Design of Improved Systolic Array Multiplier and Its Implementation on FPGA

Miss. Pritam H. Langade

Department of Electronics and Telecommunication,

SSGMCE,Shegaon.

pritam.langade@gmail.com

Prof. S. B. Patil

Department of Electronics and Telecommunication,

SSGMCE,Shegaon.

Abstract— In the recent technology, there is a need of high speed and powerful data processing. Such complex problem is overcome by using parallel computing technology which uses the concept of pipelining for this application. This project provides the implementation issues in systolic array multiplier for high speed data processing. This also introduces the concept of parallel processing and pipelining which improves the speed of execution. In this project, a new method of multiplying matrices using systolic array multiplier is designed and each processing element in systolic array is replaced by Dadda multiplier. Here the VHDL code is written for matrix multiplication with systolic architecture. It is compiled and simulated by using Modelsim SE 6.3f, synthesized by using Xilinx ISE 14.5 and targeted to the device XC3S500E-4-FG320.

Keywords— Parallel Processing, Pipelining, Systolic Array, Multiplication, Dadda Multiplier.

INTRODUCTION

Systolic algorithm is the form of pipelining, which is in more than one dimension [16]. In this algorithm, data flows from a memory in a rhythmic way and it passes through many processing elements before it returns to memory. Systolic arrays have balanced, uniform, grid like architectures in which each line indicates a communication path and each intersection represents a cell or systolic elements [13]. High speed multiplication is a basic requirement of digital systems giving high performance [10]. Execution time for Parallel multiplication has been decreased by using the technique of column compression [8]. There are two classes of parallel multipliers, which are array multipliers and tree multipliers. Tree multipliers are also known as column compression multipliers. Column compression multipliers are faster than array multipliers. Two of the most well-known column compression multipliers have been presented by Dadda and Wallace. Dadda multiplier has a faster performance, so it is used for the implementation of the proposed multiplier. Systolic Arrays are computational networks, with local data storage, that uses the parallel processing and pipelining techniques to achieve a high throughput [6]. Matrix multiplication is a computation intensive operation and plays an important role in many scientific and engineering applications [2]**.** This work demonstrates an effective design and efficient implementation of the Matrix Multiplication using Systolic Architecture. In this project, matrices are multiplied by using systolic array multiplier and each processing element in this Systolic array multiplier is replaced by Dadda multiplier. Dadda multiplier is a binary multiplier which uses the concept of column compression to reduce the partial products and hence increases the speed of multiplication.

PROPOSED WORK

Multipliers play an important role in today's digital signal processing and various other applications [9]. With advances in technology, many researchers have tried and are trying to design multipliers which offer high speed, low power consumption, regularity of layout and hence less area or even combination of them in one multiplier thus making them suitable for various high speed, low power and compact VLSI implementation.

1. ARRAY MULTIPLIER

Array multiplier is an efficient layout of a combinational multiplier. An n x n array multiplier requires n(n-1) adders and n^2 AND gates. Figure 1 shows a 4 bit array multiplier. Array multipliers have a large critical path and are very slow. The main advantage of these multipliers is the regular structure which leads to ease of layout and design. Array multiplier requires n (n-2) full adders, n half-adders and n^2 AND gates.

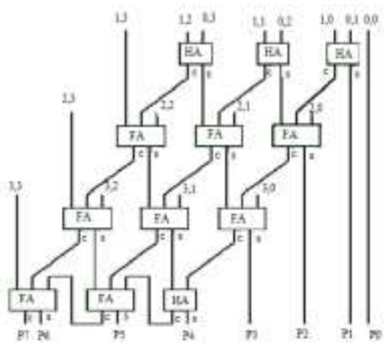


Fig 1: Architecture of 4 bit array multiplier.

2. WALLACE MULTIPLIER

The Wallace tree multiplier belongs to a family of multipliers called column compression multipliers[12]. Wallace multiplier is an efficient parallel multiplier. A Wallace tree multiplier offers faster performance for large operands. In Wallace's scheme, the partial products are reduced as soon as possible which is shown in fig. 2.

Conventional Wallace tree multiplier includes the following steps:

I. First step is to form partial product array (of n^2 bits).

II. In the second step, groups of three adjacent rows each, is collected. Each group of three rows is reduced by using full adders and half adders. Full adders are used in each column where there are three bits whereas half adders are used in each column where there are two bits. Any single bit in a column is passed to the next stage in the same column without processing. This reduction procedure is repeated in each successive stage until only two rows remain.

III. In the final step, the remaining two rows are added using a carry propagating adder. In a conventional Wallace multiplier, the number of rows in subsequent stages can be calculated as:
ri+1 = 2[ri/3] + ri mod 3 -------(1)

Where, ri mod 3 denotes the smallest non-negative remainder of ri/3.

Length of carry propogation adder is given by 2N-2-S,where N is the bit number and S is the number of stages required.

For example, in 4 bit Wallace multiplier, $r_i = r_0 = N = 4$, required number of stages are 2 and carry propogation adder is of 4 bit.

$$r_1 = 2\left(\frac{4}{3}\right) + 4 \bmod 3 = 3$$

Height is reduced upto 3, in first stage.

$r2 = 2\left(\frac{3}{3}\right) + 3 \bmod 3 = 2$. Thus in second stage, the height is reduced to 2.

The Wallace tree consists of numerous levels of such column compression structures. Wallace tree reduces the number of partial products to be added into 2 final intermediate results [14]. These two operands can then be added using fast carry-propagate adder to obtain the product result.What differentiates the Wallace tree multiplier from other column compression multipliers is that in the Wallace tree, every possible bit in every column is covered by the (3:2) or (2:2) compressors repetitively until finally the partial product matrix has a depth of only 2. Thus the Wallace tree multiplier uses as much hardware as possible to compress the partial product matrix as quickly as possible into the final product.

3. DADDA MULTIPLIER

Dadda's method does minimum reduction necessary at each level and requires the same number of levels as Wallace multiplier.In Dadda multiplier, less reductions are carried out in earlier stages.

The Dadda multiplier architecture can be divided into three stages.

I. Generation of partial products.
II. Partial product reduction.
   a) Let h1 = 2 and repeat hj+1 = floor (1.5*hj) for increasing values of j. Continue this until the largest j is reached, for which there exists at least one column in the present stage of the matrix with more dots than hj. Using this equation we get h1=2, h2=3, h3=4, h4=6, h5=9 and so on. As shown in figure 3, for example, in the first stage of the 4bit Dadda multiplication, the maximum height of columns is 4 therefore, the value of hj is 3 means heights of the columns are reduced to a maximum of 3. Similarly in the second stage, the maximum height of column is 3 and value of hj is 2 means heights of the columns are reduced to a maximum of 2.
   b) All the columns, with heights greater than hj, are reduced to a height of hj using either half adder or full adder. If the column height has to be reduced by one, use a half adder else use a full adder and continue this step till the column height is reduced to hj.
   c) Stop the reduction if the height of the matrix becomes two, after which it can be fed to final adder.
III. Once the height of matrix is reduced to two, the remaining two rows are added using a lookahead carry adder.

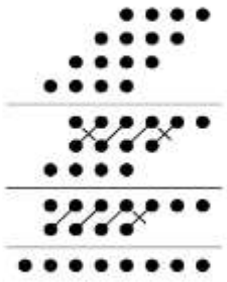


Fig 2: 4 bit Wallace multiplier

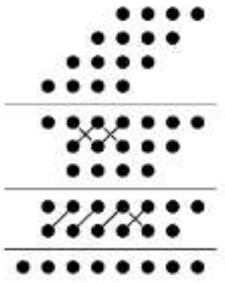


Fig 3: 4 bit Dadda multiplier

The Parallel Matrix Multiplication has many different identifications, but all with the similar implementation [4]. Parallelism is relevant not only for the data operations but also for the data transfer [5]. In this project, the PE is replaced with n bit Dadda multiplier to enhance the speed of Systolic Architecture. Also, aim is to compute the following equation with a two dimensional systolic array.

$$C_{m\times m} = A_{m\times m} \times B_{m\times m} \text{ -----------(1)}$$

Where A, B and C are the matrices with order $m\times m$. Each PE of systolic array performs binary multiplication of two n bit numbers. A systolic array multiplier is an arrangement of processing elements in an array where data flows synchronously across the array between neighbours usually with different data flowing in different directions[7],[15]. PE at each step takes input data from each step, outputs result in the opposite direction[11].Systolic Array is a set of interconnected cells which performs simple operation[1].

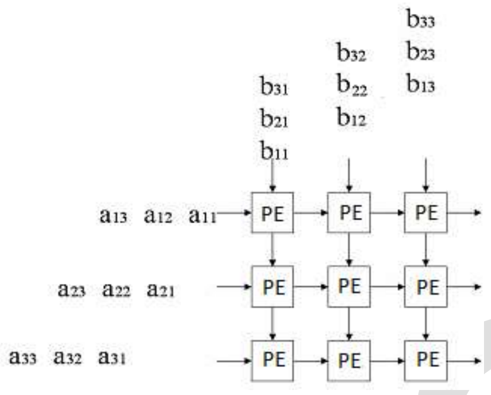


Fig 4: Systolic Array for matrix multiplication

The resulting output of this matrix is

$$\begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{pmatrix}$$

For example, when multiplying two 3*3 matrix we need 27 operations by conventional method according to the given formula:

```
For I = 1 to N
    For J = 1 to N
        For K = 1 to N
            C[I,J] = C[I,J] + A[J,K] * B[K,J];
        End
    End
End
```

But using systolic arrays, it can be done in only 9 clock pulses. So systolic array multiplier is used for matrix multiplication. The proposed two dimensional systolic array architecture is shown in figure 4, where each processing element is replaced with n bit Dadda multiplier. As in Dadda multiplier, fewer columns are compressed in the initial stages,and more columns in the later levels of the multiplier, it requires less expensive reduction phase and less hardware. Since the Dadda multiplier has a faster performance, we implement the proposed technique using this multiplier. During multiplication of two matrices, elements of first matrix are entered in normal way and second matrix elements are transposed before entering in processing element.

RESULTS AND DISCUSSION

The proposed Systolic array multiplier is very efficient multiplier which uses the concept of pipelining to enhance the speed of matrix multiplication. In this project, a new method of multiplying matrices using systolic array multiplier is designed. Here the speed of Array multiplier, Wallace multiplier and Dadda multiplier is verified. Out of these multipliers, Dadda multiplier is found to be having high speed and less area is utilized to design it. So, in proposed systolic array multiplier, Dadda multiplier is used as processing element to increase speed of matrix multiplication. It is compared with previously designed systolic array multiplier and it is found that the proposed design is faster and requires less number of registers as compared to previous one.

Simulation Results

The fig. 5, fig 6 and fig 7 shows the simulation results of Array multiplier, Wallace multiplier and Dadda multiplier respectively in which two 4 bit binary numbers 'a' and 'b' are given as input to produce 8 bit prod as output.

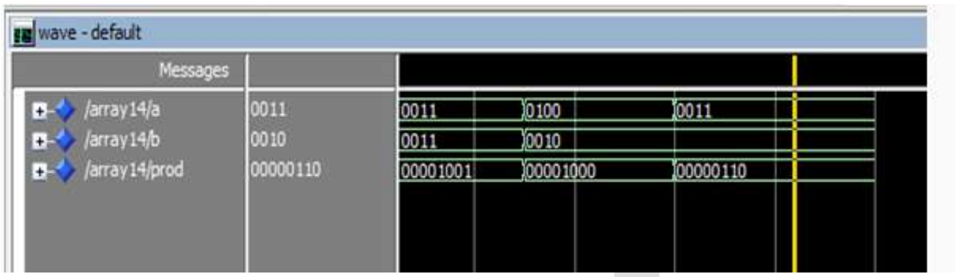


Fig 5 : Simulation of 4 bit Array Multiplier

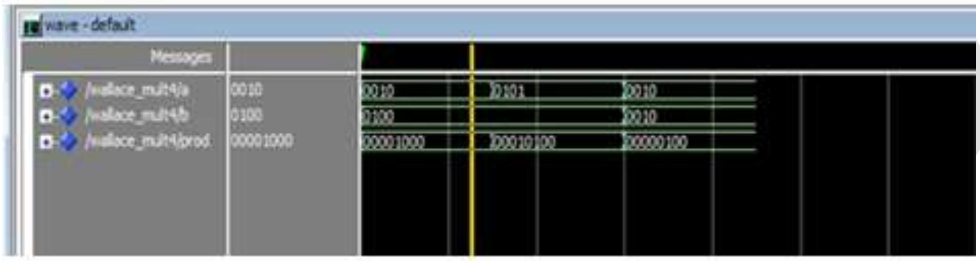


Fig 6 : Simulation of 4 bit Wallace multiplier

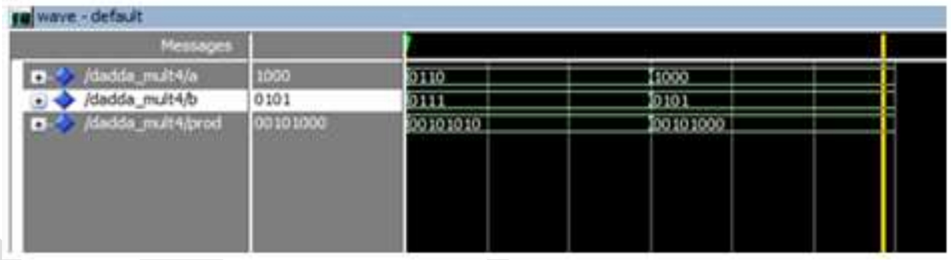


Fig 7: Simulation of 4 bit Dadda multiplier

The figure 8 shows the simulation result of proposed systolic array multiplier for matrix multiplication where each processing element is Dadda multiplier.

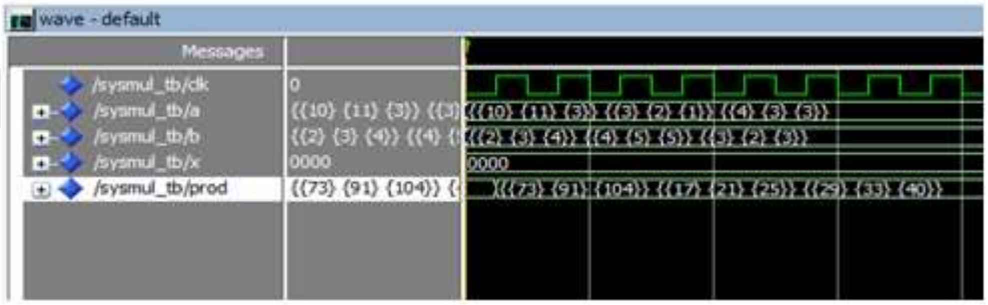


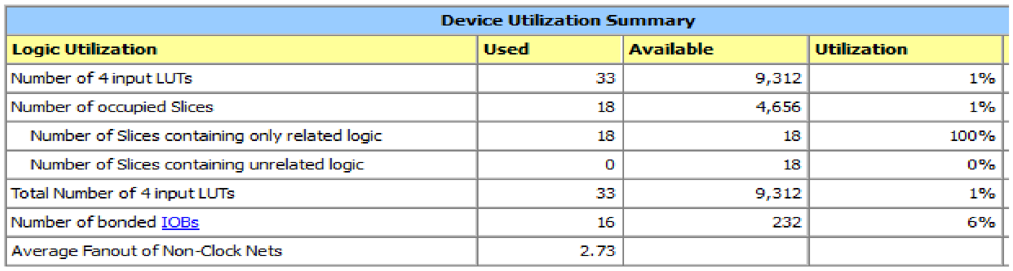
Fig 8 : Simulation of proposed systolic array multiplier

From simulation result, it is clear that the systolic array multiplier requires less number of clock cycles to multiply two matrices than mathematical calculation. Hence systolic array multiplier is the effective method for multiplication of matrices.

Design utilization summary and delays

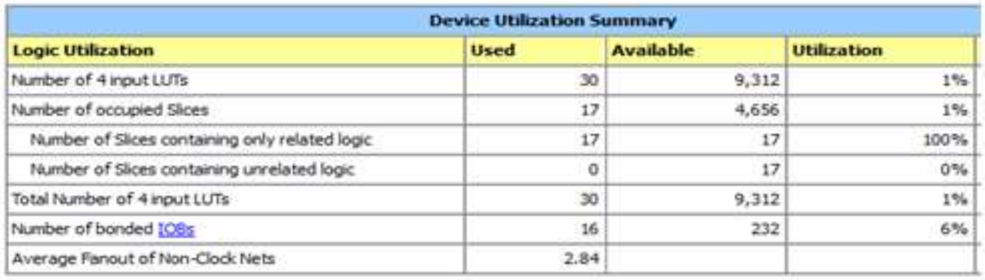
Following Table I, Table II and Table III shows design utilization summary of Array multiplier, Wallace multiplier and Dadda multiplier respectively.

Table I : Summary of Array multiplier



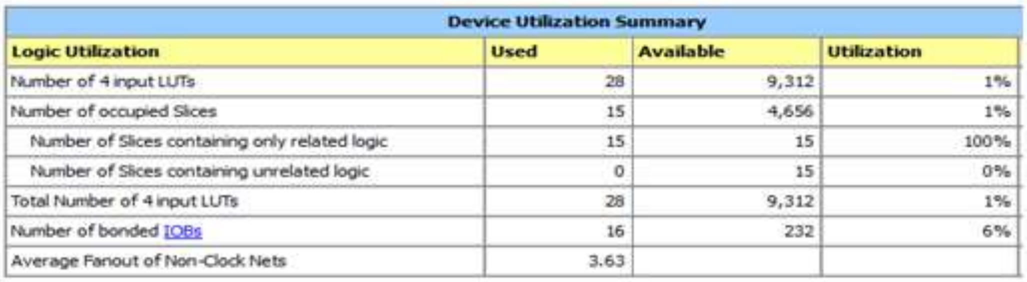
Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	33	9,312	1%
Number of occupied Slices	18	4,656	1%
Number of Slices containing only related logic	18	18	100%
Number of Slices containing unrelated logic	0	18	0%
Total Number of 4 input LUTs	33	9,312	1%
Number of bonded IOBs	16	232	6%
Average Fanout of Non-Clock Nets	2.73		

Table II : Summary of Wallace multiplier



Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	30	9,312	1%
Number of occupied Slices	17	4,656	1%
Number of Slices containing only related logic	17	17	100%
Number of Slices containing unrelated logic	0	17	0%
Total Number of 4 input LUTs	30	9,312	1%
Number of bonded IOBs	16	232	6%
Average Fanout of Non-Clock Nets	2.84		

Table III : Summary of Dadda multiplier



Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of 4 input LUTs	28	9,312	1%
Number of occupied Slices	15	4,656	1%
Number of Slices containing only related logic	15	15	100%
Number of Slices containing unrelated logic	0	15	0%
Total Number of 4 input LUTs	28	9,312	1%
Number of bonded IOBs	16	232	6%
Average Fanout of Non-Clock Nets	3.63		

The above design utilization summary of multipliers shows that Dadda multiplier used less number of slices, 4 i/p LUTs than Array and Wallace multiplier. It also has minimum path delay than Array and Wallace multipliers which is shown in the following Table IV. Hence it has more speed and is used as processing element in proposed systolic array multiplier.

Table IV : Comparison of Delays in Multipliers

Multipliers	Delays
Array Multiplier	10.466 ns
Wallace Multiplier	10.335 ns.
Dadda Multiplier	10.191 ns.

The implementation of n x n matrix multiplication is done by using systolic array multiplier having n bit Dadda multiplier as a processing element. The simulation result in fig 8 shows input and output matrices $A_{3\times3}$, $B_{3\times3}$, $C_{3\times3}$ respectively where the matrix elements are of 4 bit each. After simulation, the design is synthesized on the platform XILINX ISE 14.5 and is targeted to the device XC3S500E-4-FG320. Critical path delay represents the core speed of the design. The proposed systolic array multiplier is compared with previously designed systolic array multiplier which is shown in following Table V. It is noticed that the core speed of systolic array multiplier is 228.93MHz which is more than previously designed systolic array multiplier and it also requires less number of 8 bit registers.

Table V : Comparison of Systolic Array Multiplier

Parameters	Improved Systolic Array Multiplier	Previously designed Systolic Array Multiplier
1) Critical path delay	4.368 ns	4.757 ns
2) 8 bit registers	9	34
3) Core speed	228.93MHz	210MHz

CONCLUSION

Systolic Array Multiplier is an arrangement of processors in an array where data flows synchronously across the array between neighbours, usually with different data flowing in different directions. Systolic array multiplier is very efficient multiplier which uses the concept of pipelining to enhance the speed of multiplication. The proposed Systolic Array Multiplier multiplies n x n matrices with less number of clock cycles. In this proposed work, each processing element is replaced by Dadda multiplier to reduce delay and hence increases the speed of multiplication. Consequently, proposed Systolic Array Multiplier can be considered as an ideal primitive for dense matrix vector multiplication which is quite adequate for most image processing applications. The proposed Systolic Array Multiplier is possible using the given software algorithm and hardware implementation so that it can achieve higher speed as compared to existing Systolic Array Multiplier. The designed matrix multiplier is simulated using ModelSim SE 6.3f and implemented on a Xilinx ISE 14.5 then targeted on xc3s500EFG320FPGA.

REFERENCES:

[1] H. T. Kung, "Why Systolic Architecture?" IEEE Computer,15(1),(1982)37-46.

[2] Syed Manzoor Qasim, Shuja Ahmad Abbasi and Bandar Almashary, " A Proposed FPGA-based Parallel Architecture for Matrix Multiplication" 978-1-4244-2342-2/08/$25.00 ©2008 IEEE.

[3] Syed M. Quasim,Ahmed A. Telba and Abdulhameed Y. AIMazroo, "FPGA implementation of matrix multiplier architectures for use in image and signal processing application", IJCSNS International Journal of Computer Science And Network Security, VOL.10 No.2,February 2010.

[4] Mahendra Vucha, Arvinda Rajawat ,"Design and implementation of systolic array architecture for matrix multiplication", IJCA International Journal of Computer Applications (0975-8887),Volume 26-No.3,July 2011.

[5] Ekta Agrawal, "Systolic and Semi-Systolic Multiplier", MIT International Journal of Electronics and Communication Engineering, *Vol. 3, No. 2, August 2013, pp. 90–93, ISSN No. 2230-7672* ©MIT Publications.

[6] Mohammad Mahdi Azadfar , "Implementation of A Optimized Systolic Array Architecture for FSBMA using FPGA for Real-time Applications", IJCSNS International Journal of Computer Science and Network Security, VOL.8 NO.3,March 2008.

[7] Ganapathi Hegde, Cyril Prasanna Raj P , P.R. Vaya: "Implementation of Systolic Array Architecture for Full Search Block Matching Algorithm on FPGA", European Journal of Scientific Research,Vol.33 No.4(2009),pp.606-616.

[8] Bhabani P. Sinha, Pradip K. Srimani, "Fast Parallel Algorithms for Binary Multiplication and Their Implementation on Systolic Architectures", IEEE Transactions on computers, Vol.38. No.3. March 1989.

[9] Jasbir Kaur and Kavita, " Structural VHDL Implementation of Wallace Multiplier" International Journal of Scientific & Engineering Research, Volume 4, Issue 4, April-2013 1829 ISSN 2229-5518.

[10] B. Ramkumar, V. Sreedeep and Harish M Kittur, Member, IEEE, " A Design Technique for Faster Dadda Multiplier ".

[11] Anuja George, "A Novel Design of Low Power, High Speed SAMM and its FPGA Implementation", International Journal of Computer Applications (0975 – 8887) Volume 43– No.4, April 2012.

[12] K'Andrea C Bickerstaff, Earl E Swartzlander and Michael J. Schulte, "Analysis of column compression multipliers," 0-7695-1150-3/01 $10.00@ 2001 IEEE.

[13] Kurtis T. Johnson and A.R.Hurson, Pennsylvania State University, "General Purpose Systolic Arrays" IEEE, Nov.1993.

[14] K.Gopi Krishna, B.Santhoshand V.S ridhar, “Design of Wallace Tree Multiplier using Compressors”, International journal of engineering sciences & research Technology,September 2013.

[15] Bairu K. Saptalakar, Deepak kale, Mahesh Rachannavar , Pavankumar M. K, “Design and Implementation of VLSI Systolic Array Multiplier for DSP Applications”, International Journal of Scientific Engineering and Technology (ISSN : 2277-1581) Volume 2 Issue 3, PP : 156-159 1 April 2013.

[16] Himani Harmanbir Singh Sidhu, “Design and Implementation Modified Booth algorithm and systolic multiplier using FPGA” International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 11, November – 2013.