DOI: 30.5232/ijearst.35.72.4973 *https://ijearst.co.in/*

ENGINEERING IN ADVANCED RESEARCH SCIENCE AND TECHNOLOGY

ISSN 2386-7876
Vol.01, Issue.03
March-2026
Pages: 356-364

SCALABLE AND AREA EFFICIENT LATENCY OPTIMIZED ALU DESIGN

[1] **K.Jyothirmai,**, [2] **Ramanaidu**, [3] **Deepthi**, [4] **A.Nandini**, [5] **K.Rakesh**
[1]M.Tech, Dept of E.C.E, BVCITS, Batlapalem, Amalapuram, AP
[2,3,4,5]B. Tech, Dept of E.C.E, BVCITS, Batlapalem, Amalapuram, AP

ABSTRACT: Digital design is an amazing and very broad field. The applications of digital design are present in our daily life, including Computers, calculators, video cameras etc. In fact, there will be always need for high speed and low power digital products which makes digital design a future growing business. ALU (Arithmetic logic unit) is a critical component of a microprocessor and is the core component of central processing unit. Furthermore, it is the heart of the instruction execution portion of every computer. ALU's comprise the combinational logic that implements logic operations, such as AND and OR, and arithmetic operations, such as ADD and multiplication. We designed an 32-bit ALU (Arithmetic logic unit) that is formed by combining both adder and multiplier using modified square root carry select adder and radix8 modified booth encoding algorithm.

Keywords: Arithmetic Logic Unit, Modified booth encoding, carry select adder, Very large scale Integration.

INTRODUCTION As wireless communication and mobility of equipment become increasingly desirable, power dissipation of circuits has become a major concern in circuit synthesis. In performance driven synthesis of VLSI circuits, low-power design has joined the ranks of area and delay as major motivations in optimization. Digital circuit has rapidly evolved over the last twenty five years .The earliest digital circuits were designed with vacuum tubes and transistors. Integrated circuits were then invented where logic gates were placed on a single chip. The first IC chips were small scale integration (SSI) chips where the gate count is small. When technology became sophisticated, designers were able to place circuits with hundreds of gates on a chip. These chips were called MSI chips with advent of LSI designers could put thousands of gates on a single chip. At this point, design process is getting complicated and designers felt the need to automate these processes. For these reasons, we designed an ALU, which is the main part of any processor. CPU works as brain of any system and it consists of fast dynamic logic circuits and have carefully optimized structures. Of total power consumption in any processor, CPU accounts a significant portion of it. ALU also contribute to one of the highest power-density locations on the processor, as it is clocked at the highest speed and is busy mostly all the time which results in thermal hotspots and sharp temperature gradients within the execution core. Therefore this motivate us strongly for a energy-efficient ALU designs that satisfy the high-performance requirements, while reducing peak and average power dissipation. Basically ALU is a combinational circuit that performs arithmetic and logical operations on a pair of n bit operands for 8 bits. In the era of growing technology and scaling of devices up to nanometer regime, the arithmetic logic circuits are to be designed with compact size, less power and propagation delay. Arithmetic operations are indispensable and basic functions for any high speed low power application digital signal processing, microprocessors, image processing etc. Addition is most important part of the arithmetic unit rather approximately all other arithmetic operation includes addition. Thus, the primary issue in the design of any arithmetic logic unit is to have low power high performance adder cell. There are various topologies and Methodologies proposed to design full adder cell efficiently. This paper utilizes the concept of GDI technique in the design of ALU and its sub blocks as Multiplexers and Full adder.

POWER DISSIPATION IN CMOS CIRCUITS:

The CMOS power dissipation has become a very hot topic during the last decade or so. The number of battery-powered hand-held applications, e.g. mobile phones and laptop computers is steadily increasing and more and more functions are integrated into the systems, e.g. multi-media applications in mobile phones. This is one of the driving

forces for analysis of the mechanisms of power dissipation and power-reduction techniques. Another driving force is the incredible power dissipation of state-of-the-art microprocessors where heat removal and current delivery are very hard and expensive to accomplish.

ALU DESIGN:

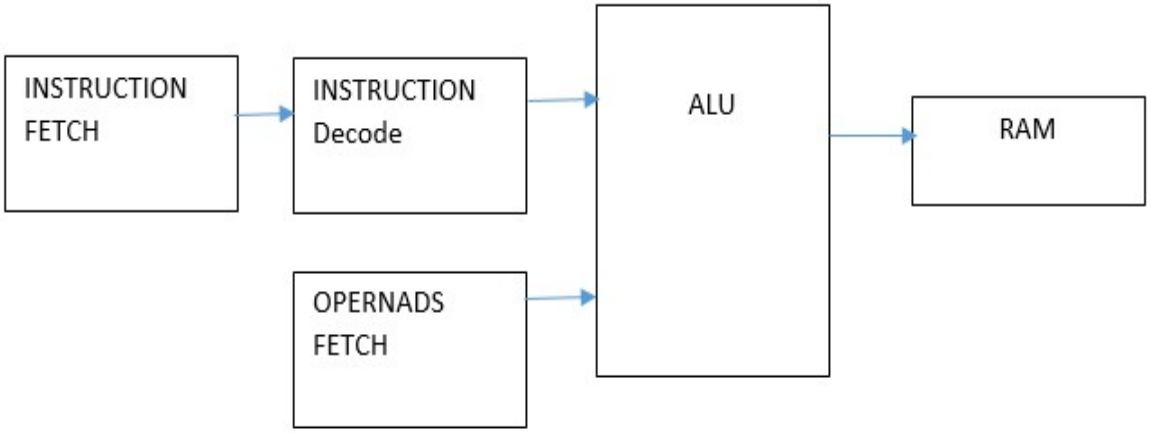
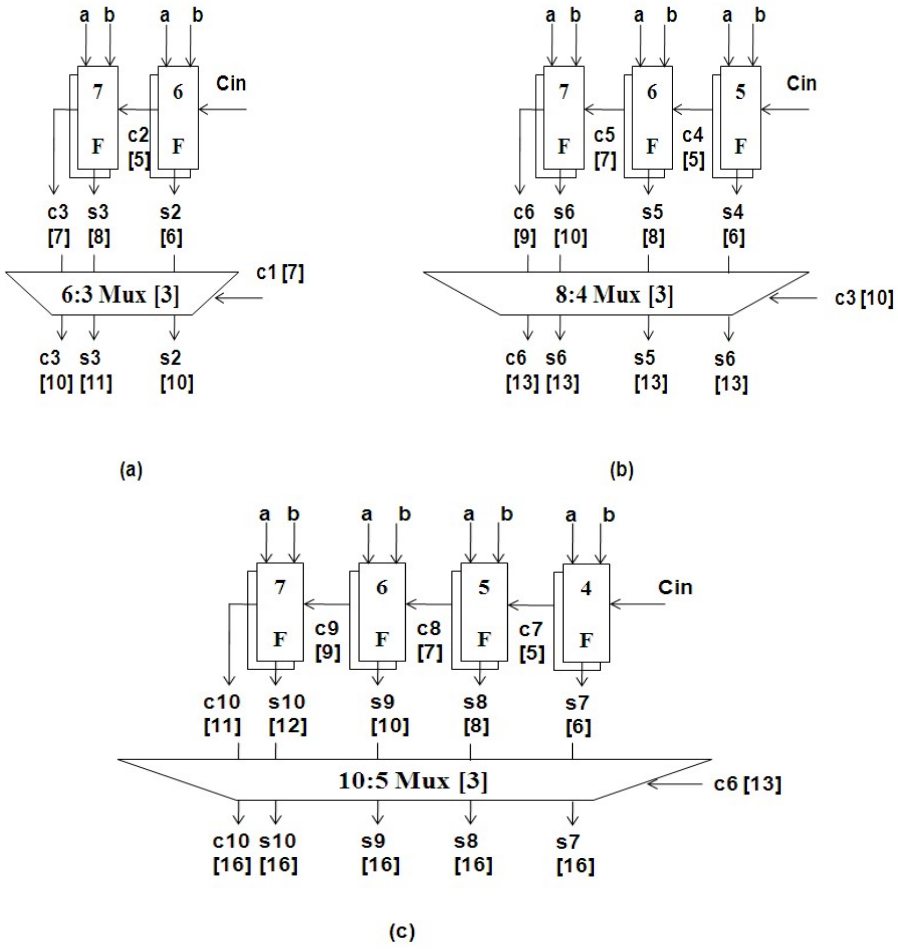
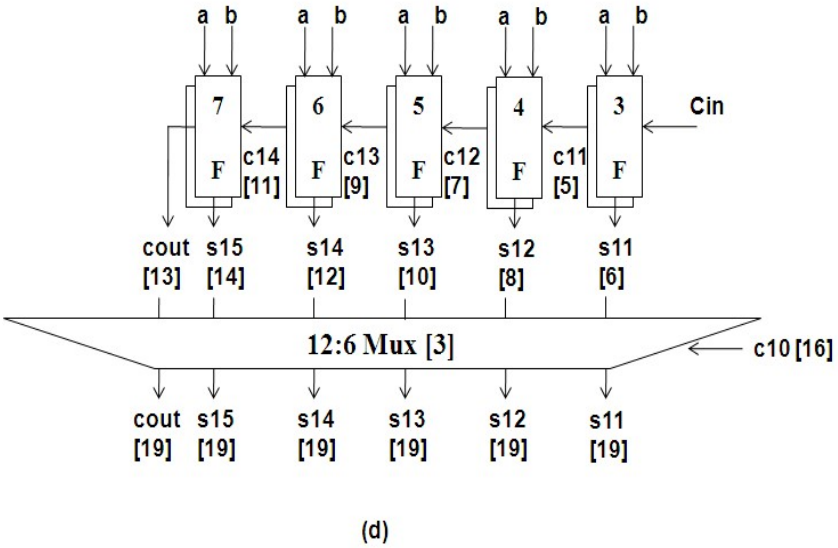


Fig1: Proposed block diagram

MODIFIED SQUARE ROOT CARRY SELECT ADDER WITH BEC





(d)

Fig2: Delay and area evaluation of regular SQRT CSLA: (a) group2, (b) group3, (c) group4 and (d) group5. F is a Full Adder.

This 16-bit square root CSLA consists of five groups where each group is of variable size. The 16-bit value data is divided as 2-bit, 2-bit, 3-bit, 4-bit, 5-bit groups. The first group consists of 2-bit ripple carry adder. The actual input carry is applied to this adder. The ripple carry adder receives the carry and performs the 2 2-bit addition (a[1:0], b[1:0]). The 2-bit sum generated from this adder is written as sum[1:0]. The carry generated by this adder is propagated to the next group with a delay. This delay is calculated using the basic circuit shown in Fig

Delay and Area Evaluation Methodology of the basic adder blocks:

The AND, OR and Inverter (AOI) implementation of an XOR gate is shown in the figur. The gates between the dotted lines are performing the operations in parallel and the numeric representation of each gate indicates the delay contributed by that gate.

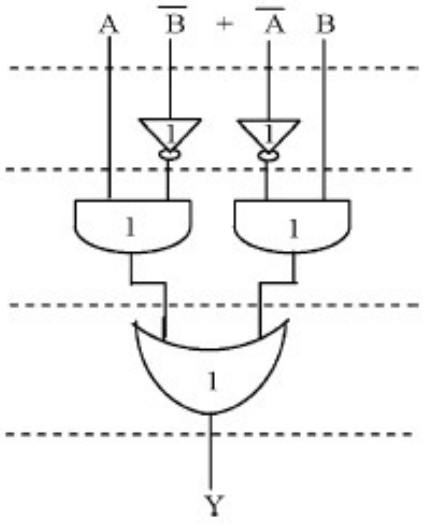


Fig3: Delay and Area evaluation of an XOR gate

The delay and area evaluation methodology considers all gates to be made up of AND, OR and Inverter, each having delay equal to 1 unit and area equal to 1 unit.

Then the number of gates, in the longest path of a logic block,is added that contributes to the maximum delay.

The area evaluation is done by counting the total number of AOI gates required for each logic block. Based on this approach, the CSLA adder blocks of 2:1 mux, Half Adder (HA) and FA are evaluated and listed in Table

Adder Blocks	Delay	Area
XOR	3	5
2:1 Mux	3	4
Half adder	3	6
Full adder	6	13

Table 1.1: Delay and area count of the basic blocks of CSLA

Delay and Area Evaluation of CSLA groups:

The structure of the 16-b regular SQRT CSLA is shown in the figure 3.5. Ithas five groups of different size RCA. The delay and area evaluation ofeach group are shown in Fig. 5, in which the numerals within [ ] specifythe delay values, e.g., sum2 requires 10 gate delays. The steps leadingto the evaluation are as follows.

1) The group2 has two sets of 2-b RCA. Based onthe consideration of delay values of Table 3.1, the arrival time ofselection input c1[time(t)= 7] of 6:3 mux is earlier than s3[t=8] and later than s2[t=6]. Thus, sum3[t=11]is summation of s3 and mux[t=3] and sum2[t=10] is summation of c1 and mux.

2) Except for group2, the arrival time of mux selection input is alwaysgreater than the arrival time of data outputs from the RCA's.Thus, the delay of group3 to group5 is determined, respectively asfollows:

{c6, sum [6:4]}=c3 [t=10] +mux

{c10, sum [10:7]}=c6 [t=13] +mux

{Cout, sum [15:11]}=c10 [t=16] +mux

3) The one set of 2-b RCA in group2 has 2 FA forCin=1 and theother set has 1 FA and 1 HA for Cin=0.

Based on the area countof Table 3.1, the total number of gate counts in group2 is determinedas follows:

Gate count =57 (HA+FA+Mux)

FA=39(3*13)

HA=6(1*6)

Mux=12(3*4)

4) Similarly, the estimated maximum delay and area of the othergroups in the regular SQRT CSLA are evaluated and listed.

The area and delay values of all the groups of square root CSLA are shown in the Table 3.2.

Group	Delay	Area
Group2	11	57
Group3	13	87
Group4	16	117
Group5	19	147

Table2: Delay and area count of regular SQRT CSLA groups

BEC stands for Binary to Excess-1 Converter.

Design of CSLA with BEC As stated above, the main idea of this work is to use BEC instead ofthe RCA with Cin=1 in order to reduce the area and power consumptionof the regular CSLA. To replace the n-bit RCA, an (n+1)-bit BECis required.

The structure and the function table of a 4-b BEC are shownin the figure and Table respectively.

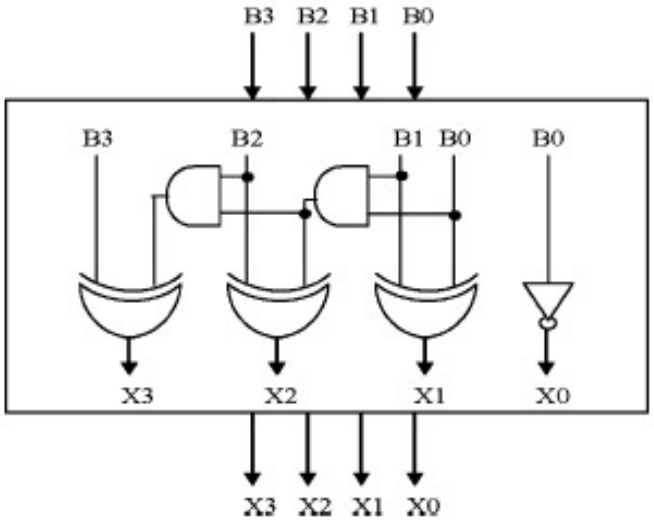


Fig:4 Structure of 4-bit BEC

The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed.

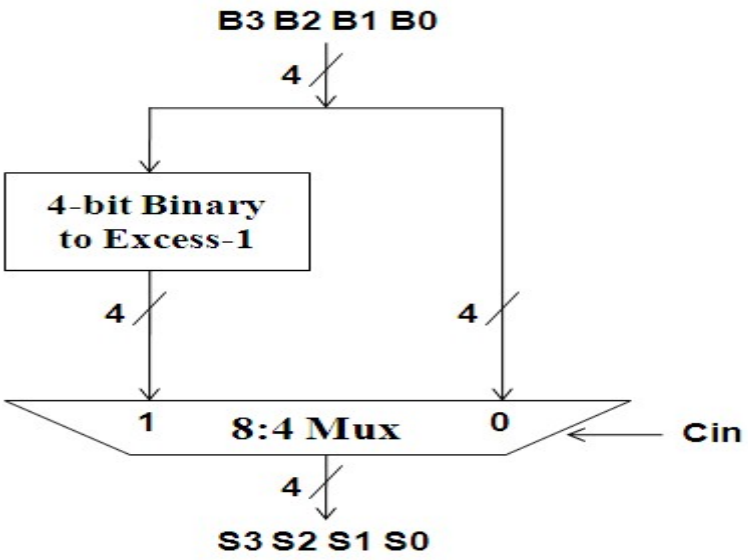


Fig5 4-b BEC with 8:4 Mux

Block diagram of CSLA with BEC:

The structure of the proposed 16-b SQRT CSLA using BEC for RCAwithCin=1 to optimize the area and power is shown in the figure 3.9.

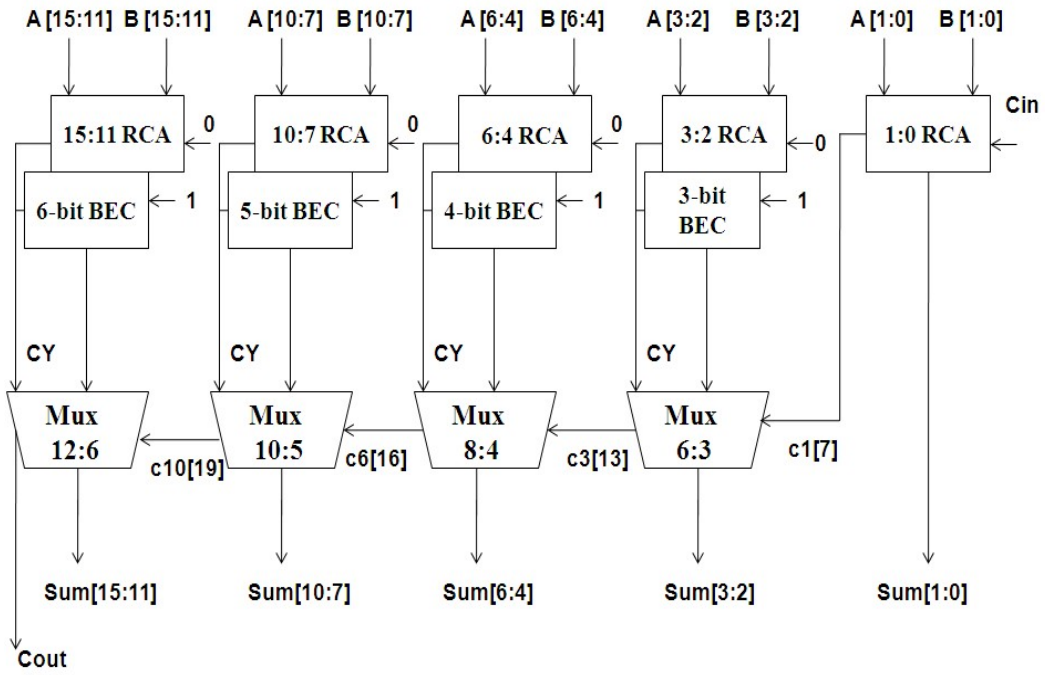


Fig6: Modified system (Modified 16-b SQRT CSLA)

Comparing the block diagram of the regular square root CSLA with the modified square root CSLA, it can be seen that the RCA with Cin=1 is replaced by BEC (binary to excess-1 converter). This is done to reduce the area consumption. This can be seen after evaluating the group delay and the number of gates required for the design.

MULTIPLICATION USING RADIX8 MODIFIED BOOTH

The Booth algorithm consists of repeatedly adding one of two predetermined values to a product P and then performing an arithmetic shift to the right on P.

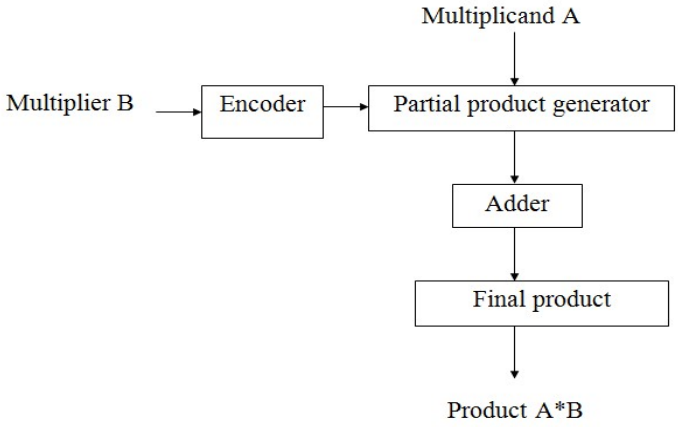


Fig7. Booth algorithm

The multiplier architecture consists of two architectures, i.e., Modified Booth. By the study of different multiplier architectures, we find that Modified Booth increases the speed because it reduces the partial products by half. Also, the delay in the multiplier can be reduced by using Wallace tree. The energy consumption of the Wallace Tree multiplier is also lower than the Booth and the array. The characteristics of the two multipliers can be combined to produce a high-speed and low-power multiplier.

The modified stand-alone multiplier consists of a modified recorder (MBR). MBR has two parts, i.e., Booth Encoder (BE) and Booth Selector (BS). The operation of BE is to decode the multiplier signal, and the output is used by BS to produce the partial product. Then, the partial products are added to the Wallace tree adders, similar to the carry-

save-adder approach. The last transfer and sum output line are added by a carry look- ahead adder, the carry being stretched to the left by positioning.

Table . Quartet coded signed-digit table

Quartet value	Signed-digit value
0000	0
0001	+1
0010	+1
0011	+2
0100	+2
0101	+3
0110	+3
0111	+4
1000	-4
1001	-3
1010	-3
1011	-2
1100	-2
1101	-1
1110	-1
1111	0

Here we have a multiplication multiplier, 3Y, which is not immediately available. To Generate it, we must run the previous addition operation: 2Y + Y = 3Y. But we are designing a multiplier for specific purposes and then the multiplier belongs to a set of previously known numbers stored in a memory chip. We have tried to take advantage of this fact, to relieve the radix-8 bottleneck, that is, 3Y generation. In this way, we try to obtain a better overall multiplication time or at least comparable to the time, we can obtain using a radix-4 architecture (with the added benefit of using fewer transistors). To generate 3Y with 21-bit words you just have to add 2Y + Y, ie add the number with the same number moved to a left position.

A product formed by multiplying it with a multiplier digit when the multiplier has many digits. Partial products are calculated as intermediate steps in the calculation of larger products.

The partial product generator is designed to produce the product multiplying by multiplying A by 0, 1, -1, 2, -2, -3, -4, 3, 4. Multiply by zero implies that the product is "0 ". Multiply by" 1 "means that the product remains the same as the multiplier. Multiply by "-1" means that the product is the complementary form of the number of two. Multiplying with "-2" is to move left one as this rest as per table.

RESULT:

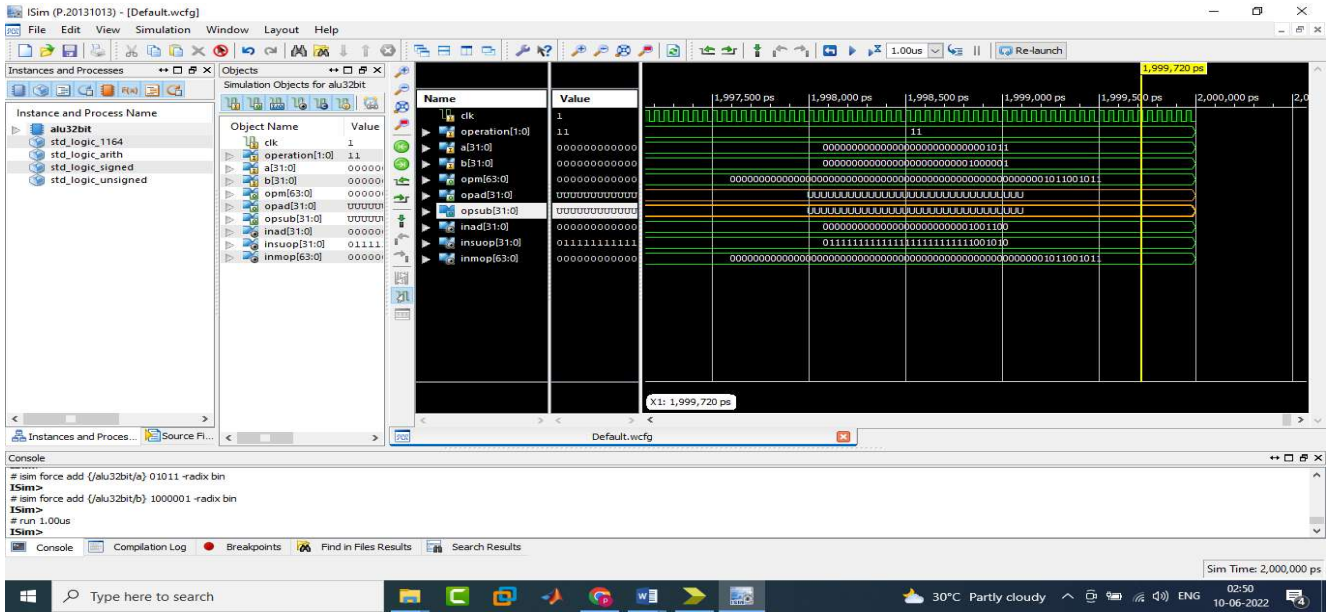


Fig: proposed multiplier in ALU

Radix-8 modified algorithm based multiplier is as shown in above figure with operand 1 as "1011" and operand 2 as "1000001" . Product is be stored in 'result' signal.

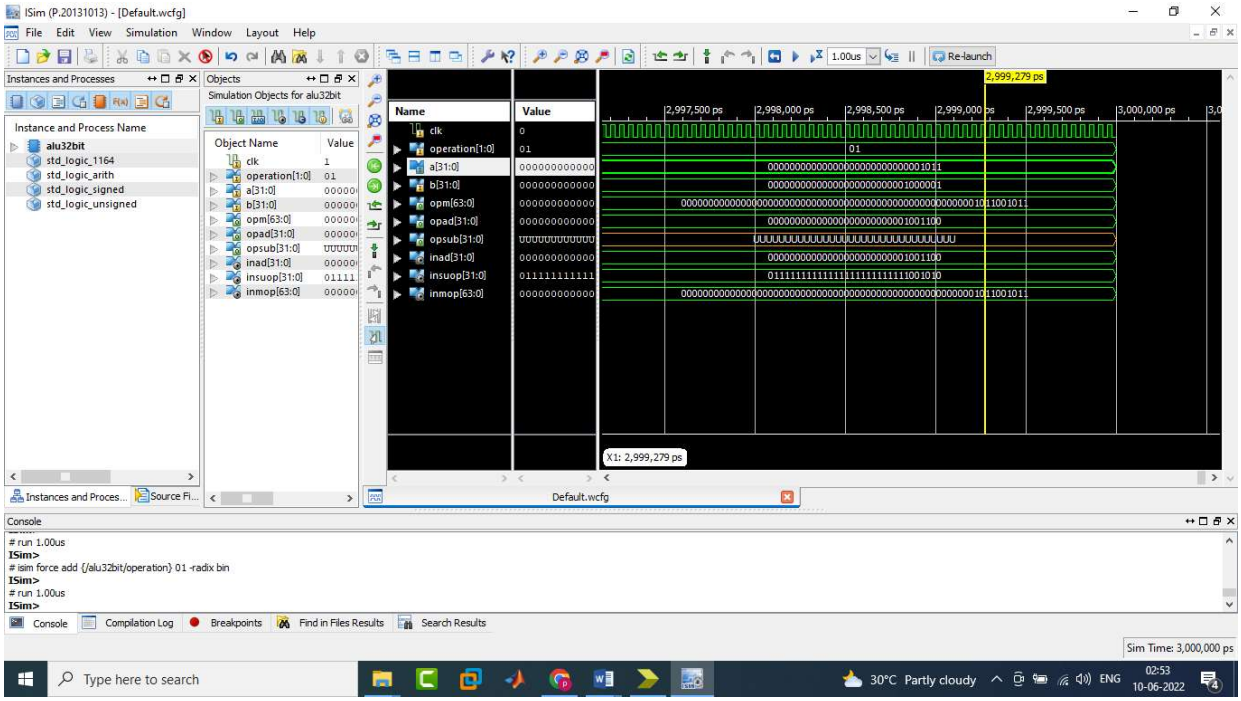


Fig: proposed adder in ALU

Square root modified carry select algorithm based addition is as shown in above figure with operand 1 as "1011" and operand 2 as "1000001" . Product is be stored in 'fsu signal.

CONCLUSION and FUTURE SCOPE In this paper, we have proposed efficient VHDL behavioral coding verification method. We have also proposed several algorithms using different design levels. Our proposals have been implemented in VHDL and verified using Xilinx ISE 14.7 analyzer. We have reduced the number of bus lines and all the designs have been implemented and tested. This ALU design using VHDL is successfully designed, implemented, and tested. In computing, an Arithmetic and logic unit is a digital circuit that performs arithmetic and logic operations. It will find its requirement in the field of Nanotechnology. Commercially it would be very useful in the smart mobile phones and calculating devices. As the numbers of input bits are increased, occupied area also increases. Now the plan is to implement ALU with more number of bits by maintaining the same area. Designers are aware that fabrication process introduces error because of the usage of large capacitance. Now, the time is to turn towards smaller capacitance. Logic Designs that are realized by smaller capacitance, consumes less power and optimizes area efficiently.

REFERENCES:

1. D. Gajski and R. Khun, Introduction: New VLSI Tools,IEEE Computer, Vol. 16, No. 12, pp. 11-14, Dec. 1983.
2. http://www.forteds.com/behavioralsynthesis/index.asp
   Douglas L. Perry, VHDL, third edition, McGraw-Hill, pp.60-63, 238, July 1999.
3. S.Yalamanchali, Introductory VHDL: From simulation to synthesis, Prentice Hall, United States, 2002.
4. http://www.xilinx.com
5. B.Stephen Brown, V.Zvonko, Fundamentals of digital logic with VHDL Design 2nd Edition , Mc Graw Hill International Edition, 2005.
6. Charles H.Roth, Jr., Digital System Design using VHDL, PWS Publishing Company, 2006.

7. Mark Zwolinski, Digital System Design with VHDL, Prentice Hall, 2000.Pedroni, Digital Logic Design using VHDL.
8. S.Kaliamurthy, R.Muralidharan, VHDL Design of FPGA Arithmetic Processor International Conference on Engineering and ICT, 2007.
9. Xilinx Technologies, Xilinx Data Sheet for XC3S100E. http:// direct. xilinx.com/bvdocs/ publications/ ds312.pdf.
10. http://www.forteds.com/behavioralsynthesis/index.asp
11. Prof. S. Kaliamurthy & Ms. U. Sowmmiya,VHDL design of arithmetic processor ,Global Journals Inc.(USA) , November 2011.
12. Geetanjali and Nishant Tripathi VHDL Implementation of 32-Bit Arithmetic Logic Unit (Alu)
13. Shikha Khurana, Kanika Kaur Implementation of ALU using FPGA
14. Mr. Abhishek Gupta , Mr. Utsav Malviya , Prof. Vinod Kapse A Novel Approach to Design High Speed Arithmetic Logic Unit Based On Ancient Vedic Multiplication Technique International Journal of Modern Engineering Research (IJMER) www.ijmer.com. Vol.2, Issue.4, July-Aug 2012 pp-2695-2698. ISSN: 2249-6645 www.ijmer.com

\