Volume 4, Issue 6, June 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Development of Delay Reduction Technique for Microprocessor or DSP Based System

Er. Neha Gupta
M.Tech (ECE), SRMIET
India

Dr. B.K Sharma
Director, SRMIET
India

ABSTRACT: Designing high-speed multipliers with low power and regular in layout have substantial research interest. Many low power designs have been found. Power reduction can be improved using structure modification. To scale back the facility consumption of multiplier factor booth coding methodology is being employed to rearrange the input bits. The operation of the booth decoder is to rearrange the given booth equivalent. Booth decoder can increase the range of zeros in variety. Hence the switching activity are going to be reduced that further reduces the power consumption of the design. The input bit constant determines the switching activity part that's once the input constant is zero corresponding rows or column of the multiplier ought to be deactivated. When multiplicand contains more number of zeros the higher power reduction can takes place. So in modified booth multiplier high power reductions can be achieved.

In this paper a modified structure with reduced switching activity is presented through optimization of design. The analysis is done on the basis of certain performance parameters i.e. Area, Speed and Power consumption and dissipation. Multipliers are considered to be an important component in DSP applications like filters. Therefore, the low power multiplier is a necessity for the design and implementation.

Keywords: Column bypassing multiplier, Modified booth algorithm, Spartan-3AN

I. Introduction

The multiplier is one of the key hardware blocks in most of the digital and high performance systems such as digital signal processors and microprocessors [1]. An increasing number of high-speed DSP applications have need of a high

Precision fixed- or floating-point multiplier suitable for VLSI implementation [2]. The multipliers are the better option for high-speed data processing. Various algorithms and architectures have been proposed to accelerate multiplication such as Booth Algorithm [3], Modified Booth Algorithm [4], Braun and Baugh-Woolley Algorithm etc. The computation of the multipliers manipulates two input data to generate many partial products.

In scaling of C-MOS circuit the most important parameter is power dissipation. Thus power reduction is the key design goal for communication design and computing system power consist of static power and dynamic power. Static power dissipation is due to leakage current and dynamic power dissipation due to charging and discharging. Many low power designs have been found. Power reduction can be improved using structure modification [6], [7], [8].

In this paper a modified structure with reduced switching activity is presented through optimization of design. Initially an Array multiplier based on Braun multiplier is defined. After that a proposed modified multiplier based algorithm design has been presented. The experimental results with various multiplications have also been shown and discussed.

II. Preliminaries

A. ARRAY MULTIPLIER

Array multiplier is an efficient layout of a combinational multiplier. Multiplication of two binary number can be obtained with one micro-operation by using a combinational circuit that forms the product bit all at once thus making it a fast way of multiplying two numbers since only delay is the time for the signals to propagate through the gates that forms the multiplication array. With its good structure, this multiplier is based on the standard add and shift operations. Each partial product is generated by taking into account the multiplicand and one bit of multiplier each time. The impending addition is carried out by high-speed carry-save algorithm and the final product is obtained employing any fast adder. The number of partial products depends upon the number of multiplier bits.

Consider the multiplication of two unsigned
n-bit numbers.
Where: -
$A=a_{n-1}, a_{n-2}, . . . , a_0$ is the multiplicand.
$B= b_{n-1}, b_{n-2}, . . . , b_0$ is the multiplier. And

C=c_{2n-1}, c_{2n-2}, . . . , c_0, is the product, which can be written as follows:

$$C=\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left(a_i * b_j\right) 2^{i+j}$$

The additions are shown in the corresponding row. It indicates that Array Multiplier gives more power consumption as well as optimum number of components required, but delay for this multiplier is larger. It also requires more number of gates because of which area is also large, therefore the array multiplier is less economical [8] [9]. Thus, it is a fast multiplier but hardware complexity is high.

Parallel multiplier: Consider the multiplication of two unsigned n-bit numbers,
Where:-
A=a_{n-1}, a_{n-2}, . . . , a_0 is the multiplicand.
B= b_{n-1}, b_{n-2}, . . . , b_0 is the multiplier. And
C=c_{2n-1},c_{2n-2}, . . . , c_0, is the product, which can be written as follows:

$$C=\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left(a_i * b_j\right) 2^{i+j}$$

The additions are shown in the corresponding row in Figure 1. On the other hand, the Baugh-Woolley multiplier uses the same array structure to handle 2's complement multiplication, with some of the partial products replaced by their complements. The multiplier array consists of (n-1) rows of CSA (carry save adder), in which each row contains (n-1) FA (full adder) cells. Each FA in the CSA array has two outputs: the sum bit goes down while the carry bit goes to the lower-left FA. For an FA in the first row, there are only two valid inputs, and the third input bit is set to 0. Therefore, it can be replaced by a two-input half-adder. The last row is a ripple adder for carry propagation. In this paper, we propose a low-power design for this multiplier.Low-power multipliers with row-bypassing: A low-power multiplier design may disable the operations in some rows to save power. If bit b_j is 0, all partial products $a_ib_{j,}$ are zero. An array implementation, known as the Braun multiplier, is shown in Figure. 2. Therefore; the additions in the corresponding row in Fig. 1 can be bypassed. The row by passing multiplier is shown in figure. 3. Each cell in the CSA (carry save adder) array is augmented with three tri-state gates and two multiplexers.

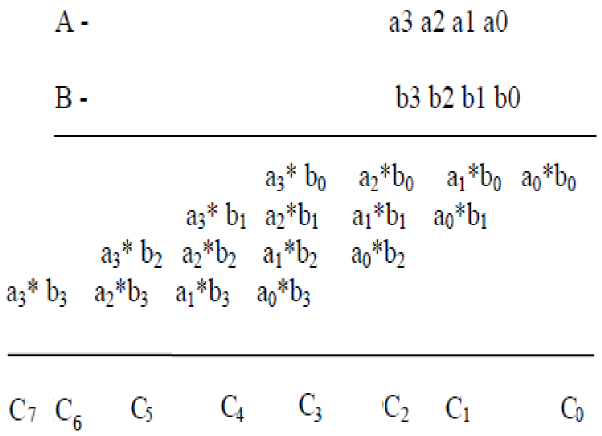

A -
a3 a2 a1 a0
B -
b3 b2 b1 b0
a_3* b_0 a_2*b_0 a_1*b_0 a_0*b_0
a_3* b_1 a_2*b_1 a_1*b_1 a_0*b_1
a_3* b_2 a_2*b_2 a_1*b_2 a_0*b_2
a_3* b_3 a_2*b_3 a_1*b_3 a_0*b_3
C_7 C_6 C_5 C_4 C_3 C_2 C_1 C_0

Figure 1: 4x4 multiplication

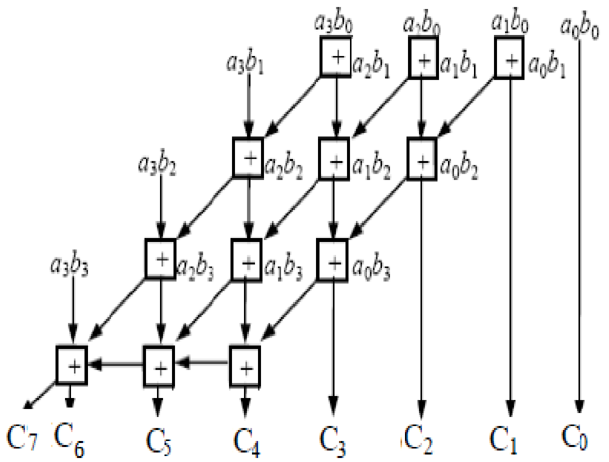


Figure 2: 4x4 Braun multiplier ref [4]

For example, let b_2 be 0 in Figure 3. In this case, the carry save adder in the second row (enclosed in the circle) can be bypassed, and the outputs from the first row are fed directly to the third row carry save adder. However, since the rightmost FA in the second row is disabled, it does not execute the addition and thus the output is not correct. To remedy this problem, an extra circuit must be added, and these elements locate in the triangle.

III. The Proposed Design

The switching activity of the component used in the design depends on the input bit coefficient. This means if the input bit coefficient is zero, corresponding row or column of adders need not be activated. If multiplicand contains more zeros, higher power reduction can be achieved. Instead of bypassing rows of full adders (FA), we propose a multiplier design in which columns of adders are bypassed. In this approach, the operations in a column can be disabled if the corresponding bit in the multiplicand is 0. Therefore, there are two advantages in this proposed approach. First, it eliminates the extra correcting circuit. Second, the modified FA is simpler than that used in the row-bypassing multiplier. Consider the multiplication shown in Figure 3, which executes 1101×1001.

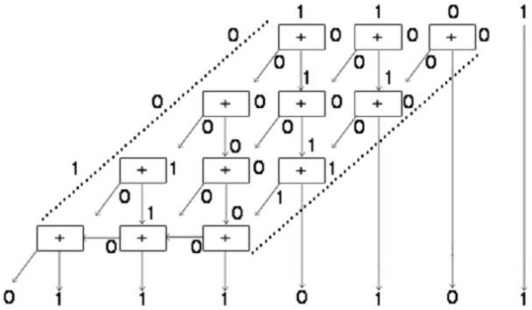


Figure 3: Column Bypass Multiplication

Note that, in the first and third diagonals (enclosed by dashed lines), two out of the three input bits are 0: the "carry" bit from its upper right FA, and the partial product a_ib_j (note that $a_0 = a_2 = 0$). As a result, the output carry bit of the FA is 0, and the output sum bit is simply equal to the third bit, which is the "sum" output of its upper FA.

A. Basic Idea.

Now, consider another multiplication of 1111 x 1000. Since multiplicand contains no zero, all columns will get Switched and consume more power. Higher power reduction can be achieved if the multiplicand contains more number of 0's than 1's. In this approach we propose Modified Booth Recoding Unit which will force multiplicand to have number of zeros, if does not have a single zero. The advantage here is that if multiplicand contains more successive number of one's then booth-recoding unit converts these ones in zeros.

B. Multiplier Design

The low power multiplier can be constructed using three units as Detection of Zero Unit, modified Booth Recoding Unit and Multiplication Unit.

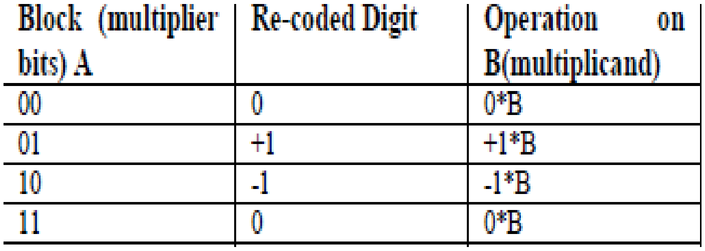
C. Detection of Zero Unit:

This unit scans the number of zeros and their respective position in the multiplicand, so as to bypass the corresponding column. If multiplicand contain at least one zero then it will feed the column bypass multiplier and multiplication will be performed using column bypassing. If multiplicand does not contain zero, multiplicand will be given to the Modified Booth Recoding Unit and after that multiplication will be performed.

IV. Proposed Modified Booth Encoder Multiplier Alorithm

Modified Booth Algorithm (MBA) is one of the powerful multiplication algorithms [17]. It is high speed multiplier to enhance parallelism which helps in reducing number of partial product (pp) row, by using modified booth algorithm. So, the overall number of partial product is decrease from N to N/2, where N is multiplier [18]. The digital bit recoding in two step encoding and selection has been shown in figure 2. The recoding method is widely used to generate the partial product for implementation of parallel multiplier. For Implementation of booth algorithm booth recoding is important and it is widely use for generating partial product for implementing parallel multiplier. The figure II is architecture of modified booth multiplier is given below. In this algorithm, we used two operands A and B as multiplicand and multiplier and decoder is used to convert the given input to equivalent booth value. Therefore, it contains more number of 0's with the output of decoder. Suppose B multiplicand and A is multiplier to form the product Z, the generating partial product is used AND operation and after adding we get the final product of A*B. So the higher representation Radix lead to fewer digits, K- bit binary number can be interpreted as K/2 digit Radix-4.

Radix-2 modified booth algorithm



Block (multiplier bits) A	Re-coded Digit	Operation on B(multiplicand)
00	0	0*B
01	+1	+1*B
10	-1	-1*B
11	0	0*B

In Modified Booth Multiplier algorithm Radix-2 append bit 0 in LSB in rightmost to fulfill two bits overlap one bit of previous adjacent bit. As shown in table I -

□□If the multiplier bit is ‘00’and ‘11’ no change put 0000.

□□If the multiplier bit is ‘01’put the value of

Multiplicand. If the multiplier bit is ‘10’ put the value of 2’s Complement of multiplicand.

□□If the multiplier bit is ‘10’ put the value of 2’s

Complement of multiplicand.

Example: Suppose a (multiplier) and B (multiplicand)

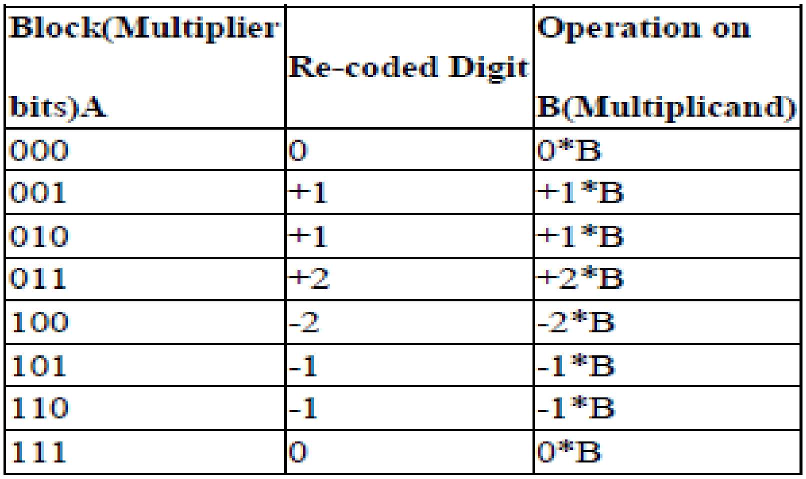
A= +4 (0100) and B= +3 (0011)

Append zeros in LSB of multiplier then by help of booth recording table find partial product and final result

If Multiplier A= 4=0100

There are some disadvantages in performing number of add/subtract operation, because it is inconvenient to design parallel multiplier. Therefore to overcome from this problem modified booth multiplier Radix-4 or higher Radix comes out. It reduces the number of partial product by N/2 or half instead of adding and shifting for every column and multiplies by 1 or 0. In this algorithm, only second column is used which is multiply by ±1, ±2 or 0 to get same result. Therefore, it is high speed multiplier which is used in present scenario for high speed processor. In fact 8.72% of all the instruction in typical processing is multiplier. Therefore, by using this modified booth multiplier one can decrease partial product by half of multiplier.

Modified Booth Recoding Radix - 4



Block(Multiplier bits)A	Re-coded Digit	Operation on B(Multiplicand)
000	0	0*B
001	+1	+1*B
010	+1	+1*B
011	+2	+2*B
100	-2	-2*B
101	-1	-1*B
110	-1	-1*B
111	0	0*B

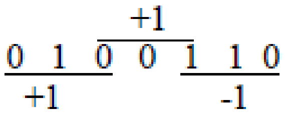
Booth Recoding

The advantage of Booth recoding is that it generates roughly one half of the partial products as compared to the multiplier implementation, which does not use Booth recoding. However, the benefit achieved comes at the expense of increased hardware complexity. Indeed, this implementation requires hardware for the encoding and for the selection of the partial products (0, ±Y, ±2Y).

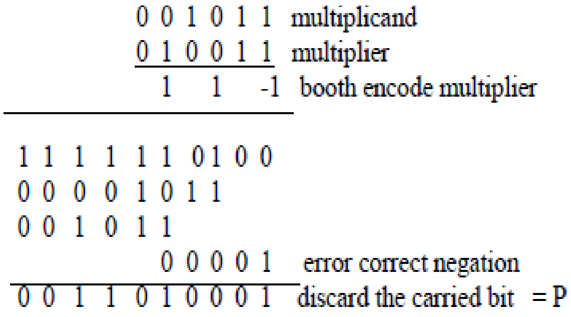
Example: - Multiplier A=010011 and multiplicand

B=01011 for using recoding technique first three bit

Pairing done for multiplier we get



Three bits pairing by extend LSB by ‘0’ to make a pair of triplet bit then calculate partial product



```
                0 0 1 0 1 1   multiplicand
                0 1 0 0 1 1   multiplier
                  1   1  -1   booth encode multiplier
     ______________________

     1 1 1 1 1 1  0 1 0 0
     0 0 0 0 1 0 1 1
     0 0 1 0 1 1
                  0 0 0 0 1   error correct negation
     ____________________
     0 0 1 1 0 1 0 0 0 1  discard the carried bit  = P
```

V. Result

To calculate the performance of this low power multiplier we implement the new design with Spartan-3AN technology. We compare the performance of this new design with Braun multiplier shown in Table 2. Our result shows that row by pass multiplier actually consume more power possibly due to extra logic circuitry. Optimization in hardware can be obtained by using VERILOG instead of VHDL. Our design produces less power dissipation

Table 2: Comparison

Multiplier Type	**Device and Family Implementation on 4x4 bit width**	
	Spartan- 3AN	**Spartan 2**
	Estimated Delay (ns)	Estimated Delay (ns)
Column bypass Multiplier	5.246	14.886
Array Multiplier	11.953	21.068

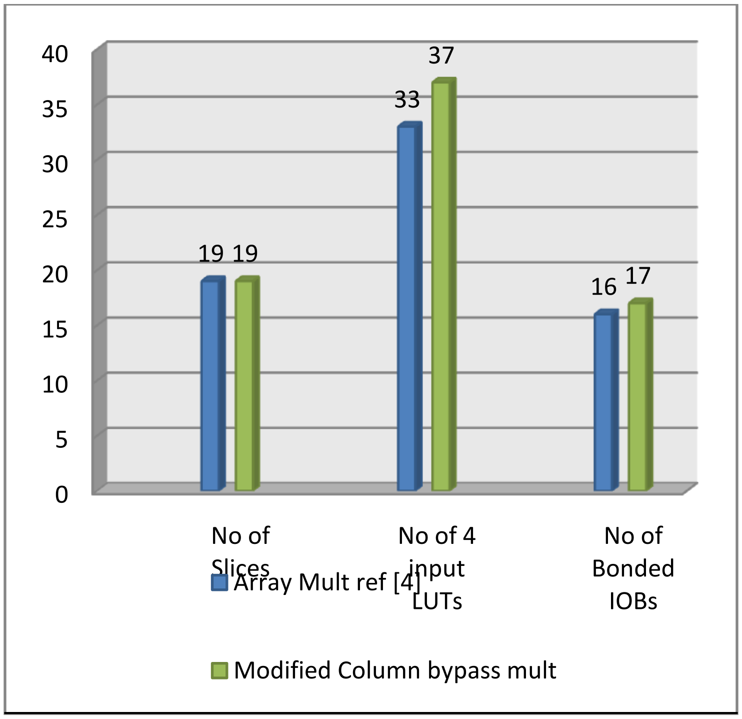


Figure 4: Comparison of hardware resources utilized in the design

The area occupancy on FPGA is measured in terms of its CLBs and IOBs shown in Figure 4. The current design of modified column bypass multiplier occupies around 1% FPGA hardware resources on Spartan 3AN- XC3s700an- 5fgg484 Device. Comparison of delay and power consumption is shown in Figure 5. The simulation result for array multiplier and modified column by pass multiplier is shown in Figure 6 and Figure 7 respectively.

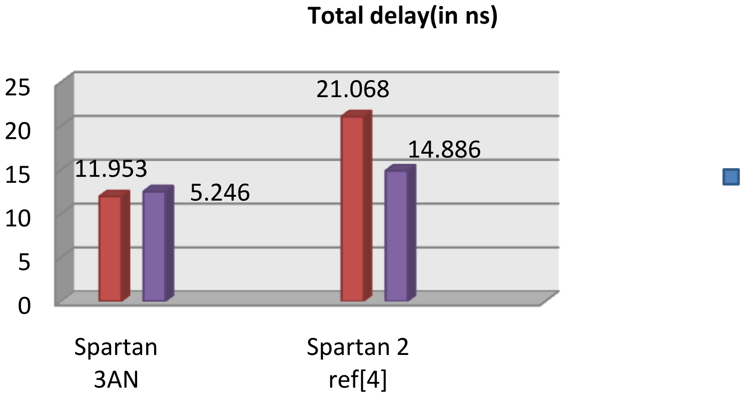
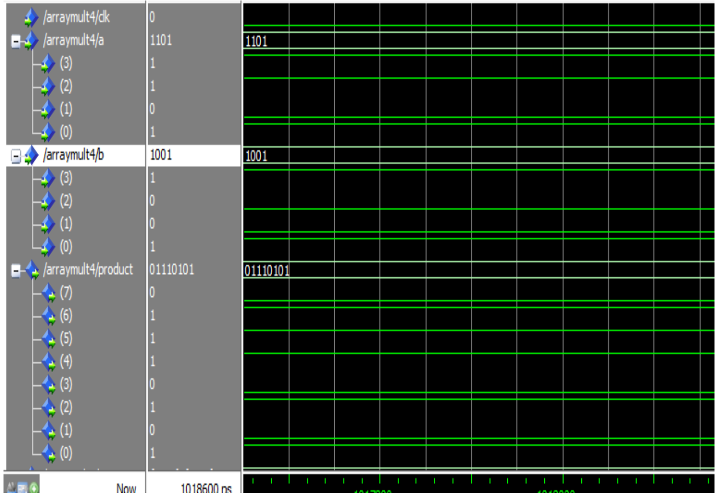


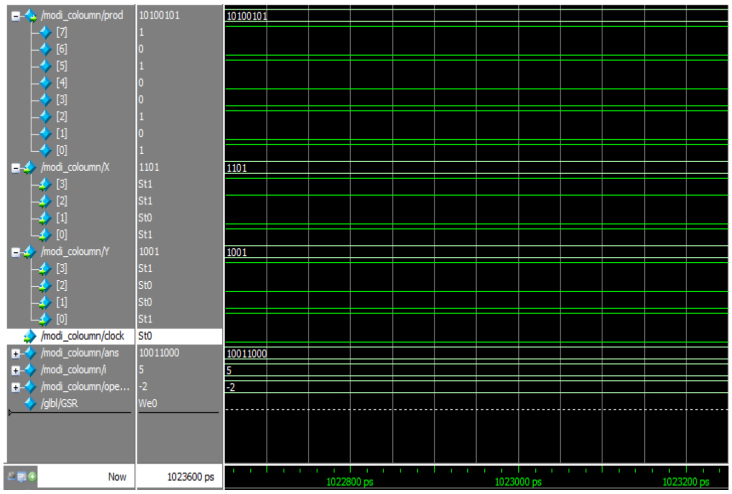
Figure 5: Comparison of propagation delays of Array and Modified Column bypass multiplier.

Graph above shows that power consumption in modified column bypass multiplier has reduced by (-1.06mW) when implemented on Spartan 3AN device as compared to Spartan 2 device which is the main objective of this research work. That also reflects increased efficiency of the multiplier.



```
/arraymult4/clk        0
/arraymult4/a          1101
  (3)                  1
  (2)                  1
  (1)                  0
  (0)                  1
/arraymult4/b          1001
  (3)                  1
  (2)                  0
  (1)                  0
  (0)                  1
/arraymult4/product    01110101
  (7)                  0
  (6)                  1
  (5)                  1
  (4)                  1
  (3)                  0
  (2)                  1
  (1)                  0
  (0)                  1
Now                    1018600 ps
```

Figure 6: Simulation result for Array multiplier



```
/modi_coloumn/prod     10100101
  [7]                  1
  [6]                  0
  [5]                  1
  [4]                  0
  [3]                  0
  [2]                  1
  [1]                  0
  [0]                  1
/modi_coloumn/X        1101
  [3]                  St1
  [2]                  St1
  [1]                  St0
  [0]                  St1
/modi_coloumn/Y        1001
  [3]                  St1
  [2]                  St0
  [1]                  St0
  [0]                  St1
/modi_coloumn/clock    St0
/modi_coloumn/ans      10011000
/modi_coloumn/i        5
/modi_coloumn/ope...   -2
/glbl/GSR              We0
Now                    1023600 ps
```

Figure 7: Simulation result for modified Column bypass

VI. Future Scope

The proposed multiplier can further be optimized in area if Robertson's algorithm is used instead of modified column bypass multiplier. But in this research work our main objective was to reduce delay consumption which has been achieved.

VII. Conclusion

In this paper we have concluded delay optimization using modified booth algorithm with implemented on Spartan 3-AN family. Optimization has been achieved using VERILOG in stead of VHDL. This technique achieves higher delay reduction with lower hardware overhead.

Acknowledgment

The authors would like to thank the management of SRM Global institute for their most support and encouragement. Also they are thankful to the staff members of ECE department for their time to time help. They are greatly indebted to their families for always encouraging them to attain their goal.

REFERENCES

[1] Wen-Chang Yeh and Chein-Wei Jen, "High speed Booth encoded parallel multiplier design", IEEE Transaction on Computers, vol. 49, pp. 692-701, July 2000.

[2] A.R. Cooper , " Parallel architecture
modified Booth multiplier" IEEE Proceedings, Vol.135, Pt. G, No.3,June 1988.

[3] A. D. Booth, "A signed binary multiplication technique", Quarterly J.Mech. and Applied Math, vol. 4, pp. 236-240, 1951.

[4] O. L. Mac-Sorley,"High Speed Arithmetic in Binary Computers", Proceedings of IRE Vol.49, No. 1, January, 1961.

[5] D. Jackuline Moni, P. Eben Sophia, "Design of low power and high speed Configurable Booth Multiplier" IEEE transaction 2011,978-1-4244-8679.

[6] Burd T.D., Brodersen R.W.: "Energy efficient CMOS microprocessor design", Proceedings 28th. Annual HICSS Conference, Jan. 1995, vol. I, pp. 288-297.

[7] Frenkil J.:"A multi-level approach to low-power IC design", IEEE Spectrum, Volume 35, Number 2, February 1998.

[8] Havinga P.J.M., Smit G.J.M.: "Design techniques for low power systems" Journal of Systems Archi tecture, Vol. 46, Iss. 1, 2000, a previous version appeared as CTIT Technical report, No. 97-32, Enschede, the Netherlands, ISSN 1381-3625

[9] Mohamed Al-Ashrafy,Ashraf Salem,Wagdy Anis "An Efficient Implementation of Floating Point Multiplier", IEEE 978-1-4577-0069-9/11/ 2011

[10] Stefan Langemeyer, Peter Pirsch, Holger Blume,"FPGA Architecture for real time processing of variable lengthFFTS", IEEE, 978-1-4577-0539,7/11/2011

[11] Muhammad H. Rais and Mohammed H. Al Mijalli, "Braun's Multipliers: Spartan-3AN based Design and Implementation",Journal of Computer Science,.ISSN 1549-3636© 2011 Science Publications. 1629-1632, 7 /11/2011

[12] Tushar V. More and Dr. R.V.Kshirsagar.Design of Low Power Column Bypass Multiplier using FPGA", IEEE, 978-1-4244-8679,3/11/2011

[13] Dr. Ravi Shankar Mishra,Prof. Puran Gour, Braj Bihari Soni ,Design and Implements of Booth and Robertson's multipliers algorithm on FPGA, International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 Vol. 1, Issue 3, pp.905-910

[14] C.-Y. Pai, W.E. Lynch and A.J. Al-Khalili "Low-power data-dependent 8 _ 8 DCT/ IDCT for video compression", IEEE, IEEE Proceedings online no. 20030564 doi: 10.1049/ip-vis:20030564 IEE Proc.-Vis. Image Signal Process., Vol. 150, No. 4, August 2003

[15] Li Shang, Alireza S Kaviani, Kusuma Bathala, "Dynamic Power Consumption in Virtex™-II FPGA Family" ACM 1-58113-452-5, February 24-26, 2002.

[16] Wen-Chang Yeh and Chein-Wei Jen, "High-speed Booth encoded parallel multiplier design", IEEE Transaction on Computers Vol. 49,july 2000.

[17] Kavita, Jasbir Kaur, "Design and Implementation of an EfficientModified Booth Multiplier using VHDL", International Conference on Emerging Trend in Engineering and Management, ISSN:2231-0347, Vol.3(3, July2013.

[18] A.S.Prabhu, V.Elakya, "Design of modified Low Power Booth Multiplier,IEEE,2012.