# A Novel Parallel Multiplier for 2's Complement Numbers Using Booth's Recoding Algorithm

Chandrashekhar T. Kukade
Dept. of Electronics Engineering
VNIT, Nagpur, Maharashtra, India
shekharkukade@ece.vnit.ac.in

Raghavendra B. Deshmukh
Dept. of Electronics Engineering
VNIT, Nagpur, Maharashtra, India
rbdeshmukh@ece.vnit.ac.in

R. M. Patrikar
Dept. of Electronics and Computer Science
VNIT, Nagpur, Maharashtra, India
rajendra@computer.org

*Abstract*

**A novel architecture of parallel multiplier using modified Booth's recoding unit for 2's complement numbers is presented in this paper. The basic Booth's recoding algorithm requires add and shift operations for multiplication, these steps makes this multiplier sequential. Parallel multiplication can be achieved using Booth's recoding algorithm and simple Brown's array of adders, but it requires more number of adders to get correct output. Other parallel multiplication techniques are available using Booth's recoding algorithm. However, these array multiplier also requires add, shift and extra control unit. The proposed design has two major features; first is modified Booth's recoding unit which produces partial products second is modified array of adders. Modified array of adder block designed, which uses less number of adders than conventional Booth's recoding multiplier. Multiplexers are basic unit used for Booth's recoding unit and synthesis has been carried out using 180 nm technology. The proposed design uses less power than conventional Booth's recoding 2's complement parallel multiplier.**

*Keyword:* **2's complement parallel multiplier, Booth's recoding algorithm**

## I. INTRODUCTION

Multipliers are fundamental blocks in today's Digital signal processing units. Several multiplication algorithms are used to achieve multiplication, some of them are sequential multiplication and others are parallel multiplication. Parallel multiplication is always advantageous over sequential multiplications. Typically, binary multiplication can be done by Brown's array multiplier [1]. However, it is not suitable for 2's complements numbers. So, to achieve parallel multiplication, Baugh-Wooly's 2's complement multiplication algorithm is widely used [1,2]. Booth's algorithm [3,4] is also used frequently for 2's complement numbers. Booth's algorithm uses radix recoding to achieve high speed. Increase in radix produces reduced number of partial products. Higher radix recoding has significant use in high speed digital arithmetic. The basic multipliers based on booth algorithm are sequential multipliers. However, Booth's array multiplication [3] is another method to achieve parallel multiplication. These types of array multipliers consist of a basic unit of add, subtract and shift with controlled input. It also requires small controller to control all the operation in the multiplier depending on input vectors. Lots of work is being done on parallel multipliers using Booth's recoding algorithm [5 – 9].

In this paper, a multiplexer based modified Booth's recoding circuit has been designed to generate efficient partial products. These partial products always have larger number of bits than the input number of bits. This width of partial product usually depends on radix scheme used for recoding. These generated partial products are added using novel array of adders to achieve parallel multiplier output. This scheme uses less number of gates so this circuit consumes less power and area.

## II. BASIC BOOTH'S RECODING USING RADIX 2 AND RADIX 4

Booth has proposed radix algorithms for high speed multiplication which reduces partial products for multiplication. It is based on the fact that partial product can be generated for group of consecutive zeros & ones which is called as Booth's recoding. So, two algorithms called radix 2 and radix 4, are explained and summarized as shown in table. 1. & table .2 respectively.

TABLE I.
BOOTH'S RADIX 2 RECODING

| $X_i$ | $X_{i-1}$ | Y | Comments | Explanation |
|---|---|---|---|---|
| 0 | 0 | 0 | Strings of zeros | Stings of zeros shift only |
| 0 | 1 | 1.A | Beginning of 1's | Add and shift |
| 1 | 0 | -1.A | End of 1's | Add and shift |
| 1 | 1 | 0 | Strings of ones | String of ones shift only |

TABLE II
BOOTH'S RADIX RECODING

| $X_{i+1}$ | $X_i$ | $X_{i-1}$ | Y | Comments | Explanation |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | Strings of zeros | Two bit shift only |
| 0 | 0 | 1 | 1.A | End of 1's | Add and two bit shift |
| 0 | 1 | 0 | 1.A | A single 1 | Add and two bit shift |
| 0 | 1 | 1 | 2.A | End of 1's | Add and two bit shift |
| 1 | 0 | 0 | -2.A | Beginning | Add and two bit |

978-1-4799-2102-7/14 $31.00 © 2014 IEEE
DOI 10.1109/ICESC.2014.103

| | | | | | |
|---|---|---|---|---|---|
| | | | | of 1's | shift |
| 1 | 0 | 1 | -1.A | A single 0 | Add and two bit shift |
| 1 | 1 | 0 | -1.A | Beginning of 1's | Add and two bit shift |
| 1 | 1 | 1 | 0 | String of ones | Two bit shift only |

Examples of multiplication using basic Booth's recoding algorithm are explained in [3]. These algorithms are used for sequential multiplication, as it comprises add and shift method. Some of the new advanced techniques are also used to achieve high speed parallel multiplication using Booth's algorithm explained in [5-9]. These designs require lot of control circuitry and more hardware. However, we can achieve parallel multiplication using basic radix-2 Booth's algorithm.

Fig.1 and 2 show basic parallel multiplication schemes using booth's recoding algorithm for multiplication of 4 and 8 bit numbers. In both the figures, at right hand sides of partial products some of bits are encapsulated with triangle that we call as correction bits. As number of input bits increases, these correction bits also increase. So, numbers of adders required to add these bits also increase. This in turn results in more power requirement of the multiplier. This is applicable to both radix 2 and radix4 methods.

```
A=> 2 X=> -5 P=> -10

    0010     <= A
x   1011     <= X
   ----------
    -110-1   <= Xr
   ----------
 1111 1110
 0000000
 000010
 1 1110
 -------------
 11110110 <= P= -10
```

Fig. 1. Parallel multiplication of 4 bits using Booth's radix 2 recoding scheme

```
A=>-5=>     11111011
X=>46=>     00101110
            ----------------
Xr==>       01-1100-10
  ----------------------------
  0000000000000000
  1111111 11111011
  0000000 0000000
  0000000000000000
  0000 0 11111011
  000 00000101
  11 11111011
  0 00000000
  ----------------------------
  1111111100011010
  P=> -230
```

Fig. 2. Parallel multiplication of 8 bits using Booth's radix 2 recoding scheme.

## III. PROPOSED NOVEL PARALLEL MULTIPLIER USING MODIFIED BOOTH'S RECODING UNIT

Parallel multiplication using basic Booth's recoding algorithm is discussed in section 2. Since this technique requires lot of adders as a result it requires more power & area. In proposed multiplier design, we have reduced number of adders required in partial product addition. Mainly correction bits are reduced. This is done without compromising correctness of multiplication of 2's complement numbers. Also, we have used multiplexer based Booth's recoding scheme. The output recoding unit has been changed. This change results in partial products which after recoding are always greater than input bit length by one bit in radix 2 scheme. Also in radix 4 schemes, it is always greater by two bits. These extra added bits work as correction bits to get proper output of multiplier. Also, at hardware realization of Booth's recoding scheme, we can remove extra select line, which is used at the time of recoding. Because of this extra select lines multiplexer size become large. We have observed that if we do not consider this extra bit at the time of hardware realization we can reduces size of one multiplexer. So, in radix 2 LSB decides first partial product. Also, in radix 4 two LSB bits decides first partial product. The working of this novel design has been explained in following sections.

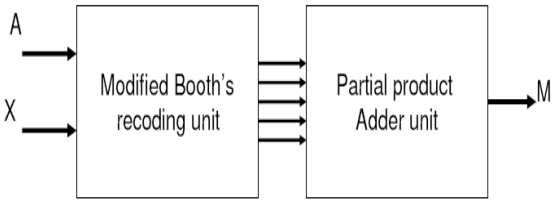


Fig.3 Block diagram of proposed multiplier

## IV. MODIFIED BOOTH'S RECODING UNIT RADIX 2 & RADIX 4

In order to achieve multiplication, Partial products can be generated using Booth's recoding unit. An idea of getting partial product for multiplication is explained in section 2. We have used same approach to get the partial products but recoded output bits are always greater than input bit sequence. These modified schemes are explained for radix 2 and radix 4 methods in following subsections**.**

### A. *RADIX 2 METHOD*

As in table 1 output partial products are added and shifted according to input sequence. Here, we have this recoding unit using multiplexers. Select lines to multiplexer are input bit sequence of multiplier and outputs are according to modified table given in table 3. So, in this scheme, partial products are always one bit more than input vector. If our multiplier is of n bit then partial products are always n+1.

TABLE III.
MODIFIED BOOTH'S RECODING TABLE FOR RADIX 2

| $X_i$ | $X_{i-1}$ | Y | Partial Product Explanation |
|---|---|---|---|
| 0 | 0 | 0 | All 0's |
| 0 | 1 | 1.A | $[A_{(n-1)}, A]$ |
| 1 | 0 | -1.A | $[\overline{A_{(n-1)}}, (-A)]$ |
| 1 | 1 | 0 | All 0's |

This can be explained with simple example.

A= 1100 (-4)
X= 1010 (-6)

So, partial products obtained for these inputs using recoding scheme are shown in table 3.

PP0= 00000
PP1= 00100
PP2= 11100
PP3= 00100

So, the hardware realization for this recoding unit is based on multiplexers and includes 2's complement unit. At the time of recoding, we assume extra '0' before LSB of multiplier, and this LSB & extra '0' bit decides Partial product according to recoding table 1. However, we have observed that at time of hardware realization only LSB is sufficient for getting partial product, because of this multiplexer becomes 2x1 rather than 4x1 and others multiplexers will remain same as per their input select lines depending upon recoding scheme. So, multiplexers become important hardware for Booth's recoding unit. Architecture of modified Booth's recoding unit is shown in fig. 4.

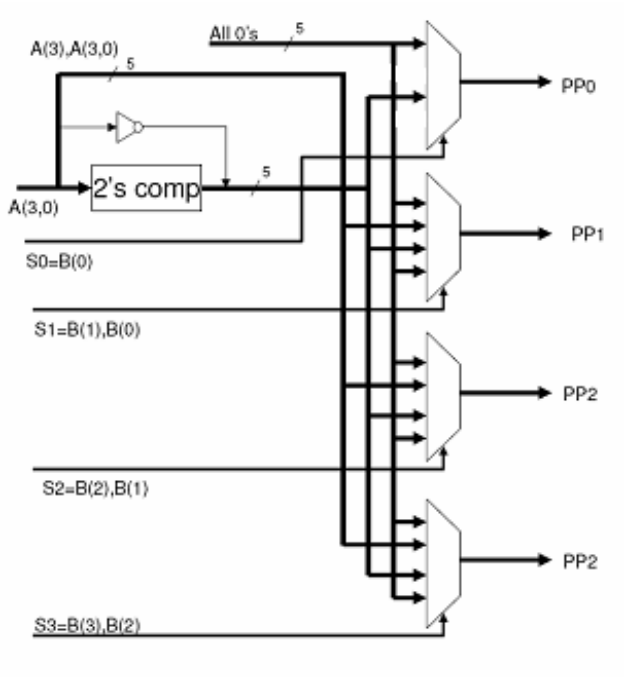


Fig. 4. Architecture of Booth's recoding unit radix 2

In this architecture, select lines for multiplexers are multiplier input bits taken according to recoding scheme given in table 3. Which are similar to General Booth's recoding scheme. We have used 4 multiplexer out of which 3 are 4x1 and one is 2x1. Each input vector is of n+1 of input bits of multiplier. $0^{th}$ and $3^{rd}$ input vector are always zero and $1^{st}$ position input vector is multiplier input(A) appended with extra zero at MSB to increase its width to n+1, it does not change original value. $2^{nd}$ position input vector is taken 2's complement of input(A) with appending inverted MSB bit of input vector. This architecture generates four partial products vector according to table 3.
Similarly, for Booth's radix 4 we have modified recoding scheme explained in following section and table 4.

B. *RADIX 4*
This is also same scheme as explained above that reduces partial products so it is very helpful for fast multiplication of long input bit sequences. But, here partial product which we got from recoding unit is always 2 bit more than input bits. If our inputs are of n bit then partial products are n+2 bit. Recoding scheme is shown in table 4, and architecture of this recoding unit is shown in fig. 5. In this radix scheme, select lines of multiplexers are 3 bits but first multiplexer can be of 2 select lines, which are two LSBs and remaining multiplexers have 3 input select lines.

TABLE IV
MODIFIED BOOTH'S RECODING UNIT FOR RADIX 4

| $X_{i+1}$ | $X_i$ | $X_{i-1}$ | Y | Partial Product Explanation |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | All 0's |
| 0 | 0 | 1 | 1.A | $[A_{(n)}, A_{(n)}, A]$ |
| 0 | 1 | 0 | 1.A | $[A_{(n)}, A_{(n)}, A]$ |
| 0 | 1 | 1 | 2.A | $[A_{(n)}, A, 0]$ |
| 1 | 0 | 0 | -2.A | $[\overline{A_{(n-1)}}, -A, 0]$ |
| 1 | 0 | 1 | -1.A | $[\overline{A_{(n-1)}}, \overline{A_{(n-1)}}, -A]$ |
| 1 | 1 | 0 | -1.A | $[\overline{A_{(n-1)}}, \overline{A_{(n-1)}}, -A]$ |
| 1 | 1 | 1 | 0 | All 0's |

The architecture of Booth's radix 4 recoding scheme is same as explained in section A shown in fig. 5. Only difference is partial products are n+2 in width of input vector according to table 4. Now, all these partial products need to be added properly to get correct output. So, we designed modified partial product adder unit (array of adders) which moreover similar to Brown's array [1].

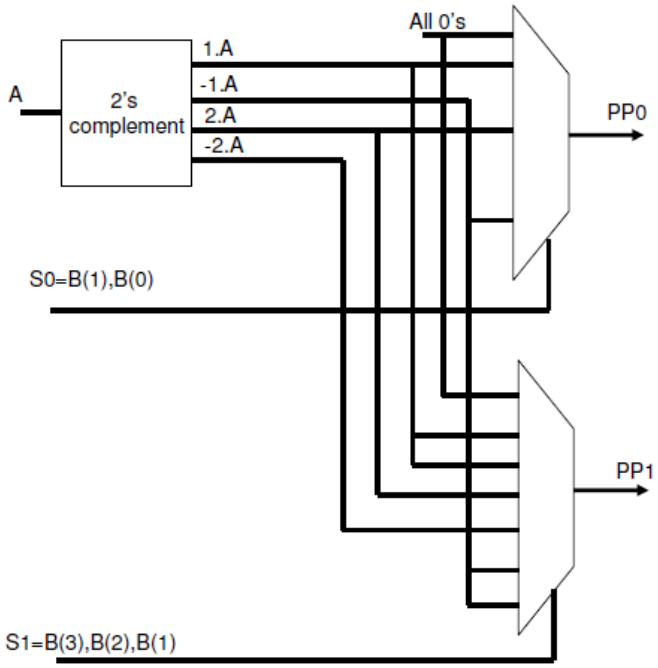


Fig.5 Architecture of booth recoding unit radix 4 for 4 bit inputs.

## V. MODIEFD PARTIAL PRODUCT ADDER(ARRAY OF ADDERS) FOR RADIX 2 & RADIX 4

The partial product obtained form Booth's recoding unit needs to be added properly to get correct output of a multiplication. The addition scheme of partial product is same as Browns array multiplier [1] except for MSBs. MSBs of partial products need to be added carefully. For that, new structure of an adder array is proposed. This modification removes the problem of large number of correction bits, which in turn require more number of adders. This is discussed in section 2 in detail. The proposed partial product adder arrays for 4 bit input sequence using radix 2 and radix 4 algorithms are shown in fig. 6 and fig. 7.

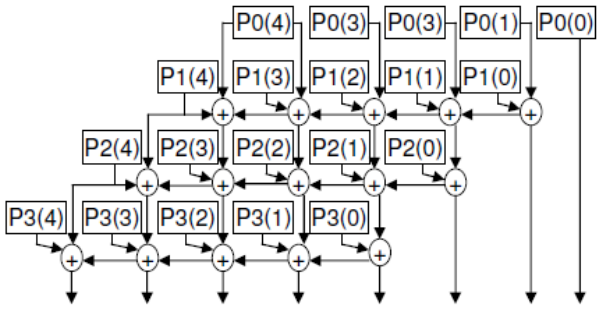


Fig.6 .Partial product adder unit for radix 2 recoding of 4 bit input.

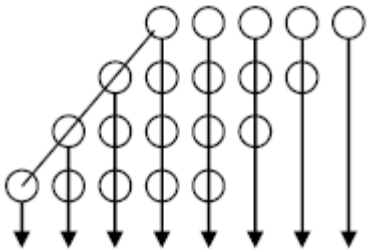

Fig.6.1. addition scheme for radix 2 (4 bit input)

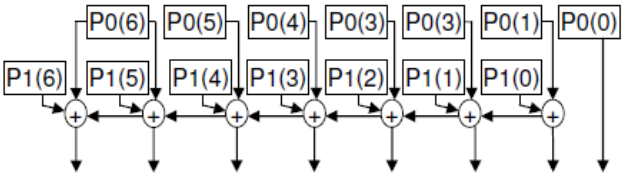


Fig. 7 Partial product adder unit for radix 4 recoding ( 4 bit input )

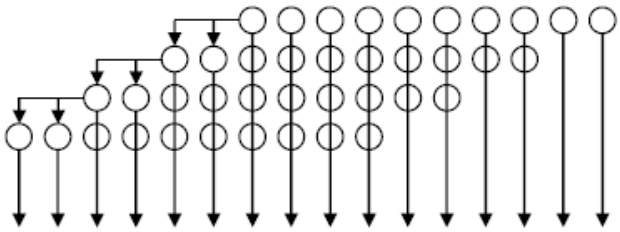

Fig.7.1 addition scheme for radix 4 (8 bit inputs)

Looking at all these addition schemes, we form structure for n bit multiplication which are true for radix2 & radix 4 methods. These recoding and additions schemes are also useful for higher radix scheme.

Let us consider n bit input binary sequence is given to modified radix 2 Booth's recoding unit. So, output partial product can be represented Pij and output multiplication M is r bit.

Now , for radix 2 , i will be n and j will be n+1.

i=n;

j=n+1=m;

So we can form n bit structure for this as shown in fig. 8.

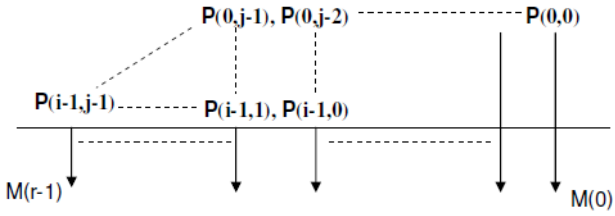


Fig. 8. n bit adder structure for radix 2.

From above n bit adder structure we have formed following equation 1 for partial product addition scheme which is true for radix 2 method.

$$[\sum_{i=0}^{n-1} Pi(j-1)]2^{r-1} + [\sum_{i=0}^{n-2} Pi(j-1) + P(i-1)(j-2)]2^{r-2} + [\sum_{i=n-2}^{n-1}\sum_{j=m-3}^{m-2} Pij + \sum_{i=0}^{n-3} Pi(j-1)]2^{r-3} + ....$$

$$..... + [\sum_{i=0}^{n-1}\sum_{j=1}^{m-1} Pij]2^{r-1} + [\sum_{i=0}^{n-1}\sum_{j=0}^{m-2} Pij]2^{r-i-1} + ..... + [\sum_{i=0}^{n=1}\sum_{j=0}^{m=1} Pij]2^{1} + [P00]2^{0}$$

----------(1)

Similarly, we draw n bit adder structure for radix 4. as shown in fig .9.

If n bit input binary input sequence is given then partial product will be Pij and multiplication output M will be of length r bit.

So i=n+2 and j=(n/2).

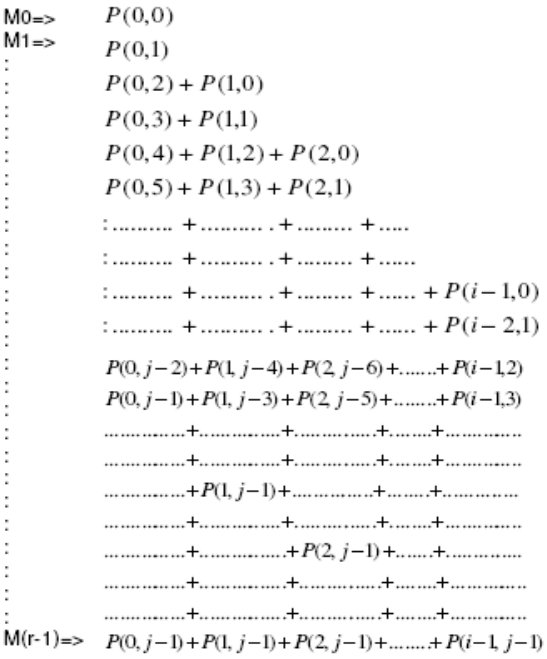


Fig.9 n bit adder structure for radix 4

## VI. RESULTS

The proposed design has been simulated and synthesized with ModelSim and Design compiler. Conventional Booth's multiplier has been taken as ideal multiplier to compare and verify output results of proposed architectures. We found that all the output results of proposed multiplier are same as conventional Booth's multiplier. A synthesis result shows that area required for proposed multiplier is less than Booth's conventional multiplier. Power is calculated using Prime power and results shows that power is reduced in both radix 2 and radix 4 scheme of proposed multiplier. As we move for larger number of input bits power reduces as compared to conventional multiplier. Fig. 10 shows the power comparison graph for radix 2 scheme. Fig. 11 shows the power comparison graph for radix 4 scheme.

## VII. CONCLUSION

From all these observations, we present novel design of Parallel multiplier using modified Booth's recoding unit. This multiplier based on multiplexers and novel partial product adder array. It reduces power and area. This multiplier is suitable for all signed & unsigned input vectors. More power and area is reduced0 for higher radix schemes.

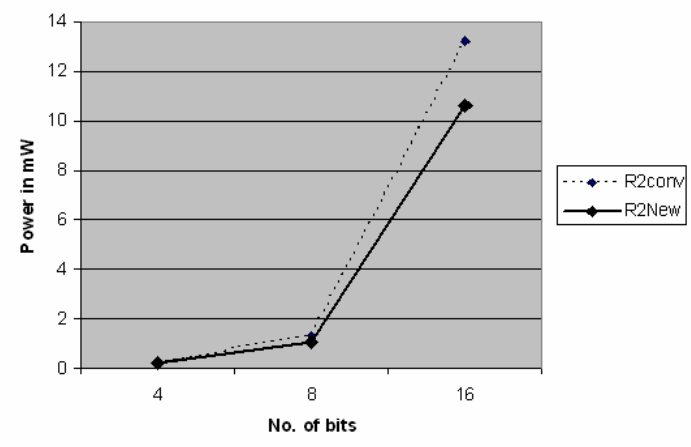


Fig. 10. Power comparison for radix 2

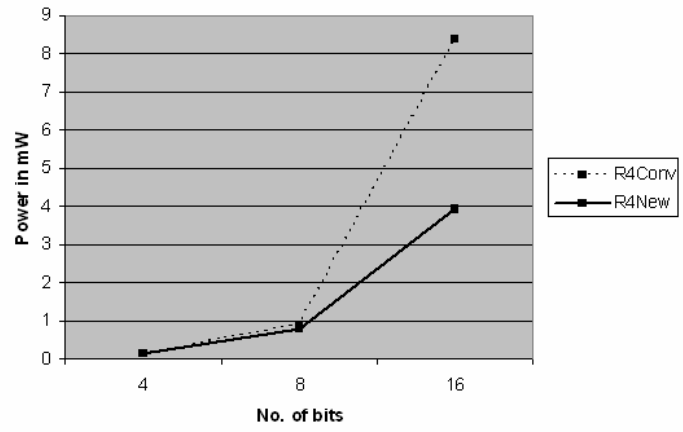


Fig. 11. Power comparison for radix 4

## REFERENCES

[1] D. A Pucknell, K. Eshraghain, *Basic VLSI Design*, Prentice-Hall, ISBN 81-203-0986-3.

[2] C.R.Baugh ,B.A.Wooly, *A two's complement parallel multiplication algorithm* IEEE Transaction on computers, vol, c-22,no.12,December 1973.

[3] Israel Koren, *Computer arithmatics algorithms* A.K.Peters Ltd. ISBN 1568811608.
[4] A.D.Booth, *A signed binary multiplication technique,* Quarterly Journal of Mechanics and Applied mathematics, vol-IV,pt-2-1951.
[5] Rajendra katti, *A Modified Booth Algorithm for High Radix Fixed-point Multiplication,* IEEE Transaction on very large scale integration VLSI systems,vol-2,no.-4 December-1994.
[6] Wen-Chang Yeh, Chein-Wei Jen, *High speed booth encoded parallel multiplier design*, IEEE transaction on computers, vol-49, no-7.July-2000.
[7] Eduardo Costa, Sergio Bampi, Jos´e Monteiro, *A New Architecture for 2's Complement Gray Encoded Array Multiplier,* Proceedings of the 15 th Symposium on Integrated Circuits and Systems Design (SBCCI'02).
[8] A.R. cooper, *Parallel architecture modified Booth multiplier*, IEEE proceedings.vol-135.pt-G.No.3June 1988.
[9] Sanjiv Kumar Mangal, Rahul M. Badghare, Raghavendra B. Deshmukh, R. M. Patrikar, *FPGA Implementation of Low Power Parallel Multiplier*, 20th International Conference on VLSI Design (VLSID'07).