iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** VIII **Month of publication:** August 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

A Review: Design of 16 bit Arithmetic and Logical unit using Vivado 14.7 and Implementation on Basys 3 FPGA Board

Prachi Sharma[1], Rama Laxmi[2], Arun Kumar Mishra[3]
[1]Student, [2,3]Assistant Professor, EC Department, Bhabha College of Engineering

Abstract: ***The paper primarily deals with review for the construction of arithmetic Logic Unit (ALU) using Hardware Description Language (HDL) using Xilinx Vivado 14.7 and implement them on Field Programmable Gate Arrays (FPGAs) to analyze the design parameters. ALU of digital computers is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve an efficient utilization of the available hardware. Speed, power and utilization of ALU are the measures of the efficiency of an algorithm. In this paper, we have simulated and synthesized the various parameters of ALUs by using VHDL on Xilinx Vivado 14.7 and Basys 3 Artix 7 FPGA board.***
Keywords: FPGA, ALU, XILINX Vivado 14.7, Basys 3 Artix 7 FPGA board.

I. INTRODUCTION

The Design and implementation of FPGA based Arithmetic Logic Unit is of core significance in digital technologies as it is an integral part of central processing unit. ALU is capable of calculating the results of a wide variety of basic arithmetical and logical computations[1]. The ALU takes, as input, the data to be operated on (called operands) and a code, from the control unit, indicating which operation to perform[2]. The output is the result of the computation. Designed ALU will perform the following operations:

A. Arithmetic operations
B. Bitwise logic operations

All the modules described in the design are coded using VHDL which is a very useful tool with its degree of concurrency to cope with the parallelism of digital hardware. The top level module connects all the stages into a higher level at Register Transfer Logic (RTL). RTL describes the requirements of data and control units in terms of digital logic to execute the desired operations. Each instruction from the architecture's instruction set is defined in detail in the RTL[3]. Once identifying the individual approaches for input, output and other modules, the VHDL descriptions are run through a VHDL simulator and then is downloaded the design on FPGA board for verification[4].

As FPGA has an application that it can incorporates much logic on a single FPGA. So as floating point ALU has many operations to be performed in the computer we are using an FPGA IC to implement it. The operations performed by the FPU are addition, subtraction, multiplication, division and logical operations as AND, OR, NOT etc[5]. FPU mainly work on Real as well as integers value.FPGA is an integrated circuit designed to be configured by the customers or designer after manufacturing- hence “Field Programmable”[6]. The FPGA configuration is generally specified using a hardware description language, similar to that used for an application specific integrated circuit (ASIC).

FPGA contain programmable logic components called “Logic Blocks”, and a hierarchy of reconfigurable interconnects that allows the block to be wired together. Logic blocks can be configured to perform complex combinational function or merely simple logic gates like AND and OR[7]. In most FPGA’s, the logic blocks also include memory elements which may be simple flip flops or more complete blocks of memory.

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

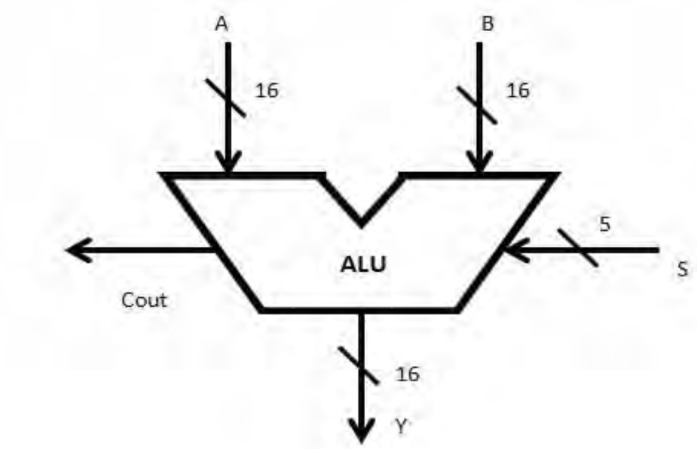


Fig.1: Symbol of 16-bit ALU.

II. DESIGN OF TOP LEVEL (RTL) VHDL MODULE OF 8 -BIT ARITHMETIC LOGICAL UNIT (ALU)

High level design methodology allows managing the design complexity in a better way and reduces the design cycle. [10]. A high-level model makes the description and evaluation of the complex systems easier. RTL description specifies all the registers in a design, and the combinational logic between them. The registers are described either explicitly through component instantiation or implicitly through inference [3]. The combinational logic is described by logical equations, sequential control statements subprograms, or through concurrent statements [3]. Designing at a higher level of abstraction delivers the following benefits [10].

Manages complexity: Fewer lines of code improves productivity and reduces error.
Increases design reuse: Implementation of independent designs as cell library & reuse in various models.
Improves verification: Helps to run process faster.

A. ALU Block Diagram

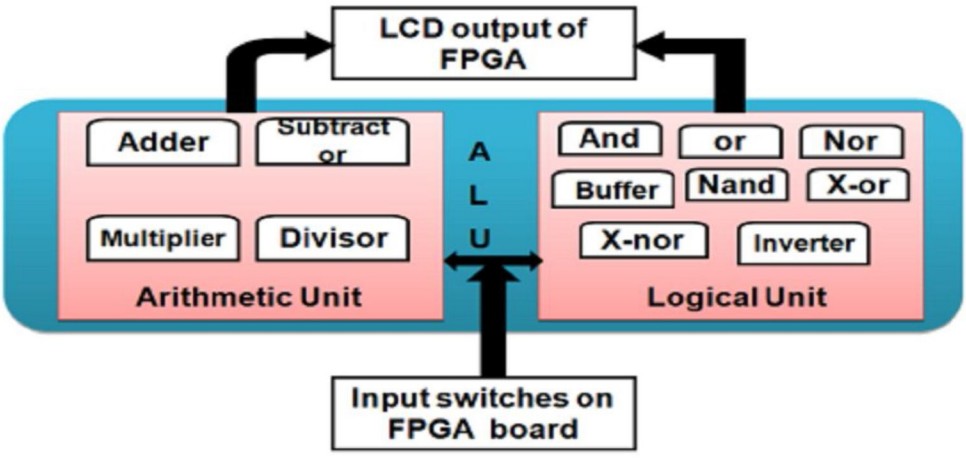


Fig.2.1: Block Diagram of ALU [6].

B. Operation Of ALU

There are two kinds of operation which an ALU can perform first part deals with arithmetic computations and is referred to as Arithmetic Unit. It is capable of addition, subtraction, multiplication, division, increment and decrement. The second part deals with the Gated results in the shape of AND, OR, XOR, inverter, rotate, left shift and right shift, which is referred to as Logic Unit. The functions are controlled and executed by selecting operation or control bits.

Table 2.1: ALU Operations

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

S3	S2	S1	S0	Operation performed
0	0	0	0	Half adder
0	0	0	1	Half subtractor
0	0	1	0	Full adder
0	0	1	1	Full subtractor
0	1	0	0	AND
0	1	0	1	OR
0	1	1	0	NOR
0	1	1	1	NOT
1	0	0	0	No shift
1	0	0	1	Logical or arithmetic left shift
1	0	1	0	Logical right shift
1	0	1	1	Arithmetic right shift
1	1	x	x	Multiplier

1) *Adder/Subtractor Unit:* In our ALU we have used the concept of adder-subtractor where the same circuit performs the functions of both adder and subtractor as shown in fig 2.2. The adder functions based on the concept of look ahead carry adder. The subtractor just uses an xor gate as an extra circuitry

The block diagram for an adder-subtractor circuit thus can be as below. To perform this we used carry look ahead adder, The carry look ahead adder reduces the consumption of power without compromising the speed of the adder [2]. This is achieved by generating carry simultaneously from all the bits. An n-bit carry look-ahead adder is formed from n stages. Carry look-ahead can be extended to larger adders. For example, four 1bit adders can be connected to form a 4bit adder and such four 4-bit adders can be connected to form the 16-bit adder.

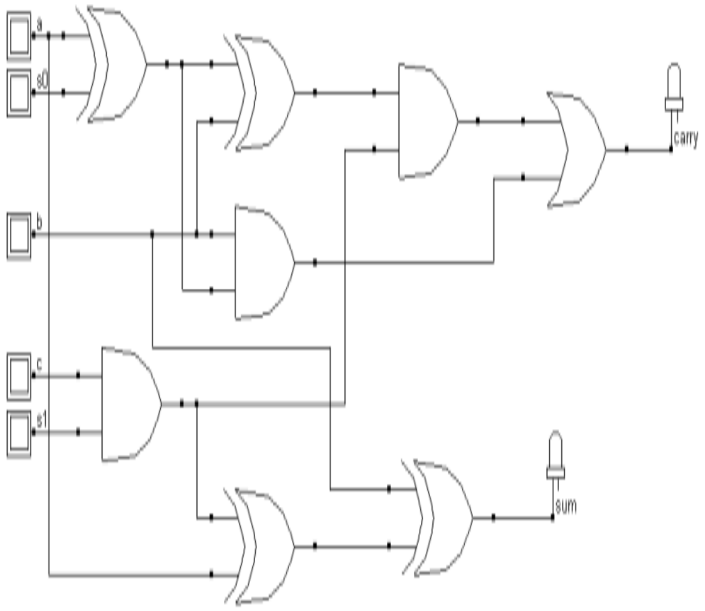


Fig 2.2: Adder/Subtractor Unit

The use of a single circuit for both adder and subtractor reduces power consumption and also area. The operation of the adder-subtractor is based on the S1 and S0 control bits.

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

S1	S0	Operation
0	0	Half Adder
0	1	Half Subtractor
1	0	Full Adder
1	1	Full Subtractor

Table 2.2: Operations of adder/subtractor circuit

2) *Logic Unit:* Fig 2.3 shows the logic unit in ALU, which performs 4 different logical operations AND, OR, XOR and NOT operations. Bitwise operation is performed on the two inputs. The operation to be performed is decided by two selections s1 and so as shown in Table 2.3.

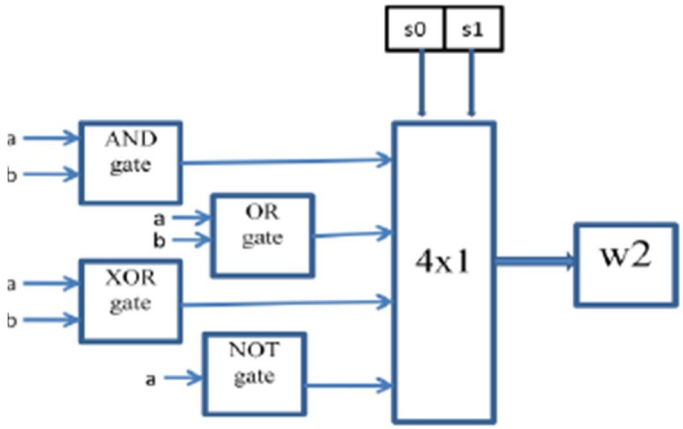


Fig 2.3: Logic Unit

Table 2.3: Logical Operations

S1	S0	Operation
0	0	AND
0	1	OR
1	0	XOR
1	1	NOT

3) *Shifter Unit:* Logical shift is an efficient way to perform division and multiplication of integers by powers of two. Shifting left by *k* bits on a binary number is equivalent to multiplying it by 2^k .Similarly shifting right by *k* bits on an binary number is equivalent to dividing it by 2^k. For example, consider the binary number 0001 0111.

a) Arithmetic shifts: Arithmetic shifts can be useful as efficient ways of performing multiplication or division of signed integers by powers of two. Shifting left by n bits on a signed or unsigned binary number has the effect of multiplying it by 2n. Shifting right

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

by n bits on a two's complement signed binary number has the effect of dividing it by 2n, but it always rounds down (towards negative infinity). Arithmetic left shift [6] move bits to the left, same order throw away the bit that pops off the MSB introduce a 0 into the LSB. This is same as logical left shift. It is shown in figure 2.4.a. Arithmetic right shift move bits to the right, same order throw away the bit that pops off the LSB reproduce the original MSB into the new MSB as shown in figure 2.4.b.

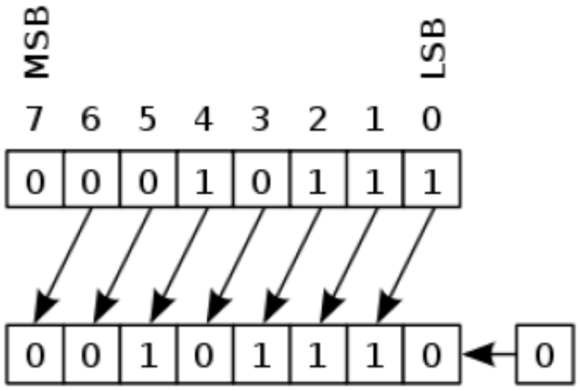


Fig 2.4.a: Arithmetic Left Shift

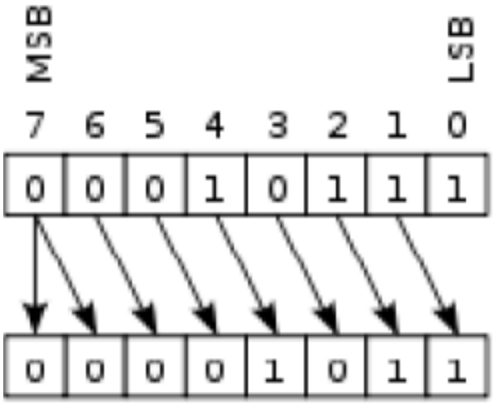


Fig 2.4.b: Arithmetic Right Shift

b) *Logical Shifts:* Logical shift is a bitwise operation that shifts all the bits of its operand. The two base variants are the logical left shift and the logical right shift. This is further modulated by the number of bit positions a given value shall be shifted, like "shift left by 1" or a "shift right by n". Unlike an arithmetic shift, a logical shift does not preserve a number's sign bit or distinguish a number's exponent from its mantissa; every bit in the operand is simply moved a given number of bit positions, and the vacant bit-positions are filled in, usually with zeros [7]. Logical **left** shift move bits to the left, same order throw away the bit that pops off the MSB introduce a 0 into the LSB as shown in figure 2.5.a. Logical **right** shift move bits to the right, same order throw away the bit that pops off the LSB introduce a 0 into the MSB as shown in figure 2.5.b.

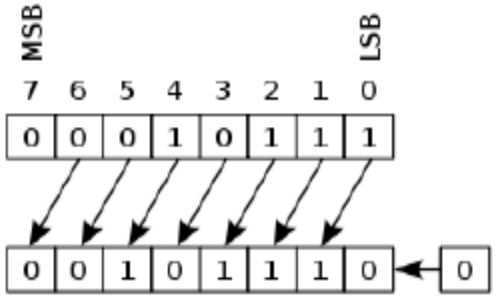


Fig 2.5.a: Logical Left Shift

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

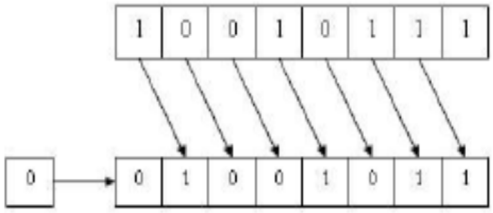


Fig 2.5.b: Logical Right Shift

Fig 2.6 shows the diagram of Shifter unit which performs arithmetic/Logical, Right and Left shifts by using Multiplexer and table 2.4 shows shifter operations controlled by 2 selection lines s1 and s0.

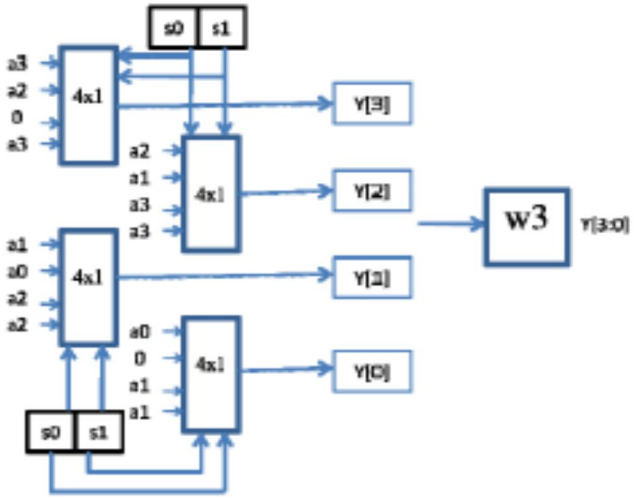


Fig 2.6: Shifter Unit

Table 2.4: Shifter unit Operations

S1	S0	Operation
0	0	No shift
0	1	Arithmetic/Logical Left Shift
1	0	Logical Right Shift
1	1	Arithmetic Right Shift

c) Multiplier Unit: The multiplication algorithm for an N bit multiplicand by N bit multiplier is shown below, AND gates are used to generate the Partial Products, PP, If the multiplicand is N-bits and the multiplier is M-bits then there is N* M partial product. The way that the partial products are generated or summed up is the difference between the different architectures of various multipliers. Multiplication of binary numbers can be decomposed into additions. Consider the multiplication of two 8-bit numbers A and B to generate the 16 bit product P. 1. If the LSB of Multiplier is „1", then add the multiplicand into an accumulator. 2. Shift the multiplier one bit to the right and multiplicand one bit to the left. 3. Stop when all bits of the multiplier are zero. Speed of the Processor mainly depends on Multiplier performance. There are Several Techniques for design of Multipliers. We need to select the appropriate technique based on factors delay, throughput, area and design complexity

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

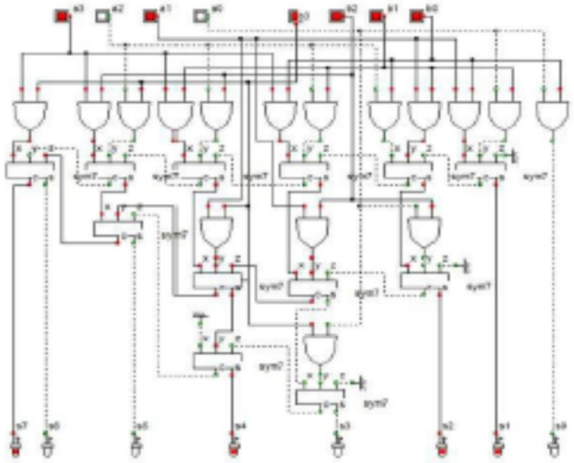


Fig 2.7: 4x4 Array Multiplier

III. IMPLEMENTATION OF 16-BIT ALU ON BASYS 3 FPGA BOARD

A. *Software Approach*

The VHDL software interface used in this design reduces the complexity and also provides a graphic presentation of the system. The key advantage of VHDL when used for systems design is that it allows the behavior of the required system to be described (modeled) and verified (simulated) before synthesis tools translate the design into real hardware (gates and wires). This software not only compiles the given VHDL code but also produces waveform results.

1) *Hardware Approach:* The VHDL code which implies the hardware part of ALU is downloaded on FPGA processor using JTAG cable interfacing PC and the hardware element. A final point is that when a VHDL model is translated into the "gates and wires" that are mapped onto a programmable logic device i.e FPGA, and then it is the actual hardware being configured, rather than the VHDL code being "executed" as if on some form of a processor chip.

IV. SIMULATION RESULT OF 16 BIT ALU DESIGN

A. *OR operation*

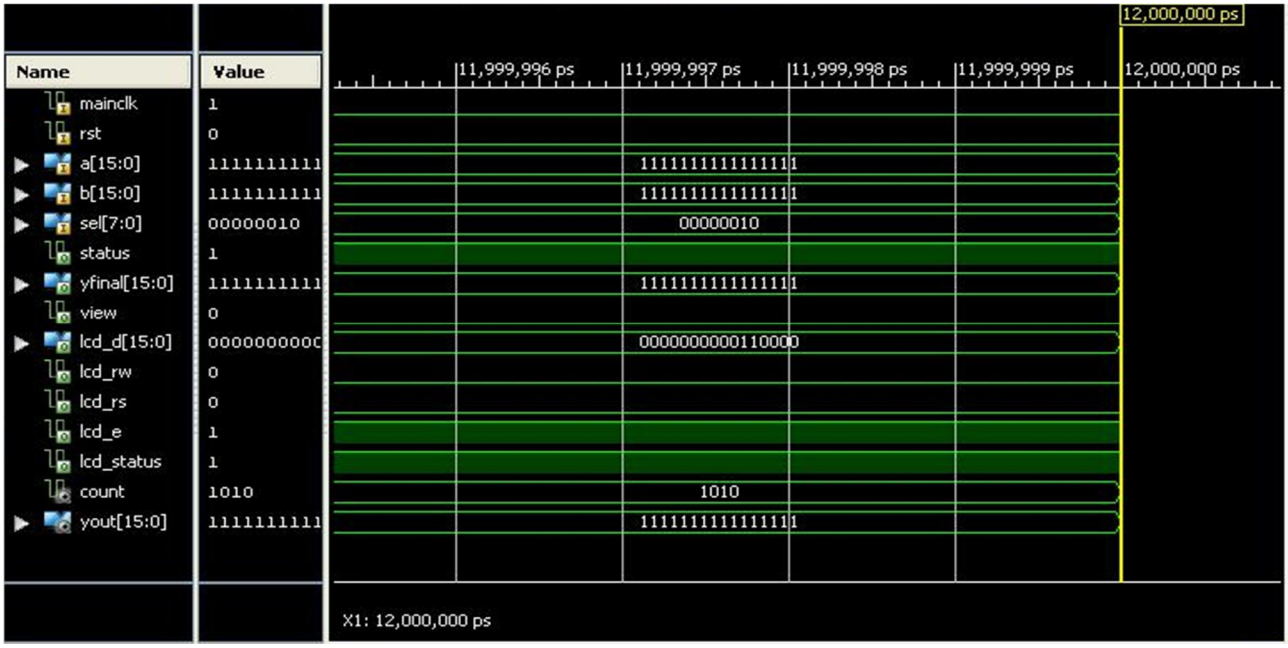


Fig 4.1: Waveform of OR operation.

©IJRASET: All Rights are Reserved

Volume 4 Issue VIII, August 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

1) XOR Operation

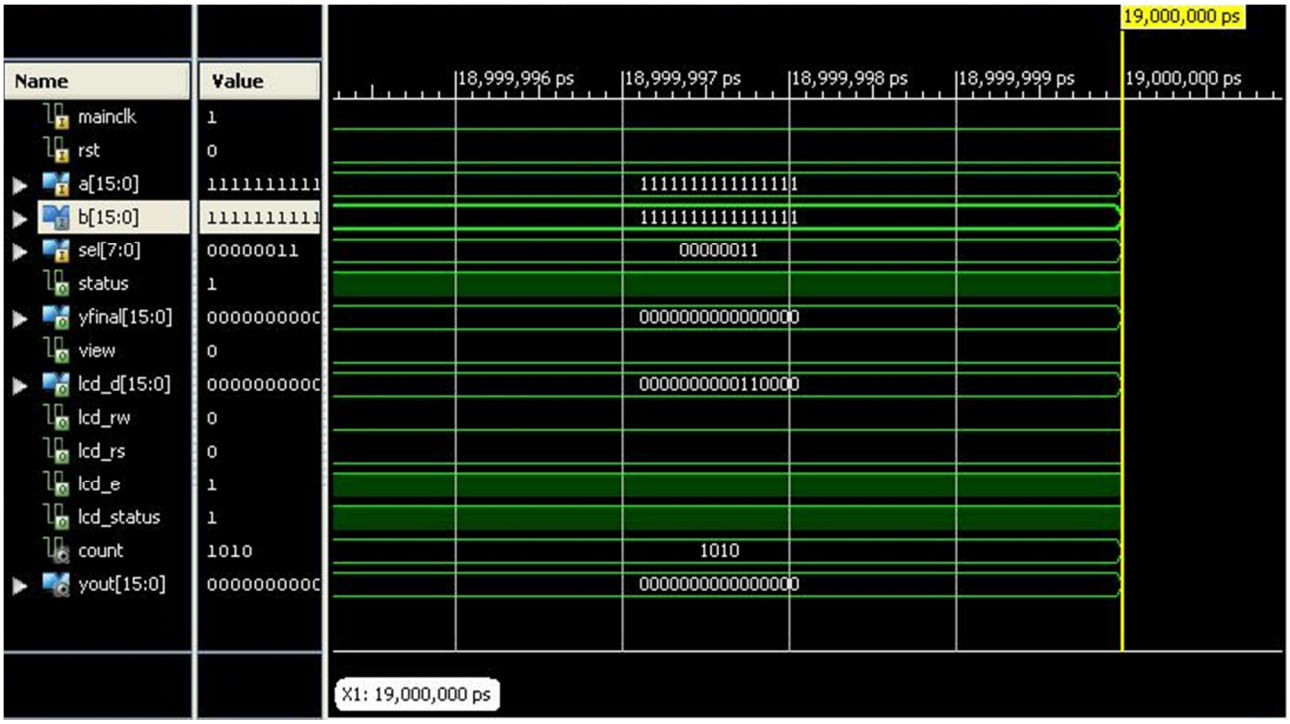


Fig 4.2: Waveform of XOR operation

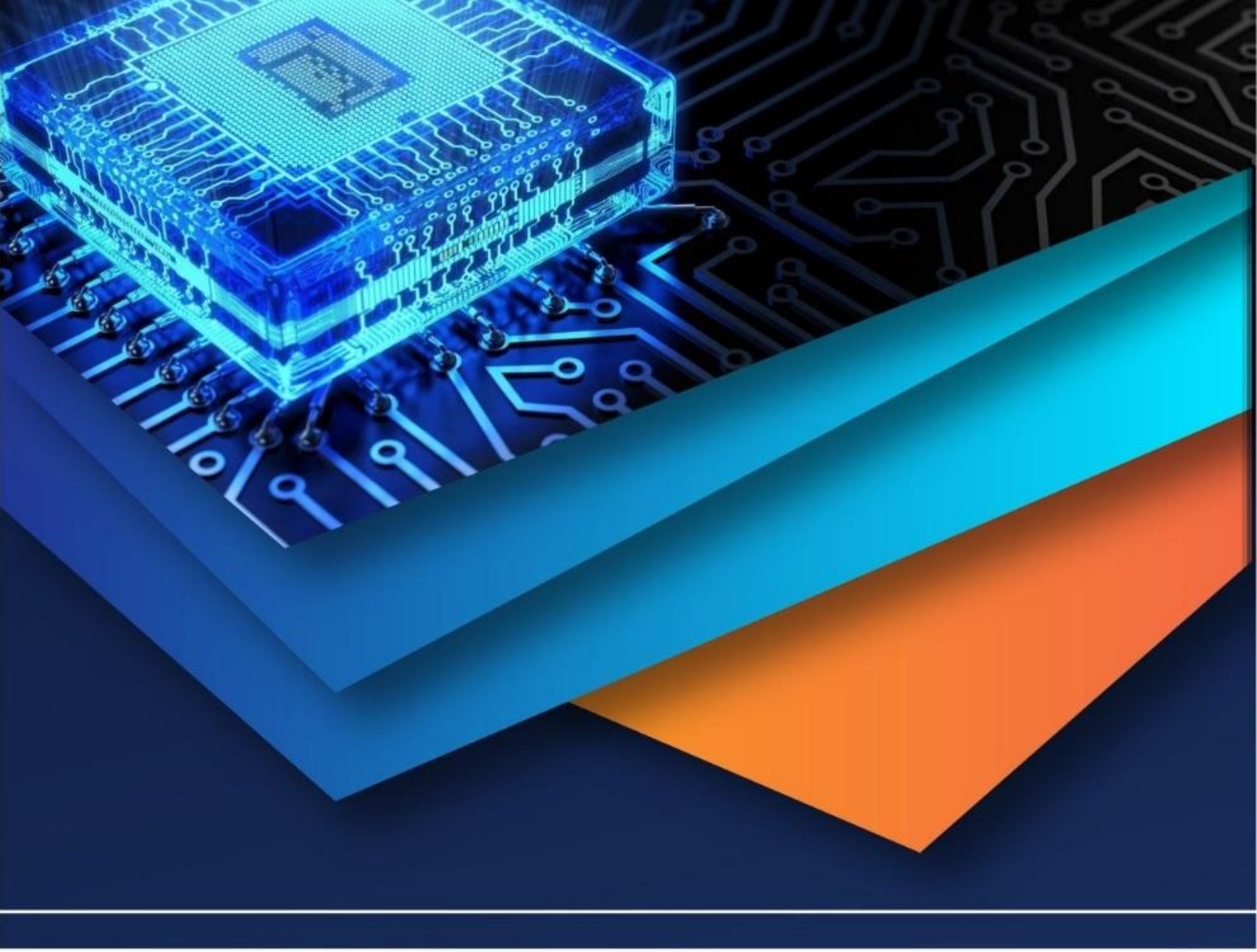
V. CONCLUSION

This study helped to understand the complete flow of RTL design, starting from designing a top level RTL module for 16-bit ALU using hardware description language, VHDL. Verification of the designed RTL code using simulation techniques, synthesis of RTL code to obtain gate level netlist using Xilinx Vivado ISE tool and Arithmetic Logic Unit was successfully designed and implemented using Very High Speed Hardware Descriptive Language and Xilinx Basys 3 Field Programmable Gate Array.

REFERENCES

[1] B.Stephen Brown, V.Zvonko, "Fundamentals of digital logic with VHDL Design"2ndEdition,Mc Graw Hill International Edition, 2005.
[2] Charles H.Roth, Jr., "Digital System Design using VHDL", PWS Publishing Company, 2006.
[3] Douglas L. Perry, VHDL, third edition, McGraw-Hill, pp. 60-63, 238, July 1999.
[4] Mark Zwolinski, "Digital System Design with VHDL", Prentice Hall, 2000.
[5] Pedroni, "Digital Logic Design using VHDL".
[6] S.Kaliamurthy, R.Muralidharan, "VHDL Design of FPGA Arithmetic Processor" International Conference on Engineering and ICT, 2007.
[7] Prof. S. Kaliamurthy & Ms. U. Sowmmiya, "VHDL design of arithmetic processor" ,Global Journals Inc.(USA) , November 2011.
[8] Landauer, R., "Irreversibility and heat generation in the computing process", IBM J.Research and Development, vol. 5 (3): pp. 183-191, 1961.
[9] Bennett, C.H., "Logical reversibility of computation", IBM J. Research and Development, vol. 17: pp. 525-532, 1973.
[10] B. Raghu Kanth1, B. Murali Krishna2, G. Phani Kumar3, J. Poornima4, K. Siva Rama Krishna " A Comparitive Study Of Reversible Logic Gates"International Journal of VLSI & Signal Processing Applications, Vol.2,Issue 1, Feb 2012.
[11] Himanshu Thapliyal ,Nagarajan Ranganathan "A New Reversible Design of BCD Adder " IEEE conference on Design and automation, 2011 pp.1-4.
[12] Zhijin Guan, Wenjuan Li, Weiping Ding, Yueqin Hang, Lihui Ni"An Arithmetic Logic Unit design based on reversible logic gates "IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim), 2011.
[13] Thapliyal H, Srinivas M.B, "Novel Reversible TSG Gate and Its Application for Designing Components of Primitive Reversible/Quantum ALU,"Fifth International Conference on Information, Communications and Signal Processing, 2006.
[14] H.Thapliyal,M.B Srinivas "Novel design and reversible logic synthesis of multiplexer based full adder and multipliers" 48th Midwest Symposium on Circuits and Systems, 2005.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)