International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

A Case Study: Design of 16 bit Arithmetic and Logical unit using Xillinx 14.7 and Implementation on FPGA Board

Manish Kumar
Jodhpur Institute of Engineering & Technology, Jodhpur
Suresh Kumar Jha
Jodhpur Institute of Engineering & Technology, Jodhpur
Ravindra Sharma
Jodhpur Institute of Engineering & Technology, Jodhpur

ABSTRACT: *The paper basically manages survey for the development of number juggling Logic Unit (ALU) utilizing Hardware Description Language (HDL) utilizing Xilinx Vivado 14.7 and actualize them on Field Programmable Gate Arrays (FPGAs) to investigate the outline parameters. ALU of advanced PCs is a part of rationale plan with the goal of creating suitable calculations keeping in mind the end goal to accomplish a proficient use of the accessible equipment. Speed, power and use of ALU are the measures of the proficiency of a calculation. In this paper, we have reproduced and joined the distinctive 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 Plan and execution of FPGA based Number juggling Rationale Unit is of center noteworthiness in computerized innovations as it is an indispensable piece of focal handling unit. ALU is fit for figuring the consequences of a wide assortment of essential arithmetical and legitimate computations [1]. The ALU takes, as information, the information to be worked on (called operands) and a code, from the control unit, demonstrating which operation to perform [2]. The yield is the consequence of the calculation. Outlined ALU will play out the accompanying operations:

A. Arithmetic operations

B. Bitwise rationale operations

Every one of the modules portrayed in the outline are coded utilizing VHDL which is an extremely valuable apparatus with its level of simultaneousness to adapt to the parallelism of advanced equipment. The best level module associates every one of the phases into a more elevated amount at Enlist Exchange Rationale (RTL). RTL portrays the necessities of information and control units as far as computerized rationale to execute the coveted operations. Every direction from the design's guideline set is characterized in detail in the RTL [3]. Once recognizing the individual methodologies for info, yield and different modules, the VHDL portrayals are go through a VHDL test system and after that is downloaded the outline on FPGA board for verification [4].

As FPGA has an application that it can consolidates much rationale on a solitary FPGA. So as drifting point ALU has numerous operations to be performed in the PC we are utilizing a FPGA IC to execute it. The operations performed by the FPU are expansion, subtraction, augmentation, division and intelligent operations as AND, OR, NOT etc [5]. FPU for the most part chip away at Genuine and also numbers value. FPGA is an incorporated

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

circuit intended to be arranged by the clients or architect subsequent to assembling thus "Field Programmable"[6]. The FPGA design is by and large determined utilizing an equipment portrayal dialect, like that utilized for an application particular incorporated circuit (ASIC).

FPGA contain programmable rationale segments called "Rationale Squares", and a chain of command of reconfigurable interconnects that enables the piece to be wired together. Rationale pieces can be arranged to perform complex combinational capacity or only straightforward rationale doors like and OR [7]. In many FPGA's, the rationale pieces additionally incorporate memory components which might be straightforward flip lemon or more total squares of memory.

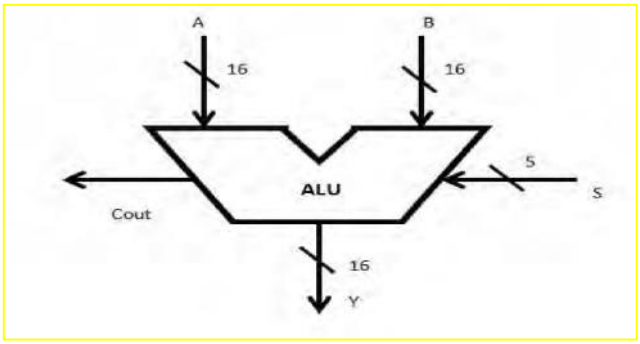

Fig.1: Symbol of 16-bit ALU.

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

ALU state plan technique permits dealing with the outline multifaceted nature betterly and diminishes the outline cycle. [10]. An ALU state demonstrate makes the portrayal and assessment of the mind boggling frameworks simpler. RTL portrayal determines every one of the registers in an outline, and the combinational rationale between them. The registers are depicted either unequivocally through segment instantiation or verifiably through induction [3]. The combinational rationale is depicted by consistent conditions, consecutive control explanations subprograms, or through simultaneous proclamations [3]. Planning at a larger amount of deliberation conveys the accompanying advantages [10].

A. ALU Block Diagram

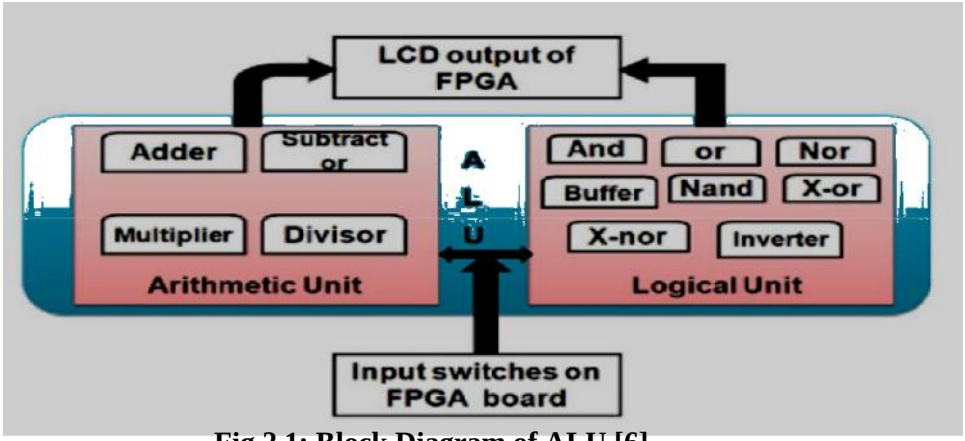

Fig.2.1: Block Diagram of ALU [6].

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

Oversees multifaceted nature: Less lines of code enhances profitability and diminishes blunder.

Expands configuration reuse: Usage of autonomous plans as cell library and reuse in different models.

Enhances check: Runs process speedier.

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 equipped for expansion, subtraction, augmentation, division, addition and decrement. The second part manages the Gated brings about the state of AND, OR, XOR, inverter, pivot, left move and right move, which is allowed to as Rationale Unit. The functions are controlled and executed by selecting operation or control bits.

Table 2.1: ALU Operations

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 utilized the idea of snake subtractor where a similar circuit plays out the elements of both viper and subtractor as appeared in fig 2.2. The snake capacities in view of the idea of look forward convey viper. The subtractor just uses a xor door as an additional hardware

The square chart for an adder subtractor circuit consequently can be as beneath. To play out this we utilized carry look ahead adder, the carry look ahead adder decreases the utilization of energy without trading off the speed of the adder [2]. This is accomplished by producing convey all the while from every one of the bits. An n-bit convey look-ahead adder is shaped from n stages. Carry look-ahead can be reached out to bigger adders. For instance, four 1bit adders can be associated with shape a 4bit viper and such four 4-bit adders can be associated with frame the 16-bit adder.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

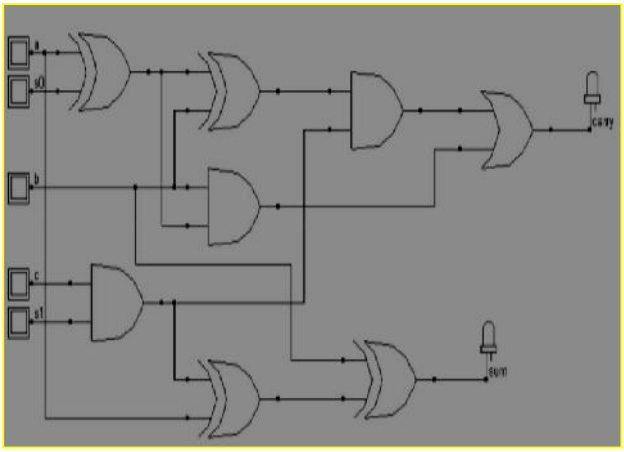

Fig 2.2: Adder/Subtractor Unit

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.

Table 2.2: Operations of adder/subtractor circuit

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

2) Logic Unit:

Fig 2.3 demonstrates the loic unit in ALU, which performs 4 different logical operations AND, OR, XOR and NOT operations. Bitwise operation is performed on the two inputs sources. The operation to be performed is chosen by two choices s1 and so thus as appeared in Table 2.3.

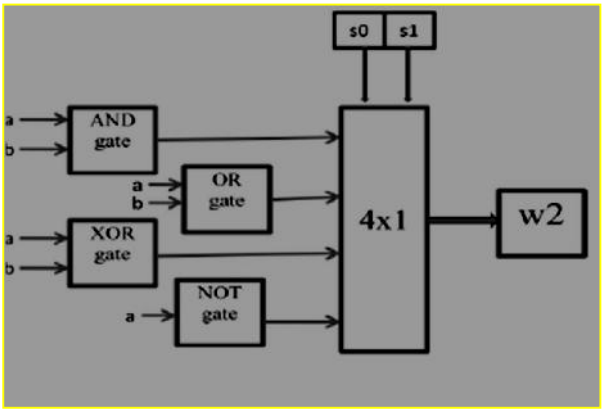

Fig 2.3: Logic Unit

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

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 effective approach to perform division and augmentation of numbers by power of two. Moving left by k bits on a binary number is proportionate to increasing it by 2k .Comparatively moving appropriate by k bits on a binary number is identical to isolating it by 2k. For instance, consider the double number 0001 0111.

a) Arithmetic shifts: Arithmetic shift can be helpful as effective methods for performing multiplication or division of marked whole numbers by forces of two. Shifting left by n bits on a signed or unsigned double number has the impact of increasing it by 2n. shifting right by n bits on a two's complement signed binary number has the impact of partitioning it by 2n, yet it generally adjusts down (towards negative interminability). Number juggling left move [6] move bits to one side, same request discard the bit that flies off the MSB bring a 0 into the LSB. This is same as coherent left move. It is appeared in figure 2.4.a. Number juggling right move bits to one side, same request discard the bit that flies off the LSB repeat the first MSB into the new MSB as appeared in figure 2.4.b.

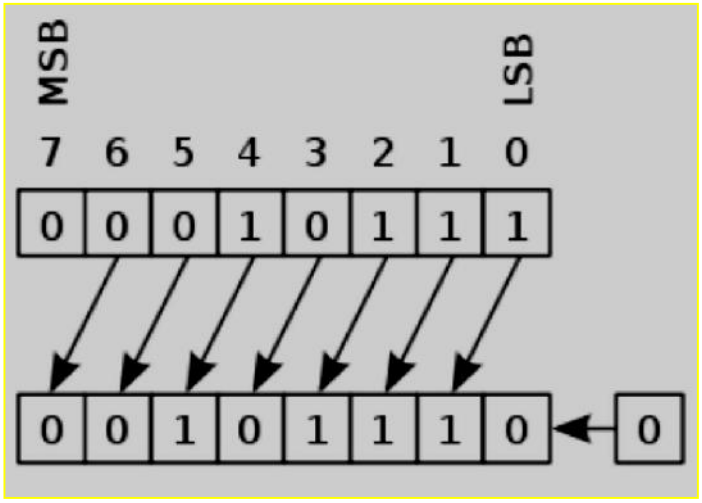

Fig 2.4.a: Arithmetic Left Shift

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

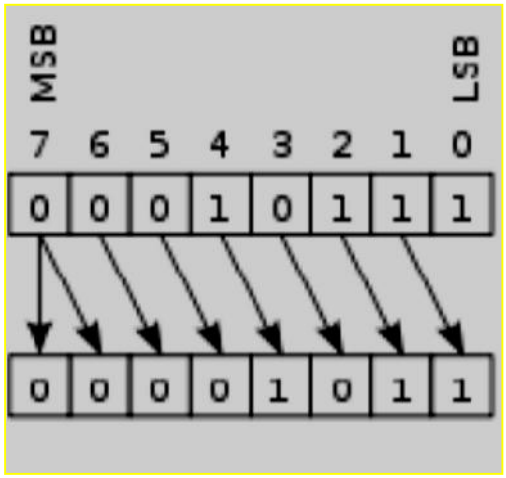

Fig 2.4.b: Arithmetic Right Shift

b) Logical Shifts: logical shift is a bitwise operation that moves every one of the bits of its operand. The two base variations are the legitimate left move and the intelligent right move. This is additionally adjusted by the quantity of bit positions a given esteem should be moved, similar to "move left by 1" or a "move ideal by n". Dissimilar to a arithmetic shift, a legitimate move does not protect a number's sign piece or recognize a number's type from its mantissa; each piece in the operand is essentially moved a given number of bit positions, and the empty piece positions are filled in, more often than not with zeros [7]. Coherent left move bits to one side, same request discard the bit that flies off the MSB bring a 0 into the LSB as appeared in figure 2.5.a.

Consistent right move bits to one side, same request discard the bit that flies off the LSB bring a 0 into the MSB as appeared in figure 2.5.b.

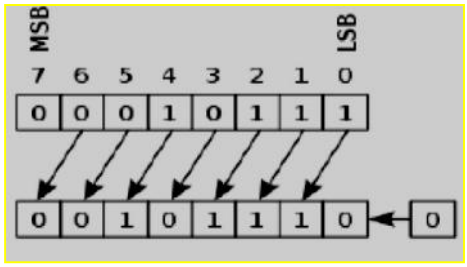

Fig 2.5.a: Logical Left Shift

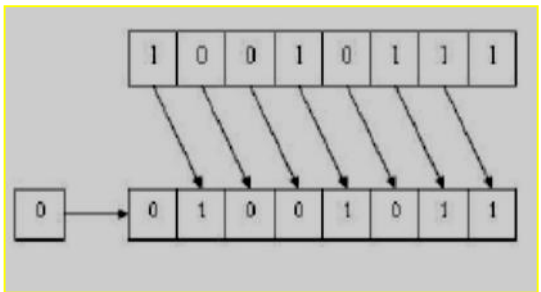

Fig 2.5.b: Logical Right Shift

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

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.

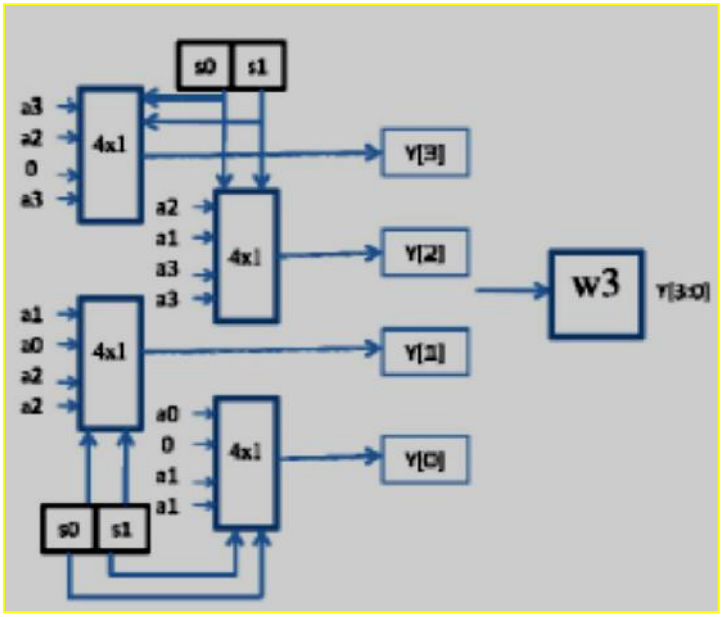

Fig 2.6: Shifter Unit

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 a N bit multiplicand by N bit multiplier is demonstrated as follows, AND logic's are utilized to create the Halfway Items, PP, If the multiplicand is N-bits and the multiplier is M-bits at that point there is N* M incomplete item. How the halfway items are produced or summed up is the contrast between the diverse designs of different multipliers. Increase of binary numbers can be decayed into augmentations. Consider the duplication of two 8-bit numbers A and B to create the 16 bit item P. 1. On the off chance that the LSB of Multiplier is „1", at that point include the multiplicand into an aggregator. 2. Move the multiplier one piece to one side and multiplicand one piece to one side. 3. Stop when all bits of the multiplier are zero. Speed of the Processor essentially relies upon Multiplier execution. There are A few Systems for outline of Multipliers. We have to choose the fitting procedure in view of variables delay, throughput, and territory and outline multifaceted nature.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

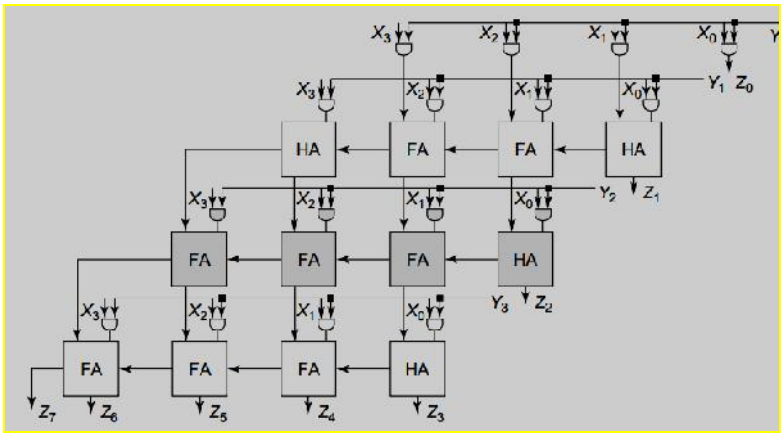

Fig 2.7: 4x4 Array Multiplier

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

A. Software Approach

The VHDL programming interface utilized as a part of this outline lessens the multifaceted nature and furthermore gives a realistic introduction of the framework. The key favourable position of VHDL when utilized for frameworks configuration is that it permits the conduct of the expected framework to be portrayed (displayed) and checked (reproduced) before synthesis apparatuses make an interpretation of the plan into genuine equipment (entryways and wires). This product aggregates the given VHDL code as well as produces waveform comes about.

1) Hardware Approach:

The VHDL code which suggests the equipment part of ALU is downloaded on FPGA processor utilizing JTAG link interfacing PC and the equipment component. A last point is that when a VHDL demonstrate is converted into the "doors and wires" that are mapped onto a programmable rationale gadget i.e FPGA, and after that it is the genuine equipment being designed, as opposed to the VHDL code being "executed" as though on some type of a processor chip.

IV. SIMULATION RESULT OF 16 BIT ALU DESIGN

A. OR operation

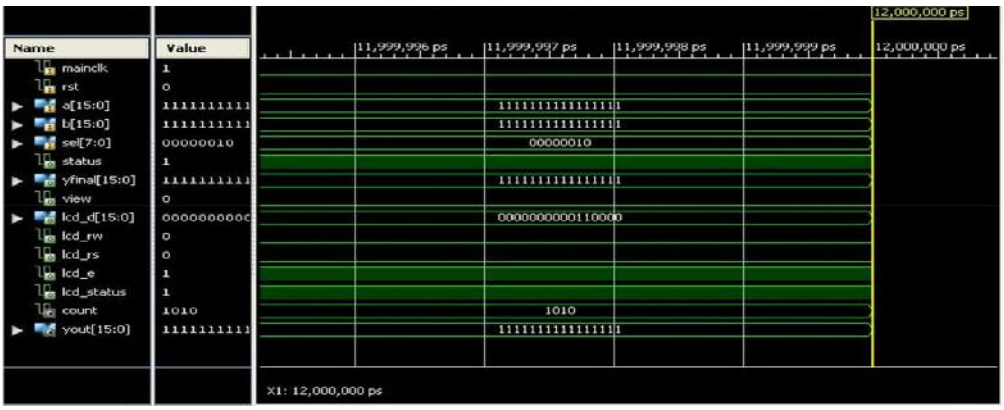

Fig 4.1: Waveform of OR operation.

International Journal of Engineering Technology Science and Research
IJETSR
www.ijetsr.com
ISSN 2394 – 3386
Volume 4, Issue 9
September 2017

1) XOR Operation

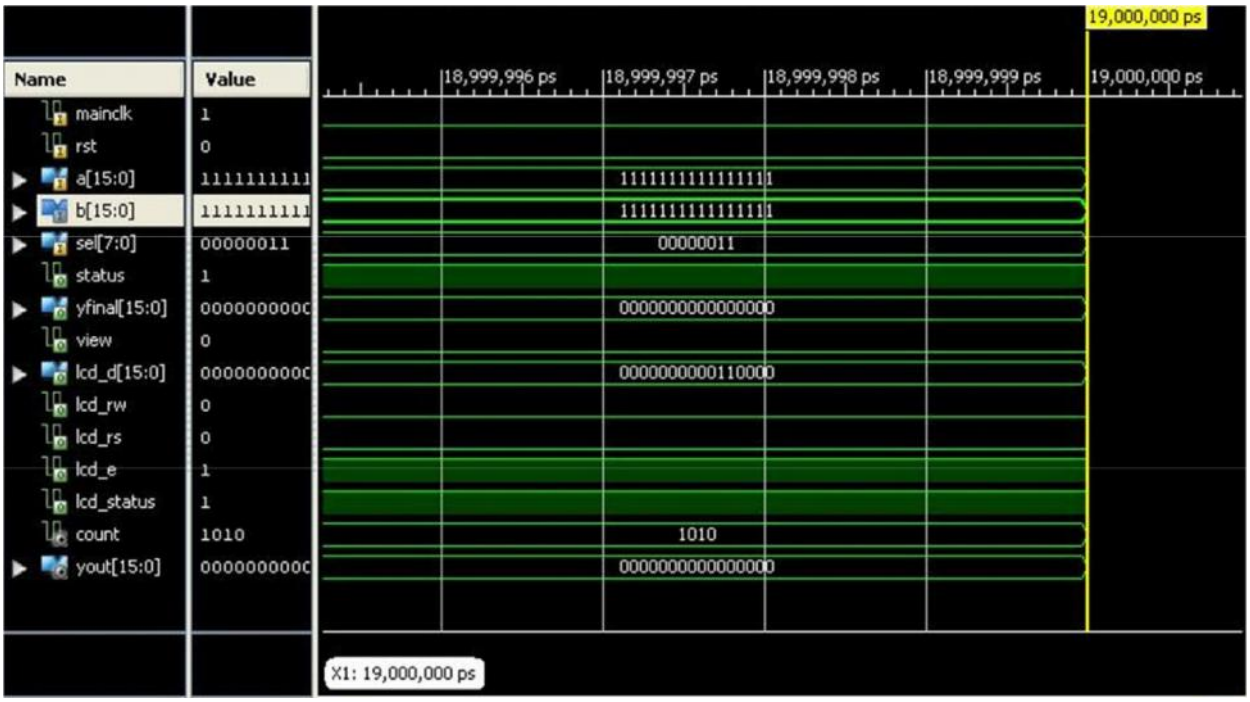

Fig 4.2: Waveform of XOR operation

V. CONCLUSION

This study comprehended the total stream of RTL configuration, beginning from planning a best level RTL module for 16-bit ALU utilizing HDL, VHDL. Confirmation of the composed RTL code utilizing reproduction methods, combination of RTL code to acquire door level netlist utilizing Xilinx Vivado ISE device and arithmetic logical Unit was effectively planned and actualized utilizing very high speed hardware description 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.