16 BIT IMPLEMENTATION OF ASYNCHRONOUS TWOS COMPLEMENT ARRAY MULTIPLIER USING MODIFIED BAUGH-WOOLEY ALGORITHM AND ARCHITECTURE.

AditiPandey*

Electronics & Communication,University Institute of Technology,
Near Airport Road, Bhopal, Madhya Pradesh 462030, India†
First author_aditipandeyrules@gmail.com‡

AvinashRai

Assistant Professor Electronics & Communication,University Institute of Technology,
Near Airport, Bhopal, Madhya Pradesh 462030, India
Second author_avinashrai@rgtu.net

***Abstract* – This paper briefly describes about the asynchronous twos complement multiplication of array multiplier using modified Baugh-Wooleyalgorithm and architecture. The paper gives a clear description about the parameters like area optimization, enhanced speed and low power consumption. In the paper we have implemented a 16 bit array multiplier which gives a regular size layout which shows proper area optimization with the LUTs. The implementation has been done on Xilinx 14.3.The family used is Spartan 3E with the device XC3S100E , package VQ100 with speed of -5 & 100us technology . The top level module implementation is done using Verilog synthesis in the ISim simulator. This is the most important application of the paper as all the processors depends on arithmetic and logical unit for storing data. These days all digital machines stores twos complement data and hence with array multiplier using Baugh-Wooley algorithm the multiplication of two 16 bits signed number becomes possible.**

Keywords: array multiplier, baugh-wooleyalgorithm ,architecture, carry save adders.

I. INTRODUCTION

With the increasing demand in the speed and power consumption in the digital signalling processes the use of multipliers is mandatory to enhance the speed , consumption with less area of chip[H.Tarun&et.all 2015].Multipliers are the key components in any digital signaling processing Thus VLSI scale is increasing at a tremendous rate with the increase in utilities.There are a number of multipliers used in various applications including the serial multipliers which uses add& shift algorithm. While the parallel multipliers uses the partial product generation. The serial multipliers can be used for unsigned as well as signed multiplication using Baugh-Wooleyalgorithm[H.Tarun&et.all 2015].Parallel multipliers are used to reduce partial products using booth algorithm that increases the silicon area with increased speed. The use of a particular multiplier depends on its usage. In the paper we have implemented the multiplication by Verilog synthesis. The kind of multiplier depends on its application. Since here our aim is to generate a regular structure with area optimization and speed have used we array multipliers.

II.ARRAY MULTIPLIERS

The array multiplier originated from multiplier parallelogram. Moreover the partial products generated are added in the next line in the form of carry out propagated signal. For non pipelined multiplier all products generate at same time. Thus for n bit multiplier the delay is same as number of full adder[G.Vishnu&et all 2016].

Since it has a regular structure thus the layout formation is a boon in its working. The layout stays small which thus means it can be implemented easily. The hardware designing thus becomes cost effective and simple. Moreover its pipelined architecture allows easier bit implementation [H.Tumiyama&et.all 2016]. With bits less than 32 it gives correct delay but if its more than it gives worst propagationdelay .Though the addition is done serially as well as in parallel. In order to improvise on the delay and area the CRAs are replaced with Carry Save Adders, in which every carry and sum signal is passed to the adders of the next stage[G.Vishnu& et all 2016]. Final product is obtained in a final adder by any fast adder (usually carry ripple adder). In array multiplication we need to add, as many partial products as there are multiplier bits.

III.MODIFIED BAUGH- WOOLEY ALGORITHM

I. TwosComplement Multiplication System

The Baugh-Wooley algorithm is the best direct method that is used to compute twos complement multiplication . Moreover it can be also used for normal unsigned multiplication and signed multiplication [Pramodni&et.all 2013]. Here we can see the multiplication cycles with no partial products. The reason being it computes the result in less time and moreover it gives accurate figures. The unsigned multiplication of two 4 bit numbers is shown below:

$a_3\ a_2\ a_1\ a_0$
$x_3\ x_2\ x_1\ x_0$

$a_3x_0a_2x_0a_1x_0a_0x_0$
$a_3x_1\ a_2x_1\ a_1x_1\ a_0x_1$
$a_3x_2a_2x_2\ a_1x_2a_0x_2$
$a_3x_3a_2x_3\ a_1x_3a_0x_3$

$P_7\ P_6P_5P_4\ P_3\ P_2\ P_1P_0$

Fig1. Unsigned Multiplication of 4 bit Numbers

The signed bit extension of baugh- wooley multiplication is shown in the next example where again we have taken 4 bit but the MSBs in each cycle is getting inverted. This clearly shows us that the carry is propagated in each cycle and the partial product generated are being added at last[M.Rambabu&et.all 2016]. This makes the final multiplication to be stored in twos complement in the system.

Not only it stores the value but it also shows regular array pattern formed during the multiplication this thus causes delay of only 4n and gives best result with Type 0 full

adder that are used in the configuration n^2-n-3. In the paper we have used 255 full adders along with 13 and gates to add the propagated carry and 16 half adders.

$a_3\ a_2\ a_1\ a_0$
$x_3\ x_2\ x_1\ x_0$

$-a_3x_0a_2x_0\ \ a_1x_0\ a_0x_0$
$-a_3x_1\ a_2x_1\ \ a_1x_1\ a_0x_1$
$-a_3x_2\ a_2x_2\ \ a_1x_2\ \ a_0x_2$
$a_3x_3-a_2x_3\ -a_1x-a_0x_3$
$P_7P_6\ P_5P_4\ \ P_3\ \ P_2\ P_1P_0$

Fig2. Signed multiplication of 4 Bits Number

B.Baugh-Wooley Algorithm& Architecture

There are many algorithms been devised for twos complement multiplication. But among them the best one is the Baugh-Wooley algorithm as it allows maximum regularity for the multiplier logic and have all partial products with positive signed bits only. The technique was developed basically for direct twos complement multiplication. While multiplying the partial products are added as signed numbers. So each partial product is given signed bit extensions thatgives the final products for correct sum by saving the carry save adders. The addition of extra bit extensions leads to removal ofnegatively generated partial products and thus occupies less area. Since its implementation is done via Type 0 full adder the delay produced is very less comparatively[M.Rambabu&et.all 2016]. The algorithm can be represented with the multiplication shownas:

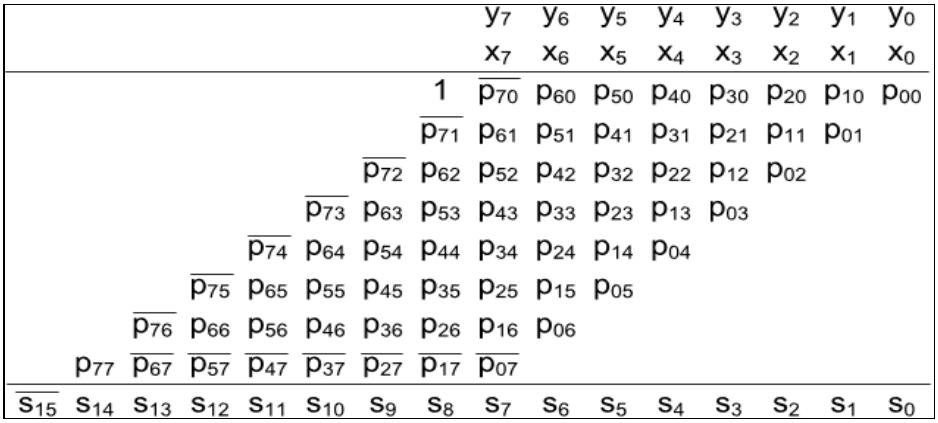

Fig3.Baugh-Wooley Algorithm

The above fig shows the signed multiplication of 8 bit numbers. It shows us the proper algorithm implementation as to how the values are calculated.

IV. RESULT AND IMPLEMENTATION

I. Implementation

For a W bit fixed point twos complement number A

$$A= a_{w-1}.a_{w-2} \ldots\ldots\ldots a_1.a_0 \quad (1)$$

the bits are $0 \leq i \leq W-1$ and the Msb is defined as the sign bit.

While performing a twos complement multiplication negating a number is first taken with 1's complement and then adding a 1 to the MSB

$$-A = a_{w-1} - \sum_{i=1}^{W-1} a_{w-1-i}2^{-i}$$

$$=a_{w-1}+ \sum_{i=1}^{W-1}(1-a_{w-1-i})2^{-i}-\sum_{i=1}^{W-1}2^{-i}$$

$$=-(1-a_{w-1})+\sum_{i=1}^{W-1}(1-a_{w-1-i})\, 2^{-i}+2^{-W-1} \quad (2)$$

For signed bit extension that solves our problem of negative weight in the earlier version of multiplication we can modify it with this statement ahead.

$$A = a_3+ a_2 2^{-1}+a_1 2^{-2}+a_0 2^{-3}$$

$$=-a_3 2+a_3+a_2 2^{-1}+a_1 2^{-2}+a_0 2^{-3} \quad (3)$$

This is the 1's complement signed bit extension which is followed by 2's complement in the next step:

$$A=-a_3 2^2+a_3 2+a_3+a_2 2^{-1}+a_1 2^{-2}+a_0 2^{-3} \quad (4)$$

Thus the final product is obtained as:

$$P = A\times B = a_{n-1}b_{n-1}2^{2n-2} +\sum_{i=0}^{n-2}\sum_{j=0}^{n-2} a_i b_j 2^{i+j} +2n-1\sum_{i=o}^{n-2}(bn-1)'a_1 2^i+2^{n-1}\sum_{j=0}^{n-2}(a_n{-}_1 b_j 2^i) -2^{2n-1}+2^{n} \quad (5)$$

Thus in the implementation we have used 255 Type 0 full adders along with 16 half adders and the remaining carry propagation of the MSBs digits were allowed to travel via a critical path and were being summed by AND gates which have been used in the configuration of 13 lines to give the final results.The typical twos complement multiplication is shown in the next figure where the actual 16 bit multiplication using Baugh-Wooley architecture is performed. For this one number must always be binary.

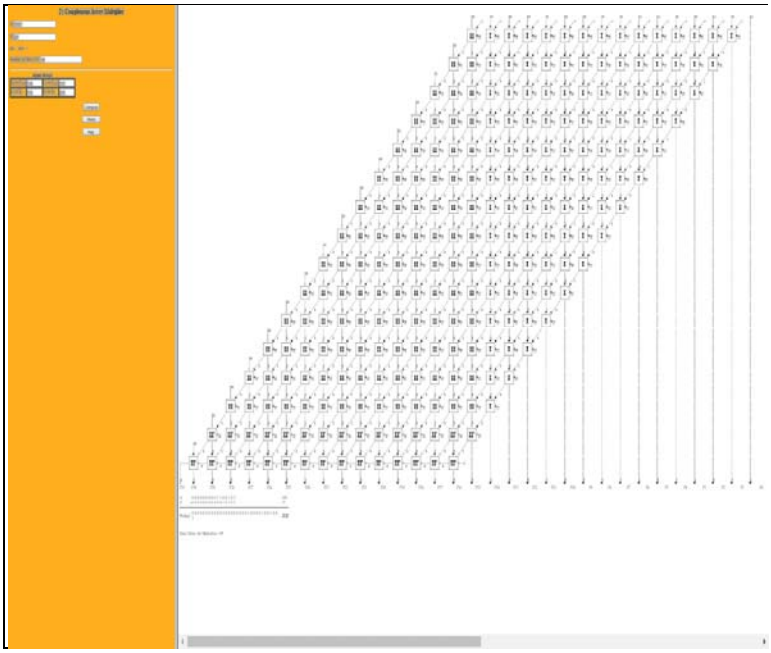

Fig.4 Baugh-Wooley 16 bit array architecture

II. Results

The simulation tool that is used in the paper is the ISE Design Suit for Xilinx with the Version 14.3. The software is compatible with almost all OS. The suit comprises of both the simulation and synthesizable parameters that allows the coding via its accurate implementation and it can be synthesized that implies that it can be run successfully and implemented on a hardware[Pramodni&et.all 2013].The coding for simulation is done in Verilog by implementing a test bench. A test bench is used to realise the characteristics of a system by showing proper simulation results and graphs. It is the most reliable method because it allows us to change the values of the input in the simulation window and can compare more results. The synthesis of the Baugh-Wooley twos complement array multiplier by RTL schematic view & Technology view is shown in the figures ahead with results.

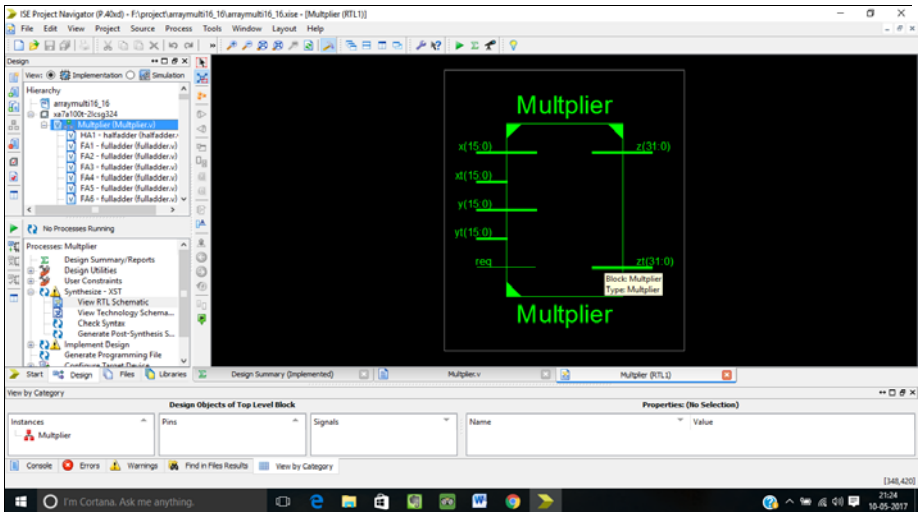

Fig5. Top Module Structure of Multiplier

The next figures show the connection of all the components used like the logical gates , full adders half adders and carry save adder(CSA).Here n^2-n-3 Type 0 full adders are used that allows to maintain the regular layout of the array multiplier[3].

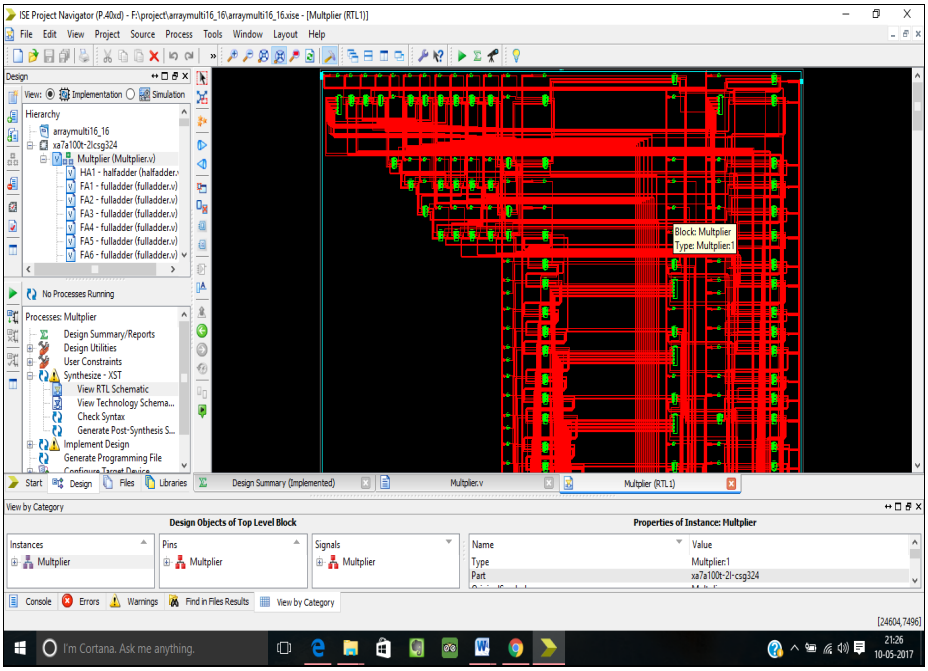

Fig6. Internal Circuit Components

The next figure shows about the hardware compatible IC. This can be synthesised with the external hardware if required. We can then develop the user constraint file that will allow mapping of the software circuitry with the external drive or hardware system. This will generate a configurable IC.

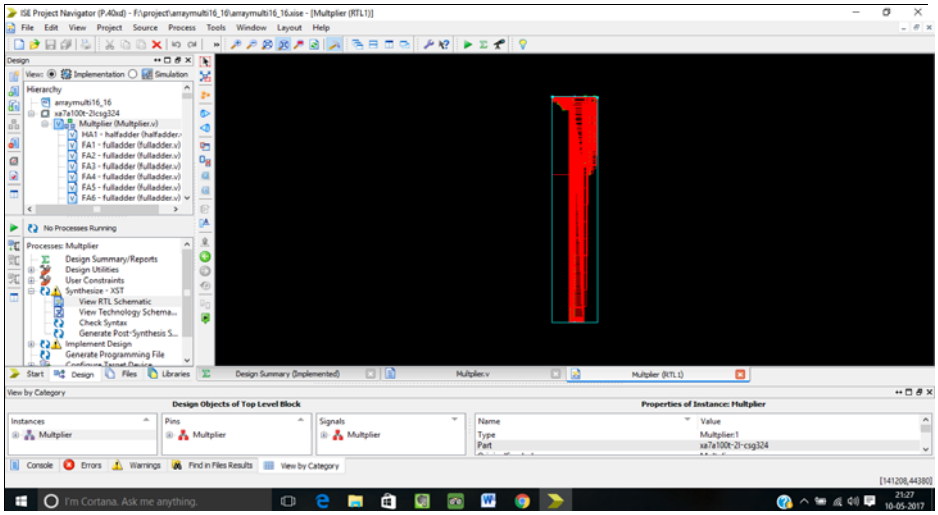

Fig7. The formation of Hardware compatible IC

The next realisation is for the optimization circuit i.e the number of Luts used, clock cycles, multiplier blocks etc. The technology view is shown in the next figure.

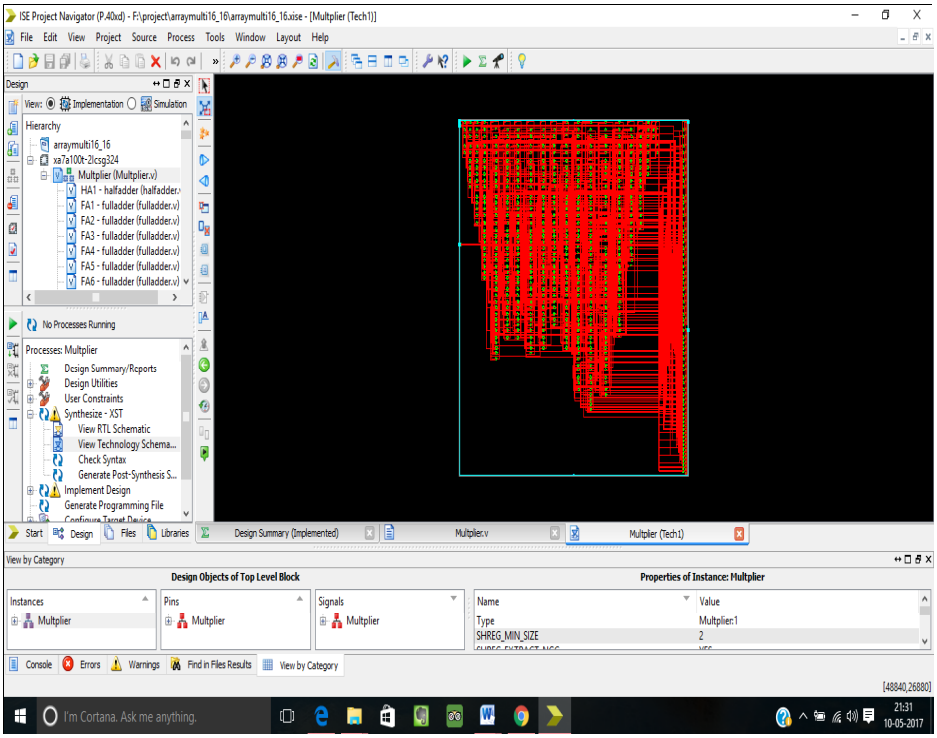

Fig8. Luts& Optimization Circuit

The final simulation results are shown here in that clearly shows that our implementation of 16 bit twos complement array multiplier using Baugh –Wooley algorithm is achieved. The following tables shows about the HDL synthesis Report and the final design report is given in form of tables.

Table1. HDL Synthesis Report

Macrostatstics	
#Tristate	2
# 1 Bit –Tristate Buffer	2
XORs	216
# 1bit Xor 2	16
# 1Bit Xor 3	210
Optimization Goal	Speed
Slice Utilization Ratio	100
BRAM Utilization Ratio	100
Slice Utilization Ratio Delta	5

Table2. Final Design Report

Design Statistics	129
IOs	
Cell Usage BELs	1
Cell Usage GND	1
Total Real time to Xst completion	10.00 sec
Total Real time CPU Xst completion	10.19 sec
Memory Usage	342844 KB
Number of Luts	28 out of 32

The two tables shows the number of cells required in order to achieve best possible optimization along with the slice utilization. This shows us about the number of input output units, real time completion,LUTsused , memory usage . Thus the regular size layout is achieved. The simulation results are shown now in the next block.

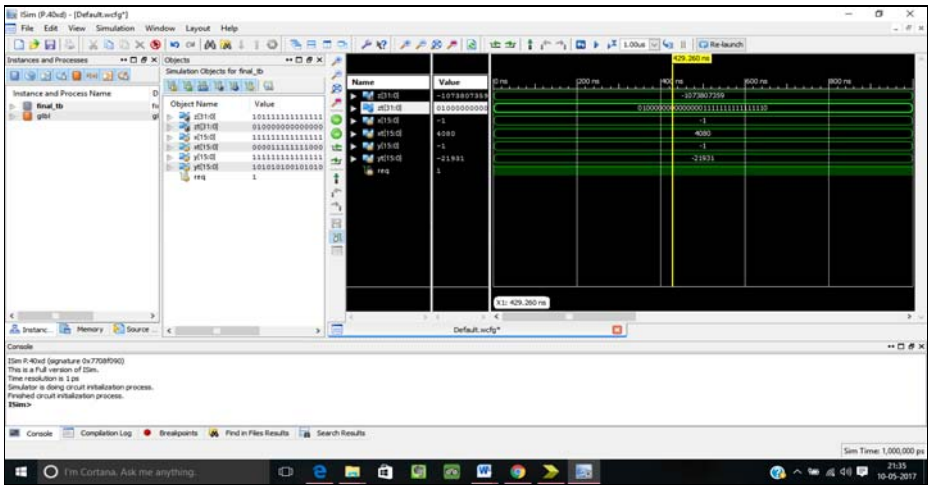

Fig.9 Final Simulation Results

V. CONCLUSION

With the final execution of the results that were desired we come to end of the thesis. In the whole process of implementation we realized that the array multiplier using Baugh-Wooley algorithm and architecture not only gives us optimized results that was the sole operation of the thesis. But eventually we were able to bring the best characteristics of the system with increased speed by more numbers of bits, cost effective circuitry, easy hardware implementation and optimized power consumption.

VI. FUTURE SCOPE

Since the Baugh-Wooley algorithm &architecture allows best optimization and is the direct method used for multiplying two 2's complement number and storing the result . It would be quite helpful to realize the circuit implementation with higher order bits but it must be <32. Thus it will be successfully if implemented with more higher order bits but <32.

ACKNOWLEDGMENT

I wish to record my appreciation for the help and guidance received in preparation of this research paper. I am deeply indebted and express deep sense of gratitude to Mr.AvianshRai, Assistant Professor in Department Of Electronics & Communication. I just appreciate his talent and the encouragement which he provided me throughout and the Rajiv Gandhi ProudyogikiVishwavidyalaya, Bhopal. I sincerely thank the journal for accepting my research work .

References

[1] Abhishek. M and AbhijitAsati (2013) "Generic Modified Baugh WooleyMultiplier".IEEE ICCPST conference.
[2] Balamugan.V&Priyanka. M (2015) " Design& Performance analysis of High Speed MAC using different multipliers ". IEEE, 5thICACC .
[3] C. Wallace "A Suggestion for a fast multiplier", IEEE Transaction on Electronics Computer Vol EC 13 pp 14-17.
[4] G. Vishnu Priya, I. ThairaBanu and S. Shrikant, (2016) "Low Power Array Multiplier using modified Full Adder " IEEE,.
[5] H.Tarunkumar and K. N. Singh (2015) "A review on various multipliers design in VLSI",IEEE,.
[6] H.Tumiyama, I.Tniguchi, S.Yamashita, T. Yamamoto, and Y.H.Azumi (2016) "A ystematic Methodology for Design & Analysis of Approximate Array Multiplier "IEEE,
[7] M. Rambabu and S.N.Begum (2016) "Design and Implementation of 16 bit Baugh Wooley Multiplier with GDI Technology" IJASTEMS.Vol.2
[8] N.A. Reddy and T.Pardu (2015) "Design Of Ultra Low Power Multiplier using Hybrid adder".IEEE, ICCSP conference.
[9] PramodiniMohanty and RashmiRajan (2012) "An Efficient Baugh Wooley Architecture for both Signed and Unsigned Multiplication". IJCSET Vol 3 No.4