iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50352

www.ijraset.com

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

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Design and Implementation of RISC MIPS Processor on FPGA

Manjula B B[1], Vidyashree H[2], Kavyashree M G[3], Tulasi V[4], Arpitha R[5]
[1]*Guide*, [2, 3, 4, 5]*Student, Department of Electronics and Communication Engineering, East West Institute of Technology Bengaluru, India*

Abstract: ***In this paper, the design and implementation of RISC Processor is proposed with MIPS (Microprocessor without interlocked pipelined stages) technique. This processor performs 16-bit operations using pipelined technique, to improve the performance. This processor performs arithmetic, logical and data movement operations, more efficiently in terms of delay and power. The processor is composed of five stages namely, instruction fetch, instruction decode, execute, memory access and write back. The proposed 16-bit RISC MIPS processor is designed using Verilog and validated using Modelism, it is synthesized using libraries of AMD 45nm technology in Xilinx tool and implemented on Xilinx Spartan 6 FPGA. The proposed RISC MIPS gives reduced overall delay of 3.565ns and overall power consumption of 0.014W.***
Keywords: FPGA, MIPS, Modelsim, pipelined, RISC, Verilog, Xilinx

I. INTRODUCTION

Computers are now indispensable tools for daily activities, with a growing demand for silicon technology to enhance performance through the use of RISC processors. The reduced cost of integrated circuits makes this a cost-effective solution for flexible systems, as a straightforward design yields excellent results [8]. A RISC (Reduced Instruction Set Computer) Processor is a type of computer processor that places an emphasis on quickly completing a small number of straightforward instructions [7]. MIPS, or a microprocessor without interlocked pipelined stages, is a particular application of the RISC architecture's focus on performance. While RISC Processors have a broader scope and cover a wide range of processors, MIPS is a specific processor architecture that follows RISC principles. A RISC MIPS Processor uses a simplified set of instructions to perform tasks more efficiently [4]. It has fewer instructions compare to traditional processor, But each instructions performs a simpler task. This means that the processor can perform tasks more quickly because it doesn't have to spend as much time decoding complex instructions. Implementing a pipelined RISC MIPS processor using Verilog [5] involves translating the processor design into a physical circuit using Verilog Hardware description language. The Verilog code is synthesized and then programmed onto a Field Programmable Gate Array. Implementation process involves translating the processor stages into Verilog code that describes the digital logic components of the processor. The Verilog code must be optimized for the target hardware, taking into consideration factors such as timing, area, power consumption. The Implemented processor is then tested using test bench to ensure that it behaves correctly and meets the design specifications. The resulting pipelined RISC MIPS processor can be applied to a variety of applications that call for effective and high-performance computing.

II. METHODOLOGY

Figure 1 shows the RISC MIPS Architecture

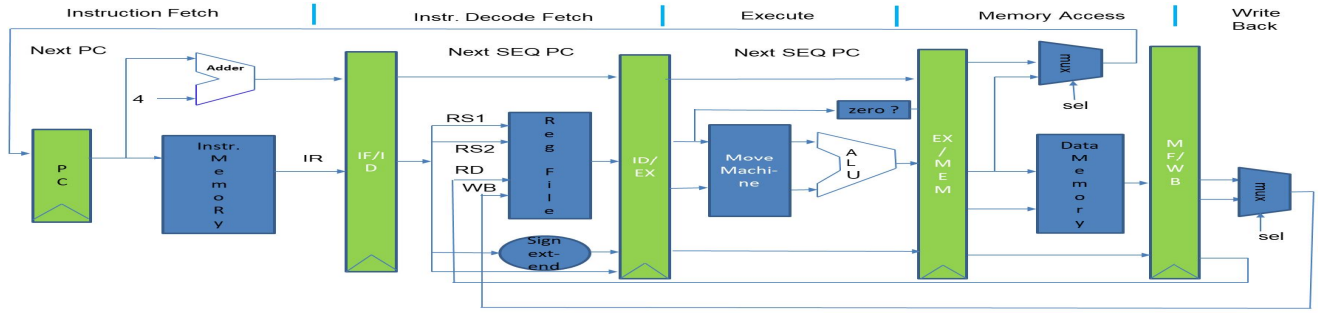


Fig. 1: RISC MIPS Architecture

Pipelining is a technique used in computer processors to improve their performance. In a pipelined processor, the Instructions are broken down into smaller parts than executed simultaneously by different parts of the processor. As a result, the processor's total performance is enhanced as it may work on several instructions at once. RISC (Reduced Instruction Set Computer) processors are designed to use simple and fast instructions, which make pipelining easier and more effective. MIPS is one such type of RISC processor, which uses a five-stage pipeline to execute instruction [4]. Five phases make up this process, which involves fetching the instructions from memory, decoding them, running them, and then storing the results in the memory. Each step is carried out by a different part of the processor, which allows the processor to execute instructions at a fast rate.

A pipelined RISC MIPS processor works like an assembly line in a factory. It breaks down instructions into smaller parts and executes them in series of stages. Each stage of the processor is responsible for carrying out a specific task on the instruction has it moves down the pipeline [5]. The "Instruction Fetch" stage, which is the first one, involves loading the instruction into the processor after it has been retrieved from memory. The "Instruction Decode" stage, which is the second, involves decoding the instruction to determine what action has to be taken. The process is performed on the data during the third stage, "Execute." The fourth stage is referred to as "Memory Access," and during this phase, depending on the operation, either data is loaded from or written to memory. The fifth stage, referred known as "Write Back," is when the operation's outcome is saved in the processor's registers. As each instruction moves to the pipeline, the next instruction can be fetched, decoded, and executed. This boosts the processor's overall efficiency by enabling it to work on several instructions at once.

RISC MIPS processors are designed to execute a fast instruction, which makes them more efficient than other types of processors. The processors can perform different types of operations, such as Arithmetic, logical and data movement operations [2]. Arithmetic operation involves performing mathematical calculations, such as addition, subtraction, multiplication. Logical operations involve evaluating conditions, such as comparing two values. Data movement operations involve moving data between different memory locations or between the processor’s registers. The implementation of RISC MIPS processor has a set of registers, which are small memory locations used for storing data temporarily. The instructions are loaded from memory into the processor, and then executed by the different stages of the pipeline. The processor’s architecture is designed to minimize the number of instructions needed to perform a task, which makes it faster and more efficient than other types of processors. The simplicity of the instruction set and the pipelined architecture make RISC MIPS processors ideal for use in applications that require high performance, such as scientific computing and multimedia processing.

III. DESCRIPTION OF FUNCTIONAL MODULES

In a pipelined RISC MIPS processor, the data unit is responsible for performing memory operations, such as loading and storing data to and from memory. The data unit interfaces with the memory subsystem of the processor to perform these operations. In a pipelined RISC MIPS processor, the instructions and decode unit is responsible for fetching and decoding instruction from memory. The instruction fetch unit fetches instruction from memory and passes them to the decode unit, which decodes the instructions and determines the necessary control signals to execute the instruction. The instructions that the processor needs to carry out are stored in the instruction memory. It functions as a library of instructions that the processor can read and execute. The processor takes the appropriate instruction from the instruction memory when it needs to complete a task and executes it. Data that is either entered or generated by the programmer is stored in the register unit. The move machine helps transfer data from a register file to an arithmetic logic unit (ALU) it acts as a middle model between the two, allowing input register values to be moved efficiently.

An ALU, or Arithmetic and Logic Unit [3], is a digital circuit that perform Arithmetic operations like addition, subtraction, multiplication, increment, decrement and logical operation AND, OR, EX-OR on the 16-bit binary number [1]. Here 16-bitShifter is a digital circuit that can shift the bits of 16-bit binary number by a specified number of positions to the left. The shifter is commonly is used in digital signal processing, microprocessor and other digital circuit. The 16-bit of the input number can be shifted to the right by up to 15 positions with this 16-bit barrel shifter. An 8x1 Multiplexer used here to select an input signal based on the opcode. The signals are the output of the ALU, Shifter, and Barrel Shifter. Only one of these input signals is selected and passed through the multiplexer as output [2]. The memory controller is responsible for managing the access to memory by the processor. It controls the flow of data between the memories, ensure that data is read and written correctly.

IV. RESULTS

The 16-bit RISC MIPS Processor is designed and tested with the simulation tool, to check the functionality of the MIPS design, a test bench was created.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

This test bench included a set of instructions that were read from memory and sent to the instruction memory for testing purposes. The simulation waveform, which showcases the results of the test, is presented in Figure 2.

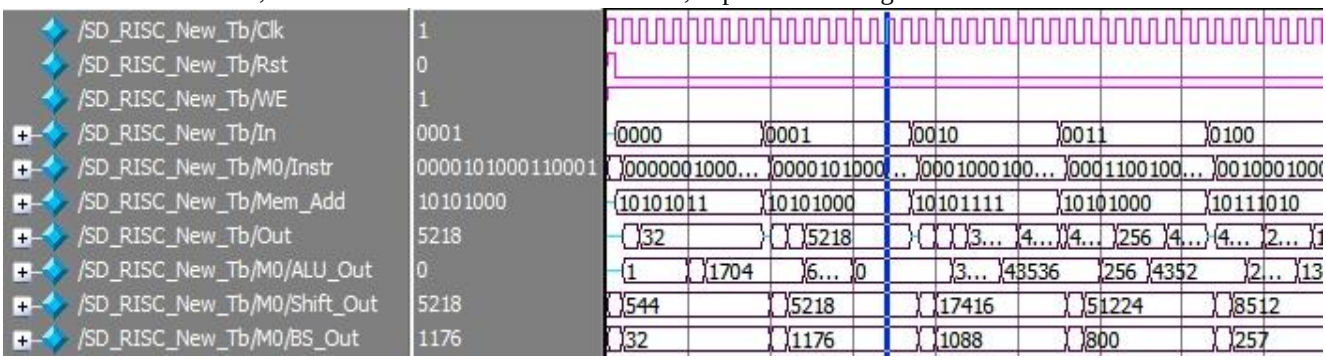


Fig. 2: Simulation result of the processor.

The presented waveform depicts the output of the Arithmetic Logic Unit (ALU), Shifter, and Barrel Shifter, as well as the primary output of the processor. Additionally, the waveform displays the memory address where the primary output is stored. This representation effectively showcases the relevant components and storage location of the processor's output.

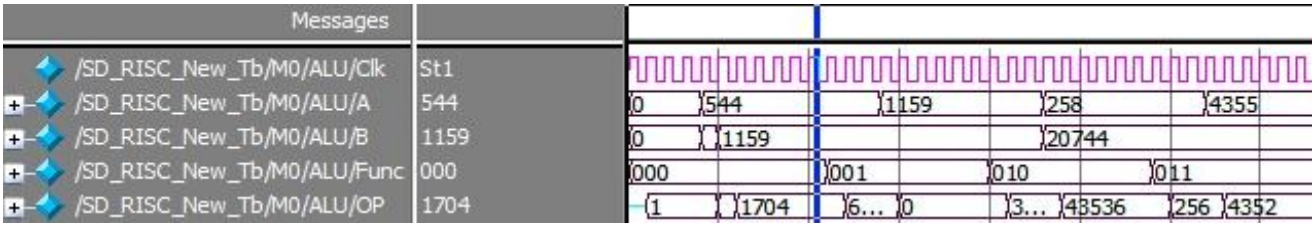


Fig. 3: ALU waveform

Table I: ALU operation

OPCODE	OPERATIONS	EXPRESSION
110	Addition	A+B
001	Subtraction	A-B
010	Multiplication	A*B
000	Increment	A+B+1
111	Decrement	A-B-1
011	AND	A&B
100	OR	A|B
101	XOR	A^B

Fig. 3 displays the waveform of an Arithmetic Logic Unit (ALU), while the Table I presents the various opcodes and their corresponding arithmetic and logic operations performed by the ALU.

```
                                        Gate     Net
Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
----------------------------------------  ------------
 IBUF:I->O            114   1.222   1.913  Rst_IBUF (Rst_IBUF)
 FDR:R                      0.430          M0/Data_Unit/Out_0
----------------------------------------
Total                      3.565ns (1.652ns logic, 1.913ns route)
                                   (46.3% logic, 53.7% route)
```

Fig. 4 Delay report

Fig. 4 displays the processor's complete delay report.

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	--	--
Logic	0.000	126	5720	2
Signals	0.000	173	--	--
DSPs	0.000	1	16	6
IOs	0.000	14	102	14
Leakage	0.014			
Total	0.014			

Fig. 5: Power Estimation

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice Registers	107	11,440	1%
Number of Slice LUTs	126	5720	2%
Number of fully used LUT-FF Pairs	61	138	44%
Number of bonded IOBs	14	102	13%
Number of Utilized Slices	41	1,430	2%

Fig. 6 Area Estimation

The power report and Area report are shown in figure 5and 6. The proposed system has an overall delay of 3.565 ns and overall power consumption off 0.014 W.

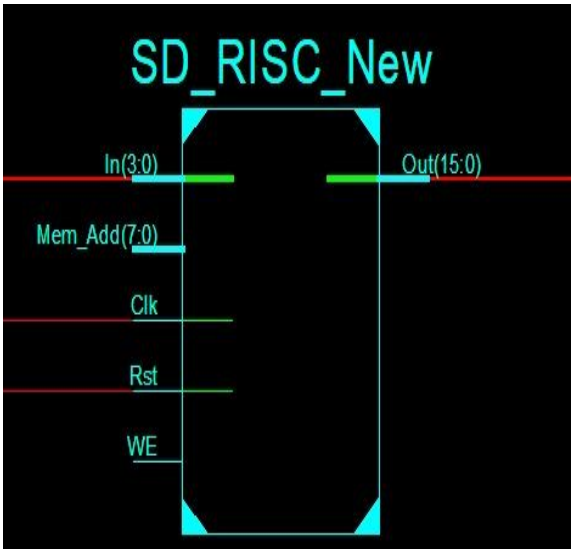


Fig.7: Top Module of processor

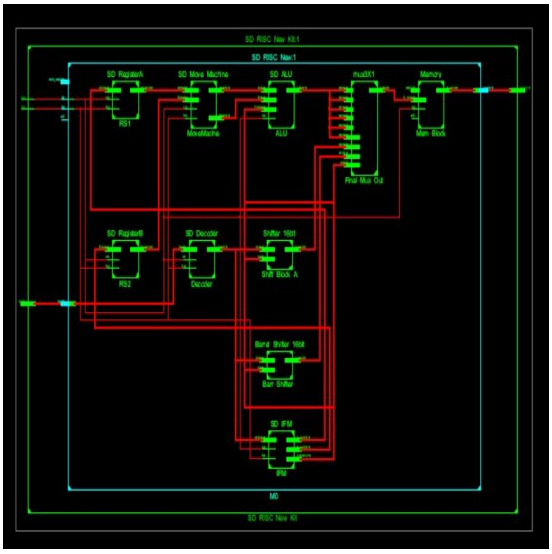


Fig. 8: RTL Schematic of RISC MIPS Processor

The RTL schematic for the RISC MIPS processor shows how the processor's components are connected and how data flows between them. It consists of modules like registers, arithmetic and logical units, data unit, and memory. The schematic is useful for designing, simulating, and verifying the processor's performance. By analyzing the schematic, designers can identify potential issues and optimizations for improving the processor's functionality.

V. FPGA IMPLEMENTATION OF RISC MIPS PROCESSOR

The RISC MIPS Processor's functionality is confirmed by simulation and FPGA prototyping. After finishing valid functional simulation and post-route simulation, FPGA prototyping can be performed. The RISC MIPS processor design is intended for implementation on Xilinx Spartan 6 FPGA.

Figure 9 shows the RISC MIPS processor's implementation on the Spartan 6 FPGA board, where the FPGA prototyping results indicate arithmetic operations, shifter, and barrel shifter functionality.



Fig. 9 : FPGA prototyping results of RISC MIPS processor

Table II: Comparison Table

Parameters	Existing System	Proposed System
Power (W)	0.14W	0.014W
Delay (ns)	15.661ns	3.565ns

The proposed system consumes less power and has lower delay compared to the existing system, as shown in the comparison table II.

VI. CONCLUSION

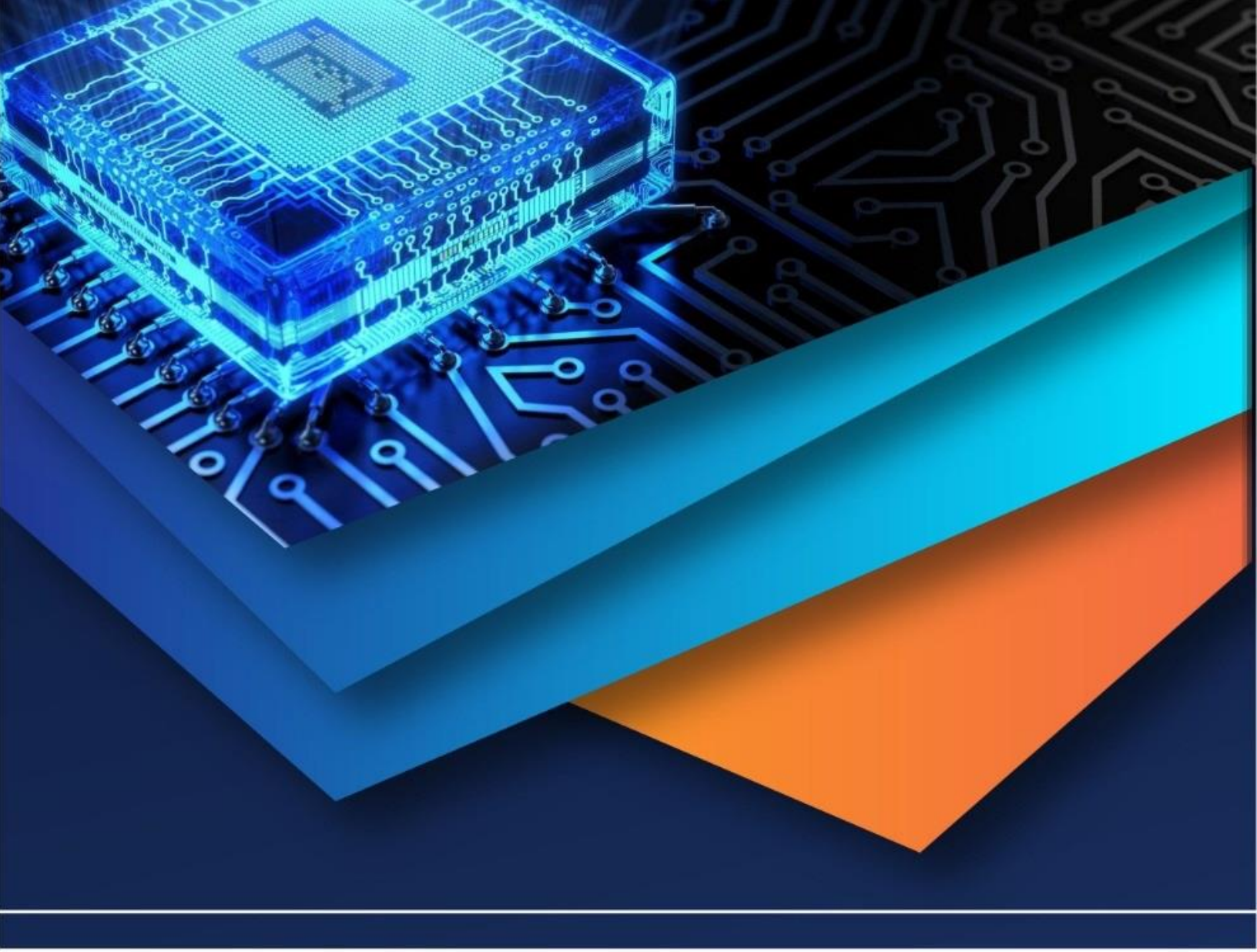
A 16-bit RISC MIPS Processor was efficiently designed using pipeline technique, which is optimized, and synthesized with a delay of 3.565ns and power consumption of 0.014W, surpassing the existing design. The design was implemented on an FPGA Spartan 6 Kit, providing an efficient solution for modern computing requirements.

VII. FUTURE SCOPE

There is potential for further modification of the RISC MIPS processor to handle to higher bit counts, such as 32-bit and 64-bit as well as for memory management. This could enhance the processing power and capabilities of the processor, making it an even more attractive solution for modern computing needs. Overall, the future scope for the RISC MIPS processor is promising, with potential for continued innovation and improvement in this technology.

REFERENCES

[1] Supraj Gaonkar, Anitha M "Design of 16-bit RISC Processor" Sir M Visvesvaraya Institute of Technology Bangalore, Karnataka, India IJSRPA 2020.
[2] Rashmi, Samanth, Ashwini, Amin, Subramanya, "Design and Implementation of 32 Functional unit for RISC Architecture applications", 2020.
[3] Archana H R, Sanjana T, Bhavana H T, Sunil S V "System Verification and Analysis of ALU for RISC Processor" B.M.S. College of Engineering Bangalore, India ICACCS 2021.
[4] Priyanka Trivedi, Rajan Prasad Tripathi "Design & Analysis of 16 bit RISC Processor Using low Power Pipelining" Galgotias University, Amity University ICCCA 2015.
[5] Jikku Jeemon "Pipelined 8-bit RISC Processor Design using Verilog HDL on FPGA", in IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016, India.
[6] Shaik Afroz , M.Sumalatha "Implementation of RISC-Based Architecture for Low power applications" Chirala Engineering College Chirala, A.P, India 2013.
[7] Nyamatulla M Patel,Pramod V Patil, Gazalatabassum M Alan,Priyanka B "Design of Pipelined RISC MIPS processor (16-bit) using VLSI technology" Hirasugar Institute of Technology Belagavi, Karnataka, India JETIR 2018.
[8] S. Suresh and R. Ganesh "FPGA Implementation of MIPS RISC Processor" CVR College of Engineering Hyderabad, India IJERT 2014.
[9] R. Uma ,"Design and performance Analysis of 8- bit RISC Processor using XILINX Tool", International journal of Engineering research and Applications, Vol. 2, no. 2, March –April 2012.
[10] Pallavi P T, Padma R, Darshan N, Dileep K V, "Implemention of 8 bit single cycle RISC processor using Cadance", 2018.



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)