Simulation of a High Performance 32-bit RISC Processor

A DESIGN REPORT

Submitted by

Satyajit Deokar
Sanjeev Krishnan
Siddesh Patil
Neil Tauro

Submitted to

Prof. Mark Faust

Content

1. Introduction 3

2. Features of RISC-V **3**

2.1 Diagram **3**

2.1.1 Block Diagram **3**

2.2 Architecture Overview **4**

3. RISC-V Instruction Set Architecture (ISA) **4**

3.1 Base Instruction Sets **4**

3.2 Extensions **4**

4. RISC-V Memory Model **4**

5. RISC-V Applications **5**

6. RISC-V Development Tools **5**

7. Advantages of RISC-V **5**

8. Appendices **6**

8.1 Simulation Result **6**

9. Conclusion **7-8**

10. Reference **8**

1. Introduction

RISC-V is an open-standard, Reduced Instruction Set Computing (RISC) architecture designed to support a broad spectrum of computing applications, ranging from embedded systems to high-performance computing. Originating from the University of California, Berkeley, RISC-V is unique among ISAs as it is free and open, allowing anyone to use, modify, and extend it without licensing fees. Unlike proprietary ISAs, it offers flexibility, modularity, and extensibility, enabling tailored implementations for specific use cases. Its open nature has driven widespread adoption across academia, research, and industry, fostering innovation in areas such as AI, edge computing, and custom hardware accelerators.

2. Features of RISC-V

2.1 Diagram

2.1.1 Block Diagram

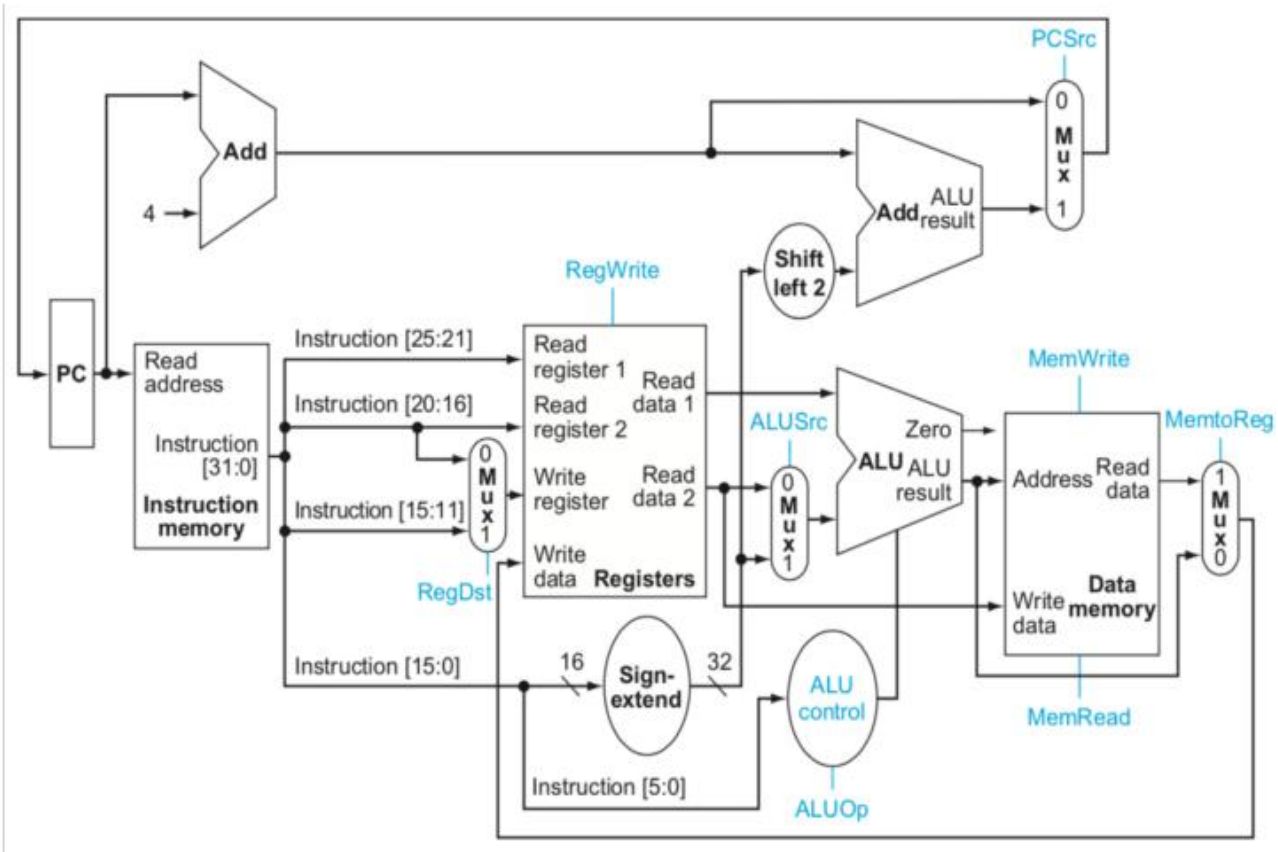

Fig. 1 Block diagram RISC V

- **Instruction Memory**: Retrieves instructions from memory.
- **Instruction Decode**: Decodes fetched instructions and determines control signals.
- **Register File**: Stores general-purpose registers used for computation.
- **Arithmetic and Logic Unit (ALU)**: Performs arithmetic and logical operations.
- **Control Unit**: Manages instruction execution flow.
- **Memory Interface**: Handles data transfer between processor and memory.

2.2 Architecture Overview

RISC-V is designed to be a simple yet powerful instruction set architecture (ISA) with key characteristics that enhance its efficiency and adaptability:

- **Load-Store Architecture**: Follows a register-based design where computations occur in registers, while memory operations are handled separately, improving performance and efficiency.
- **Scalability**: Supports multiple data path widths, including **RV32I** (32-bit), **RV64I** (64-bit), and **RV128I** (128-bit), making it suitable for a wide range of applications from microcontrollers to high-performance computing.
- **Modular ISA**: Features a minimal base instruction set with optional extensions such as floating-point arithmetic, atomic operations, vector processing, and custom extensions tailored to specific workloads.
- **Privilege Levels**: Defines multiple execution modes—**Machine, Supervisor, User, and Hypervisor**—providing enhanced security, isolation, and system control.
- **Open and Extensible**: Unlike proprietary ISAs, RISC-V is open-source, allowing designers to modify and extend it without licensing restrictions, fostering innovation in custom hardware accelerators, AI processors, and specialized computing.

These features make RISC-V a highly versatile and future-proof architecture, driving its adoption across industries, academia, and research.

3. RISC-V Instruction Set Architecture (ISA)

3.1 Base Instruction Sets

- **RV32I:** 32-bit base integer instruction set.

3.2 Extensions

- **M (Multiplication & Division):** Provides integer multiply and divide instructions.

4. RISC-V Memory Model

- **Little-endian memory model (default).**
- **Load and Store architecture (no direct memory-to-memory operations).**
- **Memory Protection Units (MPU) and Physical Memory Protection (PMP) for security.**

5. RISC-V Applications

- Embedded Systems
- Internet of Things (IoT)
- Machine Learning Accelerators
- High-Performance Computing (HPC)
- Cloud Computing and Datacenters

6. RISC-V Development Tools

- **GNU Compiler Toolchain (GCC, RVGCC Cross Compiler)**

7. Advantages of RISC-V

- Open-source and community-driven development.
- Scalability across multiple domains.
- Power-efficient and performance-optimized.
- Industry and academic support with growing ecosystem.

8. TestCases

- Refer the Verification Plan Document

9. Appendices

8.1 Simulation Result

Actual Output:

```
PC: 0000004c
Instruction type: sub   Instruction: 41390a33
x2 (sp): 00010000
x3 (gp): 00000019
x4 (tp): 00000000
x5 (t0): 00000019       x6 (t1): 00000032       x7 (t2): 00000014
x8 (s0): 0000001e
x9 (s1): fffffe2
x10 (a0): 0000000f      x11 (a1): ffffffd3      x12 (a2): 00000028      x13 (a3): fffffec
x14 (a4): 0000003c      x15 (a5): 80000000      x16 (a6): 00000001      x17 (a7): 7fffffff
x18 (s2): 7fff07ff      x19 (s3): ffffffff      x20 (s4): 7fff0800      x21 (s5): 00000000   x22 (s6): 0000
0000
x23 (s7): 00000000      x24 (s8): 00000000      x25 (s9): 00000000      x26 (s10): 00000000  x27 (s11): 000
00000
x28 (t3): 00000000      x29 (t4): 00000000      x30 (t5): 00000000      x31 (t6): 00000000
```

Expected Output (Cornell Interpreter):

```
Register Expected Results
## expect[0] = 0x00000000
## expect[1] = 0x00000000
## expect[2] = 0x00000000
## expect[3] = 0x00000019
## expect[4] = 0x00000000
## expect[5] = 0x00000019
## expect[6] = 0x00000032
## expect[7] = 0x00000014
## expect[8] = 0x0000001e
## expect[9] = 0xfffffe2
## expect[10] = 0x0000000f
## expect[11] = 0xffffffd3
## expect[12] = 0x00000028
## expect[13] = 0xfffffec
## expect[14] = 0x0000003c
## expect[15] = 0x80000000
## expect[16] = 0x00000001
## expect[17] = 0x7fffffff
## expect[18] = 0x7fff07ff
## expect[19] = 0xffffffff
## expect[20] = 0x7fff0800
## expect[21] = 0x00000000
## expect[22] = 0x00000000
## expect[23] = 0x00000000
## expect[24] = 0x00000000
```

10. Conclusion

The RISC-V processor offers a simple yet powerful and highly adaptable architecture, making it an excellent choice for diverse applications, from academic research and education to cutting-edge industry solutions. Its modular design, open-source nature, and structured pipeline stages enable easy customization and scalability, meeting the demands of both resource-constrained embedded systems and high-performance computing.

Key Takeaways:

- **Efficiency through Simplicity**: The streamlined instruction set enhances execution efficiency while maintaining ease of design and implementation.
- **Scalability and Adaptability**: RISC-V's extensible nature allows developers to integrate custom instructions for specialized applications without disrupting core functionality.
- **Broad Educational and Industry Impact**: As an open-source ISA, RISC-V fosters innovation, collaboration, and accessibility, making it a driving force in hardware research, development, and commercialization.

With its emphasis on clarity, modularity, and openness, RISC-V serves as both a robust foundation for hardware engineers and an essential educational tool for understanding modern processor architecture.

11. Reference

https://csg.csail.mit.edu/6.375/6_375_2019_www/resources/riscv-spec.pdf

https://github.com/riscv/riscv-isa-manual

https://www.cs.cornell.edu/courses/cs3410/2019sp/riscv/interpreter/