International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

DESIGN OF AREA OPTIMIZED ARITHMETIC AND LOGICAL UNIT FOR MICROCONTROLLER

DEVARAPALLI HARSHITHA *1, SAI DURGA*2, BHIMISETTY SUDHEER*3, AMARA MOUNIKA*4, ALUGUNDLA SIVA REDDY*5, CHUNDURI VAMSI*6

* 1,3,4,5 B. Tech Students, *2 Associate Professor
Dept. of Electronics and Communication Engineering,
RISE Krishna Sai Gandhi Group of Institutions

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.

1. 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.

The 3 main aspects considered in designing an ALU are:

- Power Dissipation
- Area
- Delay

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.

MECHANISMS OF POWER DISSIPATION

Mechanisms of power dissipation are usually divided into two classes: dynamic and static power dissipation. Dynamic power dissipation occurs when the circuit is operational, i.e. the circuit is performing some task on some data. Static power dissipation becomes an issue when the circuit is inactive or in a power-down mode.

There are 4 main sources of power dissipation in digital CMOS circuits. They are:

- Dynamic Power Dissipation
- Short-Circuit Power Dissipation

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

- Glitch Power Dissipation
- Static Power Dissipation

2. LITERATURE SURVEY:

Optimization of power and delay in VLSI circuits using transistor sizing and input ordering

A fast and efficient low power design method using cell libraries is developed. This optimization routine utilizes accurate and efficient statistical power estimation methods, transistor sizing and input ordering. Algorithms that select the best cell versions to use in a circuit are developed. Circuits are first mapped with minimum-sized versions to ensure low power and then gate versions are replaced by larger versions as necessary to satisfy the delay constraint with minimal power increase. Statistical power estimation methods are used to accurately estimate power in a circuit and the switching probabilities of each node in the circuit obtained from such an analysis are used to make decisions regarding the use of sized cell versions at a local level. In addition to accurate power estimation, input ordering is also used in the algorithm to further optimize the circuit.

Several unique circuit transversal algorithms are developed, each utilizing different aspects of circuit topology and node characteristics. Switching probabilities, output load, different raise and fall times and critical path analysis are all used in several different options to the main algorithm for future enhancement of the selection process. The heuristic methods developed have produced an optimization routine that takes little computation time, but produces circuits with desirable delay and power performance.

Minimum dynamic power CMOS design with variable input delay logic

Glitches are power wastage in CMOS circuits and need to be eliminated for low power applications. The technique is based on new variable input delay logic, which is a design style where logic gates have different delays along different I/O paths through a single gate. Three new ways are been proposed to design gates at transistor level. Variable input delay gates can be designed by input capacitance manipulation, single NMOS transistor addition or a CMOS pass transistor addition. A first approximation is derived from lookup table and the fine tuning is done by a steepest descent method. The technique achieved an average power savings of 58% and peak power savings of 68%. Thus, the technique is scalable for large circuits as well.

Energy-efficient hard fault detection, diagnosis and isolation in the ALU

Digital circuits are expected to increasingly suffer from more hard faults due to technology scaling. Especially, a single hard fault in the ALU might lead to a total failure in the embedded systems. In addition, energy efficiency is critical in these systems. To address these increasingly important problems in the ALU, we propose a novel energy-efficient fault-tolerant ALU design called Lizard. Lizard utilizes two 16-bit ALUs to perform 32-bit

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

computations with fault detection and diagnosis. By exploiting predictable operations, fault detection is performed in a single cycle. The 16-bit ALUs can be partitioned into two 8-bit ALUs. When a fault occurs in one of the four 8-bit ALUs, Lizard diagnoses and isolates a faulty 8-bit ALU for itself. After the faulty 8-bit ALU is isolated, Lizard continues its operation using the remaining three 8-bit ALUs, which can detect and isolate another fault. In this way, Lizard can survive faults on at most two sub-ALUs increasing its lifetime and fault tolerance. We conducted comparative evaluations with an unprotected ALU, triple modular redundancy ALU, and quadruple time redundancy ALU in terms of area, energy consumption, performance, and reliability. It is demonstrated that Lizard out performs other ALU designs in most cases, especially in energy efficiency.

BINARY MULTIPLICATION

Binary multiplication is one of the four binary arithmetic. The other three fundamental operations are addition, subtraction and division. In the case of a binary operation, we deal with only two digits, i.e. 0 and 1. The operation performed while finding the binary product is similar to the conventional multiplication method. The four major steps in binary digit multiplication are:

- $0 \times 0 = 0$
- $0 \times 1 = 0$
- $1 \times 0 = 0$
- $1 \times 1 = 1$

Note: The binary product of the two binary numbers 1 and 1 is equal to 1 only. And no additional number is borrowed or carried forward in this operation.

The binary multiplication operation is actually a process of addition and shifting operation. This process has to be continued until all the multiplier is done, and finally, the addition operation is made.

Similar to the decimal system, the multiplication of the binary numbers is done by multiplying the multiplicand with the multiplier. It is noted that the multiplication by zero makes all the bits zero, and this step may be ignored in the intermediate steps. The multiplication by 1 makes all the multiplicand value unchanged.

Binary Multiplication Table

The multiplication table for binary numbers is given below.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Binary Number	Multiplication Value
0 x 0	0
1 x 0	0
0 x 1	0
1 x 1	1

Let us take two binary numbers A = 1001 and B = 101 we want to find out A × B

```
1 0 1 0 1 1 0 0
  1 0 1 0 1 0 0
-----------------
              0
```

This is the first step in this step the least significant bit or the right most bit of B is multiplied with all the digits of A from the right side and the result is written. Here the steps took place are $1 \times 1 = 1, 1 \times 0 = 0, 1 \times 0 = 0, 1 \times 1 = 1$

```
1 0 0 1
1 0 1
-------
1 0 0 1
0 0 0 0
```

Similarly in this step all the elements of A are respectively multiplied with the second most significant bit i.e. 0.

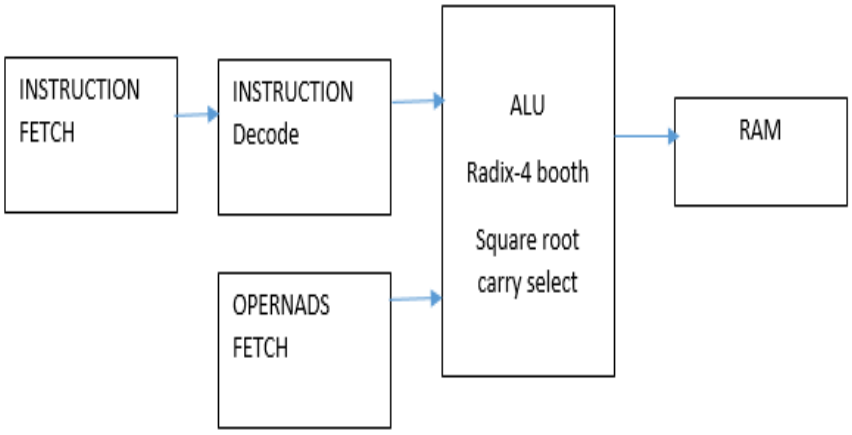
Suppose we wish to multiply two four-bit numbers, 1011 and 1010:

```
        1 0 1 1    this is 11(10)
    ×   1 0 1 0    this is 10(10)
-------------------
        0 0 0 0    1011 x 0
      1 0 1 1      1011 x 1, shifted one position to the left
    0 0 0 0        1011 x 0, shifted two positions to the left
+ 1 0 1 1          1011 x 1, shifted three positions to the left
-------------------
  1 1 0 1 1 1 0    this is 110(10)
```

Fig:1.Binary multiplication example

3. EXISTING METHOD:

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86



CARRY SELECT ADDER:

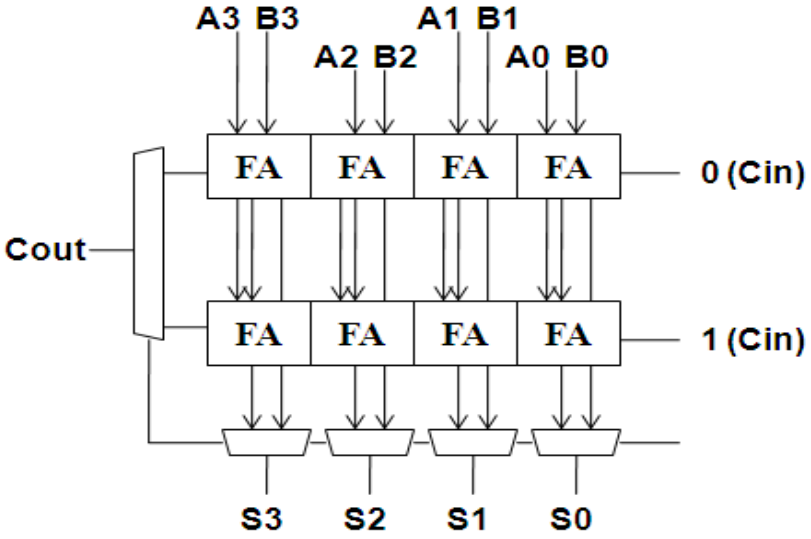


Fig:2. Basic building block of CSLA

The basic building block of the square root carry select adder of block size '4' is shown in the figure.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

BLOCK DIAGRAM OF CARRY SELECT ADDER:

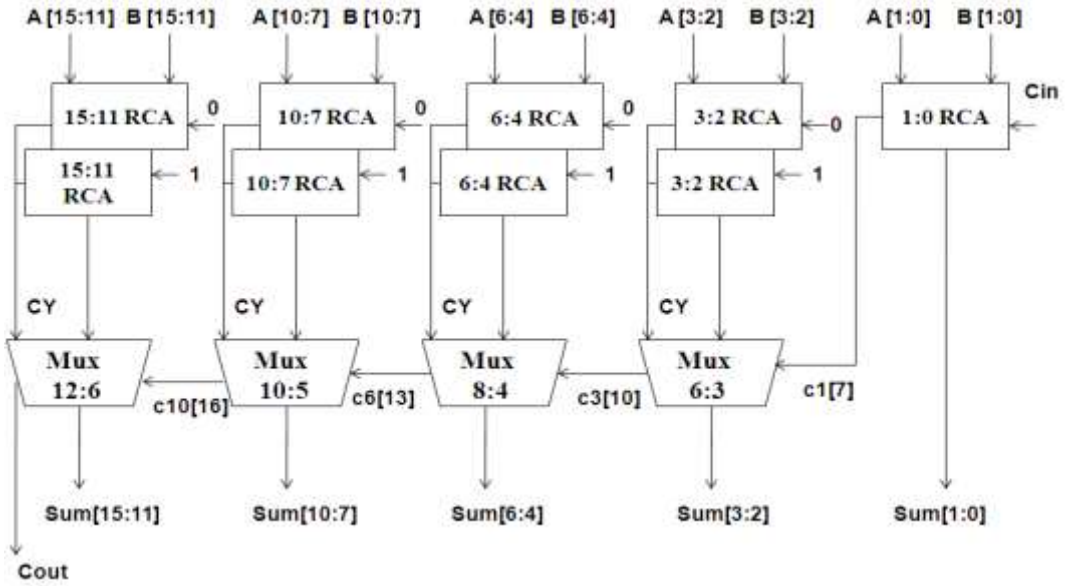


Fig:3.Existing system (Regular 16-bit Carry select adder)

The block diagram of the regular 16-bit square root CSLA is shown in the figure 3.2. This adder is a variable sized adder.

The carryselect adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

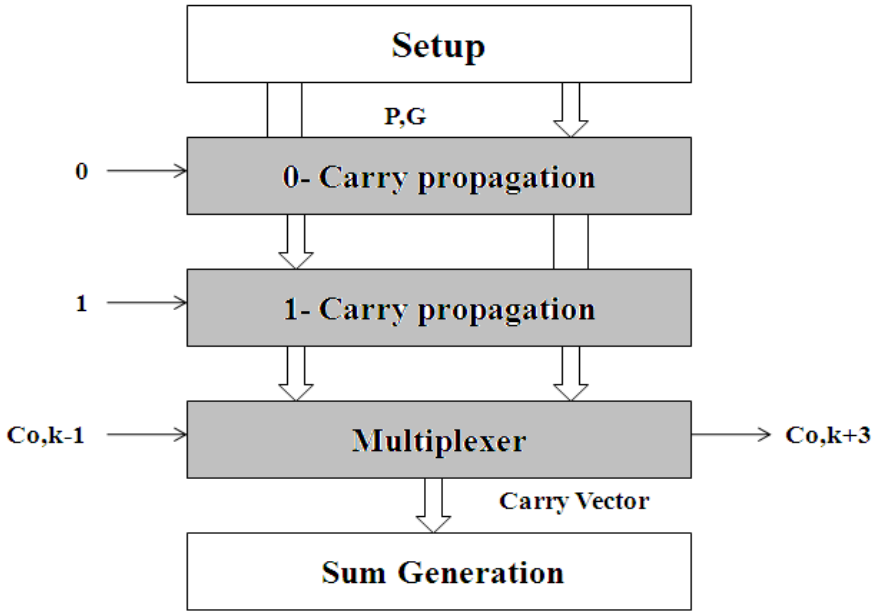


Fig:4. 4-bit carry select adder module topology

Seeing at the figure, the hardware overhead of the carry select adder is restricted to an additional carry path and a multiplexer and equals about 80% with respect to ripple carry adder. A full carry select adder is now constructed by chaining equal number of adder stages.

The critical path is shaded in gray color. From inspection of the circuit, we can derive the first order model of the worst case propagation delay of the module written as,

$T = tsetup + P \times tcarry + (2N)^{\wedge}½ \times tmux + tsum$

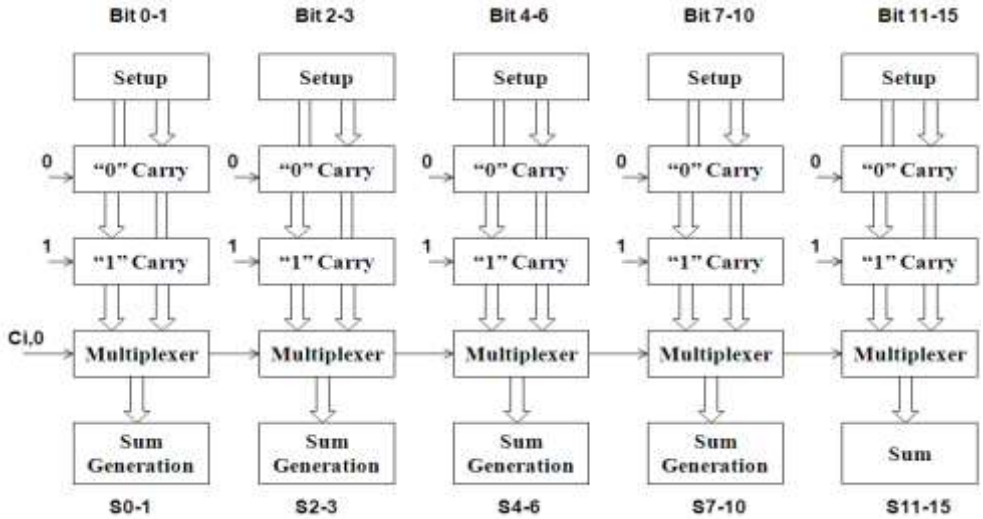


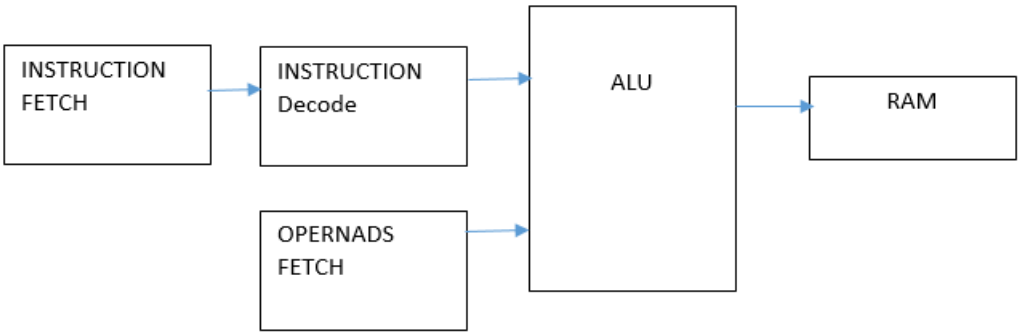
Fig:5. Delay propagation of 16-bit CSLA

The design procedure and the delay propagation of the 16-bit square root CSLA can be best explained from the figure. As from the figure, it can be seen that the model consists of 5 groups of different size. The addition process is carried out by considering the carry Cin=0

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

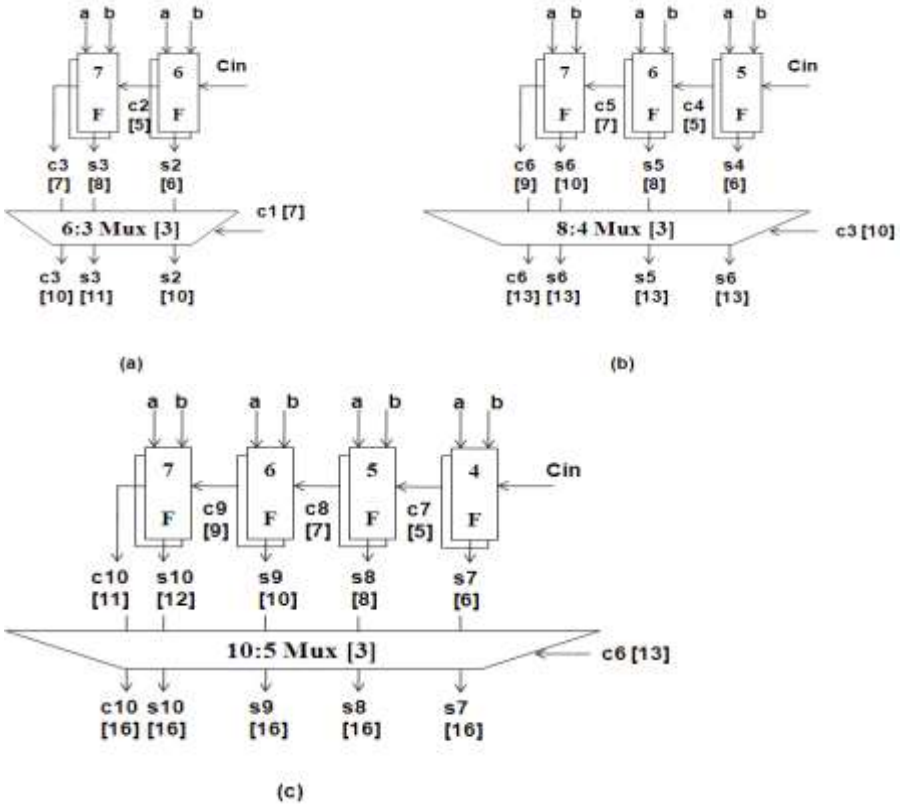
and Cin=1 and then generating the actual sum and carry using the actual carry from the previous stage is accomplished.

4. PROPOSED METHOD:



MODIFIED SQUARE ROOT CARRY SELECT ADDER WITH BEC

Architecture of 16-bit square root CSLA:



International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

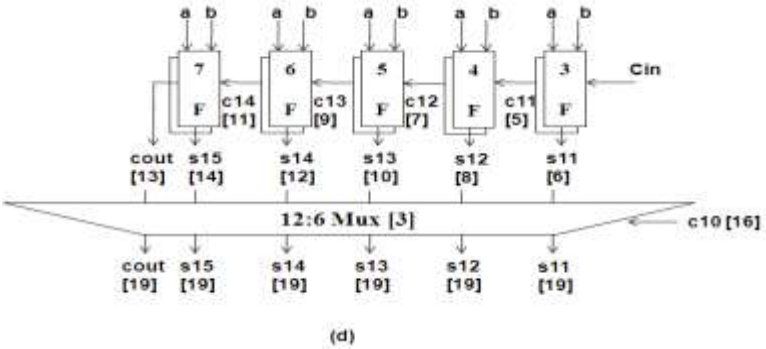


Fig:5. Delay and area evaluation of regular SQRT CSLA: (a) group2, (b) group3, (c) group4 and (d) group5. F is a Full Adder.

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.

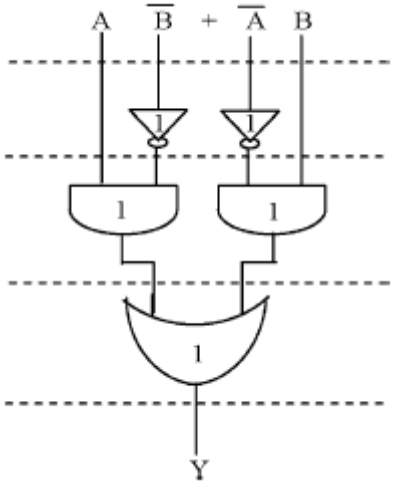


Fig:6. Delay and Area evaluation of an XOR gate

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

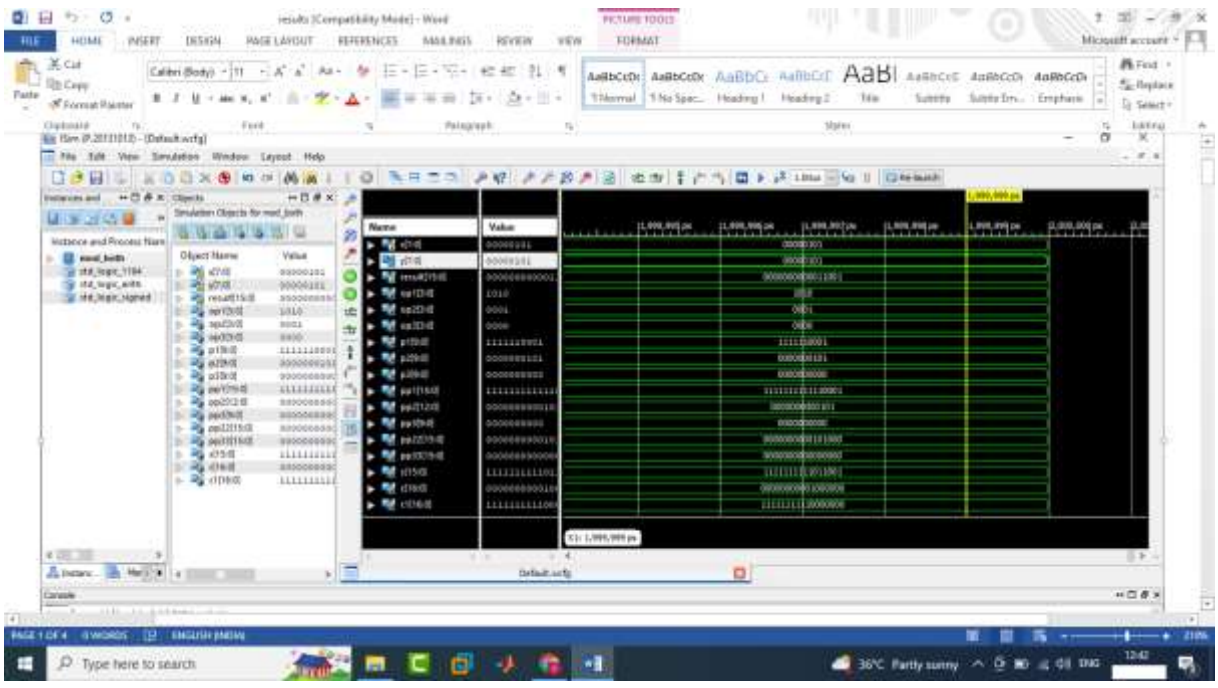
International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

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

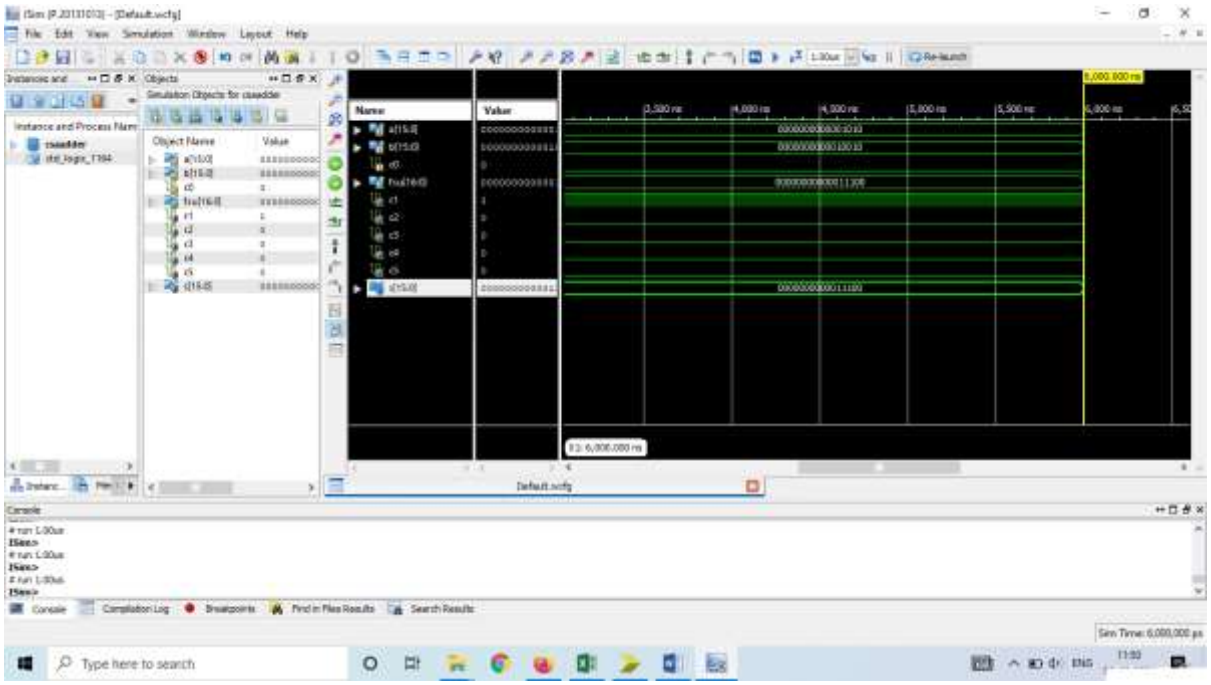
Table 4.1: Delay and area count of the basic blocks of CSLA

Delay and Area Evaluation of CSLA groups:

5. RESULT:



International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86



ADVANTAGES:

Low area occupancy
Low latency
Less power consumption with more throughput

APPLICATIONS:

FIR filters,
FFT processors,
Computers
Transmitters and Receivers modules

6. CONCLUSION:

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.

FUTURE SCOPE:

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