THOMSON REUTERS ENDNOTE

[Ukande* *et al.*, 5(10): October, 2016]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 4.116
CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

HIGH SPEED LOW POWER 32 BIT ALU IMPLEMENTATION

Vikesh Ukande*, Ankit Pandit
[*] M.Tech Scholar, Department of ECE, AISECT University, Institute of science and technology, Bhopal
Assistant Professor, Department of ECE, AISECT University, Institute of science and technology, Bhopal

DOI: 10.5281/zenodo.159286

ABSTRACT

The main objective of project is to design and verify different operations of Arithmetic and Logical Unit (ALU). We have designed an 32 bit ALU which accepts two 32 bits numbers and the code corresponding to the operation which it has to perform from the user. The ALU performs the desired operation and generates the result accordingly. The different operations are arithmetical, the coding was written in VHDL and verified in I-Sim. The waveforms were obtained successfully. After the coding was done, the synthesis of the code was performed using Xilinx-ISE.

KEYWORDS: logic unit, arithmetic unit, shift unit, arithimatic and logic unit.

INTRODUCTION

The Arithmetic Logic Unit (ALU) is a fundamental building block of the Central Processing Unit (CPU) of a Computer. Even one of the simplest microprocessor contains one ALU for purposes Such as maintaining timers. We can say that ALU is a core component of all central processing unit within in a computer and is an integral part of the execution unit. ALU is capable of calculating the results of a wide variety of basic arithmetical and logical computations. 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. The output is the result of the computation. The ALU implemented will perform the following operations: Arithmetic operations (addition, subtraction increment, decrement, transfer) Logic operations (AND, NOT, OR, NAND, NOR, EX-OR, EX-NOR) The output of the circuit is calculated from ALU.

Microprocessors/Microcontrollers have a single module that performs arithmetic operations on integer values. This is because many of the different arithmetic and logical operations can be performed using similar (if not identical) hardware. The component that performs the arithmetic and logical operations is known as the Arithmetic Logic Unit, or ALU. The ALU is one of the most important components in a microprocessor, and is typically the part of the Processor that is designed first. Once the ALU is designed, the rest of the microprocessor is implemented to feed operands and control codes to the ALU.

FUNCTION TABLE OF ALU

In our project "Design and Implementation of a 32-bit ALU on Xilinx using VHDL" we have designed and implemented a 32 bit ALU. Arithmetic Logic Unit is the part of a computer that performs all arithmetic computations, such as addition and subtraction, increment, decrement, shifting and all sorts of basic logical operations. The ALU is one component of the CPU (Central Processing Unit). The Table Gives Below show The Various Function Perform By The ALU In This Project

Table 1.1 functions of ALU

S_3	S_2	S_1	S_0	C_{in}	RESULT	Operation
0	0	0	0	0	A + B	Addition
0	0	0	0	1	A + B + 1	Addition with carry
0	0	0	1	0	A + $\bar{B}$	Subtraction with borrow
0	0	0	1	1	A + $\bar{B}$ + 1	Subtraction
0	0	1	0	0	A − 1	Decrement
0	0	1	0	1	A	Transfer
0	0	1	1	0	A	Transfer
0	0	1	1	1	A + 1	Increment
0	1	0	0	x	AB	AND
0	1	0	1	x	A+B	OR
0	1	1	0	x	A ⊕ B	XOR
0	1	1	1	x	$\bar{A}$	Complement
1		0		x	A	Shift Right
1	0	1	x	x	LSL A	Shift Left

The ALU performs the desired operation and generates the result accordingly. The different operations are arithmetical, logical and relational. Arithmetic operations include arithmetic addition, subtraction, multiplication and division. Logical operations include AND, OR, NAND, XOR, NOT and NOR. These take two binary inputs and result in output logically operated. To implement ALU, the coding was written in VHDL and verified in I-Sim. The waveforms were obtained successfully. After the coding was done, the synthesis of the code was performed using Xilinx-ISE. Circuit we will first design 32 bit Adder, Subtractor, OR, AND, NOT, XOR, LEFT Shift, Right Shift Unit. These bit-slices can then be put together to make a 32-bit Adder, Subtractor, OR, AND, NOT, XOR, Left shift, right shift unit Fig. 1 Show RTL view of ALU.

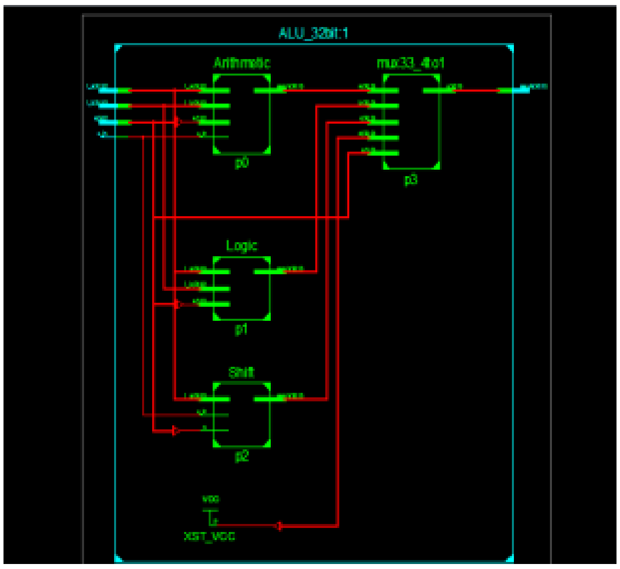

Fig. 1 Show RTL view of block diagram ALU.

32-BIT LOGIC UNIT

A Logic unit does the following task: Logical AND, Logical OR, Logical XOR and Logical NOT operation. We will design a logic Unit that can perform the four basic logic micro-operations: OR, AND, XOR and Complement, because from these four micro-operations, all other logic micro-operations can be derived. A one-stage logic unit for these four basic micro-operations is shown in the Fig. 2. The logic unit consists of four gates and a 4:1 multiplexer. The outputs of the gates are applied to the data inputs of the multiplexer. Using to selection lines S_0 and S_1 one of the data inputs of the Multiplexer is selected as the output. For a logic unit of 32-bit, the output will be of 33-bit with 33th bit to be High-impedance. The common selection lines are applied to all the stages.fig.1 shown RTL view of 32 bit ALU logic unit which represent OR Gate NOT Gate , OR Gate, NOR Gate

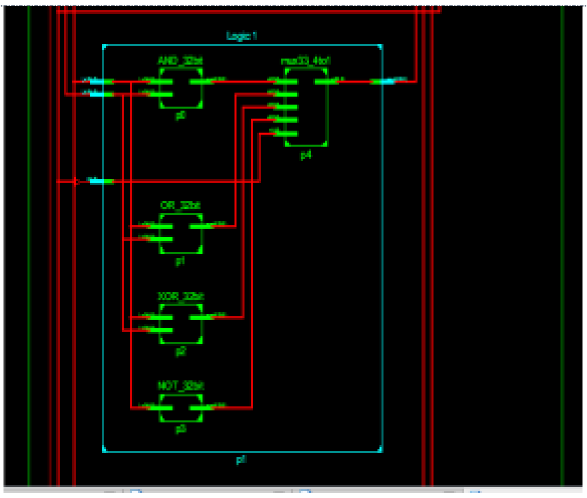

Fig.1 shown RTL view of 32 bit ALU logic unit

32-BIT ARITHMETIC AND LOGICAL UNIT

32-bit Arithmetic and Logical Unit The approach used here is to split the ALU into three modules, one Arithmetic, one Logic and one Shift module. The arithmetic, logic and shifter units introduced earlier can be combined into ALU with common selection lines. The shift micro-operations are often performed in a separate unit, but sometimes the shifter unit made part of overall ALU. Since the ALU is A particular arithmetic or logic or shift operation is selected according to the selection inputs S_0 and S_1. The final output of the ALU is determined by the set of multiplexers with selection lines S_2 and S_3. The function table for the ALU is shown in the Table. 1.

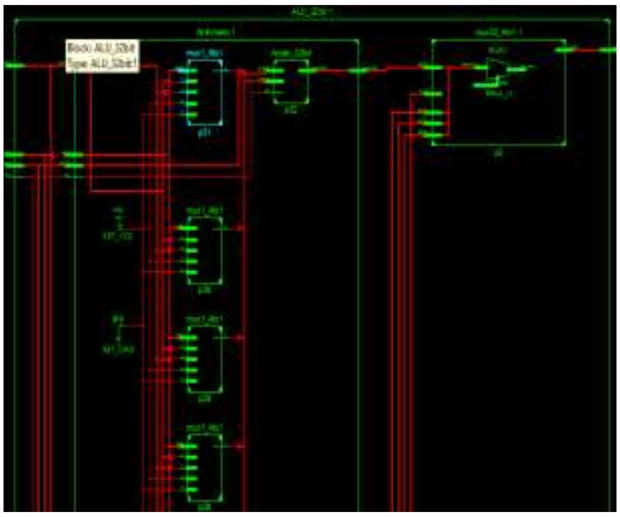

Fig. 2 shown RTL view of 32 bit ALU logic unit

The table lists 14 micro-operations: 8 for arithmetic, 4 for logic and 2 for shifter unit. For shifter unit, the selection line S_1 is used to select either left or right shift micro-operation.

32-BIT SHIFTER UNIT

Shifter unit is used to perform logical shift micro-operation. The shifting of bits of a register can be in either direction- left or right. A combinational shifter unit can be constructed as Fig. 7. The content of a register that has to be shifted first placed onto common bus. This circuit uses no clock pulse.

For a shift unit of 32-bit, the output will be of 33-bit with 33th bit to be the outgoing bit. The circuit of shift unit is shown in Fig. 8.

When S_1= 0:	RESULT= Shift Right A.
When S_1= 1:	RESULT= Shift Left A.

When the shifting unit is activated the register is shifted left or right according to the selection unit.

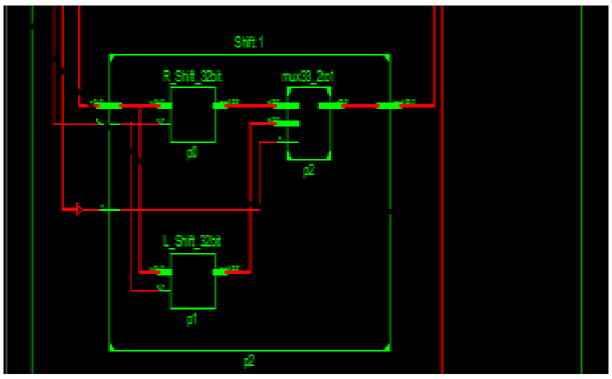

Fig. 3 shown RTL view of 32 bit ALU shifting logic unit

SIMULATION & RESULT

Once the all VHDL modules are prepared, they should be simulated before they are put in actual Hardware chip. We can generate a test counter waveform from the Project, New Source menu of ISE and it will support in setting up the simulation. Once we simulate our design and feel it is functioning properly, and then we can

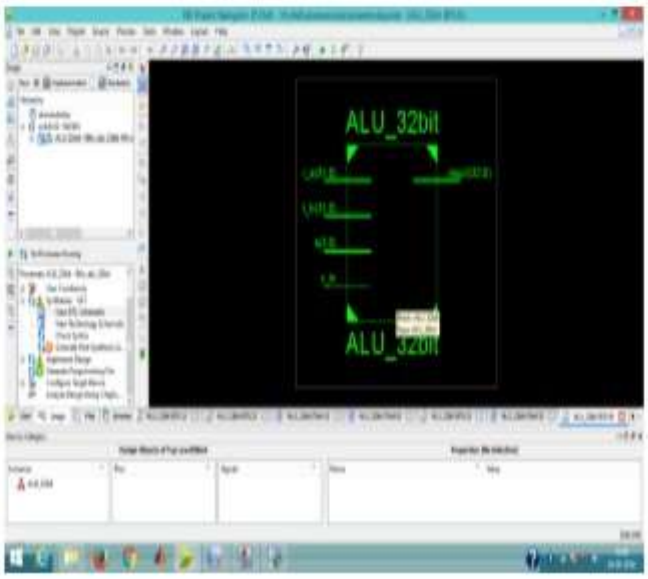

Fig. 5 Shown Simulation of 32 bit ALU

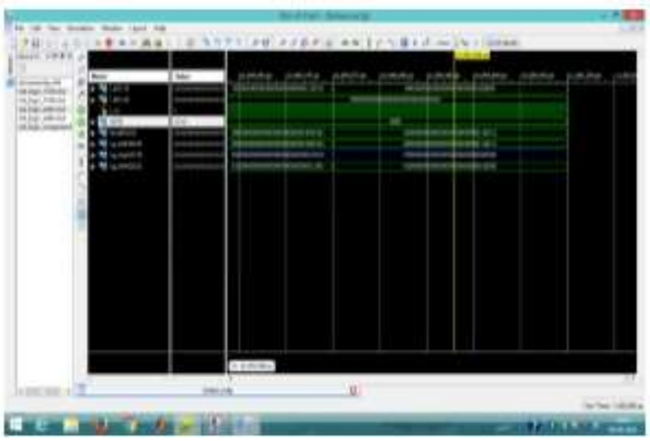

Fig. 6 Shown Simulation of 32 bit ALU

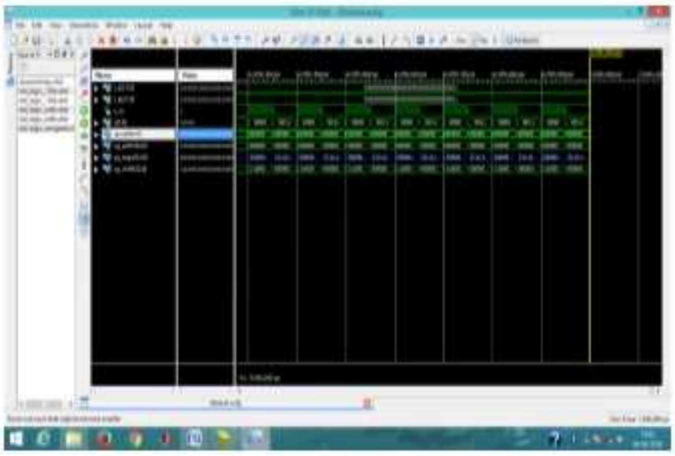

Fig. 6 Shown Simulation of 32 bit ALU

Move on to generating the data needed to essentially program

COMPARISON TABLE

design	delay (ns)	power [w]	area
Design [1]			
Proposed desi	Delay: 33.468ns (Levels o Logic = 37)	0166	193 used out of 10944 [1%]
Design [a]	-	120 @3.30V	-

CONCLUSION

In our project "Design and Implementation of a 32-bit ALU on Xilinx Fusing VHDL" we have designed and implemented a 32 bit ALU. Arithmetic Logic Unit is the part of a computer that performs all arithmetic computations, such as addition and subtraction, increment, decrement, shifting and all sorts of basic logical operations. The ALU is one component of the CPU (Central Processing Unit). Here, using VHDL we have designed a 32 bit ALU which can perform the various arithmetic

REFRENCES

[1] Design and Implementation of 4-Bit Arithmetic and Logic Unit Chip with the Constraint of Power Consumption Priyanka Yadav, Gaurav Kumar, Sumita Gupta Assistant Professor Department of Electronics & Communication HITM,Agra Research Analyst ONE97 Communication LTD.
[2] Noida Assistant Professor Department of Electronics & Communication HITM,Agra
[3] N. Shirazi, A. Walters, and P. Athanas, "Quantitative analysis of floating point arithmetic on fpga based custom computing machines," in Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, pp. 155–162, 1995.
[4] P. Belanovic and M. Leeser, "A library of parameterized floating-point modules and their use,"in Proceedings of the International Conference on Field Programmable Logic and Applications, 2002.
[5] J. Dido, N. Geraudie, L. Loiseau, O. Payeur, Y. Savaria, and D. Poirier, "A flexible floating-point format for optimizing data-paths and operators in fpga based dsps," in Proceedings of the ACM International Symposium on Field Programmable Gate Arrays, (Monterrey, CA), February 2002.

[6] A. A. Gaar, W. Luk, P. Y. Cheung, N. Shirazi, and J. Hwang, “Automating customisation of floatingpoint designs,” in Proceedings of the International Conference on Field Programmable Logic and Applications, 2002.

[7] J. Liang, R. Tessier, and O. Mencer, “Floating point unit generation and evaluation for fpgas,” in Proceedings of the IEEE Symposium on FieldProgrammable Custom Computing Machines, (Napa Valley, CA), pp. 185–194, April 2003.

[8] IEEE Standards Board, “IEEE standard for binary floating-point arithmetic,” Tech. Rep. ANSI/IEEE Std. 754-1985, The Institute of Electrical and Electronics Engineers, New York, 1985.

[9] B.Fagin and C. Renard, “Field programmable gate arrays and floating point arithmetic,” IEEE Transactions on VLSI, vol. 2, no. 3, pp. 365–367, 1994.

[10] W. B. Ligon, S. P. McMillan, G. Monn, F. Stivers, K. Schoonover, and K. D. Underwood, “A reevaluation of the praticality of floating-point on FPGAs,” in Proceedings of the IEEE Symposium on FPGAs for Custom Computing Machines, (Napa Valley, CA), pp. 206–215, April 1998.

[11] Z. Luo and M. Martonosi, “Accelerating pipelined integer and floating-point accumulations in configurable hardware with delayed addition techniques,” IEEE Transactions on Computers, vol. 49, no. 3, pp. 208–218, 2000.

[12] X. Wang and B. E. Nelson, “Tradeoffs of designing floating-point division and square root on virtex fpgas,” in Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines, (Napa Valley, CA), pp. 195–203, April 2003.

[13] IEEE 754-2008, IEEE Standard for Floating-Point Arithmetic, 2008.

[14] Simulink HDL Coder 1; User‟s Guide; 2006-2010 by the MathWorks, Inc.

[15] Hikmat N. Abdullah and Hussein A. Hadi “Design and Implementation of FPGA Based Software Defined Radio Using Simulink HDL Coder”. Engineering and Technology Journal, Iraq, ISSN 1681-6900 01/2010; Vol.28 (No.23):pp.6750-6767.

[16] B. K. Mishra, S. Save, R. Mane. A frame work for model based designing of analog circuits using Simulink. ICWET '11 Proceedings of the International Conference & Workshop on Emerging Trends in Technology. Pages 1225-1228;