

Design of High-Performance 32-bit Embedded Processor

Dr. Y. Rama Krishna¹, M Durga Viswanadhan²

*¹Professor, Dept. of Electronics and Communication Engineering,
Seshadri Rao Gudlavaluru Engineering College, Andhra Pradesh,
India*

*²Students, Dept. of Electronics and Communication Engineering, Seshadri Rao
Gudlavaluru Engineering college, Andhra Pradesh, India*

Abstract:

The increasing call for for excessive-speed and lowpower computing structures has made Reduced Instruction Set Computing (RISC) architecture one of the most desired processor layout procedures. This paper affords the layout and implementation of a 32-bit RISC processor the use of Verilog ardware Description Language. The proposed processor follows a simple and efficient structure with a uniform practise layout and a pipelined datapath to improve overall performance. The design consists of vital additives including the education memory, sign up document, mathematics common sense unit, manage unit, and information memory. A 5-degree pipelining method is adopted to beautify the guidance throughput. Functional verification is carried out via simulation, and the processor is synthesized for FPGA implementation. The results show that the designed processor achieves reliable overall performance with decreased hardware complexity, making it suitable for embedded and educational packages.

I. INTRODUCTION

Processors shape the core of all contemporary digital structures, allowing computation in diverse applications along with embedded systems, cellular gadgets, computers, and communique structures. The continuous call for for faster execution, reduced electricity consumption, and efficient hardware utilization has led to the improvement of optimized processor architectures. One such widely adopted structure is the Reduced Instruction Set Computing (RISC) structure. RISC architecture is based totally at the principle of executing a small set of simple instructions at a completely high pace. Unlike Complex

Instruction Set Computing (CISC), which uses complicated instruction codes, RISC processors use constant-duration commands and a load-store architecture that simplifies hardware layout and improves overall performance. Due to those advantages, RISC processors are normally used in high-performance and low-energy programs. A 32-bit RISC processor can handle 32-bit data and addresses, permitting it to address a larger memory space and higher precision as compared to 16-bit processors. In this paper, a 32-bit RISC processor is designed using the usage of Verilog HDL. The design focuses on simplicity, modularity, and efficiency. The processor consists of a pipelined structure to enhance throughput and overall device performance.

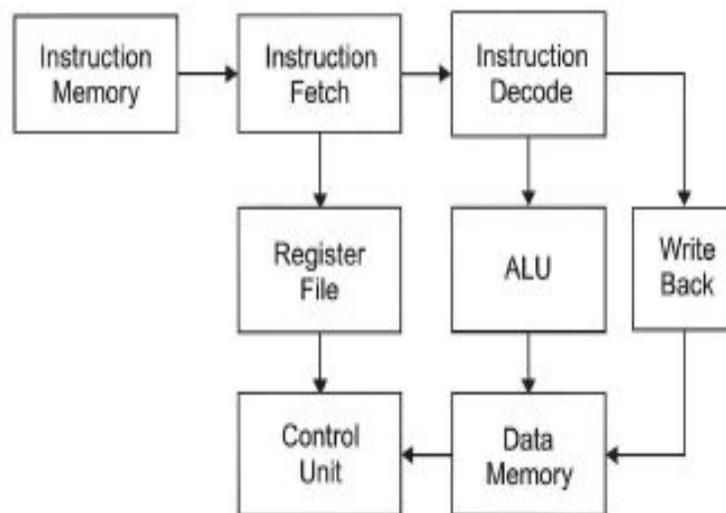


Fig. 1. Block Diagram of 32-BIT RISC Processor

II. RELATED WORK

The idea of RISC structure was brought within the early 1980s via research tasks carried out at main universities and companies. One of the earliest a hit RISC-based processors was the MIPS architecture, which proven how simplified practise units and pipelining should substantially improve overall performance. MIPS processors have become a famous version for each instructional and commercial processor designs. ARM processors later followed RISC ideas and have become widely utilized in mobile and embedded structures due to their low power intake and high efficiency. Several studies have centered on optimizing ARM-based totally processors for strength, performance,

and region. More these days, the open-source RISC-V structure has gained attention for academic studies and industrial improvement. RISC-V allows designers to develop custom processors with out license restrictions. Many researchers have proposed special 32-bit RISC cores based on RISC-V, focusing on pipelining, danger handling, and coffee-electricity optimization. Most existing works spotlight the importance of modular design, efficient datapath organization, and pipeline implementation. The proposed paintings on this paper follows comparable concepts while maintaining a simplified and educationally targeted 32-bit RISC processor design.

III. ARCHITECTURE of 32-BIT RISC PROCESSOR

The architecture of the proposed 32-bit RISC processor is designed to attain high performance with minimal hardware complexity. The processor is primarily based on a load-keep structure and helps a hard and fast-length 32-bit instruction layout. The primary additives of the processor architecture are defined under.

1. Instruction Set Architecture (ISA)

The training set includes fundamental arithmetic, logical, memory access, and manage instructions. These encompass ADD, SUB, AND, OR, LOAD, STORE, JUMP, and BRANCH operations. The commands are labeled into R-kind, I-kind, and J-kind formats for easier interpreting and execution.

2. Program Counter (PC)

The Program Counter holds the address of the next instruction to be finished. It is updated sequentially or changed all through department and bounce operations.

3. Instruction Memory

Instruction memory shops this system commands. During every clock cycle, the practise similar to the current PC value is fetched.

4. Register File

The sign up document includes a fixed of popular-reason registers used for storing operands and intermediate results. It supports simultaneous examine and write operations.

5. Arithmetic Logic Unit (ALU)

The ALU plays mathematics and logical operations as determined with the aid of the manipulate unit. It executes operations including addition, subtraction, AND, OR, and contrast.

6. Data Memory

Data memory is used for load and save operations. It allows read and write get admission to to keep transient records.

7. Control Unit

The manage unit generates the necessary manage indicators for every training to control records flow and execution.

8. Pipelining

The processor uses a 5-degree pipeline including Instruction Fetch, Instruction Decode, Execute, MemoryAccess, and Write Back levels. This permits parallel execution of more than one commands.

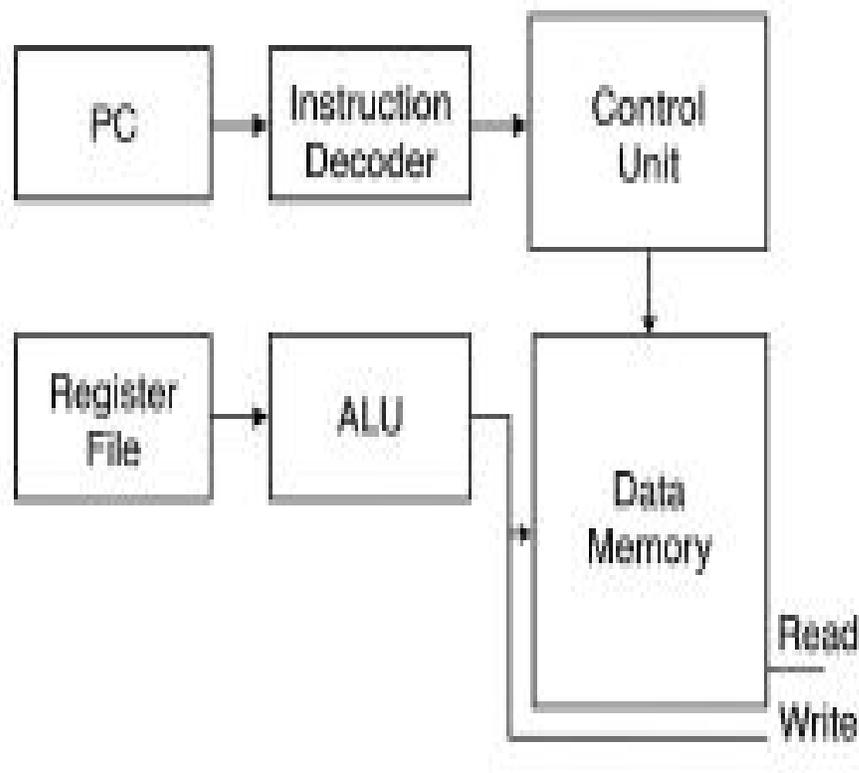


Fig. 2. Architecture Of 32-bit RISC Processor

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.

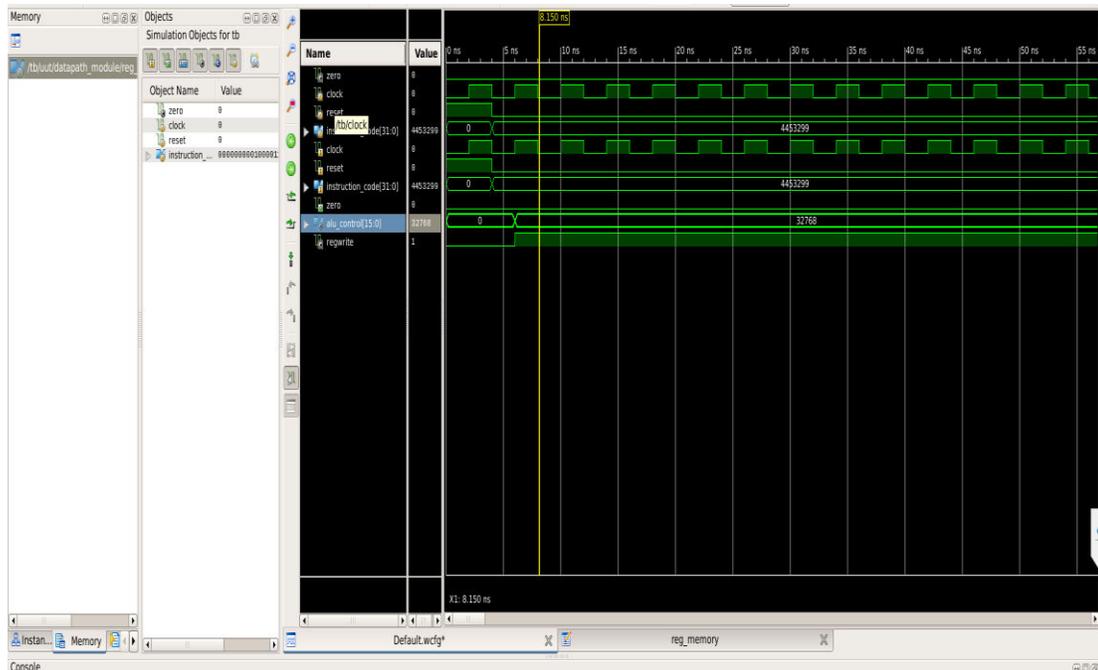


Fig. 3. simulation of waveform of 32-bit RISC Processor

V. CONCLUSION AND FUTURE WORK

Conclusion

This paper supplied the design and implementation of a 32-bit RISC processor primarily based on essential RISC concepts. The processor structure was designed with a simplified practise set, modular datapath, and pipelined execution to achieve better instruction throughput. The layout turned into carried out the use of Verilog HDL and proven through simulation and synthesis. The outcomes suggest that the proposed processor operates correctly and efficaciously with reduced hardware complexity. The modular structure of the layout makes it appropriate for educational use and embedded system packages.

Future Work

Although the modern-day design meets the fundamental practical necessities, several upgrades can be delivered in future work:

- Implementation of facts forwarding and risk detection units
- Addition of cache reminiscence for faster reminiscence get admission to

- Integration of power optimization strategies
- Support for interrupt dealing with
- Extension towards RISC-V like minded 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] Li Jiemin, Zhang Shancong. Research and performance optimization of five-stage pipeline RISC-V processor[J/OL]. Microelectronics and Computer: 1-7[2021-12-26].DOI: 10.19304/J .ISSN1000-7180.2021.0856.
- [8] Raveendran A, Patil VB, Selvakumar D, et al. A RISC-V instruction set processor-micro-architecture design and analysis[C]//2016 International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI SATA), 2016: 1-7
- [9] Banakar R, Steinke S, Lee B S, et al. Scratchpad memory; design alternative for cache on-chip memory in embedded systems[C]//2015 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS),2002:73-78.
- [10] Keller B.A RISC-V processor SoC with integrated power management at submicrosecond timescales in 28 nmFD-SOI[J].IEEE Journal of Solid-State Circuits, 2017, 1:1863-1875.
- [11] Patil V, Raveendran A, Sobha PM, et al. Out of order floating point coprocessor for RISC V ISA[C]//201519th International Symposium on VLSI Design and Test, 2015:1-7.

- [12] Rao J, Ao T, Xu S, et al. Design exploration of SHA-3 ASIP for IoT on a 32-bit RISC-V processor[J].IEICE Transactions on Information and Systems, 2018 (11): 2698-2705.
- [13] Hao Zhenhe, Jiao Jiye, Li Yuqian. Design and implementation of RISC-V microprocessor based on AHB bus[J]. Computer Engineering and Applications, 2020, 56(20): 52-58.
- [14] “Xilinx UG230 Spartan-3E FPGA Starter Kit Board User Guide”, Available at http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf
- [15] “Vivado Design Suite User Guide: Implementation-Xilinx” Available at http://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_2/ug904-vivado-implementation.pdf

