# Implementation of 4-BIT ALU with GCD

(USING EUCLID'S ALGORITHM DUMPED ON SPARTAN 3 FPGA FAMILY)

Saddam Khan[1], Yogiraj Kadikar[2], Shivam Dubey[3], Vrushal Kore[4]
U.G. Student, Department of Electronics and Telecommunication
St. John College of Engineering and Technology, Palghar, Mumbai, Maharashtra, India
[1]*khansaddam109@live.com* , [2]*yogiraj.kadikar@gmail.com* , [3]*shivamdubey.008@gmail.com* , [4]*vrushaldkore@gmail.com*

***Abstract*—** This work deals with the implementation of a ALU Processor with GCD using Xilinx ISE 9.1i and Spartan 3 FPGA kit. The GCD processor has been implemented using Euclid's algorithm. ALU performs arithmetic operations and logical operations. Code is dumped on FPGA kit and output is analyzed. The select line is used to decide the whether which operation to perform either GCD or ALU. The work also involves the simulation of the work on Xilinx ISE 9.1i. The implemented program was simulated and output waveforms were observed. We are using Spartan 3 FPGA board for observing the output. The program is dumped using JTAG( Joint Test Action Group) cable on the FPGA Board.

***Keywords-*** *Euclid's Algorithm; Greatest Common Divisor; Joint Test Action Group Field Programmable Gate Array; Xilinx ISE.*

*****

## I. INTRODUCTION

### A. ALU

An arithmetic and logic unit(ALU)[1] is a digital circuit that performs arithmetic and logical operations. The ALU is a fundamental building block of the central processing unit (CPU) of a computer. The inputs to the ALU are the data to be operated on (called operands) and the code from the control unit indicating which operation to perform. Its output is the result of the computation.

The processors found inside modern CPUs accommodate very powerful and very complex ALUs; a single component may contain a number of ALUs. Most of a processor's operations are performed by one or more ALUs. An ALU loads data from input registers, an external Control Unit then tells the ALU what operation to perform on that data, and then the ALU stores its result into an output register. The Control Unit is responsible for moving the processed data between these registers, ALU and memory.

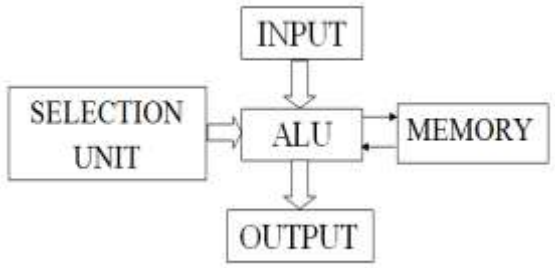


Fig 1. Block Diagram of ALU

The above Fig.1 is the Block diagram of Basic ALU. It is an ALU which operates on 4 bit value. It takes 4-bit input after that according to the opcode of selection lines it will perform the assigned operations. The output and other important data is stored in the memory

### B. GCD

GCD is an abbreviation for the "Greatest common divisor". GCD is also known as Greatest Common Factor(GCF) and Highest Common Factor (HCF). GCD calculation is highly utilized process as far as modern day computing is concerned. GCD finds its applications n data encryption and compression. Most of the encryption algorithms and techniques like Chinese remainder theorem, Euler's theorem all require a knowledge of GCD. GCD can be used in communication systems for the process of channel coding, error correction and cryptography. Various algorithms can be used for the calculation of GCD.

The primary algorithms used for computing GCD are Euclid's algorithm and extended binary algorithm also known as stein's algorithm.

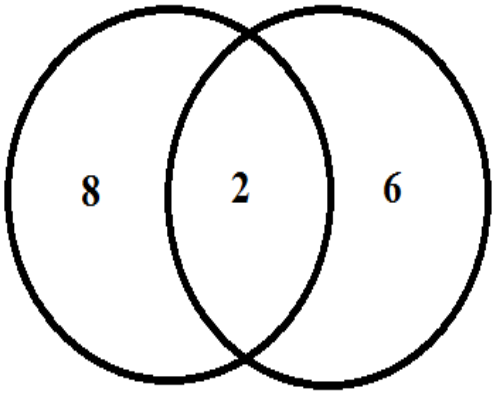


Fig 2. GCD

### C. XILINX ISE

**Xilinx ISE** (**I**ntegrated **S**ynthesis **E**nvironment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.
The **Xilinx ISE** is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors. The Xilinx ISE is primarily used for circuit synthesis

 ISSN: 2321-8169
1682 - 1687

and design, while the ModelSim logic simulator is used for system-level testing. Other components shipped with the Xilinx ISE include the Embedded Development Kit (EDK), a Software Development Kit (SDK) and Chip Scope Pro.

- *User Interface*

The primary user interface of the ISE is the Project Navigator, which includes the design hierarchy (Sources), a source code editor (Workplace), an output console (Transcript), and a processes tree (Processes). The Design hierarchy consists of design files (modules), whose dependencies are interpreted by the ISE and displayed as a tree structure. The Processes hierarchy describes the operations that the ISE will perform on the currently active module[] The hierarchy includes compilation functions, their dependency functions, and other utilities. The window also denotes issues or errors that arise with ecah function. The Transcript window provides status of currently running operations, and informs engineers on design issues. Such issues may be filtered to show Warnings, Errors, or both.

- *Simulation*

System-level testing may be performed with the ModelSim logic simulator, and such test programs must also be written in HDL languages. Test bench programs may include simulated input signal waveforms, or monitors which observe and verify the outputs of the device under test

## II. ALGORITHM USED

### A. *EUCLID'S ALGORITHM*

The Euclidean algorithm or Euclid's algorithm is a reliable method for computing GCD. It is named after the great Greek mathematician Euclid. The Euclid algorithm calculates the GCD of two non-negative integers and performs a repetitive comparison and subtraction to obtain the result. The algorithm is basically as follows:

gcd (x,0) = x.
gcd (x,y) = gcd (y, x mod y).

If above equations are both greater than zero, then

gcd (x, x) = x
gcd (x, y) = gcd (x-y, y); if y < x.
gcd (x, y) = gcd (x, y-x); if x < y.

For e.g. gcd (2,8) is 2.

Similarly, gcd (8,6) = gcd(8-6,6) = gcd (2,6) = gcd (6-2,2) = gcd(4,2) = gcd (4-2, 2) = gcd (2,2) = 2.

It is approximately 60% less efficient as compared to Stein's algorithm.

## III. FLOWCHART

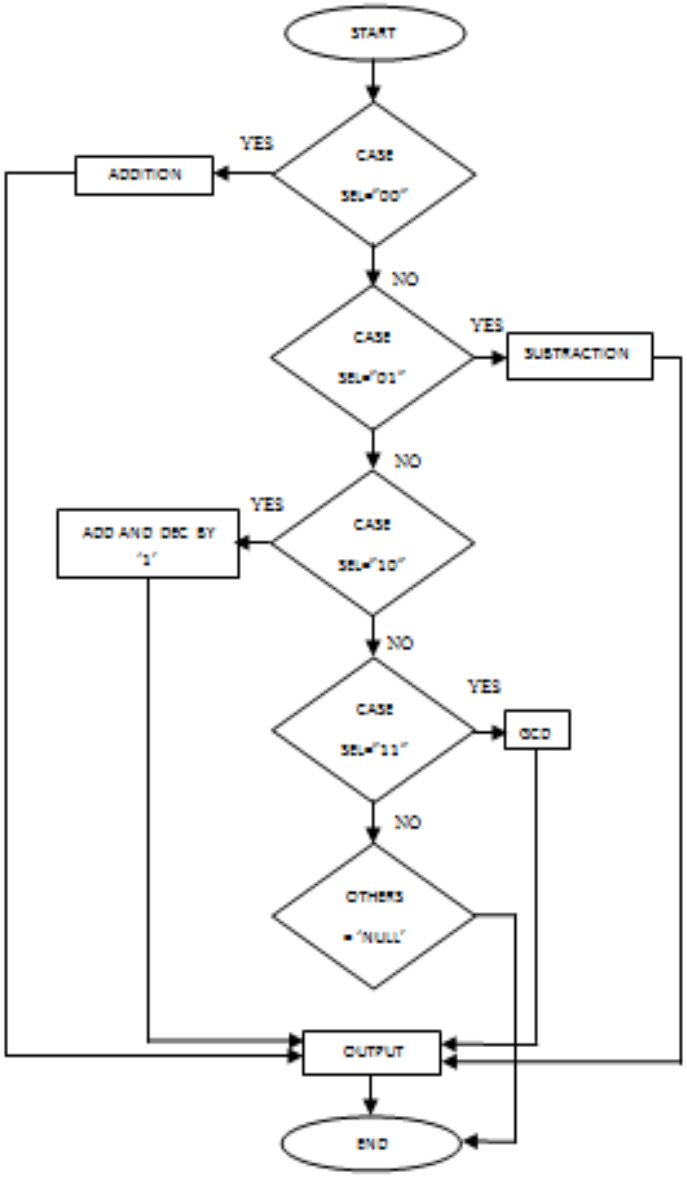


Fig 3. Flowchart of ALU Operations

The flowchart shown in Fig. 3 above is the pictorial representation of operations performed by Arithmetic and Logic Unit. Two inputs are given by the users, then according to the user requirements using the case statements, any of assigned operations can be performed.

For e.g.: if the user wants to calculate the GCD of those two numbers then user will select the '11' using select lines. After that the calculated GCD is shown on LEDs(Indicated in Fig. 7 below).

 ISSN: 2321-8169
1682 - 1687

## IV. DESIGN PARAMETERS

TABLE I. DESIGN SUMMARY

| Sr. No. | Device Utilization Summary | | |
|---|---|---|---|
| | **Logic Utilization** | **Used** | **Available** |
| 1 | Number of Slice Latches | 4 | 7168 |
| 2 | Number of 4 input LUTs | 112 | 7168 |
| 3 | Logic Distribution | | |
| 4 | Number of occupied Slices | 57 | 3584 |
| 5 | Number of Slices containing only related logic | 57 | 57 |
| 6 | Number of Slices containing unrelated logic | 0 | 57 |
| 7 | Total Number of 4 input LUTs | 112 | 7168 |
| 8 | Number of bonded IOBs | 14 | 141 |
| 9 | Total equivalent gate count for design | 764 | |
| 10 | Additional JTAG gate count for IOBs | 672 | |

TABLE II. POWER SUMMARY

| Sr. No. | Power summary | I(mA) | P(mW) |
|---|---|---|---|
| 1. | Total estimated power consumption | - | 56 |
| 2. | Vccint 1.20V | 15 | 19 |
| 3. | Vccaux 2.50V | 15 | 38 |

TABLE III. THERMAL SUMMARY

| Serial no. | Thermal summary | Temperature(in Celsius) |
|---|---|---|
| **1.** | **Estimated junction temperature:** | **27C** |
| **2.** | **Ambient temp:** | **25C** |
| **3.** | **Case temp:** | **27C** |
| **4.** | **Theta J-A:** | **35C/W** |

## V. SYNTHESIS AND SIMULATION

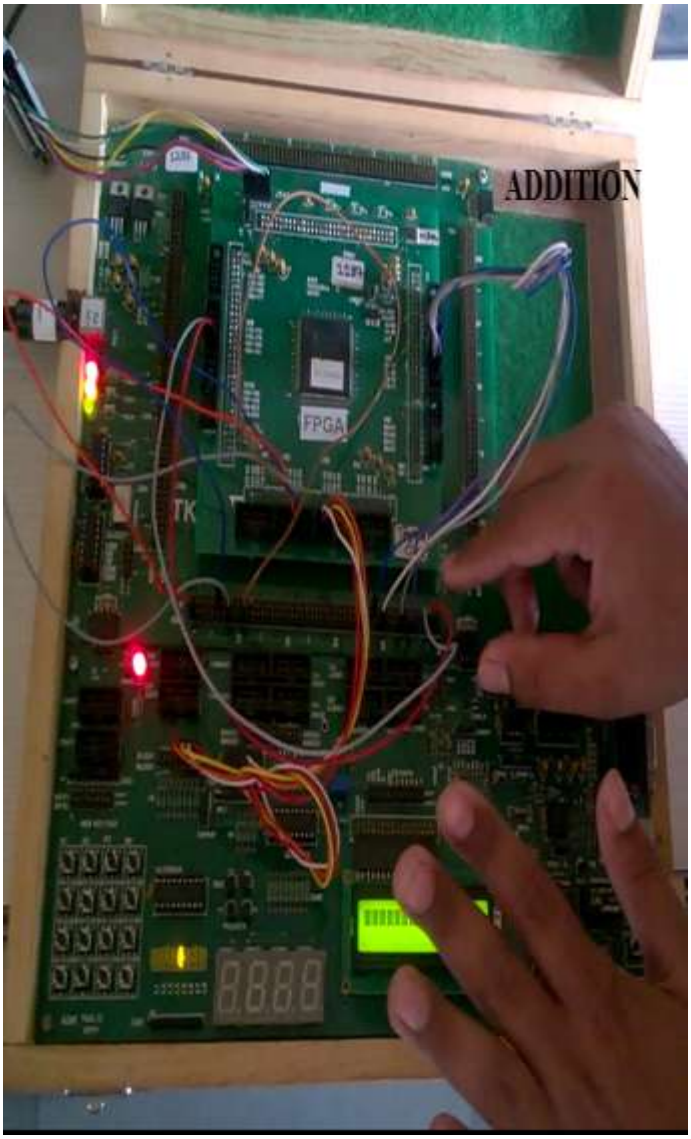


Fig. 4 Output on LED for op-code '00'

When select lines are given as '00' and the two input are given as binary '0100' and '0100'. So the result of addition operation is '8' i.e. '1000'(shown in Fig. 4 above).

Similarly, When select lines are given as '01' and the two input are given as binary '0100' and '0100'. So the result of subtraction operation is '0' i.e. '0000'(Shown in Fig. 5 below).

 ISSN: 2321-8169
1682 - 1687

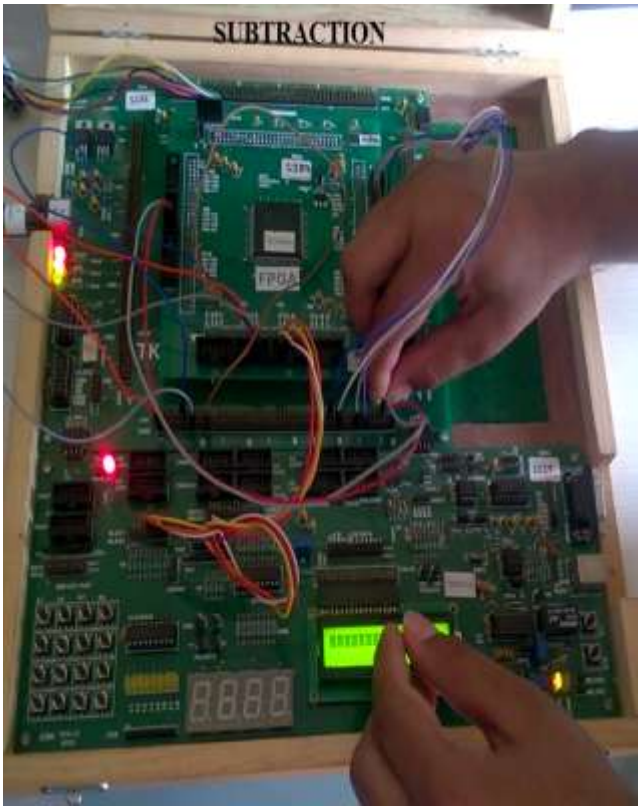


Fig. 5 Output on LED for op-code '01'

Also,When select lines are given as '10' and the two input are given as binary '0100' and '0100'. So the result of addition and the decrementing by 1 operation is '7' i.e. '0111'(indicated in Fig.6 below).

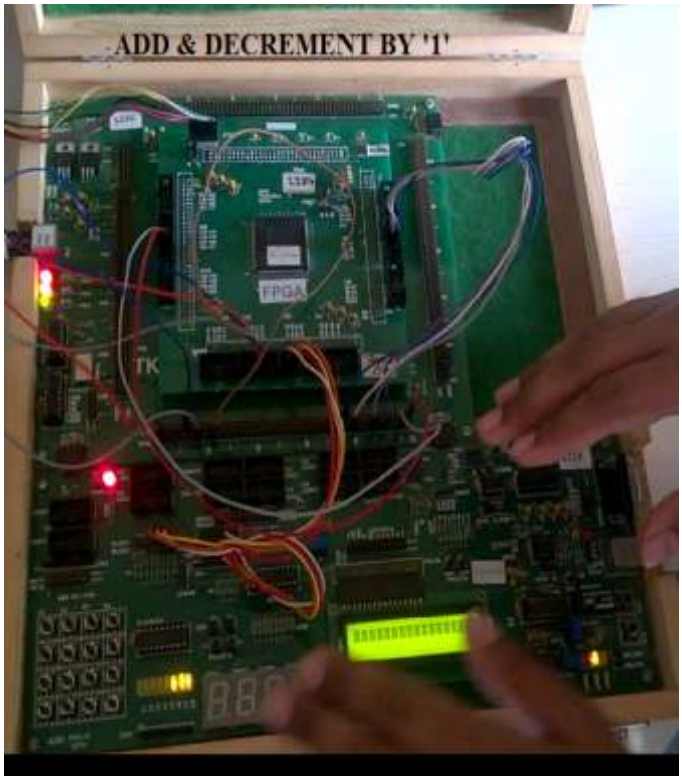


Fig. 6 Output on LED for op-code '10'

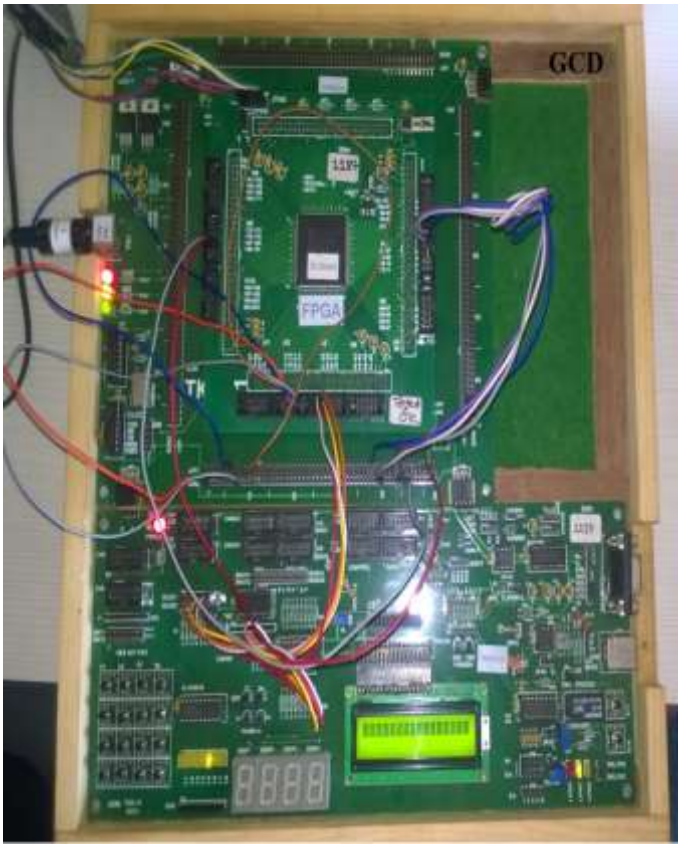


Fig. 7 Output on LED for op-code '11'

When select lines are given as '11' and the two input are given as binary '0100' and '0100'. So the result of GCD operation is '4' i.e. '0100'(shown in Fig. 7 above).

## CONCLUSION

The Arithmetic and Logic Unit(ALU) designed has also GCD operation implemented in that, which is used not only for arithmetic operations like addition and subtraction, but also for greatest common divisor(gcd) calculations. The gcd calculated in the ALU is by using Euclid's algorithm. The program is dumped on FPGA kit using Xilinx emulator. JTAG cable is used to connect computer and the FPGA kit. The Simulation Parameters , Power Summary and Thermal Summary of ALU design is shown in Table I, Table II and Table III above. From the Table I above , it is been observed that Slice Registers required is less, also the number of Bounded IOB's. Analyzed Power Consumption of ALU is calculated as 56 mW as shown in Table II above.

Fig. 8 below shows RTL view of implemented Arithmetic and Logic Unit indicating Input's and Output's.

Also, Fig. 9 below shows Simulation Output of ALU implementation.

 ISSN: 2321-8169
1682 - 1687

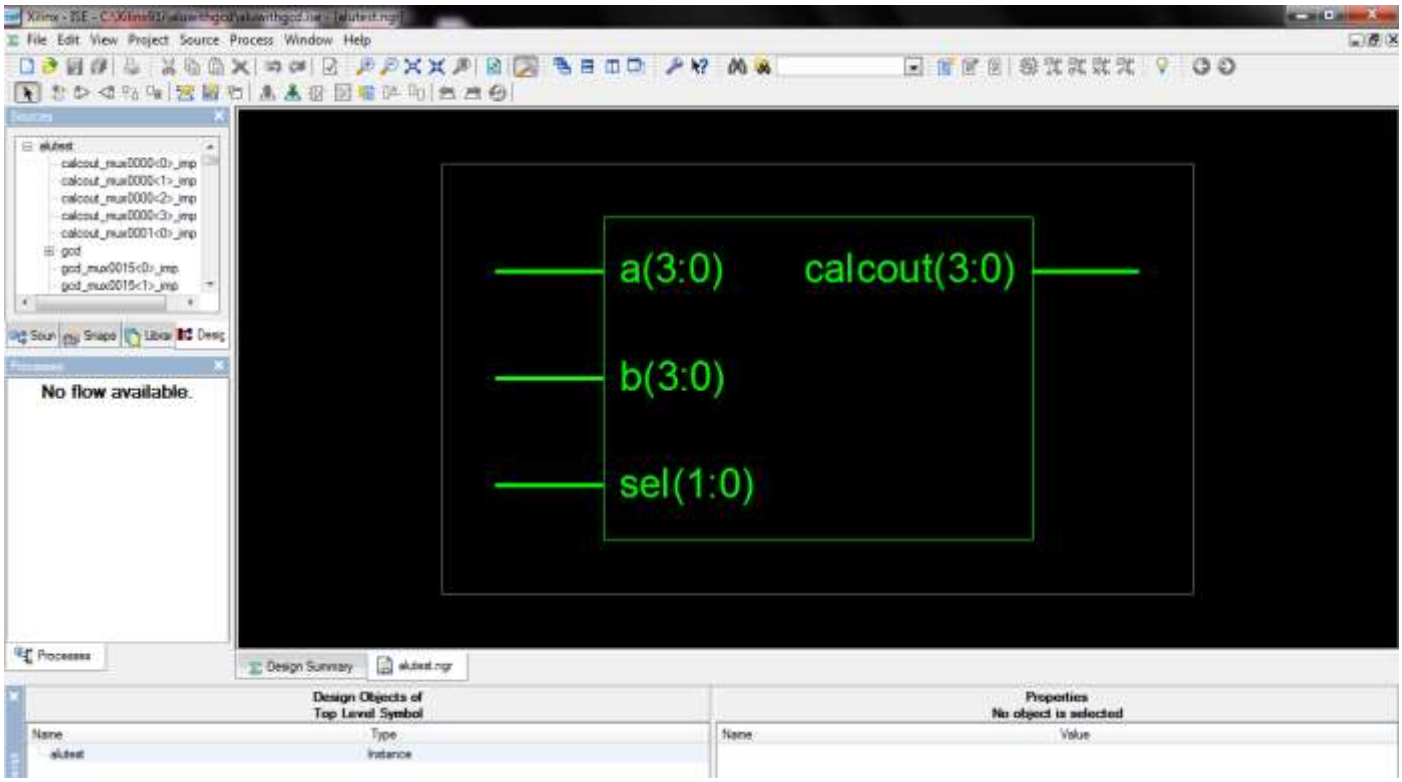


Fig 8. RTL View of ALU

a = first 4-bit input. b = second 4-bit input. sel = 00 / 01 /10 /11 for addition / for subtraction/ for addition and decrement by '1' / for GCD calculation using Euclid's algorithm. calcout = showing out simulation.

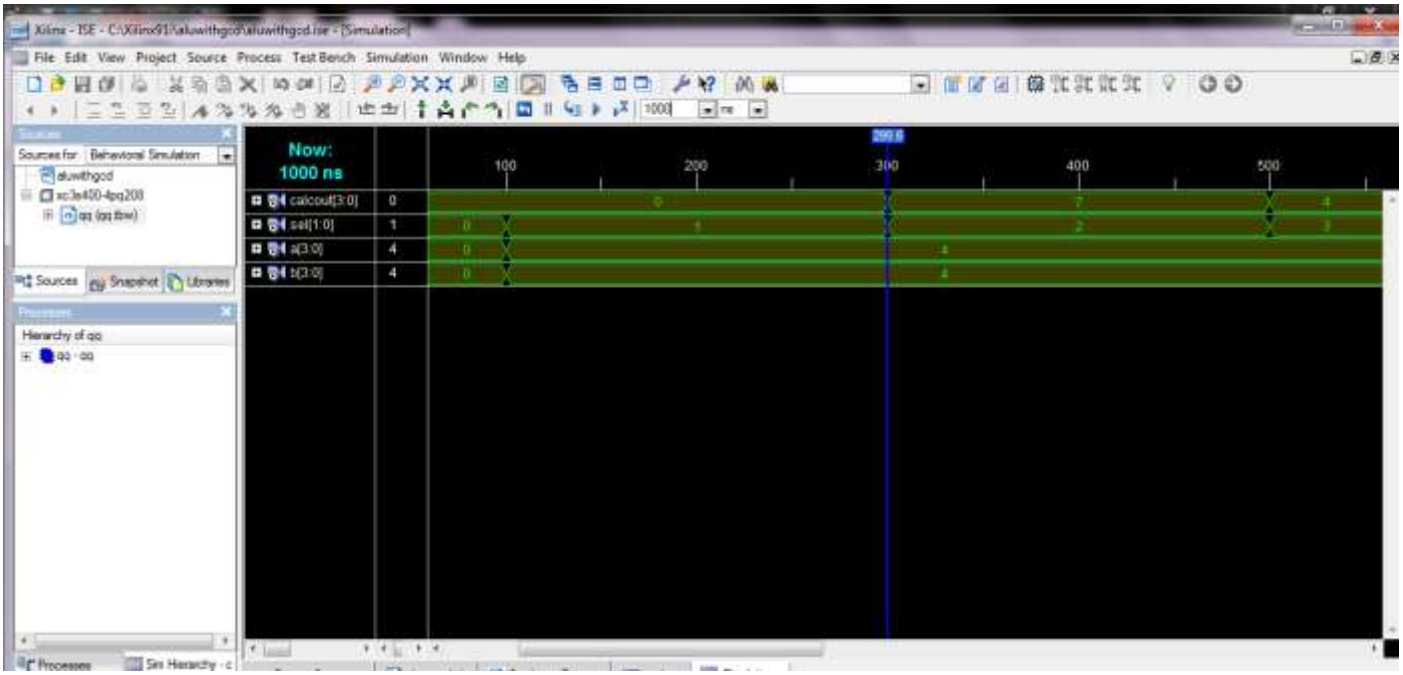


Fig 9. Simulation Result of ALU

.

## REFERENCES

[1] Jijil Kurian, Lijo Antony Alex, Venugopal G, "Design and FPGA implementation of a low power Arithmetic Logic Unit", IOSR Journal of VLSI and signal processing, Vol. 2, issue 3, May-June 2013, pp. 57-61
[2] Shikha Khurana, Kanika kaur, "Implementation of ALU using FPGA", Vol. 1, Issue 2, July-August 2012, pp. 146-149.
[3] Suchita Kamble, Prof. N. N. Mahala, "VHDL Implementation of 8-Bit ALU" IOSR Journal of Electronics and Communications Engineering, (IOSRJECE) ISSN: 2278-2834 Vol. 1, Issue 1, May-June 2012, pp. 07-11.
[4] Geetanjali, Nishant Tripathi, "VHDL Implementation of 32-bit arithmetic Logic unit (ALU)", International Journal of Computer Science and Communication Engineering IJCSCE Special issue on "emerging Trends in Engineering" ICETIE 2012, pp.41-44
[5] Geetanjali, Nishant Tripathi, "VHDL Implementation of 32-bit arithmetic Logic unit (ALU)", International Journal of Computer Science and Communication Engineering IJCSCE Special issue on "emerging Trends in Engineering" ICETIE 2012, pp.41-44.
[6] Y. Syamala, A.V.N. Tilak, "Reversible Arithmetic Logic Unit", 3rd International Conference on Electronics Computer Technology(ICECT), 5, pp. 207-211, 2011.
[7] [7] Rekha Devi, Jaget Singh, Mandeep Singh, "VHDL Implementation of GCD Processor with Built in Self-Test Feature", International Journal of Computer Applications (0975-8887), Vol. 25, July 2011, pp. 50–54.
[8] J. Bhaskar," VHDL Synthesis Primer", Pearson Education, 1st Edition, 2002.