International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 2 Feb 2022, pp: 571-577 www.ijaem.net ISSN: 2395-5252

8 X 8 Signed Booth Encoded Parallel Multiplier for High Speed Applications

Dr V.Raju
Asst.Prof., Dept.of.ECE

Submitted: 01-02-2022 Revised: 07-02-2022 Accepted: 10-02-2022

ABSTRACT—

Several architectures are available for multipliers in the literature. A parallel multiplier can perform faster multiplication than a serial multiplier. Multiplication usually requires three steps. First step is to generate partial products, second step involves summing up the partial products until two rows remain and these two rows are added to form the final product in the third step. In this project, the multiplication is implemented with modified Booth Encoding technique. For generating a partial product a new radix-4 modified booth encoding (MBE) scheme will be used to improve the performance of traditional MBE schemes. Addition of the partial products is done by using partial product reduction tree (PPRT). The PPRT will be constructed using Three-Dimensional-reduction-Method (TDM). The addition at the final stage will be done by using multiple-level conditional-sum adder (MLCSMA) algorithm.

The design is implemented in Verilog HDL on Xilinx Vivado software and synthesized on FPGA board.

Keywords—booth encoding, partial product reduction tree, three dimensional reduction method, multi-level conditional sum adder,

I. INTRODUCTION

Arithmetic calculations are very important in many of the applications. Multiplication plays a major role in Digital Computing Systems, Digital Signal Processing applications like FFT, DFT, convolution, etc. This project involves the design and implementation of signed 8-bit parallel multiplier. The significant requirement of the multiplier to be designed is speed and high performance. The multiplication process involves three stages: generation of partial products, summation of all partial products until only two rows remain, and addition of the partial products. A carry propagation adder can be used for addition of the two remaining rows of partial products to compute the final product. In the first step, that is to generate partial products, two methods are commonly used. First method is directly to use an AND gate with two inputs. Second method employs an algorithm based on radix-4 modified booth's encoding [1],[2]. All the partial products can be summed up using a Partial Product Reduction Tree (PPRT). The summing up is usually done by a carry save tree or a Wallace tree. They usually have a basic element which is 3:2 counter i.e a full adder. Building a PPRT with large counters is not beneficial. By using 4:2 compressor instead of counters, the delay paths are balanced as 4:2 compressor has only a latency of three XOR gate delays instead of two full adder delays. To enhance the speed of the process, a Three Dimensional Reduction Method (TDM) is introduced [4],[5]. This project examines a faster approach by constructing PPRT with TDM and MBE [3]. PPRT requires an adder with fast carry-propagation for the addition of the final two rows of partial products to generate the product in two's complement format. The final adder is to be designed to handle the input signals which do not arrive simultaneously. This is contrary to the conventional carry-propagation adder design which handles all the inputs arriving simultaneously. In this design a high speed Booth encoded parallel multiplier is developed. The method uses a novel MBE scheme and an improved partial product array. Final addition is done by using MLCSMA algorithm. Fig.1 depicts the sequence of the process to be followed.

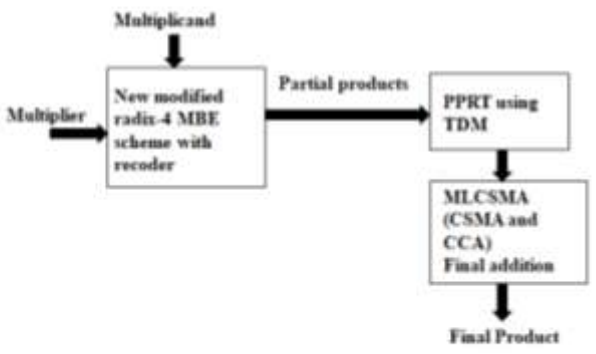

Multiplicand
Multiplier
New modified radix-4 MBE scheme with recoder
Partial products
PPRT using TDM
MLCSMA (CSMA and CCA) Final addition
Final Product

Fig. 1.Block Diagram of the Design

II. RADIX-4 MODIFIED BOOTH ENCODING SCHEME

Three MBE schemes are compared to analyze the best MBE scheme. The first MBE-I scheme from [6] uses ten transistors which makes it area efficient but its delay is more and may have poor power efficiency. The decoders of MBE-II and MBE-III are examined. In MBE-II scheme from [7] the x signal arrives at t=1.5 unit delays, the total delay are 3.5 units. The transistor count in the decoder can be reduced to 16 if 2-input XNOR gate is implemented with a pair of transmission gates and a pair of inverters and by grouping the two input AND and NOR gates into a complex gate.

In MBE-III scheme from [8] the incoming X signals reach at t = 1.0 unit delay, with a reduction of two unit delays in the total latency. The decoder of MBE-III can be implemented using 20 transistors.

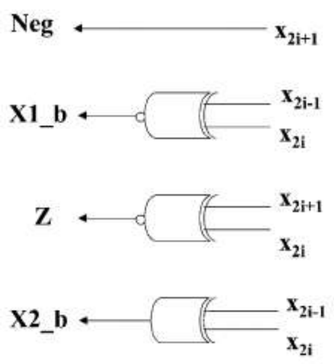


Fig.2. Encoder used in New MBE

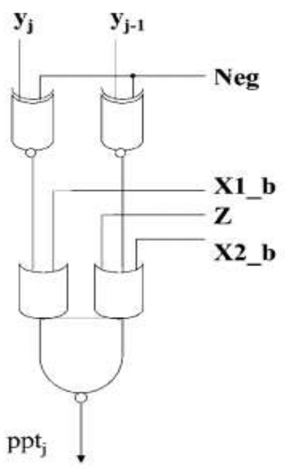


Fig. 3. Decoder used in new MBE

Table 1 shows the table of truth of the latest scheme of encoding. The output zero is produced by the Z signal to normalize the wrong X_{2_b} and NEG signals. In Figure 3.4, the encoder and decoder pair obtains 3-bit x and n-bit y inputs respectively. Fig. 2 and fig. 3 shows the encoder and decoder circuit diagram. The three input x-signals are encoded by the encoder to produce X_{1_b}, X_{2_b}, and Z signals. The LSB Y_{LSB} of the y signal in combination with x-signals evaluates the Row_LSB and Neg_cin signals. In a

X2i+1	X2i	X2i-1	value	X1_b	X2_b	Neg	z
0	0	0	0	1	0	0	1
0	0	1	1	0	1	0	1
0	1	0	1	0	1	0	0
0	1	1	2	1	0	0	0
1	0	0	-2	1	0	1	0
1	0	1	-1	0	1	1	0
1	1	0	-1	0	1	1	1
1	1	1	0	1	0	1	1

Table 1 Truth Table of New MBE Scheme

Similar way, to evaluate the sign extension signals [9],[10], Y_{MSB} is paired with x-signals. The Row_LSB, Neg_cin, and sign extension circuitry are omitted in Fig.2 and fig.3. Fig.4. displays the partial product array description. From figures 2,3,5, it is simple to check that the delay is only equal to two unit delays for any path. The new recoder exhibits the same

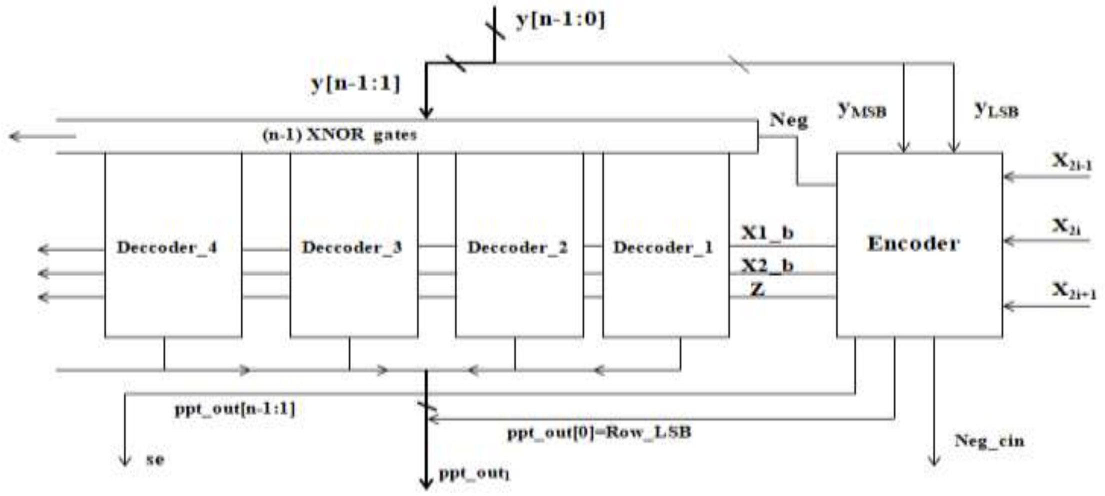


Fig. 5. Structure of MBE used in the Design

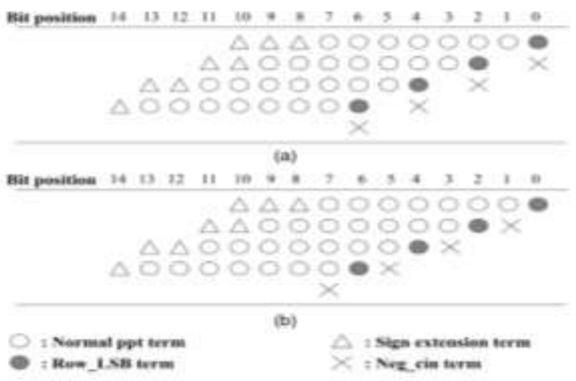

Bit position 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
(a)
Bit position 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
(b)
: Normal ppt term
: Row_LSB term
: Sign extension term
: Neg_cin term

Fig. 4. (a) Traditional MBE partial product array. (b) New MBE partial product array

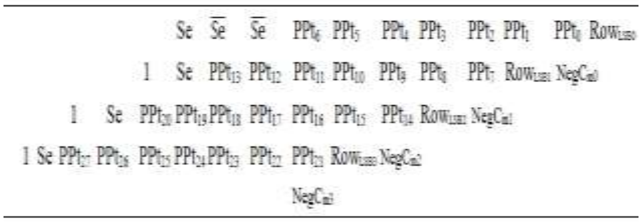
Speed efficiency as MBE III, but the decoder can be implemented with 18 transistors only. Figure 4 (a) represents a partial product array of an 8 x 8 multiplier using conventional MBE scheme. Sign extension circuitry is used. The conventional MBE suffers two drawbacks. At (n-2)th bit position, an additional partial product term and poor LSB-part performance when using the TDM algorithm compared to the non-Booth design. These drawbacks can be overcome by modifying the LSB part of partial product array. From Fig.4(a) using Boolean minimization, the Row LSB and Neg_c_{in} words are combined for further simplification. The equations for Row-LSB, Neg-cin and sign extension can be written as (1), (2) and (3), respectively.

$Row\_LSB_i = y_{LSB}.(x_{2i-1} \oplus x_{2i})$ (1)

$Neg\_c_{in}i = x_{2i-1} . ((x_{2i-1} + x_{2i})'. (x_{2i-1} + y_{LSB})'. (x_{2i}+y_{LSB})')'$ (2)

$Se = (x_{2i+1}+x_{2i}+x_{2i-1})' + (x_{2i+1} . x_{2i} . x_{2i-1}) + (y_{MSB} \oplus x_{2i+1})'$ (3)

x_{2i-1} is zero for the first row. The new partial product array is shown in Fig. 4(b). The partial product term which is additional is now moved to (n-1)th bit position, and the array of LSB-part is now more normal. No overheads are caused by this optimization. Fig 6 shows the final partial product matrix generated from the MBE.



					Se	$\overline{Se}$	$\overline{Se}$	PPt_6	PPt_5	PPt_4	PPt_3	PPt_2	PPt_1	PPt_0	Row_{LSB0}
				1	Se	PPt_{13}	PPt_{12}	PPt_{11}	PPt_{10}	PPt_9	PPt_8	PPt_7	Row_{LSB1}	$NegC_{in0}$	
		1	Se	PPt_{20}	PPt_{19}	PPt_{18}	PPt_{17}	PPt_{16}	PPt_{15}	PPt_{14}	Row_{LSB2}	$NegC_{in1}$			
1	Se	PPt_{27}	PPt_{26}	PPt_{25}	PPt_{24}	PPt_{23}	PPt_{22}	PPt_{21}	Row_{LSB3}	$NegC_{in2}$					
								$NegC_{in3}$							

Fig. 6. Partial Product Matrix of 8X8 multiplication

III. PPRT USING THREE DIMENSIONAL REDUCTION METHOD (TDM)

PPRT stands for partial product reduction tree. TDM PPRT utilizes column circuits (except column 0) for generation of two sum bits for each of the columns. The circuit of each column is implemented with minimum number of full adders required to generate the output bits from the partial product bits for that column and the carry bits (if any) from the preceding column (plus one half adder on the smallest inputs for odd number of inputs). The essential theme of this technique is the sort inputs and outputs which are fast. The interconnection of the elements used in the PPRT correctly is the most crucial step. Appropriate have therefore been defined such that the delay from each input to each output is computed.

An example of this TDM process is illustrated in [4] which depicts a 4:2 compressor with 3 XOR gate delay. It has also been shown that, TDM currently generates an optimal. An instance of the PPRT generation of TDM is shown in [4],[5]. Each column of the partial product matrix is given to a column compressor. The compressor generates sum and carry terms, reducing the partial product arrays to a pair of rows which are further given to MLCSMA for final addition.

IV. MULTIPLE LEVEL CONDITIONAL SUM ADDER (MLCSMA) ALGORITHM

MLCSMA stands for multiple-level conditional-sum adder. It consists of carry sum adder (CSMA) and a conditional-carry adder (CCA). CSMA is selected as the basic building block, since a tree-based CSMA exhibits better performance compared to that of a carry-look ahead adder. In addition, a tree-based CSMA has very regular structure.

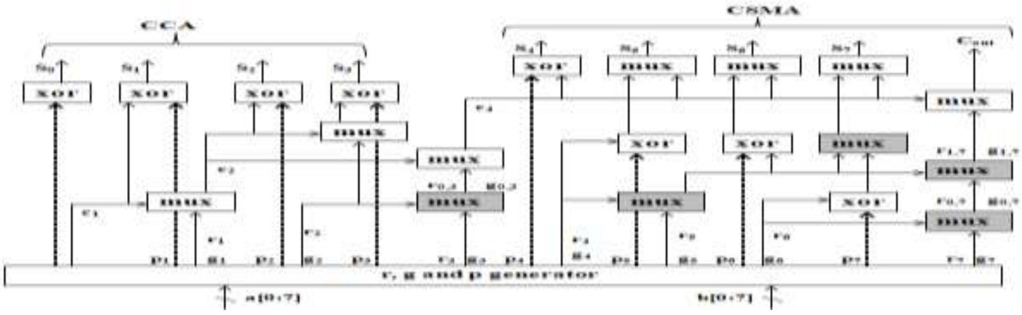


Fig. 7. Hybrid Structure of MLCSMA

In Fig. 7, 8 and 9 the gray blocks are implemented with a pair of gates. For instance, a gray mux block has structure two 2-input multiplexers. In Fig. 7, to conserve area for the first four bits, the CCA structure is used. The speed is retained by using CSMA structure for the last block [11], i.e bits 4 to 7. This structure has the same latency as CSMA and use of CCA saves area [12]. Initially, with the arrival timing data from PPRT, the algorithm receives the outputs and then the inputs are processed from LSB to MSB. Let n denote the index of the bit position and Cn represent the nth bit carry signal. Inorder to create CCA or CSMA, all the r terms, g terms, and p terms are to be initially generated from the inputs. For r, g, and p, the equations are:

$r_{n=}a_n+b_n$ (4)

$g_{n\ =}\ a_n\ .\ b_n$ (5)

$p_{n\ =}\ a_n \oplus b_n$ (6)

To generate and , respectively, the proposed design uses a 2-input NOR gate and a 2-input NAND gate to minimize the delay. By using inverted MUX gates, they are then corrected. The next levels of r and g terms are generated by using the following equations,:

$r_{i,n} = r_{i-1,n} \cdot r_{j,m} + r_{i-1,n}{'} \cdot g_{j,m}$ (7)

$g_{i,n} = g_{i-1,n} \cdot r_{j,m} + g_{i-1,n}{'} \cdot g_{j,m}$ (8)

The first and second indices represent the level for the r and g subscripts and the bit position. The notations $r_{-1,n}$ and $g_{-1,n}$ are identified as r_n and g_n respectively. It can be observed that the previous level specifies the values of j and m. The following equations are needed to generate carry and sum:

$c_{n+1}=c_n \cdot r_n + c_n{'}\ g_n$ (9)

$s_n = c_n \oplus p_n$ (10)

By decomposing (7) and representing the carrying signal recursively by r, g, and ck, where k = 2^i and i = 0, 1, ..., [$\log_2(n)$], we get the same equations which produce carry signals for CSMA or CCA. But, k is not required to be 2^i. Therefore, the selected carry signals and the intermediate r and g terms will produce an arbitrary carry or sum signal. For CSMA, two XOR gates are used to combine Pn with r_{n-1} and g_{n-1} at the beginning of the addition. Similarly, for CCA, one XOR gate is used to combine P_n with C_n to generate the sum at the last step. A typical case, shown in Fig. 8, may be used to demonstrate the MLCSMA algorithm. The C_{26} carrier signal arrives at t = 10 and the a_{26}, a_{27}, b_{26}, and b_{27} inputs hit t = 7.5.

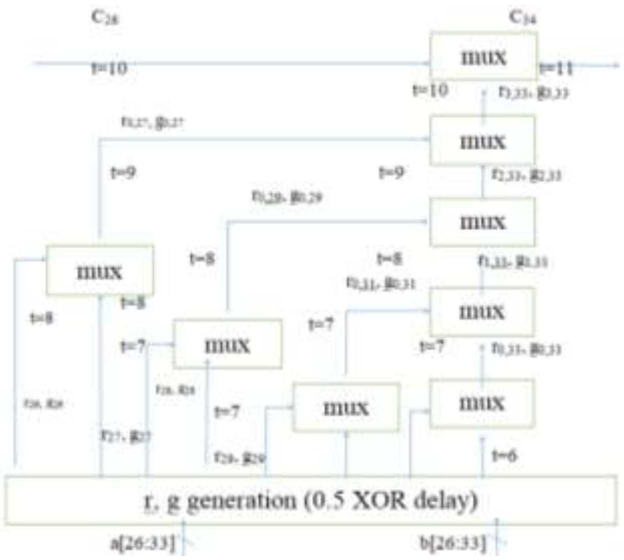


Fig. 8. Carry generation at MSB part

The software realizes a 2-bit CCA segment and generates $r_{0,27}$ and $g_{0,27}$ by using (6) and (7). The timing gap is

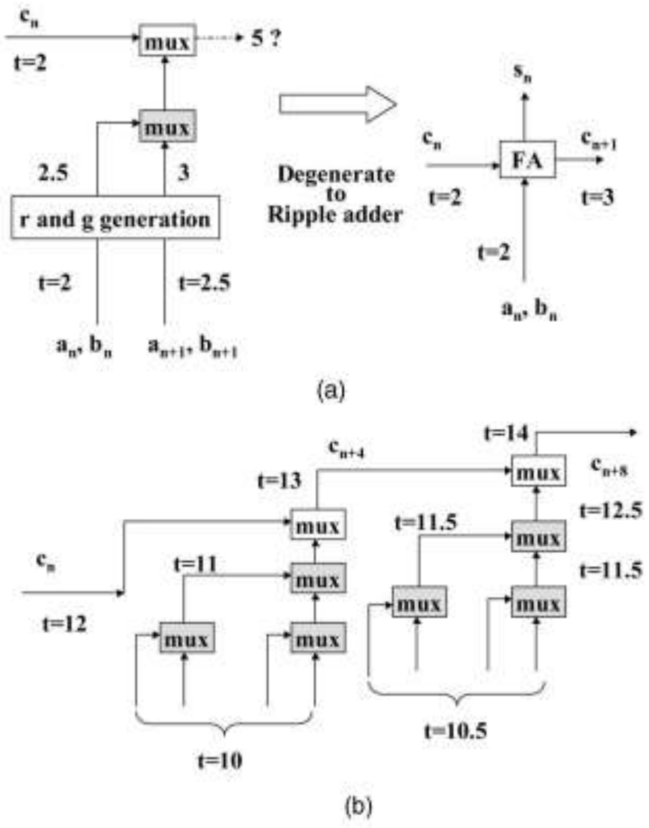


Fig. 9. Examples for LSB part: (a) Ripple carry adder (b) LSB part (two blocks with four inputs)

Sufficiently high to cover another stage of CCA.A 2-bit CCA is constructed for the next successive six bits. $r_{1,33}$ and $g_{1,33}$ are combined by using (6) and (7). c_{33} is produced at t=11 and the sum signals from n=26 to n=33 are generated. The algorithm completes 8-bit block with one delay in a span of four levels.. The LSB part can also be easily handled. In fig. 9(a) a_n and C_n arrive at the same time and an FA is fast enough to generate c_{n+1}. The fig. 9(b) is a typical case where 8-bit addition is completed in two unit delays to complete. The resulting delay is small compared to CSMA as the carry propagation path has multiple levels and less capacitance loading. Most significantly, this algorithm is independent of technology and, in linear time, solves the final adder problem. The algorithm can therefore be used for profile of any delay.

V. SIMULATION

The design is simulated by using Xilinx Vivado software. The code is written using Verilog HDL. The design of new MBE scheme is done by using the decoder and encoder modules. TDM design is done by using half adder and full adder modules. MLCSMA design is simulated using data flow modeling. Multiplier design is simulated by using MBE, TDM and MLCSMA modules.

VI. CONCLUSION

A high speed booth encoded parallel multiplier is designed. The MBE multiplier can perform better than the non booth design by combining proposed new MBE recoder and modified partial product array. For the final addition a new MLCSMA algorithm is described. This algorithm solves the final adder problem and improves its performance. The simulation and synthesis are done using Xilinx Vivado. Verilog HDL language is used to design the modules of the project. Synthesis is done by using FPGA. Spartan 6 FPGA is used in this project. By considering the timing and area analysis of several MBE schemes, it is feasible to use the new MBE scheme. While using different logic styles the recoders must be considered carefully. The MLCSMA algorithm can be implemented to other design profiles with different delay. If the output of a multiple constant multiplication comes out with different delays the the MLCSMA algorithm can be used.

REFERENCES

[1] A.D. Booth, ªA Signed Binary Multiplication Technique,º Quarterly J. Mechanical and Applied Math., vol. 4, pp. 236-240, 1951.

[2] O.L. MacSorley, ªHigh Speed Arithmetic in Binary Computers,º Proc. IRE, vol. 49, pp. 67-91, 1961.
[3] Wen-Chang Yeh and Chein-Wei Jen, "High-speed Booth encoded parallel multiplier design," in IEEE Transactions on Computers, vol. 49, no. 7, pp. 692-701, July 2000, doi: 10.1109/12.863039.
[4] V.G. Oklobdzija, D. Villeger, and S.S. Liu, ªA Method for Speed Optimized Partial Product Reduction and Generation of Fast Parallel Multipliers Using an Algorithmic Approach,º IEEE Trans. Computers, vol. 45, no. 3, pp. 294-306, Mar. 1996.
[5] P.F. Stelling, C.U. Martel, V.G. Oklobdzija, and R. Ravi, ªOptimal Circuits for Parallel Multipliers,º IEEE Trans. Computers, vol. 47, no. 3, pp. 273-285, Mar. 1998.
[6] G. Goto et al., ªA 4.1ns compact 54 54-b Multiplier Utilizing Sign-Select Booth Encoders,º IEEE J. Solid-State Circuits, vol. 32, no. 11, pp. 1,676-1,682, Nov. 1997. 700 IEEE TRANSACTIONS ON COMPUTERS, VOL. 49, NO. 7, JULY 2000 Fig. 12. Gate-level simulation and the estimated delay from normalized delay model
[7] G. Goto et al., ªA 54 54-b Regularly Structured Tree Multiplier,º IEEE J. Solid-State Circuits, vol. 27, no. 9, Sept. 1992.
[8] R. Fried, ªMinimizing Energy Dissipation in High-Speed Multipliers,º Proc. 1997 Int'l Symp. Low Power Electronics and Design, pp. 214-219, 1997.
[9] J. Fadavi-Ardekani, ªMN Booth Encoded Multiplier Generator Using Optimized Wallace Trees,º IEEE Trans. VLSI Systems, vol. 1, no. 2, June 1993.
[10] A.A. Farooqui et al., ªGeneral Data-Path Organization of a MAC Unit for VLSI Implementation of DSP Processors,º Proc. 1998 IEEE Int'l Symp. Circuits and Systems, vol. 2, pp. 260-263, 1998.
[11] K. Hwang, Computer Arithmetic: Principles, Architecture, and Design, chapter 3, p. 81. John Wiley & Sons, 1976.
[12] K.H. Cheng et al., ªThe Improvement of Conditional Sum Adder for Low Power Applications,º Proc. 11th Ann. IEEE Int'l ASIC Conf., pp. 131-134, 1998.
[13] Wikipedia contributors. "Binary multiplier." Wikipedia, The Free Encyclopedia. Wikipedia, the Free Encyclopedia, 2 Dec. 2020. Web. 4 Jan. 2021.
[14] DeekshaKiranD, K et al. "VLSI Implementation of Braun Multiplier using Full Adder." 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC) (2017): 499-504.
[15] Oliveira, Leonardo & Santos, Cristiano & Ferrão, Daniel & Costa, Eduardo & Monteiro, José & Martins, Joao & Bampi, Sergio & Reis, Ricardo. (2007). A Comparison of Layout Implementations of Pipelined and Non-Pipelined Signed Radix-4 Array Multiplier and Modified Booth Multiplier Architectures. 10.1007/978-0-387-73661-7_3.
[16] Indrayani Patle, Akanksha Bhargav, Prashant Wanjari, "Implementation of Baugh-Wooley Multiplier Based on Soft-Core Processor", IOSR Journal of Engineering, Vol. 3, Issue 10, Oct. (2013).Based on Soft-Core Processor", IOSR Journal of Engineering, Vol. 3, Issue 10, Oct. (2013).
[17] Shanthala S, S. Y. Kulkarni "VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique" European Journal of Scientific Research
[18] Wen-Chang Yeh and Chein-Wei Jen, "High-speed Booth encoded parallel multiplier design," in IEEE Transactions on Computers, vol. 49, no. 7, pp. 692-701, July 2000, doi: 10.1109/12.863039.
[19] "FPGA Implementation of Booth's and Baugh-Wooley Multiplier Using Verilog", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-3, Issue-1, June 2013.
[20] Digital Multipliers: A Review Volume-4, Issue-3, June-2014, ISSN No.: 2250-0758 International Journal of Engineering and Management Research