International Journal of Engineering & Technology, 7 (1.5) (2018) 31-36

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Improvement of the efficiency of booth multiplier

M Siva Kumar, Sanath Kumar Tulasi, N Srinivasulu, G S Krishnam Naidu Yedla, E Raghuveer, K Hari Kishore

Department of ECE, Koneru Lakshmaiah Educational Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India 522502
**Corresponding author E-mail: siva4580@kluniversity.in*

Abstract

Objective: To Improve the performance of Booth Multiplier and reduce power consumption.
Method: The most essential form of multiplication consists of framing the result of two unsigned (positive) binary numbers.
Finding: Booth Multiplier consists of pre-defined table. According to this algorithm, multiplication of two numbers x and y (x*y) is same as multiplication of y and x (y*x). At times this rule fails due to which we modify the logic by converting the decimal number in to 4 bit binary number and appending (n-2) zeros at most significant bit and one zero at least significant bit.
Improvement: By using this method we can get accurate results in multiplication by multiplying like (x*y) and (y*x).

Keywords: *Booth Multiplication, Layout, Partial Product, Booth Algorithm*

1. Introduction

The most basic form of multiplication[1] consists of framing the result of two unsigned (positive) binary numbers. The mostly used multiplier is booth multiplier and it consists of predefined table . According to the principle of booth multiplier if we multiply two numbers x and y (x*y) then the result of these two numbers is same as if we do
multiplication in reverse order that is y*x but it fails to give correct output so we are modifying the logic by converting these in to 4 bit binary number[1] and appending n-2 zeros at msb and one zero at lsb.so we write code to convert them in to 4 bit binary and appending zeros to it and we analyse in Xilinx

2. Booth Multiplication

Booth multiplication[1, 2, 3] is one of the increase methods that takes into account smaller, speedier operation on the numbers.It is the standard strategy utilized as a part of chip plan, and gives significant enhancements over the "long multiplication procedure. Booth's multiplication calculation is a multiplication calculation that increases two signed[8] binary numbers in two's complement documentation.

The partial products can be identified as follow

2.1 Booth Algorithm

In the booth algorithm it uses predefined table which was given by the scientist booth. In this booth algorithm we multiply the two numbers and codes the number which is to be multiplied by using this table.
In the booth algorithm[1,6] the multiplier is taken as a 4 bit binary number and appends zero at lsb and from left to right we pair 3 bits as a pair which are used for partial product generation. the steps of the booth algorithm is mentioned below.

Table 1: Showing Booth Algorithm Partial product

Inputs			**Partial products**
X_2	X_1	X_0	Pp
0	0	0	0
0	0	1	Y
0	1	0	Y
0	1	1	2y
1	0	0	-2y
1	0	1	-y
1	1	0	-y
1	1	1	0

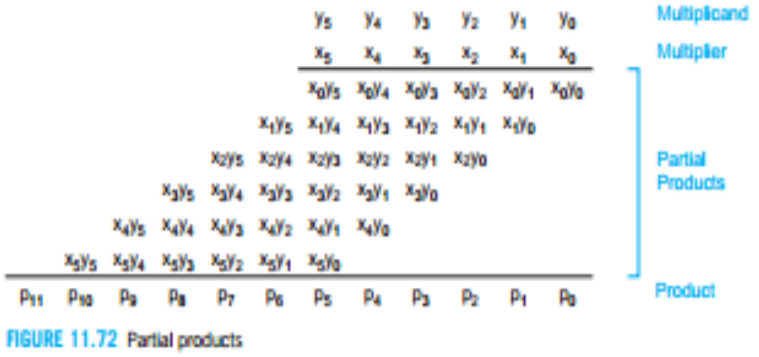

Fig.1: Partial Products Steps for booth example

Figure 1 gives the total number of partial products that can be obtained for the booth operation with multiplicand, multiplier, partial products and product

For example

8*2=16

Number of flip flops required: x+y-1

X=1000

Y=0010

X+y-1=7flipflops

Y=00100

001 100

p1 p2

P1=p0+100=p0+(-2x)

P2=p1+001= p1+(x)

3. Proposed model

In the above algorithm the multiplicand is multiplied with multiplier For an example six is multiplied with 2 (6*2=12)that means six is added twice And similarly two is multiplied[1] with 6 (2*6=12) that means two is added six times will results to output and this procedure gives more delay when compared both.In the above algorithm the process is done by appending zero at lsb and from lsb to msb 3 bits are paired and the partial products can be obtained if the number of paired bits increases then the partial products increases,and the procedure is followed and if we not get the required output that is based on binary multiplication process output then add one zero at msb and do the same process and by this time of execution increases.

To eliminate this delay we can do this process by embedding two zeros at most significant bit to the multiplier and do the process and the output will get in the same time The example is that condition is not satisfying the already existed algorithm and satisfies the proposed algorithm

And the example is (8*2=16,2*8=16)

4. Outputs of the proposed algorithm

The multiplicand is =1000 (8)

Multiplier is =0010(2) Output=8*2=16

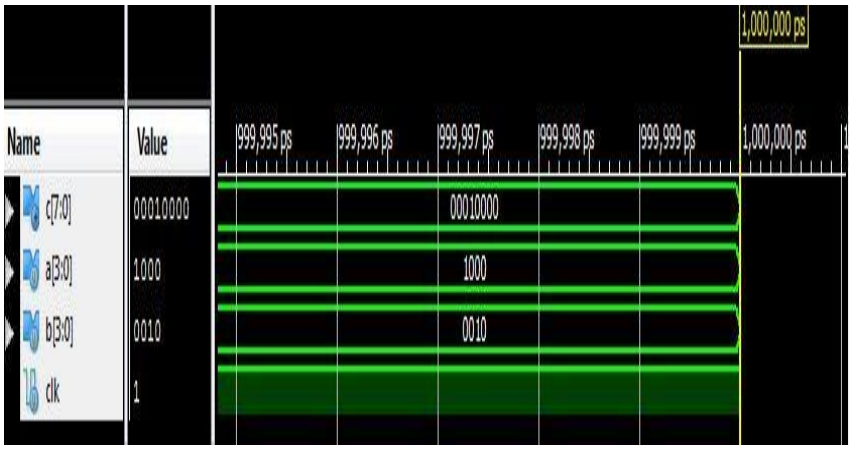

Figure 2. Showing Output=2*8=16

Figure 2 shows when multiplicand is 1000 and multiplier is 0010 and then output obtained is 16.

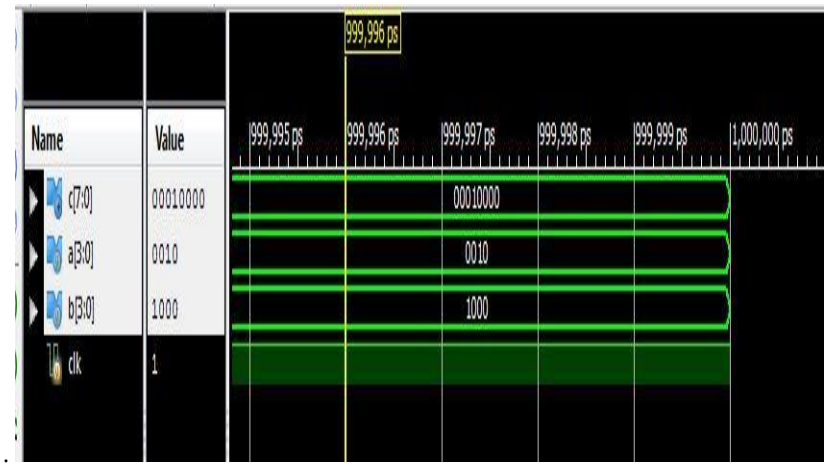

Figure 3.Test bench Diagram

In the above figure 3 represents the inputs are given as a,b and the output is stored in c=a*b,and the inputs are changed as b,a and the output is c=b*a..

5. Layout using micro wind software

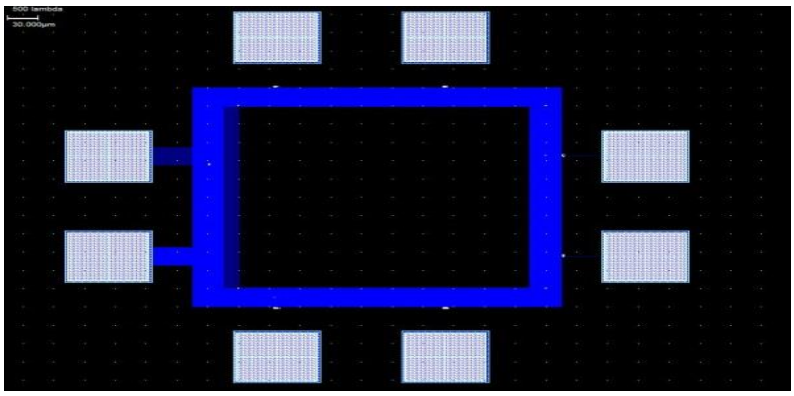

Fig.4: Layout of the booth multiplier

The above figure 4 shows the layout of booth multiplier when Verilog code is taken and the layout of this obtained by using the microwind software.

Three-dimensional view of the booth multiplier when implemented on the substrate

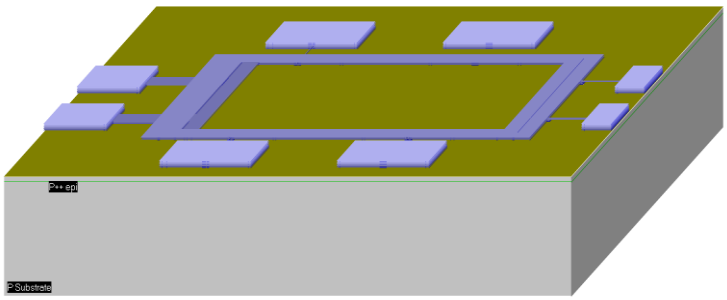

Figure 5: 3d view after fabrication

After processing several steps like photoresist, masking oxidation, metallization techniques the output circuit looks like a

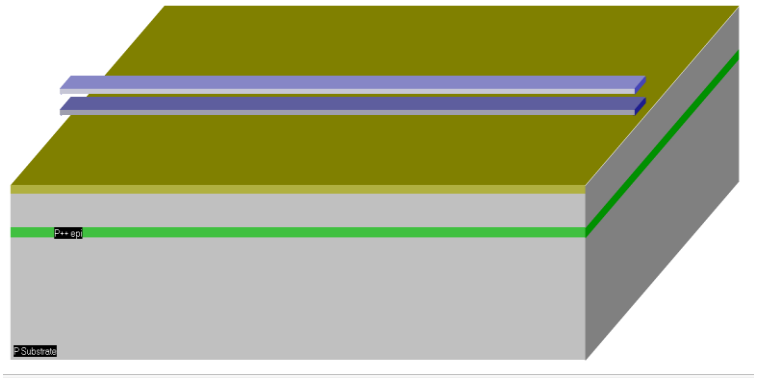

Fig.6: Final 3d view

Figure 6 is the final 3d view after processing technique completion. The green color line shows the connection between the silicon substrate to oxidation layer which is called epitaxy layer

Middle view shows the diagram is

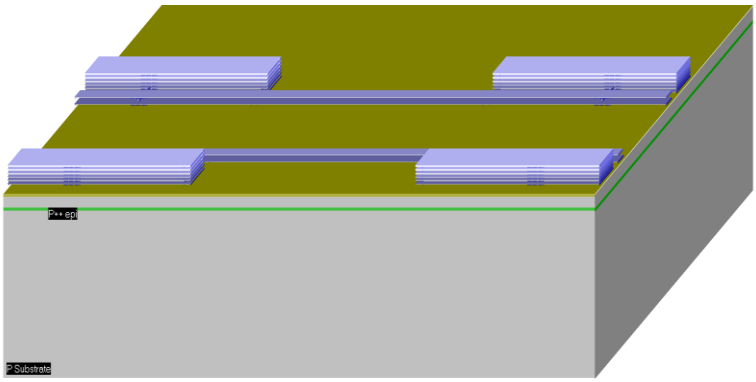

Figure 7: Middle View for the Algorithm

6. Schematics

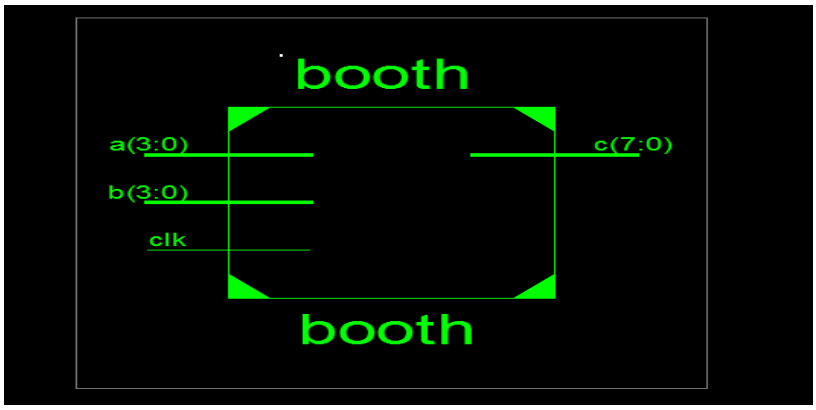

Fig. 8: is the schematic for the booth algorithm when implemented in xilinx

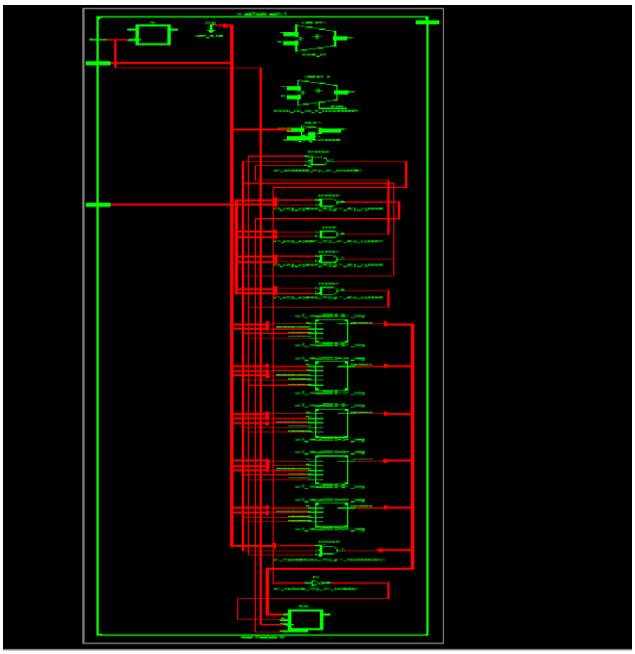

Fig.9: RTL VIEW

Figure 9 shows the register transfer level view for the booth algorithm.
The number of LUTS is used and the no of input output blocks usage can be known after synthesizing the code in Xilinx software.

When implemented in Xilinx table 2 shows the various parameters with their device utilization summary.

Table 2: Table shows device utilization summary

Logic utilization	Used	Available	Utilization
No. of Slices	16	4656	0%
No. of slices flip-flops	12	9312	0%
4inputLUT	29	9312	0%
Bonded IOB	17	232	7%
No. of Gclk	1	24	4%

7. Delay Analysis

Before clock minimum input arrival time: 4.929 ns
After clock maximum output required time: 7.204 ns ref 4
Maximum combinational path delay:No path found

8. Advantages

The speed increases for the reversible logic of the multiplication process will occur in same time so by this the system reaction is not different for different inputs and all will gives output at same time

9. Limitations

The main limitation of this is that if we increase the number of appended bits by this the usage of flip flops increases7 so the size increases by this the Power consumption[5] increases .

10. Applications

- Micro processors
- Filtering process(FIR)
- Digital signal processors
- IF stages of the receiver

11. Conclusion

In this paper we have obtained the reversible logic of multiplication process that is x is multiplied with y and y is multiplied with x gives same results with same time and no delay is obtained by using modified booth multiplier.

12. Future scope

The limitations of this paper can be over comed by separating the booth existing table in two parts if there is any partial product which consists negative outcome and this can be defined by using this booth multiplier table separate for negative and positive outcome.

References

[1] AmitaNandal,T.Vigneswaran,Ashwanik.rana Booth Multiplier using Reversible Logic with Low Power and Reduced Logical Complexity,
[2] Indian Journal of Science and Technology, Vol 7(4), 525–529, April 2014

[3] Neha Goyal, Khushboo Gupta, Renu Singla Study of Combinational and Booth Multiplier, International Journal of Scientific and Research publications. Volume 4, Issue 5,May 2014.
[4] Thomas M. Design and simulation of radix-8 booth encoder multiplier for signed & unsigned numbers. International Journal for Innovative Research in Science and Technolgy. 2014 Jun.
[5] Jonnalagadda Raghavendra and T. Vigneswar,
[6] Design of Fused Add-Multiply Operator using Modified Booth Recoder for Fast Arithmetic Circuits. Indian Journal of Science and Technology, Vol 8(19), IPL0149, August 2015.
[7] M.V.Jithin Kumar and K.B. Jayanthi, A Low Power Hybrid Multiplication Technique for Higher Radix Hard multiples Suppression, Indian Journal of Science and Technology Vol 8(13), 54495, July 2015.
[8] Sukhmeet Kaur, Suman and Manpreet Signh
[9] Implementation of Modified Booth Algorithm and its Comparison with Booth Algorithm. Advance in Electronic and Electric Engineering. Volume 3, Number 6 (2013
[10] Kang J-Y, Gaudiot J-L. A simple high speed multiplier design. IEEE Trans on Comput. 2006
[11] Swee KLS, HaiHiung L. Performance comparision review of radix-based multiplier designs International Conference on Intelligent and Advanced Systems; Kuala Lumpur. 2012.
[12] Dr. Seetaiah Kilaru, Hari Kishore K, Sravani T, Anvesh Chowdary L, Balaji T “Review and Analysis of Promising Technologies with Respect to fifth Generation Networks”, 2014 First International Conference on Networks & Soft Computing, ISSN:978-1-4799-3486-7/14,pp.270-273,August2014.
[13] Meka Bharadwaj, Hari Kishore "Enhanced Launch-Off-Capture Testing Using BIST Designs” Journal of Engineering and Applied Sciences, ISSN No: 1816-949X, Vol No.12, Issue No.3, page: 636-643, April 2017.
[14] N Bala Dastagiri, Kakarla Hari Kishore "Reduction of Kickback Noise in Latched Comparators for Cardiac IMDs” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.43, Page: 1-6, November 2016.
[15] A.Murali, K Hari Kishore, D Venkat Reddy "Integrating FPGAs with Trigger Circuitry Core System Insertions for Observability in Debugging Process” Journal of Engineering and Applied Sciences, ISSN No: 1816-949X, Vol No.11, Issue No.12, page: 2643-2650, December 2016.
[16] Mahesh Mudavath, K Hari Kishore "Design of CMOS RF Front-End of Low Noise Amplifier for LTE System Applications Integrating FPGAs” Asian Journal of Information Technology, ISSN No: 1682-3915, Vol No.15, Issue No.20, page: 4040-4047, December 2016.
[17] P Bala Gopal, K Hari Kishore, B.Praveen Kittu “An FPGA Implementation of On Chip UART Testing with BIST Techniques”, International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 10, Number 14 , pp. 34047-34051, August 2015.
[18] S Nazeer Hussain, K Hari Kishore "Computational Optimization of Placement and Routing using Genetic Algorithm” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.47, page: 1-4, December 2016.
[19] N Bala Gopal, K Hari Kishore "Analysis of Low Power Low Kickback Noise in Dynamic Comparators in Pacemakers” Indian Journal of Science and Technology, ISSN No: 0974-6846, Vol No.9, Issue No.44, page: 1-4, November 2016.
[20] T. Padmapriya and V. Saminadan, “Improving Throughput for Downlink Multi user MIMO-LTE Advanced Networks using SINR approximation and Hierarchical CSI feedback”, International Journal of Mobile Design Network and Innovation-Inderscience Publisher, ISSN : 1744-2850 vol. 6, no.1, pp. 14-23, May 2015.
[21] S.V.Manikanthan and K.srividhya "An Android based secure access control using ARM and cloud computing", Published in: Electronics and Communication Systems (ICECS), 2015 2nd International Conference on 26-27 Feb. 2015,Publisher: IEEE,DOI: 10.1109/ECS.2015.7124833.
[22] Rajesh, M., and J. M. Gnanasekar. "Path observation-based physical routing protocol for wireless ad hoc networks." International Journal of Wireless and Mobile Computing 11.3 (2016): 244-257.