International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 08, August 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

An approach of Modified Radix-8 Booth Multiplier using Verilog

T. Satish Kumar[1], Dr. G.ManmadhaRao[2]

[1]*E.C.E Department, GMRIT College, (India)*

[2]*E.C.E Department, GMRIT College ,(India)*

ABSTRACT

The multiplication operation is present in many parts of a digital system and mostly used in signal processing applications. Several techniques have been proposed to design multipliers, which offer high speed, low power consumption and reduction in area. Booth multiplier has been generally utilized for higher performance by encoding and decreasing the number of partial products. As we know that the performance of radix 8 booth multiplier is slow due to their complexity in nature, proposed an modified radix 8 booth multiplier. The proposed system reduces the complexity and helps to perform faster. The thesis of the work mainly for the design and simulation of modified Radix-8 Booth Encoder multiplier for signed-unsigned numbers. In the approximate radix8 booth multiplier, recoding adder is used which is slow due to complexity when compared to the modified Radix-8 Booth multiplier. The Radix-8 Booth circuit generates n/3 the partial products in parallel. with the extension sign bit of the operands and generating an additional partial product the signed of unsigned Radix-8 Booth Encoder multiplier is obtained .finally the Carry save Adder used to improve speed of the multiplier operation. The extension of the work will propose Modified radix8 booth multiplier to get better performance compared to existing systems.

Keywords**: **Booth Multiplier, Recoding Adder, Partial Product Generator, Carry Save Adder.

I. INTRODUCTION

Now a Days IC technology is obtaining additional advanced in terms of its performance analysis. A multiplier with lower power consumption and smaller space is implicit to the trendy electronic styles. Continuous advancement in electronics technology makes improved the use of energy, communicate info way more firm, etc. In these application systems, a multiplier is a basic arithmetic unit and widely used in circuits that the multiplication method concerns to be optimized properly. in signal processing applications the speed of multiplier operation plays a great significance. In the earlier methodologies, multiplication operation was followed by the successive steps like addition, subtraction, and shift operations. The multiplicand is nothing but the number which is to be added the multiplier, is nothing but the number of times that it is added and finally the result is the product. This repeated addition method is slow that is always replaced by an algorithm because of a low speed. An efficient multiplier should have following characteristics:

Accuracy: A good multiplier should give correct results.

Speed: Multiplier should perform operation at high speed.

Area: A multiplier should occupy less number of slices and LUTs.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Power: Multiplier should consume less power.

Classification of Multipliers:

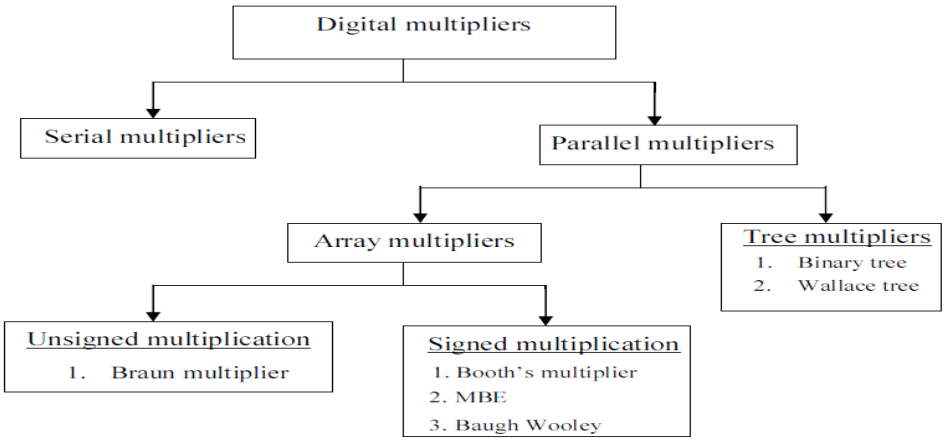

Fig.1. Classification of Multipliers

Multipliers can be classified into two, serial and parallel multipliers. In serial multiplier, for evaluating the partial products each bit of multiplier is used whereas in parallel multipliers, partial product from each bit of multiplier is computed in parallel. In parallel multipliers performance mainly determines the number of partial products that is to be added. In serial-parallel multipliers speed is conceded to achieve good performance in terms of area and power consumption [2]. Parallel or serial multiplier can be used on the type of application. Modified Booth algorithm is one of the most popular algorithms used for reducing the number of partial products.

It is possible to disintegrate multipliers into 2 parts.

1) To the generation of partial products,

2) To the reduce the no. of partial products,

3) Final addition.

For the multiplication of an n-bit multiplicand with an m bit multiplier, m partial products are generated and product formed is n + m bits long. However different multipliers used in present days among these booth multipliers shows better performance. so we discuss about the booth multiplier.

A. Booth multiplier

The algorithm gives a procedure for multiplying binary integers in signed -2's complement representation. Following steps are used for implementing the booth algorithm.

-Let X and Y are two binary numbers and having m and n numbers of bits (m and n are equal) respectively.

1. From two numbers, choose multiplier (X) and multiplicand (Y).

2. Take 2's complement of multiplicand (Y) and given to encoding then bits are formed .

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

3. These bits are multiplied with multiplier and produce partial products.

4. Partial products are the inputs to the recoding adder and produces carry, sum.

5. These are inputs to the precise adder and finally we get result.

To extend the multiplication to both signed and unsigned numbers for obtaining the larger no. of bits Modified Booth Multiplier is used. Modified Booth Multiplier improves speed and reduces the power when compared to the approximate radix8booth multiplier. Booth's algorithmic rule, Wallace Tree methodology etc are some of the algorithms were developed for this purpose. For the summation method many adders like Ripple Carry Addition, Carry Save Addition, Carry Look-ahead Addition, etc.however to scale back the facility consumption the summation design of the number need to be rigorously chosen.

Related work:

In the approximate radix 8booth multiplier, To reduce the ripple carry propagation, two adjacent bits are added [1] (instead of adding just one bit each time as in a conventional scheme) to take advantage of the duplication of the same bit, as shown in the box in Fig. 2. Take any 2- bit addition ($y_{i+1}y_i + y_i y_{i-1}$, where *i*is 1,2,.....,15) as an example; the addition result is given by

$$\begin{aligned} &= 2^i c_{in} + 2^i y_{i-1} + 3\times 2^i y_i + 2^{i+1} y_{i+1}, \\ &2^{i+2} c_{out} + 2^{i+1} s_{i+1} + 2^i s_i \end{aligned} \quad (1)$$

where (y_{i+1}, y_i, y_{i-1} are the 3 bits of the .multiplicand, *Yi* is the duplicated bit, *Cin*is the carry-in from the previousaddition, *Si* and *Si*+1 are the first and second sum bits of the2-bit addition, and *Cout*is the carry-out of the 2-bit adder.

In the approximate radix8 bit adder cannot be used to add the totalof 16 bits in the operands, because a large error would occur when the partial product 3*Y* is required in the multiplier's most significant part. the less significant part of the recoding adder implemented with the help of approximate adder and the most significant part can be implemented by a precise adder[1]. Fig2.

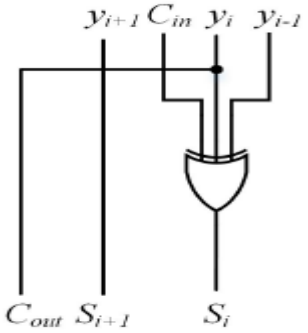

Fig 2.Circuit for approximate 2bit adder

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 08, August 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

Approximate Multiplier Designs

In this section, two approximate 16-bit signed multipliers based on the radix-8 Booth algorithm are proposed [1]. A Booth multiplier consists of stages of multiplier encoding, partial product generation, partial product accumulation and the final addition. In the radix-8 Booth algorithm partial products are generated by the multiplier encoder and the partial product generator. Moreover, a recoding adder is used to implement the sum of the partial products to reduce the total multiplication time. The selection of the partial products as inputs to the recoding adders. Sum and carry as outputs. The input signals of are the multiplier recoding results according to the radix-8 algorithm [11]. Yi is one bit of the multiplicand, and 3Yiis the corresponding bit of 3*Y* calculated by the recoding adder.

Consider a 16-bit signed multiplier, the preliminary addition is shown in Fig. 1 The least significant bit of 3*Y* (*S*0) is the same as *y*0, and the sign bit of 3*Y* is given by *y*15, because the sign does not change when the multiplicand is multiplied by 3.

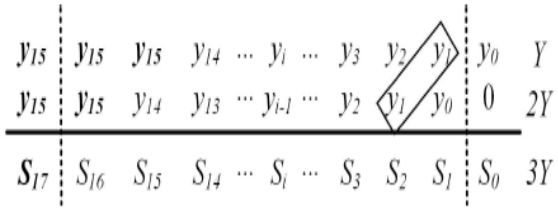

Fig.3.16 bit preliminary addition

Table 1.Truth table for approximate 2bit adder

$C_{out}S_{i+1}S_i$		y_iy_{i-1}			
		00	01	11	10
$C_{in}y_{i+1}$	00	000	001	100	101
	01	010	011	110	111
	11	011	010	111	110
	10	001	000	101	100

Radix-8 Booth recoding applies the same algorithm as that of Radix-4, but now we take quartets of bits instead of triplets. Each quartet is codified as a signed digit using Table II. Radixalgorithm reduces the number of partial products to n/3, where n is the number of multiplier bits.

The partial products in the radix-2 and radix-4 algorithms can be easily generated by shifting. or 2's complementing is implemented by inverting each bit and then adding a '1' in the partial product accumulation stage. However, in the radix-8 algorithm, an odd multiple of the multiplicand Y (i.e., (3Y)) is required and must be calculated.

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 08, August 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

System Architecture of the Approximate Radix 8Booth Multiplier

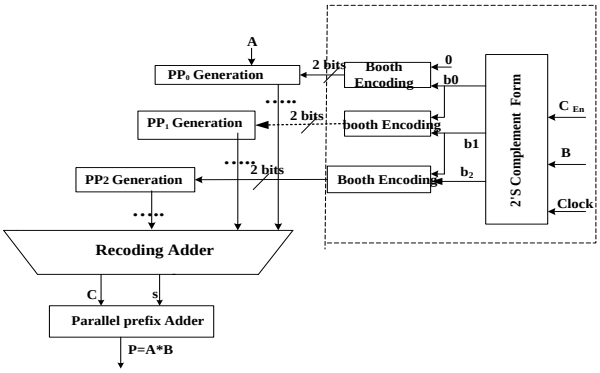

Fig 4.Multiplier with RCA

Approximate radix 8 multiplier is better when compared to the radix-4 algorithms. In Approximate radix 8 multiplier, which incurs additional delay and power cost because by implementing Y + 2Y each time according to recoding adder. so we need to improve the performance the multiplier we design modified radix8 booth multiplier.

II. PROPOSED MODEL

The architecture of the proposed modified radix8 booth multiplier is shown in Fig. It consists of four major modules: Booth encoder, partial product generation, carry save adder. Usually shift and add algorithm had been used but these are not suitable for faster multiplication and delay point of view as more adders will be needed.

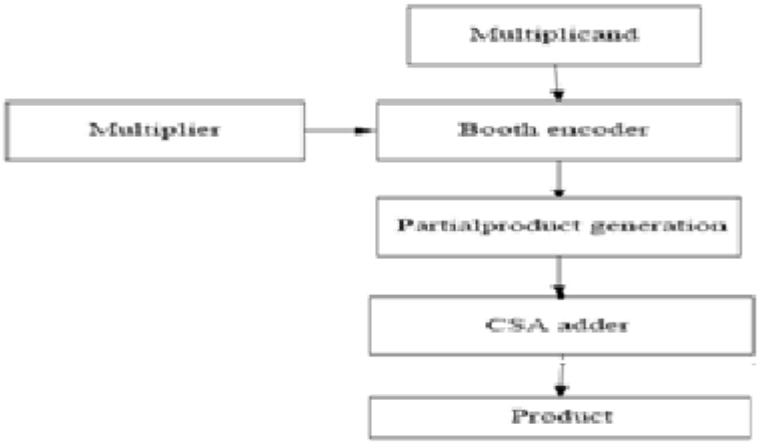

Fig.5.Modified Booth Multiplier

The speed of multiplication can be increased by reducing the number of partial products. The Booth encoder performs Radix-2 or Radix-4encoding of the multiplier bits. Based on the multiplicand and the encoded multiplier, partial products are generated. For large multipliers of 32 bits, the performance of the modified Booth algorithm is used. So the obtained partial products are collected and added by the use of carry save adder.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

carry save adder produces carry and sum. The results are finally added using a Carry Look-ahead Adder (CLA) to get the final product

A. Modified Booth's Algorithm

One of the various solutions of high speed multipliers is enhancing correspondence that helps in decreasing the quantity of ulterior calculation levels.

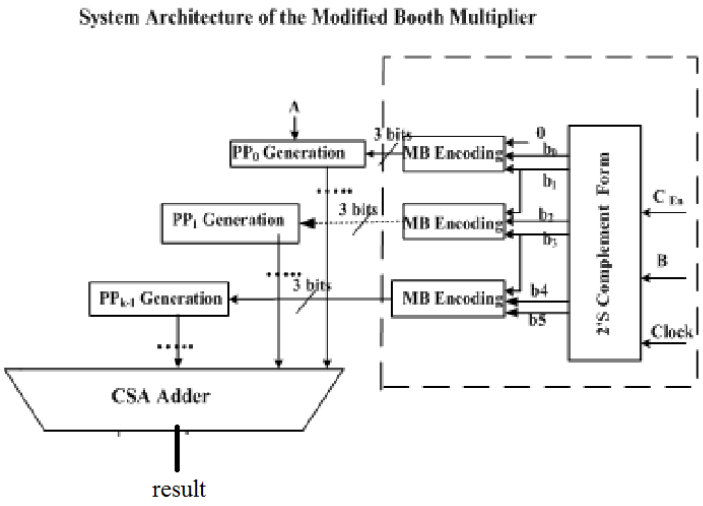

Fig 6.Modified Booth Multiplier

In modified radix 8 booth multiplier for the given inputs A and B the following steps are taken.

- 1st we do 2's compliment for B, the results is given to modified booth encoding algorithm.
- Then 3 bits are generated.
- these bits are multiplied with input A.
- Then we obtain the partial products.
- These partial products are added using Carry select adder..
- Finally, the multiplier output isbeing obtained.

III. RESULTS

The results after implementation have been shown below. The time analysis, power analysis reports, RTL Schematic diagrams are been showed. By which we can state the speed of the system and the area is given by stating the number of device utilization summary.

A. Simulation output

International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 08, August 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

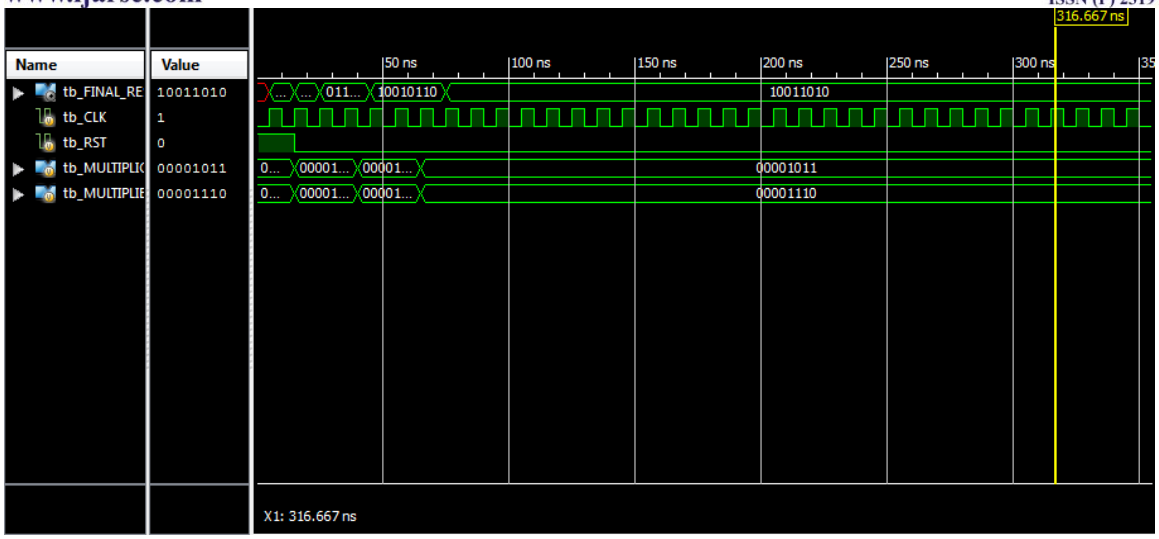

Fig. 7. Wavwform for Radix -8

B. Timing Report

```
Data Path: MULTIPLICAND<4> to FINAL_RESULT_3
                                   Gate     Net
   Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
   ----------------------------------------  ------------
    IBUF:I->O             29   1.218   1.436  MULTIPLICAND_4_IBUF (MULTIPLICAND_4_IBUF)
    LUT4:I0->O             1   0.704   0.000  PP_GEN1/n3/FPP2_6_not0000113_F (N126)
    MUXF5:I0->O            1   0.321   0.595  PP_GEN1/n3/FPP2_6_not0000113
(PP_GEN1/n3/FPP2_6_not0000113)
    LUT3:I0->O             1   0.704   0.424  PP_GEN1/n3/FPP2_6_not0000150_SW0 (N80)
    LUT4:I3->O             1   0.704   0.595  PP_GEN1/n3/FPP2_6_not0000150
(PP_GEN1/n3/FPP2_6_not0000150)
    LUT4:I0->O             3   0.704   0.566  PP_GEN1/n3/FPP2_6_not0000163
(PARTIAL_PRODUCT3<6>)
    LUT4:I2->O             3   0.704   0.566  RA/f7/carry1 (RA/C<7>)
    LUT3:I2->O             2   0.704   0.622  RA/f16/carry1 (RA/C<16>)
    LUT4:I0->O             4   0.704   0.762  RA/f17/carry_and00001 (ADD2<0>)
    LUT4:I0->O             4   0.704   0.587  PREFIX1/S2/n1/Gout1 (PREFIX1/h<1>)
    MUXF5:S->O             3   0.739   0.531  PREFIX1/S2/n6/Gout_f5 (PREFIX1/h<3>)
    MUXF5:S->O             4   0.739   0.622  PREFIX1/S2/n10/Gout_f5 (PREFIX1/h<5>)
    LUT3:I2->O             2   0.704   0.482  PREFIX1/S2/n12/Gout_SW2 (N24)
    LUT3:I2->O             7   0.704   0.883  PREFIX1/S2/n12/Gout (PREFIX1/h<7>)
    LUT3:I0->O             1   0.704   0.455  PREFIX1/Mxor_SUM<5>_Result_SW2 (N32)
    LUT4:I2->O             1   0.704   0.000  PREFIX1/Mxor_SUM<5>_Result (FINAL_SUM<5>)
    FDR:D                      0.308          FINAL_RESULT_5
   ----------------------------------------
   Total                      20.899ns (11.773ns logic, 9.126ns route)
                                       (56.3% logic, 43.7% route)
```

Fig. 8. Timing Analysis

Table 2: comparison of existing & proposed methods

Existing method		Proposed method	
Gate delay	Path delay	Gate delay	Path delay
12.230ns	10.771ns	11.773ns	9.126ns
23.001ns		20.899ns	

IV. CONCLUSION

In this project,we design the high performance of modified radix- 8 Booth multiplier. Initially, an approximate 2-bit adder is employed to implement the approximate recoding adder for generating multiplicand without carry propagation; approximate radix- 8 Booth multiplier designs have been proposed with recoding adder these are low speed and more complexity compared to the modified radix- 8 Booth multiplier. The area and speed of the

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

system is further increased by using modified booth algorithm. In the modified booth multipliers, we use carry save adder to obtain better results. The simulation results are obtained.

REFERENCES

[1] Honglan Jiang, Jie Han "Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation" IEEE Transactions On Computers, VOL. 65, 2016.

[2] G Ganesh Kumar, Subhendu K Sahoo "Implementation of A High Speed Multiplier for High-Performance and Low Power Applications" IEEE Transactions,2015.

[3] Arish S R.K.Sharma "Run-time reconfigurable multi-precision floating point multiplier design for high speed, low-power applications " IEEE Traactions,2015.

[4] M. Vinod Kumar Naik, Mohammed Aneesh. Y, "Design of Carry Select Adder for low power and High Speed VLSI Applications" IEEE Transactions,2015.

[5] Jiun-Ping Wang, Shiann-RongKuang" High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications" IEEE Traactions,2011.

[6] Y.-H. Chen and T.-Y.Chang, "A high-accuracy adaptive conditional-probability estimator for fixed-width booth multipliers," IEEE Transactions on Circuits and Systems I, vol. 59, no. 3, pp. 594–603, 2012.

[7] C. Liu, J. Han, and F. Lombardi, "A low-power, high-performance approximate multiplier with configurable partial error recovery," in DATE, 2014, p. 95.

[8] N. Zhu, W. L. Goh, and K. S. Yeo, "An enhanced low-power high- speed adder for error-tolerant application," in ISIC. IEEE, 2009.

[9] Kuan-Hung Chen and Yuan-Sun Chu "A Low-Power Multiplier With the Spurious Power Suppression Technique" IEEE Transactions2007.

[10] Kyung-Ju Cho, Kwang-Chul Lee, Jin-Gyun Chung, "Design of Low-Error Fixed-Width Modified Booth Multiplier"IEEE Transactions 2004.