IJIREEICE

ISSN (Online) 2321-2004
ISSN (Print) 2321-5526

International Journal of Innovative Research in
Electrical, Electronics, Instrumentation and Control Engineering

Vol. 7, Issue 4, April 2019

Low - Power and Error Tolerant Multi - Precision Approximate Multipliers using Voltage and Frequency Management Unit

Dr.V.Suresh Babu[1], V.Amrutha[2]

Professor, Department of ECE, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India[1]

PG Scholar, Hindusthan Institute of Technology, Coimbatore, Tamilnadu, India[2]

Abstract: Approximate computing is an emerging trend in digital design that trades off the requirement of exact computation for improved speed and power performance. The proposed method uses a novel approximate compressors and an algorithm to exploit them for the design of efficient approximate multipliers. The approximate compressors are a key element in the design of power-efficient approximate multipliers, the number of faulty rows in the compressor's truth table is significantly reduced by encoding its inputs using generate and propagate signals. Further it is converted to Multi Precision (MP) reconfigurable multiplier that incorporates variable precision, Parallel Processing (PP), razor-based Dynamic Voltage Scaling (DVS), and dedicated MP operands scheduling to provide optimum performance for a variety of operating conditions. All of the building blocks of the proposed reconfigurable multiplier can either work as independent smaller-precision multipliers or work in parallel to perform higher-precision multiplications. Given the user's requirements (e.g., throughput), a dynamic voltage/frequency scaling management unit configures the multiplier to operate at the proper precision and frequency. Based on this improved compressor, two 4×4 multipliers are designed with different accuracies and then are used as building blocks for scaling up to 16×16 and 32×32 multipliers. Comparison with previously presented approximated multipliers shows that the proposed circuits provide better power or speed for a target precision.

Keywords: Approximate computing, Compressors, Multi-precision, Image processing

I. INTRODUCTION

Approximate computing is an emerging trend in digital design [1], [2], relaxing the requisite of exact computation to gain substantial performance improvement in terms of power, speed and area. This approach is becoming more and more important for embedded and mobile systems, characterized by severe energy and speed constraints. Approximate computing can be fruitfully applied in several error-resilient applications. Examples are multimedia processing [3], data mining and recognition, machine learning. Multipliers are fundamental subsystems for microprocessors, digital signal processors, and embedded systems with applications ranging from filtering to convolutional neural networks. Unfortunately, multipliers are characterized by complex logic design [4] and constitute one of the most energy-hungry digital blocks. Therefore, approximate multiplier design has become an important research subject in recent years. A multiplier includes a few basic blocks: partial products generation, partial products reduction and carry-propagate addition. Approximations can be introduced in any of these blocks. For example, truncation of the partial products is a well established approximation technique in which some of the partial products are not formed and the truncation error is reduced with the help of suitable correction functions.

For embedded applications, it has become essential to design more power-aware multipliers. Given their fairly complex structure and interconnections, multipliers can exhibit a large number of unbalanced paths, resulting in substantial glitch generation and propagation. This spurious switching activity can be mitigated by balancing internal paths through a combination of architectural and transistor-level optimization techniques. In addition to equalizing internal path delays, dynamic power reduction can also be achieved by monitoring the effective dynamic range of the input operands so as to disable unused sections of the multiplier and/or truncate the output product at the cost of reduced precision. This is possible because, in most sensor applications, the actual inputs do not always occupy the entire magnitude of its word-length. For example, in artificial neural network applications, the weight precision used during the learning phase is approximately twice that of the retrieval phase. Besides, operations in lower precisions are the most frequently required. In contrast, most of today's full-custom DSPs and Application-Specific Integrated Circuits (ASICs) are designed for a fixed maximum word-length

DOI 10.17148/IJIREEICE.2019.7409

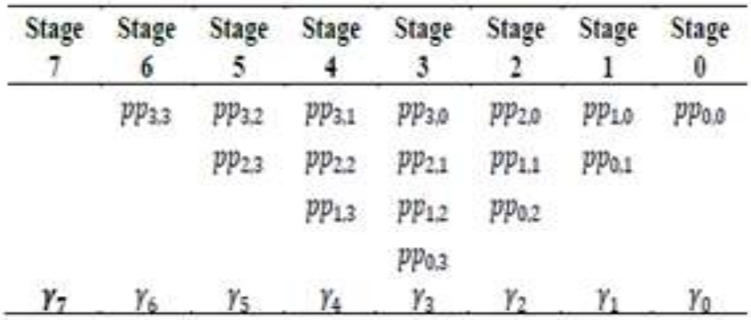
so as to accommodate the worst case scenario. Therefore, an 8-bit multiplication computed on a 32-bit Booth multiplier would result in unnecessary switching activity and power loss. In addition, the critical path may change as a result of the varying supply voltage or process or temperature variations. If this occurs, computations will completely fail regardless of the safety margins. The aforementioned limitations of conventional DVS techniques motivated recent research efforts into error-tolerant DVS approaches, which can run-time operate the circuit even at a voltage level at which timing errors occur. A recovery mechanism is then applied to detect error occurrences and restore the correct data. Because it completely removes worst case safety margins, error-tolerant DVS techniques can further aggressively reduce power consumption. The rest of the paper is organized as follows: Section II briefs about partial product generation and exact compressors. Section III explains the proposed multiplier designs in detail. Razor-based Multi-precision design with voltage and frequency management unit is explained in section IV. Results are evaluated in section V and section VI concludes the paper.

II. PARTIAL PRODUCT GENERATION AND EXACT COMPRESSION

The multiplication process consists of 3 steps:

- Partial product generation,
- Partial product reduction and
- Final carry propagating addition.

Various recoding schemes are used to reduce the number of partial products. Compressors have been widely used for reduction process which usually contributes the most to the delay, power and area of the multiplier. To achieve a better performance, the use of higher order compressors instead of conventional compressors, e.g. 3:2 compressors, have been considered. Fig.1 shows the partial product generation of 4X4 Multiplier.



Stage 7	Stage 6	Stage 5	Stage 4	Stage 3	Stage 2	Stage 1	Stage 0
	$pp_{3.3}$	$pp_{3.2}$	$pp_{3.1}$	$pp_{3.0}$	$pp_{2.0}$	$pp_{1.0}$	$pp_{0.0}$
		$pp_{2.3}$	$pp_{2.2}$	$pp_{2.1}$	$pp_{1.1}$	$pp_{0.1}$	
			$pp_{1.3}$	$pp_{1.2}$	$pp_{0.2}$		
				$pp_{0.3}$			
r_7	r_6	r_5	r_4	r_3	r_2	r_1	r_0

Fig.1 Partial Product of 4X4 Multiplier

The function of the exact 4:2 compressor is implemented by using two appropriately connected full adders is shown in Fig.2.

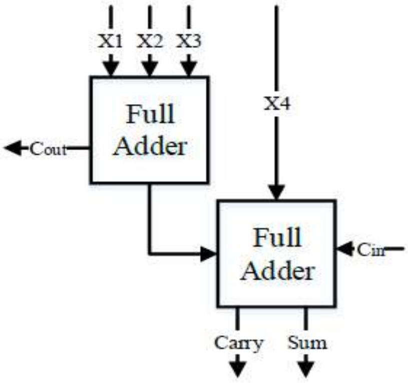


Fig.2. Exact 4:2 Compressors

sum = $x1 \oplus x2 \oplus x3 \oplus x4 \oplus$ cin,
cout = $(x1 \oplus x2).\ x3 + \overline{(x1 \oplus x2)}.\ x1$,
carry = $(x1 \oplus x2 \oplus x3 \oplus x4).$ cin + $\overline{(x1 \oplus x2 \oplus x3 \oplus x4)}.\ x4$. -------(1)

DOI 10.17148/IJIREEICE.2019.7409

The sum output has the same weight as the four input signals while the cout is used as the carry in for the next higher-order compressor and the output carry is weighted like a pp bit in a one-bit-higher position. Note that cout and carry have the same weight.

III. THE PROPOSED MULTIPLIER DESIGN

The proposed 4×4 approximate multipliers considers the carry from the previous stage (c4) and uses an exact full-adder to add pp terms pp3,2, pp2,3, and $c4$.Since c4 is generated from the four LSBs, it does not introduce a large error in an 8×8 multiplier. Note that ignoring c4 breaks the longest path (that is, the carry propagation) and it is a common technique to reduce the circuit's latency. The sixth sum output of the full adder in design are both denoted by $\gamma 5$ and the corresponding carry signal, $c5$, goes to the next stage to be added to pp3,3 using an exact half adder. The sum and carry outputs of this final half adder produce $\gamma 6$ and $\gamma 7$, respectively. Fig. 3 shows different blocks used for reducing the partial products. These blocks include: (1) half adders, (2) full adders, and (3) 4:2 compressors.

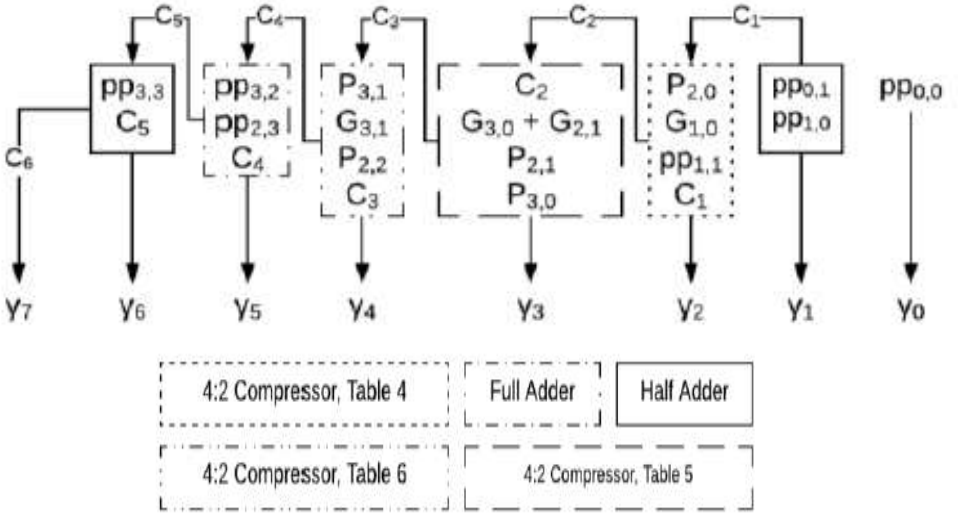


Fig.3. Partial product reduction in multiplier

The function of an exact 4:2 compressor can be approximated to reduce the hardware cost. It has been shown that cout does not have a significant impact on the compressor's accuracy, so cout is ignored in our design. Moreover, our SPICE simulations confirm that an XOR gate consumes more power and is slower than the AND and OR gates, Ignoring cout and not using XOR gates as well as our goal to use as few gates as possible led to the approximate compressor.

The sum and carry signals in the compressor for Stage 2 can be simplified as
sum =x1+x3, carry =x2+x4.------------(2)
The sum and carry signals in the compressor for Stage 3 can be simplified as
sum =x1+x3+x4, carry =x1.x2+x3.x4.----(3)
The sum and carry signals in the compressor for Stage 4 can be simplified as
sum =x1+x2+x3, carry =x2+x3.x4.-------(4)

In order to construct larger, e.g. 16×16 and 32×32, approximate multipliers, the two proposed 4×4 multipliers are combined in an array structure. For instance, to construct an 8×8 multiplier using a 4× 4 design, the two 8-bit operands A and B are partitioned into two 4-bit nibbles, namely αH and αL for A and βH and βL for B. Note that αH and βH are the 4 MSBs and αL and βL indicate the 4 LSBs of A and B, respectively. Each two of these four nibbles (in total 4 possible combinations) are multiplied using 4×4 multipliers and the partial products are then shifted (based on the nibble's importance) and added together (using a Wallace tree architecture) to produce the final multiplication result. Building 2n×2n multipliers using n×n multipliers is specified in Fig. 4.
The proposed approximate compressor can also be utilized in signed Booth multipliers. In a Booth multiplier, the partial products are generated using a Booth encoder, and the major difference between the unsigned and signed Booth multiplication is in the generation of the partial products. Therefore, the partial products in Booth multipliers can be accumulated using approximate compressors.

DOI 10.17148/IJIREEICE.2019.7409

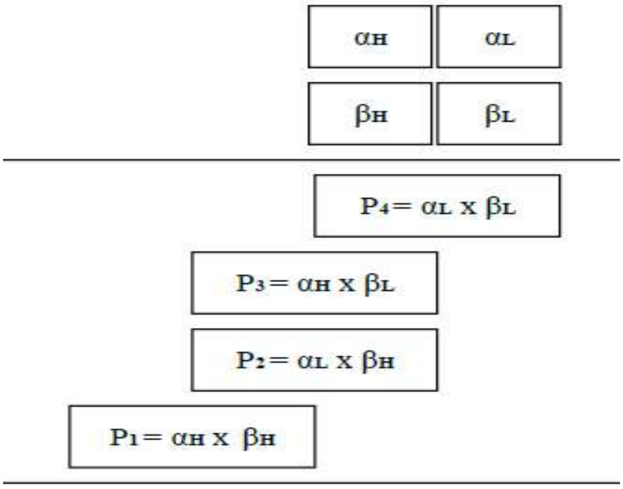


Fig.4. Building 2n×2n multipliers using n×n multipliers.

IV. RAZOR-BASED MULTIPRECISION DESIGN WITH VOLTAGE AND FREQUENCY MANAGEMENT UNIT

The MP multiplier system comprises five different modules that are as follows:
The MP multiplier;
The input operands scheduler (IOS) whose function is to reorder the input data stream into a buffer, hence to reduce the required power supply voltage transitions;
The frequency scaling unit implemented using a voltage controlled oscillator. its function is to generate the required operating frequency of the multiplier;
The voltage scaling unit implemented using a voltage dithering technique to limit silicon area overhead. Its function is to dynamically generate the supply voltage so as to minimize power consumption;
The dynamic voltage and frequency management unit that receives the user requirements.

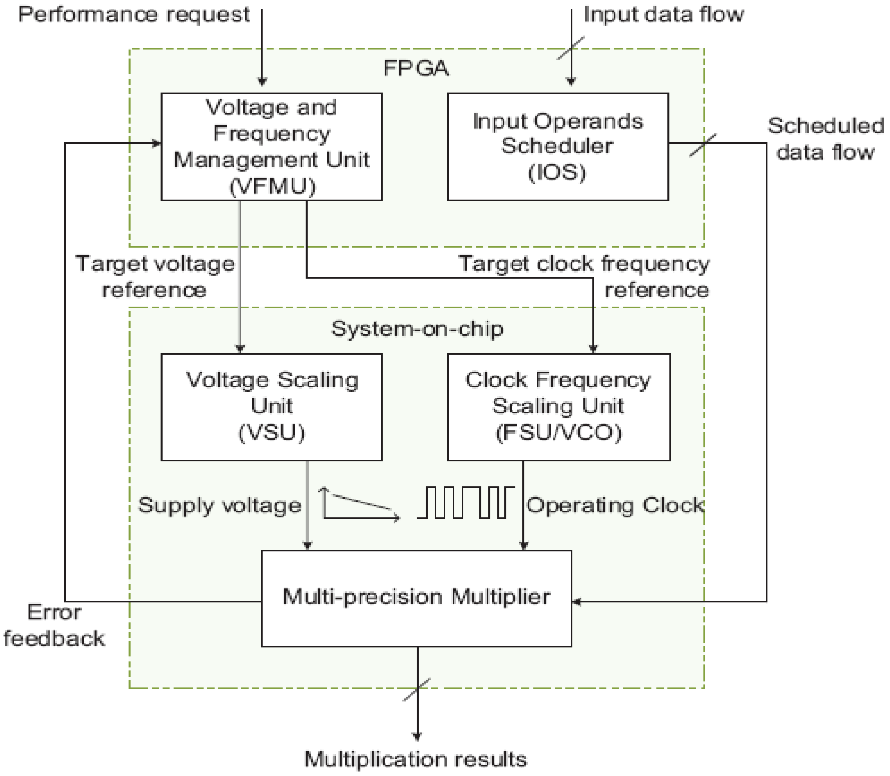

Performance request
Input data flow
FPGA
Voltage and Frequency Management Unit (VFMU)
Input Operands Scheduler (IOS)
Scheduled data flow
Target voltage reference
Target clock frequency reference
System-on-chip
Voltage Scaling Unit (VSU)
Clock Frequency Scaling Unit (FSU/VCO)
Supply voltage
Operating Clock
Multi-precision Multiplier
Error feedback
Multiplication results

Fig.5. Multi-Precision Multiplier with VFMU

In the 32 × 32 bit MP multiplier, dynamic frequency tuning is used to meet throughput requirements. This frequency range is selected to meet the requirements of general purpose DSP applications. The reported multiplier can operate as a 32-bit multiplier or as nine independent 8-bit multipliers.

A. Razor flip-flop

The razor flip-flop operates as a standard positive edge triggered flip-flops coupled with a shadow latch, which samples at the negative edge. Therefore, the input data is given in the duration of the positive clock phase to settle down to its correct state before being sampled by the shadow latch. The minimum allowable supply voltage needs to be set, hence the shadow latch (Fig. 6) always clocks the correct data even for the worst case conditions. This requirement is usually satisfied given that the shadow latch is clocked later than the main flip-flop.

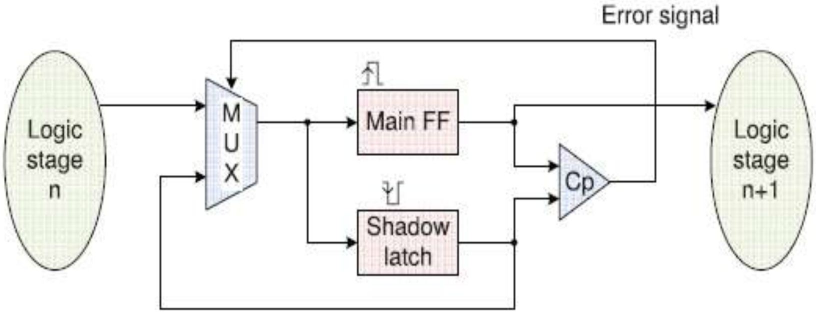


Fig.6. Architecture of razor flip-flop

A comparator flags a timing error when it detects a discrepancy between the speculative data sampled at the main flip-flop and the correct data sampled at the shadow latch. The correct data would subsequently overwrite the incorrect signal. The key idea behind razor flipflops is that if an error is detected at a given pipeline stage X, then computations are only re-executed through the following pipeline stage X + 1. This is possible because the correct sampled value would be held by the shadow latch. This approach ensures forward progress of data through the entire pipeline at the cost of a single-clock cycle. An error correction mechanism, based on global clock gating, is implemented in the proposed multiplier. In this correction scheme, error and clock signals are used to determine when the entire pipeline needs to be stalled for a single clock cycle. A global error signal is fed to the VFMU so as to alert the controlling unit whenever the current operating voltage is lower than necessary. The VFMU will then increase the voltage reference. This will in turn result in the VSU generating a new supply voltage level based on the new target voltage reference. When an error occurs, results can be recomputed at any pipeline stage using the corresponding input of the shadow latch. Therefore, the correct values can be forwarded to the corresponding next stages. Given that all stages can carry out these recomputations in parallel, the adopted global clock gating can tolerate any number of errors within a given clock cycle. After one clock cycle, normal pipeline operation can resume. The actual implementation of razor flip-flops requires careful design to meet timing constraints and avoid system failure.

V. EVALUATION RESULTS

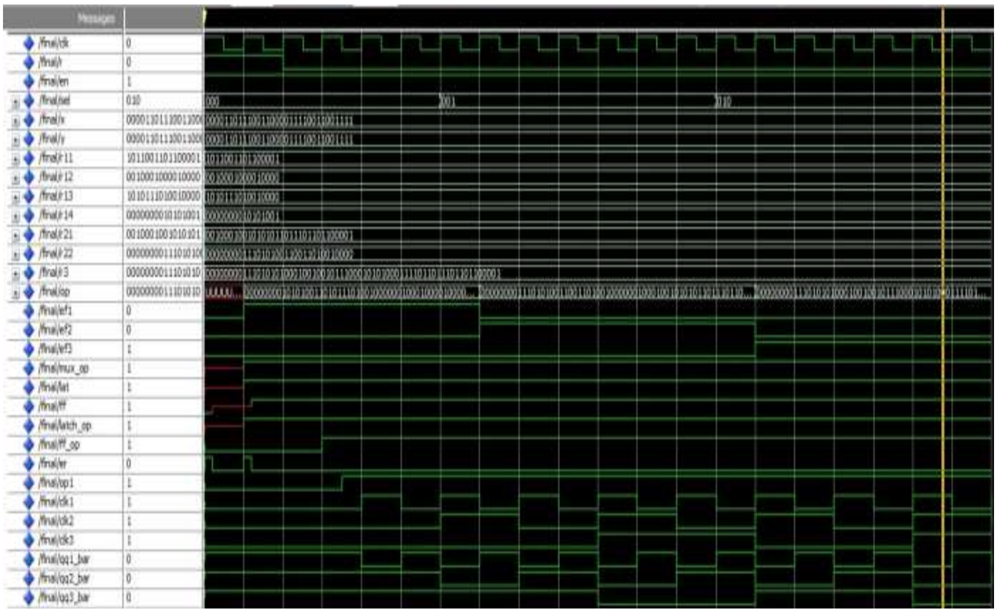
Table 1 show that various parameters like area, power and delay of 32 bit Approximate and Multiprecision multiplier. From the fundamental 4 bit approximate multiplier 8, 16 and 32 bit multiplier are computed. The 5th and 6th type results in low error and is used in image processing applications.

Table.1. Comparison Results of Multiplier

Parameters	Area (Gate Count)	Power (mW)	Delay(ns)
32-bit Approximate Multiplier	16167	479.18	33.806
32-bit Approximate Multiprecision Multiplier	3292	188.65	22.371

The proposed Multiprecision multiplier results in low power consumption and higher performance compared to conventional approximate multiplier. Fig 7 shows the simulation result of Multiprecision multiplier. Based on the operand selection 8 bit, 16 bit or 32 bit operation is performed. Voltage and frequency are generated based on the error feedback value. Clock divider is used to generate different frequencies for variable length operation.

DOI 10.17148/IJIREEICE.2019.7409



Messages	
/final/clk	0
/final/r	0
/final/en	1
/final/sel	010
/final/x	00001101110011000
/final/y	00001101110011000
/final/r11	1011001101100001
/final/r12	0010001000010000
/final/r13	1010111010010000
/final/r14	0000000010101001
/final/r21	0010001001010101
/final/r22	00000000111010100
/final/r3	0000000011101010
/final/op	0000000011101010
/final/ef1	0
/final/ef2	0
/final/ef3	1
/final/mux_op	1
/final/lat	1
/final/ff	1
/final/latch_op	1
/final/ff_op	1
/final/er	0
/final/op1	1
/final/clk1	1
/final/clk2	1
/final/clk3	1
/final/qq1_bar	0
/final/qq2_bar	0
/final/qq3_bar	0

Fig.7 Simulation Result of Multi-precision Multiplier

VI. CONCLUSION

A low power Multi-precision multiplier based on approximate 4:2 compressor is proposed. The two 4×4 multipliers with different accuracies are constructed. The 4×4 designs are then scaled up to 16×16 and 32×32 multipliers that provide a wide range of accuracy-performance trade-offs. The proposed novel dedicated operand scheduler rearranges operations on input operands, hence to reduce the number of transitions of the supply voltage and, in turn, minimized the overall power consumption of the multiplier. The proposed MP razor-based DVS multiplier provided a solution toward achieving full computational flexibility and low power consumption for various general purpose low-power applications. The proposed method results in 33.8% performance improvement along with 60.6% power reduction compared with existing design.

REFERENCES

[1]. A. J. Sanchez-Clemente, L. Entrena, R. Hrbacek, and L. Sekanina, “Error mitigation using approximate logic circuits: a comparison of probabilistic and evolutionary approaches,” IEEE Transactions on Reliability, vol. 65, no. 4, pp. 1871-1883, 2016.

[2]. J. Schlachter, V. Camus, K. V. Palem, and C. Enz, “Design and applications of approximate circuits by gate-level pruning,” IEEE Transactions on Very Large Scale Integration Systems, vol. 25, no. 5, pp.1694-1702, 2017.

[3]. B. Moons and M. Verhelst, “Energy-efficiency and accuracy of stochastic computing circuits in emerging technologies,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 4, no. 4, pp. 475–486, 2014.

[4]. J. Han and M. Orshansky, “Approximate computing: an emerging paradigm for energy-efficient design,” IEEE European Test Symposium, pp. 1-6, 2013.

[5]. C. Liu, “Design and analysis of approximate adders and multipliers,” Master's Thesis, University of Alberta, Canada, 2014.

[6]. A. Wang and A. Chandrakasan, “Energy-aware architectures for a realvalued FFT implementation,” in Proc. IEEE Int. Symp. Low Power Electron. Design, Aug. 2003, pp. 360–365.

[7]. T. Kuroda, “Low power CMOS digital design for multimedia processors,” in Proc. Int. Conf. VLSI CAD, Oct. 1999, pp. 359–367.

[8]. H. Lee, “A power-aware scalable pipelined booth multiplier,” in Proc. IEEE Int. SOC Conf., Sep. 2004, pp. 123–126.

[9]. S.-R. Kuang and J.-P. Wang, “Design of power-efficient configurable booth multiplier,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 568–580, Mar. 2010.

[10]. O. A. Pfander, R. Hacker, and H.-J. Pfleiderer, “A multiplexer-based concept for reconfigurable multiplier arrays,” in Proc. Int. Conf. Field Program. Logic Appl., vol. 3203. Sep. 2004, pp. 938–942.

[11]. N. Maheshwari, Z. Yang, J. Han, and F. Lombardi, “A design approach for compressor based approximate multipliers,” IEEE International Conference on VLSI Design, pp. 209-214, 2015.

[12]. S. Venkatachalam, S. B. Ko, “Design of power and area efficient approximate multipliers,” IEEE Transactions on VLSI, vol. 25, no. 5, pp. 1–5, 2017.

[13]. W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, “Design of approximate Radix-4 Booth multipliers for error-tolerant computing,” IEEE Transactions on Computers, vol. 66, no. 8, pp. 1435-1441, 2017.

[14]. L. Qian, C. Wang, W. Liu, F. Lombardi, and J. Han, “Design and evaluation of an approximate Wallace-Booth multiplier,” IEEE International Symposium on Circuits and Systems, pp. 1974-1977, 2016.

[15]. 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, 2013.

[16]. S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, “Energy-efficient approximate multiplication for digital signal processing and classification applications,” IEEE Transactions on Very Large Scale Integrgration Systems, vol. 23, no. 6, pp. 1180–1184, 2014.

DOI 10.17148/IJIREEICE.2019.7409