5th International Conference on Recent Development in Engineering Science, Humanities and Management
National Institute of Technical Teachers Training & Research, Chandigarh, India (ESHM-17)
16th April 2017, www.conferenceworld.in ISBN: 978-93-86171-39-9

DESIGN AND IMPLEMENTATION OF HIGH SPEED MODIFIED BOOTH ENCODED WALLACE TREE MULTIPLIER

Divya Govekar[1], Ameeta Amonkar[2]

[1]Student, [2]Proessor,Department of Electronics and Telecommunication,
Goa College of Engineering Farmagudi, Goa (India)

ABSTRACT

Multiplier is one of the most desirable components in DSP processors, Fast Fourier Transform Units and Arithmetic Logic Units. Since long, VLSI engineers are working towards speed optimization of a multiplier to ultimately improve the performance of processing units. These units play a major role in devices such as mobile, laptops etc. This paper aims to design and implement several parallel multiplication approaches and their comparative analysis based on the parameters such as speed and area. These approaches increase speed by reducing the partial products generated as well as Wallace tree algorithm is used to increase the speed of the multiplier. In this paper, the speed of a multiplier is measured in terms of delay obtained and the area in terms of number of LUTs utilized. All the multiplier designs have been synthesized using Xilinx ISE 10.1 design tool. The programming language used is Verilog HDL.

Keywords:Booth Algorithm, Booth Encoder, Parallel Multiplication Scheme, Partial Product Generator, Wallace Tree structure.

I. INTRODUCTION

Based on the theory of multiplication there are many processes that happen in the backend which earlier was done by us manually in order to get the result. Since there are many processes which cause many phases and since it is automated in a computing system, it ended up with huge hardware resources as well. In order to design an efficient multiplier architecture, the major area of improvement considered by the researchers was the Operating Speed, Area, and the Power Consumption[1]-[2]. Amongst the three factors, most of the research work was done on improving the speed of the multiplier, in order to enhance the utility of processor. Conventional multiplication is carried out in two steps that are, partial product generation and partial product addition. So in order to increase the speed of multiplier, we can reduce the number of partial products generated and we can also parallelize the addition of partial products generated. Booth multiplier algorithm can be used both to produce as well as reduce the number of the partial products generated. Compared to the conventional multiplication algorithm, this algorithm reduces the number of additions required to produce the result, where two bits of the multiplier are scanned at a time to reduce the partial product production and added together[2] .There is no need to take the sign of the number into consideration, with unsigned multiplication. In signed

ISBN: 978-93-86171-39-9

multiplication, the same process cannot be applied because the signed number is in a 2'complement form which would yield an incorrect result if multiplied in a similar fashion to unsigned multiplication. Booth's algorithm preserves the sign of the result. Most of the communication systems and multimedia are now using Digital Signal Processors (DSP), which is using booth multiplier to improve the speed. Other applications of booth multiplier are Fast Fourier Transform processors and high-speed floating point arithmetic logic units.

II. RELATED RESEARCH

In general, the multiplier consists of three parts primarily: A tree to compact the partial products, Booth encoder and the final adder. A Wallace tree is simply a logical function which is used for the addition of the partial products. The processing speed of a multiplier can be increased by reducing the amount of the partial products. To achieve the above intended aspect, Booth's algorithm is hired mostly where Wallace tree improves the speed with which the partial products are added[3]using Carry Save Adders .Booth encoder multiplier with Carry select Adder utilizes the minimum hardware, reduced chip area, low power dissipation and reduced the cost of the system.The approximate scheme has been proposed to achieves significant improvements in power consumption, delay and combined metrics [4]. In [6], the truncation error is appreciably reduced since the error compensation is generated in which, the compensation is adaptively modified according to the input data. By Fixed width multiplier there is reduction in the truncation error, appropriate compensation biases can be generated and added to the retained adder cells, all the while striving to reduce the number of adder cells used [5].The error values of proposed architecture [5] are less compared to the existing architecture [6].The proposed method is pretty close to the rounding operation and much better than the fixed-width modified Booth multiplier design method in [6]. Low error is necessary for accurate output and speed. Also complexity and power can be reduced.

In this paper, a booth encoding 8-bit Multiplier is implemented. Booth encoding is an effective method which greatly increases the speed of our algebra. Attempt is made to reduce the number of partial products generated in a multiplication process by using the modified booth algorithm .The multiplier takes in two 8 bit operands the multiplier and the multiplicand , then produces 16-bit multiplication result of the two as its output. The architecture comprises four parts: complement generator, booth encoder, partial product .This structure is then compared with other multipliers.

III. Multiplier Structure

At the most basic level, a binary multiplier is an electronic hardware device used in digital electronics or a computer or other electronic device to perform multiplication of two numbers in binary representation. Consider the multiplication of two numbers: the multiplier X, and multiplicand Y,where X is an n-bit number with bit representation {xn-1,xn-2,...,x0}, the most significant bit being xn-1 and the least significant bit being x0; Y has a similar bit representation {yn-1,yn-2,...,y0}. For unsigned multiplication, always n shifted copies of the multiplicand are added to form the result. The entire procedure is divided into three steps: partial product (PP) generation, partial product reduction, and final addition.

ISBN: 978-93-86171-39-9

3.1. Array Multiplier

The array multiplier has a regular structure. Most of the area of the multiplier is devoted to the adding of partial products. The initial step in multiplication is to generate n shifted copies of the multiplicand. Whether a given shifted copy of the multiplicand is added depends on the value of the multiplier bit corresponding to this multiplicand copy. The bit representation of this function can be implemented using a logical AND gate, which performs AND(ai,bj), i = 0 to n-1, j = 0 to n-1. The resulting values are called partial products; if we line up these partial products by bit position (bp = i + j). The disadvantage of array multiplier is large critical path delay and hence are slow.

3.2. Carry Save Array Multiplier

In conventional array structure addition was done by conventional carry adders, the delay of all the adders would consume a large amount of time, as each shifted version of the multiplicand would contribute a delay which is proportional to the width of the multiplicand. In this multiplier the delay is reduced using technique called carry save addition. In carry save we use full adders to add the three bits in each column, while adding three bit vectors of array at a time.The outputs of this first set of full adders can now serve as inputs to another row of full adders, along with another bit vector in array multiplier structure. The delay of this multiplier block is a function of the number of rows, *O(n)*, which is a big improvement over the scheme of using conventional adders for each row.

3.3. Wallace Tree Multiplier

C.S. Wallace in 1964, observed that the later stages of the array structure must always wait for all the earlier stages to complete before their final values will be established. It is possible to create a structure which has less delay by performing the addition operations in parallel when performing a series of independent add operations. Parallelizing the carry-save operations will yield a delay which is much shorter than the array's sequential series of operation The number of stages, and hence the delay of the sequential operation will be O(n), which will be further reduced to O(log3/2 n) in the parallel method.

3.4 Booth Multiplier

The Booth algorithm was invented by A. D. Booth, which forms the base of Signed number multiplication algorithms, and has the potential to speed up signed multiplication. Booth algorithm is used for partial product generation as well as reduction stage. It reduces the number of partial products formed during the generation stage by scanning two bits at a time.Hence booth multiplier increases the speed by reducing the number of the partial product.

3.4. Modified Booth Multiplier

Similar to the radix2 booth multiplier this multiplier is also used during the partial product generation and reduction stage. Since it reduces the number of partial products by half, by using the technique of radix 4 booth recoding. The basic idea is that, instead of shifting and adding for every column of the multiplier term and multiplying by 1 or 0, we only take every second column, and multiply by ±1, ±2.or 0, to obtain the same result. Booth Encoder will compare 3 bits at a time with overlapping technique.Grouping starts from the LSB and the first block only uses two bit of the multiplier and assumes a zero for the third bit as shown in fig.

ISBN: 978-93-86171-39-9

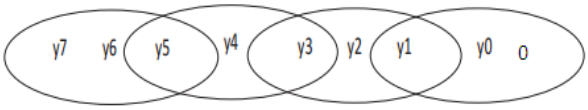


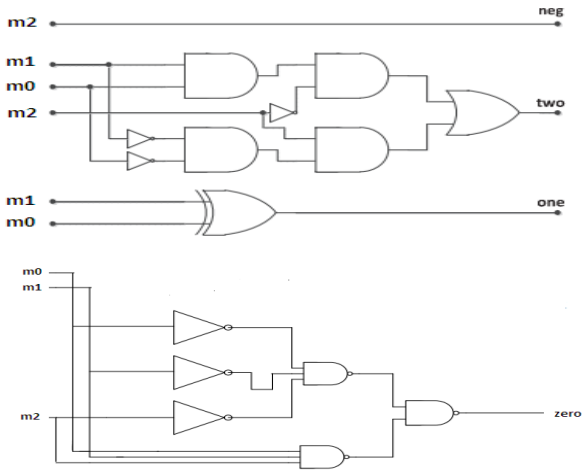
fig 1. Grouping of bits in the multiplier for encoding the partial products[8]

VI. ARCHITECTURE OF MODIFIED BOOTH ENCODED WALLACE TREE MULTIPLIER

The Modified Booth Encoded Wallace Tree Architecture is divided into four modules and each module has been explained in the following subsections.

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PP_0						$s_0^{`}$	s_0	s_0	$p_{0,7}$	$p_{0,6}$	$p_{0,5}$	$p_{0,4}$	$p_{0,3}$	$p_{0,2}$	$p_{0,1}$	$p_{0,0}$
PP_1					1	$s_1^{`}$	$p_{1,7}$	$p_{1,6}$	$p_{1,5}$	$p_{1,4}$	$p_{1,3}$	$p_{0,2}$	$p_{1,1}$	$p_{1,0}$		
PP_2			1	$s_2^{`}$	$p_{2,7}$	$p_{2,6}$	$p_{2,5}$	$p_{2,4}$	$p_{2,3}$	$p_{2,2}$	$p_{2,1}$	$p_{2,0}$				
PP_3	1	$s_3^{`}$	$p_{3,7}$	$p_{3,6}$	$p_{3,5}$	$p_{3,4}$	$p_{3,3}$	$p_{3,2}$	$p_{3,1}$	$p_{3,0}$						
	P_15	P_14	P_13	P_12	P_11	P_10	P_9	P_8	P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0

Fig 2. 8*8 Partial Product matrix



Fg 3. Booth Encoder

4.1. Two's Complement

It is one of the basic requirements for signed multiplication and it is obtained by simply complementing the bits of Multiplicand and adding one to it. The booth decoder requires this to differentiate the data required between the positive and negative multiplicand to be added.

4.2. Booth Encode

The main purpose of using booth encoder is to decrease the number of partial products as compared to basic booth multiplier [6]. With the use of Booth encoder one can multiply not only positive but also negative numbers and at the same time reduce the number of partial products[4]. By using booth encoder whose encoding values are given in Table 1 the multiplier ' X ' can be represented by considering three bits (triplets) at a time.

ISBN: 978-93-86171-39-9

This triplet is formed by considering three bits of multiplier X. The operation –X can be realized by the inversion of the multiplicand and adding 1 to the LSB. The operations 2X and -2X can be realized by shifting X and –X by one bit to left and filling the LSB with zero. In this paper for further discussion an N x N multiplication with N=8 is considered. While computing all the value the resultant partial productmatrix for 8*8 modified booth multiplications is given in fig 2

A simple gate equivalent circuit of booth encoded table is realized in Table 1. The three bits of B (multiplier) are selected at once. Neg bit indicates whether the multiplicand is multiplied with negative weight or not. Zero indicates whether to be set as partial product. one can be interpreted as multiplicand that is used directly and finallytwo tells whether the left shift operation is to be done on the multiplicand or not. The booth encoder diagram is in fig 3.

Table 1. Booth Encoded Values

M1(x2i)	M0(x2 i-1)	Fn	neg	two	one	zero
0	0	+0	0	0	0	1
0	1	+X	0	0	1	0
1	0	+X	0	0	1	0
1	1	+2X	0	1	0	0
0	0	-2X	1	1	0	0
0	1	-X	1	0	1	0
1	0	-X	1	0	1	0
1	1	-0	1	0	0	1

4.3. Partial Product Generator

The use of Modified Booth Encoder has led to a decrease in the number of partial products. The Fig 4 shows a basic multiplexer circuit which generates the partial product PP.The multiplexer unit which has two inputs the multiplicand A and its complement Abar. The neg signal generated by the Modified booth encoder circuit decides to pass either input or it's 2'complement.The other input select line is a combination of {two, one, zero}. If two is high then it will pass the last state of the input else if both two and neg is high then input is 2'complimented last state of input , if one is high then output is A or else its ABAR .

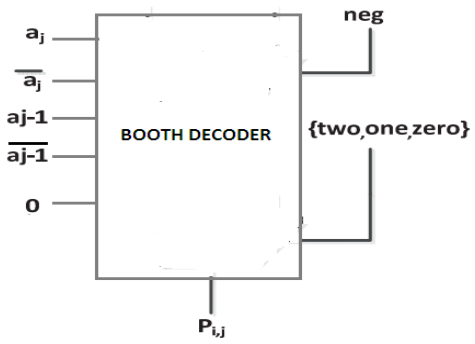


Fig 4. Booth Decoder

ISBN: 978-93-86171-39-9

If zero is set high then it sets the output to '0'. Now for every combination of the triplet formed through Fig 2 a partial product PP is formed.

4.1 Wallace Tree Structure

A Wallace tree architecture implements a faster addition as the data bits need not wait for the previous data to appear at the input. The final stage is a simple carry adder which will add all the carry propagated into a final value. The Wallace tree architecture when used in conjunction with booth encoder leads to high speed computation and yields higher efficiency [8].

V. EXPERIMENTAL RESULTS

A Modified Booth Wallace Tree multiplier is implemented and simulated in ModelSim5.7g.The simulation result is show in Fig 5 . The RTL compiler of Xilinx ISE 10.1 has been successfully created synthesized top level module modified booth this is shown in Fig 6.

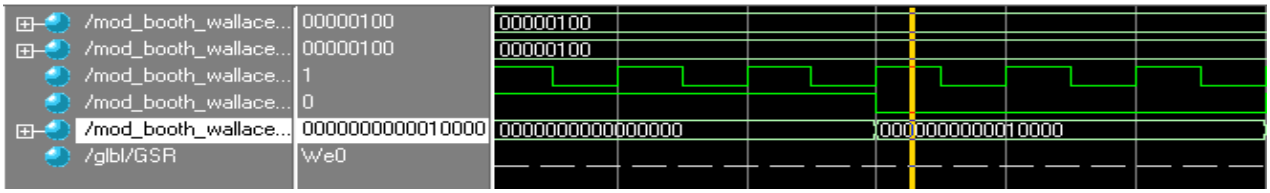


fig 5 .Simulation Results of Modified Booth Wallace Multiplier.

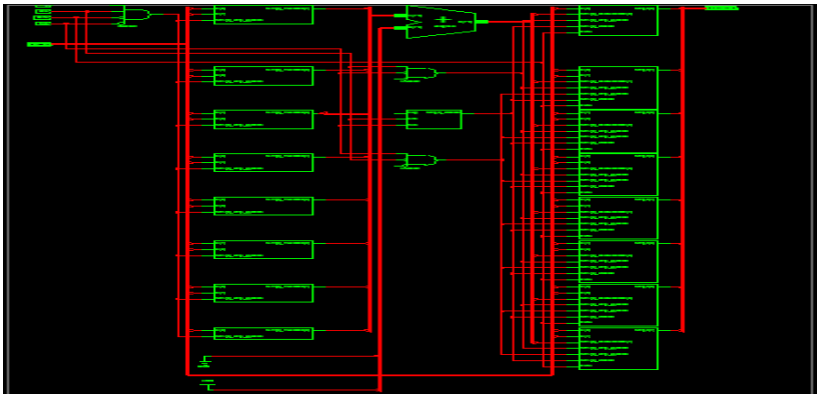


fig 6. Synthesized Top Level Module

Table 2 shows comparison results between Modified Booth Encoded Wallace Tree multiplier and other multipliers. It shows that the number of slices and LUT's are reduced in Modified Booth Encoded Wallace Tree over other multipliers like Wallace and Carry Save Array Multiplier. Array multiplier uses less number of LUTs but lacks in terms of speed compared to Modified Booth Encoded Wallace multiplier. Table below also states that the speed of the Modified Booth Encoded Wallace multiplier is increased compared with the other multiplier.

ISBN: 978-93-86171-39-9

Table 2. Comparison of Multipliers in terms of Delay and Area

Multiplier Design	Number of LUTs	Number of Bonded IOBS	Number of bits	Delay(ns)
Array	94 out of 28800	34 out of 440	8	10.825ns
Carry Save Array	106 out of 28800	34 out of 440	8	9.592 ns
Wallace Tree	199 out of 2880	34 out of 440	8	7.515ns
Modified Booth Encoded Wallace Tree	105 out of 2880	34 out of 440	8	6.5456ns

VI. CONCLUSION

In this paper 8 bit Modified Booth Encoded Wallace Tree multiplier is implemented .Its performance is compared with Array and Wallace tree multiplier designs which are implemented for the purpose of comparison. It is found that Modified Booth Wallace Tree multiplier has the highest speed of operation and least delay. In Modified Booth Encoded Wallace Tree multiplier the delay is reduced by using booth encoding scheme and fast Wallace tree addition.

REFERENCES

[1] Naga Mani Mendu ,A Simple Method to Improve the throughput of A Multiplier, International Journal of Electronics and Communication Engineering pp. 9-16,2013.

[2] . Wen-Chang Yeh, Chein-Wei Jen High-speed Booth encoded parallel multiplier design, IEEE Transactions on Computers pp(74) ,2000.

[3] R. Balakumaran , E. Prabhu Design of high speed multiplier using modified booth algorithm with hybrid carry look-ahead adder, IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT) pp:1-7, 2016.

[4] Liangyu Qian, Chenghua Wang,Weiqiang Liu, Fabrizio Lombardi, Jie Han"Design and evaluation of an approximate Wallace-Booth multiplier "IEEE International Symposium on Circuits and Systems (ISCAS) ,pp.1974-1977,2016.

[5] Saroja S. Bhusare; V. S. Kanchana Bhaskaran Design of a Low Error Fixed-Width Radix-8 Booth Multiplier IEEE Fifth International Conference on Signal and Image Processing pp: 206 - 209, 2014.

[6] K.-J. Cho, K.-C.Lee, J.-G. Chung, and K. K.Parhi,,Design of low error fixed-width modified Booth multiplier, IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst., Vol. 12, No. 5,pp. 522-531, May 2004.

[7] Pascal Constantin Hans Meier,Analysis and Design of Low Power Digital Multipliers, Pittsburgh, Pennsylvania August, 1999.

[8] Rahul D Kshirsagar; E. V. Aishwarya; Ahire Shashank Vishwanath; P. Jayakrishnan,Implementation of pipelined Booth Encoded Wallace tree Multiplier architecture ,IEEE, 2013International Conference on Green Computing, Communication and Conservation of Energy (ICGCE) Pages: 199 – 204.