International Journal of Scientific Research & Engineering Trends
Volume 5, Issue 4, July-Aug-2019, ISSN (Online): 2395-566X

Implementation on Low Power and Less Area Multiplier Using Adder

M.Tech. Scholar Sarvesh Mani Pandey   Asst. Prof. Anil Khandelwal
Department Electronics and Communication
VNS Group of Institution
Bhopal, MP, India

***Abstract*- The main intention of the paper is to use booth multiplier algorithm for designing the binary multiplier with the help of Ripple carry adder. The reason for using the booth's algorithm is that, using booth's algorithm we can reduce the number of partial products during multiplication. The adder here we have used is ripple carry adder. This adder has a very simple architecture and is very easy to implement. As here we are dealing with high bits, If we see overall including the adder and booth's algorithm we get a binary multiplier which has comparatively high speed because of less partial products and less power consumption because of the adder architecture we have used. The results are obtained by executing VHDL code in Xilinx 14.1 ISE ripple + booth2514 used out of 63400 power dissipated in the design 0.42WA And Total Delay 8.042the area occupied by the intended by calculating the number of LUTs used out of total available LUTs. Here in the intended the number of LUTs used is 1895 out of 63400 LUTs available. Design summary also gives other important information related to intend.**

Keywords –ripple Adder ,Multiplier, booth multiplier ,array multiplier, look ahead adder

I. INTRODUCTION

Speed of the multiplier is highly dependent upon the number of partial products generated and the adder architecture used to add these partial products. The main intention of the project is to use booth multiplier algorithm for designing the binary multiplier with the help of Ripple carry adder.[12][3] The reason for using the booth's algorithm is that, using booth's algorithm we can reduce the number of partial products during multiplication. The adder here we have used is ripple carry adder. This adder has a very simple architecture and is very easy to implement. As here we are dealing with high bits, this adder is very useful because of its simple architecture. If we see overall including the adder and booth's algorithm we get a binary multiplier which has comparatively high speed because of less partial products and less power consumption because of the adder architecture we have used. We have checked

The results for both signed and unsigned numbers. They require for low-power VLSI system arises from two main forces primary, among the steady growth of operating frequency and processing capacity per chip, large currents have to be delivered and the warmth due to large power consumption must be removed by proper cooling techniques. Second, battery life in portable electronic devices is limited. Low power design directly leads to prolonged operation time in these transferable devices. This has given technique to the augmentation of new circuit algorithms, with the plan of reducing the power consumption of multiplication algorithms with having high-speed structures and appropriate performance. The multiplier is fairly large block of a computing system. The standard method of multiplying two *n*-digit numbers requires n^2 multiplications With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets: high speed, low power consumption, less area or even combination of them in one multiplier. However the fact remains that the area and the speed are the two conflicting performance constraints. Hence, innovating increased speed always result in larger area [2][6].

Multiplication possibly will be a a great deal used procedure that statistics obviously in signal development and scientific application Multiplication may be a terribly hardware intensive subject and thus we as users area unit mostly involved by means of get hold of low-power, slighter space and enhanced speed[11][8][9]. The primary required apprehension in classic multiplication for the nearly everyone part accomplished by K-cycles of shifting and adding, is to hurry up underlying multi-operand addition of partial product.

During this thesis we'll design the Booth multiplier using Ripple Carry Adder architecture. Additionally multipliers are designed for each radix-2 and radix-4. Results can show that the multiplier is able to multiply two 32 bit signed numbers and how this technique reduces the number of partial products, which is an important factor to be achieved in this paper The standard method of multiplying two n-digit numbers requires n2 multiplications With advances in technology, many researchers have tried and are trying to design multipliers which offer either of the following design targets high speed, low power consumption, less area or even combination of them in one multiplier. However the fact remains that the area and the speed are the two conflicting performance constraints. Hence, innovating increased speed always result in larger area [7][10]

Overview of Research- This paper focuses on the design and synthesis of efficient binary multiplier architecture for high performance multiplier based on adder. The design makes use of Binary ripple adder the decimal unit has multi operand module. The multiplier has to be an efficient, high speed multiplier, less complexity to achieve high performance. The multiplier can be designed using booth multiplier and approaches [1][5][8] this Paper presents two novel techniques for binary multiplier based on adder. The first approach has multiplier using booth technique. Simultaneously, a novel design for multiplier proposed high performance multiplier based on adder algorithm [15]

II. PROPOSED WORK

The main of our Thesis is design a 32-bit multiplier by using either binary adder based on area, delay time and power required for the multiplier. Digital computer arithmetic is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve an efficient utilization of the available hardware. The basic operation of additions implemented to the operation of multiplication. Multiplications and additions are most widely used arithmetic computations performed in all digital signal processing applications. In this project we are going to the performance of different adders implemented to the multipliers based on area and time needed for calculation.

III. VHDL SIMULATION

The VHDL simulation of the multiplier is presented waveforms, timing diagrams and the design précis for both the ripple adder and based multipliers. The VHDL code for both multipliers, using binary adder, are generated. A design and implementation of a VHDL-based 32 bit unsigned multiplier with ripple and adder was presented. VHDL was used to model and simulate our multiplier. Using binary Adder improves the overall performance of the multiplier. Thus a 72 % area delay product reduction is possible with the use of the Ripple Adder based 32 bit unsigned parallel multiplier than binary adder based 32 bit unsigned parallel multiplier. The main motive of the proposed algorithm is to execute significantly that should be capable of executing fixed bit binary multiplier in terms of power and area. Most of the newly proposed multiplier use 7-bit booth multiplier. The proposed algorithm has been intentionally designed for such converters. The following section explains the proposed algorithm used in this paper

Performance Analysis Area Analysis The performance analysis for the area of ripple adder based multipliers are represented in the form of the diagram binary adder Area analysis

Delay Analysis-The performance analysis for the delay time of binary adder based multipliers is represented in the form of the diagram. Delay analysis

Area Delay Product Analysis:- The performance analysis for the area delay product of, ripple adder ,look ahead adder , based multipliers are represented in the form of the diagram. Area delay product analysis chart

IV. IMPLEMENTATION

The main logic of Ripple Adder is to compute alternative results in parallel and subsequently selecting the correct result by using mux according to the control bit. In Ripple Adder both sum and carry bits are calculated for two alternatives C_{in}=O and 1. Once C_{in} is obtained, the correct computation is taken using a mux to produce the actual output. Instead of waiting for C_{in} to calculate the sum, the sum is correctly output as soon as C_{in} gets there. The extra time taken to compute the sum (because of propagation delay) is then avoided which results in good improvement in speed.

The architecture of multiplier had shown in Figure 2formerly the each and every one VHDL module are ready, they be supposed to be simulated previous to they are put in actual hardware chip. We can produce a test oppose waveform from the plan new-fangled Source bill of fare of ISE and it will hold up in surrounds up the simulation. one time we replicate our intend and experience it is presentation appropriately and then we can shift on to generate the data needed to fundamentally program the principle tool through our arrangement design.

1.Multiplier Implementation -Completing of multiplier and RTL have been completed using Xilinx 14.1 and simulator has mention out by ISim 14.1e tool. Show in fig. 1

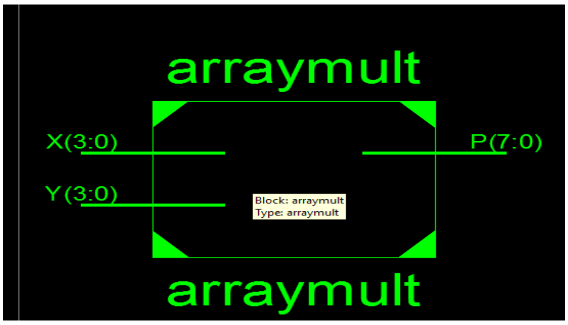
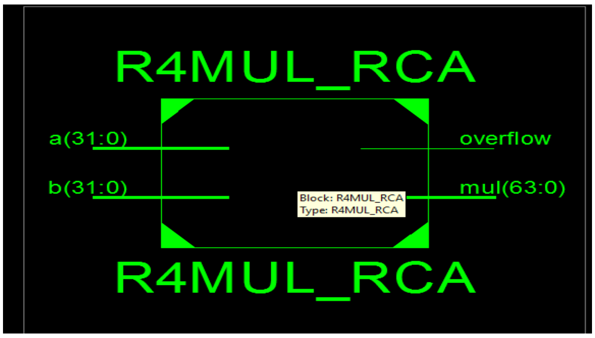


Fig.1. RTL View of Multiplier

Fig 3 Shows Resistor Transistor Logic view of Multiplier in which multiplicand ,multiplier, clk, **a**nd product is output .Fig.4 represnts the various combination of multiplexers/ adders /ff and flip flops and I/O lines related to LUTs and combination of various multiplier unit connected to lines.By double clicking on any of the Fig 2 Shows Resistor Transistor Logic view of Multiplier in which multiplicand, multiplier, clk, **a**nd product is output.



\

Fig..2. RTL View of adder/Multiplier

Fig.3 represents the various combination of multiplexers/ adders /ff and flip flops and I/O lines related to LUTs and combination of various multiplier unit connected to lines. By double clicking on any of the

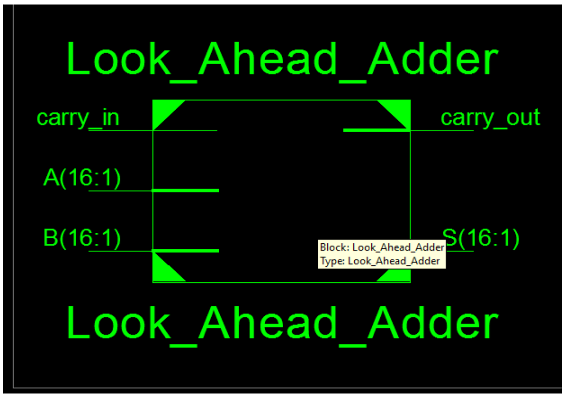


Fig.3. RTL View of look ahead adder

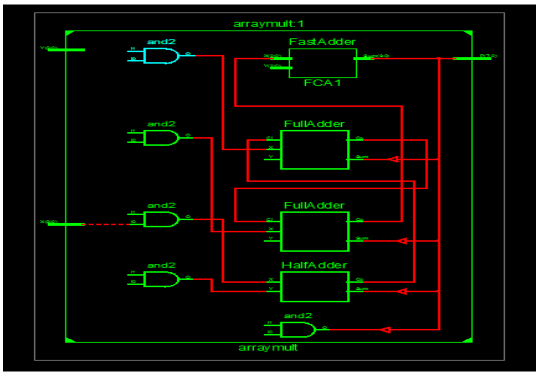


Fig.4 RTL View of adder/Multiplier.

Fig.4 demonstrate the multiplier top level view of sequential output like counter and flip flop which counts signal propogation and show I/O lines of mux and flip flops adders.

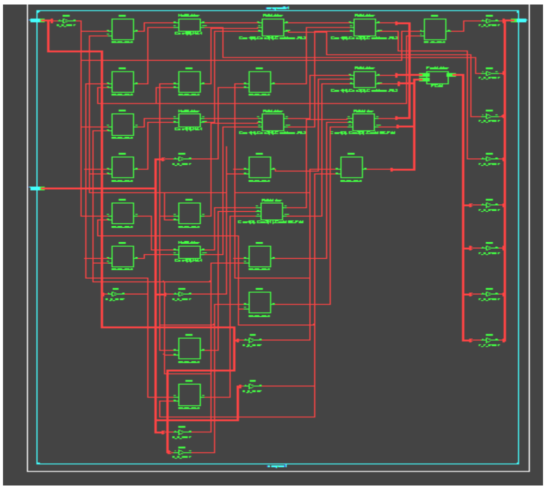


Fig.5. RTL View of adder/Multiplier

Fig.6 demonstrate the multiplier top level view and LUTs of sequential output like counter and flip flop which counts signal propogation and show I/O lines of mux and flip flops adders.

\

Fig.6 RTL View of adder/Multiplier.

Fig.6 demonstrate the multiplier top level view of sequential output like counter and flip flop which counts signal propogation and show I/O lines of mux and flip flops adders.

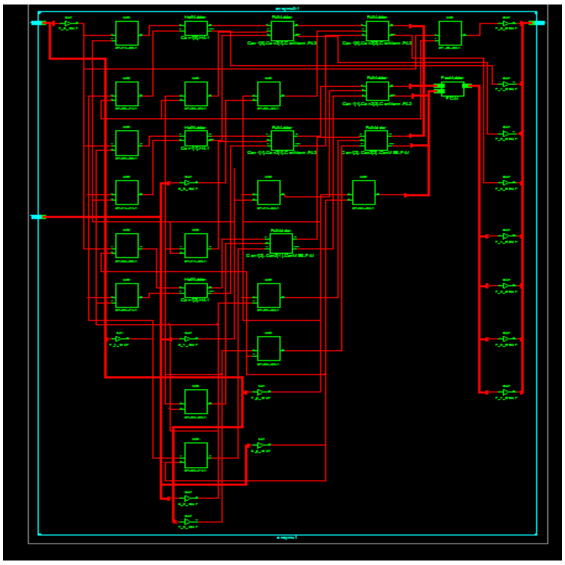


Fig.7 RTL View of adder/Multiplier.

Fig.7 demonstrate the multiplier top level view and LUTs of sequential output like counter and flip flop which counts signal propogation and show I/O lines of mux and flip flops adders.

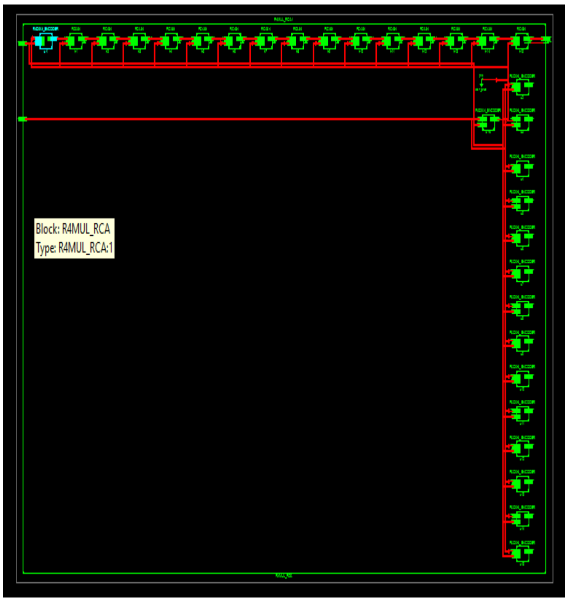


Fig 8 transient View of adder/Multiplier.

Which is show logic equvation of combination logic and karnaugh map of binary multiplier in show in fig.7 Fig illustrate of truth table of multiplier and shows look-up table entries of combinational logic of various digital logic used in multiplier Fig 9 illustrate of truth table of multiplier and shows look-up table entries of combinational logic of various digital logic used in multilier implimentation

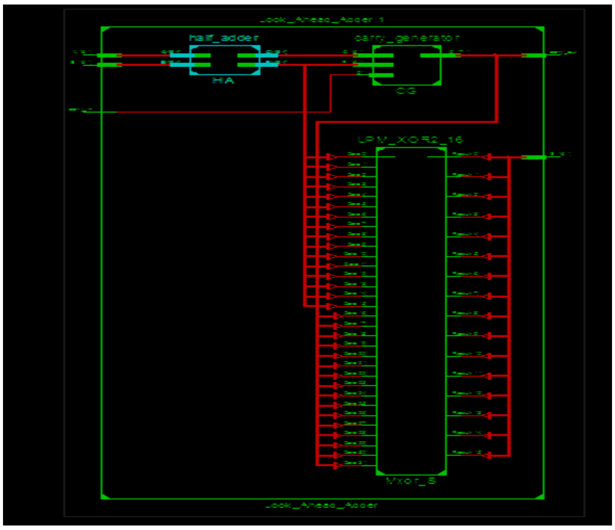
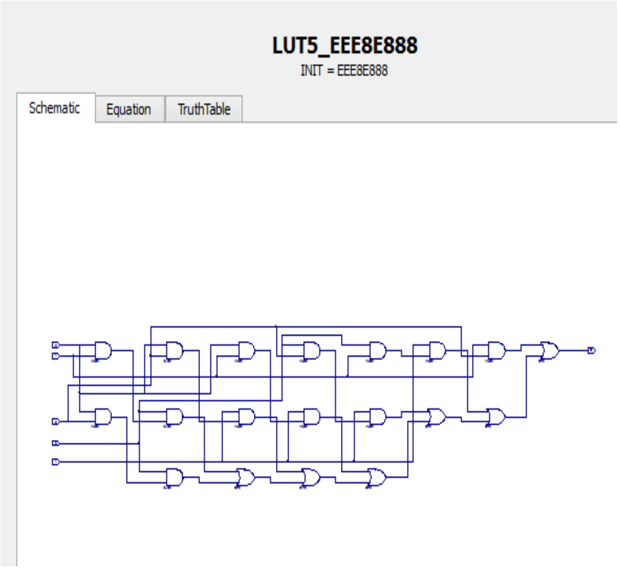


Fig.9 RTL View of adder/Multiplier.



LUT5_EEE8E888

INIT = EEE8E888

Fig. 10 LUTs of adder/Multiplier.

Fig 10 illustrate combination logic diagram of multiplier converter. Figure illustrate the combinational logic of AND OR logic states Which is show logic equvation of combination logic and karnaugh map of binary multiplier in show in fig.10 Fig 11 illustrate of truth table of multiplier and shows look-up table entries of combinational logic of various digital logic used in multiplier Fig 12 illustrate of truth table of multiplier and shows look-up table entries of combinational logic of various digital logic used in multilier implimentation

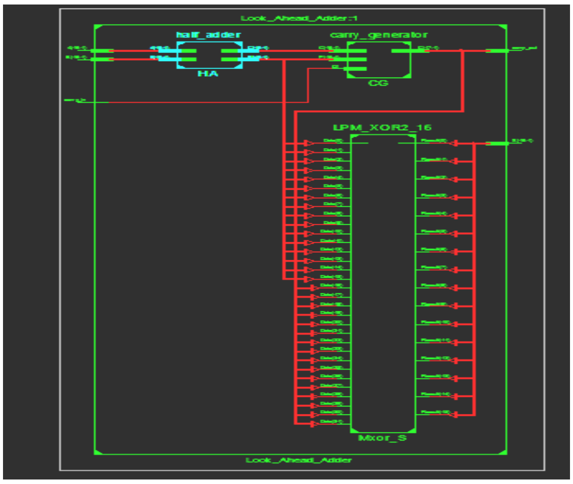


Fig.12 RTL View of lookahaed adder.

V. SIMULATION PROCESS AND RESULTS

In this work, Xilinx and I-sim tools are used for timing analysis and synthesis. The simulation output for multiplier - is presented. After verifying the block diagram, the behavior of multiplier is checked by simulation process.

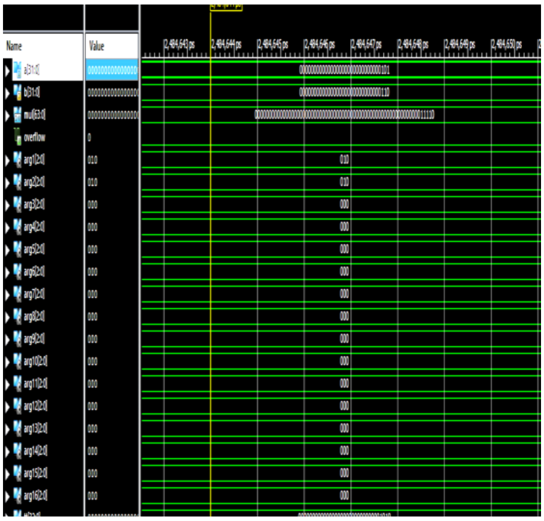


Fig.13. simulation wave form of the multiplier

In this work, Xilinx and I-sim tools are used for timing analysis and synthesis. The simulation output for multiplier is presented. After verifying the block diagram, the behavior of multiplier is checked by simulation process.

1. Power Analysis- Multiplier is a modest additional but it provides option for addition with bulky number of input and also in more efficient way in expression of area and power . So proposed work is very efficient and useful in higher level addition with increasing numeral of inputs with the rising difficulty in computation application which takes lesser time and less power consumption given away in fig 15 Power analysis figure of the intend shows the total power consumed by the design in tenure of leakage current and also tell about the thermal properties and supply power.

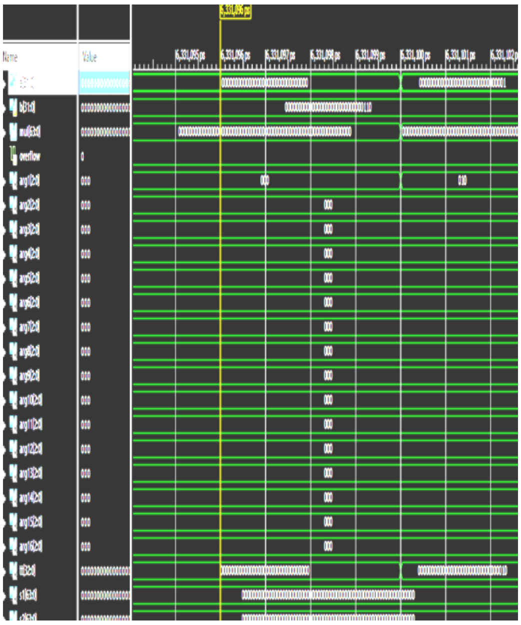
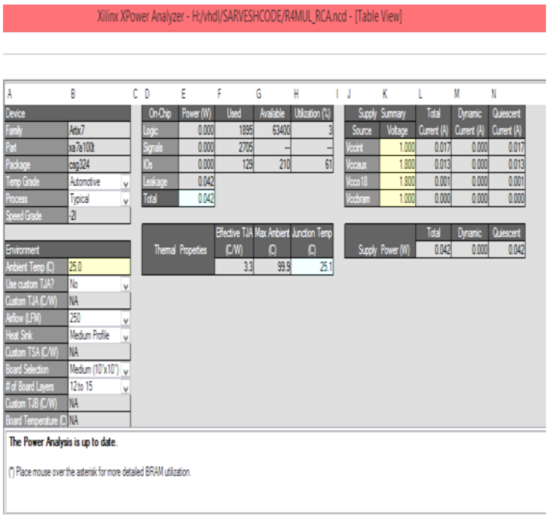


Fig.14. simulation wave form of the multiplier.

Of the intend. Figure below shows all the output parameters related to power analysis of the intend is shown the fig 15


Xilinx XPower Analyzer - H:/vhdl/SARVESHCODE/R4MUL_RCA.ncd - [Table View]

Device	
Family	Artix7
Part	xa7a100t
Package	csg324
Temp Grade	Automotive
Process	Typical
Speed Grade	-2I

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
Custom TJB (C/W)	NA
Board Temperature (C)	NA

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	1895	63400	3
Signals	0.000	2705	--	--
IOs	0.000	129	210	61
Leakage	0.042			
Total	0.042			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	3.3	99.9	25.1

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.017	0.000	0.017
Vccaux	1.800	0.013	0.000	0.013
Vcco18	1.800	0.001	0.000	0.001
Vccbram	1.000	0.000	0.000	0.000

	Total	Dynamic	Quiescent
Supply Power (W)	0.042	0.000	0.042

The Power Analysis is up to date.

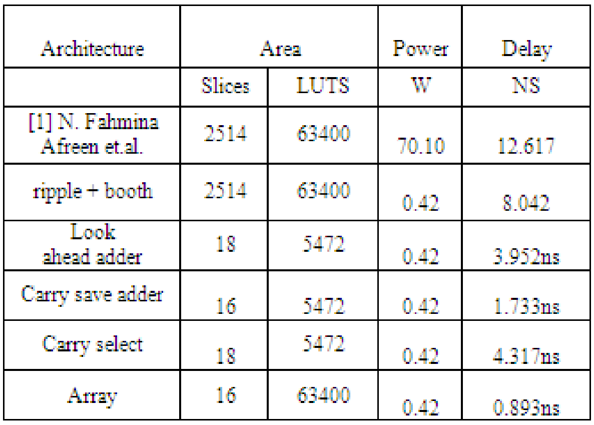
(*) Place mouse over the asterisk for more detailed BRAM utilization.

Fig.15 Shown Power expenditure of multiplier.

2. Design Summary- Table represents the summary of whole simulation of multiplier **.** Which represents the delay and time of whole combination logic of multiplier design and all the combination logic of multiplier organization is explained using VHDL data flow modeling and run using simulator (Isim) 14.7multiplier stand on the suggested algorithm is designed in the proposed algorithm.

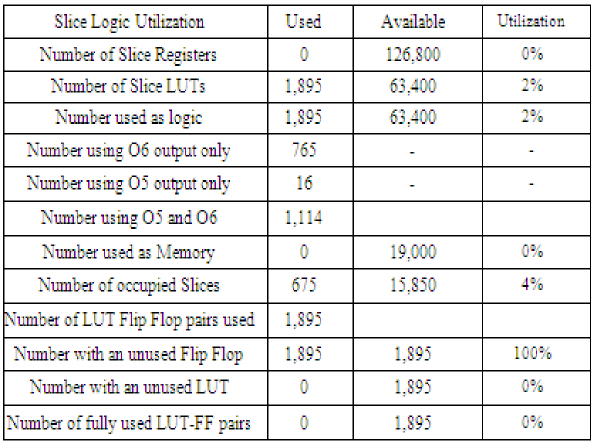
Design summary demonstrate the area occupied by the intended by calculating the number of LUTs used out of total available LUTs. Here in the intended the number of LUTs used is 1895 out of 63400 LUTs available. Design summary also gives other important information related to intend. This is shown below in table .1

Table .1 Design Summaries



Architecture	Area		Power	Delay
	Slices	LUTS	W	NS
[1] N. Fahmina Afreen et.al.	2514	63400	70.10	12.617
ripple + booth	2514	63400	0.42	8.042
Look ahead adder	18	5472	0.42	3.952ns
Carry save adder	16	5472	0.42	1.733ns
Carry select	18	5472	0.42	4.317ns
Array	16	63400	0.42	0.893ns

Table 2 Synthesis Result Comparison.



Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	0	126,800	0%
Number of Slice LUTs	1,895	63,400	2%
Number used as logic	1,895	63,400	2%
Number using O6 output only	765	-	-
Number using O5 output only	16	-	-
Number using O5 and O6	1,114		
Number used as Memory	0	19,000	0%
Number of occupied Slices	675	15,850	4%
Number of LUT Flip Flop pairs used	1,895		
Number with an unused Flip Flop	1,895	1,895	100%
Number with an unused LUT	0	1,895	0%
Number of fully used LUT-FF pairs	0	1,895	0%

VI. CONCLUSIONS

The paper that square measure to implement Booth's formula for the look of a binary multiplier factor mistreatment ripple carry adder design. in any case we have a tendency to came to a conclusion that Ripple Carry Adders square measure best suited to our Applications. Then we have a tendency to turned our focus into the look of Multipliers. initial of all we have a tendency to designed a Booth's Radix-4 multiplier factor. If we have a tendency to comparison information between Radix-2 and Radix-4 booth multipliers we have a tendency to noted that radix-4 consumes less power than radix-2, as a result of radix-4 uses virtually a 0.5 variety of iterations than radix-2. As radix-4 appeared a lot of appropriate for the look we have a tendency to dispensed additional analysis on radix-4 multiplier factor by mistreatment ripple carry adder design.

REFERENCES

1. N. Fahmina Afreen Design And Implementation Of Area- Delay Power Efficient CSLA Based 32-Bit Array Multiplier 2017 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology (RTEICT), May 19-20, 2017, India 978-1-5090-3704-9/17/$31.00 © 2017 IEEE
2. Shuchi Nagaria Anushka Singh Efficient FIR Filter Design using Booth Multiplier for VLSI Applications 2018 International Conference on Computing, Power and Communication Technologies (GUCON)Year: 2018 Pages: 581 – 584
3. Squarers H Haritha Design of an Enhanced Array Based Approximate Arithmetic Computing Model for Multipliers and S R Ramesh2017 14th IEEE India Council International Conference (INDICON)Year: 2017Pages: 1 – 5
4. Weiqiang Liu Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing IEEE Transactions on Computers Year: 2017 Volume: 66 , Issue: 8Pages: 1435 - 1441Cited by: Papers (19)2
5. SomnathRakshi T A novel reversible synthesis of array multiplier 2018 International Symposium on Devices, Circuits and Systems (ISDCS)Year: 2018Pages: 1 – 4
6. S. Srikanth I. ThahiraBanu Low power array multiplier using modified full adder; 2016 IEEE International Conference on Engineering and Technology (ICETECH)Year: 2016Pages: 1041 – 1044 Cited by: Papers (1)
7. .P.V. Sridevi K. Babulu Low Power and High Speed Optimized 4-bit Array Multiplier Using MOD-GDI Technique Pinninti Kishore ; 2017 IEEE 7th International Advance Computing Conference (IACC) Year: 2017 Pages: 487 – 49
8. Anderson Martins Optimal combination of dedicated multiplication blocks and adder trees schemes for optimized radix-2m array multipliers realization 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS) Year: 2015 Pages: 352 – 355
9. Junli Liang Sparse Array Beampattern Synthesis via Alternating Direction Method of MultipliersIEEE Transactions on Antennas and Propagation Year: 2018 Volume: 66 , Issue: 5 Pages: 2333 – 2345 Cited by: Papers
10. R Balakumaran ; E Prabhu Design of high speed multiplier using modified booth algorithm with hybrid carry look-ahead adder2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT) Year: 2016Pages: 1 – 7 Cited by: Papers (10)

11. SakshiRajput ;Rohit Shukla Implementation of High Speed and Low Power Hybrid Adder Based Novel Radix 4 Booth Multiplier 2013 International Conference on Communication Systems and Network Technologies Year: 2013 Pages: 741 – 743
12. J. M. Rudagi, V. Amblr, V. Munavalli, R. Patil, V. Sajjan, “Design and implementation of efficient multiplier using Vedic Mathematics,” in proc. IEEE International Conference on Advances in Recent Technologies in Communication and Computing, November 2011, pp. 162-166.
13. T. Arunachalam, S. Kirubaveni, “Analysis of High Speed Multipliers,” in proc. IEEE International Conference on Communication and Signal Processing, April 2013, pp. 211-214.
14. S. Swee, V. Elakya, “Design of modified low power Booth multiplier,” in proc. IEEE International Conference on Computing, Communication and Applications, June 2011, pp. 854-859.
15. K. S. Yeo, K. Roy, Low Voltage, Low Power VLSI, Tata Mcgraw-Hill Education, 2nd edition.