Design and Simulation of Low Power and Area Efficient 16x16 bit Hybrid Multiplier

Juili Borkar [1], Dr.U.M.Gokhale[2]

[1]*M.Tech VLSI, Electronics and Telecommunication, GHRIETN, Nagpur, Maharashtra, India*.

[2]*Professor, Electronics and Telecommunication, GHRIETN, Nagpur, Maharashtra, India.*

***Abstract*— Multiplication or repeated addition is the basic arithmetic operation used in both Mathematics and Science. Multiplier is one of the important hardware elements in most of the digital processing system such as digital signal processors, FIR filters and ALU in microprocessors etc. The two important parameters of a multiplier design are its area and speed that are inversely proportional. The main key problem in design of VLSI circuits are larger area utilization, high power consumption and delay which affect the speed of computation and also result in power dissipation. In general, consideration, speed, and power are the essential factor in VLSI design. For solving this problem, a new architecture has been proposed. In proposed system, two high speed multipliers are used such as; modified booth multiplier and Wallace tree multiplier are hybridized with modified carry select adder (CSLA) which delivers high speed computation with low power consumption. Modified booth multiplier is proposed to reduce the partial product where as a Wallace tree multiplier is used for fast addition of partial products and CSLA used for final accumulation. This paper presents design of 16x16 Hybrid Multiplier based on modified booth and Wallace tree architecture.**

***Keywords*: Modified booth algorithm, Wallace tree structure, Carry select adder**

I. INTRODUCTION

A multiplication process consists of two stages. In the first stage, partial products are calculated and then in the next stage the result of these partial products are accumulated. Thus, the speed of a multiplier can be increased either by increasing the speed of partial product calculation or by increasing the speed of accumulation. However, the speed of accumulation depends on the number of partial products to be accumulated. Therefore, the speed of execution can only be increased by decreasing the number of partial products or by increasing the speed of partial product calculation. Here, this research concentrates on decreasing the partial product terms and increasing the speed of partial product calculations with the help of Modified Booth's Algorithm (MBA) and Wallace tree structure. The MBA decreases the number of partial product terms and the Wallace tree structure by using CSA accelerates the accumulation [1].

II.THEORY OF TECHNQUES

A. Ripple Carry Adder (RCA): - A ripple carry adder is a circuit in which n numbers of 1-bit full adders are cascaded in such a way that the carry out of previous stage is connected to the carry in of next stage. Figure below shows the block diagram of a 4-bit Ripple carry adder. Here, in ripple carry adder the final carryout depends on the carry from previous stages. Thus, for the final carry out and sum the circuit will take time equal to the sum of propagation delay of all the n adders which is biggest limitation of this type of adder [2].

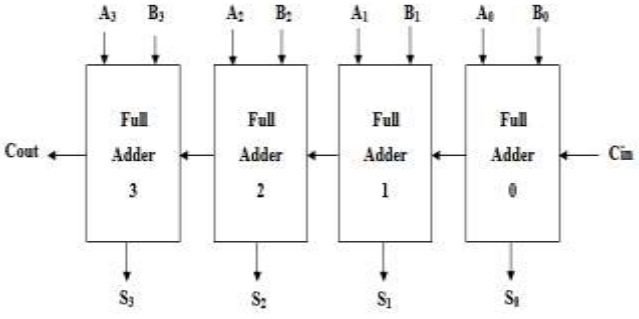

Fig1: Block diagram of Ripple Carry Adder

B. Carry Look-ahead Adder (CLA):- The biggest disadvantage of ripple carry adder is the time taken by the final carry to generate. This problem is overcome in carry look-ahead adder by calculating the carry signal in advance from the applied

inputs. This has been done by modifying the carry output equation full adder. The carry out equation of a full adder is given by

$$C_{i+1} = X_i C_i + Y_i C_i + X_i Y_i \quad -(1)$$

$$C_{i+1} = X_i C_i + C_i(X_i + Y_i) \quad -(2)$$

$$C_{i+1} = G_i + P_i C_i \quad -(3)$$

$$Where, G_i = X_i Y_i \ (G_i is\ Generate\ function)$$

$$P_i = (X_i + Y_i) \ (P_i\ is\ Propagate\ function)$$

Here both the functions Gi and Pi can be generated only in single step since it contains only AND and OR functions .[3] Similary, by using above equations the carry out of each 1-bit full adder of 4-bit ripple carry adder shown in figure.1 is given by

$$C_1 = G_0 + P_0 C_0 \quad -(a)$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) \quad -(b)$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2[G_1 + P_1(G_0 + P_0 C_0)] \quad -(c)$$

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3\{G_2 + P_2[G_1 + P_1(G_0 + P_0 C_0)]\} \quad -(d)$$

C. Carry Save Adder (CSA):-

A carry save adder is a circuit that is used for addition of three or more n-bit numbers. A carry save adder can simply be made by renaming the third input pin i.e. Cin and the carry out pin i.e. Cout of full adder. Fig. below shows the block diagram of 1-bit CSA [3].

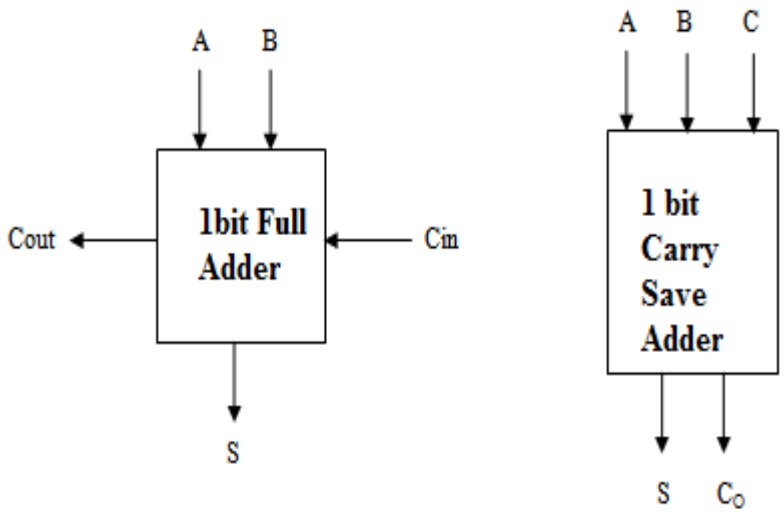

Fig2: Block diagram of Full Adder and Carry save Adder

III.SYSTEM ARCHITECTURE

A conventional multiplier consists of three stages:

1. Partial Products Generation (PPG)
2. Partial Product Manipulation (PPM)
3. Accumulation

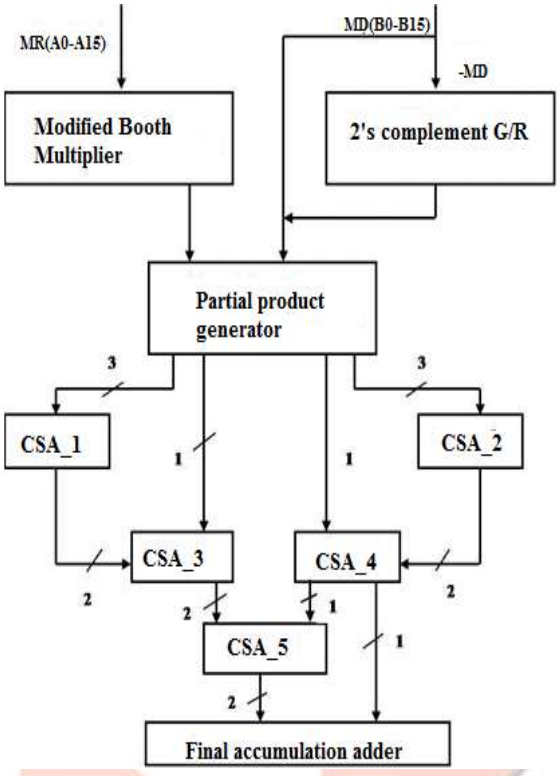

Fig3: Block diagram of Hybrid Multiplier

In proposed work, we are designing a 16x16 hybrid multiplier based on modified booth and Wallace tree architecture. The figure 3 shows the block diagram of Hybrid multiplier. The architecture consists of three stages first stage is modified booth stage, second stage is Wallace tree stage, and third stage is final accumulation stage. Multiplier (MR) and multiplicand (MD) are considered as two inputs to modified booth stage. If negative number(-MD) taken as input then it is, represented in 2's complement and if positive number is consider as input then it will keep as it is. Modified booth stage consists of partial product generation by using modified booth algorithm. In modified booth algorithm, the reduction of partial product takes places by using efficient encoding method which save multiplier area and reduce delay at the same time .Wallace tree structure is used for fast addition and for final accumulation, we used modified CSLA which is a combination of RCA and MUX.

A. Modified Booth stage: -

Modified Booth multiplier:-The modified booth multiplier is work on modified booth algorithm. Modified booth algorithm is used to produce at most n/2+1 partial product where n is number of multiplier bits. There were two drawbacks in original Booth's Algorithm.

1. The numbers of addition, subtraction, and shift operations were not constant causing inconvenience to design parallel adders.
2. When there were isolated ones, the algorithm becomes inefficient. [5]

These two problems are overcome in Modified Booth's Algorithm. O. L. Macsorley first introduced the Modified Booth's Algorithm in 1961. [6]

Modified Booth algorithm [8] is given below-

1. Pad the LSB with one zero value.
2. Pad the MSB with 2 zeros if n is even and 1 zero if n is odd.
3. Divide the multiplier into overlapping groups of three-bits.
4. Determine partial product scale factor from modified booth encoding table as shown in table 1 where M (i+1), M (i) and M (i-1) are multiplier (MR) bits.
5. Compute the Multiplicand Multiples

6. Sum Partial Products

Table 1: Modified Booth Recoding Table for Radix-4[7-8]

M(i+1)	M(i)	M(i-1)	Partial Products
0	0	0	+0*Multiplicand
0	0	1	+1*Multiplicand
0	1	0	+1*Multiplicand
0	1	1	+2*Multiplicand
1	0	0	-2*Multiplicand
1	0	1	-1*Multiplicand
1	1	0	-1*Multiplicand
1	1	1	+0*Multiplicand

2'S COMPLEMENT GENERATOR

When the input (MD) is signed number, The 2's complement of that number will be taken then it will be processed for generating outputs i.e. exactly solving simple traditional method by using 1's complement and adding 1 in generated output.

PARTIAL PRODUCT GENERATOR

Partial product reduction takes place by using Modified booth algorithm by the efficient encoding method. This algorithm can also save multiplier layout area and reduced delay at the same time.

B. Wallace tree stage:-

A Wallace tree is used to multiply two integers. It involves following steps:
First, each bit of one argument is multiplied with each bit of another argument. This will yield 2n results. Now, depending on the bit positions different wires will carry different weights.

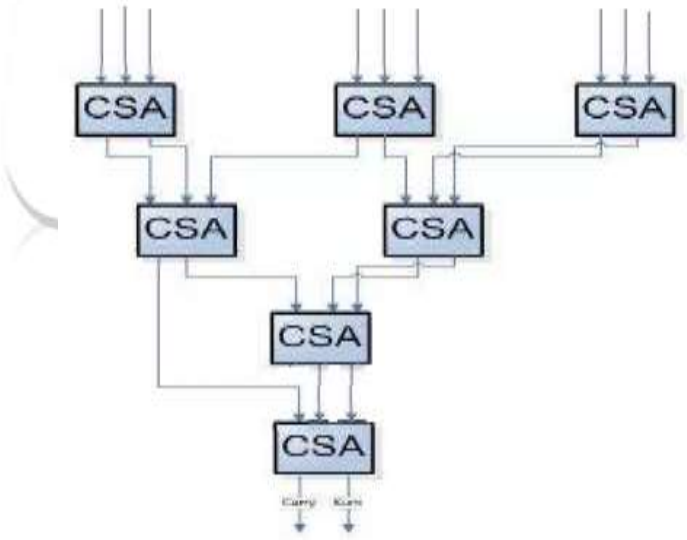

Fig4: Wallace tree structure

Then by using, the layers of half adders and full adders reduce the partial products obtained to two. Now, make a group of two wires and add them by using a conventional adder. Fig. above shows the structure of Wallace tree multiplier.

C. Final accumulation stage:-

Final accumulation stage consists of combination of RCA (ripple carry adder) and MUX i.e. carry select adder (CSLA). As we discuss above CSLA considered as a compromise solution between RCA and CLA because they offer a good trade-off

between compact area of RCAs and short delay of CLAs [9]. Generally, a CSLA has the two ripple carry adder stages whose output is given to a multiplexer. A Carry select adder is used to select the correct result with the help of multiplexer using single stage or multiple stages. For two stages of ripple carry adders, we have the two outputs (2 sum, 2 carry). The correct result will be selected by the multiplexer and speed will be high when comparing with the different adders. [4]

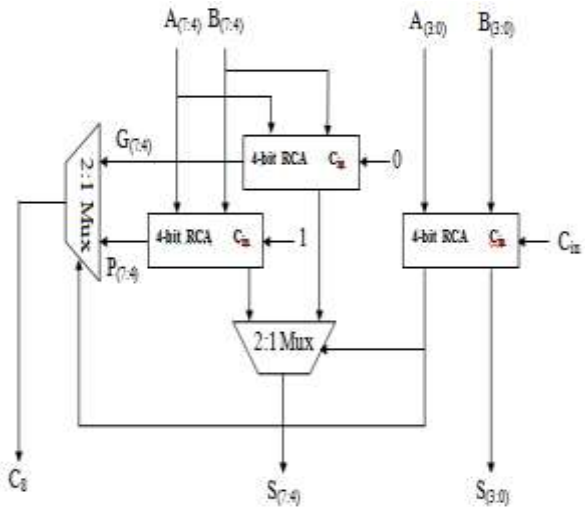

Fig5: Carry select adder

IV. RESULTS

Verilog code is written to generate the design of Hybrid multiplier in Xilinx ISE software and after the successful compilation, the RTL view generated is shown in Fig.6.The figure.9. Shows internal view of hybrid multiplier which consist of modified booth stage, Wallace tree stage and final accumulation stage.

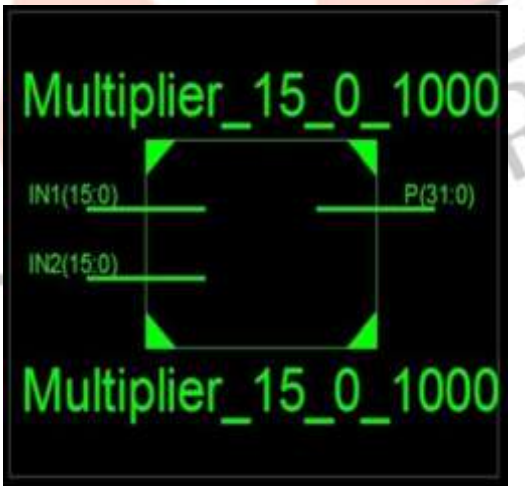

Fig-6: RTL VIEW of Hybrid Multiplier

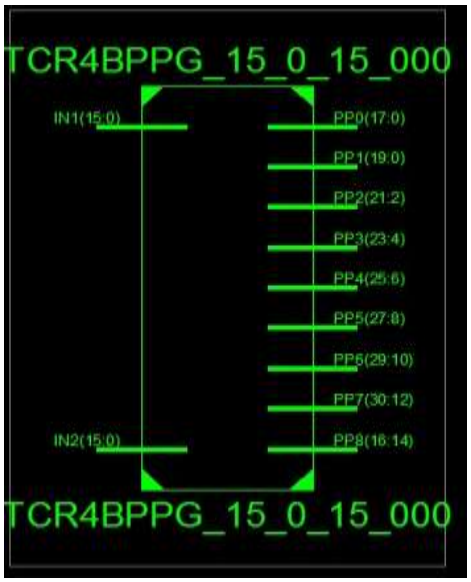

Fig-7: RTLVIEW of Modified Booth Stage

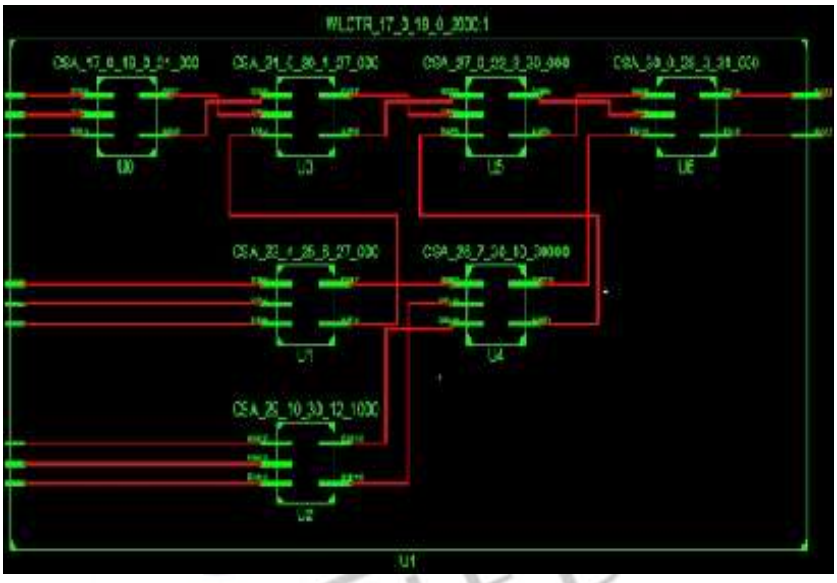

Fig-8: RTL VIEW of Wallace tree stage

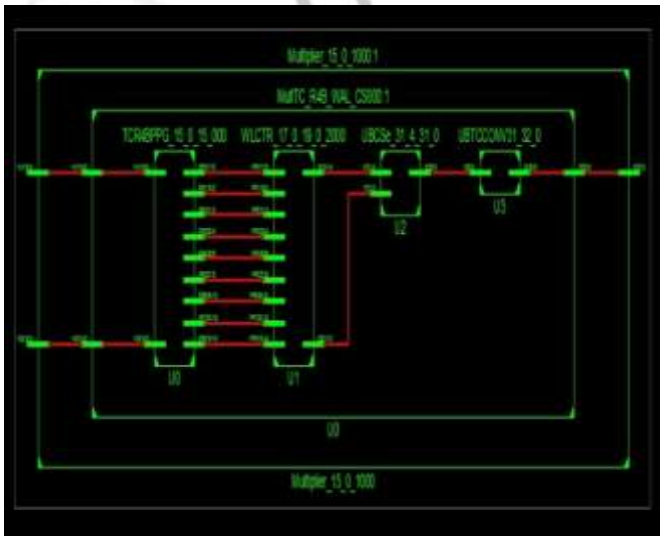

Fig-9: RTL VIEW of Hybrid Multiplier

Fig.10 shows the simulation results of Hybrid multiplier for signed-unsigned number. from figure Hybrid multiplier has two inputs IN1 and IN2 each of 16bit and gives output P of 32bit .for example (27)x(-13)=(-351)

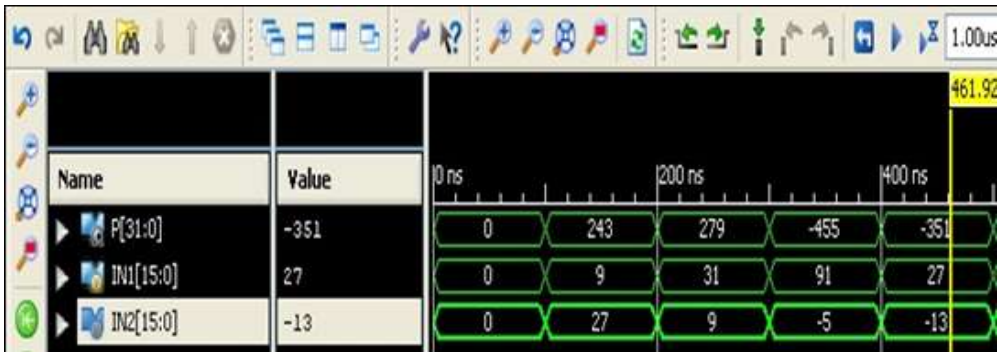

Fig-10: Simulation output of Hybrid Multiplier

Logic Utilization	Used	Available	Utilization
No. of sliced LUT's	369	2400	15%
No. of fully used LUT flip-flop pairs	0	369	0%
No of bonded IOBs	64	102	62%

Table 2: Device Utilization Summary (Estimated Values)

V. CONCLUSION

In this paper, we have designed a Hybrid multiplier based on modified booth and Wallace tree architecture. In proposed system, we get combinational delay of 9.300nsec and power is found to be 0.014 watts. All the circuits are design using verilog language and simulated using Xilinx ISE simulator. The proposed methodology which consists of modified booth algorithm and Wallace tree structure. Modified Booth algorithm used for reduction of partial product which takes places by using efficient encoding method which save multiplier area and reduce delay at the same time .Wallace tree structure design is used for fast addition of partial products and for final accumulation (i.e. for final addition) modified CSLA is used.

REFERENCES

[1] Kostas Tsoumanis, Sotiris Xydis, Constantinos Efstathiou, Nikos Moschopoulos, and Kiamal Pekmestzi, "An Optimized Modified Booth Recoder for Efficient Design of the Add Multiply Operator", IEEE Transactions on circuits and systems—: regular papers, vol. 61, no. 4, April 2014

[2] Yogita Hiremath, Akalpita L. Kulkarni, Dr. J. S. Baligar, "Design And Implementation of Ripple Carry Adder Using Area Efficient Full Adder Cell in 180nm Cmos Technology", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 3, Issue 5 May 2014

[3] N. G. Nik Daud, F. R. Hashim, M. Mustapha and M. S. Badruddin, "Hybrid Modified Booth Encoded Algorithm-Carry Save Adder Fast Multiplier," The 5th International Conference on Information and Communication Technology for the Muslim World (ICT4M), Kuching, 2014, pp. 1-6."

[4] Tauheed Khan, Radhe Shyam Mishra, Ashish Pratap Singh, "High Speed Hybrid Wallace Tree Multiplier," International Journal of Electronics and Computer Science Engineering ISSN 2277-1956/V3 N1-69-75.

[5] Razaidi Hussin, Ali Yeon Md. Shakaff, Norina Idris, Zaliman Sauli, Rizalafande Che IIsmail, and Afzan Kamaraudin, "An Efficient Modified Booth Multiplier Architecture", International Conference on Electronic Design, 978-1-4244-2315-6/08, 2008 IEEE.

[6] Kavita, Jasbir Kaur, "Design And Implementation Of An Efficient Modified Booth Multiplier Using VHDL," International Conference on Emerging Trend in Engineering and Management, *ISSN: 2231-0347,* Volume.3, 3 July2013.

[7] Sukhmeet Kaur, Suman and Manpreet Signh Manna, "Implementation of Modified Booth Algorithm (Radix 4) and its Comparison with Booth Algorithm (Radix-2)," Advance in Electronic and Electric Engineering, *(ISSN 2231-1297),* Volume 3, Number 6 , pp. 683-690,2013.

[8] EPPILI JAYA, K.CHITAMBARA RAO, "Power, area and delay comparison of different multipliers," International Journal of Science, Engineering and Technology Research (IJSETR), *ISSN: 2278 – 7798,* Volume 5, Issue 6, and June 2016.

[9]Manjunath, Venama Harikiran, Kopparapu Manikanta,Sivanantham S*, Sivasankaran K, "Design and Implementation of 16x16 Modified Booth Multiplier'', *Online International Conference on Green Engineering and Technologies* (IC-GET 2015).