



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 11, November 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.118

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com



Unsigned and Signed High Speed Radix-4 Booth Multiplier: A Review

Rahul Kumar¹, Prof. Sourabh Pandey²

M. Tech. Scholar, Department of ECE, LNCT, Bhopal, India¹

Assistant Professor, Department of ECE, LNCT, Bhopal, India²

ABSTRACT-This paper presents the design and implementation of signed-unsigned Modified Booth Encoding (SUMBE) multiplier. The present Modified Booth Encoding (MBE) multiplier and the Baugh-Wooley multiplier perform multiplication operation on signed numbers only. The array multiplier and Braun array multipliers perform multiplication operation on unsigned numbers only. Thus, the requirement of the modern computer system is a dedicated and very high speed unique multiplier unit for signed and unsigned numbers. Therefore, this paper presents the design and implementation of SUMBE multiplier. The modified Booth Encoder circuit generates half the partial products in parallel. By extending sign bit of the operands and generating an additional partial product the SUMBE multiplier is obtained. The Carry Save Adder (CSA) tree and the final Carry Look-ahead (CLA) adder used to speed up the multiplier operation. Since signed and unsigned multiplication operation is performed by the same multiplier unit the required hardware and the chip area reduces and this in turn reduces power dissipation and cost of asystem.

KEYWORDS:-SUMBE, MBE, CSA, CLA, Baugh-wooleymultiplier

I. INTRODUCTION

In advanced processing frameworks augmentation is a math activity. The increase activity comprises of delivering halfway items and after that including these incomplete items the last item is acquired. In this way the speed of the multiplier relies upon the quantity of incomplete item and the speed of the snake. Since the multipliers significantly affect the exhibition of the whole framework, some elite calculations and models have been proposed [1-2]. The rapid and committed multipliers are utilized in pipeline and vector PCs.

The fast Booth multipliers and pipelined Booth multipliers are utilized for advanced sign preparing (DSP) applications, for example, for sight and sound and correspondence frameworks. Rapid DSP calculation applications, for example, Fast Fourier transform (FFT) require augmentations and increases. The paper [3] presents a structure approach for fast Booth encoded parallel multiplier. For halfway item age, another Modified Booth encoding (MBE) plot is utilized to improve the presentation of customary MBE plans. In any case, this multiplier is just for marked number duplication task.

The regular changed Booth encoding (MBE) creates an unpredictable incomplete item exhibit in view of the additional fractional item bit at any rate huge piece position of every halfway item push.

Along these lines papers [4] presents a straightforward way to deal with create an ordinary incomplete item cluster with less halfway item pushes and immaterial overhead, in this manner bringing down the intricacy of fractional item decrease and lessening the territory, postponement, and intensity of MBE multipliers. Yet, the disadvantage of this multiplier is that it capacities just for marked number operands.

The changed Booth calculation is widely utilized for fast multiplier circuits. Once, when exhibit multipliers were utilized, the diminished number of created fractional items altogether improved multiplier execution. In structures dependent on decrease trees with logarithmic rationale profundity, be that as it may, the diminished number of halfway items limitedly affects generally speaking execution. The Baugh-Wooley calculation [5] is an alternate plan for marked increase, however isn't so broadly received in light of the fact that it might be confounded to send on sporadic decrease trees. Again the Baugh-Wooley calculation is for just marked number increase. The exhibit multipliers [6] and Braun cluster multipliers [7] works just on the unsigned numbers. In this manner, the necessity of the cutting edge PC framework is a committed and extremely rapid multiplier unit that can perform increase task on marked just as unsigned numbers. In this paper we structured and executed a devoted multiplier unit that can perform augmentation task on both marked and unsigned numbers, and this multiplier is called as SUMBE multiplier.



II. LITERATURE REVIEW

Nikhil AdvaithGudala et al. [1], one of the most substantial areas of research in VLSI is the design of power-efficient and high-speed data path logic systems. The speed of addition is constrained in digital adders by the time taken to propagate a carry through the adder. When the previous bit has been added and the carry generated from this addition is propagated onto the next bit, the sum for each bit in an elementary adder is generated sequentially in this manner. In several computer systems, the Carry Select Adder (CSLA) is used to mitigate the issue of carry propagation delay by generating multiple carry bits and then selecting a carry for the desired output. The CSLA, however, is not area-efficient since it utilizes several Ripple Carry Adders (RCA) pairs to produce partial sum and carry by considering carry data, and then multiplexers pick the final sum and carry (mux). The core concept of this study is to use the Binary to Excess-1 Converter (BEC) to achieve high speed and low-power consumption instead of the RCA in the regular CSLA. The results have been analyzed and compared for implementation of three adders (conventional CSA, CSA with RCA, CSA with BEC). The results from the performance evaluations of the adders are compared with each other. All the simulation is carried out in 45nm technology. The delay of CSA and CSA with RCA are same, but the main difference is in reduction of area and power. Similarly when CSA(BEC) and CSA(RCA) are compared, the area has been reduced by approximately 18.67% and power has been reduced by 25.85%.

S. Venkatachalam et al. [2], approximate computing is an emerging technique in which power-efficient circuits are designed with reduced complexity in exchange for some loss in accuracy. Such circuits are suitable for applications in which high accuracy is not a strict requirement. Radix-4 modified Booth encoding is a popular multiplication algorithm which reduces the size of the partial product array by half. In this paper, three Approximate Booth Multiplier Models (ABM-M1, ABM-M2, and ABM-M3) are proposed in which approximate computing is applied to the radix-4 modified Booth algorithm. Each of the three designs features a unique approximation technique that involves both reducing the logic complexity of the Booth partial product generator and modifying the method of partial product accumulation. The proposed approximate multipliers are demonstrated to have better performance than existing approximate Booth multipliers in terms of accuracy and power. Compared to the exact Booth multiplier, ABM-M1 achieves up to a 23 percent reduction in area and 15 percent reduction in power with a Mean Relative Error Distance (MRED) value of 7.9×10^{-4} . ABM-M2 has area and power savings of up to 51 and 46 percent respectively with a MRED of 2.7×10^{-2} . ABM-M3 has area savings of up to 56 percent and power savings of up to 46 percent with a MRED of 3.4×10^{-3} . The proposed designs are compared with the state-of-the-art existing multipliers and are found to outperform them in terms of area and power savings while maintaining high accuracy. The performances of the proposed designs are demonstrated using image transformation, matrix multiplication, and Finite Impulse Response (FIR) filtering applications.

D. Kalaiyarasi et al. [3], this paper shows the plan of a proficient High speed Radix-4 Booth multiplier for both marked and unsigned numbers. The Proposed Booth multiplier is the competent multiplier which treats both positive and negative number reliably unlike ordinary multiplier. For the most part augmentation can be performed by include and move activity, in which each multiplier bit makes one various piece of the multiplicand that must be added to the halfway item. The bigger number of multiplicand must be included when the multiplier is bigger in size; thus the deferral of the multiplier is high. Since the deferral relies on the quantity of expansion activity. To acquire better execution, we have to limit the quantity of expansion which thus lessens the quantity of halfway item. The productive calculation that will limit the quantity of multiplicand is Booth calculation. It has been demonstrated that it tends to be helpful to apply the proposed Booth engineering in rapid multipliers on account of the increase in time acquired because of decrease of fractional items to $k/2$. The proposed multiplier builds the speed by 62.411% than the Array multiplier with practically same region (LUTs).

ElisardoAntelo Paolo et al. [4], in this paper, we portray an enhancement for paired radix-16 (changed) Booth recoded multipliers to diminish the most extreme tallness of the incomplete item segments to $\lceil n/4 \rceil$ for $n = 64$ -bit unsigned operands. This is rather than the customary greatest tallness of $\lceil (n + 1)/4 \rceil$. In this manner, a decrease of one unit in the greatest stature is accomplished. This decrease may include adaptability during the plan of the pipelined multiplier to meet the structure objectives, it might permit further advancements of the halfway item cluster decrease organize as far as territory/delay/control or potentially may enable extra addends to be incorporated into the incomplete item exhibit immediately. The strategy can be stretched out to Booth recoded radix-8 multipliers, marked multipliers, consolidated marked/unsigned multipliers, and different estimations of n .



T. Adiono et al. [5], in this paper, a 21×21 -piece, 2s-supplement double numbers, radix-8 multiplier unit plan for explicit object is introduced. Explicit reason implies that the multiplicand has a place with a recently known arrangement of numbers which are put away in a memory, so it very well may be utilized to make an adjustment in the radix-8 duplication calculation. We can adjust the past snake with the goal that it runs quicker. A full-custom format level structure and electrical recreation of the multiplier unit has been finished. Results have been contrasted with what we could get applying an increasingly regular radix-4 multiplier design.

H. Jiang et al. [6], the Booth multiplier has been generally utilized for superior marked duplication by encoding and along these lines decreasing the quantity of fractional items. A multiplier utilizing the radix-4 (or changed Booth) calculation is effective because of the simplicity of halfway item age, though the radix-8 Booth multiplier is moderate because of the intricacy of creating the odd products of the multiplicand. In this paper, this issue is eased by the utilization of surmised structures. An estimated 2-bit viper is intentionally intended for ascertaining the total of $1 \times$ and $2 \times$ of a paired number. This snake requires a little region, a low power and a short basic way delay. Along these lines, the 2-bit snake is utilized to actualize the less huge segment of a recoding viper for creating the triple multiplicand with no convey spread. In the quest for an exchange off among precision and power utilization, two marked 16×16 piece inexact radix-8 Booth multipliers are structured utilizing the surmised recoding viper with and without the truncation of various less noteworthy bits in the halfway items. The multiplier with 15-bit truncation accomplishes the best generally speaking exhibition as far as equipment and precision when contrasted with other surmised Booth multiplier plans. At long last, the estimated multipliers are connected to the plan of a low-pass FIR channel and they show preferred execution over other rough Booth multipliers.

Honglan Jiang et al. [7], the Booth multiplier has been broadly utilized for superior marked increase by encoding and in this way lessening the quantity of fractional items. A multiplier utilizing the radix-4 (or altered Booth) calculation is effective because of the simplicity of incomplete item age, while the radix-8 Booth multiplier is moderate because of the unpredictability of creating the odd products of the multiplicand. In this paper, this issue is mitigated by the use of inexact plans. An inexact 2-bit viper is intentionally intended for ascertaining the whole of $1 \times$ and $2 \times$ of a double number. This viper requires a little territory, a low power and a short basic way delay. In the quest for an exchange off among exactness and power utilization, piece surmised radix-8 Booth multipliers are structured utilizing the inexact recoding snake with and without the truncation of various less noteworthy bits in the incomplete items. The proposed surmised multipliers are quicker and more power proficient than the precise Booth multiplier. At last, the surmised multipliers are connected to the plan of a low-pass FIR channel and they show preferable execution over other estimated Booth multipliers.

III. DIFFERENT TYPES OF MULTIPLIER

These are the conventional multipliers having regular structures. Add and shift algorithm is used for its operation and hence its circuit is based on this algorithm. By direct mapping of the manual multiplication into hardware, an Array multiplier circuit can be developed [8]. An array of adder circuits can be used to accumulate partial products. The partial products are generated by multiplying the multiplicand with each bit of multiplier. The bit order decides the amount of shift of partial products. At the final stage, the partial products are added. The number of generated partial products is equal to that of multiplier bits. If multiplier length is equal to N, then N-1 numbers of adders are required to implement array multiplier [9].

Vedic Multiplier

Vedic Mathematics is an ancient system of mathematics existed in India. In this eminent approach, methods of basic arithmetic are simple, powerful and logical. Another advantage is its regularity. These advantages make Vedic Mathematics an important topic for research. Vedic Mathematics rules are mainly based on sixteen Sutras. Out of these sixteen Sutra's UrdhvaTriyakbhyam sutras and Nikhilamsutras are used for multiplication. Vedic multipliers are considered to be the best compared with conventional multipliers and UrdhvaTriyakbhyam Sutra based multiplication is more efficient compared to that of Nikhilam Sutra [10]. Implementation of Vedic mathematics on FPGA is easy due to its regularity and simplicity [4].

All partial products required for multiplication are calculated much before actual multiplication begins. This is the big advantage of this multiplication. Based on the Vedic Mathematics algorithm, these partial products are added to obtain final product which leads to a very high speed approach. Multiplier designs based on Vedic mathematics are with high speed and consume relatively low power. Multipliers are the basic and key blocks of a Digital Signal processor.



Multiplication is the key process in improving the computational speed of Digital Signal Processors [6]. Convolution, Fast Fourier transforms and various other transforms make use of multiplier blocks [11]. Among various methods of multiplications in Vedic mathematics, UrdhvaTiryagbhyam is efficient. UrdhvaTiryagbhyam is a general multiplication formula applicable to all cases of multiplication.

Booth Series of Multipliers

There is no need to take the sign of the number into deliberation in dealing with unsigned multiplication. However in signed multiplication the process will be changed because the signed number is in a 2’s compliment pattern which would give a wrong result if multiplied by using similar process for unsigned multiplication [6]. Booth’s algorithm is used for this. Booth’s algorithm preserves the sign of the result. Booth multiplication allows for smaller, faster multiplication circuits through encoding the signed numbers to 2’s complement, which is also a standard technique used in chip design, [6] and provides significant improvements by reducing the number of partial product to half over “long multiplication” techniques. Radix 2 is the conventional booth multiplier.

Radix 2

In booth multiplication, partial product generation is done based on recoding scheme e.g. radix 2 encoding. Bits of multiplicand (Y) are grouped from left to right and corresponding operation on multiplier (X) is done in order to generate the partial product [19]. In radix-2 booth multiplication partial product generation is done based on encoding which is as given by Table1. Parallel Recoding scheme used in radix-2 booth multiplier is shown in the Table 1.

Table 1: Booth recoding for radix 2

Q _n	Q _{n+1}	Recoded Booth	Operation
0	0	0	Shift
0	1	+1	Add x
1	0	-1	Subtract x
1	1	0	Shift

Radix 4

One of the solutions attaining high speed multipliers is to improve parallelism. It helps in decreasing the number of consecutive calculation stages [1]. The Original version of Booth’s multiplier (Radix – 2) had two drawbacks [7]. The number of Add or Subtract operations became variable and hence became difficult while designing Parallel multipliers. The Algorithm becomes disorganized when there are isolated 1s. These problems are overthrown by using Radix 4 Booth’s algorithm which can browse strings of three bits with the algorithm. The above recoding has the nice feature that they translate into the partial products shown in table 2.

Table 2: Booth recoding for radix 4

Multiplier Bits Block			Recoded 1-bit pair		2-bit booth	
i+1	i	i-1	i+1	i	Multiplier Value	Partial Product
0	0	0	0	0	0	Mx0
0	0	1	0	1	1	Mx1
0	1	0	1	-1	1	Mx1
0	1	0	1	0	2	Mx2
1	0	0	-1	0	-2	Mx-2
1	0	1	-1	1	-1	Mx-1
1	1	0	0	-1	-1	Mx-1
1	1	0	0	0	0	Mx0

IV. METHODOLOGY

According to this algorithm the multiplier term is encoded. For the coding, three successive bits of the multiplier operand are considered, such that each block overlaps the previous block by one bit. Grouping starts from the LSB, and the first block only uses two bits of the multiplier; the first bit is the (n – 1)th bit of the multiplier. The encoding process is shown as in the figure 1.

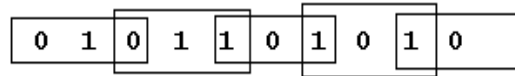


Figure 1: Grouping of bits from the multiplier term, for using Radix-4 Booth recoding

In modulo $2^n + 1$ multiplication, the result and one input use weighted representations, while the other input uses diminished-1 representation.

Let $d[A] = (a_n a_{n-1} \dots a_0)$ be the diminished-1 representation of the weighted binary number A , $B = (b_n b_{n-1} \dots b_0)$ and $P = |A \times B|_{2^n+1}$ all be weighted binary numbers. The algorithm uses $d[A]$ and B to compute the value of P , that is

$$P = (A \times B) \bmod (2^n + 1) = |A \times B|_{2^n+1} = F(d[A], B) \quad (1)$$

Where $F()$ is a function that maps the inputs $d[A]$ and B to the output P .

The modulo $2^n + 1$ multipliers can be used for the situation where one operand being constant and another operand being variable. In that situation, the diminished-1 representation of the constant operand and can be pre-computed and be present in the hardware, so its conversion circuits are not required.

The algorithm uses the radix-4 Booth recoding [11] where successive overlapping triplets of B are examined and encoded. The outputs of the encoder uses the $d[A]$ to form the partial products. The partial product reduction scheme uses the well-known inverted end-around-carry (IEAC) adder tree [12]. The final adder generates the product. To avoid $(n+1)$ -bit circuits, we distinguish the following two cases:

When $\overline{a_n} + \overline{b_n} = 1$; and when $\overline{a_n} + \overline{b_n} = 0$

Table 3: Encoding for $-b_{n-1} + b_0 - 2b_1$ and $-b_{n-1} + b_0 + 2b_1$

b_1	b_0	b_{n-1}	Encoding	Formulas
0	0	0	0	$-b_{n-1} + b_0 - 2b_1$
0	1	0	1	
1	0	0	-2	
1	1	0	-1	$-b_{n-1} + b_0 + 2b_1$
0	0	1	-1	
0	1	1	0	
1	0	1	1	
1	1	1	1	

Booth Encoder and Booth Selector

Due to the encoding scheme accordant with the radix-4 Booth recoding, the partial product generator (PPG) can be constructed with the well-known Booth encoder (BE) and Booth selector (BS). Many implementations for BE block and BS block were published, but they can be reduced to two categories: one having 4-bit bus and the other having 3-bit bus. Here, the 3-bit bus approach is used.

The BE block examines successive overlapping triplets of $b_{2i-1} + b_{2i} - 2b_{2i+1}$ and encodes each to 0, ± 1 , and ± 2 . Each BE block produces 3 bits: 1x, 2x and Sign. The 3 bits along with the multiplicand $d[A]$, are used to form partial products. The partial products are produced by the BS blocks. Each BS block takes as inputs two successive bits of $d[A]$. There are two types of BS blocks in the multipliers BS^+ and BS^- since the inverted multi-bit left-circular shifts of $d[A]$ and $d[-A]$ are different. For the i -th partial products, $0 \leq i \leq K$, there are $2iBS^-$ blocks and $(n - 2i)BS^+$ blocks are required.

V. CONCLUSION

In all multiplication operation product is obtained by adding partial products. Thus the final speed of the multiplier circuit depends on the speed of the adder circuit and the number of partial products generated. If radix 4 Booth encoding technique is used then there are only 3 partial products and for that only one CSA and a CLA is required to produce the final product. Due to the encoding scheme accordant with the radix-4 Booth recoding, the partial product generator (PPG) can be constructed with the well-known Booth encoder (BE) and Booth selector (BS). In step second, due to one operand A is used as diminished-1 number representation $d[A]$, one correction is generated. In step third, the partial products and the correction term which are n -bits each are added through a inverted end around carry (IEAC) adder tree.



REFERENCES

- [1] Nikhil AdvaitGudala, TrondYtterdal, John J. Lee and Maher Rizkalla, "Implementation of High Speed and Low Power Carry Select Adder with BEC", International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE 2021.
- [2] S. Venkatachalam E. Adams H. J. Lee and S.-B. Ko "Design and analysis of area and power efficient approximate booth multipliers" IEEE Trans. Comput. vol. 68 no. 11 pp. 1697-1703 Nov. 2019.
- [3] D. Kalaiyarasi and M. Saraswathi, "Design of an Efficient High Speed Radix-4 Booth Multiplier for both Signed and Unsigned Numbers", 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), IEEE 2018.
- [4] ElisardoAntelo Paolo Montuschi Alberto Nannarelli "Improved 64-bit Radix-16 Booth Multiplier Based on Partial Product Array Height Reduction" IEEE Transactions On Circuits And Systems-I: Regular Papers vol. 64 no. 2 February 2017.
- [5] T. Adiono H. Herdian S. Harimurti "Full-Custom Design Implementation of Serial Radix-4 8-bit Booth Multiplier" The 2nd International Conference on Electrical Engineering and Computer Science (ICEECS 2016) 2016.
- [6] H. Jiang J. Han F. Qiao 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.
- [7] Honglan Jiang Jie Han FeiQiaoFabrizio 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.
- [8] A. Rama Vasantha M. Tech M. Sai Satya Sri "Design and Implementation of FPGA Radix-4 Booth Multiplication Algorithm" International Journal of Research in Computer and Communication Technology vol. 3 no. 9 September 2014.
- [9] Ravi H Bailmare, S. J. Honale And Pravin V Kinge, "Design And Implementation of Adaptive FIR Filter using Systolic Architecture", In International Journal of Current Engineering And Technology , Vol.4, No.3, June 2014.
- [10] M. Usha, R. Ramadoss, "An Efficient Adaptive Fir Filter Based On Distributed Arithmetic", International Journal of Engineering Science Invention, Vol. 3, Issue. 4, pp. 15-20, April 2014.
- [11] Sukhmeet Kaur Suman ManpreetSingh Manna "Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2)" Advance in Electronic and Electric Engineering vol. 3 no. 6 pp. 683-690 2013 ISSN 2231-1297.
- [12] Sang Yoon Park and Pramod Kumar Meher, "Low power, High-throughput And Low- Area Adaptive FIR Filter Based on Distributed Arithmetic", in IEEE Transactions On Circuits And Systems-ii, Vol. 60, No. 6, pp. 346- 350, 2013.
- [13] Basant K. Mohanty, And Pramod Kumar Meher, "A High-Performance Energy- efficient Architecture For FIR Adaptive Filter Based On New Distributed Arithmetic Formulation Of Block LMS Algorithm", In IEEE Transactions On Signal Processing, Vol. 61, No. 4, February, 2013.
- [14] PallaviSaxena, Urvashi Purohit, Priyanka Joshi, "Analysis of Low Power, Area Efficient and High Speed Fast Adder", In International Journal Of Advanced Research In Computer And Communication Engineering, Vol. 2, Issue 9, September 2013.
- [15] A. Fathi S. Azizian R. Fathi H.G. Tamar "Low latency glitch-free booth encoder-decoder for high speed multipliers" IEICE Electronics Express vol. 9 no. 16 pp. 1335-1341 2012.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details