative error is shown in Figure 7(b). It can be seen clearly from these figures that the observability-based analysis is only accurate for small gate error rates, in which case the probability for single gate failure is significantly higher than that for multiple gate failures.

To reduce the computational complexity of the above observability-based reliability analysis, [26] also proposed

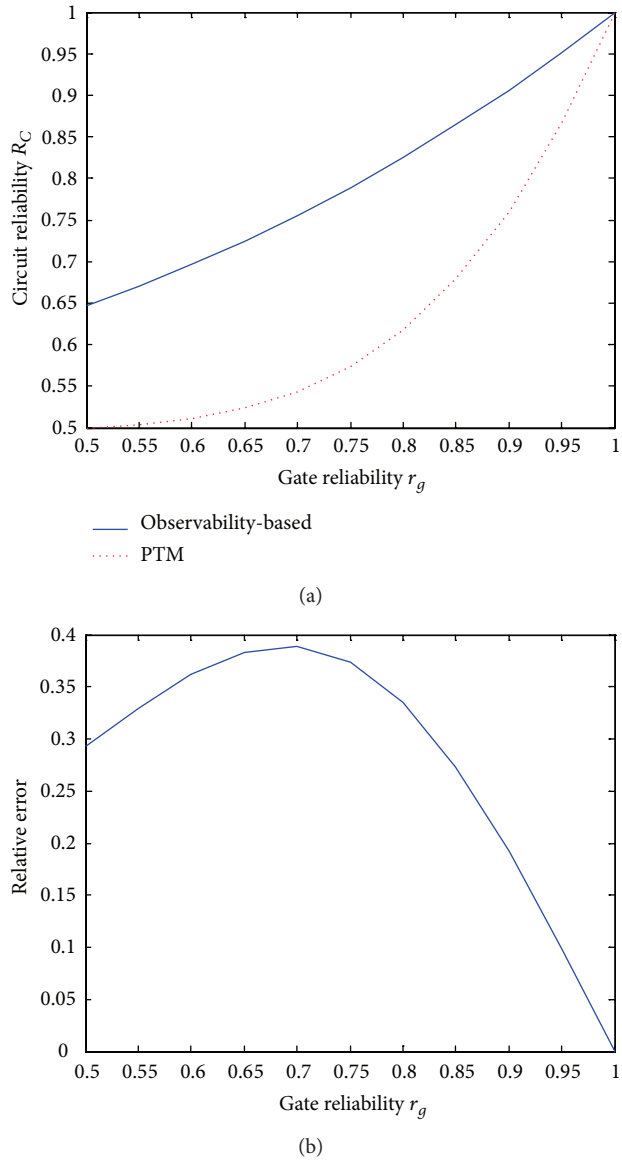


FIGURE 7: (a) Circuit reliability R_C versus gate reliability and (b) relative error versus gate reliability for the example circuit of Figure 3.

a sampling algorithm by considering the constraint that only a maximum of k gates can fail simultaneously. This algorithm first generates a set of samples for failed gates and guarantees that the total number of gates with error is no more than k. Then, a single-pass reliability analysis algorithm [26] was used to evaluate the error probability at the primary outputs, leading to the computational complexity of $O(N_g \cdot k^2)$, where N_g is the number of gates with error. For a specific sample, the reliabilities of gates in the sampling are set to be 0 and the rest are set to be 1. Finally, the overall circuit reliability is estimated by averaging the reliabilities over all samples. Therefore, this maximum-k gate failure model can be viewed as a hybrid method that makes a trade-off between the accuracy of simulation-based method and the efficiency of analytical approach. It provides more accurate results than the single-pass algorithm [26] and takes the shorter runtime than MC or SCM.

7. Correlation Coefficient Method (CCM)

CCM is a widely used approach that evaluates the signal probabilities for (fault-free) combination circuits [30]. As mentioned before, the reliability analysis can be transformed to signal probability computation. Therefore, the CCM can be used to evaluate the reliability estimation [8, 26–28, 30]. The main idea of CCM is briefly described below.

In order to compute the signal probability, the correlation coefficient between two probabilistic signals (denoted by i and j) is defined as [8]

$$C_{i,j} = C_{j,i} = \frac{P(ij)}{P(i)\,P(j)} = \frac{P(j \mid i)}{P(j)}, \tag{10}$$

which is equal to 1 for signals i and j are independent. It should be noted that, here, only the first order correlation coefficients are considered, and the correlation of two signals with a third one (denoted by h) is approximated as $C_{ij,h} = C_{i,h} \cdot C_{j,h}$. For reliability computation, four correlation coefficients for a pair of signals are needed. Each coefficient corresponds to a combination of events (i.e., $0 \to 1$ or $1 \to 0$ error) on the signal pair. In other words, the signal error (or reliability) correlation coefficient between signals i and j is defined as [26]

$$\begin{aligned}
C_{ij} &= \frac{P(i_{0\to1} j_{0\to1})}{P(i_{0\to1})\,P(j_{0\to1})}, \\
C_{i\tilde{j}} &= \frac{P(i_{0\to1} j_{1\to0})}{P(i_{0\to1})\,P(j_{1\to0})}, \\
C_{\tilde{i}j} &= \frac{P(i_{1\to0} j_{0\to1})}{P(i_{1\to0})\,P(j_{0\to1})}, \\
C_{\tilde{i}\tilde{j}} &= \frac{P(i_{1\to0} j_{1\to0})}{P(i_{1\to0})\,P(j_{1\to0})},
\end{aligned} \tag{11}$$

where the $P(i_{0\to1})$ is the probability that the value of signal i flips to 1 from its correct value 0, that is, the error probability of i given that its error-free value is 0. Once the error correlations and error-free signal probabilities are generated, the single-pass analysis is conducted using the forward topological order with the computational complexity of $O(N_g)$. Since the computation complexity of CCM is linear with the number of levels (L) and pseudoquadratic with the number of gates per level (N_L), the overall complexity of CCM-based reliability analysis turns out to be $O(N_g^{1.5})$ if a square circuit is assumed (i.e., $N_L = L = N_g^{0.5}$). This complexity is an upper bound as not all signals are correlated in real circuits.

In [26] which uses the CCM, an average relative error of up to ~13% over all outputs was reported for circuits with significant fanout (e.g., C499 and C1355) when the gate error rates range within [0, 0.5] (for other benchmark circuits,

the error was around 2~6%). Also, the relative errors may not be mitigated significantly by using more correlation coefficients. For instance, by using 0, 4, and 16 correlation coefficients, the relative errors for C499 are only improved to 13.1%, 11.2%, and 11.11%, respectively [26], where the zero-coefficient case means that all signals are treated as independent with the computation complexity of $O(N_g)$. It is shown in [26] that the runtime of using 4 coefficients is several orders of magnitude longer than the zero-coefficient case (~100 s versus ~1 s, for circuit with ~1000 gates). Therefore, it may not be worthwhile to calculate more correlation coefficients for slightly improved accuracy. In [30], the relative error for large circuits (with hundreds of gates) was reported at ~7% on average with the runtime of ~10 s, which is comparable to those from [26].

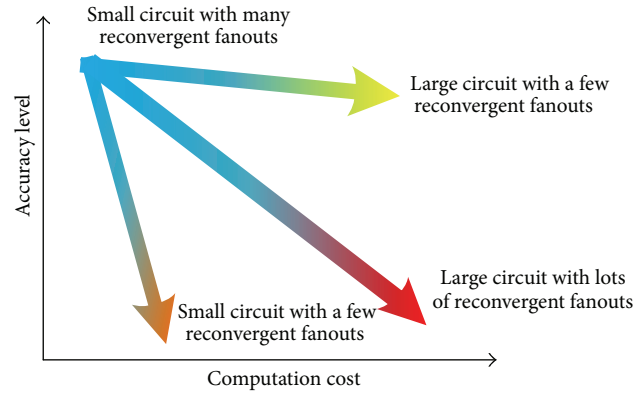


Figure 8: A general sketch of solution space for different circuit categories.

8. Comparison and Future Work

In summary, the ultimate goal of existing approaches for reliability analysis is to achieve more accurate results with as low computational cost as possible. Both accuracy and efficiency depend on specific circuit structures and their size, and, in most cases, the tradeoff between them needs to be made. The main features of each approach are described as follows.

(a) If circuits have no reconvergent fanouts (e.g., a circuit with tree structure), both signal probability and reliability can be calculated exactly with linear time (i.e., $O(N_g)$). The readers are referred to [29] for further details.

(b) For those circuits with reconvergent fanouts, the PTM method and accurate PGM model can promise exact results, while their computation costs are exponentially high. The PTM approach requires the space complexity of $O(2^{N_{\text{in}}+N_{\text{out}}})$, and the accurate PGM has the computation complexity of $O(N_g \cdot 2^{N_{\text{in}}+N_f})$. Thus, some sampling techniques are usually needed to handle large-scale circuits in these computation frameworks, leading to less accurate results.

(c) Simulation-based methods (such as MC or SCM) can provide the results with high level of accuracy, as long as enough simulation sequences are applied. To achieve a required level of accuracy, the number of simulation runs need to be determined statistically or empirically. The time complexity can be estimated by $O(N_g \cdot N_{\text{MC}})$ or $O(N_g \cdot N_{\text{SCM}})$, where N_g is again the circuit size and N_{MC} (or N_{MC}) represents the number of simulation runs. The SCM is more efficient than MC especially for small gate error rates, as the runtime of the former is approximately scaled by a constant factor.

(d) The observability-based approach has some theoretical implications, since it gives reasonable results only for circuits with extremely-low gate error rates. The maximum-k method can be viewed as the combination of CCM-based and simulation-based methods. It shows better performance than the observability-based approach in terms of both accuracy and efficiency for lower gate error rates.

(e) If all reconvergent fanouts within circuits originate from primary inputs, the simple PGM method gives exact results with the computational complexity of $O(N_g \cdot 2^{N_{\text{in}}})$. For circuits consisting of a few logic modules that are repetitively used, the modular PGM method is a good option that can provide good accuracy with short runtime.

From the above discussions, it can be concluded that errors in reliability analysis are mainly due to the reconvergent fanouts (or signal correlation) inherent in many circuits under consideration in the sense that the accurate results can be obtained efficiently for circuits with no or a few reconvergent fanouts. On the other hand, the circuit size (i.e., a large number of primary inputs or a large number of gates or both) is the main contributor to high computational costs for reliability analysis. Therefore, the most challenging problem is to analyze the reliability for large-scale circuits with a lot of reconvergent fanouts. Figure 8 illustrates the expected solution space in general in terms of accuracy and computational cost for different circuit categories. Any existing approach for the reliability analysis corresponds to a specific point in this space, which represents a tradeoff between accuracy and efficiency. For instance, the results from PTM, PGM, or MC fall into the right-upper corner of this figure with expensive computation and high level of accuracy. An ideal approach should be able to provide results somewhere near the left-upper corner where both accuracy and efficiency can be ensured.

While gate-level reliability analysis methods are well documented, there are some other important issues that remain to be tackled. First of all, most existing methods only deal with the reliability of each individual output and/or the averaged reliability over all outputs. However, the joint reliability for multiple outputs (i.e., the probability that all outputs are error-free simultaneously) is what really matters. This joint reliability could be totally different from any individual output reliability or the averaged output reliability,

depending on the possible correlation among individual output reliabilities. For an extreme case where all individual output reliabilities are independent, the joint reliability will simply be the product of all these reliabilities, which leads to a minimum value. As the correlation of output reliabilities becomes strong, the joint reliability tends to rise. In general, the complexity of computing the joint reliability would be an exponential function of the number of primary outputs. It is still an open question how to estimate the joint reliability for multiple-output circuits in an efficient way. Secondly, most of the current reliability analysis frameworks assume that the reliability for an error-free output being "0" (denoted by r_0) is the same as that for an error-free output being "1" (denoted by r_1). This is the so-called symmetric reliability model. However, this assumption does not always hold true in the real world. Thus, an asymmetric reliability model (where $r_0 \neq r_1$) would make more sense for better estimation of reliability. This requires further research work that can take the asymmetric model into consideration. Finally, there is also plenty of room for gate-level reliability improvement using reliability-critical gates as well as considering other performance metrics (such as circuit area and delay and power consumption). Unfortunately, to the best of authors' knowledge, little or limited study has been done so far in this regard.

9. Conclusion

We have reviewed the state-of-the-art methods for reliability analysis and shown their advantages and disadvantages. Some of these methods have been implemented on benchmark circuit examples to compare their performance in terms of accuracy and efficiency. While these methods seem to be effective for some specific cases/circuits, no single one of them stands out as an all-time winner due to the nature and complexity of the reliability analysis problem. Further work has also been suggested for the future research in this area.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] S. Borkar, "Designing reliable systems from unreliable components: The challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, 2005.

[2] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang, and F. Lombardi, "A stochastic computational approach for accurate and efficient reliability evaluation," *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1336–1350, 2014.

[3] S. Krishnaswamy, S. M. Plaza, I. L. Markov, and J. P. Hayes, "Signature-based SER analysis and design of logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 1, pp. 74–86, 2009.

[4] C. Chen and Y. Mao, "A statistical reliability model for single-electron threshold logic," *IEEE Transactions on Electron Devices*, vol. 55, no. 6, pp. 1547–1553, 2008.

[5] C. Chen, "Reliability-driven gate replication for nanometer-scale digital logic," *IEEE Transactions on Nanotechnology*, vol. 6, no. 3, pp. 303–308, 2007.

[6] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies*, C. E. Shannon and J. McCarthy, Eds., pp. 43–98, Princeton University Press, Princeton, NJ, USA, 1956.

[7] J. Han and P. Jonker, "A system architecture solution for unreliable nanoelectronic devices," *IEEE Transactions on Nanotechnology*, vol. 1, no. 4, pp. 201–208, 2002.

[8] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Estimate of signal probability in combinational logic networks," in *Proceedings of the 1st European Test Conference*, pp. 132–138, Paris, France, April 1989.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, Calif, USA, 1979.

[10] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Accurate reliability evaluation and enhancement via probabilistic transfer matrices," in *Proceedings of the Design, Automation and Test in Europe*, vol. 1, pp. 282–287, March 2005.

[11] S. Krishnaswamy, G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Probabilistic transfer matrices in symbolic reliability analysis of logic circuits," *ACM Transactions on Design Automation of Electronic Systems*, vol. 13, no. 1, article 8, 2008.

[12] W. Ibrahim, V. Beiu, and M. H. Sulieman, "On the reliability of majority gates full adders," *IEEE Transactions on Nanotechnology*, vol. 7, no. 1, pp. 56–67, 2008.

[13] T. Rejimon, K. Lingasubramanian, and S. Bhanja, "Probabilistic error modeling for nano-domain logic circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 55–65, 2009.

[14] T. Rejimon and S. Bhanja, "Scalable probabilistic computing models using Bayesian networks," in *Proceedings of the IEEE International 48th Midwest Symposium on Circuits and Systems (MWSCAS '05)*, pp. 712–715, August 2005.

[15] J. T. Flaquer, J. M. Daveau, L. Naviner, and P. Roche, "Fast reliability analysis of combinatorial logic circuits using conditional probabilities," *Microelectronics Reliability*, vol. 50, no. 9–11, pp. 1215–1218, 2010.

[16] R. I. Bahar, J. Chen, and J. Mundy, "A probabilistic-based design for nanoscale computation," in *Nano, Quantum and Molecular Computing: Implications to High Level Design and Validation*, S. Shukla and R. I. Bahar, Eds., chapter 5, Kluwer Academic, Norwell, Mass, USA, 2004.

[17] R. I. Bahar, J. Mundy, and J. Chen, "A probability-based design methodology for nanoscale computation," in *Proceedings of the International Conference on Computer-Aided Design*, pp. 480–486, November 2003.

[18] A. R. Kermany, N. H. Hamid, and Z. A. Burhanudin, "A study of MRF-based circuit implementation," in *Proceedings of the International Conference on Electronic Design (ICED '08)*, pp. 1–4, December 2008.

[19] D. Bhaduri and S. Shukla, "NANOLAB—a tool for evaluating reliability of defect-tolerant nanoarchitectures," *IEEE Transactions on Nanotechnology*, vol. 4, no. 4, pp. 381–394, 2005.

[20] X. Lu, J. Li, and W. Zhang, "On the probabilistic characterization of nano-based circuits," *IEEE Transactions on Nanotechnology*, vol. 8, no. 2, pp. 258–259, 2009.

[21] H. Chen and J. Han, "Stochastic computational models for accurate reliability evaluation of logic circuits," in *Proceedings*

of the 20th Great Lakes Symposium on VLSI (GLSVLSI '10), pp. 61–66, May 2010.

[22] J. B. Gao, Y. Qi, and J. A. B. Fortes, "Bifurcations and fundamental error bounds for fault-tolerant computations," *IEEE Transactions on Nanotechnology*, vol. 4, no. 4, pp. 395–402, 2005.

[23] J. Han, E. Taylor, J. Gao, and J. Fortes, "Faults, error bounds and reliability of nanoelectronic circuits," in *Proceedings of the IEEE 16th International Conference on Application-Specific Systems, Architectures, and Processors (ASAP '05)*, pp. 247–253, July 2005.

[24] J. Han, H. Chen, E. Boykin, and J. Fortes, "Reliability evaluation of logic circuits using probabilistic gate models," *Microelectronics Reliability*, vol. 51, no. 2, pp. 468–476, 2011.

[25] J. Han, E. R. Boykin, H. Chen, J. H. Liang, and J. A. B. Fortes, "On the reliability of computational structures using majority logic," *IEEE Transactions on Nanotechnology*, vol. 10, no. 5, pp. 1009–1022, 2011.

[26] M. R. Choudhury and K. Mohanram, "Reliability analysis of logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 3, pp. 392–405, 2009.

[27] L. Chen and M. B. Tahoori, "An efficient probability framework for error propagation and correlation estimation," in *Proceedings of the IEEE 18th International On-Line Testing Symposium (IOLTS '12)*, pp. 170–175, Sitges, Spain, June 2012.

[28] S. Ercolani, M. Favalli, M. Damiani, P. Olivo, and B. Ricco, "Testability measures in pseudorandom testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 6, pp. 794–800, 1992.

[29] N. Mohyuddin, E. Pakbaznia, and M. Pedram, "Probabilistic error propagation in logic circuits using the boolean difference calculus," in *Proceedings of the 26th IEEE International Conference on Computer Design (ICCD '08)*, pp. 7–13, October 2008.

[30] S. Sivaswamy, K. Bazargan, and M. Riedel, "Estimation and optimization of reliability of noisy digital circuits," in *Proceedings of the 10th International Symposium on Quality Electronic Design (ISQED '09)*, pp. 213–219, March 2009.

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 343960, 6 pages
http://dx.doi.org/10.1155/2014/343960

Research Article

Low-Area Wallace Multiplier

Shahzad Asif and Yinan Kong

Department of Engineering, Macquarie University, Sydney, NSW 2109, Australia

Correspondence should be addressed to Shahzad Asif; shahzad.asif@mq.edu.au

Received 18 March 2014; Accepted 23 April 2014; Published 12 May 2014

Academic Editor: Yu-Cheng Fan

Multiplication is one of the most commonly used operations in the arithmetic. Multipliers based on Wallace reduction tree provide an area-efficient strategy for high speed multiplication. A number of modifications are proposed in the literature to optimize the area of the Wallace multiplier. This paper proposed a reduced-area Wallace multiplier without compromising on the speed of the original Wallace multiplier. Designs are synthesized using Synopsys Design Compiler in 90 nm process technology. Synthesis results show that the proposed multiplier has the lowest area as compared to other tree-based multipliers. The speed of the proposed and reference multipliers is almost the same.

1. Introduction

Multiplication is one of the most widely used arithmetic operations. Due to this a wide range of multiplier architectures are reported in the literature providing flexible choices for various applications. Among them the simplest is array multiplier [1] which is also the slowest. Some high performance multipliers are presented in [2–5]. The focus of this paper is Wallace multiplier [6]. Wallace multiplier uses full adders and half adders to reduce the partial product tree to two rows, and then a final adder is used to add these two rows of partial products. We call this design "TW (traditional Wallace) multiplier" in this text. TW multiplier performs its operation in three steps. (1) Generate all the partial products. (2) The partial product tree is reduced using full adders and half adders until it is reduced to two terms. (3) Finally, a fast adder is used to add these two terms.

Waters and Swartzlander [7] presented a reduced complexity Wallace multiplier by reducing the number of half adders in the reduction process. We call this design "RCW (reduced complexity Wallace) multiplier" from now on. The speed of the RCW multiplier is expected to be the same as of TW multiplier due to the equal number of reduction stages in both multipliers. The RCW uses a larger final adder as compared to the TW multiplier. A number of strategies are reported in [8–10] to improve the speed of the RCW. However, the focus of their research is to reduce the delay by using a faster final adder while still using the same reduction tree as RCW. As a result, the final adder size for the multipliers in [8–10] is the same as that of RCW.

The focus of this paper is to optimize the reduction tree in a way that can reduce the size of the final adder. The reduced size of the final adder resulted in low area of the multiplier without incurring any extra delay. We call our design "PW (Proposed Wallace) multiplier." We also considered Dadda multiplier [11] for comparison due to its similarity with the Wallace multiplier.

This paper makes a contribution in the design of Wallace treed based multipliers by proposing a strategy to reduce the area of reduced complexity Wallace (RCW) multiplier. This innovative method allows for an effective utilization of half adders in such a way that the size of the final adder is reduced. It also provides a more regular structure of the reduction tree and the final adder.

The rest of the paper is organized as follows. Section 2 discusses some previous approaches for partial product tree reduction. In Section 3, the proposed Wallace multiplier is presented. In Section 4, the choice of final adder is discussed. Section 5 evaluates the results for all the designs synthesized in Synopsys. The work is concluded in Section 6.

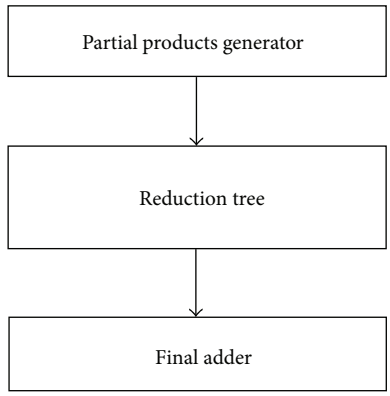


Figure 1: Block diagram of tree-based multipliers.

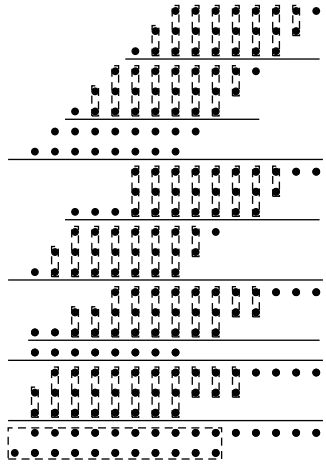


Figure 2: 8-bit traditional Wallace reduction.

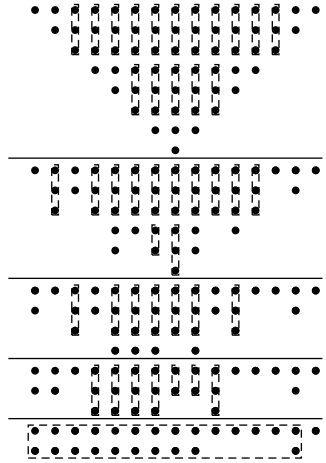


Figure 3: 8-bit reduced complexity Wallace reduction.

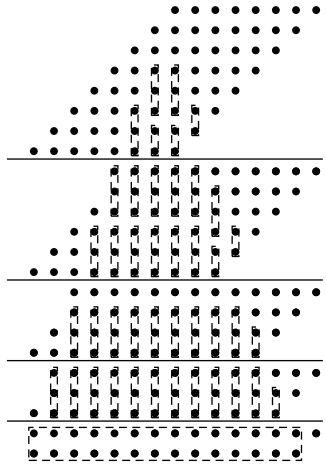


Figure 4: 8-bit Dadda reduction.

2. Previous Architectures

This section discusses some previous Wallace tree-based multiplier architectures. The general block diagram of tree-based multipliers is shown in Figure 1.

The dot notation [11] is used to represent the partial product tree in all the architectures discussed in this section as shown from Figures 2 to 5. The full adders and half adders are represented by boxes around the dot products. The box which encloses three dot products represents a full adder, whereas the box containing only two dot products is used to represent a half adder. The stages are separated by a thick horizontal line.

2.1. Traditional Wallace (TW) Multiplier. In TW multiplier architecture, the partial product tree is divided into groups [6]. Each stage can have one or more groups as shown in the 8-bit TW reduction process in Figure 2.

The groups in a stage are separated by a thin horizontal line. Each group consists of three rows except the last group where the number of rows can be less than three. Equation (1) calculates the number of rows in the last group for each stage as

$$\text{Last\_Group}_i = r_i \bmod 3. \tag{1}$$

An N-bit multiplier has N rows in the first stage. The number of rows in remaining stages can be calculated by using

$$r_i = \left\lfloor \frac{2r_{i-1}}{3} \right\rfloor + r_{i-1} \bmod 3. \tag{2}$$

Reduction is performed using a full adder or a half adder depending on the number of elements in that particular column of the group. If a column has only one element then that is passed on to the next stage without any reduction. If the last group of a stage contains less than three rows then no reduction is performed on that group as shown in stage 1 of Figure 2.

The size of the final adder for an N-bit TW multiplier with S stages can be calculated by

$$\text{FinalAdder}_{\text{TW}} = (2N - 1) - S. \tag{3}$$

Table 1: Final adder size for different multipliers.

N	Final adder size				Logic levels
	TW	RCW	Dadda	PW	
8	11	14	14	10	4
16	25	30	30	24	5
24	40	46	46	39	6
32	55	62	62	54	6
64	117	126	126	116	7

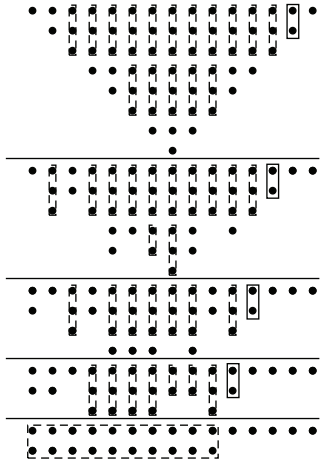


Figure 5: 8-bit proposed Wallace reduction.

2.2. Reduced Complexity Wallace (RCW) Multiplier. Waters and Swartzlander [7] presented a modification in the TW multiplier to reduce the complexity of the reduction tree. An 8-bit RCW reduction process is shown in Figure 3.

The partial products are readjusted in a reverse pyramid style which makes it easy to analyse the tree for efficient reduction. The number of stages for RCW multiplier remains the same as that of TW multiplier. RCW tries to reduce the partial product tree using only full adders. Half adders are used only where they are necessary to satisfy the number of rows in a stage according to (2). This approach allows RCW multiplier to reduce the area of the reduction process. However, RCW multiplier uses a much larger final adder as compared to TW multiplier. The size of the final adder for an N-bit RCW multiplier can be computed by

$$\text{FinalAdder}_{\text{RCW}} = 2N - 2. \tag{4}$$

2.3. Dadda Multiplier. Dadda multiplier [11] tries to reduce the number of full adders and half adders by performing the reduction only where it is essential to satisfy (2). An 8-bit Dadda reduction process is shown in Figure 4.

Dadda has the same number of stages as that of TW and RCW. We can see from Figure 4 that Dadda performed reduction only on four columns in stage 1. This is because the other columns already satisfy (2). The same approach is used in all stages to reduce the tree until we achieve a tree of only two rows. The size of the final adder is the same for Dadda and RCW multiplier as computed in (4).

3. Proposed Wallace (PW) Multiplier

In this section, we proposed a modification in the RCW multiplier to further reduce its area by reducing the size of the final adder. PW multiplier has the same number of stages and the same rule for maximum number of rows in a stage as in the other multipliers discussed in this paper.

An 8-bit PW reduction process is shown in Figure 5. PW uses an additional half adder in each stage in order to reduce the size of the final adder. The algorithm scans from the right side and starts the reduction by using a half adder when it finds the first column where the number of elements is greater than one. The additional half adders are shown in solid boxes at each stage of PW multiplier in Figure 5.

Compared with RCW in Figure 3, the introduction of half adders in Figure 5 makes the final adder in PW "less wide", namely, a smaller size. This is because, in each stage, the half adder that we introduced computes the final product bit for that particular column of the partial product tree. Therefore, the size of the required final adder is decreased by one in each stage. The least significant bit (LSB) of the product, P_0, is produced by the partial product generation block by computing $A_0 \times B_0$. In the first stage of the reduction process, product bit P_1 is computed by using the additional half adder. In the second stage, P_2 is computed. Similarly, stage 3 and stage 4 compute the product bits P_3 and P_4, respectively. Thus, when the partial product tree is reduced to two rows, five LSBs ($P_4 - P_0$) of the product are already computed as shown in Figure 5. Therefore, the size of the final adder in 8-bit PW is reduced by four as compared to the final adder in RCW multiplier. The size of the final adder for an N-bit PW multiplier with S stages can be computed by

$$\text{FinalAdder}_{\text{PW}} = (2N - 2) - S. \tag{5}$$

The comparison of (4) and (5) shows a reduction of S in the size of final adder from RCW to PW. This is achieved at the expense of an increased area for reduction process due to the insertion of additional half adders in the PW. However, the effect of additional half adders is very small as compared to the area saved by reducing the final adder size. Therefore, the overall area of the PW is less than that of the RCW.

The size of the final adder and their required logic levels for different multipliers are given in Table 1. The PW has the smallest final adder as compared to all other multipliers. All the multipliers need the same number of logic levels to implement the final adder, which means that all the multipliers will have almost the same delay. The architecture of the final adder is discussed in Section 4.

Table 2: Synthesis parameters for Synopsys Design Vision.

Technology	90 nm CMOS
Supply voltage	1.2 V
Temperature	25°C
Process model	Typical
Interconnect model	Balanced tree

4. Final Adder Design

The third step of the Wallace tree-based multipliers is to add the remaining two rows using a fast adder. Some of the most widely used parallel-prefix adders used for high speed operations are Kogge-Stone [12], Sklansky [13], and Brent-Kung [14]. These adders use the same tree topology but differ in terms of logic levels, fanout, and interconnect wires. We used Kogge-Stone adder in all the multipliers discussed in this paper. The logic levels for implementation of an N-bit Kogge-Stone adder can be calculated by using

$$\text{LogicLevels} = \lceil \log_2 (N) \rceil . \quad (6)$$

5. Results

In this section, we will discuss the verification of designs for correct operation, synthesis tool, and the results.

5.1. Functional Verification. The multipliers are implemented in VHDL with the test programs to verify the designs. All the possible input combinations are applied to thoroughly test the 8-bit multipliers. Since an exhaustive testing of bigger multipliers was not practical, they are tested with random inputs applied. The Galois-type linear feedback shift registers (LFSRs) are designed to generate pseudorandom binary sequence (PRBS) of maximum cycle for the multipliers under test [15]. All the designs are compiled and simulated using Synopsys VCS.

5.2. Synthesis Tool. All the multipliers are synthesized in Synopsys Design Compiler (DC) using 90 nm technology. The designs can be optimized for delay, power, and area by setting the appropriate options in the DC. The designer has the option of setting the various synthesis parameters such as fanout, wire load models, interconnect strategy, and PVT (process, voltage, and temperature).

The scripts are written to synthesize the TW, RCW, Dadda, and PW multipliers for optimized area. In order to have a fair comparison, the same synthesis parameters are specified for all the designs. Table 2 shows different parameters from SAED 90 nm library used for synthesis.

5.3. Synthesis Results. The detailed synthesis reports are generated by Design Compiler for area and timing. The area report includes number of cells, the area used by cells, and the interconnect area. The timing report shows the complete critical path along with the delay associated with each cell in the path. Table 3 shows the synthesis results for delay and area for different multipliers.

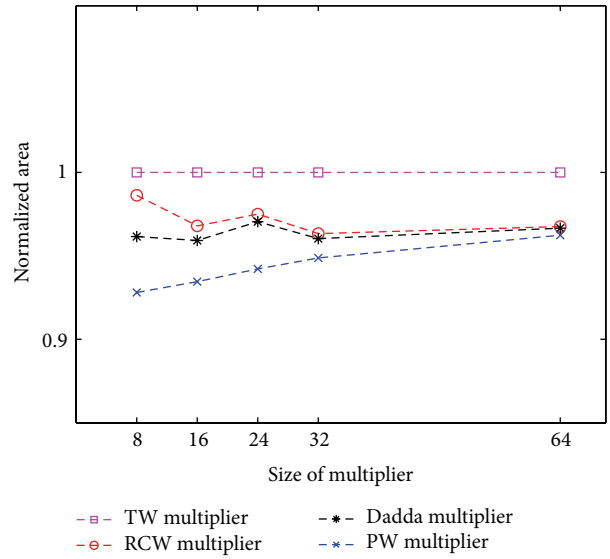


Figure 6: Area of different multipliers on Synopsys 90 nm technology.

Figure 6 shows the normalized area for each multiplier. The area of all multipliers is normalized with respect to TW multiplier by using

$$\text{Norm\_Value}_{\text{Mult\_XX}} = \frac{\text{Original\_Value}_{\text{Mult\_XX}}}{\text{Original\_Value}_{\text{TW\_Mult}}}. \quad (7)$$

It is clear from Figure 6 that the TW has the largest area as expected. Areas of RCW and Dadda are almost the same which also conforms to the results of [7]. PW has the lowest area for all the multiplier configurations. The reduction in area is more prominent when the size of the multiplier is small. As the size of the multiplier increases, the area of PW tends to asymptotically approach the area of RCW and Dadda multiplier. Therefore, the PW is particularly useful when the final adder has a significant area in the multiplier, while the area advantage might decrease in larger multipliers.

Figure 7 shows the normalized delay for each multiplier. The delays are normalized according to (7). All the multipliers use the same number of reduction stages and the same logic levels in the final adder. Therefore their delays are expected to be the same. However, the Design Compiler uses different cells to optimize the area in each design due to their different architectures and different final adder sizes. This can result in larger delays for the designs where the synthesizer can optimize the area by using relatively slower standard cells. It can be seen in Figure 7 that the PW has the least delay in 24-bit multiplier. In the rest of the multiplier sizes, PW has almost the same delay as of RCW which is less than the Dadda and TW. One exception to this is the 16-bit multiplier where PW has larger delay than RCW multiplier.

Figure 8 shows the normalized power consumption for each multiplier. The power consumption of all multipliers is normalized with respect to TW multiplier by using (7).

Table 3: Synthesis results from Synopsys DC on 90 nm technology.

Size	Delay (ns)				Area (μm^2)				Power (mW)			
	TW	RCW	Dadda	PW	TW	RCW	Dadda	PW	TW	RCW	Dadda	PW
8	2.81	2.64	2.64	2.66	3392	3346	3262	3148	1.92	2.04	1.94	1.96
16	4.13	3.65	3.80	3.75	14847	14372	14242	13876	11.09	11.41	11.22	11.20
24	4.64	4.58	4.62	4.44	34337	33479	33323	32352	27.69	28.24	28.32	28.04
32	14.83	14.62	14.82	14.62	61526	59271	59086	58375	51.85	52.74	53.11	52.48
64	22.88	21.57	21.98	21.62	246842	238843	238597	237553	216.50	219.17	222.64	218.92

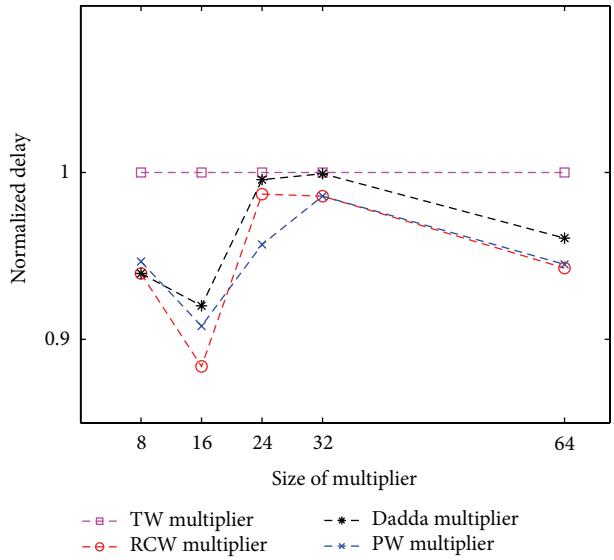


Figure 7: Delay of different multipliers on Synopsys 90 nm technology.

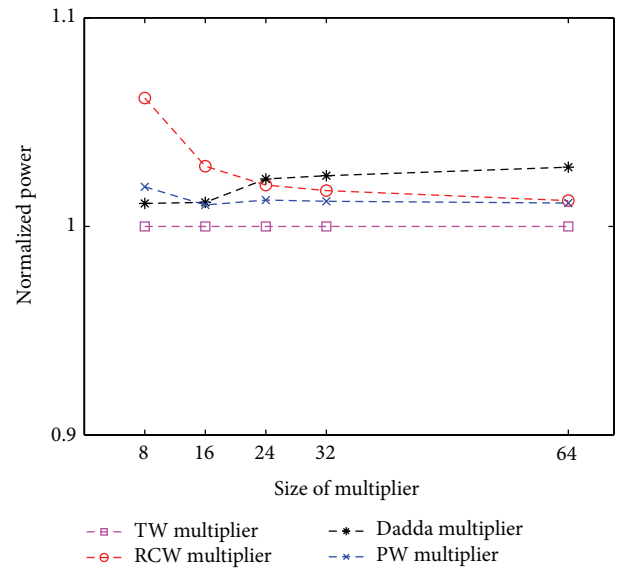


Figure 8: Power consumption of different multipliers on Synopsys 90 nm technology.

It is clear from Figure 8 that the TW multiplier has the lowest power consumption as compared to the other multipliers. One reason for this could be that the Design Compiler was able to find the low-power cells to synthesize the TW multiplier. The regular structure of TW multiplier could also be a reason of its low-power consumption. The power consumption of PW is less than that of the RCW multiplier due to the smaller final adder used in PW multiplier. It can be noted that the difference in power consumption of PW and RCW is very little for large multipliers, as expected, due to the small difference in their area.

6. Conclusion and Future Work

This paper presents a method to reduce the area of the Wallace multiplier. The proposed architecture, named as PW (proposed Wallace) multiplier, uses a smaller final adder to reduce the area of a multiplier. The designs are synthesized in Synopsys Design Compiler using 90 nm process technology. The synthesis results verify that the PW multiplier, as expected, has the smallest area as compared to the other Wallace based multipliers. The speed of the PW multiplier is almost the same as of other multipliers.

As our future work, we plan to implement the designs using Synopsys IC Compiler to analyze the postlayout results for area and delay. Synopsys Prime Time can be used to analyze the multipliers for their power consumption.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] N. H. E. Weste and D. M. Harris, *Integrated Circuit Design*, Pearson, 2010.

[2] J.-Y. Kang and J.-L. Gaudiot, "A fast and well-structured multiplier," in *Proceedings of the EUROMICRO Systems on Digital System Design (DSD '04)*, pp. 508–515, September 2004.

[3] C. R. Baugh and B. A. Wooley, "A twos complement parallel array multiplication algorithm," *IEEE Transactions on Computers*, vol. C-22, no. 12, pp. 1045–1047, 1973.

[4] S.-R. Kuang, J.-P. Wang, and C.-Y. Guo, "Modified booth multipliers with a regular partial product array," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 56, no. 5, pp. 404–408, 2009.

[5] B. C. Paul, S. Fujita, and M. Okajima, "ROM-based logic (RBL) Design: a low-power 16 bit multiplier," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 11, pp. 2935–2942, 2009.

[6] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Transactions on Electronic Computers*, vol. EC-13, no. 1, pp. 14–17, 1964.

[7] R. S. Waters and E. E. Swartzlander, "A reduced complexity wallace multiplier reduction," *IEEE Transactions on Computers*, vol. 59, no. 8, pp. 1134–1137, 2010.

[8] S. Rajaram and K. Vanithamani, "Improvement of Wallace multipliers using parallel prefix adders," in *Proceedings of the International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN '11)*, pp. 781–784, July 2011.

[9] P. Jagadeesh, S. Ravi, and K. H. Mallikarjun, "Design of high performance 64 bit mac unit," in *Proceedings of the International Conference on Circuits, Power and Computing Technologies (ICCPCT '13)*, pp. 782–786, March 2013.

[10] M. Kumaran and M. Kamarajan, "Multicore embedded system using parallel processing technique," *International Journal of Emerging Trands in Electrical and Electronics*, vol. 5, no. 3, 2013.

[11] L. Dadda, "Some schemes for parallel multipliers," *Alta Frequenza*, vol. 34, pp. 349–356, 1965.

[12] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations," *IEEE Transactions on Computers*, vol. C-22, no. 8, pp. 786–793, 1973.

[13] J. Sklansky, "Conditional-sum addition logic," *IRE Transactions on Electronic Computers*, vol. EC-9, pp. 226–231, 1960.

[14] R. P. Brent and H. T. Kung, "A regular layout for parallel adders," *IEEE Transactions on Computers*, vol. C-31, no. 3, pp. 260–264, 1982.

[15] R. Ward and T. Molteno, "Table of linear feedback shift registers," Datasheet, Department of Physics, University of Otago, 2007.

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 652187, 11 pages
http://dx.doi.org/10.1155/2014/652187

Research Article

Efficient Hardware Trojan Detection with Differential Cascade Voltage Switch Logic

Wafi Danesh, Jaya Dofe, and Qiaoyan Yu

Department of Electrical and Computer Engineering, University of New Hampshire, Durham, NH 03824, USA

Correspondence should be addressed to Qiaoyan Yu; qiaoyan.yu@unh.edu

Received 26 February 2014; Accepted 7 April 2014; Published 11 May 2014

Academic Editor: Chih-Cheng Lu

Offshore fabrication, assembling and packaging challenge chip security, as original chip designs may be tampered by malicious insertions, known as hardware Trojans (HTs). HT detection is imperative to guarantee the chip performance and safety. Existing HT detection methods have limited capability to detect small-scale HTs and are further challenged by the increased process variation. To increase HT detection sensitivity and reduce chip authorization time, we propose to exploit the inherent feature of differential cascade voltage switch logic (DCVSL) to detect HTs at runtime. In normal operation, a system implemented with DCVSL always produces complementary logic values in internal nets and final outputs. Noncomplementary values on inputs and internal nets in DCVSL systems potentially result in abnormal power behavior and even system failures. By examining special power characteristics of DCVSL systems upon HT insertion, we can detect HTs, even if the HT size is small. Simulation results show that the proposed method achieves up to 100% HT detection rate. The evaluation on ISCAS benchmark circuits shows that the proposed method obtains a HT detection rate in the range of 66% to 98%.

1. Introduction

The growing number of ICs manufactured offshore increases the threats to chip security [1–3]. Research has exposed an increase in existence of hardware Trojans (HTs), which are malicious additions or modifications to the circuit design that alter the original function. Malicious inclusions of hardware have the potential to degrade system performance, surreptitiously delete data, leave a backdoor for secret key leaking, or eventually destroy the chip [4, 5]. It is imperative to detect HTs.

HTs can be detected by destructive approaches such as the chemical mechanical polishing (CMP) method. The CMP approach detects HTs by analyzing pictures of the demetalized chips under an electron microscope [6]. In addition to being expensive, this type of technique is also time consuming (takes several months) and loses its efficiency when the transistor density increases. Nondestructive HT detection methods are broadly classified into two categories: logic testing and side-channel analysis (SCA) approaches [6]. Automatic test pattern generation (ATPG) approaches examine whether the measured outputs match the expected one for given inputs and work well for a functional unit with a small set of inputs, as the probability of rare events is relatively high. When the circuit complexity increases, the number of test vectors for ATPG will significantly increase to an unaffordable degree. The benefits relative to the testing efforts of ATPG become worse if the input nodes for the HT's trigger circuit are spread out throughout the system. The main challenge with logic testing approaches [7, 8] is the generation of stimulus for sequential HTs. Voltage inversion technique alternates supply voltage and ground grids in CMOS-based functional blocks to change the original logic function and thus increases the HT trigger probability [9]. Dummy flip-flops are inserted into the design to increase transition probability of particular paths and reduce Trojan activation time [10]. Alkabani [11] introduces the concept of creating dual circuits for a given design. By testing the dual with a few random input vectors, a HT inserted in the original design can be detected.

SCA approaches examine the anomalous behavior (resulting from HTs) in system parameters such as transient current, power, and path delay [12–15]. A multiple-parameter

side-channel analysis method and a platform are developed to reliably test, analyze, and detect a wide range of HTs for both combinational and sequential designs [14]. Recently, HT detection approaches rely on multiple-parameter side-channel analysis technique, which can be integrated with statistical logic testing in order to improve the detection of HTs with very small design area [16]. SCA-based methods [17, 18] achieve a high coverage and are effective for finding HTs that span a large area of a system. However, the sensitivity of SCA-based methods is challenged by the increasing process variation [16, 19]. False detection on small HTs can happen when process variation effects exceed the signal threshold (e.g., power) for side-channel analysis.

To address the challenge from process variation on HT detection, region-based approach [4] magnifies the region potentially affected by HTs and forces the remaining regions to be inactive. Postsilicon spatial thermal and power maps are simultaneously utilized in a multimodal characterization procedure to improve the HT detection sensitivity [13]. A unified framework combines different HT detection methods in a systematic analysis platform, which studies the impact of small HTs [20].

In this work, we propose a method to remove the need of golden design for comparison and detect small HTs at runtime. The difference with other side-channel analysis approaches is that our method focuses on enhancing the side-channel signals by using a logic family's inherent characteristics. We exploit the special characteristic of differential cascade voltage switch logic (DCVSL) to detect HTs. Trojan detection using DCVSL can be performed using the constant, abnormal power consumption peaks, or erroneous outputs. The method is inexpensive as there is no extra hardware overhead required in order to implement the HT detection platform. Simulation results show that the power consumption of a DCVSL system with a HT triggered is constantly three orders of magnitude higher than that of the system with inactive HTs. This unique, abnormal power consumption phenomenon complements existing power-based side-channel analysis methods.

The remainder of this work is organized as follows. In Section 2, we highlight the basis for abnormal power consumption in DCVSL and introduce the proposed HT detection method. In Section 3, we thoroughly evaluate the area, power, and HT detection rate of our method in full adders and ISCAS benchmark circuits. Conclusions and future work are provided in Section 4.

2. Proposed DCVSL-Based HT Detection Method

2.1. Method Overview. HT detection is typically carried out during test stages, when numerous test vectors are simultaneously applied to both the device under test (DUT) and a golden reference. As it is difficult to obtain a golden version, the behavioral model is often used as a reference. Because of the effects of process variation and imperfect device libraries for computer-aided design tools, a behavioral model based golden version is not precise enough to detect small and ultrasmall HTs. Moreover, due to the demand of short time-to-market, the verification and testing period has been reduced significantly. Although it is always desired, thorough testing is not economically feasible. It is imperative to develop a HT detection method that is not limited by the HT size and does not take very long time to perform chip testing and authorization.

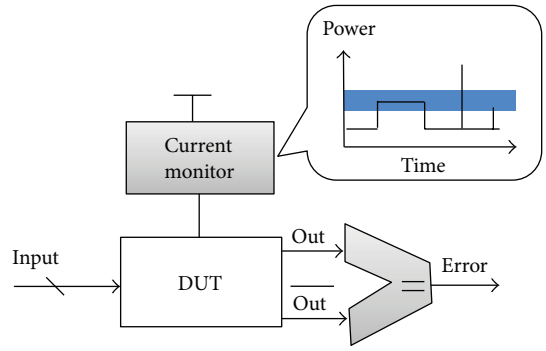


Figure 1: Proposed HT detection system.

We propose a HT detection method that allows users to detect potential HTs *at runtime* and *without a golden reference*. The proposed method exploits the abnormal instantaneous power for DUT to detect small HTs. Figure 1 shows the overview of the proposed method. Given a stable supply voltage, we examine the current through the current monitor for abnormal power behavior. A notable difference with offline power-based side-channel analysis methods is that we are not interested in a particular power value; instead, the current monitor detects the current (we can interpret it to power consumption) staying at a constant high value for a relatively long duration. As shown in Figure 1, the current monitor will trigger an alarm circuit, when the power value falls in and remains in the blue shadow region for a relatively long period. This duration is comparable with the duration of an input vector, rather than input rising and fall times. The triggered HT in the DUT causes the abnormal power period. We propose to implement DUTs with DCVSL, which always produces Out and Out bar, a pair of complementary outputs. Such complementary outputs will be used as inputs for next stage. In DCVSL, noncomplementary inputs (invalid inputs) result in short-circuit power remaining for a long period of time until the noncomplementary inputs disappear.The proposed method exploits this inherent feature of DCVSL to detect the presence of HTs.

Besides power detection, the proposed method further examines the complementary characteristic of the output pair, Out and Out bar. The noncomplementary output pair indicates a potential hardware Trojan insertion in the DUT. These noncomplementary outputs can be utilized for HT detection when no abnormal power values appear due to the HT being triggered.

The current monitor is connected with the DUT on a separate platform at the user end. If the current monitor is integrated on the same chip with the DUT, this potentially leaves an opportunity for an attacker to tamper or remove

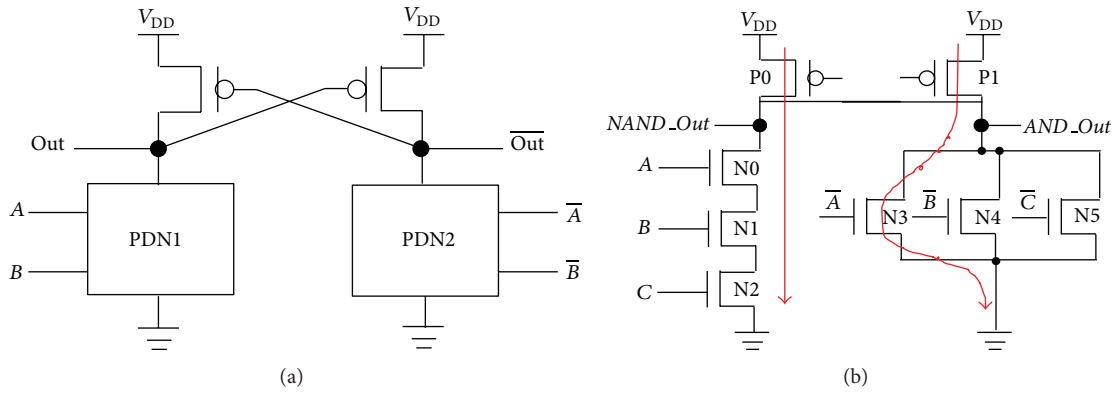


FIGURE 2: DCVSL logic gates. (a) General gate structure and (b) circuit schematic of NAND3-AND3. Current track highlighted in the figure is for noncomplementary inputs on A and $\overline{A}$.

the HT detection mechanism. A current sensor is needed to convert the transient current of the DUT and produce an analog voltage that is proportional to the measured DUT current. A programmed microcontroller can sample the analog voltage signal at specific intervals using interrupts. When the voltage value stays approximately constant for multiple interrupts, it indicates an abnormal short-circuit power due to a HT creating a short-circuit path from supply voltage V_{DD} to ground. The microcontroller can be further configured to set off an alarm or trigger a light-emitting diode to indicate HT detection to the user.

2.2. Short-Circuit Power-Based HT Detection

2.2.1. Unique Short-Circuit Power in DCVSL. Each DCVSL gate needs complementary inputs and produces complementary outputs [21], as shown in Figure 2(a). In normal operation, short-circuit power consumption of DCVSL gate is close to that of CMOS logic gate, as the time period for the direct current path from V_{DD} to ground is extremely short compared with that in switching and steady state conditions.

When the input pair is noncomplementary (both inputs being either logic 0 or logic 1), a DCVSL gate loses its complementary nature. More specifically, the output pair may be noncomplementary, resulting in the short-circuit power consumption lasting for a significantly longer time than the case with complementary inputs.

Take a 3-input NAND-AND gate as an example. The circuit schematic is shown in Figure 2(b). In normal operation conditions, we give the input vector of $A = B = C = 1$ and $\overline{A} = \overline{B} = \overline{C} = 0$. The *NAND_Out* port is pulled down to logic low through NMOS transistors N0, N1, and N2; this in turn activates PMOS transistor P1. As P1 is turned on, the *AND_Out* node is pulled to logic high and thus P0 is turned off. The time period when both PMOS and NMOS transistors are on is extremely short. Let us reconsider the 3-input NAND-AND gate with the same input vector, except that we make $A = \overline{A} = 1$. Now, there exist two paths from V_{DD} to the ground terminal: one is through N0, N1, and N2 and another one is through N3. The path through N0, N1, and N2 pulls the *NAND_Out* port low as before, which turns on P1. P1 then tries to pull the *AND_Out* port high. At the same time, the path to ground through N3 tries to pull the *AND_Out* node low. If N3 is stronger than P1 (which is typically the case), the *AND_Out* port is pulled low and this activates P0. Therefore, a path from V_{DD} to ground is created through P0, N0, N1, and N2, resulting in a *high* and *constant* short-circuit power. The constant short-circuit power remains as long as the duration of the input vector.

Figures 3(a) and 3(b) show the power waveforms with complementary and noncomplementary inputs, respectively. As shown in Figure 3(b), in the duration of the input vector $A = \overline{A} = B = C = 1$ (from 7 to 8 μs on the time axis), the peak power has a constant high value. This is because the noncomplementary input pair ($A = \overline{A}$) makes *NAND_Out* and *AND_Out* both stay at logic low. The time from 7 to 8 μs represents the high time of the shortest input pulse A. As a result, the two PMOS transistors, P0 and P1, are both turned on; thus, the two current paths from V_{DD} to ground (highlighted in Figure 2(b)) exist till the input vector is changed. The amplitude of short-circuit power is typically three orders of magnitude higher than the leakage power. This significant power difference between the cases using complementary and noncomplementary inputs is large enough for a monitoring device to indicate the presence of a HT.

We examine the average power for complementary and noncomplementary inputs for basic DCVSL gates using a typical IBM7RF technology library. As shown in Table 1, the increase on the average power (averaging power for all possible input patterns) caused by noncomplementary inputs is over three orders of magnitude. This is the basis for choosing DCVSL to implement functional units that facilitate HT detection. If the triggered HT flips the internal node of a functional unit, it will create a noncomplementary signal in the middle of that functional unit. Consequently, the power consumption will stay high for a long time, which is different from normal switching power.

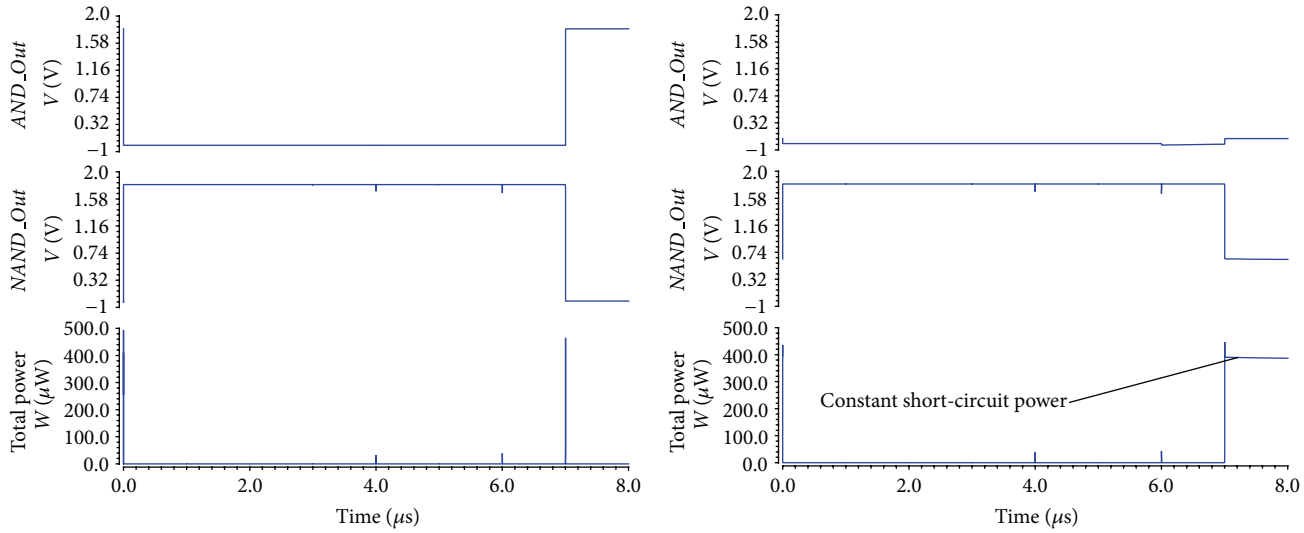


Figure 3: Voltage and power waveforms for DCVSL NAND3-AND3 gate. (a) Complementary inputs and (b) noncomplementary input $A = \overline{A}$.

Table 1: Power increase caused by noncomplementary inputs.

Logic gates	Average power	
	Power for complementary inputs	Power for noncomplementary inputs
Inverter	20.51 nW	205.25 μW
NAND2-AND2	12.76 nW	92.84 μW
NOR2-OR2	11.85 nW	92.61 μW
XNOR2-XOR2	19.97 nW	171.2 μW
NAND3-AND3	7.501 nW	40.36 μW
NOR3-OR3	6.673 nW	39.81 μW
XNOR3-XOR3	16.49 nW	84.90 μW
D-Flip-Flop	17.23 nW	181.8 μW

2.2.2. Probability of Abnormal Short-Circuit Power. The key reason for DCVSL gate having abnormal short-circuit power is the noncomplementary output nodes turning on the two PMOS transistors simultaneously. We assume that the consequence of a HT insertion on DCVSL functional units is flipping one of the complementary inputs. This is similar to HT insertion in other technologies; that is, a triggered HT is used to change the logic value of a logic gate or memory element.

Because of electrical and logical masking, the noncomplementary inputs (caused by HTs) do not always yield abnormal short-circuit power. As the logic gate topology varies between gates, it is difficult to obtain a closed-form expression for the probability of abnormal power occurrence. We summarize the general procedure for how to analyze the HT detection probability in DCVSL systems through abnormal power observation. Figure 4 is the flowchart for the analysis procedure.

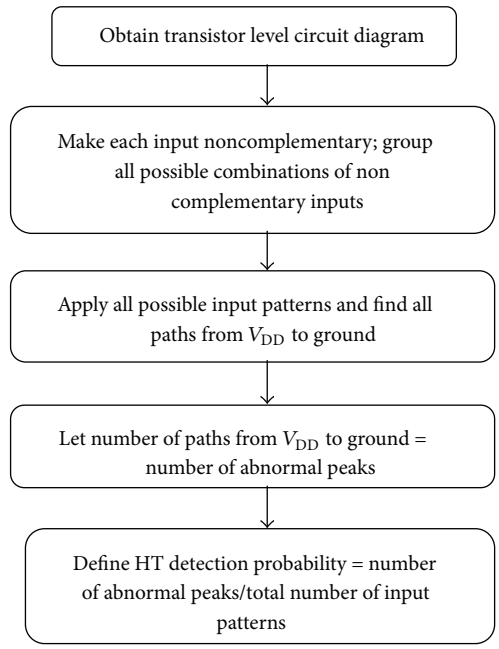


Figure 4: Flowchart for analyzing the HT detection probability for a DCVSL gate.

In order to create an erroneous output in DCVSL, a HT has to make one or more of the inputs noncomplementary. This may result in an erroneous output if the effect of the noncomplementary input is propagated and reaches the output port. An important point to note is that not all erroneous outputs are accompanied by abnormal power peaks. Only if the erroneous output creates at least one path from V_{DD} to ground, will we observe the abnormal short-circuit power.

Table 2: Probability of abnormal power and output error rate over all possible input patterns for DCVSL logic gates.

DCVSL gates	Percentage of abnormal power over all input patterns	Percentage of output error over all input patterns
Inverter	50.00%	100%
XOR2	41.66%	100%
XOR3	33.92%	100%
AND2	25.00%	25.00%
AND3	12.50%	12.50%
OR2	25.00%	50.00%
OR3	12.50%	25.00%
OAI21	23.21%	28.57%
AOI21	26.78%	50.00%
AOI22	25.89%	45.08%
OA22	25.89%	35.27%
MUX21	30.35%	55.00%
Average	27.72%	52.00%

We examine the probability of abnormal power and output error occurrence for all input patterns. Table 2 shows the ratio of the total number of abnormal power peaks over the total number of all input patterns for various basic DCVSL gates. The average probability for power exception and output mismatch are 27.7% and 52%, respectively. This means our HT detection method has over 50% chance to detect HTs, even if the HT trigger circuit is implemented with a single gate. This is a significant advantage over other power-based side-channel analysis methods, which have a lower bound on the size of detectable HTs.

Moreover, we observe that abnormal power occurs more often on the input pattern that produces the rare output value. For example, an AND3 gate produces high output only when all three inputs are high; the abnormal power appears at the exact input pattern if one of the inputs is not in the complementary form. To hide a HT, hackers often utilize the rare case to trigger the HT. As discussed above, our approach inherently achieves a higher detection rate for the HT triggered by rare cases. This means a system equipped by our method will pose a greater challenge to attackers in order to conceal HTs.

3. Experimental Results

3.1. Experimental Setup. We evaluated the proposed method on the 64-bit ripple carry adder, ISCAS'85 and ISCAS'89 benchmark circuits. The schematic and layout of the 64-bit adder were implemented in Cadence Virtuoso with the IBM CMOS7RF technology. We set all transistor lengths to 220 nm (minimum length in the CMOS7RF technology) and set the PMOS and NMOS transistor widths to 500 nm and 600 nm, respectively. The average power, leakage power, and peak dynamic power were obtained from schematic-level simulations by examining all possible input patterns. The area for DCVSL modules was obtained from customized layout in Virtuoso. Five metal layers were used in layout design. The fastest switching period for input is 1 μs. We synthesized the Verilog codes of ISCAS benchmark circuits in Synopsys Design Compiler with IBM CMOS7RF technology. The synthesized netlist is modified with an in-house python-based netlist generator, which converts CMOS netlist to DCVSL netlist. The behavior model of CMOS library is modified according to the gate output and power performance obtained from simulation in Cadence Virtuoso.

Table 3: Number of transistors for DUTs and HTs in this work.

Circuit	CMOS 64-bit adder	HT-1	HT-2	HT-3
Transistor number	2560	8	28	100
Circuits	DCVSL 64-bit Adder	C432	C1908	C3540
Transistor number	1644	2070	5516	9874
Circuits	S526	S832	S1196	S1488
Transistor number	1682	1408	3056	2824

Table 4: Power consumption for two 64-bit full adders and HT insertions.

Unit under test		Dynamic power (mW)	Leakage power (nW)
CMOS-based 64-bit full adder	Adder	24.6	65.78
	HT-1	0.444	0.419
	HT-2	0.942	1.243
	HT-3	1.028	2.583
DCVSL-based 64-bit full adder	Adder	8.002	47.50
	HT-1	0.566	0.328
	HT-2	0.892	0.993
	HT-3	1.544	2.465

HT detection rate is evaluated through gate-level simulation in Cadence NCVerilog. To observe the accumulated HT-induced effects through the system, we inserted the HTs payload on the inputs of DUTs. We particularly did so to model the propagation of HT effect in a large-scale system. To compare the area and power consumption of DUT and HTs, we designed three HTs. HT-1 is OR3 trigger circuit with XOR2 payload. HT-2 is OR(XOR(AND(x,y),z),w) trigger circuit with XOR2 payload. HT-3 is AND4 plus modulo-8 counter trigger circuit with XOR2 payload. The complexity of the DUTs and HTs in this work is listed in Table 3. As can be seen, the HTs are significantly smaller than the target design.

3.2. Case Study on a 64-Bit Full Adder. We implemented a 64-bit full adder using CMOS and DCVSL in Cadence Virtuoso. The layout area for these two adders is shown in Table 3. Because less PMOS transistors are needed in DCVSL, the area of DCVSL-based full adder is less than that of CMOS full adder when optimization is applied on both implementations. HTs are rarely triggered and the leakage power for HTs is a few orders of magnitude less than the adder switching power, as shown in Table 4.

All possible input patterns were applied to the 64-bit ripple carry adder. We placed a HT circuit to alter one complementary input pin in the adder. The power over time waveform is shown in Figure 5. As can be seen in

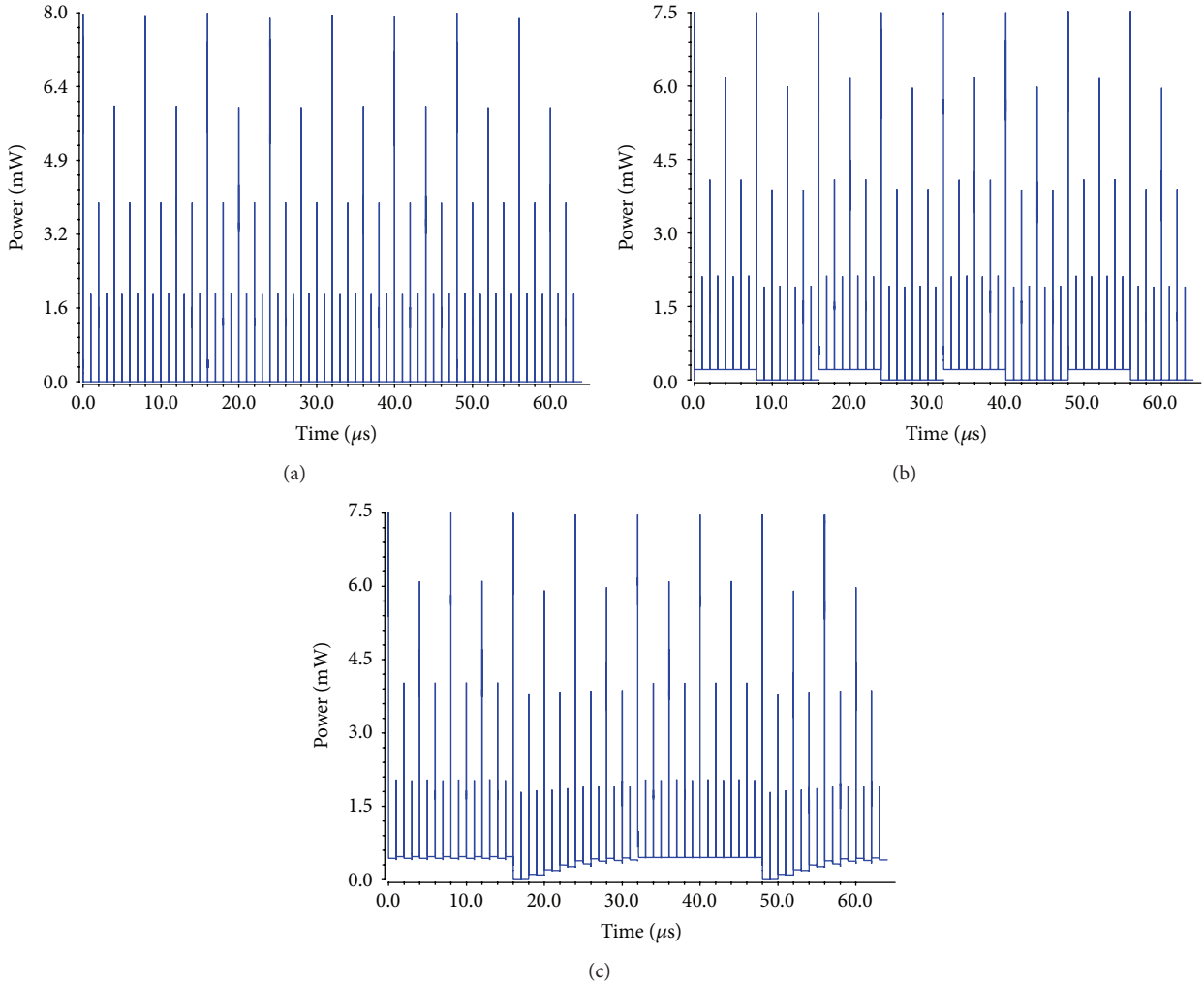


Figure 5: Power consumption for a 64-bit DCVSL full adder. (a) No HT, (b) HT on the 49th 1-bit full adder carry in port, and (c) HT on the 2nd 1-bit full adder carry in port.

Figure 5(a), when no HT is triggered, the switching power has instantaneous peaks whereas the leakage power remains flat (close to zero). Figure 5(b) shows the power for the adder with one HT inserted at the 49th 1-bit full adder. As can be seen, the power has an extra periodical increase, which is noticeably higher than the leakage power. This is the short-circuit power (discussed in Section 2.2) induced by the noncomplementary inputs from HT insertion. We placed the HT payload circuit to the 2nd 1-bit full adder and observed different power behavior. As shown in Figure 5(c), the increased short-circuit power appears in almost all input patterns. This is because the 2nd 1-bit full adder with noncomplementary inputs yields noncomplementary outputs, and those outputs are further propagated to other 1-bit full adders. Because of the propagation of HT effects, the power consumption is exceptionally higher than that in normal cases.

CMOS circuits have more PMOS transistors than the DCVSL version. Consequently, the dynamic power consumption of CMOS is higher than that of DCVSL. As shown in Figure 6, DCVSL has less average power consumption than CMOS. However, when the HT is triggered to change the noncomplementary inputs for the DCVSL-based full adder, the increased short-circuit power results in a dramatic increase on the average power. Figure 6 also shows that the average power difference between original and HT affected version is over 50X. If the HT is inserted at the early stage in the functional block, the average power difference increases to over two orders of magnitude. This is favorable for power-based side-channel analysis HT detection methods.

To assess the HT detection rate, we assume that HTs are inserted to change the complementary inputs. As input vectors A and B for a 64-bit full adder are equivalent, we select 64-bit input A to receive the potential impact from HTs. Besides half of the inputs, A, the carry-in bit for the first 1-bit full adder is another potential location for HT insertion. As the proposed method is independent of the particular HT trigger circuit, we flipped one of the complementary inputs to model the effect of HT insertion. As shown in Figure 7, for the HTs on A, the HT detection rate reaches 1. Given a HT area over chip area ratio below 1%, the HT detection rate is higher

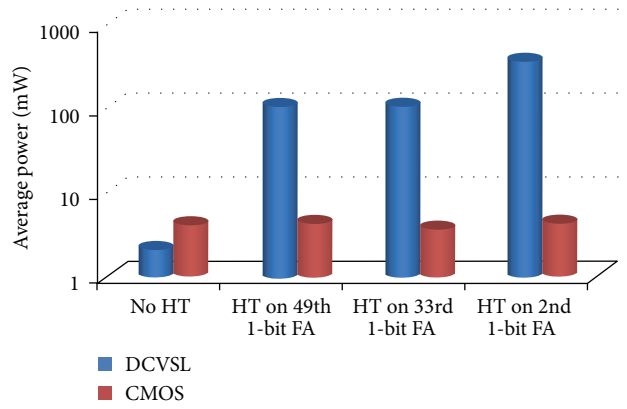


Figure 6: Impact of HT location on average power of 64-bit DCVSL adder.

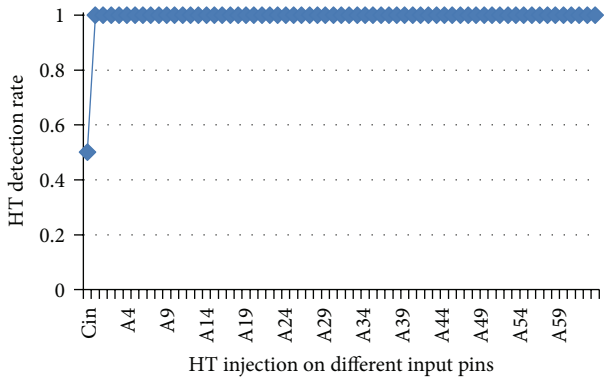


Figure 7: Impact of HT insertion locations on HT detection rate.

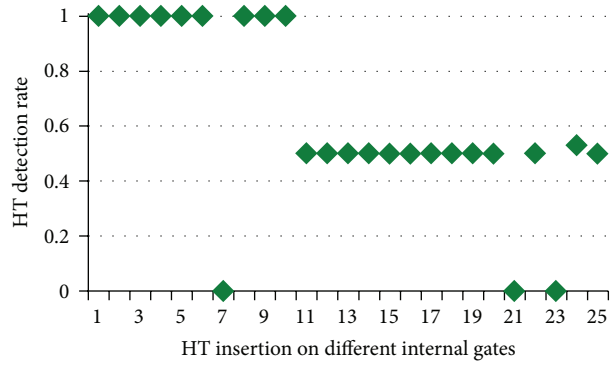


Figure 8: Impact of HT insertion location on HT detection rate.

than the one reported in [13]. Such high HT detection rate is mainly contributed by the noncomplementary inputs, which lead to internal noncomplementary outputs. Those outputs are further propagated to the remaining gates. Consequently, one HT injection possibly leads to more gate failures. Figure 7 also shows that the HT inserted on the carry-in (Cin) input can be detected with a HT detection rate of 0.5, which can be compensated by comparing outputs. Our simulation results show that, after the output comparison, the HT detection rate can be enhanced close to 1. The simulated HT detection rate was obtained from 200,000 random input patterns.

HTs placed on input pins at earlier stages in the design have higher potential to be detected, because of the propagation of noncomplementary outputs. We examine the impact of HT insertion locations on the HT detection rate. As shown in Figure 8, as the HT insertion location shifts towards the final output, the HT detection rate decreases to around 0.5.

The earlier the HT is inserted, the higher the probability of obtaining abnormal power behavior which can be used to determine the presence of HTs will be. For HT injection on the very early inputs, each HT detected case will have about 1.7 gates experiencing high short-circuit power, as shown in Figure 9(a). According to Tables 1 and 4, the short-circuit power for one gate is one order of magnitude higher than the leakage power of a full adder. Therefore, the power difference is high enough for use in HT detection. As shown in Figure 9(b), the HT inserted in the early 1-bit full adder stage yields an abnormal energy that is up to three orders of magnitude higher than normal leakage energy. HT insertion location approaching the final output yields less abnormal power, in terms of absolute energy value and the frequency of abnormal energy. As explained before, the latter HT injection location has a higher probability to demonstrate errors on the final outputs.

3.3. Evaluation on Benchmark Circuits. The proposed method is further evaluated with ISCAS benchmark circuits, which are composed of various logic gates listed in Table 2. In the experiments below, we assume that single HT is inserted in the benchmark circuit. More HT insertions in the target circuit lead to a higher HT detection rate, as more gates experience abnormal short-circuit power. The HT detection rate is defined as the number of cases experiencing abnormal short-circuit power over the total number of test cases. Three combinational benchmark circuits, c432, c1908, and c3540, are used to assess the HT detection rate of our method. 500,000 random input patterns were applied to the evaluation of c432 and c1908 circuits. Because of larger scale, c3540 was evaluated with 1,000,000 random input patterns.

As shown in Figure 10, our method achieves the HT detection rate up to 1 in the c432 circuit. The lowest HT detection rate is 0.7333. The majority of logic gates in c432 are Inverter and AND2; thus the HT rates are centered around two particular regions, 1 and 0.73. The scales of c1908 and c3540 are larger than c432; the kind of logic gates in c1908 and c3540 is more diverse than c432. These two factors affect the HT detection rate. Figures 11 and 12 show that the HT detection rate is distributed over the whole range, but the HT detection rate stays mostly above 0.7. We averaged the HT detection rate over all test cases in Figure 13. As can be seen, our method achieves a HT detection rate over 0.8 in c432 and c1908. The HT detection rate for c3540 is slightly low; however, our HT detection rate is still significant, as our method is not limited by the size of HTs and can be used to detect extremely small HTs. The average HT detection rate for the examined ISCAS'85 benchmark circuits is 0.76.

To examine the amount of power increased by each HT insertion, we first investigated the number of gates having

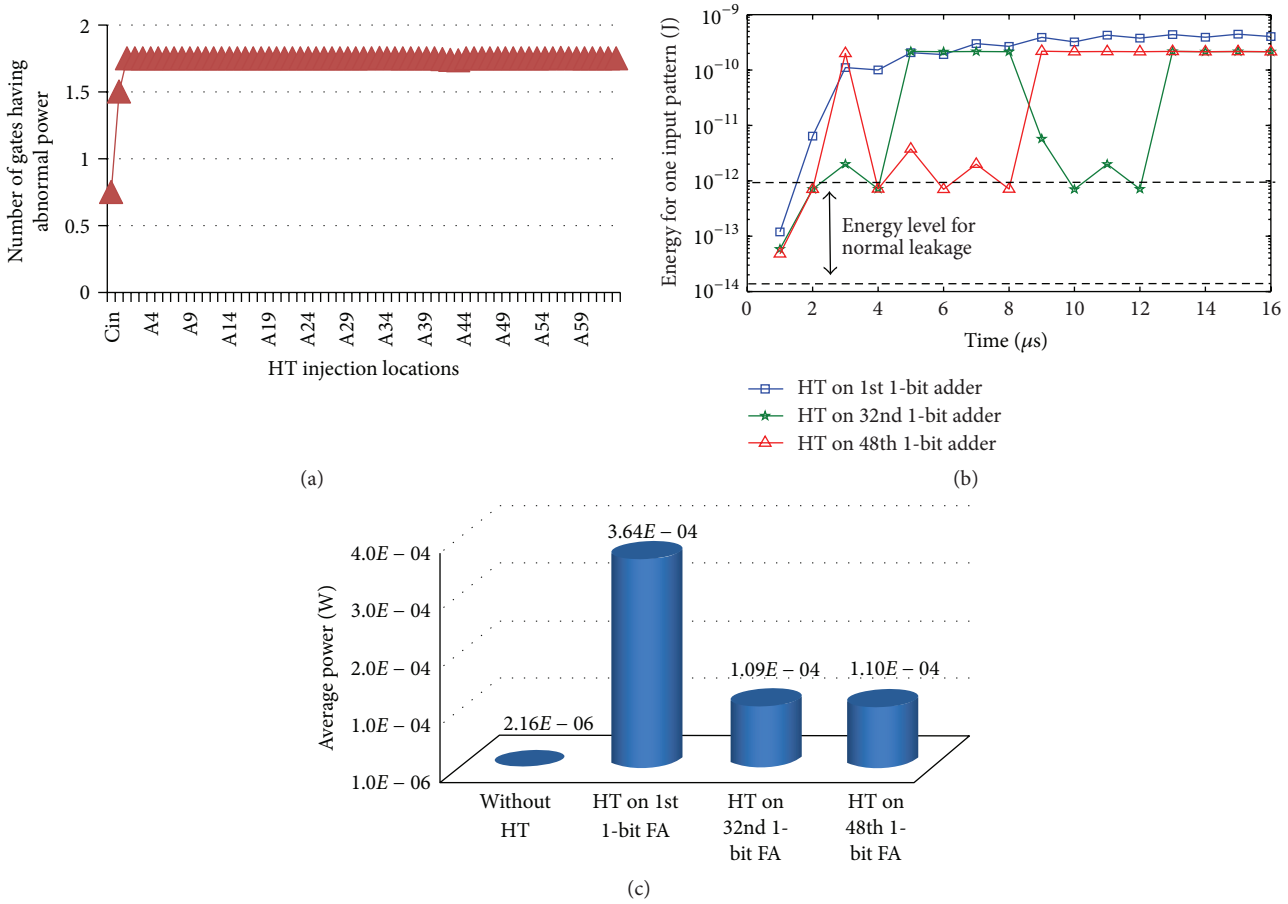


FIGURE 9: Results for HT-induced abnormal power assessment. (a) Average number of gates experiencing high short-circuit power per HT inserted case. (b) Abnormal energy caused by HT insertion over regular leakage energy. (c) Average power for three different HT injection locations.

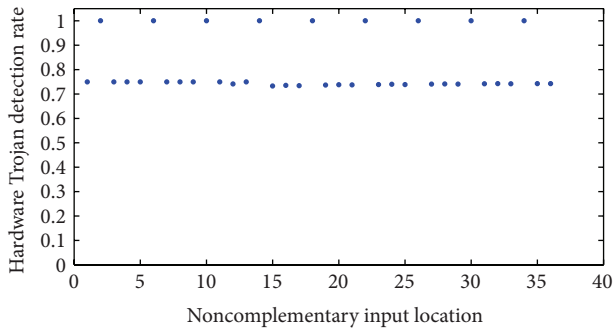


FIGURE 10: HT detection rate in c432.

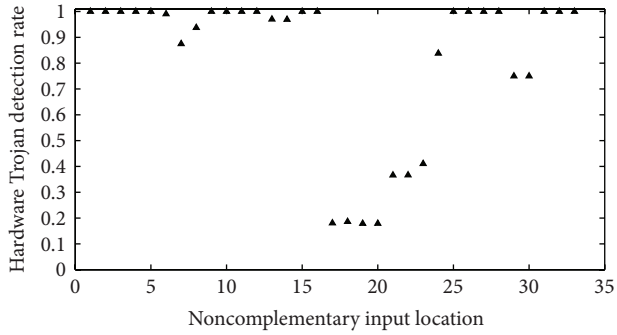


FIGURE 11: HT detection rate in c1908.

abnormal power upon HT insertion in the different locations of three ISCAS'85 benchmark circuits. Figure 14 shows the number of gates that are affected by one HT insertion. As can be seen, the number of gates yielding abnormal power generally increases with the circuit size and complexity. As shown in Figure 14, c3540 has the highest number of gates experiencing abnormal power per each HT insertion, compared to c1908 and c342. As HT insertion position moves towards the final output, the number of gates with abnormal power behavior decreases because the path of HT effect propagation is reduced. Since the abnormal short-circuit power also depends on input patterns of the target gate, the results reported in Figure 14 is not always integer valued. We averaged the number of gates affected by each HT insertion in three benchmark circuits. As shown in Figure 15, the average affected gate number for c3540 exceeds three. The higher number means more significant power will be induced by

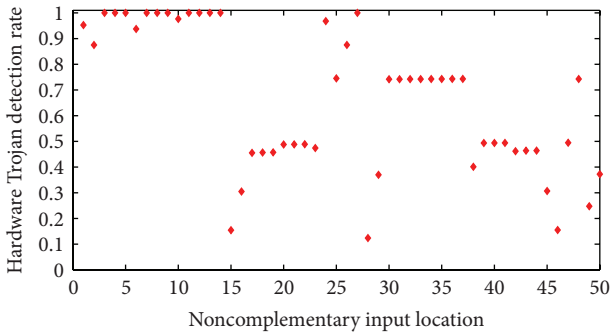


Figure 12: HT detection rate in c3540.

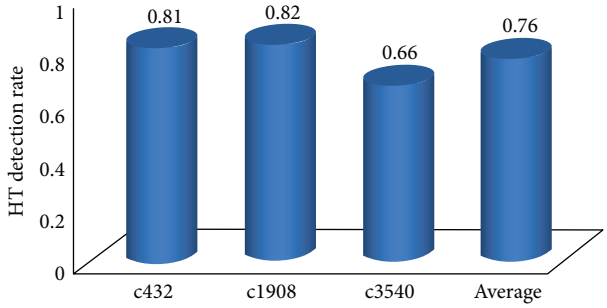


Figure 13: Average HT detection rate.

HT insertions; this feature has potential to be used in power-based HT detection.

Detecting the noncomplementary final output of DUT helps to improve the HT detection rate. As shown in Figure 16, not all test cases have abnormal power behavior. We collected the number of cases that have noncomplementary outputs (i.e., output error) and observed that the cases of noncomplementary DUT final output can achieve a HT detection rate of 1. This outstanding performance depends on circuit topology and the employed logic gates. Sometimes, the output error occurs at the same moment when abnormal short-circuit power is observed.

Sequential circuits are more likely to be affected by HT effect propagation, as latches and flip-flops have a higher probability to remain high with short-circuit power than combinational logic gates. We injected single HT on the inputs of benchmark circuits, s526, s832, s1196, and s1488, to model the impact of HT on circuits. As shown in Figure 17, on average, the HT detection rate on sequential circuit is higher than that in combinational circuits. The HT detection of s1488 and s1196 is close to 1. The average HT detection rate for the examined ISCAS'89 benchmark circuits is 0.85.

4. Conclusion

Hardware Trojans (HTs) challenge the chip security because of the increasing number of chips being fabricated, assembled, and packaged offshore. To enforce the confidence of chip security, efficient HT detection is imperative. HT detection can be performed during chip testing stage, although it

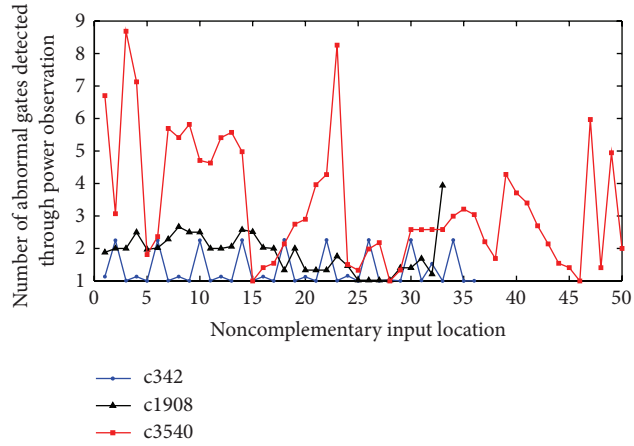


Figure 14: The number of gates experiencing abnormal power during each HT insertion.

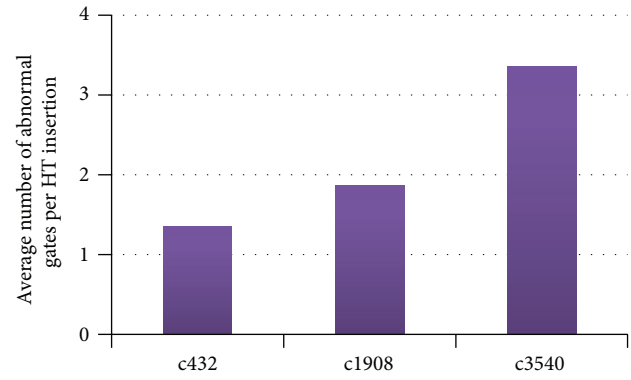


Figure 15: Average number of gates with abnormal power per each noncomplementary input pair.

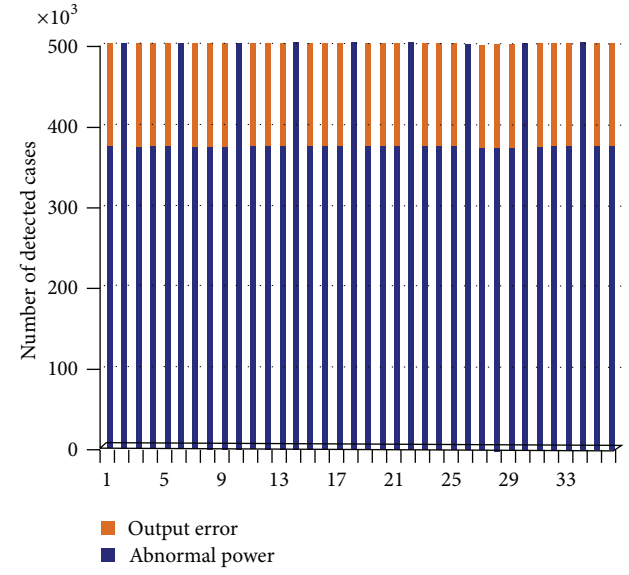


Figure 16: HT detection rate improvement by comparing complementary outputs in c432 circuit.

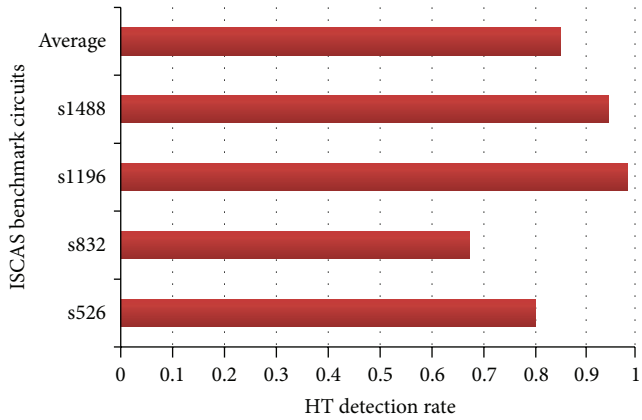


Figure 17: Average HT detection rate of different sequential benchmark circuits.

requires large numbers of test vectors and long verification times. As argued by many researchers, testing approaches may not be practical in identifying the rare events caused by HTs in a short period of time. Chip fingerprint is examined in IC authorization stages through side-channel analysis. Existing side-channel analysis approaches are challenged by process variation, lack of a perfect golden chip for comparison, and the presence of small-scale HTs. To address this need, we propose to use the inherent characteristic of DCVSL to detect HTs at runtime, without requiring a golden chip and a large number of test vectors. Our method is low-cost, convenient for user, and complementary to existing power-based side-channel analysis methods.

In this work, we exploit DCVSL's complementary feature on both inputs and outputs to detect hardware Trojans at runtime, rather than offline. Noncomplementary inputs in DCVSL-based systems lead to constant and abnormal short-circuit power peaks, which remain until the noncomplementary inputs disappear. A case study on a 64-bit ripple carry adder shows that the proposed method achieves from 50X to two orders of magnitude higher average power difference than CMOS-based power analysis. Such high power difference between normal operation and HT triggered conditions is desirable for power-base side-channel analysis. Evaluation on a 64-bit adder shows that our method achieves a HT detection rate approaching 100%, if HTs are inserted to flip one of the adder inputs logic value. As HT payload circuits are placed close to the final outputs, our abnormal power-based HT detection slightly loses its efficiency. The examination on the complementary characteristic of the outputs can improve the HT detection rate. Assessment on ISCAS'85 and ISCAS'95 benchmark circuits shows that the HT detection rate is in the range of 66% to 98%. On average, our method can detect 76% and 85% of HTs inserted in ISCAS'85 and ISCAS'89 benchmark circuits, respectively. By examining the complementary nature of the final output, we further improve the HT detection rate. Simulation on ISCAS'85 c432 circuit shows that the HT detection rate can be compensated to reach 100%.

In future work, we will validate the proposed method in larger-scale circuits. In addition, we will integrate our method with a current monitor to demonstrate the significance of proposed concept in real applications.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] J. Markoff, "Old Trick Threatens the Newest Weapons," October 2009, http://www.nytimes.com/2009/10/27/science/27trojan.html?pagewanted=all&_r=1&.

[2] J. Ellis, "Trojan integrated circuits," http://chipsecurity.org/2012/02/trojan-circuit/.

[3] S. Johnson, "Fake chips threaten military," San Jose Mercury News, September 2010, http://www.mercurynews.com/breaking-news/ci_15990184.

[4] M. Banga and M. S. Hsiao, "A region based approach for the identification of hardware Trojans," in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08)*, pp. 40–47, June 2008.

[5] D. Mukhopadhyay and R. S. Chakraborty, "Testability of cryptographic hardware and detection of Hardware Trojans," in *Proceedings of the 20th Asian Test Symposium (ATS '11)*, pp. 517–524, November 2011.

[6] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Design and Test of Computers*, vol. 27, no. 1, pp. 10–25, 2010.

[7] F. Wolff, C. Papachristou, S. Bhunia, and R. S. Chakraborty, "Towards Trojan-free trusted ICs: problem analysis and detection scheme," in *Proceedings of the Design, Automation and Test in Europe (DATE '08)*, pp. 1362–1365, Munich, Germany, March 2008.

[8] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia, "MERO: a statistical approach for hardware Trojan detection," in *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 396–410, 2009.

[9] M. Banga and M. S. Hsiao, "VITAMIN: voltage inversion technique to ascertain malicious insertions in ICs," in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '09)*, pp. 104–107, July 2009.

[10] H. Salmani, M. Tehranipoor, and J. Plusquellic, "A novel technique for improving hardware trojan detection and reducing trojan activation time," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 1, pp. 112–125, 2012.

[11] Y. Alkabani, "Trojan immune circuits using duality," in *Proceedings of the 15th Euromicro Conference on Digital System Design (DSD '12)*, pp. 177–184, 2012.

[12] Y. Jin and Y. Makris, "Hardware Trojan detection using path delay fingerprint," in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08)*, pp. 51–57, June 2008.

[13] K. Hu, A. N. Nowroz, S. Reda, and F. Koushanfar, "High-sensitivity hardware Trojan detection using multimodal characterization," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE '13)*, pp. 1271–1276, 2013.

[14] C. Bell, M. Lewandowski, and S. Katkoori, "A multi-parameter functional side-channel analysis method for hardware trust verification," in *Proceedings of the 31st IEEE VLSI Test Symposium (VTS '13)*, pp. 1–4, 2013.

[15] L. Wang, H. Xie, and H. Luo, "Malicious circuitry detection using transient power analysis for IC security," in *Proceedings of the International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE '13)*, pp. 1164–1167, 2013.

[16] S. Narasimhan, D. Du, R. S. Chakraborty et al., "Hardware Trojan detection by multiple-parameter side-channel analysis," *IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2183–2195, 2013.

[17] S. Narasimhan, X. Wang, S. Bhunia, W. Yueh, and S. Mukhopadhyay, "Improving IC security against Trojan attacks through integration of security monitors," *IEEE Design & Test of Computers*, vol. 29, no. 5, pp. 37–46, 2012.

[18] T. Huffmire, J. Valamehr, T. Sherwood et al., "Trustworthy system security through 3-D integrated hardware," in *Proceedings of the IEEE International Workshop on Hardware-Oriented Security and Trust (HOST '08)*, pp. 91–92, June 2008.

[19] X. Wang, H. Salmani, M. Tehranipoor, and J. Plusquellic, "Hardware Trojan detection and isolation using current integration and localized current analysis," in *Proceedings of the 23rd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT '08)*, pp. 87–95, October 2008.

[20] F. Koushanfar and A. Mirhoseini, "A unified framework for multimodal submodular integrated circuits trojan detection," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 1, pp. 162–174, 2011.

[21] D. A. Rennels and H. Kim, "Concurrent error detection in self-timed VLSI," in *Proceedings of the 24th International Symposium on Fault-Tolerant Computing*, pp. 96–105, June 1994.

Hindawi Publishing Corporation
VLSI Design
Volume 2014, Article ID 801241, 14 pages
http://dx.doi.org/10.1155/2014/801241

Research Article

On-Chip Power Minimization Using Serialization-Widening with Frequent Value Encoding

Khader Mohammad,[1] Ahsan Kabeer,[2] and Tarek Taha[3]

[1] *Birzeit University, P.O. Box 14, Birzeit, West Bank, Palestine*
[2] *Clemson University, Clemson, SC 29634, USA*
[3] *University of Dayton, Dayton, OH 45469, USA*

Correspondence should be addressed to Khader Mohammad; hajkhader@gmail.com

Received 19 January 2014; Accepted 2 April 2014; Published 6 May 2014

Academic Editor: Qiaoyan Yu

In chip-multiprocessors (CMP) architecture, the L2 cache is shared by the L1 cache of each processor core, resulting in a high volume of diverse data transfer through the L1-L2 cache bus. High-performance CMP and SoC systems have a significant amount of data transfer between the on-chip L2 cache and the L3 cache of off-chip memory through the power expensive off-chip memory bus. This paper addresses the problem of the high-power consumption of the on-chip data buses, exploring a framework for memory data bus power consumption minimization approach. A comprehensive analysis of the existing bus power minimization approaches is provided based on the performance, power, and area overhead consideration. A novel approaches for reducing the power consumption for the on-chip bus is introduced. In particular, a serialization-widening (SW) of data bus with frequent value encoding (FVE), called the SWE approach, is proposed as the best power savings approach for the on-chip cache data bus. The experimental results show that the SWE approach with FVE can achieve approximately 54% power savings over the conventional bus for multicore applications using a 64-bit wide data bus in 45 nm technology.

1. Introduction

There is a need for high-performance, high-end products to reduce their power consumption. The high-performance systems require complex design and a large power budget having considerable temperature impact to integrate several powerful components. Therefore, low energy consumption is a major design criterion in today's design. Low energy consumption improves battery longevity and reliability, and a reduction in energy consumption lowers both the packaging and overall system costs [1]. As the technology scaling down the power consumption is also decreasing and results in more sensitivity to soft errors so reliability would be affected. There are tradeoffs between power consumption and reliability in different ways. In future work overall reliability will be discussed and it will be evaluated how it can be improved by reducing the power consumption.

The primary goal of this research is for bus power minimization by reducing the switching activity while at the same time improving bus bandwidth for the compression technique and reducing the bus capacitance for the SW approach. The goal is similar to using switching activity and capacitance reduction in bus power savings; the key difference between the prior work and the work presented here is that the primary focus of this work is to explore a framework for bus power minimization approaches from an architectural point of view. As a result, this paper presents a comprehensive analysis of most of the possible bus power minimization approaches for the on-chip. This research explores a framework for power minimization approaches for an on-chip memory bus from an architectural point of view. It also considers the impact of coupling capacitance for estimating the on-chip bus power consumption. Finally this paper proposes a serialized-widened bus with frequent value encoding (FVE) as the best power savings approach for the on-chip (L1-L2 cache) data bus.

The organization of the rest of the paper is as follows. Section 2 presents background. Section 3 presents framework

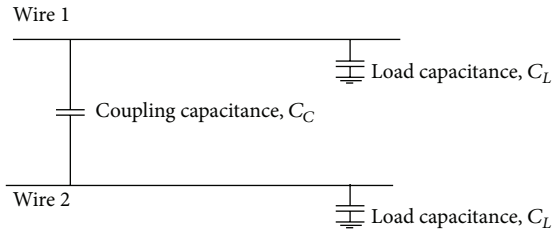


Figure 1: Load capacitance of a wire and coupling capacitance between the wires.

and proposed on-chip bus power model, a framework for bus power minimization approaches and their efficacy. Section 4 present experiment setup followed by Section 5 which presents the experiment results, a thorough comparison of the proposed technique with the other approaches.

2. Background

Memory bus power minimization techniques can be categorized as bus serialization [2–4], encoding [5–8], and compression techniques [9–14]. Non-cache-based encoding techniques reduce power by reordering the bus signals. Bus serialization reduces the number of wire lines, eventually reducing the area overhead. A serialized-widened bus reduces the capacitance of on-chip interconnections. Cache-based encoding techniques reduce the number of switching transitions using encoded hot-code. These techniques keep track of some of the previous transmitted data using a small cache on both sides of the data bus. Compression techniques reduce the number of wire lines contributing a reduction on in area overhead and an increase in the bus bandwidth. These compression techniques also reduce the switching activity. Serialization changes the data ordering transmitted through the data bus. This method contributes to reducing the switching activity as well. It may also improve the chance of data matching by incorporating it with cache-based encoding techniques because partial data matching is three times more frequent than full-length data matching [7].

Jacob and Cuppu [3] explored the dynamic random access memory (DRAM) system and memory bus organization in terms of performance, presenting design tradeoffs for the bank, channel, bandwidth, and burst size. They also measured the performance in relation to optimize the memory bandwidth and bus width. Suresh et al. [7] presented a data bus transmit protocol called the power protocol to reduce the dynamic power dissipation of off-chip data buses. Hatta et al. [2] proposed the concept of bus serialization-widening (SW) to reduce wire capacitance; their work focused on the power minimization of the on-chip cache address and data bus. Li et al. [15] proposed reordering the bus transactions to reduce the off-chip bus power.

In this chapter we present on-chip bus power model, a framework for bus power minimization approaches and their efficacy. We also discuss in detail the proposed technique and present a thorough comparison of our proposed technique to the possible approaches from power savings stand point.

The general equation of the bus power calculation is given as follows:

$$P = \alpha f W C V_{DD}^2, \tag{1}$$

where α is the switching activity, f is the frequency of the bus, W is the number of parallel data bus lines, C is the total capacitance of the bus, and V_{DD} is the swing voltage. The capacitance C of (1) can be divided into two parts as load capacitance C_L which is the parasitic capacitance to substrate with a constant potential and coupling capacitance C_C which is the parasitic capacitance between the adjacent lines (see Figure 1). In a deep submicron technology, the total capacitance no longer only depends on load capacitance of the wire. Coupling capacitance between the wires is a large factor as coupling capacitance is some order of load capacitance of the wire line [16–20].

The total capacitance is the sum of the load capacitance and coupling capacitance and it can be expressed as $C = C_L + 2C_C$ [2, 16, 21–23]. The equation of the power consumption calculation of the conventional bus line will be

$$P_C = \left(\alpha_{C_L} C_{C_L} + \alpha_{C_C} C_{C_C}\right) f_C W_C V_{DD}^2, \tag{2}$$

where α_{C_L} is the signal transition switching activity, C_{C_L} is the load capacitance, α_{C_C} is the coupling transitions switching activity, and C_{C_C} is the coupling capacitance between the conventional bus lines. The signal transition switching activity [2, 22] is given by

$$\alpha_{C_L} = \begin{cases} 0 & \text{if data transition from } 0 \longrightarrow 0 \text{ or } 1 \longrightarrow * \\ 1 & \text{if data transition from } 0 \longrightarrow 1. \end{cases} \tag{3}$$

The coupling switching activity [2, 22] depends on the transitions activity between two adjacent bus lines as follows:

$$\alpha_{C_C} = \begin{cases} 0 & \text{if wire 1 transition } 0 \longrightarrow 1 \text{ and} \\ & \text{wire 2 transition } 0 \longrightarrow 1 \\ 1 & \text{if wire 1 transition } 0 \longrightarrow 1 \text{ and} \\ & \text{wire 2 transition } 0 \longrightarrow 0 \\ 1 & \text{if wire 1 transition } 0 \longrightarrow 1 \text{ and} \\ & \text{wire 2 transition } 1 \longrightarrow 1 \\ 2 & \text{if wire 1 transition } 0 \longrightarrow 1 \text{ and} \\ & \text{wire 2 transition } 1 \longrightarrow 0. \end{cases} \tag{4}$$

Two of the main approaches to minimize the power consumption of a bus are to reduce the bus switching activity and the bus wire capacitance. Switching activity can be reduced through encoding techniques while the wire capacitance can be reduced by changing the wire width and spacing.

2.1. Bus Serialization and Widening. Bus serialization involves reducing the number of wires on the bus. If the number of transmission lines in a conventional bus is NC and the serialization factor is S, then the number of transmission of lines in the serialized version of the bus is given by NS = NC/S. The serialization factor can be any

Table 1: Serialization may increase or decrease switching activity. Parts (a) and (b) illustrate two different 16-bit data streams passing through a conventional 8-bit bus and a serialized 4-bit bus. In example (a) switching activity decreases, while in (b) it increases.

(a) 16-bit data stream → 0011 0011 0011 0011

Passing through 8-bit bus →	Data sequence	Signal	Coupling
	0000 0000	—	—
	00**11** 00**11**	4	3
	0011 0011	0	0
	Total number of transitions	4	3
Passing through 4-bit bus →	Data sequence	Signal	Coupling
	0000	—	—
	00**11**	2	1
	0011	0	0
	0011	0	0
	0011	0	0
	Total number of transitions	2	1

(b) 16-bit data stream → 0011 1100 0011 1100

Passing through - bit bus →	Data sequence	Signal	Coupling
	0000 0000	—	—
	00**11 11**00	4	2
	0011 1100	0	0
	Total number of transitions	4	2
Passing through 4 bit-bus →	Data sequence	Signal	Coupling
	0000	—	—
	00**11**	2	1
	1100	2	2
	00**11**	2	2
	1100	2	2
	Total number of transitions	8	7

integer multiple of 2. The throughput of a bus serialized by a factor of two is halved. To prevent a reduction in the throughput, the bus frequency can be doubled. This requires the increasing of the wire widths to support higher switching speeds. The advantage of serialization is that the bus occupies less area than a conventional bus. Serialization on its own may not necessarily reduce the switching activity and thus the energy consumption of a bus (see Table 1). Loghi et al. [5] examined the use of bus serialization combined with data encoding for power minimization. In this case, the bus area

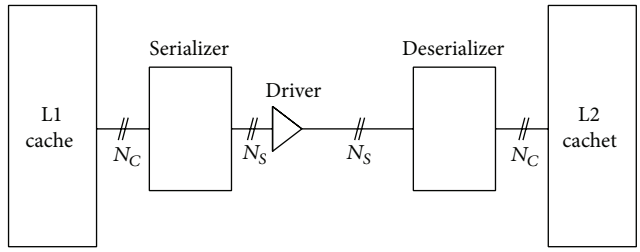


Figure 2: Basic structure and position of serializer and deserializer.

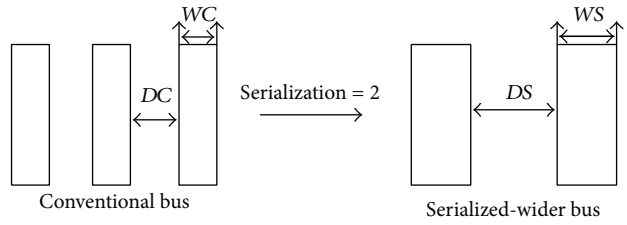


Figure 3: Basic structure of conventional and serialized bus lines.

was smaller, but the throughput of the bus was halved (since the frequency remained the same).

In a deep submicron technology, the switching energy consumed due to coupling capacitance is dominant [16, 17, 24–26]. The disadvantage of bus widening is that the bus occupies more area than a conventional bus. Hatta et al. [2] looked at combining bus serialization with bus widening in order to reduce bus power without increasing the bus area. In that study, the bus frequency was increased to keep the throughput constant. Although this required increasing the width of the wires, the extra spacing between the wires allowed this to be accommodated without a bus area overhead. Hatta et al. [2] also looked at combining a serialized-widened bus with differential data encoding and found that it helped on the address bus but not on the data bus.

In a serialized-widened bus, the operating frequency can be increased to keep the throughput the same as in a conventional bus. In this case, the serialized frequency is given by $fS = S \cdot fC$, where S is the serialization factor and fC is the frequency of the conventional bus. In order to implement bus serialization at a higher frequency, a serializer and deserializer are required at the sending and receiving ends of the bus, respectively (as shown in Figure 2).

Figure 3 shows the structure of data lines of a conventional bus and those of a serialized-widened bus. The relationship of the wire width and spacing between the wires of a conventional bus and a serialized-widened bus is

$$WS + DS = (WC + DC)\,S, \tag{5}$$

where WC is the wire width of the conventional bus, DC is wire spacing between the lines in the conventional bus, WS is the wire width of a serialized-widened bus, DS is wire spacing between the lines in the serialized-widened bus, and S is the serialization factor.

Table 2: Comparison of possible approaches to reduce on-chip data bus power.

	Approach	Bus freq.	Switching activity	Line cap.	Bus area
1	C (conventional)	f	α_C	C_C	Original bus area
2	S (serial)	$2f$	α_S	C_S	Reduced
3	W (widened)	f	α_C	C_W	At least double
4	E (encoded)	f	α_E	C_C	Unchanged
5	SW	$2f$	α_S	C_{SW}	Unchanged
6	SE	$2f$	α_{SE}	C_S	Reduced
7	WE	f	α_E	C_W	At least double
8	SWE	$2f$	α_{SE}	C_{SW}	Unchanged

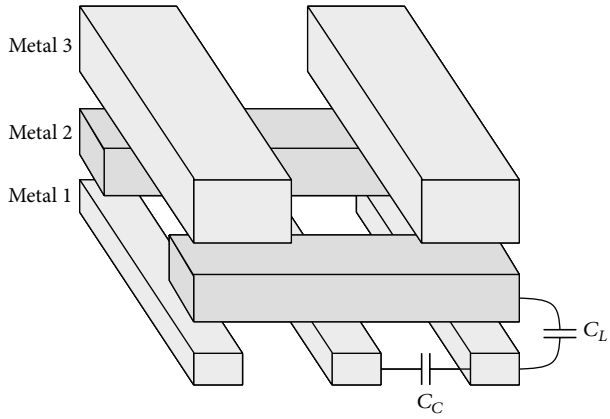


Figure 4: Line-to-line and crossover capacitance of a multilevel metal layer.

The width WC is different from the width WS to allow a higher frequency. Since the wire widths have to be changed to accommodate the higher operating frequency, the load capacitance of the bus wires (given in (5)) will change. In addition, the increase in wire spacing changes the cross-coupling capacitance. Thus the power consumption of the bus is given by

$$P_S = \left(\alpha_{SL} C_{SL} + \alpha_{SC} C_{SC}\right) f_S W_S V_{DD}^2, \quad (6)$$

where α_{SL} is the signal transition switching activity, α_{SC} is the coupling switching activity, C_{SL} is the load capacitance, and C_{SC} is the coupling capacitance of the serialized-widened bus. Figure 4 shows the capacitance values in a multilevel metal layer. The wire configurations values are taken from ITRS 2004 Update [27] and those values are used in the Chern et al. [23] equations to calculate the capacitance values. The frequency of the bus is given by the Kawaguchi and Sakurai [28] equation:

$$\frac{1}{f} \approx \left(1.63 \cdot \frac{C_C}{C_C + C_L} + 0.37\right) \cdot R \cdot \left(C_C + C_L\right). \quad (7)$$

Here, R is the resistance of the wire given by its width W, thickness T, and rate of resistance α (dependent on material property).

Consider

$$R = \frac{\alpha}{WT}. \quad (8)$$

Equations (7) and (8) can be used to determine the optimum wire width for the serialized-widened bus at the higher frequency.

3. Framework and Proposed Technique

The three fundamental approaches discussed earlier in this section to reduce bus power are serialization (S), encoding (E), and widening (W) of the bus. Combinations of these approaches are also possible, and in fact yield better results. Table 2 lists the possible types of buses based on these three approaches and their combinations (the first of which is a conventional bus (C) not employing any of the approaches). These approaches reduce the power through changes in the switching activity and the line capacitance of the bus.

Table 2 lists the relation between the switching activity and the line capacitance of the different approaches. It also lists the change in bus area and frequency due to the approaches. Two other important methods to reduce bus power are variations in the swing voltage and operating frequency. These two techniques can be applied in conjunction with all of the methods listed in Table 2. The framework shown in Table 2 can be used to categorize many of the approaches used to minimize the bus switching activity and wire capacitance. The encoding techniques proposed in [12–23, 27–47] fall under the category E listed in the table. The narrow bus encoding technique presented by Loghi et al. [5] falls under the category SE, while Hatta's serialized-widened bus [2] falls under the category SW.

There are four unique capacitance values and switching activities listed in Table 2. The relation between these capacitance values can generally be described as CW < CSW < CS < CC. If the serialized bus is running at a higher frequency to preserve the bus throughput, the wires and their spacing may have to be widened, thus possibly reducing their capacitance CS from the original bus value, CC. In the widened bus, the wires spacing is increased, making this type of bus having the lowest wire capacitance. However, there is a significant bus area overhead in this approach. The serialized-widened bus running at a higher frequency to preserve the throughput will have slightly less wire spacing than the widened bus since the wires will have to be made wider for the higher frequency. Thus the capacitance of this bus, CSW, will be more than that of the widened bus, CW, but

Table 3: Architectural configuration of the simulator used in the experiment.

System Parameters	
Number of processor cores	2, 4, 8
Super scalar width	4, out-of-order
L1 instruction cache	16/32/64 KB, direct-mapped, 1-cycle
L1 data cache	16/32/64 KB, 4-way, 1-cycle
L1 block size	32 B
Shared L2 cache	1 MB, 4-way, unified,, 12-cycle
L2 block size	64 B
RUU/LSQ	16/8
Memory ports	2
TLB	128-entries, 4-way, 30-cycle
Memory latency	96-cycle
Memory bus width	1/2/4/8 B

Table 4: Benchmarks, types, and number of warm up instructions used in the experiment.

Benchmarks	Type	Warm up instructions
gzip (pro)	Int	2000 M
gzip (src)	Int	1400 M
Wupwise	FP	2000 M
Gcc	Int	2000 M
Mesa	FP	700 M
Art	FP	2000 M
Mcf	Int	1000 M
bzip2 (pro)	Int	2000 M
bzip2 (src)	Int	2000 M
Twolf	Int	900 M
mpeg2d	MB	0 M
Gsm	MB	0 M
mpeg2e	MB	0 M

still less than the serialized bus, CS (since the wires are more spaced out than a serialized bus).

The relation between the switching activities is highly dependent on the data values passed on the bus. Therefore a strict relation between the switching activities cannot be shown. However, in general it can be expected that an encoded bus will have less switching than a conventional bus (hence $\alpha E < \alpha C$). In addition the serialized-encoded bus (SE) will also likely have a lower switching activity than a conventional bus (hence $\alpha SE < \alpha C$). The relation between the switching activity of a serialized bus (αS) and a conventional bus (αC) is hard to predict.

This paper proposes data bus power reduction techniques for the SWE approach. This work compares these approaches with existing power reduction methods that fall under the different categories in Table 2. This work finds that the SWE approach works best since this method reduces both the wire capacitance and the switching activity significantly.

4. Experimental Setup

This section discusses the target system of the experiment and the memory structure used to collect the memory traces. The first subsection describes the architecture of sim-outorder, the superscalar simulator from the Simplescalar tool suite [48]. In the subsection followed we discusses the benchmarks suite and the input sets that are used in this paper. In the last part of this section we present the switching activity computation methodology.

4.1. Simulator. This experiment uses a modified version of Simplescalar 3.0d's sim-outorder simulator [48] to collect our cache request traces. The model architecture has mid-range configuration. Table 3 summarizes the architectural configuration of our simulator. The baseline configuration parameters are typical those of a modern chip multiprocessors and out-of-order simulator. This work keeps the L1 cache size smaller to get more memory access which results in more accurate behavior of memory access and memory bus. This work develops another simulator written in program C to calculate the switching activity for the bus power estimation.

4.2. Benchmark Suites. This experiment uses 6 integers and 3 floating point benchmarks from SPEC2000 suite [49] and 3 benchmarks from MediaBench suite [47]. This selection is motivated by finding some memory intensive programs (mcf, art, gcc, gzip, and twolf) [3] and some memory nonintensive programs. The simulation wants to use reference inputs of the SPEC2000 suite because of having smaller data sets of test or training inputs. For each of the benchmark of SPEC2000 suite, this work divides the total run length by 5 and warm up for the first 3 portions with a maximum of 2 billion instructions using fast-forward mode cycle-level simulation. A 200 million instruction window is simulated using the detailed simulator. For MediaBench suite, this work simulates the whole program to generate the required traces without any fast forwarding. Table 3 lists the reference inputs that are chosen from the SPEC2000 benchmark and MediaBench suite and the number of instructions for which the simulator is warmed up. Among these benchmarks, a group of benchmarks are selected to run in multicore processor units qs in Table 4. This selection gives importance to group the memory intensive programs to get more accurate behavior of memory access than to group memory nonintensive programs. Table 5 summarizes the list of benchmarks used for 8, 4, and 2 cores processing units.

4.3. Switching Activity Computation. A power simulator written in C is integrated with the modified Simplescalar sim-outorder simulator [48] to calculate the switching activity of the data transitions between L1 and L2 cache through L1-L2 cache bus. The simulator has several functionalities for calculating the switching activity for all six different kinds of encoding techniques listed in Table 6.

During serialization-widening, the simulator uses two sets of value cache (VC) for LSB and MSB data matching instead of using one unified VC. Figure 5 shows the different

Table 5: Combination of benchmarks used for multiprocessing cores.

Number of cores	Set	Name of benchmarks
8	1	mcf, art, gcc, twolf, mpeg2d, gzip (pro), mesa, bzip2 (pro)
	2	gzip (src), mcf, gcc, gsm, wupwise, mpeg2e, art, bzip2 (src)
4	1	mcf, art, mpeg2e, gzip (pro)
	2	twolf, bzip2 (pro), mesa, art
	3	gcc, gzip (src), bzip2 (src), gsm
2	1	mcf, art
	2	gcc, twolf

Table 6: Listing of different encoding techniques implemented in this experiment.

Name	Abbreviation
Bus-invert coding	bi
Transition signaling	xor
Frequent value encoding with one hot-code	Fv
Frequent value encoding with two hot-code	fv2
TUBE with one hot-code	Tube
TUBE with two hot-code	tube2

structures of two sets of VC with serialization. The data bus size is varied frequently to compare the effectiveness of different possible approaches and encoding techniques keeping the total amount of data the same. For example, if a data stream of 64-bit wide requires 1 transition using 64-bit wide data bus, it requires 8 transitions using 8-bit wide data bus.

5. Results and Analysis

This section presents the experimental results. It has a general comparison of the cache bus power minimization using the seven possible approaches listed in Table 2. It further examines in detail three of the approaches that do not change the bus area and finds that the SWE approach performs the best. It also presents an in depth analysis of the SWE approach performance under various architecture and technology configurations. At the end of this section we discuss the performance, power, and area overhead for the proposed technique.

5.1. Power Savings for Different Possible Approaches. The seven possible bus power savings approaches listed in Table 2 earlier are different combinations of serialization (S), bus widening (W), and encoding (E). Figure 6 shows the power savings on the L1-L2 cache data bus for the different architecture-benchmark combinations listed in Table 5 using these approaches. A 64-bit data bus implemented on 45 nm technology is assumed. The techniques reduce bus power by

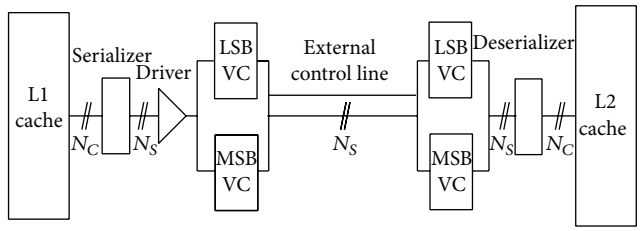


Figure 5: Structure of 2 sets of value cache combined with serialization.

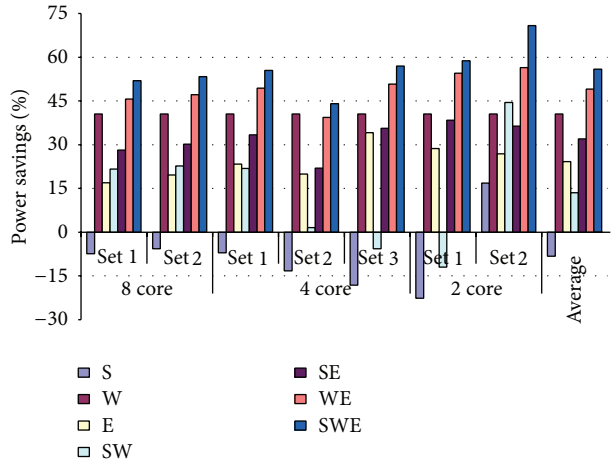


Figure 6: Comparison of the % of power savings using the different data bus power reduction approaches. Results are compared to a conventional 64-bit L1-L2 cache data bus at 45 nm technology.

minimizing bus switching activity, bus wire capacitance, or both.

When the three approaches for power reduction are applied on their own, bus widening performs the best. The serialization (S) approach performs poorly for most of the architecture configurations listed in Figure 6 (the bus power is generally increased). This is primarily due to the fact that serialization generally increases switching activity. The bus capacitance is actually reduced partially since the wires are spaced out further to allow the frequency to be doubled. However, this reduction in capacitance is not enough to offset the increased switching activity. The widening (W) approach performs very well since it reduces the bus wire capacitance significantly. The disadvantage of the approach is that it almost doubles the bus area. There are six different encoding techniques (E) that are tested (see Table 6). Figure 6 shows the result from the best encoding technique for each architecture configuration. Encoding reduces switching activity without affecting the bus capacitance and so does minimizing the bus power. This approach does not change the bus area or frequency.

When using combinations of the three approaches, the serialized-widened-encoded (SWE) method performs the best. The serialized-widened (SW) approach reduces the bus capacitance by widening the wire spacing, but generally increases the switching activity through serialization. The net

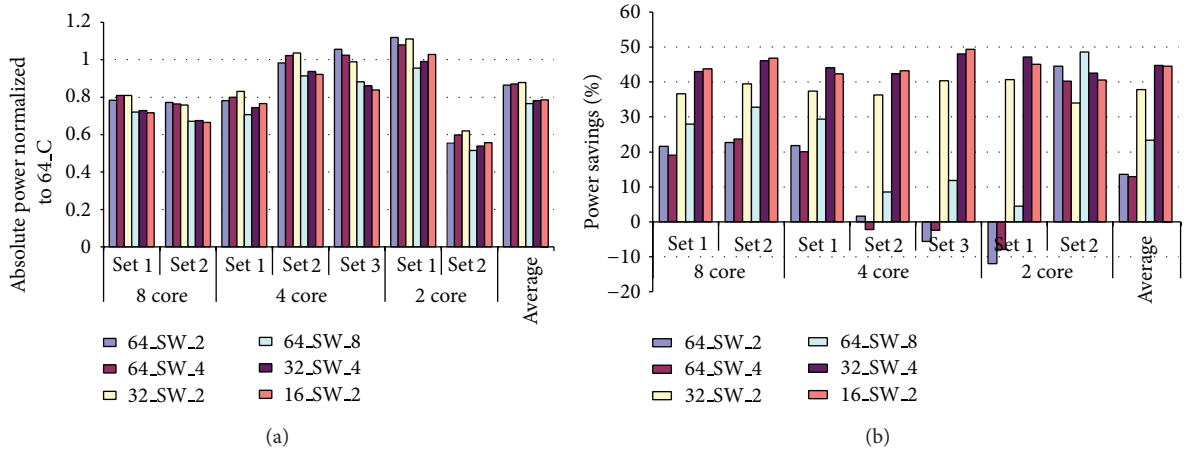


Figure 7: (a) % of power savings achieved and (b) absolute power normalized to 64-bit conventional bus power using bus serialization for 64-, 32-, 16-bit wide bus for different serialization factors. The figure legend indicates the first number as bus width, S as serialization, and the last number as the serialization factor.

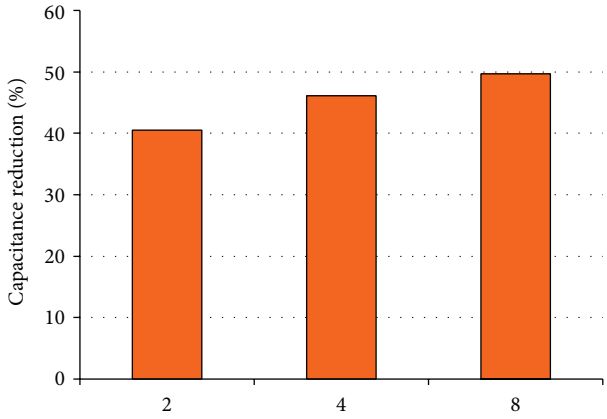


Figure 8: % of capacitance reduction using serialized-widened data bus for different serialization factors in 45 nm technology.

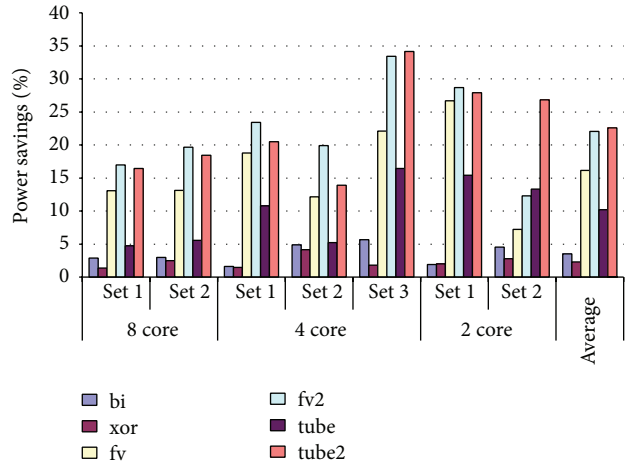


Figure 9: % of power savings for using different encoding techniques for 64-bit wide data bus for different number processing cores with several benchmark combinations.

result of these two opposing effects is generally a decrease in the power consumption (although there are cases where power is actually increased). This is the approach proposed by Hatta et al. [2] for both the address and data buses. The serialized-encoded (SE) method reduces the bus power mainly through a reduction in switching activity. There is also a slight reduction in capacitance due to the serialization. The widened-encoded (WE) approach reduces the power by minimizing both the switching activity and bus capacitance. It however has the disadvantage in increasing the bus area. Finally the serialized-widened-encoded (SWE) approach produces the best results for the architectures in Figure 6 by minimizing the bus capacitance and switching activity while keeping the bus area constant.

The rest of this chapter considers primarily the SW, E, and SWE approaches as these do not change the bus area. Unless explicitly stated, a 45 nm technology implementation is assumed.

5.2. Serialization-Widening (SW). Figure 7(a) shows the power savings of using a serialized-widened bus (as proposed by Hatta et al. [2]) for different bus widths and serialization factors. The results show that the SW approach performs well for narrow buses. Figure 7(b) shows the absolute power consumption of the SW approach with different architectural configurations normalized with a 64-bit wide conventional data bus. The average power consumption of a specific bus width does not vary to each other irrespective of serialization factors.

Figure 8 shows the percentage of capacitance reduction using the serialization-widening data bus approach for different serialization factors. The figure shows that a serialization factor of 4 or 8 does not provide a significant reduction of capacitance over a serialization factor of 2.

5.3. Encoding (E). Figure 9 compares the power savings from the different encoding schemes presented in Table 6 for a 64-bit L1-L2 cache data bus. Table 7 shows the power savings of the encoding techniques for various cache bus widths. For

Table 7: % of power savings for different bus widths and encoding techniques.

	64-bit	32-bit	16-bit	8-bit
bi	3.491879	13.10983	12.55175	**10.22098**
xor	2.292305	7.015497	4.343778	3.213289
fv	16.15159	38.09723	**25.42917**	5.665817
fv2	22.04569	37.30793	17.56435	2.736691
tube	10.20364	29.08284	9.316818	1.79227
tube2	**22.58817**	**43.95828**	20.94449	2.792153

Table 8: Hit rate and number of hit in one or two transition cache locations using FV and FV2 techniques for 8-core dataset 1.

	FV	FV2
Hit rate (%)	68.08	79.87
Number of 1-transition hit	11586571	2047667
Number of 2-transition hit	0	11545453

the 64-bit and 32-bit wide buses, the frequent value or TUBE approaches with two hot-codes (FV2 and TUBE2) perform the best. This is mainly because the wide bus allows for a large number of entries in these encoding caches. With a 16-bit data bus width, a frequent value cache using one hot-code performs better. This is because the larger cache size of FV2 than FV increases the hit rate, but large number of them hit in the location that requires a switching activity of two instead of one. Table 8 lists the hit rate and the number of one or two transition cache location hit of FV2 and the number of one transition cache location hit of FV for simulating 8-core set 1 application. It is obvious from the data of the table that FV2 performs poorly as large data matching hits in two transition cache locations. An improvement of this situation is to map the most frequent data value in the cache location of smaller number of transitions. This type of encoding technique is proposed by Suresh et al. [7]. It can be easily implemented in advance as their proposed context independent codes works for known dataset of embedded processing systems. But, it requires very complex hardware design to implement for a real-time data arrangement. For the 8-bit cache bus width, none of the cache-based approaches work well as their hit rates are low (since values get replaced too often). In this case bus-invert has the best performance.

5.4. Serialization-Widening with Encoding (SWE). Figure 10 compares the power savings from the different encoding schemes presented in Table 6 using the serialized-widened-encoded (SWE) scheme for a 64-bit L1-L2 cache data bus. Table 9 shows the power savings of the encoding techniques for various cache bus widths and a serialization factor of 2. For the 64-bit and 32-bit wide buses, the frequent value approach (FV) performs the best. This is mainly because the wide bus allows for a large number of entries with a higher number of switching activity (as given example in Table 8) in these encoding caches. With a 16-bit data bus width, a bus invert performs better. This is because we end up with an 8-bit bus after serialization, and the cache hit rates become too low for this configuration.

Table 9: % of power savings for different bus widths and encoding techniques.

	64-bit	32-bit	16-bit
bi	25.23832	35.94936	**41.25804**
xor	16.42357	17.68467	22.45874
fv	**55.94778**	**59.1091**	27.96345
fv2	49.89886	51.8574	10.45392
tube	44.20373	44.15911	3.55008
tube2	53.97197	54.80531	10.87319

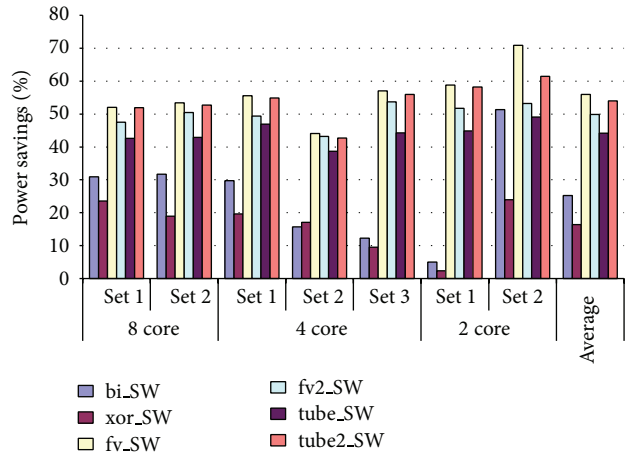


Figure 10: % of power savings using SWE approach for different encoding techniques for 64-bit wide data bus.

Table 10: Variation of value cache table size with encoding techniques.

Encoding technique	Table size		Max. possible switching activity
	Data size	Number of entry	
FV	32-bit	32	1
FV2	32-bit	528	2
TUBE	32-bit	5	1 or 2
	24-bit	5	
	16-bit	5	
	8-bit	8	
	16-bit	16	
TUBE2	32-bit	30	2 or 4
	24-bit	30	
	16-bit	30	
	8-bit	68	
	16-bit	68	

5.5. Power Savings under Different Architecture Options. Figure 11 presents the percentage of power savings for the SWE approach using frequent value encoding (FVE) and the

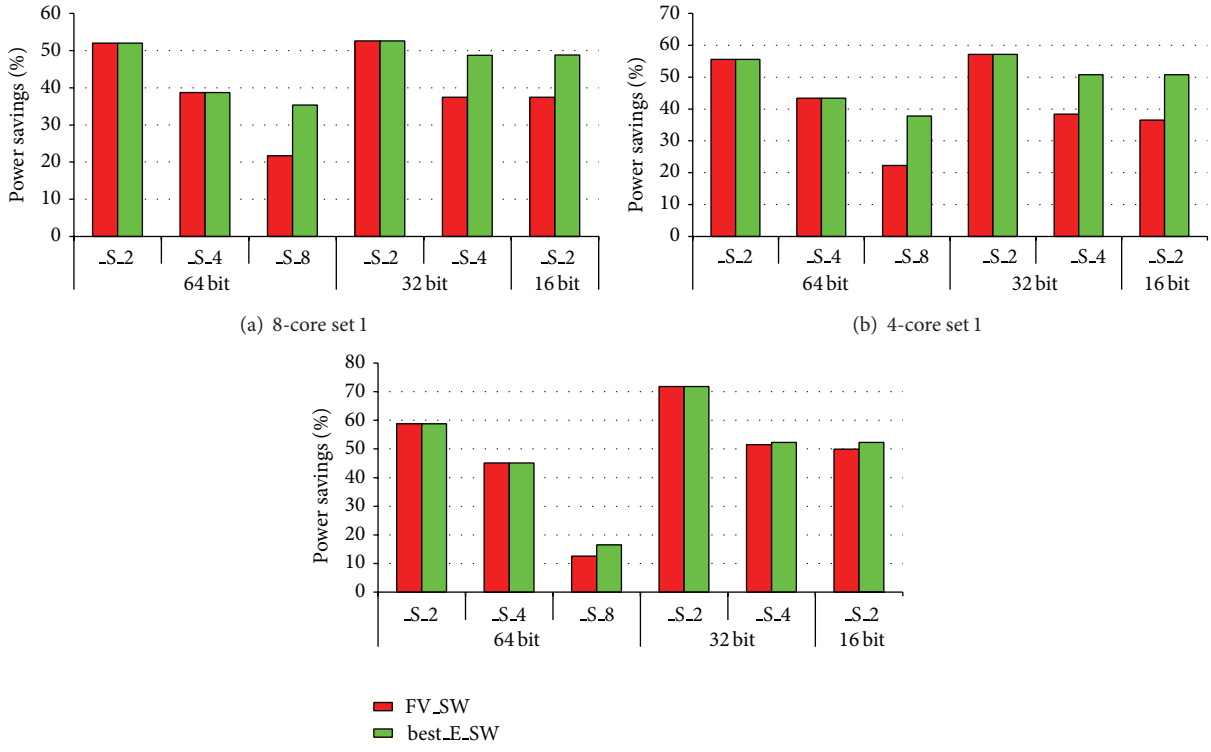


Figure 11: Comparison of % of power savings between different serialization factors with different cache bus width.

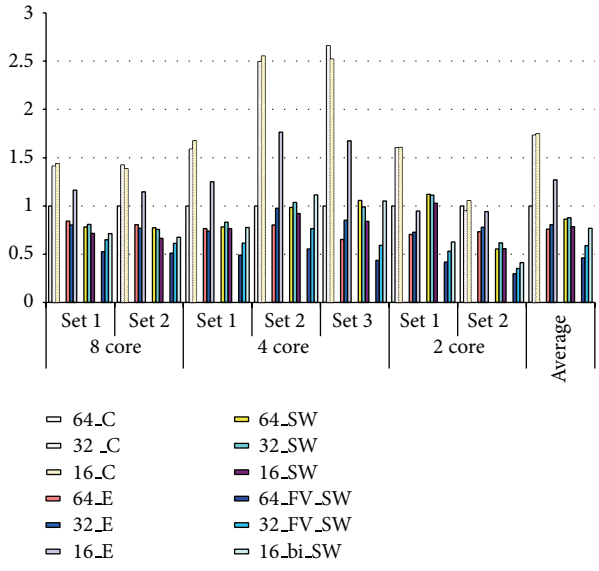


Figure 12: Absolute power consumption for 64-bit, 32-bit, and 16-bit bus, with encoding (E), serialization-widening (SW), and serialization-widening with encoding (SWE) normalized to 64-bit bus width for 32 KB L1 cache.

best encoding for different bus widths and serialization factors. The amount of power savings achieved by this approach depends on several factors. These factors include cache data bus width, types of applications, number of processing cores, L1 cache size, and type of technology used. For a specified bus width, a serialization factor of 2 with encoding gives more power savings than any other combinations. Although higher serialization factor can contribute in more capacitance reduction, it reduces the number of bus lines. This reduction of the number of bus lines decreases the chance of data matching for cache-based encoding. To choose a cache bus width for L1-L2 cache bus design, Figure 11 gives a comparative view of power savings for different cache bus width using the proposed technique with other best encoding technique. The proposed technique works well for wide data bus, but poorly performs for narrow bus.

Cache bus power consumption can be varied with bus width, application sets, and different approaches (E, SW or SWE). Figure 12 is a comparative view of cache bus power for a 32 KB L1 cache with 64-/32-/16-bit bus size. The graph shows that a 32-bit wide bus consumes more power than a 64-bit wide bus for most of the application sets used in this experiment. For a 16-bit wide bus, it consumes almost similar or sometimes more power than a 32-bit wide data bus. Encoding (E) approach consumes almost the same amount of power for 64-/32-bit wide data buses. This indicates that the power consumption of the E approach is independent of the bus size. A 16-bit data bus requires a bit higher power than either a 64-bit or 32-bit wide data bus using E approach. SW approach gives us a similar result for the 64-bit and 32-bit data buses. But, a 16-bit data bus requires quite less power than a 64-bit or 32-bit using the SW approach. Using the SWE

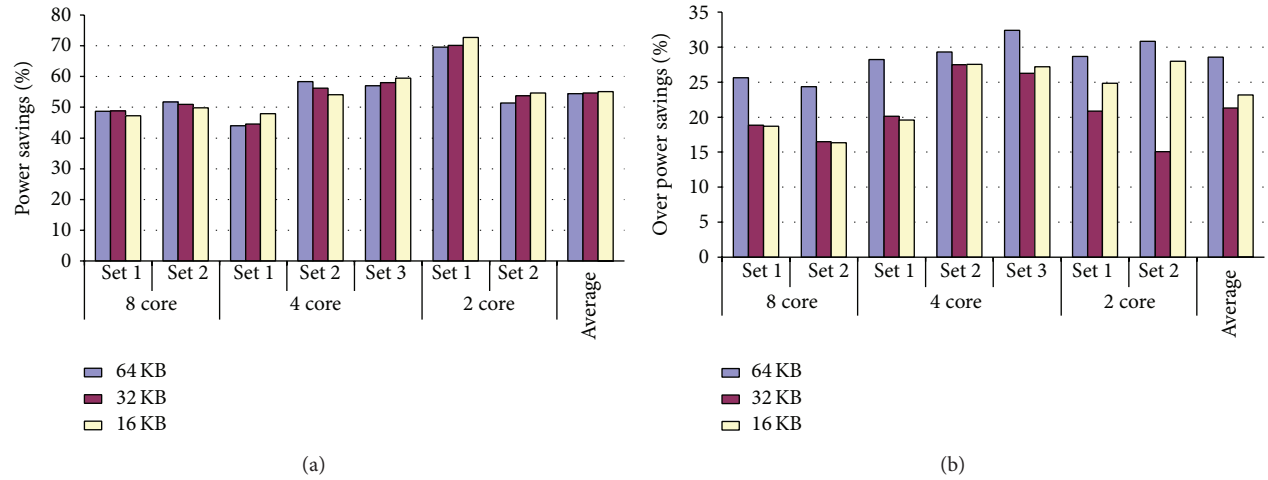


Figure 13: Comparison of (a) % of absolute power savings using different L1 cache sizes for a 64-bit wide data bus using serialization-widening with frequent value encoding and (b) % of relative power savings using a 64-bit wide bus compared to a 32-bit wide bus (both of the bus used serialization-widening with frequent value encoding).

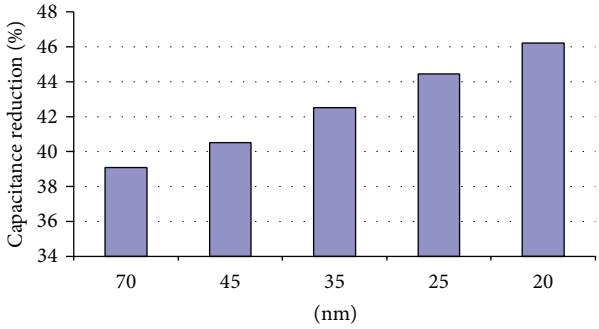


Figure 14: % of capacitance reduction for using serialized-widened bus with respect to conventional bus for a serialization factor of 2 for different technologies.

approach, a 64-bit wide data bus consumes approximately 22% less power than a 32-bit wide data bus for the same application sets. The best encoding that supports the SWE approach is frequent value encoding (FVE). FVE works much better with SWE approach than other cache-based techniques because of the reduced number of bus lines. The value cache size of the cache-based encoding depends on the number of bus lines. The reduced number of bus lines reduces the value cache entry which hurts in a data matching chance for TUBE. For FV2 and TUBE2, it increases the table size to a large number, but the overhead increases yielding a large number of switching activity. Table 10 gives a comparison of value cache size among different cache-based encoding techniques for a 32-bit data bus. The comparison of the same study for the 32-bit and 16-bit data bus gives us a good indication that the SWE approach (FVE as the best encoding) for a 32-bit wide bus consumes approximately 17% less power than that of a 16-bit wide data bus. The results also notice that both SW approach and SWE approach more or less performs the same for a narrow bus (a 16-bit wide data bus).

Reliability is also another concern which points to the need for low-power design. There is a close correlation between the power dissipation of circuits and reliability problems such as electromigration and hot-carrier. Also, the thermal caused by heat dissipation on chip is a major reliability concern. Consequently, the reduction of power consumption is also crucial for reliability enhancement. As a future work, we will be working in another paper to evaluate constraint on reliability and power.

5.6. Different L1 Cache Size. Figure 13 gives a comparison of the absolute power savings of using a 64-bit wide data bus with SWE (FVE) approach having 64 KB, 32 KB, and 16 KB L1 cache size. According to the results, the cache size does not affect in power savings of the proposed technique. Although the cache size can change the order of data transitions through the cache bus, the proposed technique works well irrespective of the changing of data transitions transmitted through the data bus. Thus, this proposed approach keeps consistent result with the variation of L1 data cache size. This figure also compares the percentage of relative power savings of using a 64-bit wide bus compared to a 32-bit bus for the same L1 cache size. Different bus size may change the ordering of the same data set and can significantly affect the number of switching activity. So, changing the cache size alters the data requests from the lower level cache and passing the data requests using different bus width may revise the number of switching activity. This effect can visualize from the Figure 13(b) but still it favors a 64-bit wide data bus from a power saving standpoint compared to a 32-bit wide data bus.

5.7. Different Technologies. This work extends the experiment for different technologies not keeping limited to different cache bus width and L1 cache size. As industry is already started to manufacture for less than 65 nm process technology, the experiment considers small gate size as 70, 45, 35, 25 and 20 nm technology. The experiment finds the capacitance

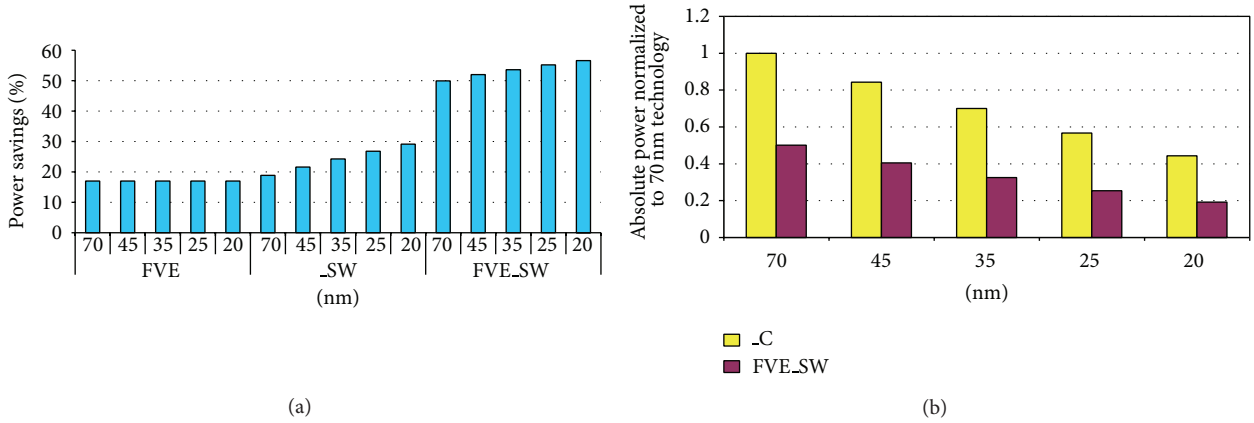


Figure 15: (a) % of power savings using different technologies for a 64-bit data bus experimenting on application set 1 in 8 processing cores, (b) absolute power consumption of the same set (8 core set 1) for different technologies (power consumption values are normalized to 70 nm technology).

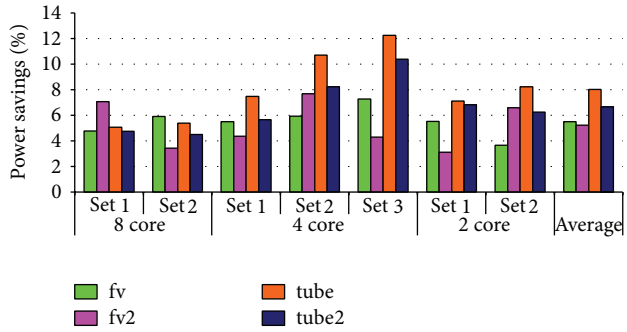


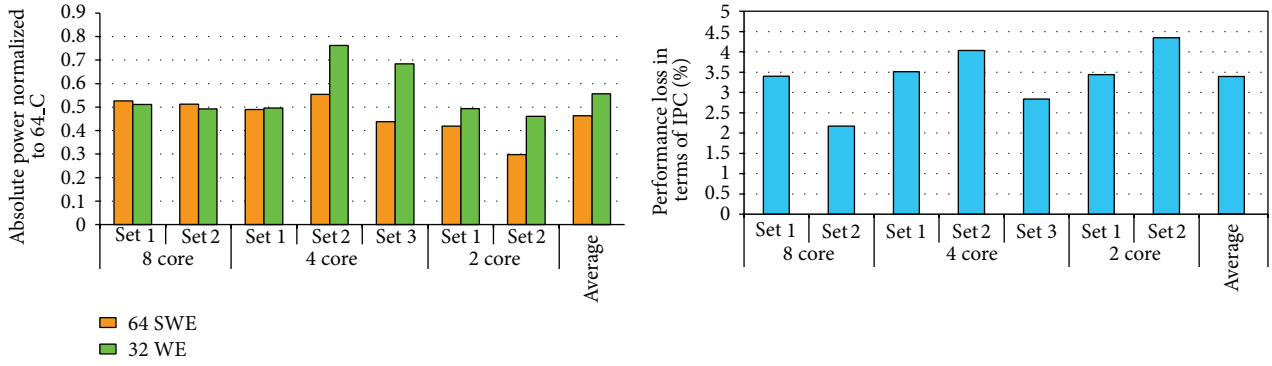
Figure 16: % of over power savings for using split cache instead of unified cache in cache-based encoding techniques.

reduction for different technology as shown in Figure 14. Figure 15(a) presents a comparison of the power savings using encoding, serialization-widening, and serialization-widening with FVE. The results shown in Figure 15 uses a 64-bit wide data bus for application set 1 in 8 processing cores. The amount of power savings is in similar fashion for different technologies, but the absolute power consumption reduces with shrinking the technology as shown in Figure 15(b). This is because the swing voltage reduces with shrinking the technology [27]. Although shrinking the technology increases the capacitance (capacitance, $C = \varepsilon * A/d$), serialization-widening gives us the advantage of using extra space between the wires which reduces the overall capacitance compared to the conventional bus and finally reduces the total power budget. Using this advantage, the proposed approach improves the power savings significantly.

5.8. Split Value Cache versus Unified Value Cache. Frequent value encoding (FVE) uses a unified value cache (VC) to implement the VC structure. The size of the VC depends on the type of pattern matching algorithm (full or partial) and type of hot-code (one or two) used in the implementation. In the proposed technique, the simulation uses two sets of VC instead a unified from the VC entry. Figure 16 presents a comparison of the power consumption VC. Two sets of VC hold the least significant bits (LSB) and most significant bits (MSB) part of the data value for implementing serialization-encoding approach as serialization breaks the whole data sequence. Utilization of two sets of VC increases the chance of hits in the VC. This also keeps consistent of the two separate VC as LSB part changes more frequent than MSB part. This removes the necessity of a frequent replacement using these two types of VC implementation. The figure shows that using two separate VC structure gives approximately 5% of more power savings for FVE-based technique and 8% for TUBE than using one unified VC.

5.9. Widened-Encoded Data Bus of 32 Bit Wide. A widened-encoded (WE) 32-bit wide data bus requires the same area as the SWE approach of a 64-bit wide. The results of Figure 6 show that WE approach works very close to SWE approach in power savings. But, the 64-bit wide WE data bus requires double area. This motivates us to compare the power consumption of the WE approach having a 32-bit data bus compared to the SWE approach having a 64-bit wide data bus. Figure 17(a) gives the absolute power consumption normalized with the 64-bit wide conventional data bus. The results show that these two approaches consume almost the same amount of power. The benefit of using WE approach is that it does not require higher operating frequency. But, it has to pay performance loss in terms of IPC for using narrow data bus. The experimental results having the performance loss are shown in Figure 17(b).

5.10. Performance Overhead. Performance overhead is a considerable issue in designing a serializer with frequent value cache (FVC) unit. Figure 18 presents the architectural configuration of a serializer-deserializer with the FVC unit between the L1-L2 cache block.



(a) (b)

Figure 17: (a) Comparison of absolute power consumption of SWE approach (64 bit wide data bus) and WE approach (32-bit wide data bus) and (b) % of performance loss of using 32-bit wide data bus instead of 64-bit wide data bus.

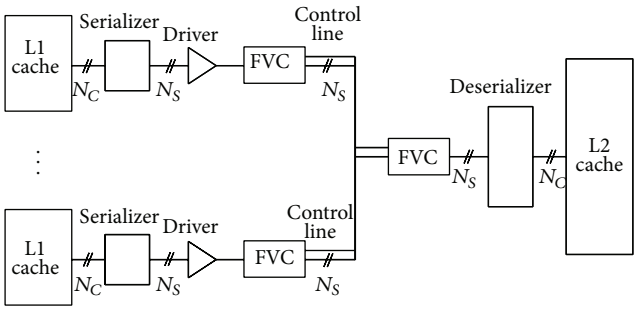


Figure 18: Architectural configuration of serializer-deserializer with FVC unit.

Hatta et al. [2] presented a novel work about bus serialization-widening and showed that the serialized-widened bus operates in faster frequency than the conventional bus. Liu et al. [32] talked about pipelined bus arbitration with encoding to minimize the performance penalty which might be less than 1 cycle. Although most of the works supports minimized performance penalty using serialization-widening with frequent value technique, it takes 2 cycles penalty in worst case. This work runs the simulation using 2 cycles and 1 cycle performance penalty for L2 data cache access using different application sets. The results present the performance loss in Figure 19. This work further includes a comparative view of absolute power savings using a 64-bit wide bus with serialization-widening and frequent value encoding for 32 KB L1 cache size at 45 nm technology.

According to the dataset of Figure 19, about 2.5% average performance loss for worst case (2 cycle penalty) if the approach cannot achieve the advantages of faster serialized bus and pipelining in data transmission. This comes down to average 1.35% performance degradation for 1 cycle penalty. Further investigation looks into the area required for additional circuitry. Different citations find that a minimum of approximately 0.05 mm^2 area is required to implement the value cache with serializer. The additional peripheral also consumes extra 2% power required by the wire [2, 7, 35].

6. Conclusion

In system power optimization, the on-chip memory buses are good candidates for minimizing the overall power budget. This paper explored a framework for memory data bus power minimization techniques from an architectural standpoint. A thorough comparison of power minimization techniques used for an on-chip memory data bus was presented. For on-chip data bus, a serialization-widening approach with frequent value encoding (SWE) was proposed as the best power savings approach from all the approaches considered.

In summary, the findings of this study for the on-chip data bus power minimization include the following.

(i) The SWE approach is the best power savings approach with frequent value encoding (FVE) providing the best results among all other cache-based encodings for the same process node.

(ii) SWE approach (FVE as encoding) achieves approximately 54% overall power savings and 57% and 77% more power savings than individual serialization or the best encoding technique for the 64-bit wide data bus. This approach also provides approximately 22% more power savings for a 64-bit wide bus than that for a 32-bit wide data bus using 32 KB L1 cache and 45 nm technology.

(iii) For a 32-bit wide data bus, the SWE approach (FVE as encoding) gives approximately 59% overall power savings and 17% more power savings than a 16-bit wide bus for the same L1 cache size and technology.

(iv) For different cache sizes (64 KB L1 cache size and 45 nm technology), a 64-bit wide data bus gives approximately 59% overall power savings and 29% more power savings than for a 32-bit wide data bus using the SWE approach with FV encoding.

In conclusion, the novel approaches for on-chip memory data bus minimization were presented. The simulation studies for the same process node indicate that the proposed techniques outperform the approaches found in the literature in terms of power savings for the various applications considered. The

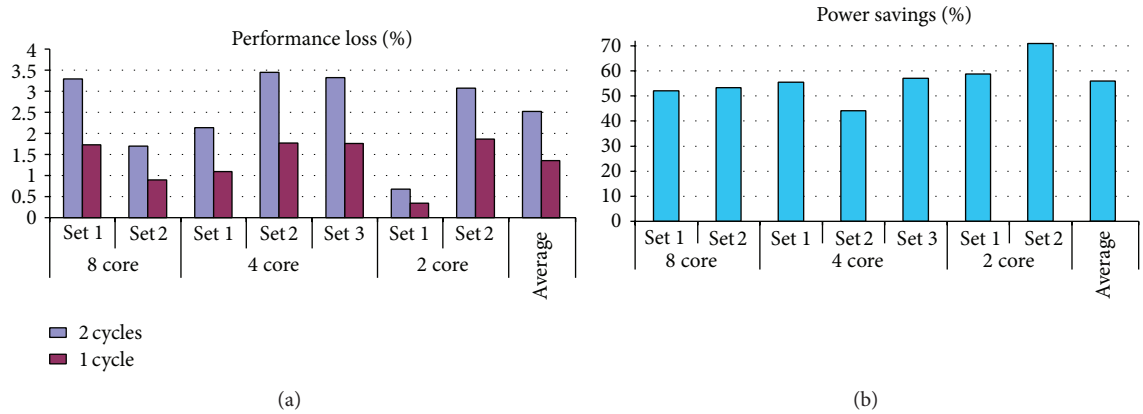


Figure 19: (a) % of performance degradation in term of instruction per cycle (IPC) for using 64-bit serialized bus with encoding for 2/1 cycle performance penalty instead of using conventional bus and (b) % of power savings.

work in this paper primarily involved the software simulation of the proposed techniques for bus power minimization considering performance overhead. As far as future work, we will continue to evaluate the proposed approach with lower process node (14 and 10 nm) for reliability especially with new process.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. R. Stan and K. Skadron, "Power-aware computing," *IEEE Computer*, vol. 36, no. 12, pp. 35–38, 2003.

[2] N. Hatta, N. D. Barli, C. Iwama et al., "Bus serialization for reducing power consumption," *Proceedings of SWoPP*, 2004.

[3] B. Jacob and V. Cuppu, "Organizational design trade-offs at the DRAM, memory bus and memory controller level: initial results," Tech. Rep. UMD-SCA-TR-1999-2, University of Maryland Systems & Computer Architecture Group, 1999.

[4] Rambus Inc, *Rambus Signaling Technologies: RSL, QRSL and SerDes Technology Overview*, Rambus Inc, 2000.

[5] M. Loghi, M. Poncino, and L. Benini, "Cycle-accurate power analysis for multiprocessor systems-on-a-chip," in *Proceedings of the ACM Great lakes Symposium on VLSI*, pp. 401–406, April 2004.

[6] K. Mohanram and S. Rixner, "Context-independent codes for off-chip interconnects," in *Power-Aware Computer Systems*, vol. 3471 of *Lecture Notes in Computer Science*, pp. 107–119, 2005.

[7] D. C. Suresh, B. Agrawal, W. A. Najjar, and J. Yang, "VALVE: variable Length Value Encoder for off-chip data buses," in *Proceedings of the International Conference on Computer Design (ICCD '05)*, pp. 631–633, San Jose, Calif, USA, October 2005.

[8] M. R. Stan and W. P. Burleson, "Coding a terminated bus for low power," in *Proceedings of the 5th Great Lakes Symposium on VLSI*, pp. 70–73, March 1995.

[9] K. Basu, A. Choudhury, J. Pisharath, and M. Kandemir, "Power protocol: reducing power dissipation on off-chip data buses," in *Proceedings of the 35th Annual ACM/IEEE International Symposium on Microarchitecture*, pp. 345–355, 2002.

[10] N. R. Mahapatra, J. Liu, K. Sundaresan, S. Dangeti, and B. V. Venkatrao, "A limit study on the potential of compression for improving memory system performance, power consumption, and cost," *Journal of Instruction-Level Parallelism*, vol. 7, pp. 1–37, 2005.

[11] A. Park and M. Farrens, "Address compression through base register caching," in *Proceedings of the Annual ACM/IEEE International Symposium on Microarchitecture*, pp. 193–199, November 1990.

[12] M. Farrens and A. Park, "Dynamic base register caching: a technique for reducing address bus width," in *Proceedings of the 18th International Symposium on Computer Architecture*, pp. 128–137, May 1991.

[13] D. Citron and L. Rudolph, "Creating a wider bus using caching techniques," in *Proceedings of the International Symposium on High Performance Computer Architecture*, pp. 90–99, January 1995.

[14] K. Sunderasan and N. Mahapatra, "Code compression techniques for embedded systems and their effectiveness," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, pp. 262–263, February 2003.

[15] L. Li, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, and I. Kadayif, "CCC: crossbar connected caches for reducing energy consumption of on-chip multiprocessors," in *Proceedings of the Euromicro Symposium on Digital Systems Design (DSD '03)*, 2003.

[16] P. P. Sotiriadis and A. P. Chandrakasan, "A bus energy model for deep submicron technology," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 3, pp. 341–349, 2002.

[17] P. P. Sotiriadis and A. Chandrakasan, "Low power bus coding techniques considering inter-wire capacitances," in *Proceedings of the IEEE 22nd Annual Custom Integrated Circuits Conference (CICC '00)*, pp. 507–510, May 2000.

[18] J. Henkel and H. Lekatsas, "A2BC: adaptive address bus coding for low power deep sub-micron designs," in *Proceedings of the IEEE 38th Design Automation Conference*, pp. 744–749, June 2001.

[19] T. Lindkvist, "Additional knowledge of bus invert coding schemes," in *Proceedings of the IEEE 5th International Workshop on System-on-Chip for Real-Time Applications (IWSOC '05)*, pp. 301–303, Alberta, Canada, July 2005.

[20] T. Lindkvist, J. Löfvenberg, and O. Gustafsson, "Deep submicron bus invert coding," in *Proceedings of the 6th Nordic Signal Processing Symposium (NORSIG '04)*, pp. 133–136, Espoo, Finland, June 2004.

[21] K.-W. Kim, K.-H. Baek, N. Shanbhag, C. L. Liu, and S.-M. Kang, "Coupling-driven signal encoding scheme for low-power interface design," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 318–321, San Jose, Calif, USA, 2000.

[22] S. Komatsu, M. Ikeda, and K. Asada, "Bus power encoding with coupling-driven adaptive code-book method for low power data transmission," in *Proceedings of the European Solid-State Circuits Conference*, 2001.

[23] J.-H. Chern, J. Huang, L. Arledge, P.-C. Li, and P. Yang, "Multilevel metal capacitance models for CAD design synthesis systems," *Electron Device Letters*, vol. 13, no. 1, pp. 32–34, 1992.

[24] K. Mohammad, A. Dodin, B. Liu, and S. Agaian, "Reduced voltage scaling in clock distribution networks," *VLSI Design*, vol. 2009, Article ID 679853, 7 pages, 2009.

[25] K. Mohammad, B. Liu, and S. Agaian, "Energy efficient swing signal generation circuits for clock distribution networks systems," in *Proceedings of the IEEE International Conference on Man and Cybernetics*, pp. 3495–3498, 2009.

[26] K. Mohammad, S. Agaian, and F. Hudson, "Efficient FPGA implementation of convolution," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, Tex, USA, October 2009, paper ID 3922.

[27] "International Technology Roadmap for Semiconductors," http://www.itrs.net.

[28] H. Kawaguchi and T. Sakurai, "Delay and noise formulas for capacitively coupled distributed RC lines," in *Proceedings of the 3rd Conference of the Asia and South Pacific Design Automation (ASP-DAC '98)*, pp. 35–43, February 1998.

[29] C.-L. Su, C.-Y. Tsui, and A. M. Despain, "Saving power in the control path of embedded processors," *IEEE Design and Test of Computers*, vol. 11, no. 4, pp. 24–30, 1994.

[30] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on VLSI Systems*, vol. 3, no. 1, pp. 49–58, 1995.

[31] L. Benini, G. de Micheli, E. Macii, D. Sciuto, and C. Silvano, "Asymptotic zero-transition activity encoding for address busses in low-power microprocessor-based systems," in *Proceedings of the 7th Great Lakes Symposium on VLSI*, pp. 77–82, March 1997.

[32] C. Liu, A. Sivasubramaniam, and M. Kandemir, "Optimizing bus energy consumption of on-chip multiprocessors using frequent values," in *Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network-based Proceedings (PDP '04)*, pp. 340–347, February 2004.

[33] J. Yang and R. Gupta, "Frequent value locality and its applications," *ACM Transactions on Embedded Computing Systems*, vol. 1, no. 1, pp. 79–105, 2002.

[34] J. Yang, R. Gupta, and C. Zhang, "Frequent value encoding for low power data buses," *ACM Transactions on Design Automation of Electronic Systems*, vol. 9, no. 3, pp. 354–384, 2004.

[35] D. C. Suresh, B. Agrawal, J. Yang, and W. Najjar, "A tunable bus encoder for off-chip data buses," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 319–322, San Diego, Calif, USA, August 2005.

[36] W.-C. Cheng and M. Pedram, "Memory bus encoding for low power: a tutorial," in *Proceedings of the International Symposium on Quality Electronic Design (ISQED '01)*, p. 1999, 2001.

[37] T. Lang, E. Musoll, and J. Cortadella, "Extension of the working-zone-encoding method to reduce the energy on the microprocessor data bus," in *Proceedings of the IEEE International Conference on Computer Design*, pp. 414–419, October 1998.

[38] L. Benini, G. de Micheli, E. Macii, M. Poncino, and S. Quer, "System-level power optimization of special purpose applications: the beach solution," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 24–29, Monterey, Calif, USA, August 1997.

[39] L. Benini, G. DeMicheli, E. Macii, M. Poncino, and C. Silvano, "Address bus encoding techniques for system level power optimization," in *Proceeding of the Design Automation and Test in Europe*, pp. 861–866, Paris, France, February 1998.

[40] N. Chang, K. Kim, and J. Cho, "Bus encoding for low-power high-performance memory systems," in *Proceedings of the 37th Design Automation Conference (DAC '00)*, pp. 800–805, June 2000.

[41] W.-C. Cheng and M. Pedram, "Power-optimal encoding for DRAM address bus," in *Proceedings of the Symposium on Low Power Electronics and Design (ISLPED '00)*, pp. 250–252, July 2000.

[42] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, "A coding framework for low-power address and data busses," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 7, no. 2, pp. 212–221, 1999.

[43] E. Musoll, T. Lang, and J. Cortadella, "Exploiting the locality of memory references to reduce the address bus energy," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 202–207, Monterey, Calif, USA, August 1997.

[44] Y. Shin, S.-I. Chae, and K. Choi, "Partial bus-invert coding for power optimization of system level bus," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 127–129, August 1998.

[45] M. R. Stan and W. P. Burleson, "Two-dimensional codes for low power," in *Proceedings of the International Symposium on Low Power Electronics and Design*, pp. 335–340, August 1996.

[46] S. Yoo and K. Choi, "Interleaving partial bus-invert coding for low power reconfiguration of FPGAs," in *Proceedings of the 6th International Conference on VLSI and CAD*, pp. 549–552, 1999.

[47] C. Lee, M. Potkonjak, and W. H. Mangione-Smith, "MediaBench: a tool for evaluating and synthesizing multimedia and communications systems," in *Proceedings of the 30th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 330–335, December 1997.

[48] SimpleScalar Simulator, "SimpleScalar LLC," http://www.simplescalar.com/.

[49] SPEC, "SPEC CPU2000 Benchmark Suite Ver 1.2," http://www.spec.org/osg/cpu2000/.