

UNIVERSIDADE FEDERAL DE VIÇOSA - *CAMPUS FLORESTAL*
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
CCF-252

TRABALHO PRÁTICO - CAMINHO DE DADOS SIMPLES DO RISC-V

ERICH PINHEIRO AMARAL - 5915
PEDRO PAULO PAZ DO NASCIMENTO - 5937

FLORESTAL - MG
2025

ERICH PINHEIRO AMARAL
PEDRO PAULO PAZ DO NASCIMENTO

TRABALHO PRÁTICO - CAMINHO DE DADOS SIMPLES DO RISC-V

Relatório do trabalho prático da disciplina CCF-252 , sendo o objetivo implementar uma versão simplificada do caminho de dados do RISC-V. Avaliação feita pelo Prof. Jose Augusto Miranda Nacif, UFV *Campus Florestal*.

RESUMO

O trabalho consiste na implementação de uma versão simplificada de um caminho de dados para a arquitetura RISC-V, seguindo especificações de uma tabela formada para cada aluno com um conjunto de instruções que deve ser implementada. Por fim, o sistema deverá receber um arquivo contendo instruções em linguagem de máquina (formato binário). Após a execução dessas instruções por meio do testbench, o sistema deve exibir no terminal os valores finais de cada registrador, permitindo a verificação do funcionamento correto do caminho de dados implementado.

Palavras-chave: caminho de dados, instruções, verilog,

ABSTRACT

The project consists of implementing a simplified version of a datapath for the RISC-V architecture, following specifications defined in a table assigned to each student, with a specific set of instructions to be implemented. Finally, the system must receive a file containing instructions in machine language (binary format). After executing these instructions through the testbench, the system should display the final values of each register in the terminal, allowing verification of the correct functionality of the implemented datapath.

Keywords: datapath, instructions, verilog

1 INTRODUÇÃO

As instruções são comandos básicos que informam ao processador quais operações devem ser realizadas, como somar valores, acessar a memória ou tomar decisões com base em comparações. Elas fazem parte de um conjunto bem definido chamado ISA. Para que essas instruções possam ser compreendidas e executadas por uma máquina, é necessário um processo de tradução. Primeiro, o código-fonte escrito em uma linguagem de alto nível é convertido pelo compilador para o Assembly. Em seguida, o montador transforma esse código Assembly em linguagem de máquina, composta por números binários, que são interpretados pelo processador, reduzindo o nível de abstração até chegar a execução.

Este trabalho tem como foco a implementação de um caminho de dados simplificado para a arquitetura RISC-V, utilizando a linguagem Verilog. A RISC-V é uma arquitetura de conjunto de instruções (ISA) moderna, amplamente adotada em pesquisas e no desenvolvimento de processadores. Cada grupo é responsável por implementar um subconjunto específico de instruções, com base na estrutura apresentada na Figura 1.

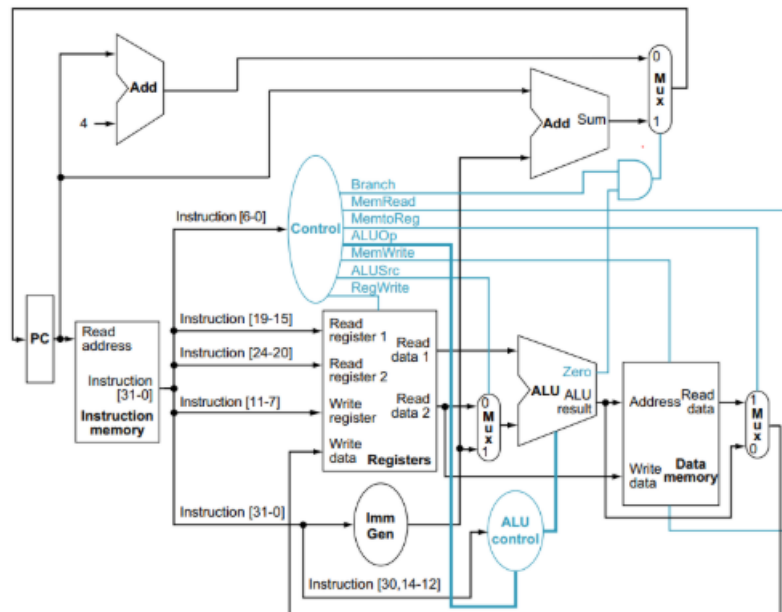


Figura 1 – Caminho de dados

Compreender como instruções em baixo nível são interpretadas pelo hardware e o caminho de dados por onde cada instrução passa é essencial para entender o funcionamento interno de uma máquina. A implementação prática de um caminho de dados, ainda que simplificado, permite visualizar de forma concreta os conceitos fundamentais de organização de computadores e arquitetura de processadores.

2 OBJETIVOS

O objetivo principal deste trabalho prático foi a implementação de um processador simples utilizando um caminho de dados simplificado da arquitetura RISC-V, com base na figura 1. Ao longo do desenvolvimento, foi necessário aplicar de forma prática diversos conceitos fundamentais estudados na disciplina. A construção do processador exigiu a implementação de um subconjunto de instruções da RISC-V, contemplando diferentes formatos (R, I, S, SB), além da conexão de diversos módulos em Verilog, como registradores, memória, ALU, unidade de controle, entre outros. Esse processo apresentou um desafio importante, pois exigiu a compreensão detalhada do funcionamento interno de cada instrução e sua integração ao caminho de dados.

O desenvolvimento do projeto proporcionou um aprendizado significativo nas seguintes áreas:

- Aprendizado da linguagem (Verilog);
- Instruções em RISC-V;
- Tipos de instruções;
- Lógica do caminho de dados;

- Integração de módulos fundamentais de um processador;
- interpretação de instruções em linguagem de máquina;

3 IMPLEMENTAÇÃO

A implementação foi inicialmente separada em módulos, que depois foram integrados no módulo principal, main, que serão apresentados e explicados a seguir.

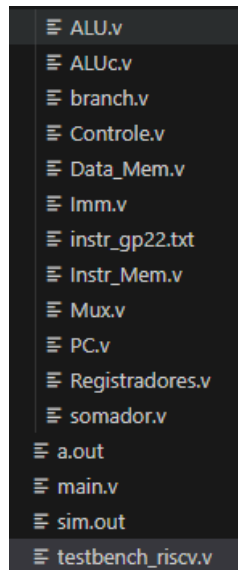


Figura 2 – implementação em módulos separados

ALU: Esse módulo implementa uma ALU básica capaz de executar soma, subtração, operações lógicas AND/OR, e deslocamento à direita. Ele também fornece uma indicação de "zero" útil em instruções condicionais.

ALUc: Atua como um tradutor entre a unidade de controle principal e a ALU. Ele garante que, dependendo do tipo de instrução e seus campos de função, a ALU receba o sinal correto para realizar a operação esperada.

Branch: Gera o sinal de controle que indica se um desvio (branch) deve ocorrer. Ele ativa o desvio apenas quando ambos os sinais de entrada Branch (controle da instrução) e zero (resultado da ALU) são iguais a 1.

Controle: É a unidade de controle principal do processador. Ele recebe como entrada o campo Opcode de uma instrução RISC-V e gera diversos sinais de controle que determinam como os demais blocos do processador devem se comportar durante a execução da instrução.

Data-Men: Implementa a memória de dados do processador. Ele permite leitura e escrita de bytes em uma memória de 64 palavras de 32 bits, com suporte à instrução lb e sb.

Imm: É o gerador de imediato do processador. Ele recebe uma instrução de 32 bits como entrada e extrai o valor imediato correspondente, com sinal estendido, de acordo com o tipo da instrução.

Instr-Men: Implementa a memória de instruções do processador, responsável por fornecer a instrução a ser executada com base no endereço do programa (PC).

Mux: Este módulo escolhe entre dois valores para decidir o que usar, por exemplo, entre usar um registrador ou um imediato, ou escolher entre resultado da ALU e dado da memória.

Pc: O módulo PC implementa o contador de programa, responsável por armazenar e atualizar o endereço da próxima instrução a ser executada.

Registradores: Implementa o banco de registradores do processador, contendo os 32 registradores de propósito geral (x0 a x31). Todos os registradores são inicializados como zero.

Somador: O módulo somador implementa um somador simples de 32 bits sendo útil para calcular incrementos no PC, endereços de memória ou resultados aritméticos da ALU.

Main: Implementa o caminho de dados e controle principal de um processador simples, integrando todos os módulos.

Testbench: O testbench cria simulação para testar o main e ao final da simulação, imprime o conteúdo final de todos os 32 registradores.

4 RESULTADOS

A implementação do caminho de dados da arquitetura RISC-V na linguagem Verilog apresentou desafios iniciais, principalmente relacionados ao entendimento da linguagem. Porém, o algoritmo do processo estava clara desde o princípio: receber as instruções em linguagem de máquina (binário), e processá-las através do caminho de dados, de acordo com o tipo de instrução. Por fim, o sistema deve apresentar os valores finais dos registradores após a execução completa das instruções.

Durante o desenvolvimento, notamos que instruções do tipo R foram implementadas com menos dificuldade, enquanto os tipos S e SB exigiram maior atenção devido a sua alta complexidade. As instruções que envolvem leitura e escrita na memória (como lb e sb), assim como as instruções de desvio condicional (como beq), representaram os maiores pontos de dificuldade no projeto, uma vez a manipulação precisa de endereços de memória e condições de salto.

Com pesquisa e o auxílio de ferramentas de IA, conseguimos solucionar problemas específicos que surgiram, como o tratamento de instruções que utilizam a lógica de desvio e problemas na leitura e escrita de dados na memória. Por fim, o resultado final foi um processador simples funcional para o conjunto de instruções referente ao grupo 22 do trabalho prático, e o processo proporcionou um aprendizado prático valioso sobre a arquitetura RISC-V.

5 CONCLUSÕES

Após a implementação de um processador RISC-V em Verilog, foi possível consolidar conhecimentos sobre a arquitetura RISC-V, incluindo o funcionamento dos diferentes formatos de instrução (R, I, S, SB) e a função de cada campo binário na execução do programa. O desenvolvimento dos módulos fundamentais, como a unidade de controle, ALU, registradores, memória

e lógica de desvio, ajudou a ter uma melhor compreensão prática de como funciona o processador em baixo nível de abstração. Além disso, o uso da linguagem Verilog aprofundou nosso conhecimento sobre essa linguagem.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- David A. Patterson, John L. Hennessy Computer Organization and Design: The Hardware/Software Interface – RISC-V Edition Morgan Kaufmann, 2017.
- The RISC-V Instruction Set Manual – Volume I: User-Level ISA, Version 2.2 RISC-V Foundation, 2017 – <https://riscv.org/technical/specifications/>
- Cornell RISC-V Web Emulator – <https://www.cs.cornell.edu/courses/cs3410/2020sp/riscv/>