iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** IV **Month of publication:** April 2026

DOI: https://doi.org/10.22214/ijraset.2026.81150

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 14 Issue IV Apr 2026- Available at www.ijraset.com

Design and Implementation of RISC-V 32M Using Verilog HDL

Borra Kailash, Chevula Praveen Kumar, Kalla Sharon Olivia, Tirnathi Thanaya Sai Venkata Suryakanth, Prof. P. V. Sridevi
Department of Electronics and Communication, Andhra University College Of Engineering, Andhra Pradesh, India

Abstract: ***RISC-V has significant demand due to its efficiency, scalability, and open-source instruction set architecture. Its simple and modular design makes it highly suitable for both academic and industrial applications. This paper presents the design and implementation of an RV32M processor using a single-cycle architecture, where each instruction is executed within one clock cycle. The proposed design includes key modules such as instruction memory, control unit, register file, and arithmetic logic unit (ALU). This pro cessor is efficient to perform arithmetic, logical, and branch operations, following a load/store architecture. This design is implemented using Verilog HDL and verified through simulation, which demonstrates the functionality and efficient execution of instructions. This provides an understanding of RISC-V processor design and foundation for pipelined implementation.***

Keywords— RISC-V, RV32M, Single-Cycle Processor, Instruction Set Architecture, ALU, M-Extension, Datapath Design.

I. INTRODUCTION

Processor design plays a crucial role in modern computing systems, influencing performance, complexity, power consumption and scalability. Traditional processors such as complex instruction set computer (CISC) execute multi-step instruction into a single instruction leads to slower execution, high power consumption, pipelining difficulties and design complexities. These limitation driven adoption to Reduced Instruction Set Computer (RISC) architecture for its simplicity and faster execution.

RISC-V is an open standard Instruction Set Architecture (ISA) which provides flexibility and extensibility, making it ideal for academic and industrial purposes. ARM and MIPS architecture follows RISC principles. X86 widely used CISC processors which have been developed over the years. These architectures are licensed and complex. Whereas RISC-V stands out for its open-source architecture and simplified instruction set. Compared to CISC, RISC processors offer better execution speed, memory access, pipelining, and optimization capabilities. It has more general purpose registers and few addressing modes whose complexity is simple and efficient.

Processor design explored in both single-cycle and pipelined architecture. The single-cycle processor executes each instruction in a single clock cycle. Single-cycle processors are simpler to design and understand, which is suitable for academic purposes. This paper focuses on the implementation of the RISC-V 32M processor using a single-cycle architecture. The RV32M includes fundamental arithmetic, logical, load/store, and branch operations. The primary objective is to design a modular and functional processor using single-cycle architecture

II. SYSTEM ARCHITECTURE OF RISC-V 32M PROCESSOR

The architecture of RV32M is based on a single-cycle architecture, where each instruction is executed at within a clock cycle. This architecture follows a top-down methodology, where the processor architecture is first designed, and then broken down into smaller units. This processor consists of separate instruction and data memory units. The processor is designed in modular approach which improves scalability, reliability and ease of implementation.

A. Datapath description

The datapath consists of functional units which are responsible for instruction execution, including the Program Counter (PC), Instruction Memory, Register File, Arithmetic Logic Unit (ALU), Immediate Generator, Data Memory, and multiplexers.

The Program Counter provides the address of the instruction. The fetched instruction is decoded, and operands are read from the register file. The ALU performs the required computation based on the instruction. For memory-related instructions, the ALU output is used as an address to access the data memory.

Multiplexers are used to select appropriate inputs and outputs at different stages of the datapath, enabling flexible data routing for various instruction formats.

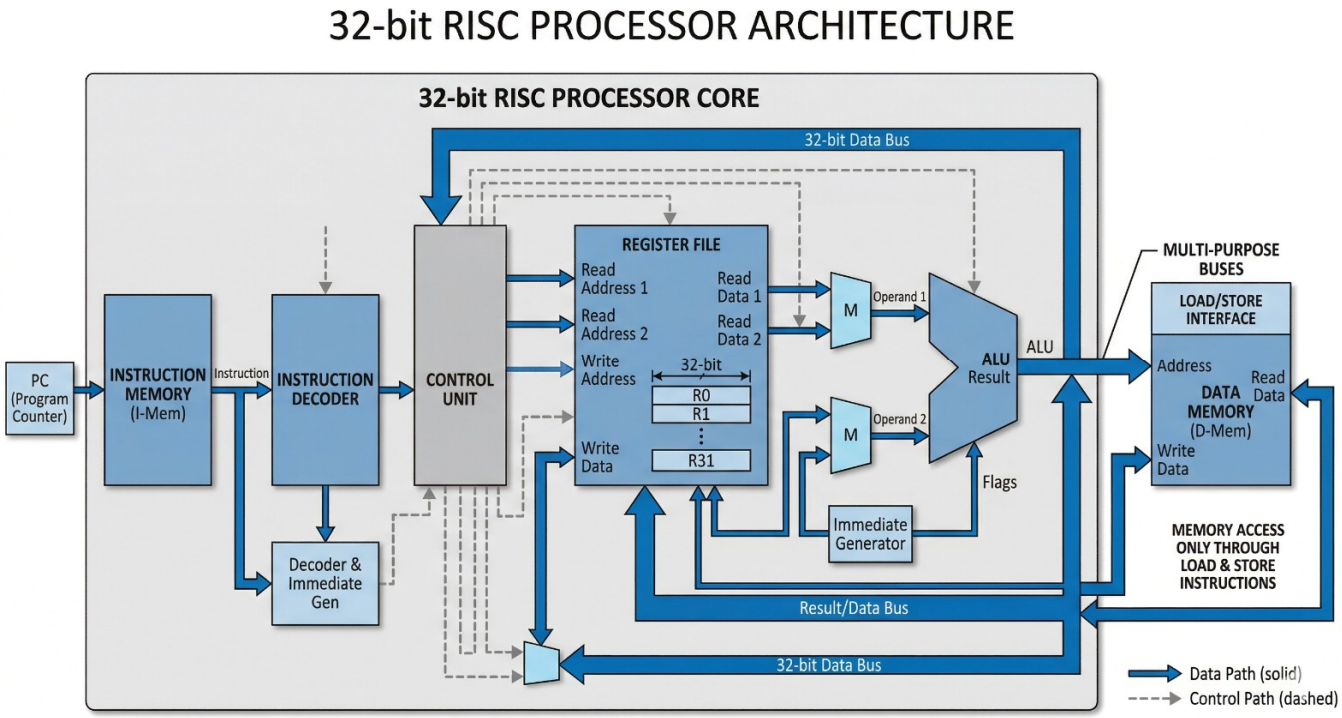


Fig. 1 RV32M Single-Cycle Processor Architecture.

B. Instruction decoder and Control

The instruction decoder extracts opcode, funct3, funct7 fields from the instruction. The control unit generates the required control signals. These signals co-ordinates the operation of various modules within the processor.

C. Register file

The register file consists of 32 general purpose registers with 32 bits wide. It consists of two read ports (rs1,rs2), and one write port (rd).

D. Arithmetic Logic Unit (ALU)

The ALU performs arithmetic, logical and shifting operations, taking input operands from register file and immediate generator. The ALU operates based on control signals provided by the control unit.

TABLE I
ALU CONTROL LINES

Instruction Opcode	ALUOp	Operation	Funct7	Funct3	Desired ALU action	ALU control input
Id	00	Load doubleword	xxxxxxx	xxx	Add	0000
Sd	00	Store doubleword	xxxxxxx	xxx	Add	0000
beq	01	Branch if equal	xxxxxxx	xxx	Subtract	0001
R-type	10	Add	0000000	000	Add	0000
R-type	10	Sub	0100000	000	Subtract	0001
R-type	10	Mul	0000001	000	Multiplication	0010
R-type	10	Div	0000001	100	Division	0100

R-type	10	Rem	0000001	110	Division	0101
R-type	10	And	0000000	111	AND	0110
R-type	10	Or	0000000	110	OR	0111
R-type	10	Xor	0000000	100	XOR	1000

E. Data memory

The Data memory is used for storing and retrieving data, which can be accessed during load/store instructions. The ALU provides memory access, while the operations are performed based on control signals.

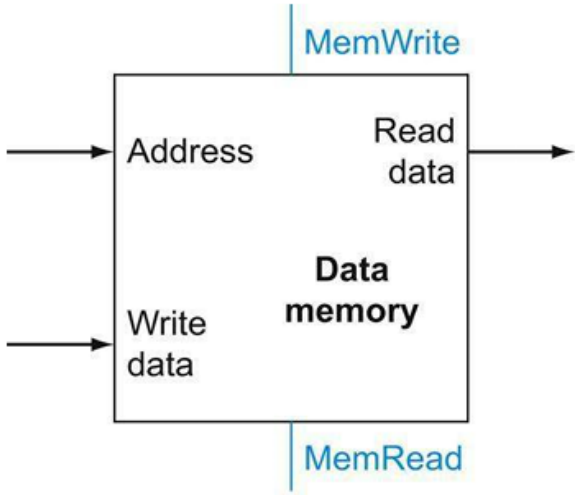
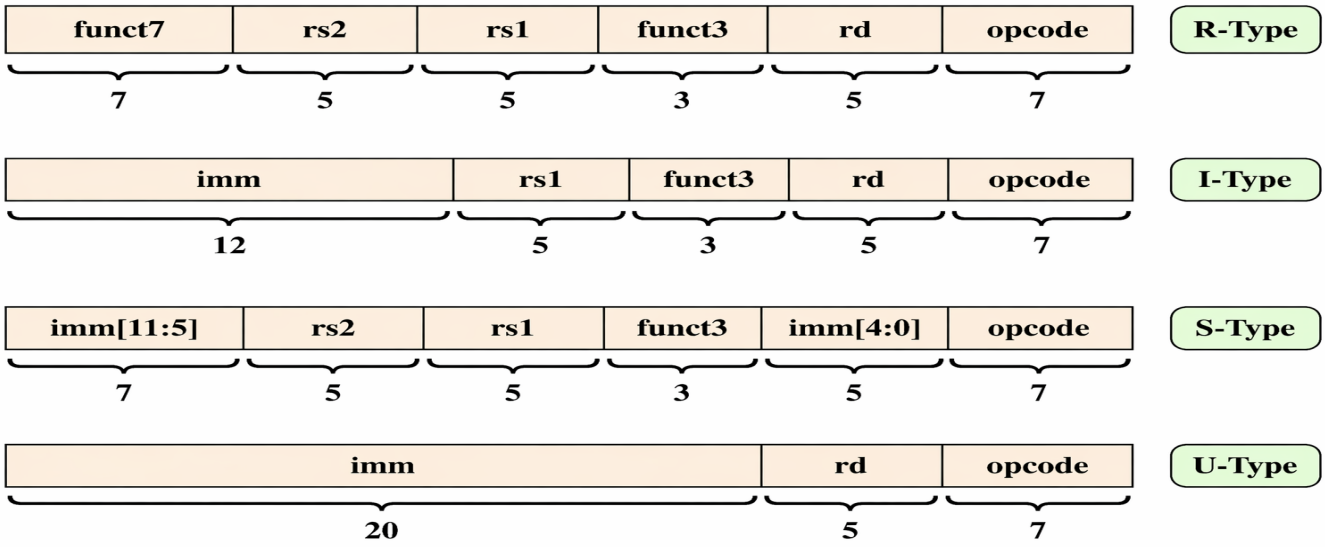


Fig. 3 Data Memory Unit

III. DESIGN METHODOLOGY

The overall design of a RV32M processor is divided into multiple functional block, each responsible for individual operation. The PC module is a 32-bit register that updates at every clock cycle. It increments either by 4 or update the address by branch/jump operations.



F

Fig. 4 RISC-V Instruction Format.

The instruction memory is developed as ROM that stores the program instructions. It provides instructions corresponding to PC value. The Control unit decodes instructions and generate appropriate signals. These signals co-ordinates the operation of other modules in the datapath. The Register file consists of 32 general purpose registers, of 32-bit wide. The source registers fetch operands, whereas the destination register is updated by the output. The Register x0 is hardwired to zero. The Arithmetic Logic unit performs arithmetic, logical and shifting operations. These operations are determined by control unit signals, along with funct3, funct7 fields. The Immediate generator extracts immediate values from the instructions, which are used for immediate operations.

TABLE 2
CONTROL UNIT SIGNALS

Signal name	Effect when desserted	Effect when asserted
RegWrite	None.	The register on the write register input is written with the value on the write data input.
ALUSrc	The second ALU operand comes from the second register file output.	The second ALU operand is the sign-extended 12-bit of the instruction.
PCSrc	The PC is replaced by the output of the adder that computes the value of PC+4.	The PC is replaced by the output of the adder that computes the branch target.
MemRead	None.	Data memory contents designated by the address input are put on the Read data output.
MemWrite	None.	Data memory contents designated by the address input are replaced by the value on the write data input.
MemtoReg	The value fed to the register write data input comes from the ALU.	The value fed to the register write data input comes from the data memory.

The address of the data memory is derived from ALU. It performs read and write operations based on control signals. Finally, all these modules are integrated in a top-level module, making proper data flow across the processor.

IV. SIMULATION AND RESULTS

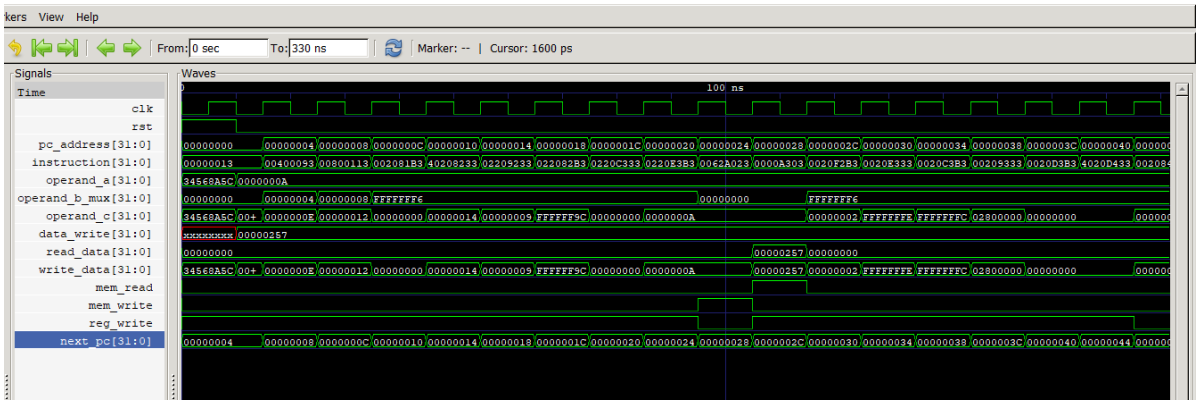


Fig. 6 Simulated waveform analysis of RV32M processor.

The RV32M processor is implemented using Verilog HDL and verified through simulation. The waveform represents correct execution, where instructions are fetched and decoded. The ALU performs arithmetic and logical operations accurately, meanwhile memory operations are validated through control signals. The simulation results verify the functional correctness of the RV32M single-cycled processor.

V. CONCLUSIONS

This paper presents the design and implementation of RISC-V 32M single-cycle processor using verilog HDL. The architecture of RV32M integrates the modules such as Control unit, Register file, ALU and memory units which supports M-extension.

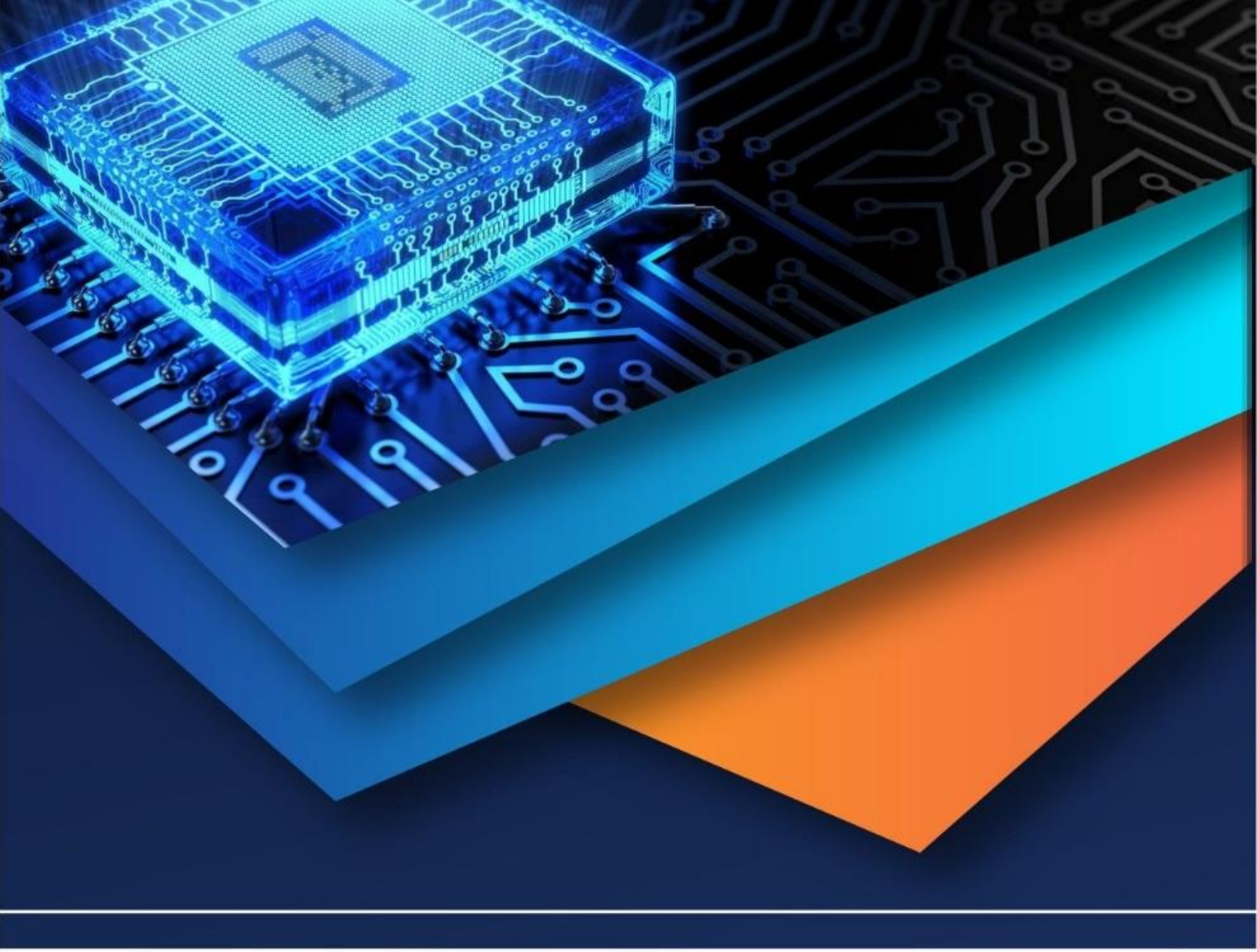
This design is a simple modular approach, makes suitable for academic purposes and foundation for pipelined architecture and performance optimization.

VI. ACKNOWLEDGMENT

We would like to express our gratitude to Prof. P. V. Sridevi for her valuable guidance, Continuous support and encouragement throughout the development of this project. Her suggestions and technical insights greatly contributed to the successful completion of this project. We also extend our thanks to the faculty members of the Department of Electronics and Communication, for their support and assistance during the course of this project.

REFERENCES

[1] D. A. Patterson and J. L. Hennessy, Computer Organization and Design: RISC-V Edition, 1st ed. Burlington, MA, USA: Morgan Kaufmann, 2017.
[2] S. R. Sarangi, Computer Architecture. New Delhi, India: McGraw Hill Education, 2015.
[3] A. Waterman and K. Asanović, The RISC-V Instruction Set Manual, Volume I: User-Level ISA, Version 2.1, 2019.
[4] T. G. Galani, R. Saini, and R. D. Daruwala, "Design and Implementation of 32-bit RISC Processor using Xilinx," International Journal of Emerging Trends in Electrical and Electronics, vol. 5, no. 1, pp. –, July 2013.
[5] S. Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis, 2nd ed. Upper Saddle River, NJ, USA: Pearson, 2003.
[6] M. M. Mano, Digital Logic and Computer Design. Upper Saddle River, NJ, USA: Prentice Hall, 1979.
[7] The RISC-V Foundation, The RISC-V Instruction Set Manual, Volume I: User-Level ISA, 2019.
[8] P. Chu, FPGA Prototyping by Verilog Examples. Hoboken, NJ, USA: Wiley, 2008.
[9] H. Kaul et al., "A 320 mW 1.5 GHz 32-bit RISC-V processor in 22 nm CMOS," in Proc. IEEE Int. Solid-State Circuits Conf. (ISSCC), 2017, pp. 1–3.
[10] Y. Lee et al., "A 45 nm 1.3 GHz RISC-V processor with integrated multiply and divide units," IEEE J. Solid-State Circuits, vol. 51, no. 1, pp. 1–10, Jan. 2016.
[11] J. L. Hennessy, "The rise of RISC-V," IEEE Micro, vol. 38, no. 2, pp. 5–7, Mar.–Apr. 2018.
[12] M. Gautschi et al., "Near-threshold RISC-V core with DSP extensions for scalable IoT endpoint devices," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 10, pp. 2700–2713, Oct. 2017.
[13] S. R. Vemuri and R. Tessier, "Energy-efficient RISC-V processor design on FPGA," in Proc. IEEE Int. Conf. Reconfigurable Comput. FPGA (ReConFig),2018.
[14] B. Zimmer et al., "A 0.6–1.3 GHz RISC-V processor with vector extensions in 28 nm CMOS," in Proc. IEEE Symp. VLSI Circuits, 2019.
[15] M.Asanović et al., "The Rocket Chip generator," EECS Dept., Univ. of California, Berkeley, CA, USA, Tech. Rep., 2016.
[16] A. Waterman, "Design of the RISC-V instruction set architecture," Ph.D. dissertation, Univ. of California, Berkeley, CA, USA, 2016.
[17] S. Mittal, "A survey of RISC-V based processors for embedded systems," J. Syst. Archit., vol. 110, 2020.
[18] Xilinx Inc., Vivado Design Suite User Guide, 2020.
[19] RISC-V Foundation, "RISC-V specifications." [Online]. Available: https://riscv.org
[20] Mentor Graphics, ModelSim User Manual, 2019.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY