Design and Simulation of 16x16 Hybrid Multiplier based on Modified Booth algorithm and Wallace tree Structure

[1]JUILI BORKAR, [2]DR.U.M.GOKHALE
[1]M.TECH VLSI (STUDENT), DEPARTMENT OF ETC, GHRIET, NAGPUR, INDIA.
[2]PROFESSOR, DEPARTMENT OF ETC, GHRIET, NAGPUR, INDIA.

Abstract — Multiplier **is one of the important elements in most of the digital processing system such as digital signal processors, Finite Impulse Response (FIR) filters, and Arithmetic Logic Unit (ALU) in microprocessors etc. The two important parameters of a multiplier design are its area and speed, which are inversely proportional to each other. The speed of a system depends on how a faster an arithmetic operations are performed. The main problem in designing of Very Large Scale Integration (VLSI) circuits are high power consumption, large area utilization and delay which affect the speed of the computation and also results in power dissipation. In general, speed and power are the two essential factors in VLSI design. For solving the issues, a new architecture has been design. In proposed design, two high speed multipliers are used such as Modified Booth multiplier and Wallace tree multiplier are hybridized with carry select adder. Modified Booth multiplier is used to reduce number of partial products by using Modified Booth algorithm whereas Wallace tree multiplier is used for fast addition of partial products and finally, modified carry select adder is used for final accumulation. This paper present design of Hybrid Multiplier using Verilog Hardware Description Language (HDL) and simulated using Xilinx ISE simulator. In proposed system, we get combinational delay of 9.300nsec and number of slice LUT's is found to be 369.**

Keywords— Radix-4 Modified Booth algorithm, Wallace tree multiplier, carry save adder, carry select adder

---

I. INTRODUCTION

Multiplication is one of the fundamental operations in most signal processing systems. The basic multiplication principle is consists of first evaluation of partial products and then accumulation of shifted partial products. As compared with many other arithmetic operations multiplication is more time and power consuming operation. Hence, improving the performance and reducing the power dissipation of multiplier are the two most important design challenges for all application in which multiplier unit dominate system performance and power dissipation. The computation speed of a multiplier is increase by reducing the number of the partial products generated in multiplication process. Fast multipliers are used in many digital signal processing (DSP) and multimedia application in which the output data has directed bearing to the accumulation of series of products over a single multiplication operation. Multipliers design consumes large area and considerable power. Hence, the low power multiplier design has been an important part in low power VLSI system design.

II. REVIEW WORK

Ravindra P Rajput and M. N Shanmukha Swamy [1] gives design and implementation of high speed 8x8 modified Booth encoder multiplier for signed and unsigned numbers. Here, carry save adder and carry look-ahead adder are used for increase the multiplier operation and different simulation output result of 8x8 modified Booth encoder multiplier for signed-unsigned numbers are given in binary form.

B. Ramkumar and H.M. Kittur [2] present the paper on design of carry select adder which gives a simple approach to reduce the area and power of square root carry select adder (SQRT CSLA) architecture. The comparative result of modified SQRT CSLA with regular SQRT CSLA on basis of power-delay product and area-delay product.

Uvaraj Subramanian and Srinivasan Alavandar [3] worked on design of 8x8 multiplier based on hybrid architecture. The 8x8 multiplier is synthesized and implemented on FPGA using verilog HDL and study of comparative output of conventional design and hybrid architecture.

Himanshu Bansal , K. G. Sharma, Tripti Sharma [4] gives the comparative study of various types of the Wallace tree multiplier with conventional Wallace tree multiplier and also reduce the complexity of Wallace tree multiplier, as the number of half adder used in Wallace tree structure is reduce to 80% with increase in number of full adders .

Damarla Paradhasaradhi, M. Prashanthi, N Vivek [5] work on design of Modified Wallace tree multiplier. In this paper, the proposed 8-bit and 16-bit Modified Wallace tree multiplier is designed and also the results are compared with conventional Wallace tree multiplier. The modified Wallace tree multiplier is also studied and designed using Ripple carry adder (RCA) and Square Root Carry Select Adder (SQRT CSLA).

Sukhmeet Kaur, Suman, Manpreet Signh Manna [6] presents the paper which describes the implementation of Radix-4 Modified Booth Multiplier and this implementation is compared with Radix-2 Booth Multiplier. Designing of this algorithm is done by using VHDL (VHSIC Hardware description language) and simulated using Xilinx ISE 9.1i software has been used and implemented on FPGA. The both multiplier are compared by using parameter such as speed in which Radix-4 Booth Multiplier is giving higher speed as compared to Radix-2 Booth Multiplier.

III. METHODOLOGY

In proposed work, we are designing a 16x16 Hybrid Multiplier based on Modified Booth and Wallace tree architecture. The figure.1 shows the architecture of Hybrid multiplier. The Hybrid multiplier architecture consists of three stages 1st stage is Modified Booth stage, 2nd stage is Wallace tree stage, and 3rd stage is final accumulation stage. Multiplier (MR) and Multiplicand (MD) are considered as two inputs to Hybrid Multiplier. If input multiplicand (MD) value is negative number then it will represent in 2's complement form 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 Radix-4 Modified Booth algorithm. In Modified Booth Algorithm (MBA), the reduction of partial product takes places by using efficient encoding method. Wallace tree structure is used for fast addition by using carry save adder (CSA) and for final accumulation we used modified Carry Select Adder (CSLA).

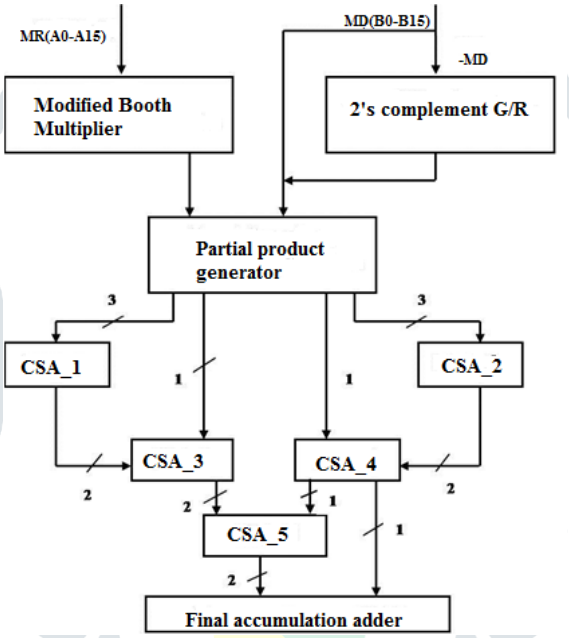

Fig.1. Modified Booth and Wallace Tree Architecture

Modified Booth and Wallace Tree Architecture:-

A. Partial Product Generation Stage:- Modified Booth Stage (MBS) is the first stage of architecture of Hybrid multiplier which consists of Modified Booth multiplier, 2's complement generator, and partial product generator. Modified Booth Stage is nothing but partial product generation stage.

(a) *Modified Booth Multiplier:* - In year 1951, Andrew D. Booth [7] devised a multiplication algorithm, which given named as Booth's algorithm. The Booth algorithm is used for multiplication of two signed number in 2's complement notation. Booth multiplier is designed by using Booth algorithm. The original version of Radix-2 Booth algorithm had two drawbacks. They are as follows [8]

1. The number of add or subtract and shift operations became variable and hence, the design became inconvenient while designing Parallel multipliers.
2. When there are isolated 1's operations (such as 010), the algorithm become inefficient.

These two drawbacks are overcome by Radix-4 Modified Booth Algorithm.

The Modified Booth Multiplier was proposed by D.L.Macsorley [9] in 1961. Modified Booth algorithm (MBA) is one of the most powerful multiplication algorithms for both signed and unsigned number for reducing the number of partial products. It is a high speed multiplier used to enhance parallelism process which helps to reduce number of partial product row, by using Radix-4 MBA overall the number of partial product are deceased from N to N/2 or [(N/2)+1] where N is multiplier bits[10]. The Modified Booth algorithm is extensively used for high-speed multiplier design. By using this technique of Radix-4 Modified Booth

encoding, it is possible to reduce the number of partial products by half. The basic idea of Radix-4 Modified Booth algorithm is that, instead of adding and shifting for every column of multiplier term and multiplying by 1 or 0, we only consider every second column, and multiply by ±1, ±2 or 0, to obtain the same results. Depending on multiplier bits, radix-4 modified Booth encoder performs the process of encoding the multiplicand. It will compare three bits at a time with overlapping technique. First, the grouping starts from the least significant bit (LSB), and the first block only uses 2 bits of multiplier and assumes a zero for the third bit as shown by figure.2 and recoded values are taken from table.1 where M(i+1), M(i), M(i-1) are multiplier bits.

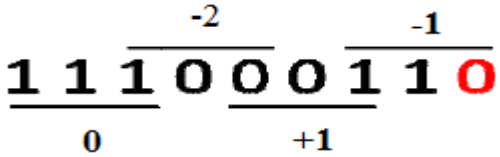

Fig.2. Three Bit pairing as per Booth encoding

TABLE 1. Booth Recoding Table for Radix-4

M(i+1)	M(i)	M(i-1)	Operations
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

(b) 2's Complement Generator: - When input (MD) is negative number then 2's complement generator is used for generating 2's complement form of that input number. Then generated 2's complement form is given as one of the input to partial product generator.

(c) Partial Product Generator: - The partial product reduction takes place by using Radix-4 Modified Booth algorithm by using efficient encoding method as discuss above in Modified Booth multiplier.

B. Partial Product Reduction Stage:-

The second stage of architecture of Hybrid Multiplier is Wallace tree stage which is nothing but partial product reduction stage. The partial product generated at Modified Booth stage is consider as input to Wallace tree stage or Wallace tree multiplier. In year 1961, C.S.Wallace [11] proposed a method for a carry save adder like reduction scheme which is valid for columns of any length which is known as Wallace tree. The figure.3 shows the Wallace tree structure by using CSA. To achieve high performance in digital signal processing system, the Wallace tree multiplier is used which enhance the speed of the system. The Wallace tree multiplier is used for fast addition of partial product with the help of carry save adder. Wallace tree structure can be used to speed up the computation by reducing the number of sequential stages. The Wallace tree structure is designed by using CSA with (3:2) compressors.

The Wallace tree structure [12] has three steps:-

1. Multiply (i.e. AND) each bit of one of the arguments, by each bit of the other. Now, depending upon the position of the resultant multiplied bits different wires carries different weights.
2. The numbers of partial products are reduce to two by layers of full and half adders.
3. Group the wires in two numbers, and add them by using conventional adder.

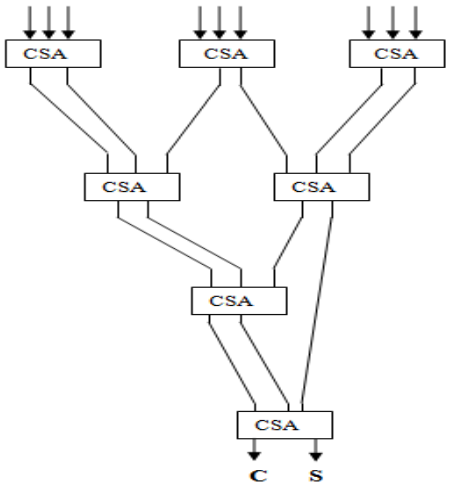

Fig.3. Wallace Tree Structure

C. Final Addition Stage:-

The Final accumulation stage is the third stage of architecture of Hybrid Multiplier. This stage is used as final addition stage which consists of combination of RCA (ripple carry adder) and Multiplexer i.e. Modified Carry Select Adder (CSLA).The modified CSLA is the final accumulation adder. Generally, the carry-select adder consists of two RCAs and a multiplexer. Adding two n-bit numbers by using carry-select adder is done with two adders (therefore two ripple carry adders). In CSLA to perform the calculation twice, one time carry-in consider as zero and the other assuming it will be one. After the results are calculated, the correct sum and carry-out, is then selected by using multiplexer once the correct carry-in is known. The output of Wallace tree stage is given as input to final accumulation stage which is a modified CSLA. Carry select adder generate sum and carry at the output. In final accumulation stage, the carry generated in CSLA is accumulated by input/output blocks. For final addition we are consider CSLA because it is consider as compromise solution between RCA and Carry look-ahead adder (CLA) because they offer a good trade-off between compact area of RCAs and short delay of CLAs [13].The figure.4 shows carry select adder structure.

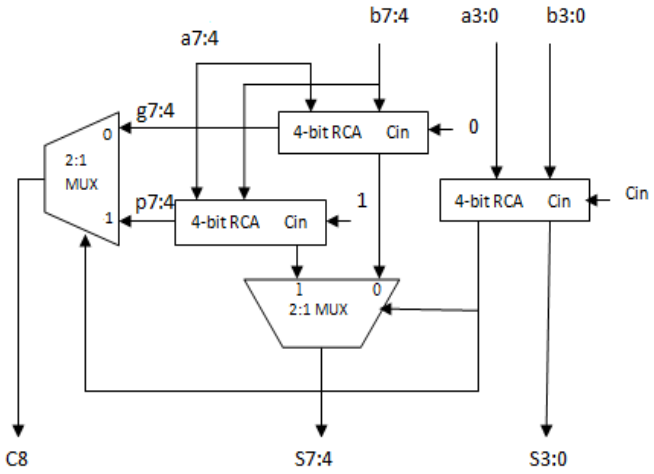

Fig.4. Carry Select Adder

IV. RESULTS

HYBRID MULTIPLIER

All design of Hybrid multiplier are designed using Verilog HDL in Xilinx ISE 13.1 software and the Schematic Technology view of Hybrid Multiplier is shown in figure.5. Multiplier is consists of two inputs i.e. multiplicand and multiplier. From figure.5 Hybrid multiplier, has two inputs IN1 is from (15:0), IN2 is from (15:0), and one output P (31:0).

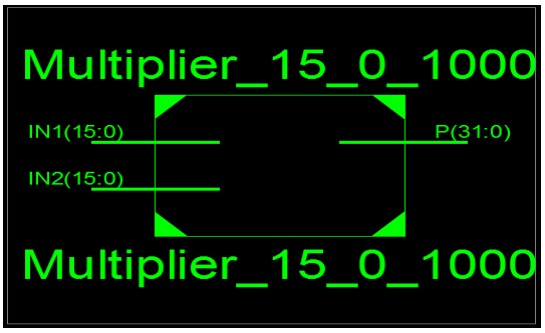

Fig.5. Schematic Technology View of Hybrid Multiplier

The RTL View of Hybrid Multiplier is shown in figure.6 which consist of two 16 bits inputs IN1, IN2 and one 32 bit output P.

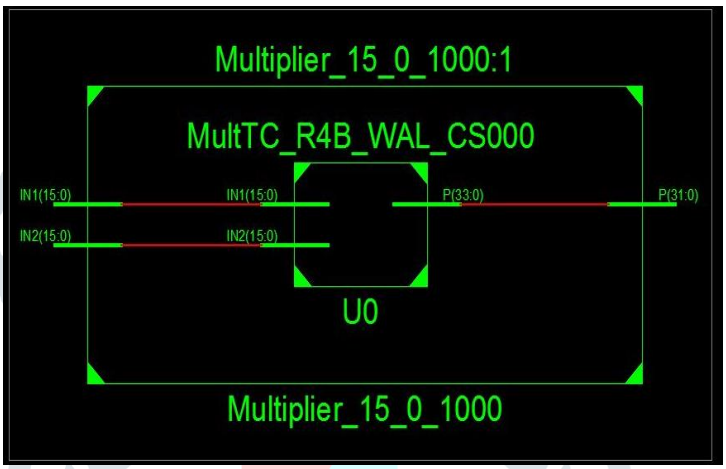

Fig.6. RTL View of Hybrid Multiplier

The simulation output of Hybrid Multiplier is shown in figure.7 in signed decimal form which consist of two inputs IN1 [15:0] and IN2 [15:0] and output is given as P [31:0]. The Figure.7 shows multiplication in signed decimal form as

$(27)_{10} * (-13)_{10} = (-351)_{10}$

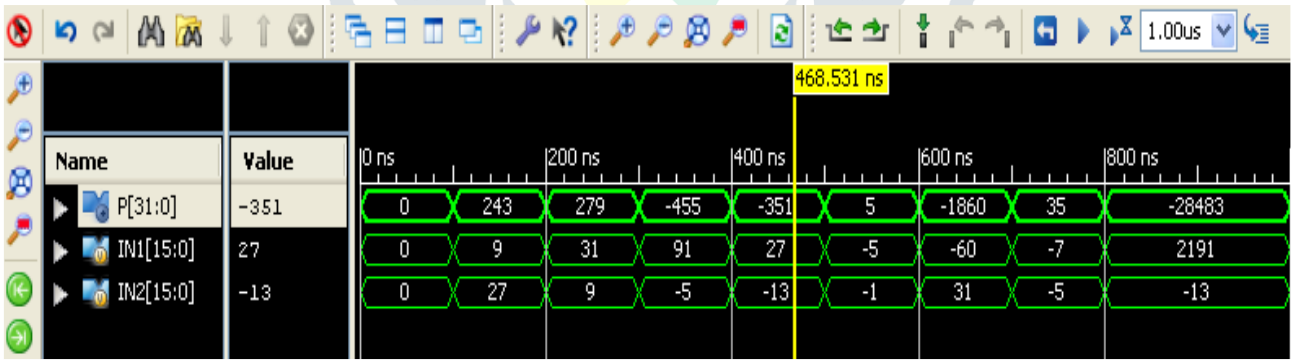

Fig.7. Simulation Output of Hybrid Multiplier in Signed decimal form

The simulation output of Hybrid Multiplier in Unsigned decimal form is shown in figure.8 which consist of two inputs IN1 [15:0] and IN2 [15:0] and one output P [31:0].

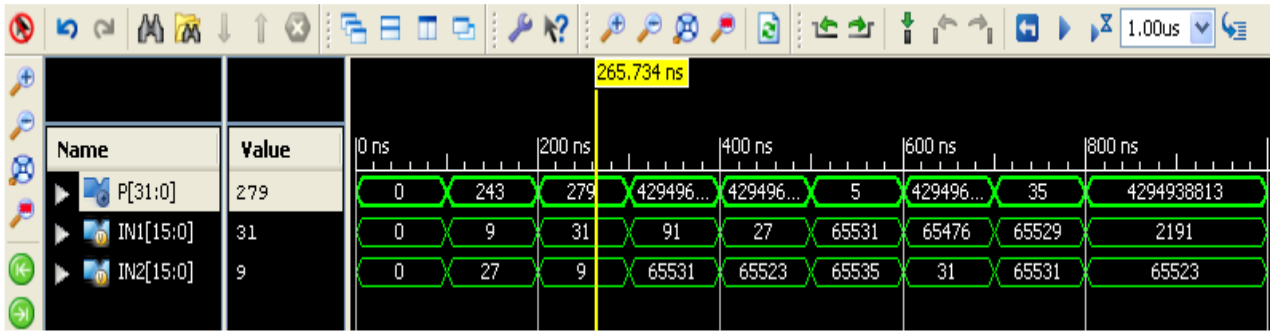

Fig.8. Simulation Output of Hybrid Multiplier in Unsigned decimal form

The simulation output of Hybrid Multiplier is shown in fig. 9 in Hexadecimal form which consist of two inputs IN1 [15:0] and IN2 [15:0] and output P [31:0].

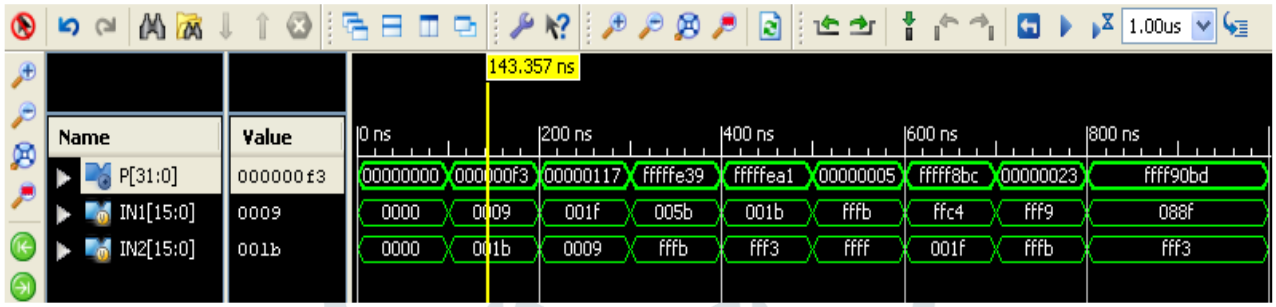

Fig.9. Simulation output of Hybrid Multiplier in Hexadecimal form

The simulation output of Hybrid Multiplier in Binary form is shown in figure.10 which consist of two inputs IN1 [15:0] and IN2 [15:0] and output P [31:0].

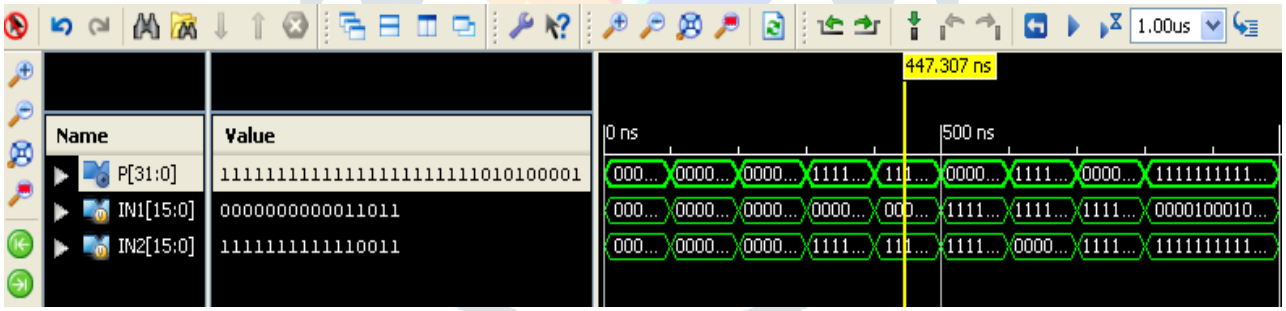

Fig.10.Simulation output of Hybrid Multiplier in Binary form

The device utilization summary of Hybrid multiplier is shown in figure.11 which consist of number of LUTs, number of fully used LUT-FF pairs and number of bonded IOBs.

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	369	178800	0%
Number of fully used LUT-FF pairs	0	369	0%
Number of bonded IOBs	64	600	10%

Fig.11. Design Utilization Summary of Hybrid Multiplier

The power calculation of Hybrid multiplier is shown in figure.12. The power report can be observed in Xilinx power analyzer the total power in Hybrid Multiplier is 0.546 Watts.

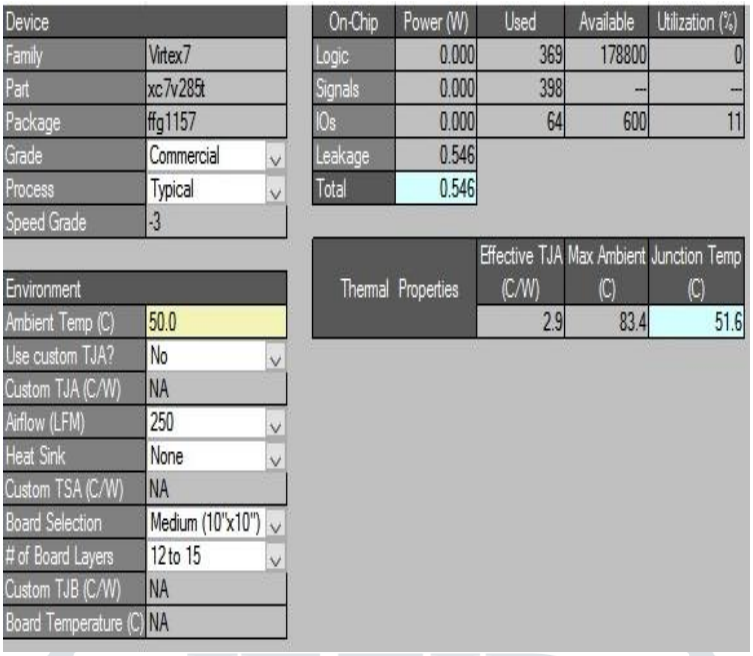

Fig.12. Power Calculation of Hybrid Multiplier

V. Comparative Study of Hybrid Multiplier

For comparative study of Hybrid Multiplier, we have designed Modified Booth Multiplier. From table.2 after analyzing both Hybrid Multiplier and Modified Booth Multiplier and compare, their characteristics such as area and delay we found that Hybrid multiplier required less area and delay than Modified Booth Multiplier for same number of inputs.

Table. 2. Comparison of Modified Booth Multiplier and Hybrid Multiplier

Parameters	**Modified Booth Multiplier**	**Proposed Hybrid Multiplier**
Device Utilization Summary		
Number of Slice LUT's	441	306
Number of occupied slices	149	110
Number of LUT flip-flop pairs used	441	306
Number of bonded IOB's	64	64
Combinational Delay	10.311 ns	9.300 ns

The table 3 shows Comparison of 16x16 Hybrid multiplier in Spartan 3 family with Booth Multiplier and Vedic Multiplier. All the above values of 16x16 Hybrid Multiplier is calculated on device xc3s400-5pq2083 of Spartan 3 family in Xilinx ISE

software. After Comparison, it is concluded that Hybrid Multiplier required less number of slices, number of 4 input LUTs as compared with Booth Multiplier and Vedic Multiplier.

Table.3. Comparison of Hybrid Multiplier in Spartan 3 family

Parameter	16x16 Booth Multiplier[14]	16x16 Vedic Multiplier [14]	Proposed 16x16 Hybrid Multiplier
No. of slices	718 out of 3584 (20%)	345 out of 3584 (9%)	299 out of 3584 (8%)
No of 4 input LUTs	1281 out of 7168 (17%)	622 out of 7168 (8%)	532 out of 7168 (7%)
Number of IOBs	64 out of 141 (45%)	64 out of 141 (45%)	64 out of 141 (45%)

The table 4 shows Comparison of 16x16 Hybrid multiplier with Vedic Multiplier using BEC and Array Multiplier. All the above values of 16x16 Hybrid Multiplier is implemented on device xc3s500-5fg320 of Spartan 3E family in Xilinx ISE software. After Comparison, it is concluded that Hybrid Multiplier required less number of 4 input LUTs and number of IOBs as compared with Vedic and Array Multiplier and it is clear that delay generated in Hybrid Multiplier is less as compared to both Multiplier.

Table.4. Comparison of Hybrid Multiplier in Spartan 3E family

Logic Utilization	16x16 Vedic Multiplier using BEC [15]	16x16 Array Multiplier [15]	Proposed 16x16 Hybrid Multiplier
Number of 4 input LUT's	1243/9312	844/9312	530/9312
Number of IOBs	66/232	66/232	64/232
Delay in ns	38.82	61.49	30.887

The table 5 shows Comparison of 16x16 Hybrid multiplier and all the values of Hybrid Multiplier is calculated on device XC5VLX30 of Vertex-5 family in Xilinx ISE software. After Comparison, we found that number of LUTs required in Hybrid Multiplier is 370 which less than proposed Vedic multiplier and Booth Multiplier and also it required less delay.

Table.5. Comparison of Hybrid Multiplier in Virtex-5 family

Device:Vertex5, XC5VLX30, Speed Grade: -3	Proposed Vedic Multiplier[16]	Booth Multiplier[16]	Proposed Hybrid Multiplier
Delay	21.387ns	26.231ns	14.371ns
Number of LUTs	447 of19200	600 of19200	370 of19200

VI. CONCLUSION

We have presented a design of 16x16 Hybrid Multiplier based on Modified Booth and Wallace tree architecture. In proposed system, we get combinational delay of 9.300nsec and number of slice LUT's is found to be 369. For comparative study of Hybrid multiplier, we have design Modified Booth Multiplier. After, analyzing both multiplier and compare their characteristics such as area and delay we found that Hybrid multiplier required less area and delay than Modified Booth Multiplier for same number of inputs. The architecture of Hybrid multiplier is the combination of high speed multipliers and adders. Hybrid Multiplier will be used for both signed and unsigned number multiplication. The design can be used for meeting the challenges in all signal processing applications by its faster operations. The speed of the computation is increased since the partial products are reduced by using Radix-4 Modified Booth algorithm. 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] Ravindra P Rajput, M. N Shanmukha Swamy, "High speed modified Booth Encoder Multiplier for Signed and Unsigned Numbers,'' 14th International Conference on Modelling and Simulation, Pages: 649 – 654, March 28-30, 2012.

[2] B. Ramkumar and H.M. Kittur, "Low-power and area-efficient Carry-Select Adder," IEEE Transactions on Very Large Scale Integration (VLSI) System, Volume 20, Number 2, Pages: 371–375, February 2012.

[3] Uvaraj Subramanian, Srinivasan Alavandar, "Low-Power and high speed computation using hybridized Multiplier," Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT) IEEE 31661, July 4-6, 2013.

[4] Himanshu Bansal, K. G. Sharma, Tripti Sharma, "Wallace tree multiplier designs: a performance comparison review", Innovative Systems Design and Engineering *ISSN 2222-1727* (Paper), *ISSN 2222-2871* (Online) Volume.5, Number.5, 2014.

[5] Damarla Paradhasaradhi, M. Prashanthi, and N Vivek, "Modified Wallace tree multiplier using efficient Square Root Carry Select Adder," International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), March 2014.

[6] 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.

[7] A.D. Booth, "A Signed Binary Multiplication Technique," Quarterly Journal of Mechanics and Applied mathematics, Volume-IV, pt-2-1951.

[8] Nishat Bano, "VLSI Design of Low Power Booth Multiplier," International Journal of Scientific & Engineering Research, *ISSN 2229-5518*, Volume 3, Issue 2, February 2012.

[9] Kavita, Jasbir Kaur, "Design and Implementation of an Efficient Modified Booth Multiplier using VHDL," International Conference on Emerging Trend in Engineering and Management (ICETEM), *e-ISSN:2231-0347, Print-ISSN: 2231-2013*, pp. 78-81, Volume.3 (3), July2013.

[10] Sumod Abraham, Sukhmeet Kaur, Shivani Singh, "Study of Various High Speed Multipliers" International Conference on Computer Communication and Informatics (ICCCI), January 08 -10, 2015.

[11] C. S. Wallace, "A suggestion for a fast multiplier," Electronic Computers, IEEE Transactions paper, volume: 13, Page(s): 14-17, February 1964.

[12] Dr.V.Malleswara Rao, K.V.Ganesh, P.Pavan Kumar, "Comparison of low power and delay using Baugh Wooley and Wallace Tree Multipliers," International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 3, Issue 9, September 2014.

[13] 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).

[14]Pranita Soni, Swapnil Kadam, Harish Dhurape, Nikhil Gulavan, "Implementation of 16x16 Bit Multiplication Algorithm By Using Vedic Mathematics over Booth Algorithm", International Journal of Research in Engineering and Technology(IJRET) *eISSN: 2319-1163 ,pISSN*: 2321-7308, Volume: 04 Issue: 05 , May-2015.

[15] G.Challa Ram, D.Sudha Rani, Y.Rama Lakshmanna, K.Bala Sindhuri, "Area Efficient Modified Vedic Multiplier", International Conference on Circuit, Power and Computing Technologies (ICCPCT), 978-1-5090-1277-0, IEEE-2016.

[16]D.Kishore Kumar, A.Rajakumari, "Modified Architecture of Vedic Multiplier for High Speed Applications," International Journal of Engineering Research & Technology (IJERT), *ISSN: 2278-0181*, Volume 1 Issue 6, August – 2012.