IOSR Journal of Engineering (IOSRJEN) www.iosrjen.org
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 08, Issue 8 (August. 2018), ||V (V) || PP 71-78

An Approximate Multiplier-Accumulator Based on Radix-4 Modified Booth Algorithm

Maede Jabbar Zare
Department of Computer Engineering, Sheikhbahaee University, Baharestan, Isfahan, Iran

Hossein Mohammadi Nejad
Department of Computer Engineering, Sheikhbahaee University, Baharestan, Isfahan, Iran
E-mail: mohammadi.n@shbu.ac.ir
Seyed Mohammad Mahdi Tabei
Department of Computer Engineering, Sheikhbahaee University, Baharestan, Isfahan, Iran
E-mail: tabei@shbu.ac.ir

Corresponding Author:HosseinMohammadinejad

Abstract- Fast multiplier-accumulator (MAC) is one of the most important requirements of today's VLSI systems and digital signal processing (DSP) applications. DSP applications are usually comprised of many multiplications and accumulations. So, for real time signal processing applications, high speed MAC is always an essential element. The speed of MAC is highly affected by the multiplier. Truncated multipliers area kind of approximate multipliers which discard a part of the partial products to reduce hardware. Although this elimination lowers accuracy, meaningful and faster results are still provided. In this paper we have proposed an approximate multiplier-accumulator. The results of synthesis show that the proposed design is faster than the traditional one. The proposed design can be used effectively for digital signal processing and embedded systems.

Keywords: Multiplier-Accumulator unit, Modified BoothMultiplier, Approximate Multiplier.

Date of Submission: 16-08-2018 Date of acceptance: 03-09-2018

I. INTRODUCTION

Rapid advances in multimedia and communication systems have caused a progressive demand for real-time signal processing like audio signal processing, video/image processing, or large-capacity data processing[1]. Most DSP methods use nonlinear functions such as discrete cosine transform (DCT)[2]or discrete wavelet transform (DWT) [3]. These functions are basically accomplished by repetitive application of multiplication and addition, so the speed of the multiplication and addition arithmetic determines the execution speed and the performance of the entire calculation.The importance of multiplication and addition motivateddigital designers to invent a new arithmetic unit which is called Multiplier-Accumulator (MAC). Multiplier and adder are the basic units of MAC. The performance of MAC is affected by both multiplier and adder. A number of designers have proposed new methods for improving MAC by improving adder[4]. Nonetheless, the multiplier has more significanceas it requires the longest delay among the basic operational blocks and the critical path is determined by the multiplier[5].

Generally, there are two kinds of multipliers, sequential multipliers and parallel multipliers. Parallel multipliers are used for their high speed performance, although high power consumption and large silicon area make them not to be perfect[6, 7].Sequential multipliers are popular for their low power consumption and small silicon area; their disadvantage is their low speed[8, 9].

Typically, multipliers are implemented using the modified Booth algorithm (MBA), in which partial product is generated from multiplicand (X) and multiplier (Y). Booth multiplication allows for the smaller, faster multiplication circuits through encoding bits and providing considerable improvement by reducing the partial products. Modified booth algorithm exists for radix 2 and higher (4,8,16 and 32). Partial products are further reduced by using higher radix booth encoder which increases complexity but improves the performance[10].For high-speed multiplication, the modified radix-4 Booth's algorithm [11] is commonly used. Nonetheless, the long critical path causes the problem not to be solved completely[1, 10, 12].Lately, there have been efforts to speed up MBA by using parallel implementations[13-17]. Researchers have been successful in developing high-speed MAC based on the Baugh–Wooley algorithm (BWA), and applying these structures to several digital filtering applications [18-26].

In Many scientific and engineering problems we need accurate, precise and deterministic algorithms to compute the result. However, applications involving signal/image processing and multimedia do not need exact and accurate computations. It is because these applications are error tolerant and produce results that are good enough for human perception [27]. In error resilient applications, conversion or elimination of some parts leads to a reduction in circuit complexity and improvement of total efficiency. Although usingapproximate computing in error tolerant applications reduces accuracy, meaningful results are still provided[28]. Researchers have proposed a number of designs for approximateadders[29-32].

Approximate multipliers have beenalsoinvestigatedsince they haveconsiderableimportance in arithmetic operations [33-39].Truncated multiplication is one of the well-known approximate techniques which is able to reduce the hardware requirements and increase the speed. In this technique, we ignore the least significant columns of the partial products. The carries generated by least significant columns are estimated, and they will be added with the most significant columns [40].

In this paper, a new approximate MAC is proposed. We have reduced the number of the partial products by using Radix-4 modified booth algorithm. We have also employed approximate computing to make our proposed design faster.

This paper is divided into several parts. In part II we present an overview of MAC. PartIIIdemonstratestruncated multiplication. In part IV the proposed MAC architecture is introduced. Part V demonstrates the results of simulation.Finally, the conclusion is presented in partVI.

II. OVERVIEW OF MAC UNIT

Multiplication and accumulation are the basic operations in DSP.The MAC unitis the key element of the DSPapplicationssuch as filtering, convolution, and inner products and it is able to performoperations such as high speed multiplication, saturationandmultiplication with cumulative addition and subtraction[41]. A multiplier basically consists of three operational steps. The firstoneis Booth encoding in which a partial product is produced from the multiplicand and the multiplier. In the second step, partial products are compressed or they are added by an array of adders. Finally, we have final addition, in which sum and carry are added to produce multiplication result. If there is any need to accumulate the final result, the number of steps will be increased to four. Fig.1 shows Basic arithmetic steps of multiplication and accumulation. General hardware architecture of MAC is shown in Fig. 2[5].

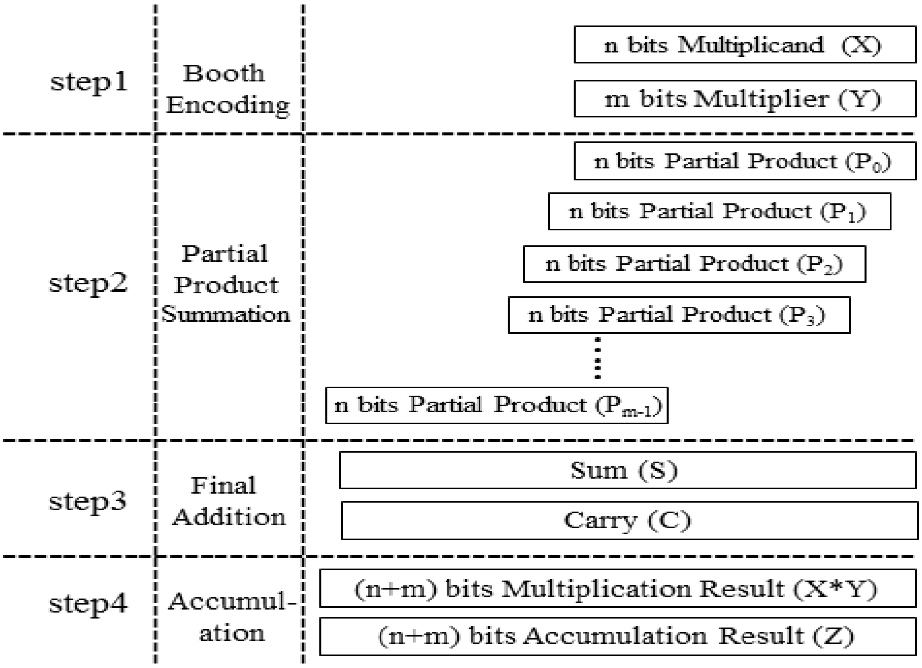

Fig. 1.Basic arithmetic steps of multiplication and accumulation[5].

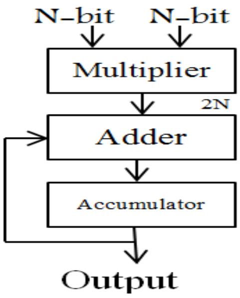

Fig. 2.Hardware architecture of MAC

1. Radix-4Modified Booth Multiplier

This multiplier is one of the most important multipliers which can be used in MAC. The reason of this importance is the technique of encoding which reduces the number of partial products and increases speed.Fig. 3shows the multiple generation part of radix-4 multiplier based on Booth's recoding[8].Initially, a zero is concatenated to multiplier.Then, multiplier bits will form 3-bit overlapping groups, and each group will be used to generate *Neg*, *Two* and *Non0* bits by utilizing Recoding Logic. *Two*is the select of multiplexer and Non0 is the enable of multiplexer. *Neg* is used to select addition or subtraction operation. If *Neg* is one the operation will be subtraction and if it is zero the operation will be addition. The output of add/subtract unit is entered to the n+1most significant bits of register and 2-bit arithmetic shift occurs.

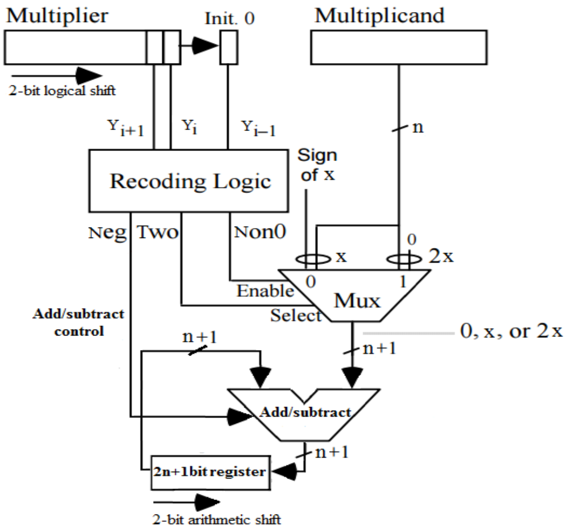

Fig. 3. The multiple generation part of a radix-4 multiplier based on Booth's recoding[8].

Fig. 4shows the general form of radix-4 multiplication with modified Booth's recoding of the 2's-complement multiplier. Each n-cell table represents an n-bit number. The 8-bit 2's-complement multiplier is recoded as a 4-digit number $Z = (\alpha\beta\gamma\omega)_{four}$.This recoded number dedicates the multiples $\omega\times X$, $\gamma\times X$, $\beta\times X$, $\alpha\times X$ to be added to the cumulative partial product in four cycles. The value of Z is determined according to table I. In all intermediate steps, the upper half of the cumulative partial product is extended 2 bits to accommodate the sign extension needed for proper handling of the negative values. Black cells represent bits which are producedby sign extension. Dark gray cells represent bits which can be produced by sign-extension or not. If the recoded number is 2 or -2 there is no need for sign-extension. Light gray cells represent zero bits. $P_i$denotes register Pin cycle i.Note the sign extension during the right shift to obtain $P_i$from$4P_i$[8, 42-44].

Table I

Booth recoding table for radix-4

i+1	i	i-1	add
0	**0**	**0**	**0*X**

0	**0**	**1**	**1*X**
0	**1**	**0**	**1*X**
0	**1**	**1**	**2*X**
1	**0**	**1**	**-1*X**
1	**0**	**0**	**-2*X**
1	**1**	**1**	**0*X**
1	**1**	**0**	**-1*X**

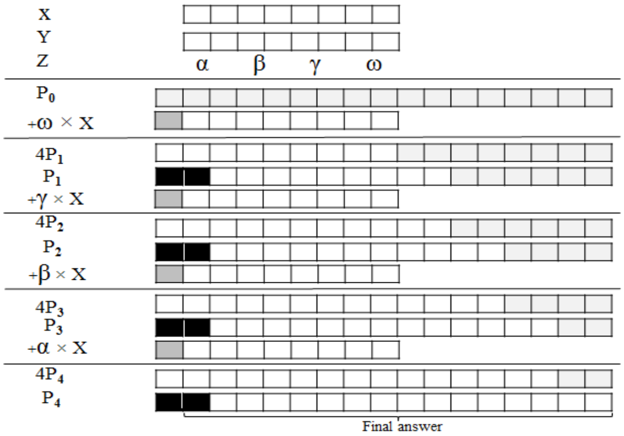

Fig. 4. General form of radix-4 multiplication with modified Booth's recoding of the 2's-complement multiplier.

III. TRUNCATED MULTIPLICATION

One of the most effective approaches for implementing approximate parallel multipliers is truncation, in which the Least Significant Part (LSP) of partial products is ignored.Actually, in many applications, it is possible to reduce the area by discarding LSP and just using the Most Significant part (MSP). Although this will cause some error in final result, but it can be compensated by using a simple circuit that produces error compensation value (ECV). As it is shown in Fig. 5, LSB can be divided in two parts: LSB_{minor} and LSB_{major}. Error compensation value can be obtained using most significant column of LSB_{minor} that is called Input Correction (IC) [45]. Error compensation techniques are devided in two types:

1- Constant error compensation: adds a constant value to final answer which is not relatedto multipplier and multiplicand[46, 47].

2- Variable error compensaton: adds a variable value to final answer which depends on multipier and multiplicand [48, 49].

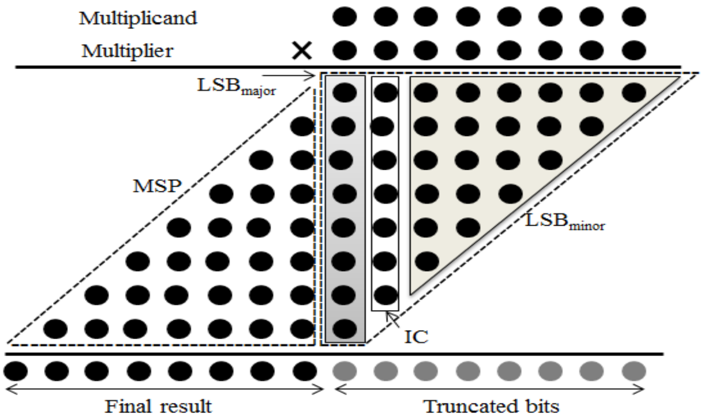

Fig. 5. Partial products matrix of 8×8 truncated multiplier.

IV. PROPOSED MAC ARCHITECTURE

Our proposed method is an approximate MAC which applies the advantages of multiplier-accumulator and approximate computing.Since the multiplier is the main part of the MAC, we have concentrated on it.

Forhigh-speed multiplication, the modified radix-4 Booth’s algorithm (MBA) is commonly used. We have applied approximation technique for MBA in order to increase the speed of multiplier and improve the total efficiency. Fig.6showsgeneral form of the proposed radix-4 multiplication with modified Booth's recoding of the 2's-complement multiplier. For improving the speed of sequential multipliers, the length or the number of cycles should be reduced. In our proposed method we have eliminated the first cycle of multiplication since it has the least efficacyin final answer. We have called our proposed multiplier "Elimination of First Cycle" or EFC.

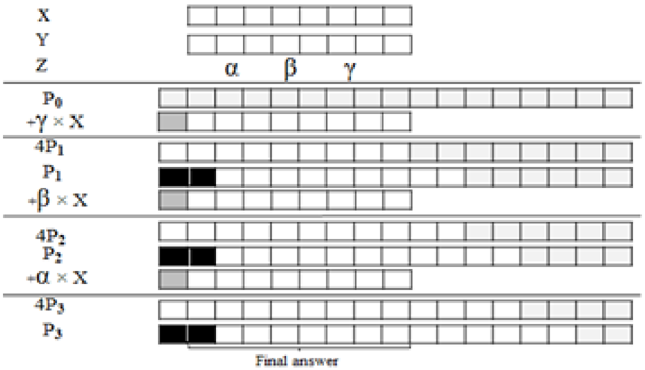

Fig. 6. General form of radix-4 multiplication with modified Booth's recoding of the 2's-complement multiplier

Fig. 7Shows the multiple generation part of the proposed radix-4 8×8 multiplier based on Booth's recoding.The main difference between this circuit and the one which is introduced in [8] is the elimination of first cycle. In EFC, there is no need to add a zero to the right hand side of multiplier. Also,theprocess of 3 bit grouping is started from the second least significant bit of multiplier. In this case Zwill be $(\alpha\beta\gamma)_{\text{four}}$ instead of $(\alpha\beta\gamma\omega)_{\text{four}}$.

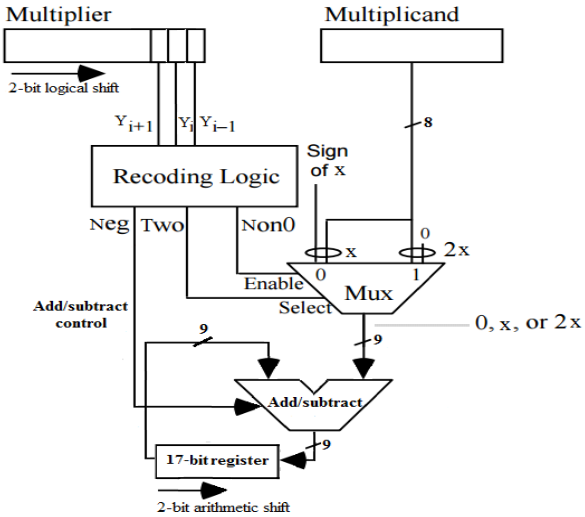

Fig. 7.The multiple generation part of the proposed radix-4 8×8 multiplier based on Booth's recoding.

V. SIMULATION AND SYNTHESIS RESULTS

In order to evaluate the precision of the proposed method and to compare it with the precise one, we have used MATLAB software. All possible values for multiplicand and multiplier which are totally 256×256 = 65536, are applied to both algorithms. We call precise modified booth algorithm "PMBA". If we compare 8 most significant bits of PMBA and EFC which is our concern, 74.8% of answers are similar and 24.98% of answers have slight difference. It means that our proposed method is reliable for applications which need n most significant bits of final answer, such as floating point.

For achieving theoreticaldelayof PMBA and EFC, We assume the delay of Recoding Logic is 3 gate-delays, the delay of a multiplexer is 2 gate-delays, the delay of n+1 bitaddition/subtraction unitis2(n+1) gate-delays(*n*indicates the number of inputs).Table II shows the delay of n-bit PMBA and n-bit EFC. The delay of an 8-bitPMBA and an 8-bit ECU is demonstrated in table III.

Table IIDelay of precise and proposed designs (n-bit)

Delay/Designs	PMBA	EFC
Delay per cycle	2n+7	2n+7
Number of cycles	$\frac{n}{2}$	$\frac{n}{2}$-1
Total delay	$\frac{n}{2}(2n+7)$	$(\frac{n}{2}-1)$(2n+7)

Table IIIDelay of precise and proposed designs (8-bit, gate-delay)

Delay/Designs	PMBA	EFC
Delay per cycle	23	23
Number of cycles	4	3
Total delay	92	69

Table IV shows the timing reportswhich are produced by synthesizing PMBA and EFC. The delay of EFC is 25% less than PMBA.It means that EFC is faster and can be used in applications which are error tolerant and the speed of multiplication is the most important criteria.

Table IV The Delay of PMBA and EFC(ns)

Delay/Design	PMBA	EFC
Recoding Logic	0.92	0.92
Multiplexer	2.26	2.26
Adder/Subtractor	6.62	6.62
Delay per Cycle	14.80	14.80
Number of cycles	4	3
Total Delay	59.2	44.4

VI. CONCLUSION

In this paper, a new approximate Multiplier-Accumulatorwasproposed. Since modified booth multiplier is one of the most important multipliers that can be used for MAC, we have introduced a new truncated modified booth algorithm. Results of simulation showthat the upper half of the EFC's answer is similar to upper half of the PMBA's answer in 74.8% of times.The proposed design is faster than its precise counterpart and results of synthesis show that its delay is 25% less than the traditional one. The proposed architecture can be used effectively in digital signal processing.

REFERENCES

[1]. Cavanagh, J.J., *Digital computer arithmetic*1983: McGraw-Hill, Inc.

[2]. Standardization, I.O.f., *Information Technology--Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to about 1, 5 Mbit/s*1996: International Organization for Standardization.

[3]. Jagadeesh, S. and S.V. Chary, *Design of Parallel Multiplier–Accumulator Based on Radix-4 Modified Booth Algorithm with SPST.* International Journalof Engineering Research and Application, 2012.

[4]. Xia, B.-j., P. Liu, and Q.-d. Yao, *New method for high performance multiply-accumulator design.* Journal of Zhejiang University-SCIENCE A, 2009. **10**(7): p. 1067-1074.

[5]. Seo, Y.-H. and D.-W. Kim, *A new VLSI architecture of parallel multiplier–accumulator based on Radix-2 modified Booth algorithm.* IEEE Transactions on very large scale integration (vlsi) systems, 2010. **18**(2): p. 201-208.

[6]. Jaberipur, G. and A. Kaivani, *Improving the speed of parallel decimal multiplication.* IEEE Transactions on Computers, 2009. **58**(11): p. 1539-1552.

[7]. Hong, S., et al. *Low power parallel multiplier design for DSP applications through coefficient optimization*. in *ASIC/SOC Conference, 1999. Proceedings. Twelfth Annual IEEE International*. 1999. IEEE.

[8]. Behrooz, P., *Computer arithmetic: Algorithms and hardware designs.* Oxford University Press, 2000. **19**: p. 512583-512585.

[9]. Hasan, M.A. and C. Negre, *Sequential multiplier with sub-linear gate complexity.* Journal of Cryptographic Engineering, 2012: p. 1-7.

[10]. 10. Omondi, A.R., *Computer Arithmetic Systems Algorithms, Architecture and Implementation.* 1994.

[11]. MacSorley, O.L., *High-speed arithmetic in binary computers.* Proceedings of the IRE, 1961. **49**(1): p. 67-91.

[12]. Waser, S. and M.J. Flynn, *Introduction to arithmetic for digital systems designers.* 1982.

[13]. Cooper, A. *Parallel architecture modified Booth multiplier*. in *IEE Proceedings G-Electronic Circuits and Systems*. 1988. IET.

[14]. Shanbhag, N.R. and P. Juneja, *Parallel implementation of a 4* 4-bit multiplier using a modified Booth's algorithm.* IEEE Journal of solid-state circuits, 1988. **23**(4): p. 1010-1013.

[15]. Goto, G., et al., *A 54* 54-b regularly structured tree multiplier.* IEEE Journal of solid-state circuits, 1992. **27**(9): p. 1229-1236.

[16]. Fadavi-Ardekani, J., *M* N Booth encoded multiplier generator using optimized Wallace trees.* IEEE Transactions on very large scale integration (vlsi) systems, 1993. **1**(2): p. 120-125.

[17]. Ohkobo, N. *A 4.4-ns CMOS 54× 54-b multiplier using pass-transistor multiplexor*. in *IEEE Proc*. 1994.

[18]. Tawfik, A., F. El-Guibaly, and P. Agathoklis, *New realization and implementation of fixed-point IIR digital filters.* Journal of Circuits, Systems, and Computers, 1997. **7**(03): p. 191-209.

[19]. Tawfik, A., et al., *High-speed area-efficient inner-product processor.* Canadian Journal of Electrical and Computer Engineering, 1994. **19**(4): p. 187-191.

[20]. Elguibaly, F., *Overflow handling in inner-product processors.* IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 2000. **47**(10): p. 1086-1090.

[21]. Fahmi, M.N., et al. *Design of novel serial-parallel inner-product processors*. in *Circuits and Systems, 1994. ISCAS'94., 1994 IEEE International Symposium on*. 1994. IEEE.

[22]. Abdel-Raheem, E., F. EL-GUIBALY, and A. Antoniou, *Efficient decimator and interpolator array structures.* Journal of Circuits, Systems, and Computers, 1995. **5**(02): p. 215-238.

[23]. Abdel-Raheem, E., F. El-Guibaly, and A. Antoniou, *VLSI array processors for linear-phase FIR filters.* Canadian Journal of Electrical and Computer Engineering, 1995. **20**(2): p. 73-77.

[24]. Fahmi, M., et al. *Area-time efficient fixed-point multiplier-accumulators for inner-product computation*. in *Proc. IEEE Int. Conf. Microelectronics, Dhahran, Saudi Arabia*. 1999.

[25]. Tawfik, A., F. El-Guibaly, and P. Agathoklis. *Efficient systolic implementation of fixed-point state-space digital filter*. in *Electrical and Computer Engineering, 1993. Canadian Conference on*. 1993. IEEE.

[26]. Abdel-Raheem, E., F. El-Guibaly, and A. Tawfik. *Systolic implementation of linear-phase FIR filters*. in *Electrical and Computer Engineering, 1993. Canadian Conference on*. 1993. IEEE.

[27]. Han, J. and M. Orshansky. *Approximate computing: An emerging paradigm for energy-efficient design*. in *Test Symposium (ETS), 2013 18th IEEE European*. 2013. IEEE.

[28]. Venkatesan, R., et al. *MACACO: Modeling and analysis of circuits for approximate computing*. in *Proceedings of the International Conference on Computer-Aided Design*. 2011. IEEE Press.

[29]. Phillips, B.J., D.R. Kelly, and B.W. Ng. *Estimating adders for a low density parity check decoder*. in *SPIE Optics+ Photonics*. 2006. International Society for Optics and Photonics.

[30]. Lu, S.-L., *Speeding up processing with approximation circuits.* Computer, 2004. **37**(3): p. 67-73.

[31]. Ashmila, E., S. Dlay, and O. Hinton, *Adder methodology and design using probabilistic multiple carry estimates.* IEE Proceedings-Computers and Digital Techniques, 2005. **152**(6): p. 697-703.

[32]. Meera, G. and T.J. Swamy, *A Novel Power Reduction Design using Approximate Adders for Inexact Computing.*

[33]. Kyaw, K.Y., W.L. Goh, and K.S. Yeo. *Low-power high-speed multiplier for error-tolerant application*. in *Electron Devices and Solid-State Circuits (EDSSC), 2010 IEEE International Conference of*. 2010. IEEE.

[34]. Kulkarni, P., P. Gupta, and M. Ercegovac. *Trading accuracy for power with an underdesigned multiplier architecture*. in *VLSI Design (VLSI Design), 2011 24th International Conference on*. 2011. IEEE.

[35]. Mahdiani, H.R., et al., *Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications.* IEEE Transactions on Circuits and Systems I: Regular Papers, 2010. **57**(4): p. 850-862.

[36]. Lin, C.-H. and C. Lin. *High accuracy approximate multiplier with error correction*. in *Computer Design (ICCD), 2013 IEEE 31st International Conference on*. 2013. IEEE.

[37]. Bhardwaj, K., P.S. Mane, and J. Henkel. *Power-and area-efficient Approximate Wallace Tree Multiplier for error-resilient systems*. in *Quality Electronic Design (ISQED), 2014 15th International Symposium on*. 2014. IEEE.

[38]. Liu, C., J. Han, and F. Lombardi. *A low-power, high-performance approximate multiplier with configurable partial error recovery*. in *Proceedings of the conference on Design, Automation & Test in Europe*. 2014. European Design and Automation Association.

[39]. Momeni, A., et al., *Design and analysis of approximate compressors for multiplication.* IEEE Transactions on Computers, 2015. **64**(4): p. 984-994.

[40]. Stine, J.E. and O.M. Duverne. *Variations on truncated multiplication*. in *Digital System Design, 2003. Proceedings. Euromicro Symposium on*. 2003. IEEE.

[41]. Mohanapriya, R. and K. Rajesh, *A Modified Architecture of Multiplier and Accumulator Using Spurious Power Suppression Technique.* International Journal of Students' Research in Technology & Management, 2015. **3**(2): p. 258-263.

[42]. Choubey, R. and M. Arif, *Low power and area optimized using modified booth algorithm Radix-2 and Radix-4.* International Journal of Emerging Technology and Advanced Engineering, 2014.

[43]. Vanama, C. and M. Sumalatha, *Implementation of High Speed Modified Booth Multiplier And Accumulator (Mac) Unit".* Iosr Journal of Electronics And Communication Engineering (Iosr-Jece), 2013. **8**(5).

[44]. Pavate, C., *Efficient Implementation of Modified Booth Algorithm in Radix-4 Form.* International Journal of Innovative Research in Computer and Communication Engineering, 2016. **4**(6): p. 11065-11069.

[45]. Tabei, S.M.M. and H. Nikmehr, *An unsigned truncated sequential multiplier with variable error compensation.* Microprocessors and Microsystems, 2017. **49**: p. 9-17.

[46]. Lim, Y., *Single-precision multiplier with reduced circuit complexity for signal processing applications.* IEEE Transactions on Computers, 1992. **41**(10): p. 1333-1336.

[47]. Schulte, M.J. and E.E. Swartzlander. *Truncated multiplication with correction constant [for DSP]*. in *VLSI Signal Processing, VI, 1993.,[Workshop on]*. 1993. IEEE.

[48]. King, E.J. and E. Swartzlander. *Data-dependent truncation scheme for parallel multipliers*. in *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*. 1997. IEEE.

[49]. Swartzlander, E. *Truncated multiplication with approximate rounding*. in *Signals, Systems, and Computers, 1999. Conference Record of the Thirty-Third Asilomar Conference on*. 1999. IEEE.

IOSR Journal of Engineering (IOSRJEN) is UGC approved Journal with Sl. No. 3240, Journal no. 48995.

MaedeJabbarZare,HosseinMohammadinejad, Seyed Mohammad Mahdi Tabei, An Approximate Multiplier-Accumulator Based on Radix-4 Modified Booth Algorithm.”IOSR Journal of Engineering (IOSRJEN), vol. 08, no. 8, 2018, pp. 71-78.