Design and Analysis of Approximate Booth Multipliers with Low Power and Area Consumption on FPGA

Mr.N.B.Jilani [(1)] Mrs.M.Haritha [(2)] Venna Siva Sankara Reddy [(3)] B Ganga Raju [(4)] Gadam Venkata Gurunadham [(5)] Mukkamalla Kailash Reddy [(6)]

[1,2] *Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.*

[3,4,5,6,] *Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.*

Abstract In this project, A developing technology called approximate computing involves designing power-efficient circuits with less complexity, albeit at the expense of some accuracy loss. These circuits are appropriate for uses when strict adherence to high precision is not necessary. A common multiplication technique that cuts the size of the partial product array in half is the Radix-4 modified Booth encoding. This project proposes three approximate booth multiplier models (ABM-M1, ABM-M2, and ABM-M3) that use the radix-4 modified Booth method and approximation computing. Every one of the three designs has a different approximation method that entails both changing the partial product accumulation procedure and lowering the logic complexity of the Booth partial product generator. It is shown that the suggested approximate multipliers perform more accurately and powerfully than the current approximate Booth multipliers. In contrast with the precise Booth multiplier, when compared to the most advanced multipliers now in use, the suggested designs perform better in terms of area and power savings while keeping a high level of accuracy. Applications such as image transformation, matrix multiplication, and Finite Impulse Response (FIR) filtering are used to illustrate the effectiveness of the suggested architectures.

Keywords: Booth Multipliers, Communications, FPGA, Verilog HDL, Low Power and Xilinx Tool.

I.INTRODUCTION

Approximate computing is a concept applied to errortolerant applications in which the accuracy of an operation is reduced to improve other measures of circuit performance. Approximate computing leverages the innate ability of some applications to tolerate error. Relaxed accuracy requirements are typically acceptable in applications such as digital signal processing, image processing, data mining, and pattern recognition. In these applications, multipliers make a notable impact on power consumption, and they stand to benefit from new inexact multiplier designs with high performance. Use of approximate circuits in such applications allow for substantial improvements in performance measures such as power, area, and/or delay [1], [2].

Arithmetic units such as adders and multipliers are extensively used in digital signal processing applications. Approximation schemes for addition are widely discussed in the literature [3] -[5]. Approximation in carry-select adders based on speculation with error detection and recovery is proposed in [3]. An error-tolerant adder based on segmentation is analysed in [4]. In [5], several imprecise adders are designed by reducing the number of transistors and are utilized in digital signal processing applications.

Multiplication is most implemented using either AND-array multipliers or Booth multipliers. For a nn multiplication, AND-array multipliers involve the use of AND-gates for partial product generation to produce a partial product matrix with n rows. Booth encoding is introduced in [6] and in [7], Booth multipliers involve recoding the input combination for use in partial product generators to produce signed and plural values of the multiplicand, thereby reducing the number of rows in partial product accumulation matrix. Truncation schemes are a widely-used traditional method of decreasing circuit complexity in fixedwidth multipliers in exchange for some loss in accuracy as in [8]–[11], where the term fixed-width indicates a multiplier that produces a n-bit output given two n-bit inputs. A posttruncated fixed-width Booth multiplier designed using a compensation vector is discussed in [8]. In [9], quantization error is compensated with approximate carry values. An error compensation circuit composed of simplified sorting networks is proposed in [10]. An adaptive estimator based on conditional probability theory is studied in [11]. For fixed-width multipliers to obtain high accuracy, such compensation strategies require additional hardware resources. Approximation provides an alternative method of achieving varying degrees of accuracy in multipliers without compensation circuits.

Approximation in multipliers has been widely discussed in recent years [12]–[20]. Many of these works focus on applying approximation to the partial product accumulation stage of the multiplier [13]–[17]. Approximate counters and compressors are investigated in [13], [14], where partial product accumulation is performed using approximate counters and compressors rather than exact models. In [13], an inaccurate counter is proposed and used in a Wallace tree structure of a 4 X 4 multiplier. In [14], two approximate 4-2 compressors are proposed and used in a Dadda tree partial product accumulation. In [15], partial products are altered and approximate arithmetic units are proposed according to the probability of the modified partial products being equal to one. In the partial product perforation multiplier from [16], approximation is achieved by reducing the number of rows in the partial product accumulation circuit in ANDarray multipliers and Booth multipliers. In [17], a broken Booth multiplier with vertical breaking levels is introduced, where the elements of partial product matrix to the right of the breaking level are made zero. The authors of these works mostly analyze the effect of applying approximation to multipliers in the partial product accumulation stage. However, Booth multipliers make use of a more complex partial product generation circuit to reduce the total number of partial products generated. While substantial work has been performed on approximating partial product accumulation, additional exploration is needed into techniques that apply approximation to partial product generation in Booth multiplication.

There are few existing works investigating approximation in partial product generation [18]–[20]. In Booth multipliers, a higher radix corresponds to a decrease in the number of rows of the partial product matrix. For instance, in radix-4 Booth multipliers, partial product generation produces values of 0, 1, and 2 multiplicand and reduces the size of the partial product matrix by

nearly half. Similarly, radix-8 multipliers further reduce the number of rows in partial product matrix where the encoding signals are 0, 01, 2, 3, and 4 multiplicands. In [18], the complexity of radix-4 partial product generation is reduced via the modification of truth tables to produce two approximate Booth partial product generators each exhibiting 4@32 and 8@32 altered truth table entries respectively. In [19], approximation is applied in the generation of partial products for radix-8 Booth multipliers. An approximate 2-bit adder composed of a 3-input XOR-gate is used to generate the 3-multiplicand term. [20] makes use of a hybrid encoding technique in which exact radix-4 encoding is used to generate the most-significant partial products and approximate higher-radix encoding is used to produce the less-significant bits.

In this project, three approximate Booth multipliers models (ABM-M1, ABM-M2, and ABM-M3) based on radix4 Booth encoding are proposed. The ABM-M1 multiplier makes use of an approximate Booth partial product generator that replaces 2 multiplicand terms with 1 multiplicand terms, producing error in 4 out of 32 cases. The same approximate partial product generator is used in ABM-M2, but the multiplicand input to the generator is consolidated by replacing a set of partial products in every row with a single reduced partial product. ABM-M3 makes use of a second proposed partial product generator that produces a partial product according to the zero-values of a single encoded signal and multiplicand.

2.LITERATURE SURVEY

This project is an extension of our conference work [21]. The main improvements and novel contributions of this paper include:

1) Error distance (the absolute difference between actual value and approximate value) of the partial product generator in ABM-M1 multipliers is discussed and analyzed using 16-bit multipliers models.

2) ABM-M2 multipliers are introduced, where partial product generation and accumulation is further simplified based on a consolidated value of the multiplicand and replacing a set of partial product generators with a single partial product generator.

3) A partial product generator based on zero-values of the multiplicand and encoded signal is proposed. The proposed partial product generator is utilized in ABM-M3 multipliers.

4) An approximation factor m is used to implement and analyze the proposed designs with varying degrees of applied approximation. In each design, approximation factor m refers to the number of columns in the partial product matrix to which approximation is applied, in order of increasing significance. As m increases, a higher number of columns make use of the approximate partial product generator, and the inexactness of the multiplier increases. Approximation factors are chosen such that the error metrics of the designs for all models are similar and therefore comparable.

Models 1 and 3 make use of a rectangular replacement scheme in which all partial products with significance less than m are replaced with approximate partial products. Specifically, models 1 and 3 implement approximation factors m = N/4, N/2, 3N/4, and N. Model 2 makes use of a diagonal replacement scheme in which approximation factor specifically indicates that, for each

row, m exact partial products are compressed into a single approximate partial product. Model 2 implements approximation factors m = N/8, N/4, 3N/8, and N/2. Smaller approximation factors are used in model 2 because a diagonal replacement scheme is used, meaning that a larger total number of exact partial product generators are replaced with approximate partial product generators than in the rectangular replacement scheme used in models 1 and 3. In all proposed multipliers, the partial product accumulation is performed using a Dadda tree structure composed of exact 4-2 compressors, full-adders, and half-adders. The exact, proposed, and existing approximate multipliers are evaluated with applications including image transformation, matrix multiplication, and Finite Impulse Response (FIR) filtering.

II.EXISTING SYSTEM AND PROPOSED SYSTEM

1.EXISTING SYSTEM: RADIX-4 BOOTH MULTIPLIERS

The output of Booth multiplication can be given as the multiplication of two signed inputs A and B of length N resulting in output Pout of length 2N. The inputs and outputs of the multiplication in two's complement representation can be given as

$$A = -a_{N-1}2^{N-1} + \sum_{n=0}^{N-2} a_n 2^n ,$$
$$B = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n ,$$
$$P_{out} = -P_{2N-1}2^{2N-1} + \sum_{n=0}^{2N-2} P_n 2^n . \quad (1)$$

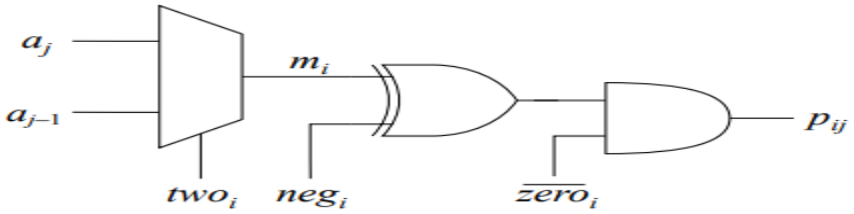

Fig. 1. Circuit schematic for the partial product generator in radix-4 encoding.

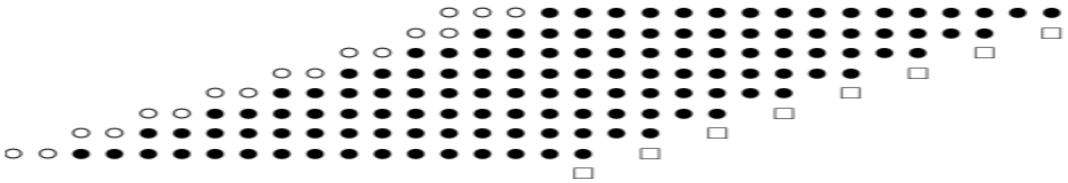

Fig. 2. Partial product matrix of a 16-bit radix-4 Booth multiplier (●: a partial product, ○: a sign-extension term, □: a correction term).

2.PROPOSED SYSTEM: APPROXIMATION IN BOOTH MULTIPLIERS

Booth multipliers are suitable candidates for applying approximation in both partial product accumulation and partial product generation. An exact radix-4 partial product generator requires all three signals negi , twoi , and zeroi , to generate the partial product. For ABM-M1 and ABMM2, an approximate partial product generator is designed using only two of the three signals, namely negi and twoi . In ABM-M3, a partial product generator is proposed which uses

only the signal zeroi . In ABM-M1, approximation is applied in partial product accumulation by combining the correction term to its respective row in the partial product matrix and thereby reducing the depth of the matrix. In ABM-M2 and ABM-M3, approximation in generation is achieved by replacing a set of partial product generators with a single approximate partial product generator, thereby reducing the number of elements in accumulation.

2.1 ABM-M1,M2,M3 Approximate Multipliers

The K-map corresponding to the partial product generator circuit in Figure 1 is approximated by modifying 4 of 32 entries as shown in Figure 3, where 1 represents a change from '0' to '1' and 0 represents a change from '1' to '0'. This results in an approximate partial product generator based on two signals, negi and zeroi , subsequently referred to as PPG-2S. The circuit schematic for this approximate partial product generator is shown in Figure 4 and can be given as

$$p_{ij} = \overline{a_j} \cdot neg_i + a_j \cdot \overline{neg_i} \cdot \overline{zero_i} \,. \qquad (3)$$

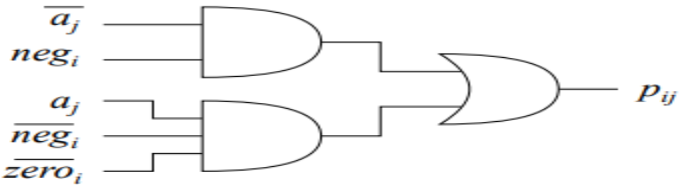

FIG2.Circuit schematic for approximate two-signal partial product generator PPG-2S.

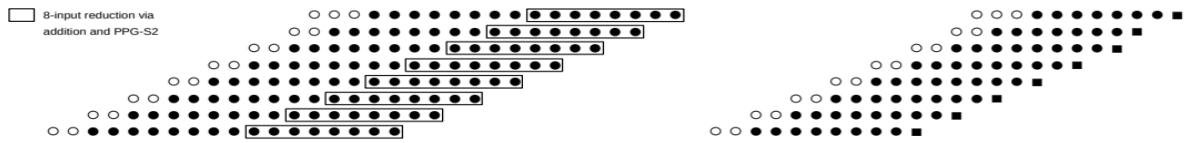

FIG3.Partial product matrix of a 16-bit ABM-M2 multiplier with m = 8. The width of the matrix is reduced by adding the m least significant bits of each partial product row, comparing the result to m, and then using the resulting 1-bit or 0-bit as an input to PPG-2S (c: a partial product, b: a sign-extension term, v: approximate partial product generated using PPG-2S).

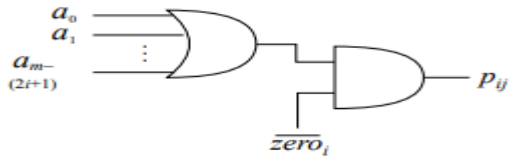

Fig. 4. Circuit schematic for approximate single-signal partial product generator PPG-1S.

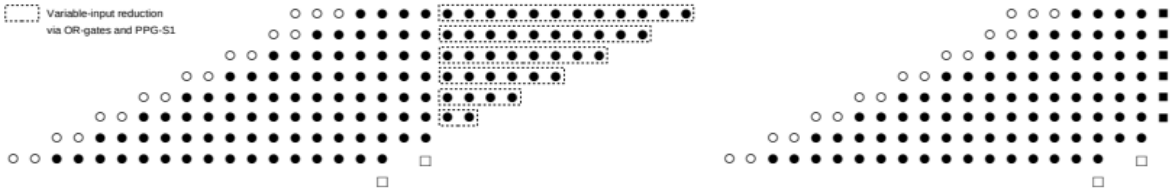

Fig. 5. Partial product matrix of a 16-bit ABM-M3 multiplier with m = 12. The width of the matrix is reduced by OR-ing together all bits with a significance less than m and then using the

result of the OR operation as an input into PPG-1S (c: a partial product, b: a sign-extension term, u: a correction bit, v: approximate partial product generated using PPG-1S).

III. Results and Analysis discussion

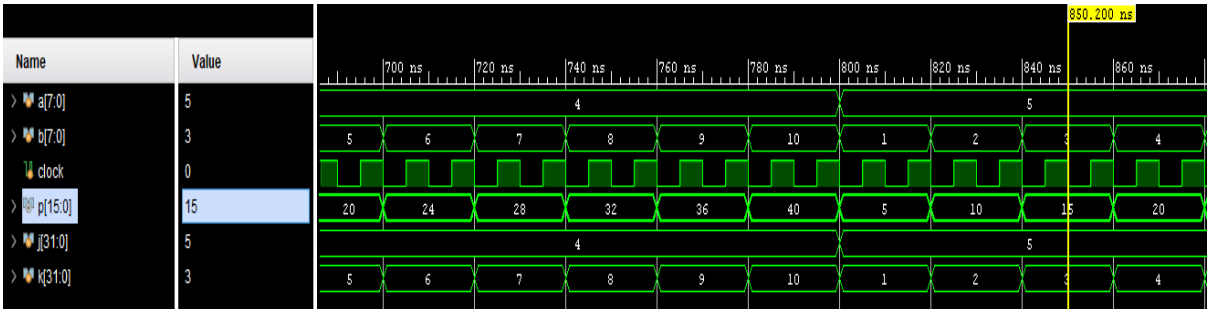

FIG1.SIMULATION OUTPUT

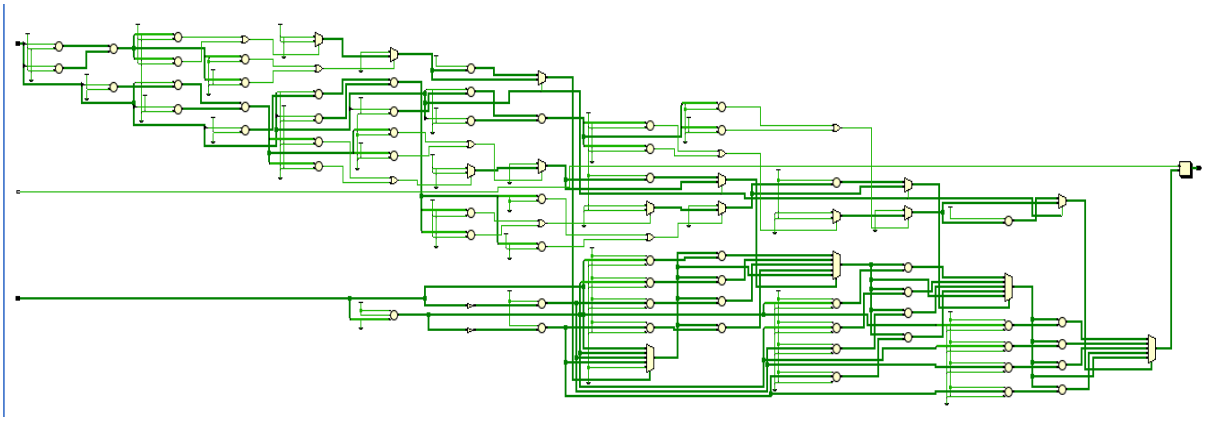

FIG2.RTL SCHEMATIC DIAGRAM

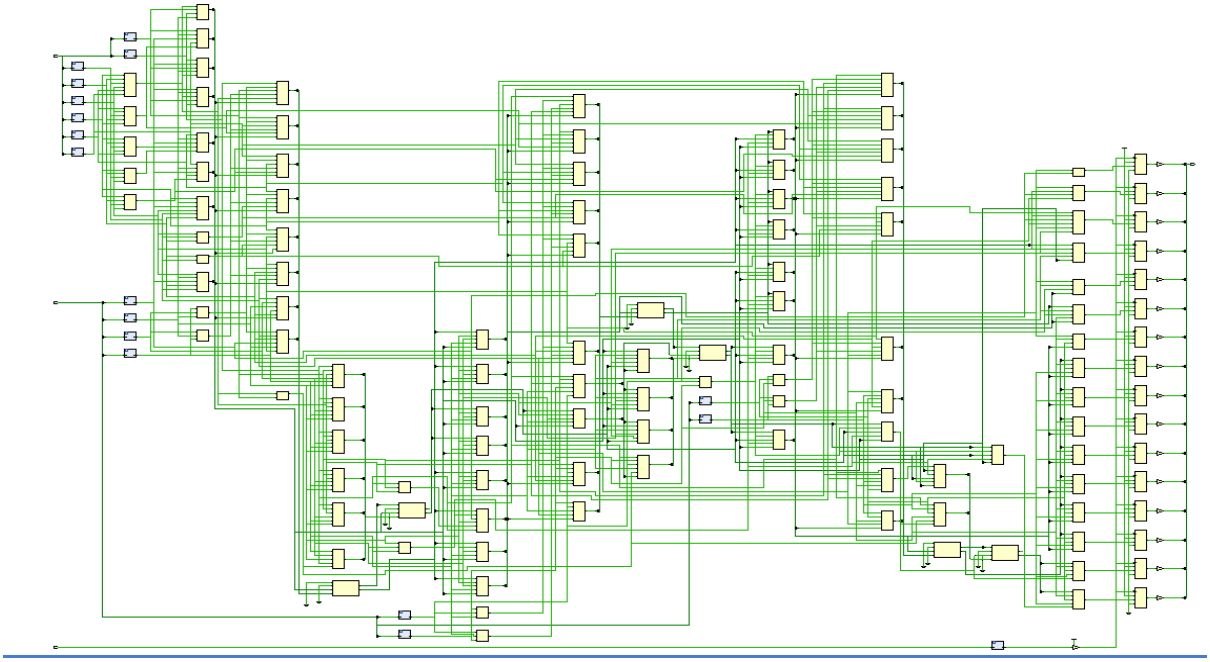

FIG3.SYNTHESIS DESIGN

Resource	Utilization	Available	Utilization %
LUT	87	117120	0.07
FF	16	234240	0.01
IO	33	204	16.18
BUFG	1	352	0.28

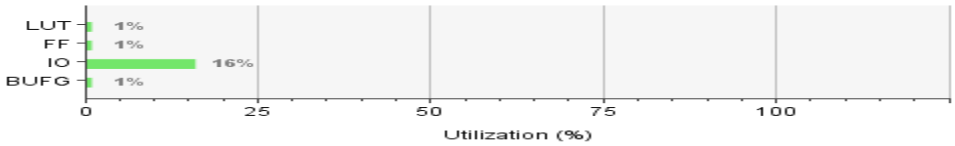

FIG4.UTILIZATION OF LUTS AND FFS IO BUFFERS

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**24.152 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**58.1°C**
Thermal Margin:	41.9°C (30.0 W)
Effective ϑJA:	1.4°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

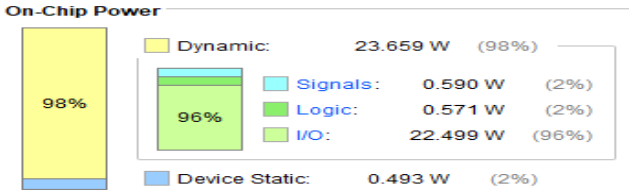

FIG5.POWER REPORT

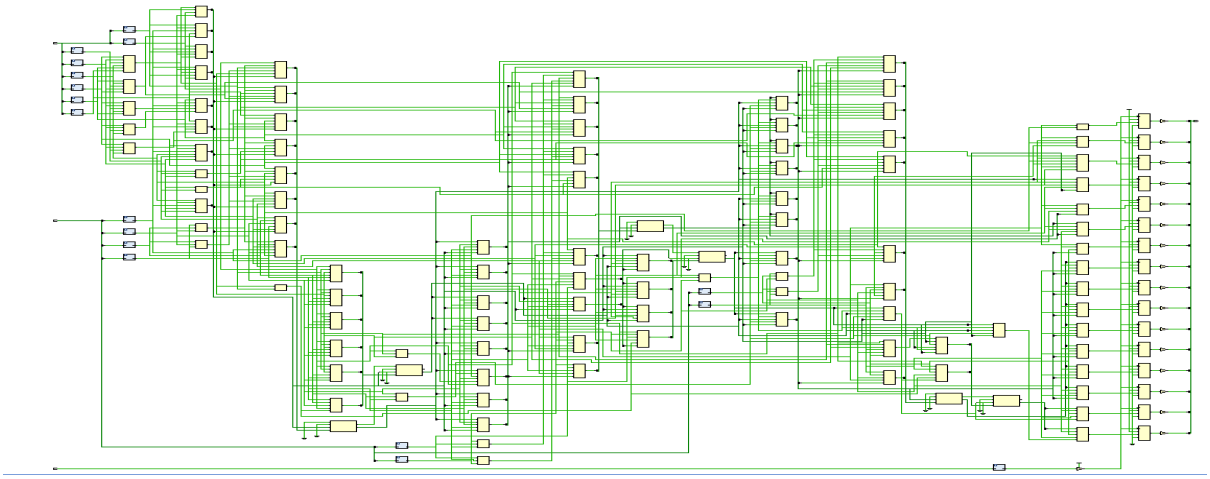

FIG6.OPTIMIZED DESIGN

Conclusion

Three models of approximate Booth multipliers are proposed, in which approximation is focussed on partial product generation. A partial product generator that makes use of only two signals is used in ABM-M1 and ABM-M2, and the partial product generator in ABM-M3 is further reduced to use only one recoded signal. An approximation factor m is used to indicate the imprecision of each model. Area-power product and error metrics of the proposed multipliers are compared with exact multipliers and existing state-of-the-art approximate Booth multipliers.

REFERNECES

[1] J. Han and M. Orshansky, “Approximate Computing: An Emerging Paradigm for Energy-Efficient Design,” IEEE ETS, 2013.

[2] S. Venkataramani, S. T. Chakradhar, K. Roy and A. Raghunathan, “Computing approximately, and efficiently,” Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015, pp. 748-751.

[3] K. Du, P. Varman and K. Mohanram,, “High performance reliable variable latency carry select addition,” Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012, pp. 1257-1262.

[4] Ning Zhu, W. L. Goh and K. S. Yeo,, “An enhanced low-power highspeed Adder For Error-Tolerant application,” Proceedings of the 2009 12th International Symposium on IC, Singapore, 2009, pp. 69-72.

[5] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, “Lowpower digital signal processing using approximate adders,” IEEE Transactions on Computer Aided Design Integrated Circuits Systems, vol. 32, no. 1, pp. 124-137, 2013.

[6] A. D. Booth, “Signed binary multiplication technique,” Quarterly Journal of Mechanics and Applied Mathematics, vol. 4, pp. 236–240, Jan. 1951.

[7] O. L. Macsorley, ”High-Speed Arithmetic in Binary Computers,” in Proceedings of the IRE, vol. 49, no. 1, pp. 67-91, Jan. 1961.

[8] S. J. Jou, M. Tsai, and Y. Tsao, “Low-error reduced-width Booth multipliers for DSP applications,” IEEE Transactions on Circuits and Systems, vol. 50, no. 11, pp. 1470-1474, 2013.

[9] K. J. Cho, K. C. Lee, J. G. Chung and K. K. Parhi, ”Design of lowerror fixed-width modified Booth multiplier,” IEEE Transactions on Very Large Scale Integration Systems, vol. 12, no. 5, pp. 522-531, May 2004.

[10] J. Wang, S. Kuang and S. Liang, ”High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications,” IEEE Transactions on Very Large Scale Integration Systems, vol. 19, no. 1, pp. 52-60, Jan. 2011.

[11] Y. Chen and T. Chang, ”A High-Accuracy Adaptive ConditionalProbability Estimator for Fixed-Width Booth Multipliers,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 59, no. 3, pp. 594-603, March 2012.

[12] P. Kulkarni, P. Gupta, and M. D. Ercegovac, “Trading accuracy for power in a multiplier architecture,” Journal of Low Power Electronics, vol. 7, no. 4, pp. 490-501, 2011.

[13] C. H. Lin and C. Lin, “High accuracy approximate multiplier with error correction,” IEEE 31st International Conference on Computer Design, 2013, pp. 33-38.

[14] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, “Design and analysis of approximate compressors for multiplication,” IEEE Transactions on Computers,, vol. 64, no. 4, pp. 984-994, 2015.

[15] S. Venkatachalam and S. B. Ko, ”Design of Power and Area Efficient Approximate Multipliers,” IEEE Transactions on Very Large Scale Integration Systems, vol. 25, no. 5, pp. 1782-1786, May 2017.

[16] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris and K. Pekmestzi, “Design-efficient approximate multiplication circuits through partial product perforation,” IEEE Transactions on Very Large Scale Integration systems, vol. 24, no. 10, pp. 3105-3117, Oct. 2016.

[17] F. Farshchi, M. S. Abrishami and S. M. Fakhraie, ”New approximate multiplier for low power digital signal processing,” The 17th CSI International Symposium on Computer Architecture & Digital Systems (CADS 2013), Tehran, 2013, pp. 25-30.

[18] 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, Aug. 2017.

[19] H. Jiang, J. Han, F. Qiao and F. Lombardi, “Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation,” IEEE Transactions on Computers, vol. 65, no. 8, pp. 2638-2644, Aug. 2016.

[20] V. Leon, G. Zervakis, D. Soudris and K. Pekmestzi, ”Approximate Hybrid High Radix Encoding for Energy-Efficient Inexact Multipliers,” in IEEE Transactions on Very Large Scale Integration Systems, vol. 26, no. 3, pp. 421-430, March 2018.

[21] S. Venkatachalam, H. J. Lee and S. B. Ko, “Power Efficient Approximate Booth Multiplier,” 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 2018, pp. 1-4.

[22] N. Burgess, ”Removal of sign-extension circuitry from Booth’s algorithm multiplier-accumulators,” in Electronics Letters, vol. 26, no. 17, pp. 1413-1415, Aug. 1990.

[23] 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, Sept. 2013.