HIGH PERFORMANCE PIPELINED SIGNED 8*8 -BIT MULTIPLIER USING RADIX-4,8 MODIFIED BOOTH ALGORITHM

Prateek Mudgal[1],Rakesh Jain[2],
[1]M.Tech Scholar, Suresh Gyan Vihar University, Jaipur, Rajasthan, India
[2] Professor, Suresh Gyan Vihar 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 provides better encoding during the multiplication first step. Radix 4 and Radix 8 multiplication. We are improving the results of LUT's and Delay by use pipelining. 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.

Keywords- Radix-32, Booth Encoder, 3:2 Compressor, 4:2 Compressor ,Wallace Tree .

I. INTRODUCTION

Today we know that multiplier is using in every basic circuit. Today all the ALU system is based on a multipliers. Complete arithmetic Logical part is based on a multiplier and if multiplier is consuming so much delay then the entire product which is based on a multiplier is fail due to fail of multiplier.

If multiplier have low speed then it will works slowly. Regarding this if our function is working in 2 second then it will also take some delay .Then its output will be 2second+ delay. That delay may greater then to basic performing delay.

In VLSI speed of any IC is depend on power consumption, Area, delay. Some we have a complex circuit and at that time we will get increment in delay and power consumption. Power consumption is also a main power factor If we reduce the power factor of an IC the it is showing that our product battery life is good.

Today everybody is using calculators and CPU. Every company is working for low power consumption circuit so that they can deliver more long life battery as comparison to other company.

If our Multiplier power consumption will be increase then heat dissipation will be increase. So it will increase leakage current. So this multiplier will be used in many ALU circuits. then all the product of this ALU have a low battery life .If we are using multiplier operations for memory allocation of mobile phone then battery of that mobile phone will not be long life because the heat dissipation , power consumption is greater than the normal range .

According to Moore low in every 18 month the transistors of any IC will be doubled which is using in a IC .According to moore low after every 18 month we will get a new IC in which we will find a ore number of transistor according to previous .Now we have a question that what is benefit from this increase in number of transistor. If number of transistor will increase that means you have to increase its functionality. Right now our IC can work more and fast work as comparison to previous IC.

We are reducing the delay of multiplier by use Booth multiplier Radix 4 and radix 8. To create fast or high speed multiplier we need some fastest adding application, we have already reduced no. of partial product. So now we have less no. of operands for adding, an efficient method of addition (Parallel prefix Adder) will be implement in partial product addition. To Create area efficient multiplier we need some effective algorithm for 2's complement process in signed no. we have already discussed about that MBE work for signed and unsigned no.

In this thesis we will work at delay and area consumption part. Basically delay will depends on a total number of Lut's . If we will reduce the total number of LUT the n it will automatically reduce the delay.

Actually if speed increase will found in IC then the performance of any circuit in which that IC is using also increase. We are using a Booth multiplier with radix 4 for better performance .

We are using Xilinx project navigator 6.1 for design and verification of VHDL code . For wave form generation we are using modelsim 5.4 . We are taking all the parameters at FPGA XC3S50E kit .

Power consumption is the main factor for design a goes performance IC so we are a software which will describe the power consumption of IC .

II. BOOTH MULTIPLICATION ALGORITHM

Booth Algorithm is representing by adding (unsigned binary numbers) with 2 defined values A, S and the product of that 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 .

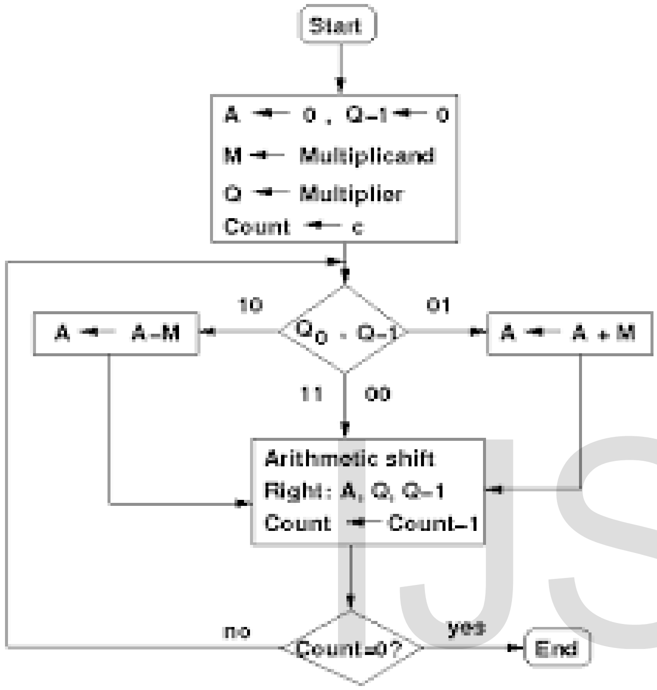
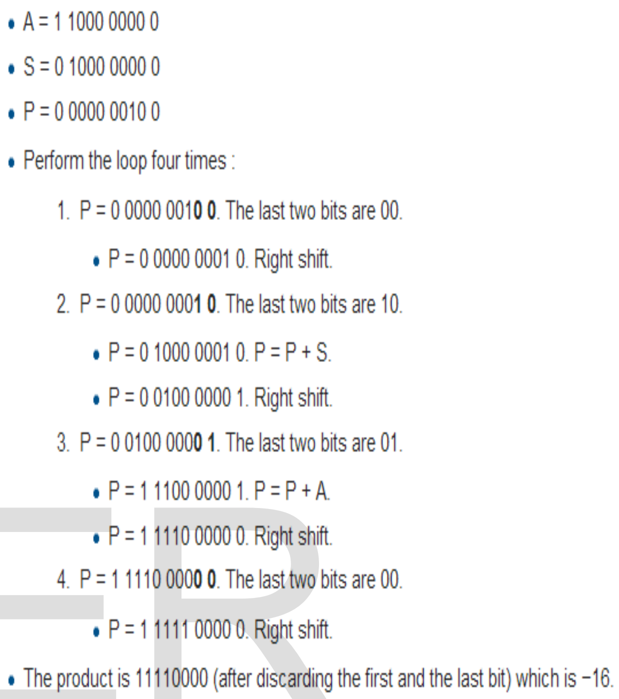
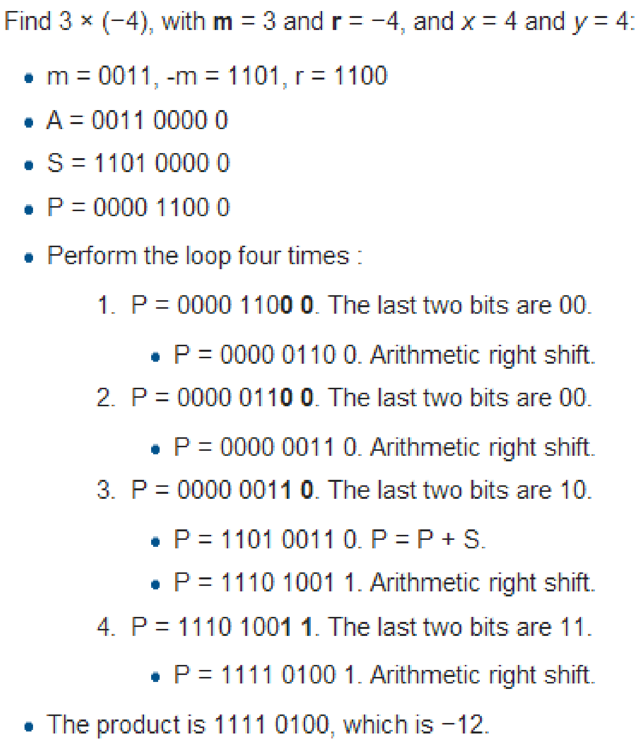


Figure 1:- Flow chart of booth algorithm



- A = 1 1000 0000 0
- S = 0 1000 0000 0
- P = 0 0000 0010 0
- Perform the loop four times :
  1. P = 0 0000 001**0** **0**. The last two bits are 00.
     - P = 0 0000 0001 0. Right shift.
  2. P = 0 0000 000**1** **0**. The last two bits are 10.
     - P = 0 1000 0001 0. P = P + S.
     - P = 0 0100 0000 1. Right shift.
  3. P = 0 0100 000**0** **1**. The last two bits are 01.
     - P = 1 1100 0000 1. P = P + A.
     - P = 1 1110 0000 0. Right shift.
  4. P = 1 1110 000**0** **0**. The last two bits are 00.
     - P = 1 1111 0000 0. Right shift.
- The product is 11110000 (after discarding the first and the last bit) which is −16.



Find 3 × (−4), with **m** = 3 and **r** = −4, and x = 4 and y = 4:

- m = 0011, -m = 1101, r = 1100
- A = 0011 0000 0
- S = 1101 0000 0
- P = 0000 1100 0
- Perform the loop four times :
  1. P = 0000 110**0 0**. The last two bits are 00.
     - P = 0000 0110 0. Arithmetic right shift.
  2. P = 0000 011**0 0**. The last two bits are 00.
     - P = 0000 0011 0. Arithmetic right shift.
  3. P = 0000 001**1 0**. The last two bits are 10.
     - P = 1101 0011 0. P = P + S.
     - P = 1110 1001 1. Arithmetic right shift.
  4. P = 1110 100**1 1**. The last two bits are 11.
     - P = 1111 0100 1. Arithmetic right shift.
- The product is 1111 0100, which is −12.

III. MODIFIED BOOTH MULTIPLIER (Radix 4)

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.

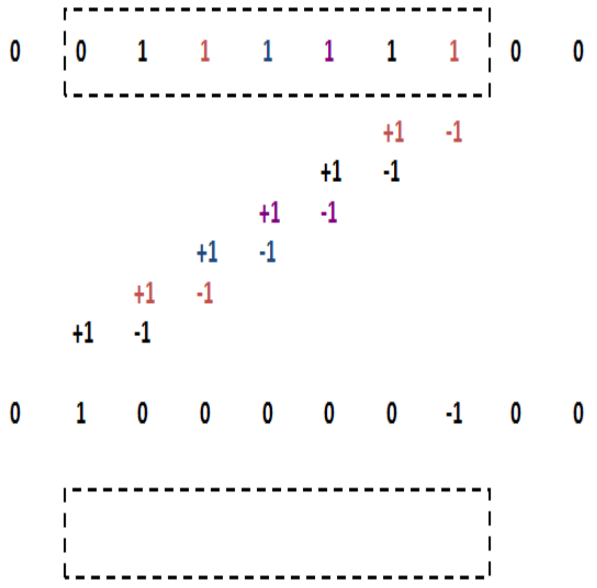


Fig 2:- modified Booth multiplier

IV. 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.

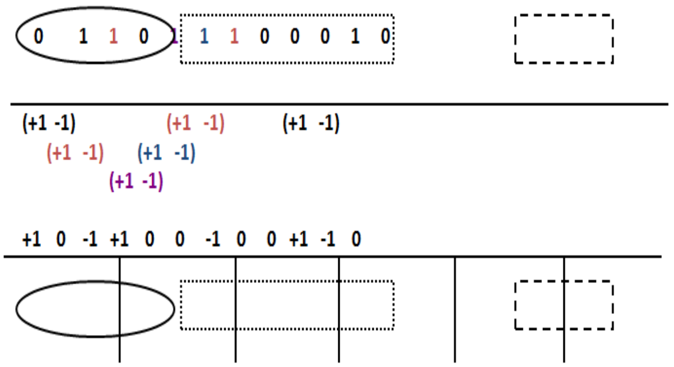


Fig 3 :- Encoding of Booth Multiplier

V. 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.

VI. PROBLEM STATEMENT

FIRST ISSUE

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 .

SECOND ISSUE

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.

VII. 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 ,

VIII. RESULTS

RESULTS FOR RADIX 4

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

According to image 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 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 .

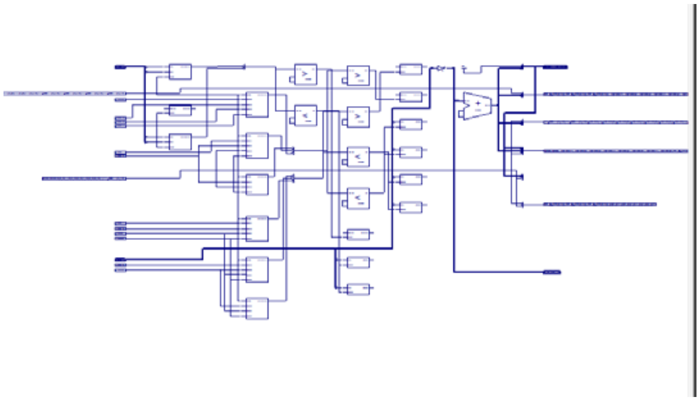


Figure 4:- RTL View

SYNTHESIS REPORT FOR RADIX 4

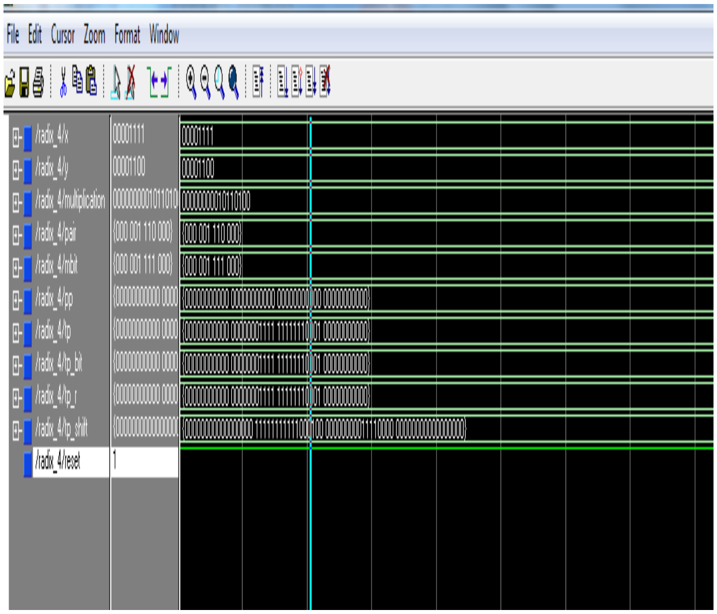
Name	Output Result
Number of Slices	88
Number of Slices Flip Flop	44
Number of 4 Input LUT's	154
Number of Bounded IOBs	33

Table 1:- Output for Radix 4

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 .

OUTPUT WAVEFORM



```
/radix_4/x               00001111            00001111
/radix_4/y               00001100            00001100
/radix_4/multiplication  000000001011010     0000000010110100
/radix_4/pair            {000 001 110 000}   {000 001 110 000}
/radix_4/mbit            {000 001 111 000}   {000 001 111 000}
/radix_4/pp              {0000000000 0000    {0000000000 0000000000 0000000000 0000000000}
/radix_4/tp              {0000000000 0000    {0000000000 0000001111 1111111001 0000000000}
/radix_4/tp_bit          {0000000000 0000    {0000000000 0000001111 1111111001 0000000000}
/radix_4/tp_r            {0000000000 0000    {0000000000 0000001111 1111111001 0000000000}
/radix_4/tp_shift        {00000000000000     {0000000000000000 1111111111000000 0000000001110000 0000000000000000}
/radix_4/reset           1
```

Figure 6:- Output waveform

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

RESULTS FOR RADIX 8

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

RTL

According to image 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 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 .

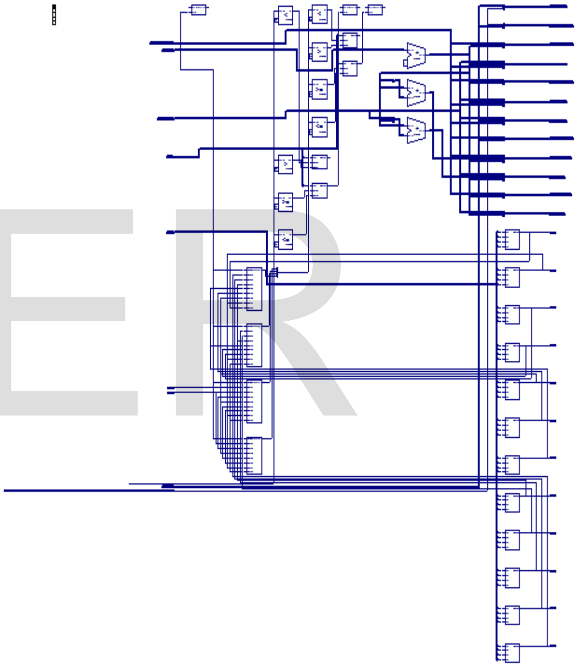


Figure 7:- RTL View

SYNTHESIS REPORT FOR RADIX 8

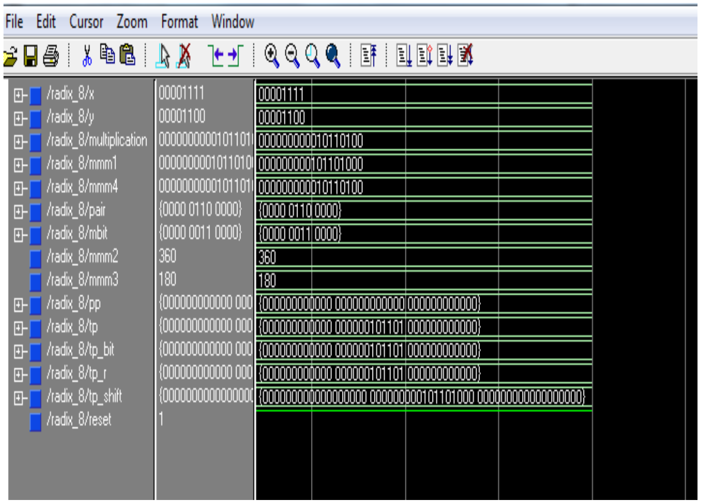
Name	Output Result
Number of Slices	123
Number of Slices Flip Flop	34
Number of 4 Input LUT's	221
Number of Bounded IOBs	34
Delay	13.565

Table2:- Output for Radix 8

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 .

4.2.4 OUTPUT WAVEFORM



```
/radix_8/x               00001111            00001111
/radix_8/y               00001100            00001100
/radix_8/multiplication  000000000010110100  0000000010110100
/radix_8/mmm1            000000000101101000  0000000101101000
/radix_8/mmm4            000000000010110100  0000000010110100
/radix_8/pair            {0000 0110 0000}    {0000 0110 0000}
/radix_8/mbit            {0000 0011 0000}    {0000 0011 0000}
/radix_8/mmm2            360                 360
/radix_8/mmm3            180                 180
/radix_8/pp              {00000000000 000    {00000000000 00000000000 00000000000}
/radix_8/tp              {00000000000 000    {00000000000 00000101101 00000000000}
/radix_8/tp_bit          {00000000000 000    {00000000000 00000101101 00000000000}
/radix_8/tp_r            {00000000000 000    {00000000000 00000101101 00000000000}
/radix_8/tp_shift        {00000000000000000  {0000000000000000 0000000101101000 0000000000000000}
/radix_8/reset           1
```

Figure 8:- Output waveform

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

4.3 COMPARISON TABLE

	RADIX 4 Base paper	Radix 4 Proposed	Radix 8 Base paper	Radix 8 Proposed
AREA(LUT's)	250	154	230	221
DELAY(ns)	28.994	12.155	24.650	13.565
Power (w)	0.034	0.026	0.032	0.031

Table 4.3:- Comparison table

IX. CONCLUSION AND FUTURE SCOPE

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.

In the future delay, area can be reduce by apply multilevel stage pipelining. As the results can be able to improve more from the given results of delay and area for Radix 4 and Radix 8.

References

[1] Chen ping-hua and ZHAO Jaun, "high-speed parallel 32x32-bit multiplier Using Radix-16 Booth Encoder" ,2009 IEEE proceeding of 2009 Third International Symposium on Intelligent Information Technology Application Workshop, IITAW 2009, 406-409.

[2] Chen ping-hua and ZHAO Jaun, XIE Guo-bo,LI Yi-jun, "An improved 32-bit Carry-Look ahead Adder with Conditional Carry-selection"[C].Proceeding of 2009 4th International conference on Computer Science & Education, ICCSE 2009: 1911-1913.

[3] Weinan Ma, Shuguo Li, "A New High Compression Compressor for Large Multiplier", Institute of Microelectronics, Tsinghua University, Beijing 100084,P.R. China, 2008 IEEE.

[4] LIU Qiang, WANG Rongsheng, "High-speed Parallel 32x32-b Multiplier Design Using Radix-16 Booth Encoder". Computer Engineering [J],2005,31:200-202

[5] B.Parhami, " Computer Arithmetic Algorithm and Hardware designs," Oxford University Press, 2000.

[6] Ki-seon Cho,Jong-on Park,Jin-seok Hong, Goang_seog Choi, ''54x54-bit Radix-4 Multiplier based on Modified Booth Algorithm" proceeding GLSVLSI 03, April 28-29,Washington, DC, USA : 233-236

[7] I. Korean, "Computer Arithmetic Algorithms," 2nd Edition. Prentice Hall,2001

[6] Lakshamanan, Masuri Othman and Mohamad Alaudin Mohd.Ali, Signal Processing Group, "High Performance Multiplier using Wallace-Booth Algorithm", 2002 IEEE proceeding ICSE2002 proc.2002, Penang Malaysia: 433-436.

[9] Dr. D.S.Dawoud, "Modified Booth Algorithm for Higher Radix fixed- point multiplication",1997 IEEE.

[10] Ahmed Elhossini, Ali Rasid, and Mohammed K.Refai, "A 16x16-bit Modified radix 16 Booth Encoder parallel Multiplier", Al_azahar University Engineering Journal, AUEJ Vol.8,No.7,Jan.2005