Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

DESIGN OF FPGA BASED 8 BIT RISC PROCESSOR

S.V. KULKARNI[1,*], A.I. NADAF[2], P.P. SHAH[1], M.K. BHANARKAR[2]
[1] Department of Electronics, Devchand College, Arjunnagar, MS-India
[2] Department of Electronics, Shivaji University, Kolhapur, MS-India
sachineln@gmail.com

ABSTRACT

After a wide survey of Embedded Applications, it is found that the eight bit microcontroller continues to play a major role in the consumer products. The trend is now become a customizing or semi-customizing processor for many of the application. In this manner FPGA play a major role for customizing the processor and reconfigure the many of the electronics. The main purpose of this paper is to design 8 bit RISC processor using Spartan 6E tool by interfacing the kit with computer. The basic modules of this processor are programmed by using Verilog Hardware Description Language (VHDL), it is then verified the simulation result using XILLINX ISE 12.4 tool..

Keywords: *FPGA, RISC Processor, VHDL, Simulation*

1. INTRODUCTION

FPGA- Field Programmable Gate Array; it is an integrated circuit designed and configured by a customer or designer after manufacturing hence called “Field Programmable”. The specification of configuration is normally done by using Hardware Description Language (HDL). It contains Logic Components which are programmable called Logic box and it has the ability to interconnect that cell or blocks to be wired together. It contains over 10000 logic cells. The individual cells are interconnecting by a matrix of wires and programmable switches [1].

FPGA plays a major role for customizing the processor and reconfigures the many of the electronics. The Hardware Description Languages (HDLs) increase the range of options available to FPGA designers by enabling hardware implementation with the flexibility that language based design provides HDLs allow designers to implement flexible intellectual Property (IP), often referred to as IP cores. IP is the implementation of reusable components, which describe and implement hardware functionality [2].

Field Programmable Gate Arrays (FPGA) is growing fast with cost reduction compareto ASIC design.This resea rchconcerned with the design and implementation of a lowcost8-bit Reduced Instruction Set Computer (RISC) processor on a FPGA. It provide the benefits of custom VLSI design while avoiding the initial cost, time delay and inherent risk of a conventional masked gate array [3]. They are customized by loading configuration data into the internal memory cell. RAM based FPGA's can be infinitely reprogrammed in-circuit in only a fraction of seconds. Design revisions even for fielded products can be implemented quickly and precisely [4].

RISC stands for Reduced Instructions Set Computer (RISC). It is a processor which uses only simple instructions, which performs low level operation in a single clock cycle. It has a very high performance capacity and capable of executing those instructions in a single microprocessor cycle. It has advantage of having pipelining therefore multiple instructions can execute in a single clock cycle. This results in high speed instruction execution. It also has Load/Store architecture where memory is accessed through particular instructions. It also has a simple Arithmetic Logic Unit (ALU) for basic operations with simple and uniform instruction set [5]. This work presents the design of reconfigurable 8 bit RISC processor.

Verilog HDL is a standard language which describes hardware of a digital system therefore it is called Hardware Description Language (HDL). It allows the user to design to be simulated earlier before implement to correct errors or experiments with different architectures [5].

2. RISC PROCESSOR ARCHITECTURE

RISC architecture was first developed by IBM in 1970’s and completed by T. J. Watson at research centre in1980’s. He was started with frequently used simple instructions. IBM was the first company to define Reduced Instruction Set Computer (RISC) and further research was done by Universities of Berkeley and Stanford to give Basic Architectural models [5]. All the processors are designed around two techniques; Complex Instruction Set Computer (CISC) and Reduced Instruction Set Computer (RISC). The CISC is the concept that approaches to the Instruction Set Architecture (ISA) which is based on doing more work in one instruction with variety of addressing modes. Due to instruction has variable length, they are generally implemented using micro programmed model. [7] Some characteristics of CISC are as follows.

- It depends on Hardware.
- Instructions can execute in many clock cycles.
- Instruction set is complex.
- Memory to Memory Load and store incorporated in instructions.
- Cycle period per instruction is high.
- Pipelining is not possible.

Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

As compared to CISC; RISC processor works on simple instructions with fixed length of instructions that results in faster execution of instructions per clock cycle as compared to CISC. Therefore nowadays RISC becomes famous and becomes more important device for computer systems [6]. Some important characteristics of RISC processor is as fallows.

- It uses hardwired units.
- Instructions are small therefore executes in a single clock cycle.
- It has reduced instruction set.
- Instruction length and size is same.
- It has general purpose registers and simple addressing modes.
- Register to Register Load and Store are independent instructions.
- Code is large but cycle period is low.
- Pipelining is possible.

3. Design of Arithmetic Logic Unit (ALU)

The fig 4 and 5 shows block diagram and RTL schematic of ALU respectively. It contains inputs ports 'a' and 'b' which is 8 bit wide and select line 's' which is 4 bit wide. The output is obtained at out1 line, which is also 8 bit wide. The ALU performs arithmetic and logical operations such as AND, OR, NOT, NOP, NAND, NOR, XOR etc. Table 1 shows details of arithmetic and logical instruction operations The instructions designed for ALU are 8 bit wide. Furthermore, ALU contains logic gates, adder, substractor, multiplexers etc for the intended operation.

Table 1:- Arithmetic and Logical instruction operations.

Sr. No	Select Lines	Operation	Inputs		Output Equation	Calculated output	Observed Output (Out 1)
			a	b			
LOGICAL OPERATIONS							
1	0000	No Change	10101010	------	Out= a	10101010	10101010
2	0001	AND	10101010	01000011	Out=a*b	00000010	00000010
3	0010	OR	10101010	01000011	Out=a+b	11101011	11101011
4	0011	NOT	10101010	------	$out=\overline{a}$	01010101	01010101
5	0100	XOR	01010101	01110111	$out=\overline{a}b+a\overline{b}$	00100010	00100010
6	0101	NOR	10101010	01000011	$out=\overline{a+b}$	00010100	00010100
7	0110	NAND	10101010	01000011	$out=\overline{a*b}$	111111101	111111101
ARIHMETIC OPERATIONS							
1	0111	ADD	01010101	01111000	Out=a+b	11001101	11001101
2	1000	INCREMENT BY 1	01010101	-------	Out=a+1	01010110	01010110
3	1001	SUB	01010101	01010011	Out=a-b	00000010	00000010
4	1010	DECREMENT BY 1	01010101	-------	Out=a-1	01010100	01010100

Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

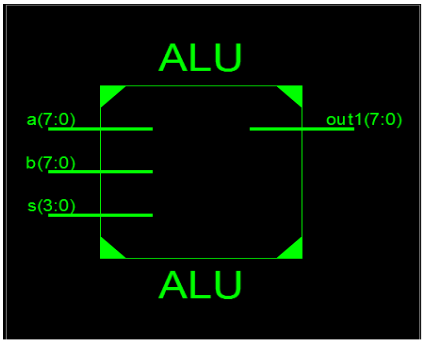

Fig 4:- Block diagram of ALU

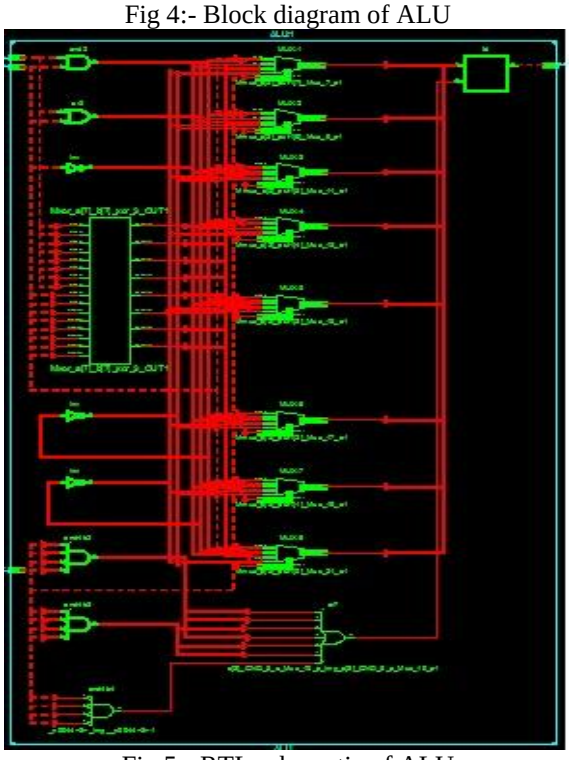

Fig 5:- RTL schematic of ALU

The logical instructions are designed to perform logical operations such as No Change (NOP) operation, AND, OR, NOT, NAND, NOR, XOR operations etc. All logical instructions are 8 bit wide. The details of each operation are as follows,

3.1 No Change operation (NOP)

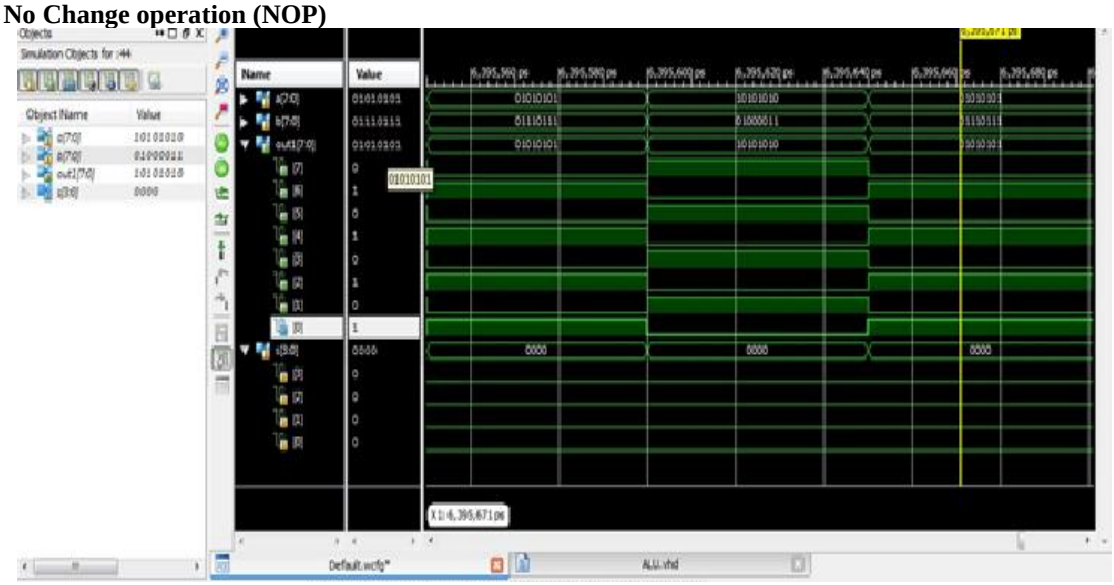

Fig:- 6 timing diagram of No change Operation (NOP)

Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

In NO CHANGE operation the output is same as input (a) that is there is no change in output. Fig 6 shows 8 bit NOP operation.

Select line (s) = 0000
Output equation out=a
Input (a) =10101010
Output (out1) =10101010

3.2 AND Instruction

In AND operation the ANDING between two inputs a and b is done. The fig 7 shows timing diagram for AND operation.

Select line (s) = 0001
*Output equation out=a*b*
Input (a) =10101010
Input (b) =01000011
Output (out1) =00000010

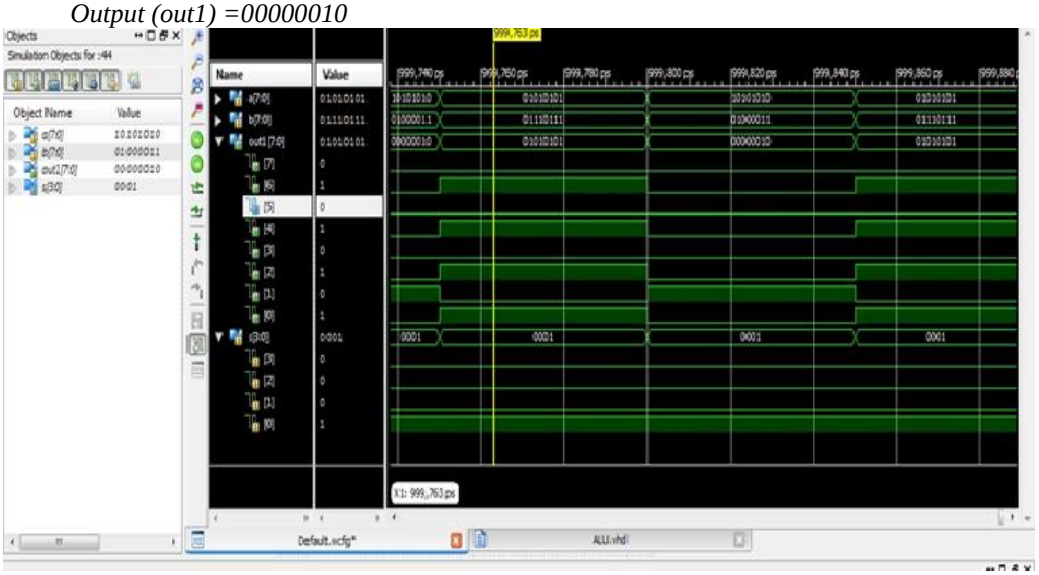

Fig 7:- Timing diagram of AND operation

3.3 OR Instruction

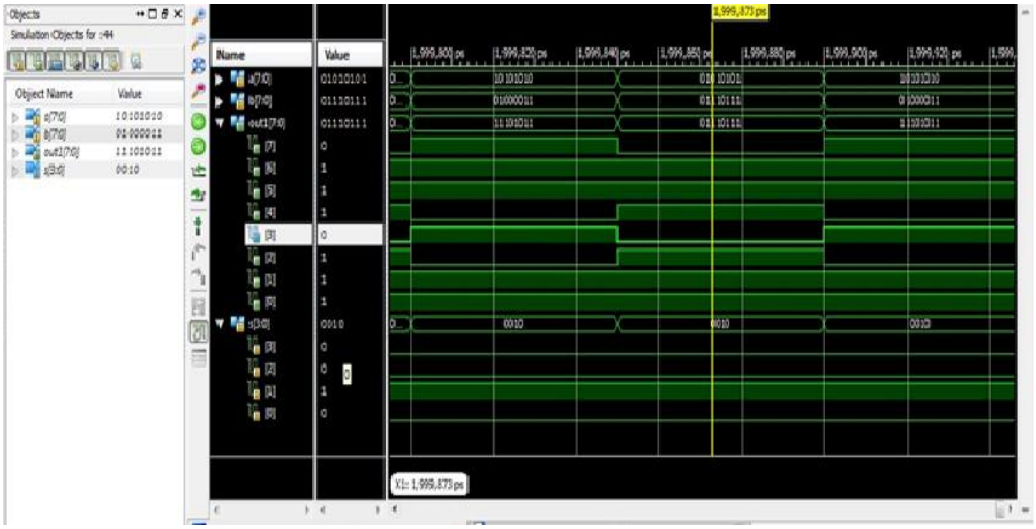

Fig 8:- Timing diagram of OR operation

In OR operation the ORING between two inputs a and b is done. The fig 8 shows timing diagram for OR operation.

Select line (s) = 0010
Output equation out= a+b
Input (a) =10101010
Input (b) =01000011
Output (out1) =11101011

Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

3.4 NOT Instruction

Fig 9:- Timing diagram of NOT operation

In NOT operation the complement of input a is done. The fig 9 shows timing diagram for NOT operation. The output gives complement value of input. Therefore it is also called as inverter.

Select line (s) = 0011

output equation=$\overline{a}$

Input (a) =10101010
Output (out1) =01010101

3.5 XOR Instructions

In XOR operation the XORING between two inputs a and b is done. The fig 10 shows timing diagram for XOR operation.

Select line (s) = 0100

Output Equation=$\overline{a}b+a\overline{b}$

Input (a) =01010101
Input (b) =01110111
Output (out1) =00100010

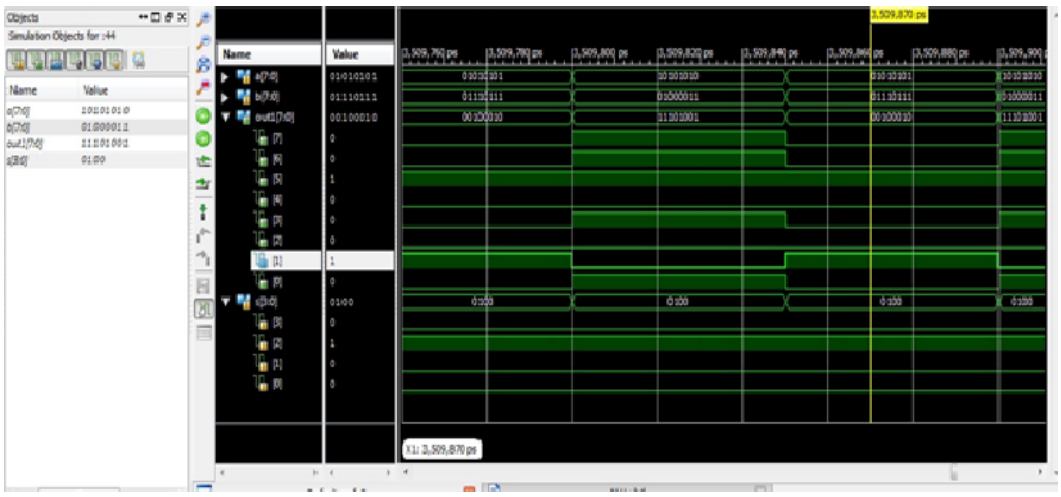

Fig 10:- Timing diagram of XOR operation

3.6 NOR Instruction

It is the combination of OR and NOT gate. NOR gate performs operation that of compliment of OR gate. The fig 11 shows timing diagram for NOR operation.

Select line (s) = 0101

Output equation=$\overline{a+b}$

Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

Input (a) =10101010
Input (b) =01000011
Output (out1) =00010100

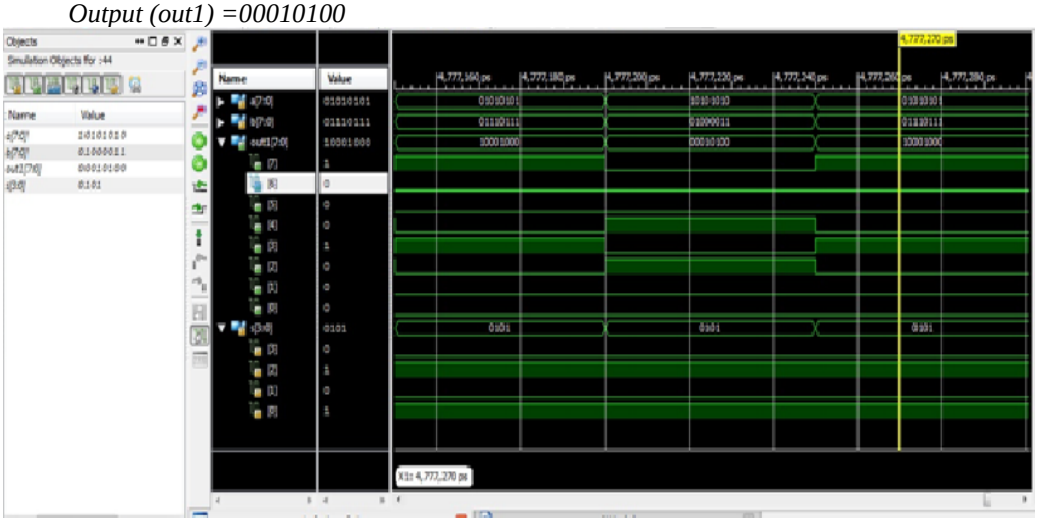

Fig 11:- Timing diagram of NOR operation

3.7 NAND Instruction

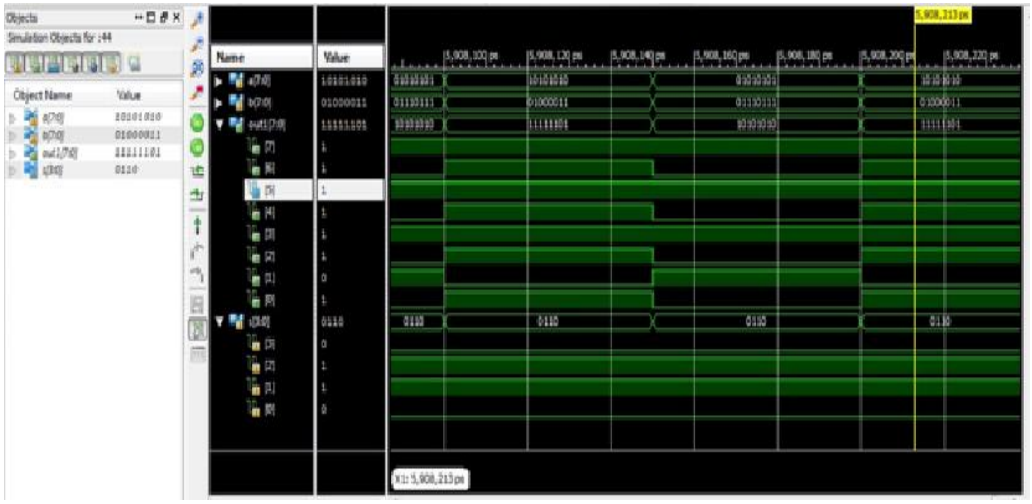

Fig 12:- Timing diagram of NAND operation

It is the combination of AND and NOT gate. NAND gate performs operation that of compliment of AND gate. The fig 12 shows timing diagram for NAND operation.
Select line (s) = 0110

$$\text{Output Equation} = \overline{a*b}$$

Input (a) =10101010
Input (b) =01000011
Output (out1) =11111101

3.8 ADD Instruction

The arithmetic instructions are designed to perform arithmetic operations such as addition, subtraction, increment, decrement etc. All arithmetic instructions are 8 bit wide. ADD instruction performs addition of two inputs a and b. The fig 13 shows timing diagram for ADD operation.
Select line (s) = 0111
Output equation out= a+b
Input (a) =01010101
Input (b) =01111000
Output (out1) =11001101

Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

Fig 13:- Timing diagram of ADD operation

3.9 SUB Instructions

SUB instruction performs subtraction between two inputs a and b. The fig 14 shows timing diagram for SUB operation.

Select line (s) = 1001

Output equation out=a-b
Input (a) =01010101
Input (b) =01010011
Output (out1) =00000010

Fig 14:- Timing diagram of SUB operation

3.10 INCREMENT (INC) Instruction

Increment instruction increments the value of a by one. The fig 15 shows timing diagram for INC operation.

Select line (s) = 1000

Output equation out=a+1
Input (a) =01010101
Output (out1) =00000110

Fig 15:- Timing diagram of INC operation

Vol 07, Article 05675; June 2016
http://ijves.com
International Journal of VLSI and Embedded Systems-IJVES
ISSN: 2249 – 6556

3.11 DECREMENT (DEC) Instruction

Fig 16:- Timing diagram of DEC operation

DECREMENT (DEC) instruction decrements the value of a by one. The fig 16 shows timing diagram for DEC operation.

Select line (s) = 1010

Output equation out=a-1
Input (a) =01010101
Output (out1) =01010100

4. Conclusion

The present paper deals with the design of FPGA based 8 bit RISC processor. The basic modules of this processor are programmed by using Verilog Hardware Description Language (VHDL), it is then verified the simulation result using XILLINX ISE 12.4 tool. The present 8 bit ALU performs arithmetic and logical operations such as AND, OR, NOT, NOP, NAND, NOR, XOR etc. Furthermore, ALU contains logic gates, adder, substractor, multiplexers etc for the intended operation. In the nutshell, the designed ALU performs its intended operations.

5. REFERENCES

1. Deepak Kumar, K. Anusudha, (2013) "RISC processor design in Xilinx" International Journal of advance research in Electrical, Electronics and Instrumentation Engineering. Vol2 pp. 1406-1411
2. Application note for FarhatMasood, RISC and CISC Computer Architecture, available at http://arxiv.org/ftp/arxiv/papers/1101/1101.5364.pdf, retrieved from: 11/07/201**2**
3. RC Cofer and Ben Harding, (2005) "FPGA Soft Processor Design Considerations", Available at http://www.eetimes.com/General/PrintView/4014791, online published, date 10/12/2005.
4. W.M Medany, K.Kooheji,"Design and Implementation of a 32-bit RISC Processor", Available at http://www.ursi.org/proceedings/procGA08/papers/DP1p2.pdf,
5. Luker, Jarrod D., Prasad, Vinod B. (2001) "RISC system design in a FPGA" MWSCAS 2001, vol2, pp. 532-536, 2001.
6. Rakesh M. R. "RISC processor design in VLSI technology using pipelining technique" International Journal of Innovative Research in Electrical. Electronics, Instrumentation and control engineering, Vol2, pp. 1359-1363.
7. Mishra K. K., Purwar R., Singh P., "International Journal of Electronics and Communication" vol3, PP 35-42, January 2015.