Impact Factor (SJIF): 5.301

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 5, Issue 3, March-2018

AREA EFFICIENT LOW POWER MULTIPLIER FOR FFT DIF ALGORITHM

[1]Sankar K, [2]Kavitha K

[1]PG Scholar,[2]Assistant Professor(Sr.Gr),Department of Electronics and Communication Engineering, Velalar College of Engineering and Technology, Erode-12.

Mail Id: [1]*sankarvlsi94@gmail.com*, [2]*kkavithamoorthy75@gmail.com*

ABSTRACT

In many multimedia and Digital Signal Processing(DSP) systems, Modified Booth Multiplier is mostly preferred multiplier design for the high speed application. In this paper, Modified Booth Algorithm (MBA) is used for both signed and unsigned multiplication process which is used for the FFT application in frequency domain whose design parameter are analyzed along with the design of FFT with Baugh Wooley multiplier. Wallace tree reduction technique is used to reduce the number of partial product which reduces the hardware complexity with the low power consumption in Modified Booth Multiplier.

Keyword: FFT,Modified Booth Multiplier, Baugh Wooley multiplier

I. INTRODUCTION

In digital signal processing systems, the most important objectives are high processing performance and low power dissipation. Among the basic operational block in a digital system, multiplier requires the longest delay. In general, the critical path is determined by the multiplier. Modified Booth Algorithm (MBA) is commonly used for high speed multiplication. The speed of the multiplication can be increased by reducing the number of partial products [6]. Various multiplication algorithm Booth, Braun and Baugh Woolley have been used.

The modified booth algorithm is considered as the fastest multiplication algorithm and reduces the number of partial products. Carry propagate adder has been used to sum the partial product in reduced time. Our goal is to reduce computation frequency by using a Modified Booth algorithm for multiplication. Ripple carry adder (RCAs) has the most compact design of all types of adders. The first stage is considered as a partial product generation. The second stage is considered as a partial product reduction stage was high-speed algorithms are used to reduce the number of partial products [5]. The final addition stage of the multiplier is Carry propagate Adder (CPA).

Modified booth encoding is used to achieve high speed which reduces the number of partial products. Due to truncation error will be introduced in the kind of Direct-Truncated Fixed-width Multiplier (DTFM). The error compensation value can be produced by the constant scheme [1] -[2] or the adaptive scheme [3] -[4]. Fixed width modified booth multiplier achieves better performance in terms of error [7]. In wireless communication applications, the most commonly used operation is Fast Fourier Transform (FFT).

Vedic Mathematics [8] and [9], Booth, Modified Booth, Barun, Array multipliers are designed to achieve the high-speed FFT application. Where Booth multiplication is used to reduce the partial products at the intermediate stage, which achieves the high-speed multiplication rather than other multipliers. In [10] for FFT application modified booth along with the Wallace tree structure which is build using Carry Save Adder (CSA).

In this paper, high-speed FFT architecture is designed based on two different multiplication algorithm such as Modified Booth Algorithm and Baugh Woolley multiplication algorithm were Modified Booth multiplication algorithm achieves better performance than Baugh Woolley multiplication algorithm.

II. FAST FOURIER TRANSFORM (FFT)

Fast Fourier Transform (FFT) as a method of efficiently evaluating a polynomial at the powers of a primitive root of unity. In other words, we will define the FFT as "a technique which efficiently evaluates a polynomial over a special collection of points" and the inverse FFT will be defined as "a technique which efficiently interpolates a collection of evaluations of some polynomial at a special set of points back into this polynomial". Fourier analysis converts a signal from time or space domain to the frequency domain and vice versa. An FFT rapidly computes such transformations by factorizing the DFT matrix into a product of sparse (mostly zero) factors. As a result, it manages to reduce the complexity of computing the DFT from (O (n^2)), which arises if one simply applies the definition of DFT, to (O (n log n)), where n is the data size.

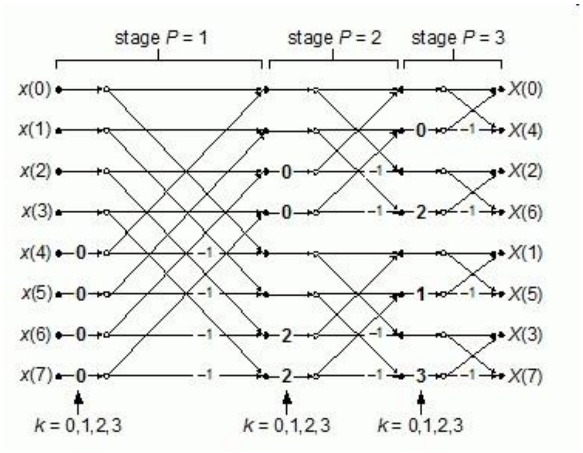


Fig. 1: Butterfly diagram of 8-point DIF-FFT

III. PROPOSED ARCHITECTURE

The proposed multiplier architecture consist of three stages they are,

1. Partial Product Generation
2. Partial Product Reduction
3. Final Addition

Consider 'a' and 'b' be the two 8-bit inputs for the multiplier, where 'a' represents the multiplicant bit and 'b' represents the multiplier bit. The multiplicant bit is encoded by using the booth encoder which reduces the bit count from 8-bit to 3-bit resulting in 4 different combinations. These 4 different combinations acts the selection line for the multiplexer.

The partial products are generated by using 8:1 8-bit multiplexer along with the multiplication algorithms like Modified Booth Algorithm (MBA) and Baugh Woolley multiplication algorithm. Then further the accumulated partial products are reduced using the Wallace tree structure. Finally, Carry Propagate Adder is used to produce the required output. The architecture of MBA used for the FFT application is shown in figure 2.

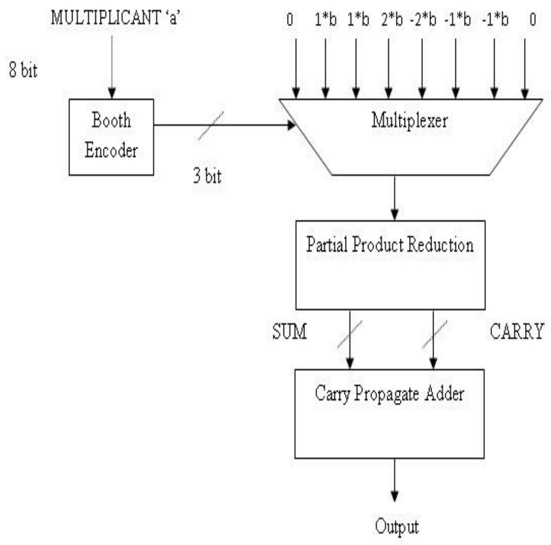


Fig. 2: Block diagram of Modified Booth Muliplier

1. Partial Product Generation

In the generation of partial products, Modified Booth Algorithm is used which consist of selection table to generate the multiplied output. This modified booth multiplier's computation time and the logarithm of the word length of operands are proportional to each other. We can reduce half the number of partial product. Radix-4 booth algorithm used here to increase the speed of multiplier and reduces the area of multiplier circuit.

In this algorithm, every second column is taken and multiplied by 0 or +1 or +2 or -1 or -2 instead of multiplying by 0 or 1 after shifting and adding of every column of the booth multiplier which is explained in Table. I. Thus, half of the partial product can be reduced using this booth algorithm. The multiplexer is used for the selection of partial products which is generated by the modified booth multiplication algorithm.

The outcome of the encoder is taken as the 3-bit select line for the multiplexer. The input of the multiplexer is the 8-bit data of count 8. Among the 8, one is used for the generation of output which is based on the select line. Thus the four different combination outcome of encoder produce the four rows of partial products which all are obtained from the previously used 8:1 multiplexer. The multiplier bit is associated with the following set of numbers to produce the output, which is explained in the Table. I.

TABLE I

SELECTION TABLE OF MODIFIED BOOTH MULTIPLIER

i+1	I	i-1	ADD
0	0	0	0*M
0	0	1	1*M
0	1	0	1*M
0	1	1	2*M
1	0	0	-2*M
1	0	1	-1*M
1	1	0	-1*M
1	1	1	0*M

2. Partial Product Reduction

Here the reduction process is carried out by using the Wallace tree structure which consist of a tree of Carry Save Adder (CSA). In order to diminish the number of PPs involved and therefore lessen the area/delay of the circuit, one operand is usually recoded into high-radix digit sets. One of the most used and widespread radix-2n algorithm is the radix-4 which has a set of digits given by {-2,-1, 0, 1, 2} for PPG.For PPR, two choices exist which can be implemented: reduction by rows, which can be performed by taking into consideration an adder array and reduction by columns, which can be performed by taking into consideration a counter array.

The adder used in the Wallace structure generates the result as the sum and carry. CSA is the most preferable adder for the partial product reduction process which consumes less time to compute the result.

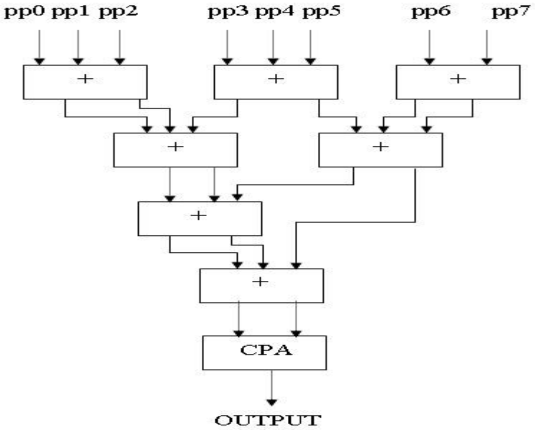


Fig. 3: Wallace Tree Structure.

3. Carry Propogate Adder

The final addition is performed by using the carry propogate adder instead of Carry Propagate Adder(CPA) to obtain the exact result. N-bit adder is designed using N 1-bit full-adder and carry bit is propagated through the proceeding block to produce the required product output.

IV. BAUGH WOOLLEY MULTIPLIER

An efficient way to analyze the sign bits, Baugh Woolley multiplication algorithm is used. Regular multipliers are designed by using Baugh Wooley multiplication technique which suited for the 2's complement numbers whose architecture is shown in figure . Let us consider two N-bit numbers INPUT 1(A) and INPUT 2(B), where A represents the multiplier bit, B represents the multiplicant bit and they are represented as,

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

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

Where a_i's and b_i's are the bits in A and B respectively, a_{n-1} and b_{n-1} are the sign bits.

The product P = A*B is given by, the following equation,

P = A*B

$$P=[-a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i]\mathrm{x}$$

$$[-b_{n-1}2^{n-1} + \sum_{i=0}^{n-2} b_i 2^i]$$

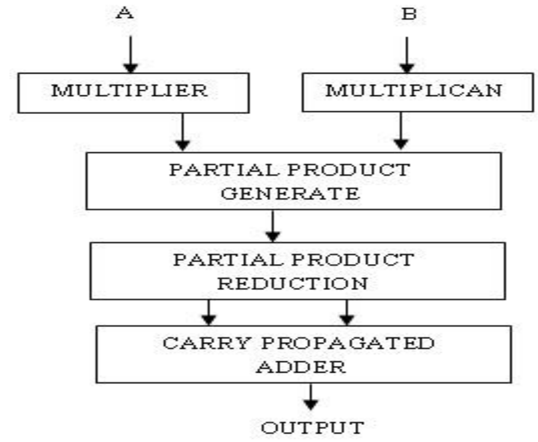
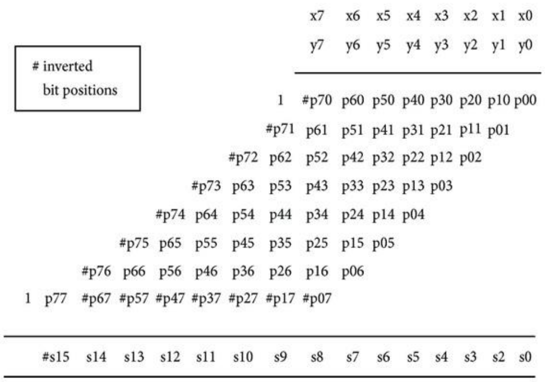


Fig. 4: Architecture of Baugh Woolley Multiplier.

The two inputs A and B are fed as the input to the Partial Product Generation (PPG) unit. In the PPG unit the partial products are generated and accumulated. The accumulated partial products are reduced by the Wallace Tree reduction technique to achieve the better hardware as well as the timing constraints. The reduction process is carried out through three stages. In each stage, the number of rows are being reduced. The reduced partial products are passed through the carry propagate adder to generate the final product. The figure shows the accumulation of partial products.



```
                                                x7   x6   x5   x4   x3   x2   x1   x0
                                                y7   y6   y5   y4   y3   y2   y1   y0
 # inverted                                    ----------------------------------------
   bit positions                          1   #p70  p60  p50  p40  p30  p20  p10  p00
                                        #p71  p61  p51  p41  p31  p21  p11  p01
                                  #p72  p62  p52  p42  p32  p22  p12  p02
                            #p73  p63  p53  p43  p33  p23  p13  p03
                      #p74  p64  p54  p44  p34  p24  p14  p04
                #p75  p65  p55  p45  p35  p25  p15  p05
          #p76  p66  p56  p46  p36  p26  p16  p06
 1   p77  #p67 #p57 #p47 #p37 #p27 #p17 #p07
---------------------------------------------------------------------------------------
    #s15  s14  s13  s12  s11  s10  s9   s8   s7   s6   s5   s4   s3   s2   s0
---------------------------------------------------------------------------------------
```

Fig. 5: Partial Product Accumulation.

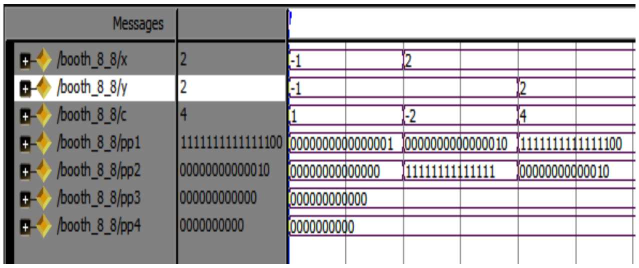
V. RESULT AND DISCUSSION

The modified booth multiplier has been designed and verified by the simulation and it is analyzed using a Xilinx Spartan 3E kit. The following figure 6 and 7 shows the simulation waveform of modified booth multiplier and baugh woolley multiplier. The comparison table of area and power of the multipliers are shown below.

TABLE II

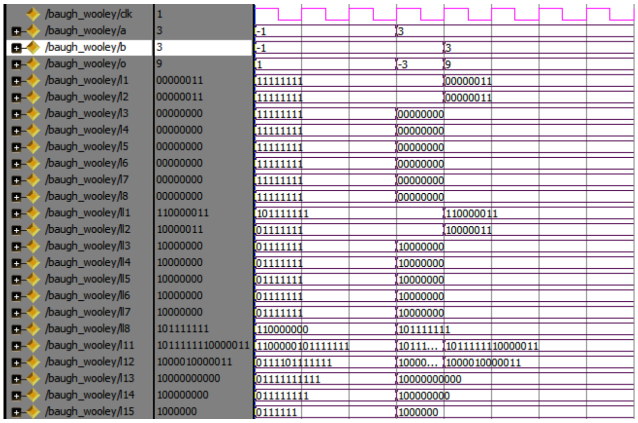
COMPARISON OF DESIGN PARAMETERS FOR FFT WITH MODIFIED BOOTH AND FFT WITH BAUGH WOOLEY MULTIPLIER

EXACT MULTIPLIER	AREA(m^2)	POWER(mw)
MODIFIED BOOTH MULTIPLIER	1533	40
BAUGH WOOLLEY	1845	46.94
FFT MULTIPLIER		
MODIFIED BOOTH MULTIPLIER	5912	62.96
BAUGH WOOLLEY	8532	80.65



```
Messages
/booth_8_8/x     2                 -1                 2
/booth_8_8/y     2                 -1                                    2
/booth_8_8/c     4                 1                  -2                 4
/booth_8_8/pp1   1111111111111100  0000000000000001   0000000000000010   1111111111111100
/booth_8_8/pp2   00000000000010    00000000000000     11111111111111     00000000000010
/booth_8_8/pp3   000000000000      000000000000
/booth_8_8/pp4   0000000000        0000000000
```

Fig.6: Simulation Result of Modified Booth Multiplier



```
/baugh_wooley/clk  1
/baugh_wooley/a    3                 -1                  3
/baugh_wooley/b    3                 -1                              3
/baugh_wooley/o    9                 1                   -3          9
/baugh_wooley/l1   00000011          11111111                        00000011
/baugh_wooley/l2   00000011          11111111                        00000011
/baugh_wooley/l3   00000000          11111111            00000000
/baugh_wooley/l4   00000000          11111111            00000000
/baugh_wooley/l5   00000000          11111111            00000000
/baugh_wooley/l6   00000000          11111111            00000000
/baugh_wooley/l7   00000000          11111111            00000000
/baugh_wooley/l8   00000000          11111111            00000000
/baugh_wooley/ll1  110000011         101111111                       110000011
/baugh_wooley/ll2  10000011          01111111                        10000011
/baugh_wooley/ll3  10000000          01111111            10000000
/baugh_wooley/ll4  10000000          01111111            10000000
/baugh_wooley/ll5  10000000          01111111            10000000
/baugh_wooley/ll6  10000000          01111111            10000000
/baugh_wooley/ll7  10000000          01111111            10000000
/baugh_wooley/ll8  101111111         110000000           101111111
/baugh_wooley/l11  101111111000001 1 1100000101111111    10111...    101111111000001 1
/baugh_wooley/l12  1000010000011     0111101111111       10000...    1000010000011
/baugh_wooley/l13  10000000000       01111111111         10000000000
/baugh_wooley/l14  100000000         011111111           100000000
/baugh_wooley/l15  1000000           0111111             1000000
```

Fig.7: Simulation Result of Baugh Wooley Multiplier

Exact modified booth multiplier attains power of about 40(mw) and occupy an area of about 1533 while, baugh woolley multiplier attains power of about 46.94(mw) and occupy an area of about 1845. Thus modified booth multiplier consumes low power and occupy less area when compared to baugh woolley multiplier. The both modified booth multiplier and baugh woolley multiplier are implemented for FFT applications. FFT implemented modified booth multiplier attains power of about 62.96(mw) and area is 5912 while baugh wooley multiplier attains power of about 80.65(mw) and area is 8532. Thus, finally FFT implemented modified booth multiplier achieves low power and area than the exact modified booth, Baugh Wooley and FFT implemented Baugh Wooley multiplier.

VI. CONCLUSION

Modified Booth and the Baugh Wooley multipliers are implemented for FFT application. The partial products are generated and accumulated which are reduced by using a Wallace tree reduction technique and finally the product output is obtained by using Carry Propagate Adder (CPA). Both the multiplier has same simulation outcome, but different operational flow. From the analysis, FFT with Modified Booth multiplier save power of about 14.78% compared with the FFT with Baugh Wooley multiplier design. Similarly FFT with Modified Booth multiplier have a reduction in hardware complexity of about 16.91% than FFT with Baugh Wooley multiplier.

REFERENCES

[1] Y. C. Lim.”Single precision multiplier with reduced circuit complexity for signal processing applications,”IEEE Transactions Computer, vol.41, no.10.pp.1333-1336, Oct. 1992.

[2] S. S. kidambi, F. EI-Gulibaly, and A. Antoniou,” Area-efficient multipliers for digital signal processing applications, “IEEE Transactions circuits system II, Exp. Briefs, vol. 43, no.2,pp.90-94.Feb.1996.

[3] J. M. jou, S.R. Kuang, and R. D. chen, “Design of low-error fixed width multipliers for DSP applications,” IEEE Transactions circuits system I, Exp. Briefs, vol. 46, no.6, pp.836-842.june.1999.

[4] G. M. stroll, N. Petra, and D.D. Caro,”Dual-tree error compensation for high performance fixed width multipliers,” IEEE Transactions circuits system II, Exp. Briefs, vol. 52, no.8,pp.501-5070.Aug.2005.

[5] MacSorley O.L, “High speed arithmetic in binary computers,” Int Proc. IRE, vol. 49, pp 67-69, Jan 1961.

[6] D. Booth,” A Signed Binary multiplication technique.” Quart. J. Math., vol. IV, pp.236-240.1952.

[7] M.A.Song, L.-D. Van, and S.-Y. Kuo.”Adaptive low-error fixed-width Booth multipliers,” IEICE Trans.Fundamentals, vol.E90-A,no.6,pp. 1180-1187, jun.2007.

[8] Jyoti Agarwal, Vijay Matta and Dwejendra Arya,” Design and implementation of FFT processor using Vedic multiplier with high throughput”, International journal of emerging technology and advanced engineering, vol.3, 2013, pp.207-211.

[9] Prabirsaha, Arindam Banerjee, Partha Bhattacharyya and Anup Dandapat,” High speed ASIC design of complex multiplier using Vedic mathematics”, proceeding of the 2011 IEEE Students Technology symposium, IIT Kharagpur,2011,pp.237-241.

[10] Manjunath,VenamaHarikiran, Kopparapu Manikanta,Sivanantham.sandSivasankarank,”Design and mplimentation of 16*16 Modified Booth Multiplier”, Online international conference on Green Engineering and Technologies,Nov 2015.

[11] Pushpalataverma,Methak.k,” Implementation of an efficient multiplier based on vedic mathematics using EDA tool”, International journal of engineering advanced technology, vol.1,2012.