

FPGA Implementation of a 32-Bit MIPS Processor

Vishala A. Tyamanavar
M.Tech Student

Department of Electronics and
Communication Engineering,
SDM College of Engineering and Technology,
Dharwad, India

Jayashree C. Nidagundi
Asst.Professor

Department of Electronics and
Communication Engineering,
SDM College of Engineering and Technology,
Dharwad, India

Abstract—Proposed paper represents the FPGA implementation of a 32-bit Microprocessor without Interlocked Pipeline Stages (MIPS) processor architecture. This processor architecture consists of blocks like memory unit, controlling unit, program counter, adder, sign expanded, multiplexers, data memory and ALU. The Verilog HDL language is used to design the MIPS processor architecture. The proposed processor execution speed is 290.166MHz.

Keywords— RISC, MIPS, Xilinx Tool, Verilog-HDL, RTL, FPGA

I. INTRODUCTION

Any Microprocessor and Microcontroller are designed by utilizing the two computing architectures, which are Complex Instruction Set Computing and Reduced Instruction Set Computing architectures. The CISC processor architecture depends on Instruction Set architecture. The CISC processor utilizes the more number of operands and addressing modes in its instruction set. The RISC processor uses a less number of directions compared with the CISC processor. The MIPS processor structured depends on the RISC processor. The MIPS processor has 32-bit architecture with 32 bit instructions, a 32 bit data and 32 bit addresses. It has 32 registers. The processor utilizes the 5 pipeline stages, which are Instruction Fetch (IF), Instruction Decoder (ID), Execute (EX), Memory Access (MEM) and Write Back (WB). The MIPS single-cycle processor plays out all the pipeline stages in one clock cycle.

The pipelined MIPS processor plays out every one of the stages in various clock cycles. In single cycle MIPS processor every one of the tasks takes a similar amount of time. In this processor we use instruction memory, data memory, an ALU, some additional adders, and progressively number of multiplexers.

The [1] concentrated on ASIC execution of 5 stages pipelined MIPS processor utilizing Verilog-HDL. The execution parameters like area, power dissipation and propagation delay all are determined by utilizing the Cadence RTL compiler.

In [2], the plan and execution of the low power 5 stages pipelined MIPS processor is proposed. In this the power is diminished by decreasing undesirable changes in the MIPS processor design.

The [3] planned the 16 bit optimized MIPS processor utilizing VHDL. This paper utilizes the two enhancement strategies for upgrading the processor. This processor is implemented on a SPARTAN 2E FPGA board.

The [4] focus on design and simulation of the 32 bit MIPS processor utilizing Xilinx device. In this paper the execution of each direction in the processor happens in a single cycle.

In [5], synthesis of the 32 bit MIPS processor utilizing VHDL is finished. In this processor every one of the stages are executed inside the single clock cycle. The 32 bit MIPS processor is implemented on a SPARTAN 2 FPGA board. This processor speed is 18.970MHz.

A Paper [6] executed the 32 bit MIPS processor with one CISC activity for augmentation which is realized by utilizing a Booth Multiplier. This processor is synthesized in a SPARTAN 3E FPGA board utilizing memory. This paper utilizes the SPARTAN 3E FPGA board for synthesis.

Paper [7] depicts the plan of an 8 bit pipelined RISC processor utilizing Verilog. RISC processor utilizes the different instruction memory and data memory. This paper utilizes the SPARTAN 3E FPGA board for synthesis.

The [8] concentrated on pipelined MIPS processor by utilizing the improved datapath. To diminish the undesirable changes, adjustment in the datapath is made.

II. BUILDING BLOCKS OF MIPS PROCESSOR

A. MIPS INSTRUCTION SET:

The instruction set in the MIPS processor design is grouped into 3 types. They are Instruction of type Register, Instruction of type Immediate and Instruction of type Jump. In all these arrangement of instructions the Opcode is of 6-bits, which is used to choose the type of operation in the ALU block.

A.1 Instruction of type Register (R):

This type of Instruction plays out all the arithmetic and logical operations. The principle utilization of this instruction is playing out the mathematical operations. For example, addition and subtraction. This type of instruction comprises of a instruction memory and ALU block. The instruction memory has 32 bit address and 32-bit instruction. Out of this 32 bits instruction the higher 6-bits are utilized for Opcode.

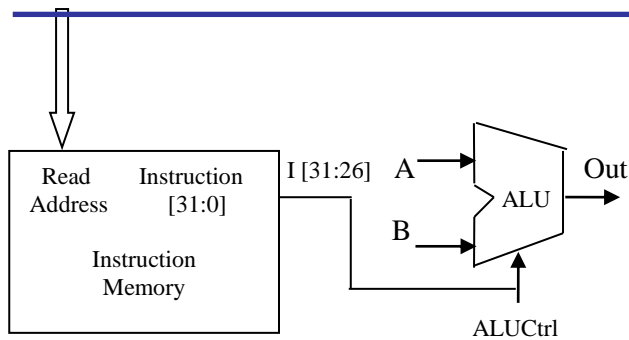


Fig.1. Data Path for the Instruction of type Register

Fig.1. demonstrates the Data Path for the Instruction of type Register. In this data path the ALU block plays out all the arithmetic and logical operations. For example, addition, subtraction, multiplication, division, AND, OR etc. The control signal for the ALU block is the 6 bit Opcode.

Ex: add Out, A, B

In this example we have to add the 32-bit inputs A and B values and the result is stored in the Out register.

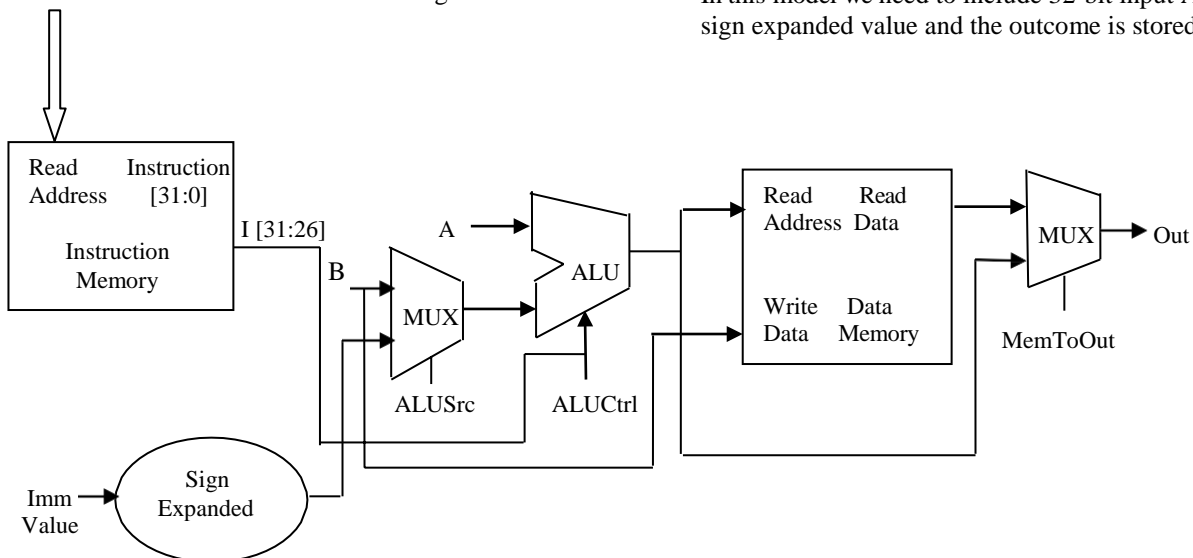


Fig.2. Data Path for the Instruction of type Immediate

A.3 Instruction of type Jump (J):

Jump Type instruction is the most part used to play out the jump instruction. In this instruction the Opcode is of 6-bit is used to pick the type of instruction. The target address is of 26-bits instruction is utilized to figure out where the location must be fanned. It is utilized just when the result of the ALU block is zero.

Fig.3 demonstrates the Data Path for the instruction of type Jump. In this type of instruction the 32 bit jump address is determined by concatenated the higher 4 bits of NPC values and the shifted left by 2 values of the 26 bit instruction. This 26 bit instruction is obtained from the instruction memory.

A.2 Instruction of type Immediate (I):

The Immediate type Instruction is utilized to play out the immediate arithmetic and logical operations. In this kind of instruction 16 bit immediate value is utilized for playing out all the operations. By utilizing this type of instruction we can plays out the load, store and branch operations. In this instruction one input is directly connected to the ALU block another input is either immediate value or the input is directly given to the ALU block, which is decided by the 2:1 multiplexer. In this the data memory is utilized to play out the load and store instructions.

Fig.2. demonstrates the Data Path for the instruction of type Immediate. By utilizing the immediate value it can plays out every one of the operations. The immediate value which is of 16 bit is given to sign expand block, which expand the 16 bit into the 32 bit, at that point the 32 bit value is sent to the ALU unit for playing out the ideal tasks. This type of instruction performs both the register type and immediate type instructions.

Ex: add Out, A, Immvalue

In this model we need to include 32-bit input A and 16-bit sign expanded value and the outcome is stored in Out register.

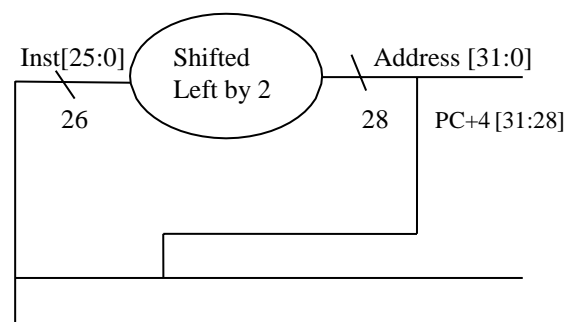


Fig.3. Data Path flow for the Instruction of type Jump

B. PROPOSED ARCHITECTURE:

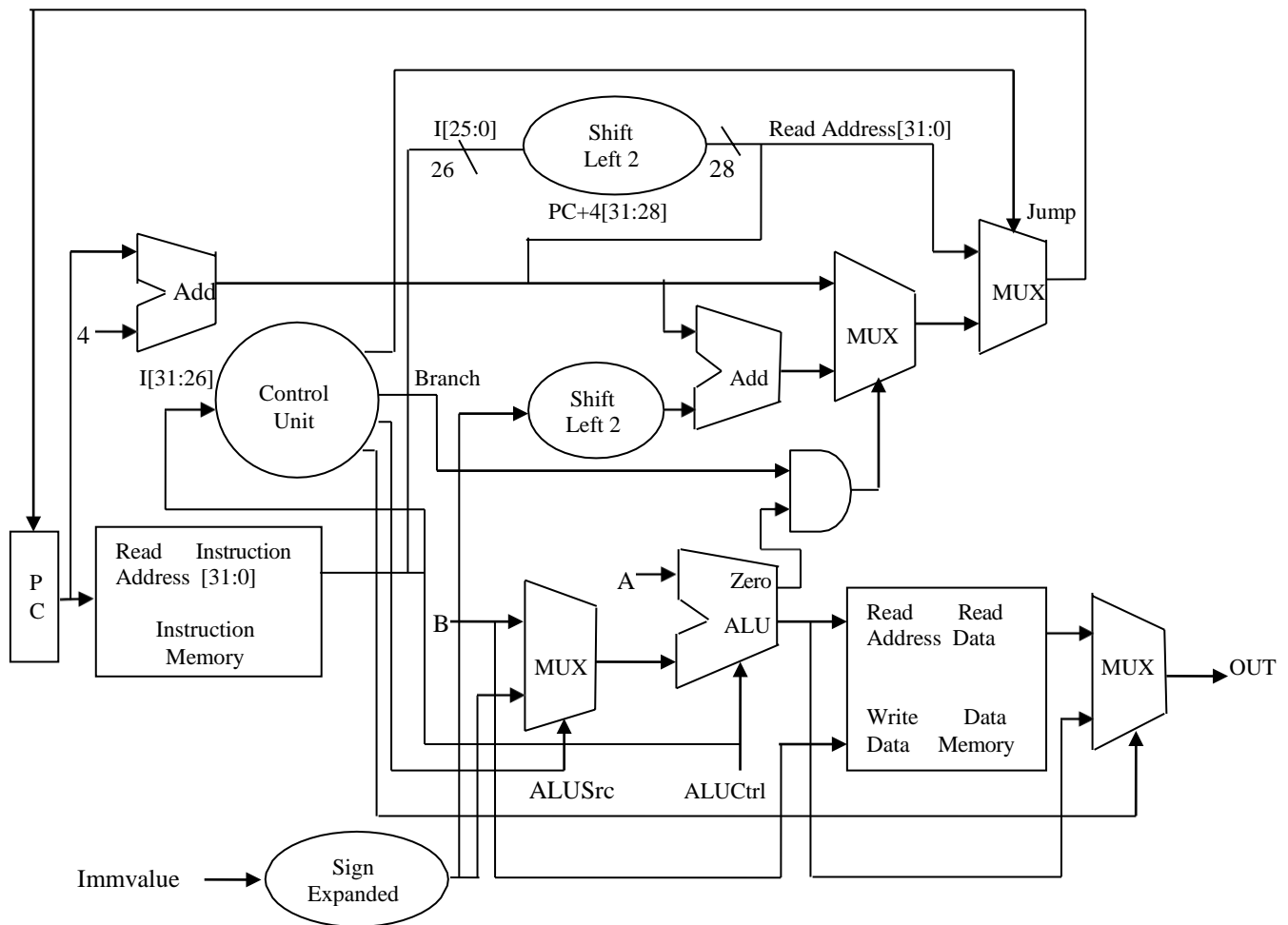


Fig.4. Data Path for the 32-bit MIPS processor design.

Fig.4 demonstrates the data path for the single cycle MIPS processor architecture. 32-bit single cycle MIPS processor architecture comprises of a Program Counter, Memory unit, ALU block, Data Memory, Adders, multiplexers and controlling unit. By utilizing every one of these blocks we can design the 32-bit Single cycle MIPS processor architecture. MIPS processor architecture consists of 5 phases which are Instruction Fetch, Instruction Decode, Execute, Memory Access and Write Back. In single cycle MIPS processor every one of the directions are executed by utilizing single clock cycle. In single cycle MIPS processor on the basis of PC value 32 bit instruction is bring from the instruction memory. In this 32-bit instruction the 6 bits are utilized as an Opcode which decides the kind of activities which ALU can do. 6-bit Opcode is the control signal for the ALU block. ALU unit plays out all the arithmetic and logical operations. In this datapath the data memory is utilized to play out the load and store instructions. In the load and store instruction the ALU unit is utilized to figure the memory address. The main block in the MIPS processor architecture is a controlling unit which gives the whole control flag. The input for the control unit is the 6-bit Opcode, which is gotten from the instruction memory.

After the completion of the one operation we increment the PC value to store the next address for performing the operation. The PC value is incremented by adding 4 to the current value, because every one of the instructions must be in a word limit. To pipeline the 32-bit single cycle MIPS processor we give the clock signal to each block in the single cycle MIPS processor architecture data path. This pipelined MIPS processor will improve the processor speed. This processor architecture data path is designed and simulated by using the Verilog-HDL language. We also get the RTL schematic of the same processor in the Xilinx tool. After simulating the processor we synthesized the MIPS processor in the Spartan 6 FPGA kit. After synthesizing we get the design utilization summary and also get the data about the delay and speed of the processor.

III. RESULTS AND DISCUSSIONS

A. Simulation Results:

The Fig.5, 6, 7 demonstrates the simulation result of a Register Type, Immediate Type, Jump Type instructions and Fig.8.shows the simulation result of an overall 32 bit single cycle MIPS processor design.

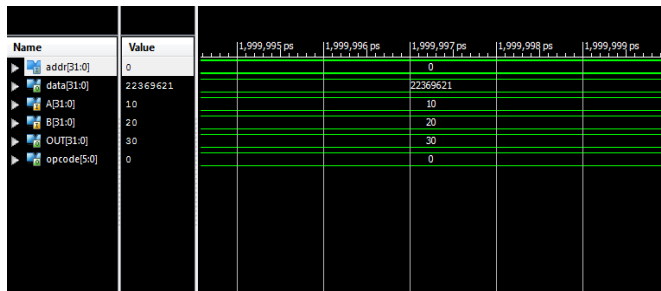


Fig.5. Simulation Result of Register Type Instruction Data Path.

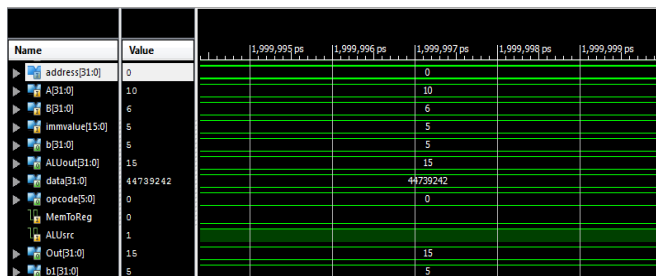


Fig.6. Simulation Result of Immediate Type Instruction Data Path

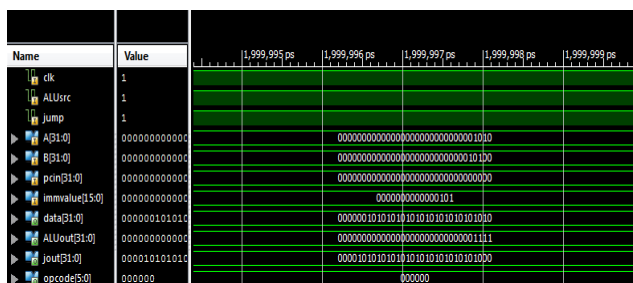


Fig.7. Simulation Result of Jump Type Instruction Data Path

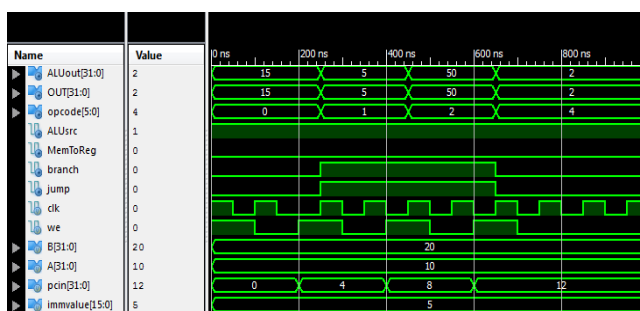


Fig.8. Simulation result of a 32-bit MIPS processor design.

B. Synthesis Result:

B.1 RTL Schematic:

The RTL (Register Transfer Logic) schematic can be viewed after the synthesis is shown below in the black box.

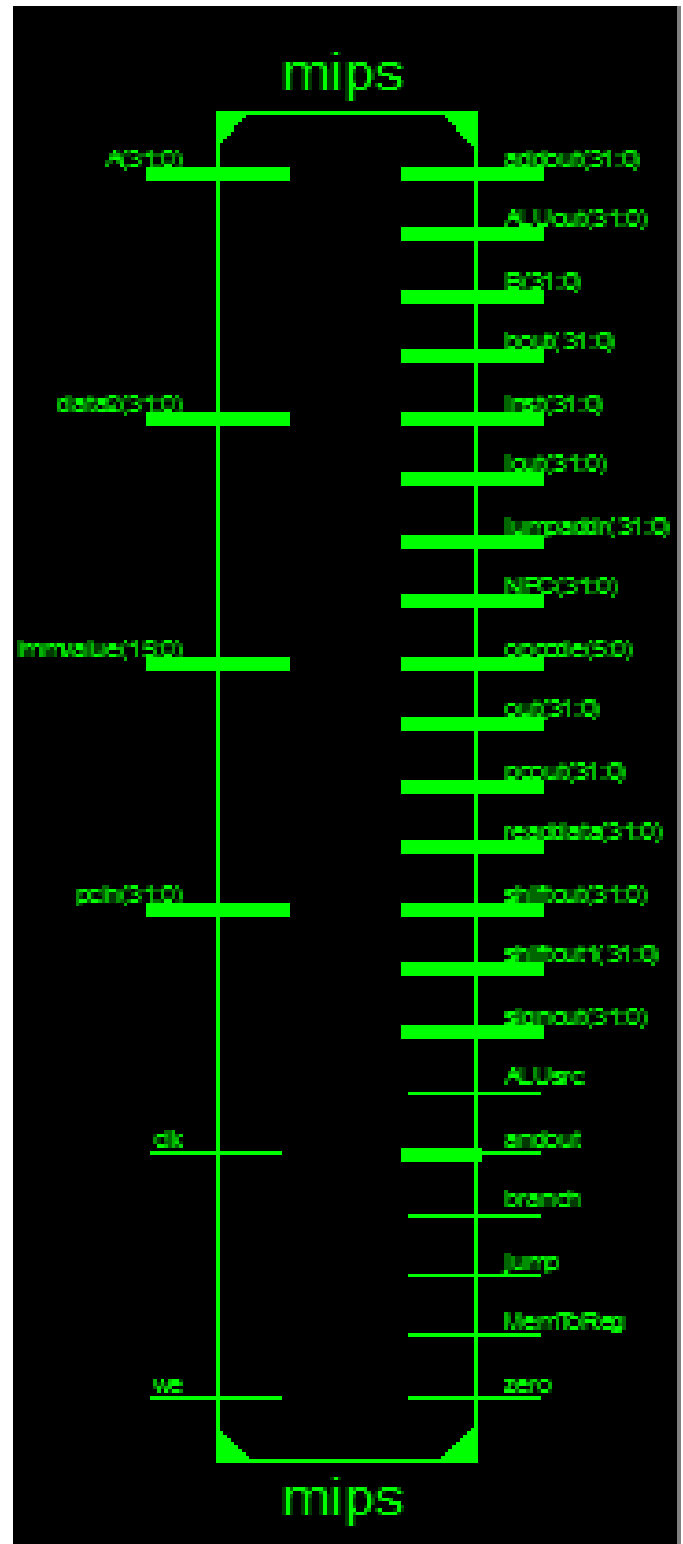


Fig.9. RTL Schematic of a 32-bit MIPS processor architecture.

After the implementation of a 32-bit MIPS processor on a Virtex 5 (Device:xc5vlx110t, Package:ff136, Speed:-1) series FPGA kit we get the following results.

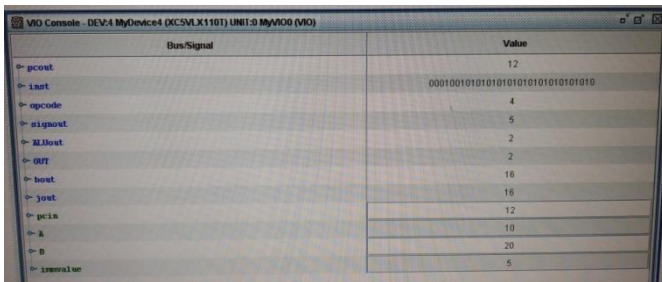


Fig.10. Implementation result of 32-bit MIPS processor architecture using Chip Scope VIO

B.2 Device Utilization Summary:

The 32-bit MIPS processor is executed by utilizing Virtex 5 series FPGA board. The outcomes got are appeared in the underneath table.

Table I. Device Utilization Summary of a 32-bit MIPS processor

Logic Utilization	Used	Available	utilization
Number of Slice Registers	3433	69,120	4%
Number of slice LUTs	1469	69,120	2%
Number of bonded IOBs	1	640	1%

Timing Summary:

Minimum Period: 6.216 ns

(Maximum Frequency = 160.867MHz)

Minimum input arrival time before clock: 0.336 ns

Minimum output required time after clock: 0.471ns

Maximum combinational path delay: No path found

IV. CONCLUSION

In this proposed work it is observed that the FPGA implementation of a 32-bit MIPS processor architecture works with the speed of 290.166MHz. It is observed that improvement in the speed of processor (271.196MHz) compared to [5]. This paper additionally outlines the utilization of Instruction of type Register, Instruction of type Immediate and Instruction of type Jump.

REFERENCES

- [1] Mohit N. Topiwala, N. Sarawathi, "implementation of a 32-bit MIPS Based RISC Processor using Cadence", International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 2014 IEEE.
- [2] Gautam P. Parthasarathy R, Karthi Balasubramaniam, "Low Power pipelined MIPS Processor design", in the proceedings of the 2009, 12th international symposium, 2009 Research in Computer Science (JGRCS), September 2011.
- [3] Neeraj Jain, "VLSI Design and Optimizaed Implementation of a MIPS PISC Processor using XILINX TOOL", International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), December 2012.
- [4] Priyavarat Bhardwaj, Siddharth Murugesan, "Design and simulation of a 32-bit RISC based MIPS Processor using Verilog", International Journal of Research in Engineering and Technology (IJRET), November 2016.
- [5] N. Alkeya, P. Ganesh Kumar, "Design of 32-bit RISC CPU based on MIPS", Journal of Global Research in Computer Science (JGRCS) September 2011.
- [6] Anu Mariam John, Shilpi Varshney, "FPGA Implementation of 32-bit MIPS Processor with CISC Multiplication Operation", *International Journal of Research in Engineering and Technology (IJRET)*, November 2015.
- [7] Akshatha S Patil, B G Shivaleelavathi, "Design and Implementation of Pipelined 8-bit RISC Processor using Verilog HDL on FPGA", *International Journal of Research in Engineering and Technology (IJRET)*, June-2017.
- [8] Kaur Harpreet, Gulati Nitika, "Pipelined MIPS processor architecture with Improved Datapath", *International Journal of Research in Engineering and Application (IJREA)*, January-February 2013.