Journal Publication of International Research for Engineering and Management (JOIREM)
Volume: 03 Issue: 12 | Dec-2025
ISSN (O) 3107-6696

Designing of 32 Bit-RISC Processor

Prof. Neeta M[1], Miss.Lavanya N Kurahatti[2], Miss.Keerti V Navalagunda[3], Miss.Kavya B Ramasali[4]

[1,2,3,4] *Department of Electronics and Communication Engineering, Tontadarya College Of Engineering , Gadag, Karnataka, India*

---***---

Abstract - The increasing demand for high-speed and low-power computing systems has made Reduced Instruction Set Computing (RISC) architecture one of the most preferred processor design approaches. This paper presents the design and implementation of a 32-bit RISC processor using Verilog Hardware Description Language. The proposed processor follows a simple and efficient architecture with a uniform instruction format and a pipelined datapath to improve performance. The design includes essential components such as the instruction memory, register file, arithmetic logic unit, control unit, and data memory. A five-stage pipelining technique is adopted to enhance the instruction throughput. Functional verification is carried out through simulation, and the processor is synthesized for FPGA implementation. The results demonstrate that the designed processor achieves reliable performance with reduced hardware complexity, making it suitable for embedded and educational applications.

Keywords - RISC processor, Verilog HDL, instruction set, processor architecture, digital design.

I. Introduction

Processors form the core of all modern digital systems, enabling computation in various applications such as embedded systems, mobile devices, computers, and communication systems. The continuous demand for faster execution, reduced power consumption, and efficient hardware utilization has led to the development of optimized processor architectures. One such widely adopted architecture is the Reduced Instruction Set Computing (RISC) architecture.

RISC architecture is based on the principle of executing a small set of simple instructions at a very high speed. Unlike Complex Instruction Set Computing (CISC), which uses complex instruction formats, RISC processors use fixed-length instructions and a load-store architecture that simplifies hardware design and improves performance. Due to these advantages, RISC processors are commonly used in high-performance and low-power applications.

A 32-bit RISC processor can process 32-bit data and addresses, allowing it to support a larger memory space and higher precision compared to lower bit processors. In this paper, a 32-bit RISC processor is designed using Verilog HDL. The design focuses on simplicity, modularity, and efficiency. The processor includes a pipelined architecture to improve throughput and overall system performance.

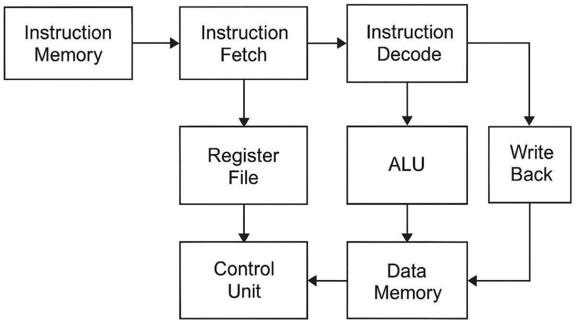

Fig. 1. Block Diagram of 32-BIT RISC Processor

II. RELATED WORK

The concept of RISC architecture was introduced in the early 1980s through research projects conducted at leading universities and organizations. One of the earliest successful RISC-based processors was the MIPS architecture, which demonstrated how simplified instruction sets and pipelining could significantly improve performance. MIPS processors became a popular model for both academic and commercial processor designs.

ARM processors later adopted RISC principles and became widely used in mobile and embedded systems due to their low power consumption and high efficiency. Several studies have focused on optimizing ARM-based processors for power, performance, and area.

More recently, the open-source RISC-V architecture has gained attention for academic research and industrial development. RISC-V allows designers to develop custom processors without license restrictions. Many researchers have proposed different 32-bit RISC cores based on RISC-V, focusing on pipelining, hazard handling, and low-power optimization.

Most existing works highlight the importance of modular design, efficient datapath organization, and pipeline implementation. The proposed work in this paper follows similar principles while maintaining a simplified and educationally focused 32-bit RISC processor design.

III. ARCHITECTURE of 32-BIT RISC PROCESSOR

The architecture of the proposed 32-bit RISC processor is designed to achieve high performance with minimal hardware complexity. The processor is based on a load-store architecture and supports a fixed-length 32-bit instruction format. The major components of the processor architecture are described below.

1. Instruction Set Architecture (ISA)
The instruction set consists of basic arithmetic, logical, memory access, and control instructions. These include ADD, SUB, AND, OR, LOAD, STORE, JUMP, and BRANCH operations. The instructions are classified into R-type, I-type, and J-type formats for easier decoding and execution.

2. Program Counter (PC)
The Program Counter holds the address of the next instruction to be executed. It is updated sequentially or modified during branch and jump operations.

3. Instruction Memory
Instruction memory stores the program instructions. During each clock cycle, the instruction corresponding to the current PC value is fetched.

4. Register File
The register file contains a set of general-purpose registers used for storing operands and intermediate results. It supports simultaneous read and write operations.

5. Arithmetic Logic Unit (ALU)
The ALU performs arithmetic and logical operations as determined by the control unit. It executes operations such as addition, subtraction, AND, OR, and comparison.

6. Data Memory
Data memory is used for load and store operations. It allows read and write access to store temporary data.

7. Control Unit
The control unit generates the necessary control signals for each instruction to manage data flow and execution.

8. Pipelining
The processor uses a five-stage pipeline consisting of Instruction Fetch, Instruction Decode, Execute, Memory Access, and Write Back stages. This enables parallel execution of multiple instructions.

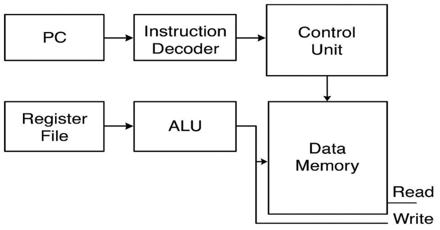

Fig. 2. Architecture Of 32-bit RISC Processor

IV. DESIGN METHODOLOGY

The design of the 32-bit RISC processor is carried out using a systematic and modular approach. The main steps involved in the design process are as follows:

1. Specification and Requirement Analysis
The processor requirements are defined based on the desired instruction set, data width, performance, and application purpose. A 32-bit instruction format and register-based architecture are selected.

2. ISA Design
Instruction formats, opcodes, addressing modes, and register structures are defined. This ensures uniform instruction decoding and efficient execution.

3. Datapath Design
The datapath consists of the register file, ALU, multiplexers, Program Counter, and memory interfaces. Each unit is designed separately and later integrated.

4. Control Unit Design
A hardwired control unit is developed using combinational logic to generate appropriate control signals based on instruction opcodes.

5. Pipelined Implementation
To improve throughput, the processor is divided into five pipeline stages. Pipeline registers are inserted between each stage to synchronize data transfer.

6. Verilog HDL Implementation

Each functional block of the processor is coded using Verilog. A top-level module is created to integrate all submodules.

7. Simulation and Verification
Testbenches are developed to verify the correct functioning of each module. Simulation is performed using tools such as EDA Playground.

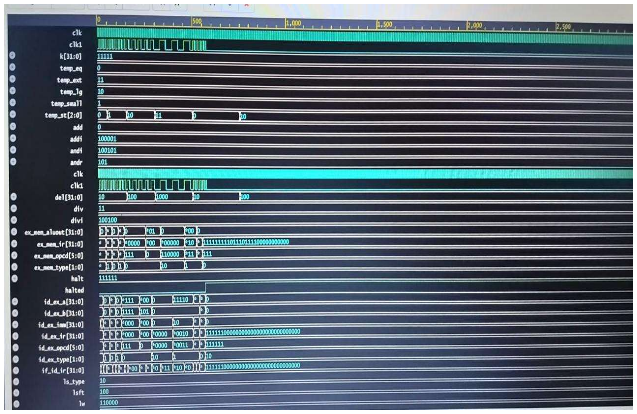

Fig. 3. simulation of waveform of 32-bit RISC Processor

V. CONCLUSION AND FUTURE WORK

Conclusion

This paper presented the design and implementation of a 32-bit RISC processor based on fundamental RISC principles. The processor architecture was designed with a simplified instruction set, modular datapath, and pipelined execution to achieve higher instruction throughput. The design was implemented using Verilog HDL and verified through simulation and synthesis. The results indicate that the proposed processor operates correctly and efficiently with reduced hardware complexity. The modular structure of the design makes it suitable for educational use and embedded system applications.

Future Work

Although the current design meets the basic functional requirements, several enhancements can be introduced in future work:

- Implementation of data forwarding and hazard detection units
- Addition of cache memory for faster memory access
- Integration of power optimization techniques
- Support for interrupt handling
- Extension toward RISC-V compatible architecture

References

[1] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 5th ed., Morgan Kaufmann, 2011.

[2] D. A. Patterson and J. L. Hennessy, Computer Organization and Design, 4th ed., Elsevier, 2012.

[3] M. M. Mano and M. D. Ciletti, Digital Design with an Introduction to Verilog HDL, 5th ed., Pearson, 2013.

[4] S. Furber, ARM System-on-Chip Architecture, Addison-Wesley, 2000.

[5] Waterman and K. Asanovic, The RISC-V Instruction Set Manual, UC Berkeley, 2019.

[6] Xilinx Inc., Vivado Design Suite User Guide: Synthesis, 2020.

[7] Mentor Graphics, ModelSim Simulation User Manual, 2019.