International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 3 Issue: 6

ISSN: 2321-8169
4227 - 4230

Improved Performance of Area and Delay for Radix 4 and Radix 8 Multiplier

Anuradha Kumari[1],Nidhish Tiwari[2]
[1]M.Tech Scholar, JaganNath University, Jaipur, Rajasthan, India
[2] Associate Professor, JaganNath University, Jaipur, Rajasthan, India

Abstract :- Booth multiplier algorithm provides a basic platform for the new advanced fast with higher performance multiplier. It is little work performed on disposal of the negative partial products .Booth multiplier algorithm provide better encoding during the multiplication first step. In this paper is working for Radix 4 and Radix 8 multiplication. We are improving the results of LUT's and Delay by use pipelining.

Keyword :- *Radix 4, Radix 8, LUT .*

*****

1. INTRODUCTION

Booth Algorithm is representing by adding (unsigned binary numbers) with two predefined values A and S to a product P. In next step We will perform rightward arithmetic shift on p . Let m and r be the multiplicand and multiplier and x , y represent the number of bits in m , r .

1:- find the values of A and S , and then initial value of P.the total number of length will be equal to (X+Y+1).

- A:- fill the MSB bits with the value of m and remaining (y+1) with 0 .
- S:- MSB bit with the value of (-m)in two's complement notation . Fill the remaining(Y+1) with 0 .
- P: MSB x bits fill with zeros, right of this append the values of r . Fill the least significant bit with 0.

2:- Now we will check the last two bits values of P

- If the values are "01", then P+A , ignore any carry
- If the value are "10",then P+S , ignore any carry
- If the value is "00" , Use P directly in the next step
- If the value is "11" , use P directly in the next step .

3:- Arithmetic shift :- in the next step the one bit right shift will be performed .

4:- repeat the 2nd and 3rd step for y times .

5:- Drop the least significant bit from P . This will be the product of m and r .

Firstly Recode each and every 1 in multiplier as "+2-1" . Now Converts sequences of 1 to 10…0(-1) . Might reduce the number of 1's.

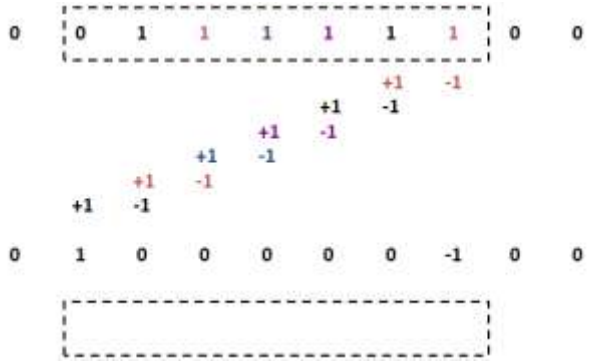

Figure 1:- Modified booth algorithms

2. ENCODING OF BOOTH MULTIPLIER

If you are using the last row in multiplication, you should get exactly the same result which was in the first row.

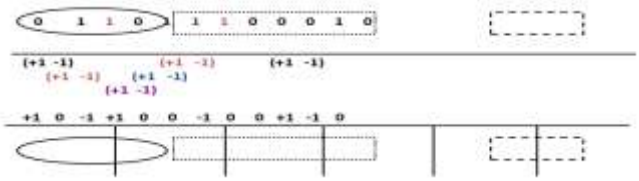

Figure 2:- Encoding of booth multiplier

3. MULTIPLICATION EXAMPLE

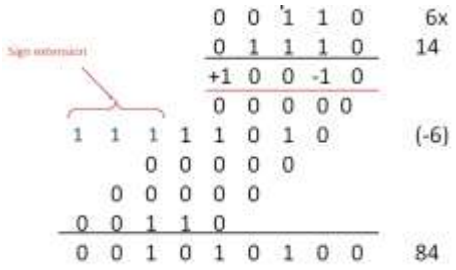

Figure 3:- Multiplication process

 ISSN: 2321-8169
4227 - 4230

4. 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

Table 1:- Booth recoding table

It must be able to add multiplicand times –2, -1, 0, 1 and 2 Since Booth recoding got rid of 3's, generating partial products is not that hard (shifting and negating).

5. RADIX 8 MULTIPLICATION

In the Radix 8 multiplication all the things are same but we will do pairing of 4 bit for radix 8. All the process will be same for radix algorithm.

6. PROBLEM STATEMENT

All the ic's of the electronics fields are improving in form of power consumption, delay, Area. These three points are the main factor. According to base paper the output of radix 4 is 28.994 ns is delay for the radix 4 and the Look up table (LUT's) are using 250. For the Radix 8 the delay is 24.650 and the area is (in the form of LUT's) 230 . We find that the main program is design only for the fix number of bits. If you want to increase or reduce the number of bits so have to change the complete program of multiplication.

7. PROPOSED METHODLOGY AND SOLUTION

For improve the performance of the Radix4 and Radix 8 multiplication, we used pipelining. The data is flowing through the pipelining so that the results are getting improved. After introduce the pipeline the Radix 4 delay and power get reduced and also reduces for the Radix 8 .Pipelining is the way in which data will flow in the form of parallel. In our project when the data is getting transfer from one connection to another connection then the data is transferring by parallel. so we introduce pipelining there for improve the results.

8. RESULTS

8.1 RESULTS FOR RADIX 4

For radix 4 the results are showing in form of RTL, Delay and Area.

8.1.1 RTL

Figure 4:- IC view of Radix4

According to image number 4 the RTL view of the Radix 4 is showing. There are two inputs x,y of 8 bit and one is output of 16 bit .

According to image number 5 the internal RTL view is showing of the IC . In this all the unsigned adder, multiplexer are using for design the circuit. D flipf lop is using for forward the input at the output at every rising edge condition. When the enable gets the value of 1 , the input goes at the output .

Finally unsigned addition is using for the addition of the outputs of the radix4,8 subsequent .

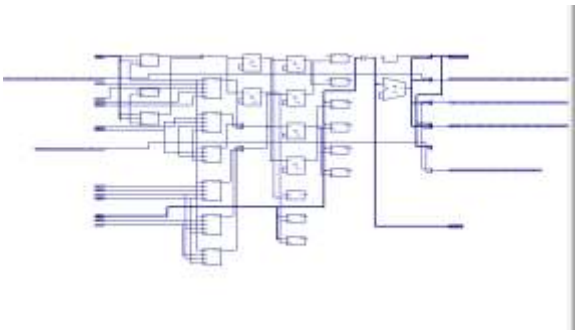

Figure 5:- RTL View

8.1.2 SYNTHESIS REPORT FOR RADIX 4

Name	Output Result
Number of Slices	88
Number of Slices Flip Flop	44
Number of 4 Input LUT's	154
Number of Bounded IOBs	33

Table 2:- Output for Radix 4

8.1.3. DELAY

The output delay for complete circuit is coming 12.155ns. It is the final delay which we get for radix 4 Design. In this 10.987 ns is getting by logic and 1.168 is getting by routing .That means 90.4% is for the logic and 9.6% is for routing .

 ISSN: 2321-8169
4227 - 4230

```
                                        Gate    Net
Cell:in->out      fanout  Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
  LDCP:G->Q             6  0.727   0.688  IP_SHIFT<1>_0 (IP_SHIFT<1>_0)
  LUT2:I1->O            1  0.720   0.000  Madd__n0088_inst_lut2_111 (Madd__n0088_inst_lut2_11)
  MUXCY:S->O            1  0.629   0.000  Madd__n0088_inst_cy_11 (Madd__n0088_inst_cy_11)
  MUXCY:CI->O           1  0.090   0.000  Madd__n0088_inst_cy_12 (Madd__n0088_inst_cy_12)
  MUXCY:CI->O           1  0.090   0.000  Madd__n0088_inst_cy_13 (Madd__n0088_inst_cy_13)
  MUXCY:CI->O           1  0.090   0.000  Madd__n0088_inst_cy_14 (Madd__n0088_inst_cy_14)
  XORCY:CI->O           1  0.899   0.240  Madd__n0088_inst_sum_15 (_n0088<15>)
  LUT2:I1->O            1  0.720   0.000  Madd_MULTIPLICATION_inst_lut2_151 (Madd_MULTIPLICATION_inst_lut2
  MUXCY:S->O            0  0.629   0.000  Madd_MULTIPLICATION_inst_cy_15 (Madd_MULTIPLICATION_inst_cy_15)
  XORCY:CI->O           1  0.899   0.240  Madd_MULTIPLICATION_inst_sum_16 (MULTIPLICATION_0_OBUF)
  OBUF:I->O                 5.412          MULTIPLICATION_0_OBUF (MULTIPLICATION<0>)
----------------------------------------
Total                      12.155ns (10.987ns logic, 1.168ns route)
                                    (90.4% logic, 9.6% route)
```

Figure 6:- Delay for Radix 4

8.1.4 OUTPUT WAVEFORM

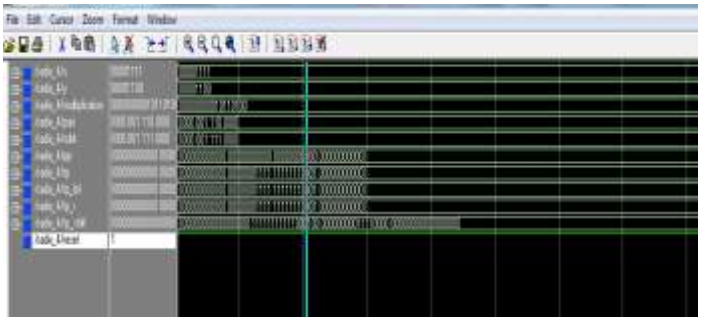

Figure 7:- Output waveform

According to image number 4.4 when the input is 00001111 and the second input is 00001100 then the output is 0000000010110100. In this we are making 4 pairing.

8.1.5 FLOORPLANER

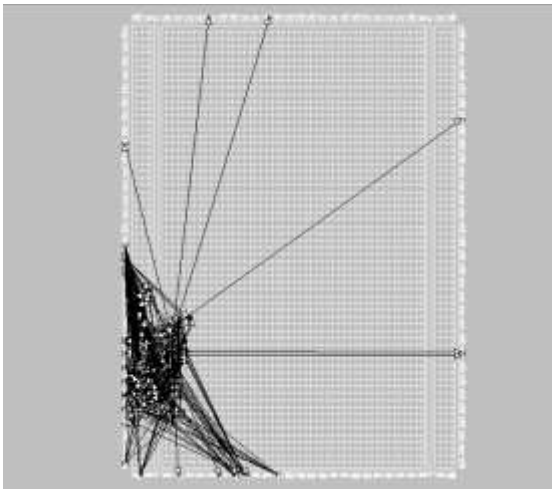

Figure 8 :- Floorplaner

8.2 RESULTS FOR RADIX 8

For radix 8 the results are showing in form of RTL, Delay and Area.

8.2.1. RTL

Figure 9:- IC view of Radix8

According to image number 9 the RTL view of the Radix 8 is showing. There are two inputs x,y of 8 bit and one is output of 16 bit .

According to image number 10 the internal RTL view is showing of the IC. In this all the unsigned adder, multiplexer are using for design the circuit. D flipf lop is using for forward the input at the output at every rising edge condition. When the enable gets the value of 1 , the input goes at the output .

Finally unsigned addition is using for the addition of the outputs of the radix4,8 subsequent .

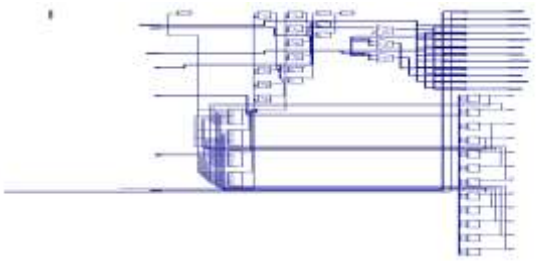

Figure 10:- RTL View

8.1.2 SYNTHESIS REPORT FOR RADIX 8

Name	Output Result
Number of Slices	123
Number of Slices Flip Flop	34
Number of 4 Input LUT's	221
Number of Bounded IOBs	34

Table 1:- Output for Radix 8

8.1.3. DELAY

The output delay for complete circuit is coming 13.565 ns. It is the final delay which we get for radix 4 Design. In this 11.752 ns is getting by logic and 1.813 ns is getting by routing .That means 86.6% is for the logic and 13.4% is for routing .

ISSN: 2321-8169
4227 - 4230

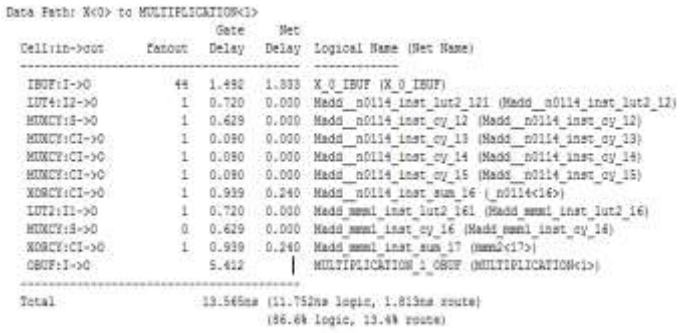

Figure 11:- Delay for Radix 8

8.1.4 OUTPUT WAVEFORM

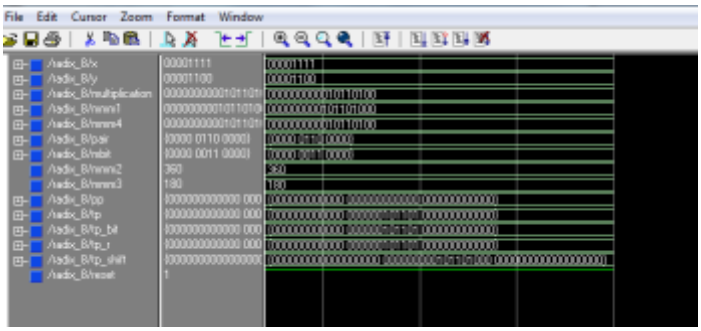

Figure 12:- Output waveform

According to image number 4.9 when the input is 00001111 and the second input is 00001100 then the output is 0000000010110100. In this we are making 4 pairing.

8.1.5 FLOORPLANER

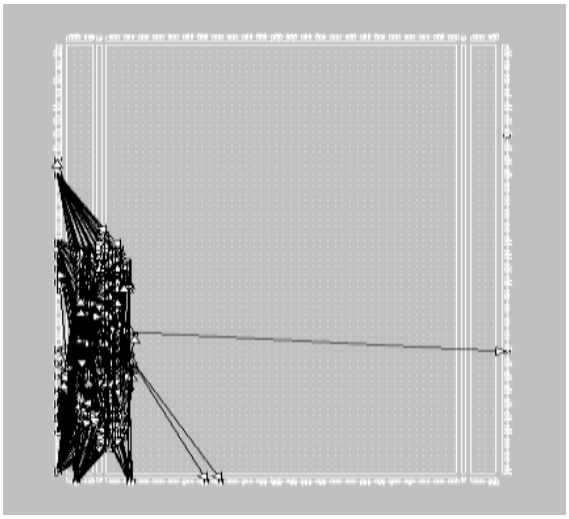

Figure 13:- Floor planer

9. CONCLUSION

As we discussed in previous section that in digital communication integrated circuit contains an important role. All the digital PCB is designed by help of IC's. The IC contains the important role in electronics world. We have to design the IC with low power, delay and area, so that it can be efficient and produce low heat in the electronics equipment. We are reducing the delay, area for this thesis. The output delay for radix 4 is 12.155ns and Look up tables is 154. For the Radix 8 delay is 13.565 ns and look up table is 221. The results of delay and Look up table are improving from the attached base paper.

References

[1] R, Gonzalez, B. M. Gordon, and M. A. Horowitz. Supply and threshold voltage scaling for low power CMOS, IEEE Journal of Solid-State Circuits, 32: 1210-1216, Aug 1997.

[2] P. Korkmaz, Bilge E.S. Akgul and K. Palem. "Characterizing the Behavior of a Probabilistic CMOS Switch through Analytical Models and its Verification through Simulations, CREST Technical Report no. TR- 05-08-01, Aug 2005.

[3] Zhu N., Goh W.L. , Zhang W. J., Yeo K. S., and Kong Z.H. , "Design of Low-Power High Speed Trancation-Error-Toleant Adder and Its Application in Digital Signal Processing, TVLSI-00206-2008.

[4] Melvin A. Breuer and Haiyang Zhu, "Error-tolerance and multi-media," In Proc of the 2006 International Information Hiding and Multimedia Signal Processing, 2006.

[5] I. S. Chong and A. Oretega, "Hardware testing for error tolerant multimedia compression based on linear transforms," Defect and Fault Tolerance in VLSI Systems Symp., 2005.

[6] Tso B. J., Shen F. H. , "Low-Error Carry Free-Width Multipliers with Low-Cost Compensation Circuits," IEEE Transactions on Circuits and Systems, vol 52, No. 6, Jun 2005.

[7] S. S. Kidambi, F. EI-Guibaly, Senior Member and A. Antoniou, Fellow IEEE, "Area-Efficient Multipliers for Digital Signal Processing Applications," IEEE Trans. On Circuits and Systems-IL vol. 43, no. 2, Feb 1996.

[8] M. J. Schutle and E. E. Swartzlander Jr. , "Truncated multiplication with correction constant," VLSI Signal Processing, vol. 6, pp. 388-396,1993.

[9] J.M. Jou, S.R. Kuang, and R.D. Chen, "Design of low-error-fixed-width multipliers for DSP applications, " IEEE Trans, Circuits System If, Analog and Digital Signal Processing, vol. 46, no. 6, pp. 836-842, lun 1999.