Documentação Caminho de Dados RISC-V Grupo 12

Mateus S. M. Peixoto[1], Adryan M. B. D. Santos[1]

[1]Instituto de Ciências Exatas e Aplicadas - Universidade Federal de Ouro Preto (UFOP) - João Monlevade - MG - Brazil

{mateus.serretti, adryan.santos}@aluno.ufop.edu.br

Abstract. This paper documents the implementation of a single-cycle RISC-V processor in Verilog, developed as a practical assignment for the Fundamentals of Computer Organization and Architecture course (CSI211). The processor supports a specific subset of 7 instructions assigned to Group 12, including data transfer (lw, sw), arithmetic-logic (sub, xor, addi), shift (srl), and control flow (beq) operations. The architecture, modular design, and simulation results are presented.

Resumo. Este artigo documenta a implementação de um processador RISC-V de ciclo único em Verilog, desenvolvido como trabalho prático da disciplina de Fundamentos de Organização e Arquitetura de Computadores (CSI211). O processador implementa um subconjunto específico de 7 instruções atribuídas ao Grupo 12, incluindo operações de transferência de dados (lw, sw), lógico-aritméticas (sub, xor, addi), de deslocamento (srl) e de desvio de fluxo (beq). A arquitetura, o design modular e os resultados da simulação são apresentados.

Introdução

Este trabalho descreve o projeto e a implementação de um processador de ciclo único para um subconjunto da arquitetura RISC-V. A atividade foi proposta como parte da avaliação da disciplina de Fundamentos de Organização e Arquitetura de Computadores (CSI211), da Universidade Federal de Ouro Preto (UFOP).

O objetivo principal é aplicar os conceitos teóricos de organização de computadores na construção de um processador funcional em Verilog, capaz de executar um conjunto pré-definido de instruções. O processador foi projetado para o subconjunto de instruções designado ao Grupo 12, que abrange os principais tipos de operações da arquitetura RISC-V. O caminho de dados geral do processador implementado é ilustrado na Figura 1.

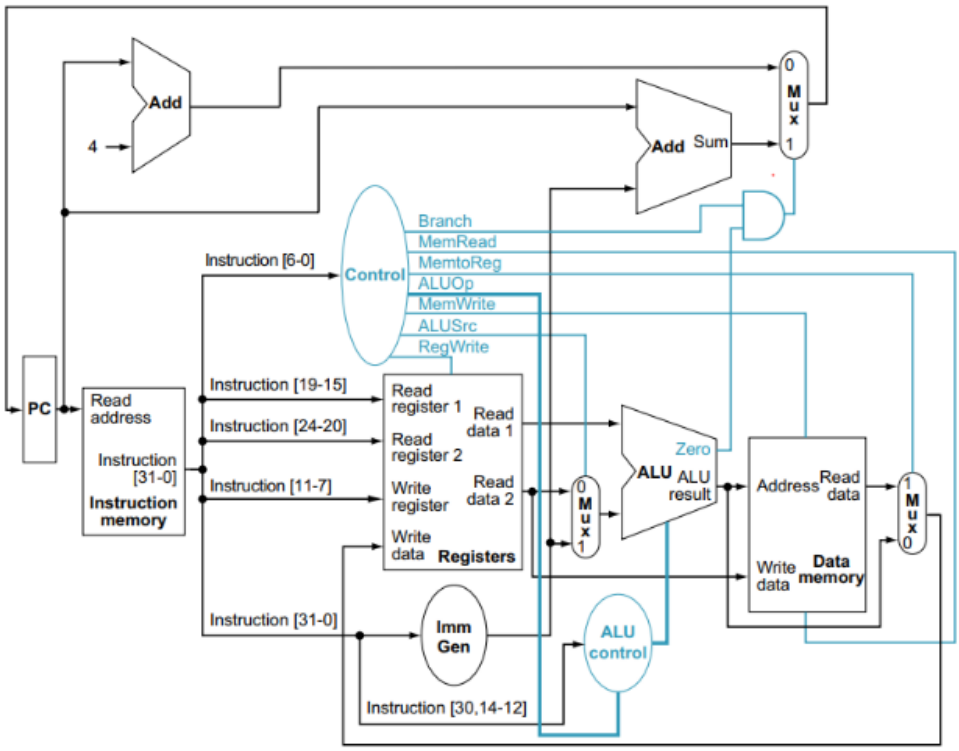

Figura 1. Caminho de dados do processador RISC-V de ciclo único.

Arquitetura e Implementação

O processador foi projetado com uma abordagem modular para facilitar o desenvolvimento, a depuração e a reutilização de componentes. A estrutura do projeto divide o processador nas unidades funcionais clássicas de um caminho de dados de ciclo único.

O projeto foi organizado em um diretório `src/`, contendo os seguintes módulos Verilog:

- **riscv.v:** Módulo de topo que instancia e interconecta todos os componentes.
- **control.v:** Unidade de controle principal, responsável por gerar os sinais de controle a partir do `opcode` da instrução.
- **alu_control.v:** Unidade de controle secundária que define a operação a ser realizada pela ULA.
- **alu.v:** Unidade Lógica e Aritmética, que executa as operações de cálculo.
- **reg_file.v:** Banco de registradores, contendo os 32 registradores de uso geral.
- **imm_gen.v:** Gerador de imediatos, que extrai e estende o sinal dos valores imediatos de diferentes formatos de instrução.
- **data_memory.v e instruction_memory.v:** Módulos que simulam as memórias de dados e de instruções, respectivamente.

Instruções Implementadas (Grupo 12)

O subconjunto de instruções suportado pelo processador é composto por 7 operações que cobrem os formatos R, I, S e B, conforme detalhado na Tabela 1.

Tabela 1. Subconjunto de instruções implementadas.

Tipo	Instrução	Descrição
I-Type	‘lw‘	Load Word
S-Type	‘sw‘	Store Word
R-Type	‘sub‘	Subtração
R-Type	‘xor‘	OU Exclusivo (XOR)
I-Type	‘addi‘	Adição com Imediato
R-Type	‘srl‘	Shift Right Logical
B-Type	‘beq‘	Branch on Equal

Resultados e Simulação

Para validar o funcionamento do processador, foi desenvolvido um `testbench` (`riscv_tb.v`) que instancia o módulo `riscv.v` e carrega um programa de teste a partir de um arquivo binário (`instruction_mem.bin`). Este arquivo é a tradução do código Assembly de teste, realizado previamente pelo montador desenvolvido na primeira etapa do trabalho prático.

A Figura 2 apresenta o estado final dos registradores após a execução do programa de teste, demonstrando os valores armazenados em cada um.

A simulação foi conduzida utilizando o Icarus Verilog. O processo consiste em compilar todos os arquivos-fonte Verilog e, em seguida, executar o simulador, que gera um arquivo de saída.

```
# Compilação (deve ser digitado em uma linha)
iverilog -o riscv_sim src/*.v
                      test/riscv_tb.v

# Execução da simulação
vvp riscv_sim
```

Ao final da execução, o `testbench` exibe no terminal o estado final dos 32 registradores. A validação é feita comparando-se os valores obtidos com os resultados esperados após a execução do programa de teste, confirmando a correta implementação de cada instrução e do caminho de dados.

Conclusão

O desenvolvimento deste projeto permitiu a aplicação prática dos conceitos fundamentais de arquitetura de computadores. A implementação de um processador RISC-V de ciclo único em Verilog, desde a definição do caminho de dados até a criação das unidades de controle e ULA, solidificou o entendimento sobre como as instruções de máquina são decodificadas e executadas pelo hardware. O projeto atendeu a todos os requisitos propostos, resultando em um processador funcional para o subconjunto de instruções especificado.

Referências

Patterson, D. A., & Hennessy, J. L. (2017). *Computer Organization and Design RISC-V Edition: The Hardware Software Interface*. Morgan Kaufmann.

Waterman, A., Asanović, K. (Eds.). (2019). *The RISC-V Instruction Set Manual, Volume I: Unprivileged ISA*. RISC-V International. Disponível em: `https://riscv.org/specifications/ratified/`.

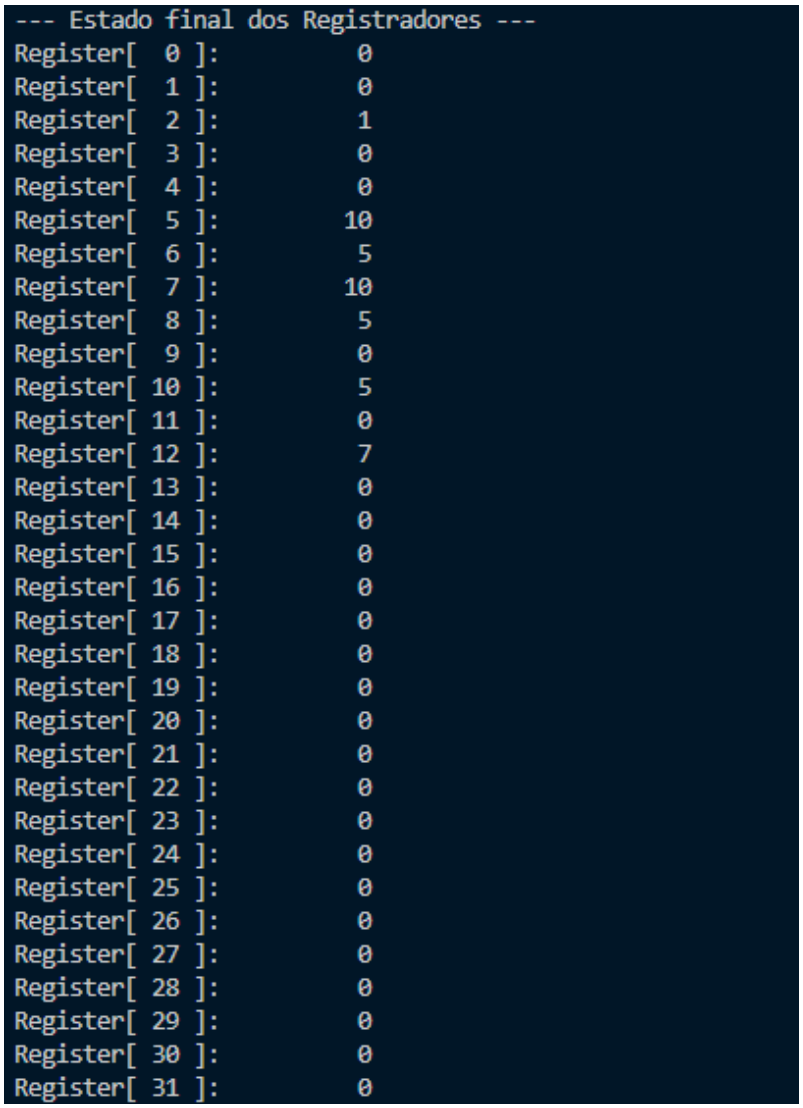

Figura 2. Estado final dos Registradores após a simulação.