International Journal of Advanced Research and Review

www.ijarr.org

A Low-Power High-Speed Accuracy-Controllable Approximate Multiplier Design

Mohammed Numan[1],Tirumani Eshwar Prakash [2],Syed Abdul Rehman[3], Mohammed Mustafa Adnan [4]

Assistant professor- Mr. B. Sai Krishna

Student, Department of Electronics and Communication Engineering, Lords Institute of Engineering and Technology,

Associate professor, Department of Electronics and Communication Engineering, Lords Institute of Engineering and Technology, Hyderabad, Telangana, India-500091

Abstract

In CMOS-based application-specific integrated circuit (ASIC) designs, total power consumption is dominated by dynamic power, where dynamic power consists of two major components, namely, switching power and internal power. In this paper, we present a low-power design for a accuracy-controllable multiplier. Multiplication is a key fundamental function for many error-tolerant applications. Approximate multiplication is considered to be an efficient technique for trading off energy against performance and accuracy. This paper proposes an accuracy-controllable multiplier whose final product is generated by a carry-maskable adder. The proposed scheme can dynamically select the length of the carry propagation to satisfy the accuracy requirements flexibly. The partial product tree of the multiplier is approximated by the proposed tree compressor. An multiplier design is implemented by employing the carry maskable adder and the compressor. Compared with a conventional multiplier, the proposed multiplier reduced power consumption. The implementation, synthesis and simulation is executed and noted in the Xilinx-ISE in verilog HDL language.

I. INTRODUCTION

Many increasingly popular applications, such as image processing and recognition, are inherently tolerant of small inaccuracies. These applications are computationally demanding and multiplication is their fundamental arithmetic function, which creates an opportunity to trade off computa' · accuracy for reduced power consumption. Approximate computing is an efficient approach for error tolerant applications because it can trade off accuracy for power, and it currently plays an important role in such application domains [1].

Different error-tolerant applications have different accuracy requirements, as do different program phases in an application. If multiplication accuracy is fixed, power will be wasted when high accuracy is not required. This means that approximate multipliers should be dynamically reconfigurable to match the different accuracy requirements of different program phases and applications.

This paper focuses on an approximate multiplier design that can control accuracy dynamically. A carry-maskable adder (CMA) is proposed that can be dynamically configured to function as a conventional carry propagation adder (CPA), a set of bit-parallel OR gates, or a combination of the two. This configurability is realized by masking carry propagation: the CPA in the last stage of the multiplier is replaced by the proposed CMA. An approximate tree compressor is utilized to reduce the accumulation layer depth of the partial product tree.

Our approach introduces a term representing the power and accuracy requirements which simplifies the partial product reduction (PPR) component as needed. An approximate multiplier is designed using the proposed adder and compressor. This multiplier, together with a conventional multiplier and the previously studied approximate multipliers, was implemented in Verilog HDL using a 45-nm library to evaluate the power consumption, critical path delay, and design area. Compared with the conventional Wallace tree multiplier, the proposed approximate multiplier reduced power consumption by between 47.3% and 56.2% and the critical path delay by between 29.9% and 60.5%, depending on the required computational accuracy. In addition, its design area was 44.6% smaller. Comparisons with the established approximate multipliers, none of which have any dynamic reconfigurability, demonstrate that the proposed multiplier provided the best trade-off of power and delay against accuracy. All the multiplier designs are then evaluated in a real image processing application.

The remainder of this paper is organized as follows. Section II reviews previous works. Section III introduces the accuracy controllable approximate multiplier after explaining the tree compressor and the CMA. Section IV evaluates the multipliers experimentally and then evaluates the proposed approximate multiplier using an image processing application. Section V presents our conclusions.

II. PREVIOUS WORK

The adder is a basic element of most multipliers. Mahdiani et al. [2] proposed the lower-part-OR adder, which utilizes OR gates for addition of the lower bits and precise adders for addition of the upper bits. It is similar to our proposed CMA in that it uses OR gates to generate the sum approximately, but our CMA is also dynamically reconfigurable.

Liu et al. [3] utilized an approximate adder to reduce carry propagation delay in partial product accumulation. They also proposed a recovery vector to improve accuracy. The bit width of the error recovery vector can be selected by the designer to satisfy accuracy requirements. Hashemi et al. [4] proposed a technique that reduces the size of the multiplier by detecting the leading one bit of the input operands and selecting the following bits as abridged operands for both inputs, where is a designer-defined value that specifies the bandwidth used in the core accurate multiplier. Both [3] and [4] allow a static tradeoff between power consumption and accuracy. The bit lengths of the recovery vector [3] and the input operands [4] are determined during the design process and the accuracy is not dynamically controllable, unlike with our proposed multiplier. Moons et al. [5] proposed a system-level technique that disables part of the combinational logic and reconfigures the pipelined registers and combinational logic. It can trade off accuracy for power dynamically by changing the numbers of pipeline stages and voltage-accuracy scaling modes. Our proposed multiplier also disables part of the combinational logic in the CPA to achieve lower power consumption, but ours does not require a pipeline system or control circuits for voltage scaling.

III. SCOPE OF THE PROJECT

The project titled "A Low Power, High Speed, Accuracy-Controllable Approximate Multiplier Design" aims to develop an arithmetic multiplier that effectively balances power efficiency, processing speed, and computational accuracy. The core objective is to design an approximate multiplier that allows users to control the level of accuracy based on the specific requirements of the application, making it highly suitable for error-tolerant and energy-constrained environments. This involves implementing various approximation techniques, such as truncation and speculative computation, to reduce logic complexity and switching activity, thereby achieving low power consumption and high-speed performance. The design will be evaluated through simulations using tools like Verilog, VHDL, MATLAB, or industry-standard EDA tools such as Xilinx and Synopsys. The project also includes a detailed analysis of the trade-offs between accuracy and key performance metrics like delay, power, and area. The proposed design will be benchmarked against traditional and existing approximate multipliers to highlight its advantages. Targeted applications include image processing, machine learning, digital signal processing, and IoT systems where a certain degree of computational error is tolerable. Additionally, the design will be made

modular and scalable to support various bit-widths and facilitate easy integration into larger digital systems.

IV. ACCURACY-CONTROLLABLE MULTIPLIER

A typical multiplier consists of three parts: (i) partial product generation using an AND gat PPR using an adder tree; and (iii) addition to produce the final result using a CPA. Power consumption and circuit complexity are dominated by the PPR [6], and the multiplier's critical path is dominated by the propagated carry chain in the CPA [7].

This section is organized as follows. Section III-A explains how the partial product layer is simplified by the approximate tree compressor. Section III-B introduces the CMA. Finally, Section III-C presents the overall structure of the accuracycontrollable approximate multiplier, which uses the proposed adder and tree compressor.

A. Approximate Tree Compressor

Figure 1(a) shows an accurate half adder, for which the following equation can be obtained:

where {,} and + denote concatenation and addition, respectively. The value c is generated by a b and s is generated by a b, so () can be generated by a b. Based on the above, consider the basic logic cell shown in Fig. 1(b), for which the following equations can be obtained:

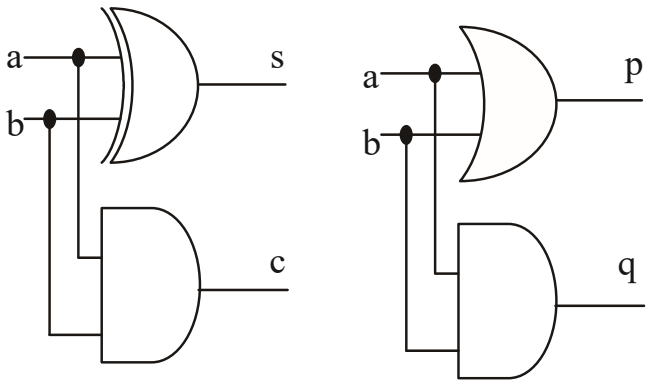
This is called an incomplete adder cell (iCAC). Table I shows the truth tables for an accurate half adder and an iCAC. Note that the bit position of c and that of s, p, and q are different. As can be seen, q is equal to c. While p is not equal to s, the precise sum can be ' 'ained by adding p and q, so the iCAC is not an approximate adder but an element of a precise a

By extending the above equation to bits, the following equation can be obtained:

where A, B, P, and Q are -bit values, the bits of which correspond to a, b, p, and q, respectively. A row of eight iCACs, used for 8-bit inputs, is shown in Fig. 2. Consider the example of an 8-bit adder with the two inputs A = 01011111 and B = 00110110. The accurate sum S is 10010101, while the row of iCACs produces P = 01111111 and Q = 00010110. Again, it is evident that the following holds:

(1)

While S is obtained from P and Q, P can be used as an approximation for S, and Q can be used as an error recovery vector for the approximate sum P.



(a) (b)

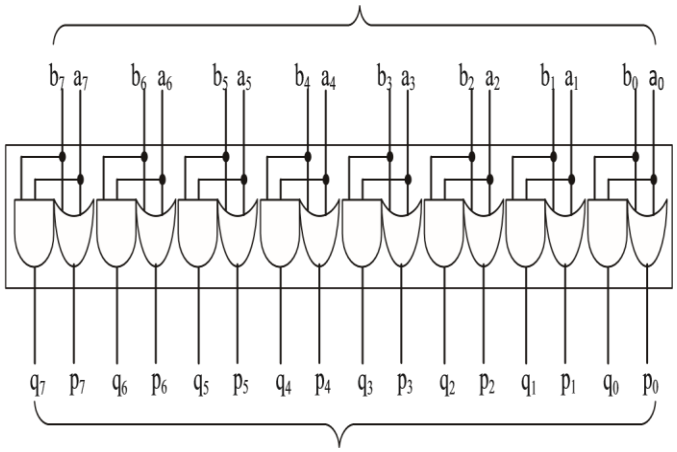
Fig. 1. (a) Accurate half adder and (b) incomplete adder cell.

TABLE I. TRUTH TABLES FOR ACCURATE HALFADDERAND INCOMPLETE ADDER CELL.

Inputs		Outputs			
		Accurate half adder		iCAC	
a	b	c	s	q	p
0	0	0	0	0	0
0	1	0	1	0	1
1	0	0	1	0	1
1	1	1	0	1	1

Two 8-bit inputs :

A = {a7, a6, a5, a4, a3, a2, a1, a0} B = {b7, b6, b5, b4, b3, b2, b1, b0}



Two 8-bit outputs :

Approximate sum : P = {p7, p6, p5, p4, p3, p2, p1, p0}

Error recovery vector : Q = {q7, q6, q5, q4, q3, q2, q1, q0}

Fig. 2. A row of incomplete adder cells with two 8-bit inputs.

By extending the row of iCACs from two to inputs, /2 Ps and /2 Qs are obtained. If the sum of the /2 Qs is used instead of the /2 Qs themselves, the number of Qs is reduced to one. Remember that P is always greater than or equal to S, and Q is equal to C. By exploiting these facts, OR gates can be used to generate the approximate sum of the /2 Qs without significant loss of accuracy. This approximate sum is called the accuracy compensation vector and is denoted by V. This method is named approximate tree compressor (ATC). An ATC with inputs is called an ATC- , and the structure of an ATC with eight inputs (ATC-8) is shown in Fig. 3. The rectangles represent rows of iCACs and the number of iCACs in each row (rectangle) is dependent on the bit width of the inputs. For example, if there are eight -bit inputs (D1, D2, …, D8), four rows of iCACs are required to build a -bit ATC-8. This reconstruction generates four approximate sums, P1, P2, P3, and P4, and four error recovery vectors, Q1, Q2, Q3, and Q4. OR gates generate the accuracy compensation vector V. As a result, the eight inputs have been reduced to five.

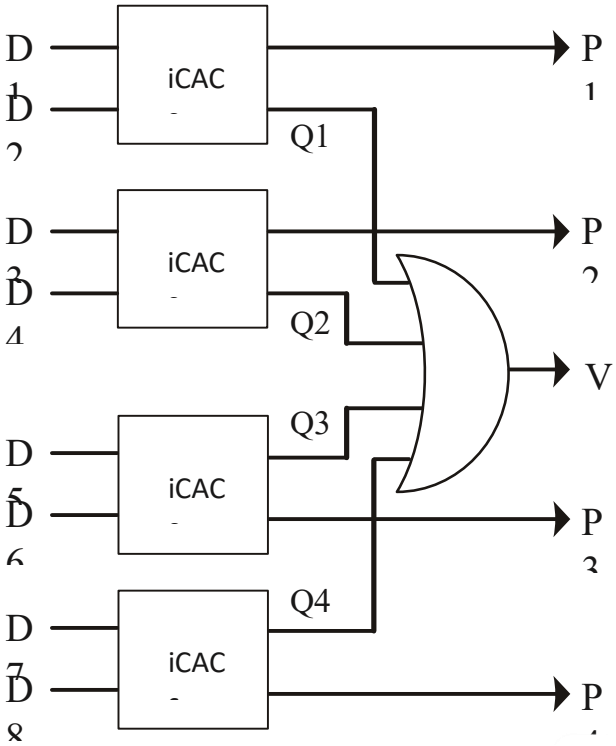


Fig. 3. Structure of an approximate tree compressor with eight inputs.

B. Carry-maskable Adder

A CMA is proposed to control the accuracy flexibly and dynamically. A -bit CMA comprises (!1) carry-maskable full adders and one carry-maskable half adder, and its structure is similar to that of a -bit CPA.

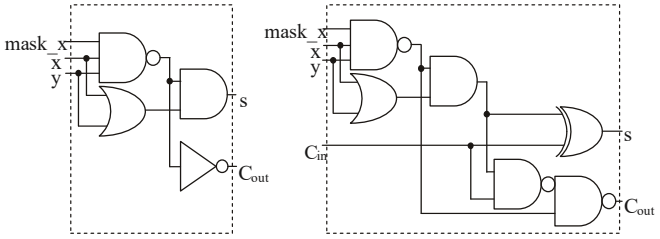
The structures of the proposed carry-maskable half and full adders are shown in Fig. 4. In the proposed half adder, when mask_x is 0, S is equal to x y and C_{out} is equal to 0. Otherwise, when mask_x is 1, S is equal to x b and C_{out} is equal to x y. In other words, the operation of the proposed half adder can be controlled by the active-low signal mask_x. When mask_x is disabled (=1), it functions as an accurate half adder, and when mask_x is enabled (=0), C_{out} is masked to 0 and it functions as an OR gate with output S. The operation of the proposed full adder is similar to the half adder: when mask_x is disabled (=1), it functions as an accurate full adder, and when mask_x is enabled (=0), C_{out} is equal to C_{in} and S is the output of an OR gate

C. Overall Structure

An -bit multiplier consists of " rows, each of which has partial products (PP), so there are PPs in total. Using the ATC- introduced in the previous section, the rows can be replaced by /2# rows. Figure 5 shows an example of an 8-bit multiplier with $$ PPs. The PPR is performed in three stages (Stage 1, Stage 2, and Stage 3) and the CPA is performed in Stage 4. The PP generation step is not shown. Each dot represents a PP. The least significant bit (right side) is bit 0, and the most significant bit (left side) is bit 14. The solid rectangles in Stage 1 represent ATCs and the dashed rectangles represent rows of seven iCACs. Every row of iCACs includes PPs that are not processed: for example, the PP at position 0 in the first row and the one at position 8 in the second row of the first iCAC block in ATC-8 are not processed.

In Stage 1, eight rows of PPs are reduced to four rows (P1, P2, P3, and P4) and one accuracy compensation vector (V1) by an ATC-8. The four rows are further reduced to two rows (P5 and P6) and another accuracy compensation vector (V2) by an ATC-4. A final row of iCACs then processes P5 and P6 and generates P7 and Q7. In summary, Stage 1 uses an ATC-8, an ATC-4, and a row of seven iCACs

to compress the $$ PPs to four rows (P7, V1, V2, and Q7).



(a) (b)

Fig. 4. (a) Carry-maskable half adder, (b) Carry-maskable full adder.

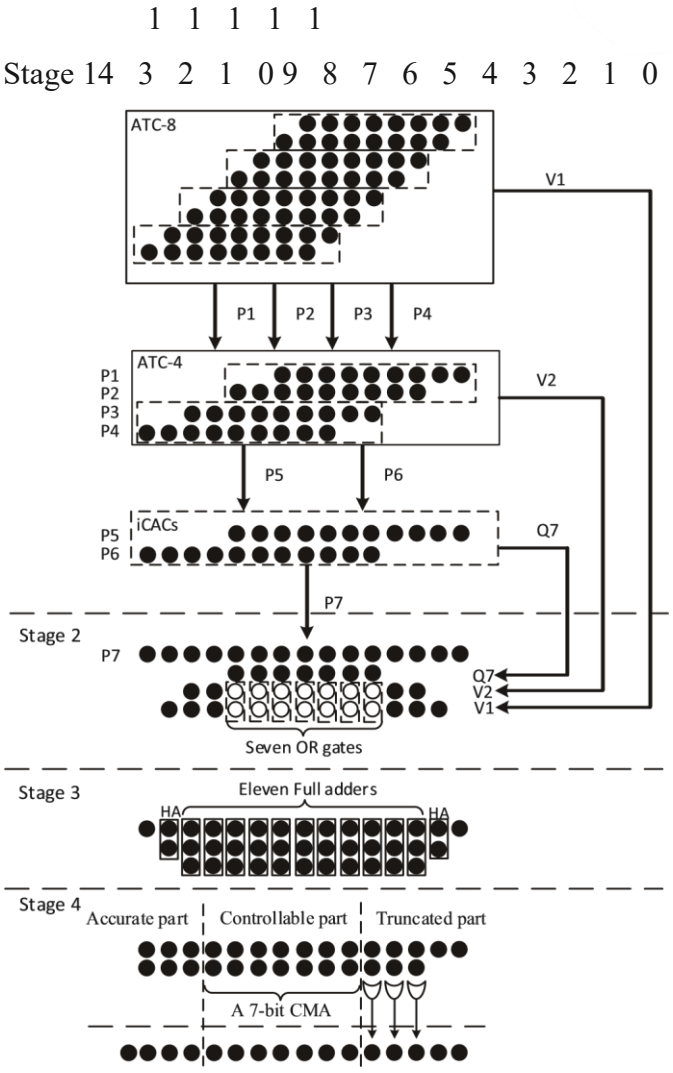


Fig. 5. Structure of an 8-bit multiplier with $$ partial products.

In Stage 2, there are four PPs for each of bits 4 to 10. In order to achieve a lower path delay, OR gates are used to sum V1 and V2 approximately. The empty circles for V1 and V2 represent the bits which are summed using OR gates. Seven OR gates are required in total and the four rows are compressed to three.

In Stage 3, full adders and half adders are used to [illegible]ws to two. Two half adders are required for bits 1 and 13, and eleven full adders are required for bits 2 to 12.

Addition using a CPA is required after PPR to produce the final result. For an 8-bit Wallace tree multiplier, the length of the CPA is 11 [7]. In our proposed multiplier, the length of the CPA is 13. In Stage 4, the CPA is divided into three parts in order to reduce the length of the carry propagation. Since the lower bits are not significant for accuracy, bits 0 to 4 are defined as the truncated part and three OR gates are used to generate the values for bits 2, 3, and 4 of the final result. Because there is no carry out from the truncated part, the length of the CPA is reduced to 10. Since the upper bits are the most significant for accuracy, bits 12 to 14 are defined as the accurate part, and three accurate adders are used to generate the values for these bits of the final result.

The accuracy-controllable part lies between the truncated and accurate parts. This part is important for both critical path delay and accuracy. In Stage

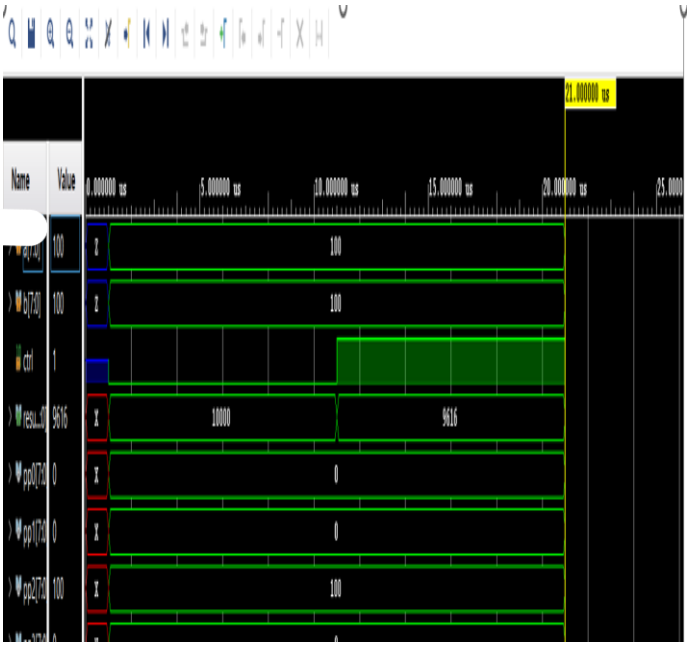
4, bits 5 to 11 in the CPA are replaced by a 7-bit CMA. Note that every 1-bit CMA has a mask_x signal. Given a value for %, the % upper bits in the accuracy-controllable part are configured as a %-bit CPA and the lower bits are configured as (&!%) 2-input OR gates by managing the seven mask_x signals appropriately. When % = 7, it functions as a 7-bit CPA, and when % = 0, it functions as seven 2-input OR gates. For each bit of S that is generated by a 2-input OR gate, power consumption is reduced because the switching activity is reduced in some of the logic gates. Furthermore, the maximum delay of the CMA is reduced.

V. Simulation

The simulation is the process which is termed as the final verification in respect to its working whereas the schematic is the verification of the connections and blocks. The simulation window is launched as shifting from implantation to the simulation on the home screen of the tool, and the simulation window confines the output in the form of the wave forms. Here it has the flexibility of providing the different radix number systems.

Waveform & Output Validation

- Partial products generated correctly
- Approximation logic functioning as expected
- CPA outputs verified against expected product values
- Errors confined to LSBs as designed
- Configurable logic toggles accuracy on/off dynamically



VI Result:

The result of this project is the successful design and simulation of an approximate multiplier that offers a configurable trade-off between power, speed, and accuracy. The implemented design demonstrates:

- Reduced Power Consumption: Through the use of approximation techniques like truncation and speculative computation, the multiplier consumes significantly less power compared to traditional exact multipliers.
- Improved Speed: Due to a simplified logic structure and reduced gate delay, the design achieves faster computation speeds, making it ideal for real-time and high-performance applications.

- Accuracy Control: The system allows users to adjust the level of computational accuracy, enabling optimization for either precision or efficiency depending on the application's need.
- Resource Efficiency: Synthesis results (from tools like Xilinx or Synopsys) show a decrease in area utilization (logic elements or gates) while maintaining acceptable error margins.
- Application Readiness: The design proves suitable for integration into low-power applications like image processing, IoT devices, and AI edge computing

VII. Conclusion:

This paper proposes an accuracy-configurable approximate multiplier, which consumes less power than the Wallace tree multiplier and the previously implemented accuracy- scalable multiplier do, respectively. The reduction ratio depends on the required accuracy. In addition, the area of the implemented multiplier is smaller than the area of those multipliers. The lower power and the smaller area are realized by borrowing the concepts of ATC and CMA with careful modifications. At the cost of accuracy, the simple circuit of the basic OR gates is chosen to generate the accuracy compensation vector. This accuracy-configurable approximate multiplier will be beneficial for applications that prioritize power and area rather than accuracy and suffer from fluctuations under working conditions, such as IoT devices.

VIII Future scope:

The future scope of the project titled "A Low Power, High Speed, Accuracy-Controllable Approximate Multiplier Design" includes significant opportunities for advancement and application. The proposed multiplier can be integrated into larger computing systems such as digital signal processors, AI accelerators, and embedded platforms where energy efficiency is critical. Future developments may include dynamic accuracy scaling, where the system can automatically adjust its accuracy level based on the operational context, thereby improving performance in real-time. The design can be further optimized for ASIC and FPGA implementations to enable use in commercial and industrial products. Additionally, it holds great potential for edge computing, IoT, and wearable devices, where low power consumption is essential, and minor accuracy loss is acceptable. Research can also explore the incorporation of intelligent error prediction or correction mechanisms to enhance the reliability of approximate computing. Overall, the project lays a strong foundation for future innovations in energy-efficient and high-performance digital systems.

REFERENCES

[1] S. Venkataramani, V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan. "Quality programmable vector processors for approximate computing," *46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),* pp. 1-12, Dec. 2013.

[2] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-Inspired imprecise computational blocks for efficient VLSI implementation of Soft-Computing applications," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 57, no. 4, pp. 850-862, Apr. 2010.

[3] C. Liu, J. Han, and F. Lombardi, "A Low-Power, High-Performance approximate multiplier with configurable partial error recovery," *Design, Automation & Test in Europe Conference & Exhibition (DATE),* Mar. 2014.

[4] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A Dynamic Range Unbiased Multiplier for approximate applications," *IEEE/ACM International Conference on Computer-Aided Design (ICCAD),* pp. 418425, Nov. 2015.

[5] B. Moons, M. Verhelst, "DVAS: Dynamic Voltage Accuracy Scaling for increased energy-efficiency in approximate computing," *IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED),* Jul. 2015.

[6] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approxiı compressors for multiplication," *IEEE Transactions on Computers,* vol. 64, no. 4, pp. 984-994, Apr. 2015.

[7] K. C. Bickerstaff, E. E. Swartzlander, and M. J. Schulte, "Analysis of column compression multipliers," *15th IEEE Symposium on Computer Arithmetic,* pp. 33-39, Jun. 2001.

[8] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for ErrorResilient multiplier design," *IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS),* pp. 183-186, Oct. 2015.

[9] NanGate, Inc. NanGate FreePDK45 Open Cell Library, http://www.nangate.com/?page_id=2325, 2008

[10] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," *IEEE Transactions on computers,* vol. 62, no. 9, pp. 1760-1771, Sep. 2013.

[11] M. S. Lau, K. V. Ling, and Y. C. Chu, "Energy-Aware probabilistic multiplier: Design and Analysis," *2009 international Conferrence on Compliers, architeture, and synthesis for embedded systems,* pp. 281-290, Oct. 2009.