COMPARATIVE ANALYSIS OF VARIOUS TYPES OF MULTIPLIERS FOR EFFECTIVE LOW POWER AND TIME

KATTA VAMSI

Post Graduation Student, Department Of ECE, JNTU, UCEK, Kakinda, Andhra Prdesh, India

ABSTRACT:

The main objective of this proposed concept is to design an efficient and enhanced multiplier using gate level implementation. In this present study, the comparison of the three types of multiplier architectures are conceded out by means of area and time design methods using VHDL code.Both 4 bit and 8 bit multiplier designs are implemented for particular special approach of applications. The sub block in the multiplier construction was also substantiated to improve low area and time. Wallace tree multiplier is used for high speed operation; Baugh Wooleymultiplier is applied for signed multiplication by a smaller amount of delay. Radix 8 modified booth encoding algorithm is used for low density and latency optimized applications. Proposed radix8 booth is elected for reduction of more partial products in order to optimize particular area constraints.

KEYWORDS: Baugh Wooley¸ Modified Booth encoding, Wallace, Radix, Density, Latency.

INTRODUCTION: Now-a-days, the arithmetic operation is the basic operation for all the innovative processes. The arithmetic operation has addition, subtraction, multiplication and division. Among these multiplication and addition is the frequently using operations. Multiplication is to add an integer to a specified number of times by itself. Multiplication is done with two numbers, multiplicand and multiplier. The Multiplication is done by adding multiplicand to itself for a number of times, as specified by another number called Multiplier. The output generated is the product value of the two binary numbers. The multiplicand is multiplied by each digit of the multiplier beginning with the LSD. Intermediate results are called as the partial products. The final product is determined by summation of all the partial-products. Multiplication involves three main steps. They are Partial product generation, Partial product reduction and Final addition.For the multiplication of an N-bit multiplicand with an M-bit multiplier, Mpartial products are generated and product formed is N+ Mbits long.Multiplication is an important fundamental function in arithmetic logic operation. Computational performance of a DSP system is limited by its multiplication performance and since, multiplication dominates the execution time of most DSP algorithms [3]; therefore high-speed multiplier is much desired. Currently, multiplication time is still the dominant factor in determining the instruction cycle time of a DSP chip. With an ever-increasing quest for greater computing power on battery-operated mobile devices, design emphasis has shifted from optimizing conventional delay time area size to minimizing power dissipation while still maintaining the high performance.Digital circuit design uses digital multipliers, which

are fast, reliable and have efficient components. They can do manyoperations with less number of components. Depending upon the arrangement of the components, the various multipliers are available. Particular multiplier architecture is chosen based on the application. In many DSP algorithms, the multiplier has critical path and determines the performance of the algorithm. The speed of multiplication operation is of great importance in DSP as well as in general processor. In the past, multiplication was implemented generally with a sequence of addition and shifting operations. Traditionally shift and add algorithm has been implemented to design however this is not suitable for VLSI implementation and also from delay point of view. Some of the important algorithm proposed in literature for VLSI implementable fast multiplication is Booth multiplier, array multiplier and Wallace tree multiplier.

LITERATURE SURVEY:

PallaviSaxena et al (2015) [1] proposed a paper on "Design of Low Power and High Speed Carry Select Adder Using Brent Kung Adder". In this paper constraint carry select adder architectures are proposed using parallel prefix adders (Brent Kung Adder) is used to design linear Carry Select Adder. By using Carry look ahead adder to derive fast results but they leads to increase in area. SandeepKakde et al (2015) [2] proposed a paper on "Design of Area and Power Aware Reduced Complexity Wallace Tree Multiplier". In this paper, the work has been done to reduce the area by using energy efficient CMOS full adder. By using the multiplier, the system complexity is increased. ShradhaAgrey et al (2015) [3] proposed a paper on "Comparative Analysis of Different Adders for Wallace Tree Multiplier". Booth algorithm is used to reduce the number of input bits required for the multiplication to be correct. Rajaram S et al (2011) [4] proposed a paper on "Improvement of Wallace Multipliers using Parallel Prefix Adders". In this paper, employing parallel prefix adders (fast adders) at the final stage of Wallace multipliers to reduce the delay.

SYSTEMATIC LITERATURE:

A)Wallace Multiplier: At 1964, C. S. Wallace has proposed the new fast multiplier scheme Known as WM based on the sequential adding stages reduction by reducing the Partial Product (PP) accumulation [4]. In 1998, M. E. Robinson and E. Swartzlander Junior has proposed WM by using 4:3 counters for optimizing the hardware performance and found up to 10% less delay [5]. In 2010, Ron S. Waters and E. Swartzlander Junior has modified the conventional WM with a nearly same delay [6]. In 2011, S. Rajaram and K. Vanithamanihas proposed a WM which reduce the delays [7], [8], [9]. The modified WM reduces 80% of HAs [10]. Fast column compression techniques in multiplication have been acquired by using combination of two different designs. The results demonstrated that fast column compression multiplier is 41.1% faster than the 64-bit regular WM [11]. 8-Bit hybrid tree multiplier is developed by combining Wallace and Dadda methods and found 40% of power reduction [12]. The modifications of Wallace/Dadda multiplier use carry-look-ahead adders as a replacement of full adders [13], [14]. In 2018, E. JagadeeswaraRao (2018) proposed high speed WM [15]. In this design a high-speed adder with 4-2 and 8-2 adder compressor was used at reduction stage and increases speed 25% in comparison of reported WMs [15].

B. Modified Booth Multiplier In 1950, Booth Donald has proposed new algorithm for multiplying two unsigned (or signed) numbers which is known as Booth multiplier [16]. In 2000, Chang Yeh and Chein Wei Jen has proposed a new modified Booth Encoding Scheme (MES) [17]. It increases 25% speed in comparison of conventional MBM [17]. In same year 2000, Fayez Elguibaly has proposed MBM with parallel MAC unit [18]. It was three times faster than the standard MAC unit. In 2007, Zhou Shun et al. has proposed a radix-4 MBM withmulti precision reconfigurable scheme which can be cascaded to comply with the different input length [19]. In 2010, S. R. Kuang and J. P. Wang has proposed low power configurable MBM [20]. But it had extra overhead circuit, it has hardware overhead in comparisons of the regular multipliers but their power consumption is significantly reduced. In 2012, R. P. Rajput and M. N. S. Swamyhas proposed a high speed MBM which uses CSA and CLA [21]. In 2014, Kostas Tsoumanis et al. has proposed MBM with Fused Add Multiply (FAM) [22]. In 2016, K. Tsoumanis and N. Axelos has developed a new MBM hardware with pre-encoded scheme, in which reduce the area at encoding stage also reduce delay and power consumption [23].

C. Baugh-Wooley Multiplier

Magnus Sjalander and Per Larsson-Edefors, An 8-bit Baugh-Wooley algorithm case, According to Hatamian's scheme the partial-product bits have been reconstructed. Here they clearly verified that the reduction arrays are basedon BW implementations. In this paper,[24][25]however, they have chosen the HPM reduction and the implementation of BW algorithm.By using a 2-input AND gate for every pair of multiplicand bits of the partial-product bits are generated. Where the partial-product bit to be inverted, rather they used a 2-input NAND gate[26].

WALLACE TREE MULTIPLIER:

Waters et al. presented reduced complexity Wallace multiplier reduction approach [2]. It is a modification to the second phase reduction method used in the conventional Wallace multipliers, in which number of the half adders is greatly reduced. In the first phase, the partial product array is formed and it is converted in the form of an inverted pyramid array. An inverted pyramid array is formed when the bits in the left half of the partial product array is shifted in the upward direction. In the second phase, this array is divided into group of three rows each and full adders are used in each column. Half adders are used only when the number of reduction stages of the modified Wallace multiplier is exceeding that of the conventional Wallace multiplier. According to equation (1) in the modified Wallace multiplier, if (ri mod 3) = 0, then half adder is needed in the reduction stage otherwise half adder is not required. The number of half adders was observed to be (N-S-1). In the modified Wallace 9 by 9 bit reduction, only one half adder is used in the first and the second phase and two half adders are used in the final phase as shown in Figure 2. In the third phase, (2N-2) bit carry propagating adder is used. Therefore, we observed that the number of the reduction stages remain same as compared to the conventional Wallace reduction whereas two more full adders and 17 fewer half adders are used in the modified Wallace multiplier. Both multipliers yield same performance in the terms of delay and have same number of the reduction stages, but the modified Wallace multiplier has the advantage of reduced complexity as number of half adders is 80% less than the conventional Wallace multiplier in the second phase. However

due to reduction in number of half adders, the total gate count in modified Wallace reduction is always less than that of the conventional Wallace reduction. The number of full adders is somewhat increased between 1-5 for 8-64 bit modified Wallace multiplier. Several popular and well-known schemes, with the objective of improving the speed of the parallel multiplier, have been developed in past. Wallace introduced a very important iterative realization of parallel multiplier. This advantage becomes more pronounced for multipliers of bigger than 16 bits. In Wallace tree architecture, all the bits of all of the partial products in each column are added together by a set of counters in parallel without propagating any carries. Another set of counters then reduces this new matrix and so on, until a two-row matrix is generated. The most common counter used is the 3:2 counter which is a Full Adder.. The final results are added using usually carry propagate adder. The advantage of Wallace tree is speed because the addition of partial products is now O (logN). A block diagram of 4 bit Wallace Tree multiplier is shown in below. As seen from the block diagram partial products are added in Wallace tree block.

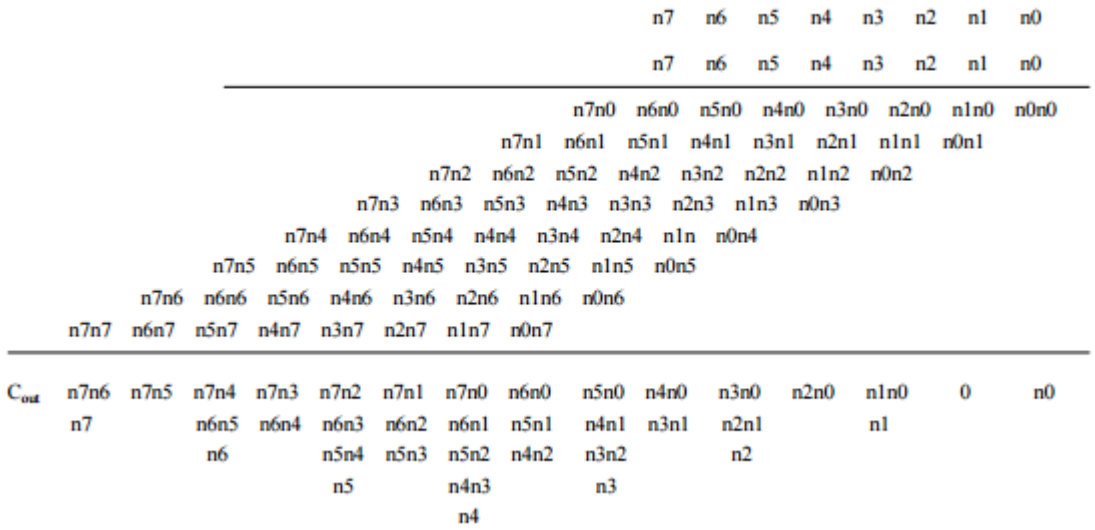

Fig1: Wallace Tree Multiplier

BAUGH-WOOLEY MULTIPLIER

This technique has been developed in order to design regular multipliers, suited for 2's-complement numbers.

Let us consider 2 numbers A and B :

$$A = (a_{n-1} \dots a_0) = -a_{n-1} \cdot 2^{n-1} + \sum_{0}^{n-2} a_i \cdot 2^i$$

$$B = (b_{n-1} \dots b_0) = -b_{n-1} \cdot 2^{n-1} + \sum_{0}^{n-2} b_i \cdot 2^i$$

The product A.B is given by the following equation :

$$A.B = a_{n-1}.b_{n-1}.2^{2n-2} + \sum_{0}^{n-2}\sum_{0}^{n-2} a_i.b_i.2^{i+j} - a_{n-1}\sum_{0}^{n-2} b_i.2^{n+i-1} - b_{n-1}\sum_{0}^{n-2} a_i.2^{n+i-1}$$

We see that subtractor cells must be used. In order to use only adder cells, the negative terms may be rewritten as :

$$-a_{n-1}\sum_{0}^{n-2} b_i.2^{i+n-1} = a_{n-1}.\left(-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{b_i}.2^{i+n-1}\right)$$

By this way, A.B becomes :

$$A.B = a_{n-1}.b_{n-1}.2^{2n-2} + \sum_{0}^{n-2}\sum_{0}^{n-2} a_i.b_j.2^{i+j}$$
$$+ b_{n-1}\left[-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{a_i}.2^{i+n-1}\right]$$
$$+ a_{n-1}\left[-2^{2n-2} + 2^{n-1} + \sum_{0}^{n-2} \overline{b_i}.2^{i+n-1}\right]$$

The final equation is :

$$A.B = -2^{2n-1} + (\overline{a_{n-1}} + \overline{b_{n-1}} + a_{n-1}.b_{n-1}).2^{2n-2}$$
$$+ \sum_{0}^{n-2}\sum_{0}^{n-2} a_i.b_j.2^{i+j} + (a_{n-1} + b_{n-1}).2^{n-1}$$
$$+ \sum_{0}^{n-2} b_{n-1}.\overline{a_i}.2^{i+n-1} + \sum_{0}^{n-2} a_{n-1}.\overline{b_i}.2^{i+n-1}$$

A and B are n-bits operands, so their product is a 2n-bits number.

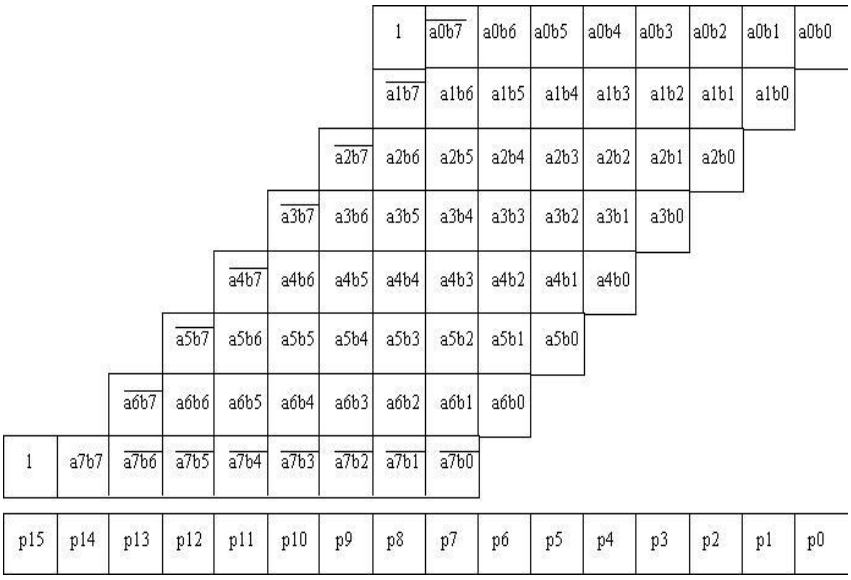

Fig2: Baugh-wooley Multiplier

BOOTH ENCODING MULTIPLIER:

It is a powerful algorithm for signed-number multiplication, which treats both positive and negative numbers uniformly. For the standard add-shift operation, each multiplier bit generates one multiple of the multiplicand to be added to the partial product. If the multiplier is very large, then a large number of multiplicands have to be added. In this case the delay of multiplier is determined mainly by the number of additions to be performed. If there is a way to reduce the number of the additions, the performance will get better.

Booth algorithm is a method that will reduce the number of multiplicand multiples. For a given range of numbers to be represented, a higher representation radix leads to fewer digits. Since a k-bit binary number can be interpreted as K/2-digit radix-4 number, a K/3-digit radix-8 number, and so on, it can deal with more than one bit of the multiplier in each cycle by using high radix multiplication.

RADIX-8 MODIFIED BOOTH ALGORITHM:

The Booth algorithm consists of repeatedly adding one of two predetermined values to a product P and then performing an arithmetic shift to the right on P.

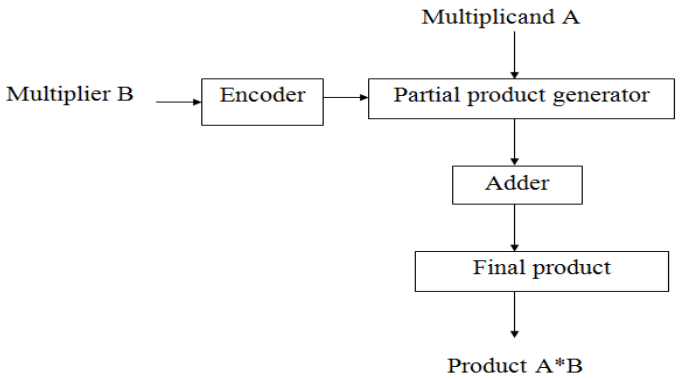

Fig3. Booth algorithm

The multiplier architecture consists of two architectures, i.e., Modified Booth. By the study of different multiplier architectures, we find that Modified Booth increases the speed because it reduces the partial products by half. Also, the delay in the multiplier can be reduced by using Wallace tree. The energy consumption of the Wallace Tree multiplier is also lower than the Booth and the array. The characteristics of the two multipliers can be combined to produce a high-speed and low-power multiplier.

The modified stand-alone multiplier consists of a modified recorder (MBR). MBR has two parts, i.e., Booth Encoder (BE) and Booth Selector (BS). The operation of BE is to decode the multiplier signal, and the output is used by BS to produce the partial product. Then, the partial products are added to the Wallace tree adders, similar to the carry-save-adder approach. The last transfer and sum output line are added by a carry look-ahead adder, the carry being stretched to the left by positioning.

Table1. Quartet coded signed-digit table

Quartet value	Signed-digit value
0000	0
0001	+1
0010	+1
0011	+2
0100	+2
0101	+3
0110	+3
0111	+4
1000	-4
1001	-3
1010	-3
1011	-2
1100	-2
1101	-1
1110	-1
1111	0

Here we have a multiplication multiplier, 3Y, which is not immediately available. To Generate it, we must run the previous addition operation: 2Y + Y = 3Y. But we are designing a multiplier for specific purposes and then the multiplier belongs to a set of previously known numbers stored in a memory chip. We have tried to take advantage of this fact, to relieve the radix-8 bottleneck, that is, 3Y generation. In this way, we try to obtain a better overall multiplication time or at least comparable to the time, we can obtain using a radix-4 architecture (with the added benefit of using fewer transistors). To generate 3Y with 21-bit words you just have to add 2Y + Y,ie add the number with the same number moved to a left position.

RESULTS:

Multipliers	Delay time (ns)	Area	power (W)
Array	32.352	259	0.236
Baugh wooley	30.614	263	0.243
walace	22.978	194	0.036
Radix8 modified booth 4bit	17.693	128	0.034
Radix8 modified booth 8bit	20.750	422	0.053

CONCLUSION:

Pipelined large word length digital multipliers are difficult to design under the constraints of core cycle time (for nominal voltage), pipeline depth, power and energy consumption and area. Low level optimizations might be required to meet these constraints. In this project , we have presented a method to reduce by one the maximum height of the partial product array for 8-bit radix-8 Booth recoded magnitude multipliers. This reduction may allow more flexibility in the design of the reduction tree of the pipelined multiplier.

REFERENCES:

1. Ron S., Waters and Earl E.Swartzlander-A Reduced Complexity Wallace multiplier reduction, IEEE transactions on Computers, 2010.

2.PallaviSaxena, UrvashiPurohit, Priyanka Joshi -Analysis of Low Power, Area- Efficient and High Speed Fast Adder, International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 9.

3.SohanPurohit and Martin Margala -Investigating The Impact Of Logic And Circuit Implementation On Full Adder Performance, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 7, 2012 .

4. Rajaram, S. and Mrs.K.Vanithamani -Improvement of Wallace multipliers using Parallel prefix adders. Proceedings of 2011 International Conference on Signal Processing, Communication, Computing and Networking Technologies IEEE, 2011.

5. Moises E. Robinson and Earl Swartzlander, Jr., "A Reduction Scheme to Optimize the Wallace Multiplier", IEEE Proc. Int. Conf., Comput. Design: VLSI in Comput. and Processors, pp. 122-127, Oct. 1998.

6. Ron S. Waters and Earl E. Swartzlander, Jr., "A Reduced Complexity Wallace Multiplier Reduction", IEEE Trans., pp. 1134-1137, Aug., 2010.

7. S. Rajaram and K. Vanithamani, "Improvement of Wallace multipliers using Parallel prefix adders", IEEE Int. Conf. on Signal Process., Comm., Computing and Networking Technologies, pp. 781-784, July, 2011.

8. D. Shin and S. Kunita, "Approximate logic synthesis for error tolerance applications", Proc. of Design, Automatic and Test in Europe Conf. and Exhibition (DATE), pp. 957-960, 2010.

9. A.B. Kahng and S. Kang, "Accuracy-configurable adder for Approximate Arithmetic Design", Proc. of Design Automatic Conf. (DAC), pp. 820-825, 2012.

10. S. Anju, and M. Saravanan, "High Performance Dadda Multiplier Implementation Using High Speed Carry Select Adder", Int. J. of Adv. Res. in Computer and Comm. Engg., 2(3), 2013.

11. Z. Wang, G. A. Jullien and W.C. Miller, "A New Design Technique for Column Compression Multipliers", IEEE Trans. on Computers, 44(8), pp. 962-970, 1995.

12. P. Anitha and P. Ramanathan, "A New Hybrid Multiplier using Dadda and Wallace Method", Proc. of Int. Conf. on Electronics and Comm. Systems (ICECS), pp. 1-4, IEEE, 2014.

13. W. Chu, A. I. Unwala, P. Wu and E. Swartzlander, "Implementation of a High-Speed Multiplier using Carry Look-Ahead Adders", Proc. of Asilomar Conf. on Signals, Systems and Computers (pp. 400-404). IEEE, 2013.

14. P. Samundiswary, K. Anitha, "Design and Analysis of CMOS Based DADDA Multiplier", Int. J. of Comp. Engg. and Management IJCEM, 1(16), 12-17, 2013.

15. E. JagadeeswaraRao, K. Jayram Kumar and T.V. Prasad, "Design of high-speed Wallace Tree Multiplier using 8-2 and 4-2 adder compressor", Int. J. f Engg. And Tech., PP. 2386-2390, 2018.

16. A.D. Booth, "A Signed Binary Multiplication Technique," Jour. Of Mech. Appl. Math., vol. 4, pp. 236-240, Oxford University Press,1951.

17. Wen-Chang Yeh and Chein-Wei Jen, "High-Speed Booth Encoded Parallel Multiplier Design",, IEEE Trans. on Compt., vol. 49, no. 7, pp.692-701, July, 2000.

18. Fayez Elguibaly, "A Fast-Parallel MultiplierAccumulator Using the Modified Booth Algorithm", IEEE Trans. Circuits and Sys. II, Analog and Digital Signal Processing, vol. 57, no. 9, pp. 902-909, Sept., 2000.

19. Zhou Shun, Oliver A. Pf¨ander, Hans JorgPfleiderer and Amine Bermak, "A VLSI architecture for a Run-time Multi-precision Reconfigurable Booth Multiplier", 14th IEEE Int. Conf. on Electronics, Circuits and Systems, pp. 975-978, Dec., 2007.

20. Shiann-RongKuang and Jiun-Ping Wang, "Design of Power-Efficient Configurable Booth Multiplier", IEEE Trans., Circuits and Sys. II Regular Papers, vol. 57, no. 3, pp. 568-580, March, 2010. 21. Ravindra P Rajput and M.N ShanmukhaSwamy, "High speed Modified Booth Encoder multiplier for signed and unsigned numbers", 14th IEEE Int. Conf. on Modelling and Simulation, pp. 649-654, 2012.

22. Kostas Tsoumanis, Sotiris Xydis, ConstantinosEfstathiou, Nikos Moschopoulos and KiamalPekmestzi, "An Optimized Modified Booth Recoder for Efficient Design of the Add-Multiply Operator", IEEE Trans., Circuits and Systems I, Regular Papers, vol. 61, no. 4, pp. 1133-1143, April, 2014.

23. K. Tsoumanis, N. Axelos, N. Moshopoulos, G. Zervakis and K. Pekmestzi," Pre-Encoded Multipliers Based on Non-Redundant Radix-4 Signed-Digit Encoding", IEEE Trans. On Computers, vol. 65, pp. 670-676, Feb. 2016.

24. Bonetti, A. Teman, P. Flatresse, and A. Burg, "Multipliers-Driven Perturbation of Coefficients for Low-Power Operation in Reconfigurable FIR Filters," IEEE Trans. Circuits Syst. I Regul.Pap., vol. 64, no. 9, pp. 2388–2400, 2017.

25. Kiran and N. Prashar, "FPGA Implementation of High Speed BaughWooley Multiplier using Decomposition Logic," pp. 2–7.

26. K. D. C. Dandade and P. R. Indurkar, "Design of High Speed 16-Bit Vedic and Booth," vol. 5, no. Viii, 2017.