

# Caminho de Dados Simples e Controle MIPS

Gabriela Marangoni Radigonda e Jean Carlos Martins Miguel

Universidade Tecnológica Federal do Paraná – UTFPR

COCIC – Coordenação do Curso de Bacharelado em Ciência da Computação  
Campo Mourão, Paraná, Brasil

[gabrielaradigonda@alunos.utfpr.edu.br](mailto:gabrielaradigonda@alunos.utfpr.edu.br)

[jeanm@alunos.utfpr.edu.br](mailto:jeanm@alunos.utfpr.edu.br)

## Resumo

O trabalho realizado tem em vista a implementação do conteúdo visto em sala de aula. A implementação consiste no algoritmo decodificador MIPS, um programa que realiza a tradução do código binário para linguagem MIPS. As instruções que são suportadas nesta implementação são: **add**, **sub**, **and**, **or**, **slt**, **lw**, **sw** e **beq**. Os circuitos foram construídos utilizando o Logisim, uma ferramenta para simulação digital de circuitos lógicos.

## Introdução

Este artigo apresenta a simulação da implementação do caminho de dados para um subconjunto de instruções presentes no MIPS. As instruções abordadas nesta implementação são: **add**, **sub**, **and**, **or**, **slt**, **lw**, **sw** e **beq**. A implementação está dividida em três partes: a primeira traz a implementação do *datapath* (caminho de dados) com o Controle da ULA e Controle Principal ambos em funcionamento manual. A segunda parte, por sua vez, apresenta o Controle Principal e Controle da ULA implementados e a última etapa traz a implementação das instruções **addi** e **jump**. Para tal propósito, foi utilizado o Logisim, uma ferramenta para simulação digital de circuitos lógicos.

## 1. Métodos que foram utilizados no trabalho

Para execução do trabalho foi utilizado como base os arquivos disponibilizados contendo o esquema principal de funcionamento do circuito (Registadores, Memória de Dados, Memória de Instruções), subcircuitos e conjuntos de instruções em Assembly para fins de testes.

A Figura 1 ilustra como era o circuito Datapath, inicialmente sem as conexões dos Caminhos de Dados.

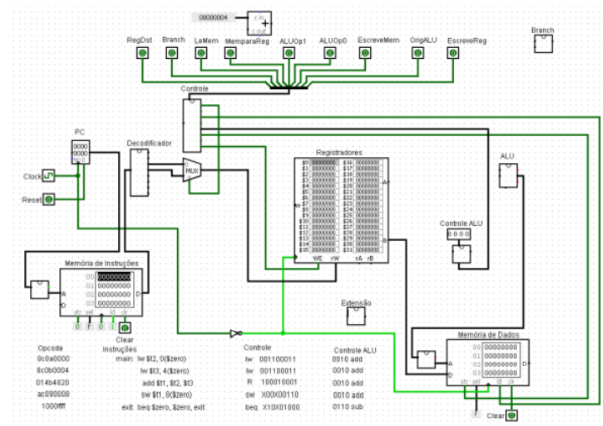


Figura 1: circuito inicial do datapath passado pelo professor.

Com base no circuito apresentado acima e no diagrama do datapath foi construído todo o circuito, realizando as conexões e inserindo os componentes necessários. A Figura 2 apresenta o diagrama utilizado como base para a montagem do circuito.

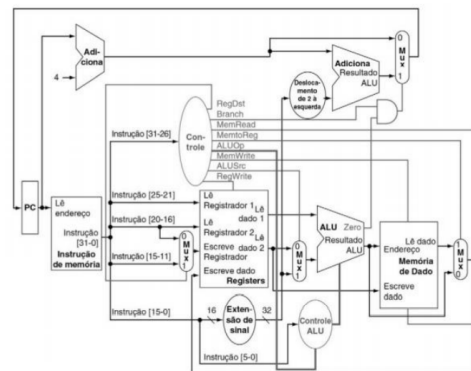


Figura 2: Diagrama do caminho de dados.

Após a construção do circuito final de Datapath no Logisim, foram realizadas as implementações do Controle da ULA e Controle Principal. Por fim, foi criado um novo circuito com base no anterior para dar suporte às instruções addi e jump.

## 2. Memória de Instrução

A primeira etapa do ciclo de instrução do MIPS é a busca de instrução. Para isso, a arquitetura utiliza uma memória de instruções que armazena o programa a executar. Temos um PC e a Figura 3 abaixo que indica o endereço da instrução a executar. Para executar a instrução basta fazer o load image da instrução que deseja utilizar e escolher a opção .mem. E uma ULA que efetua uma operação de adição para calcular o endereço da próxima instrução a ser executar.

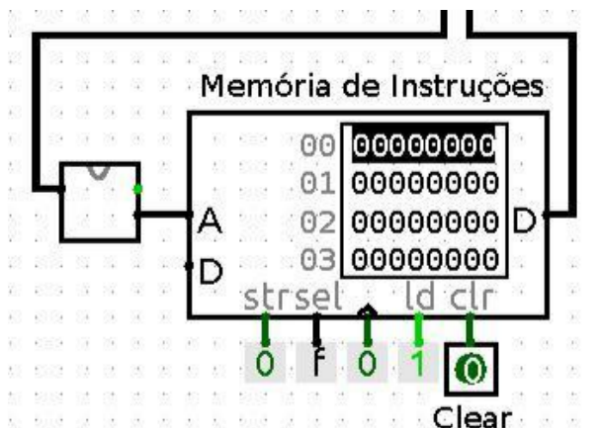


Figura 3: armazenamento de instruções a serem efetuadas

## 3. Decodificador

A segunda etapa é a decodificação da instrução, onde é determinado a operação para ser executada, os argumentos a serem usados e onde o resultado será armazenado. Na parte da decodificação temos: Banco de registradores, unidade de controle, unidade de memória de dados, unidade de extensão de sinal, unidade lógica aritmética (ULA) e junção das unidades.

- **Banco de registradores**

Possibilita o acesso ao valor dos registradores do processador.

- **Unidade de Controle**

A unidade de controle é responsável pela configuração de todos os sinais de controle de modo que cada instrução seja executada apropriadamente. A maioria dos sinais podem ser gerados a partir das próprias opções da instrução.

Como o caminho de dados contém todas as unidades funcionais e conexões necessárias para implementar o conjunto de instruções da arquitetura. A unidade de controle manipula o caminho de dados baseado na instrução que está sendo executada. O processador possui dez sinais de controle que regulam o caminho de dados. Os sinais de controle podem ser gerados a partir de um circuito combinatório que recebe como entrada com a codificação da instrução de 32 bits. A seguir, a Figura 4 mostra os sinais produzidos pelo circuito.

Operation	RegDst	RegWrite	ALUSrc	ALUOp	MemWrite	MemRead	MemToReg
add	1	1	0	010	0	0	0
sub	1	1	0	110	0	0	0
and	1	1	0	000	0	0	0
or	1	1	0	001	0	0	0
slt	1	1	0	111	0	0	0
lw	0	1	1	010	0	1	1
sw	X	0	1	010	1	0	X
beq	X	0	0	110	0	0	X

Figura 4: Tabela de sinais de controle.

Para gerar sinais de controle a UC precisa como entrada 13 bits. Sendo seis bits usados como o opcode da instrução. Outros seis bits para o campo func da instrução e uma saída Zero da ULA. A unidade de controle gera 10 bits de saída que correspondem aos sinais mencionados na tabela acima.

- **Unidade de memória de dados**

Permite ler e escrever na memória. O sinal Address indica a posição de memória a ler ou escrever e o sinal Write data indica os dados a escrever. O sinal Read data contém o valor lido da memória. Os sinais MemRead e MemWrite permitem selecionar a operação de leitura ou escrita, respectivamente

- **Unidade de extensão de sinal**

Efetua a extensão de valores de 16 bits, em complemento de dois, para valores de 32 bits. Nos acessos à memória o endereço a ser somado é calculado através da adição de uma constante de 16 bits com o valor de um registrador. Os 16 bits provêm de um campo de instrução, sendo necessário estender o valor para 32 bits e ligar à ULA, assim como a primeira saída do banco de registradores.

- **Unidade lógica aritmética(ULA)**

Uma unidade lógica aritmética (ULA) é um circuito eletrônico digital que executa operações lógicas e aritméticas bit a bit em números binários inteiros. A ULA realiza as operações possuindo duas entradas de 32 bits, uma saída de 32 bits para o resultado e um sinal adicional que indica se o resultado da operação for zero. O sinal de controle Operation da ULA permite selecionar a operação a ser aplicada.

- **Junção das unidades**

Em ciclos diferentes as várias unidades podem ser utilizadas para instruções distintas. E como uma mesma unidade não pode ser usada várias vezes no mesmo ciclo, o uso de multiplexadores se torna necessário uma vez que este componente permite selecionar uma entrada entre várias através de um sinal de controle.

O valor que é escrito no banco de registradores pode vir da memória (lw) ou da ULA (outras instruções). A origem do valor é escolhida pelo multiplexador destacado pelo círculo azul.

## 4. Resultados

A partir das implementações realizadas, foi obtido como resultado final um circuito com controle manual e um circuito com controle combinacional que dão suporte às instruções add, sub, and, or, slt, lw, sw e beq. Além desses, foi implementado um terceiro circuito com controle combinacional acrescentando suporte às instruções addi e jump. Abaixo, as Figuras 5, 6 e 7 apresentam o circuito Datapath, subcircuito de Controle Principal e sub circuito de Controle da ULA.

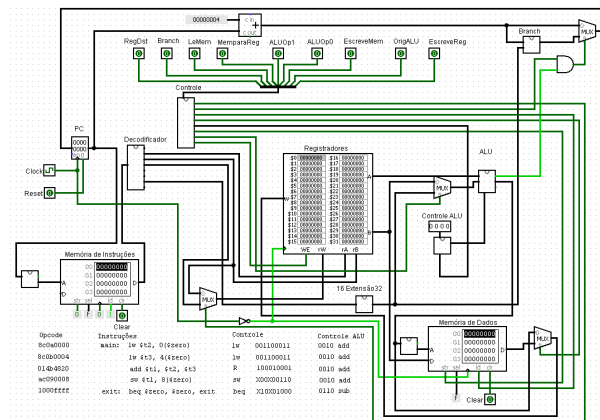


Figura 5: Circuito datapath com controle manual

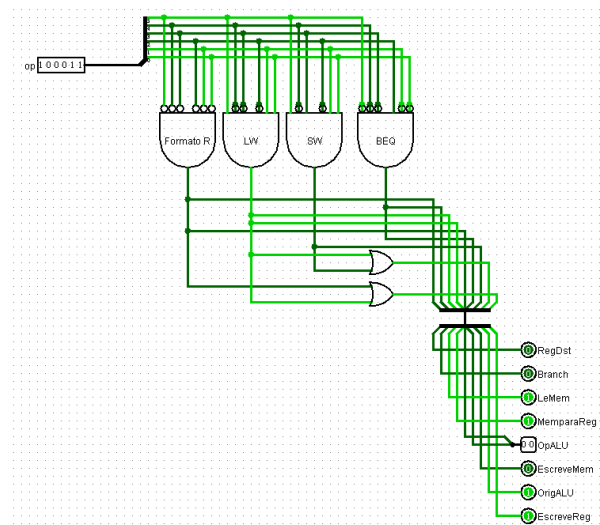


Figura 6: Sub Circuito de controle principal

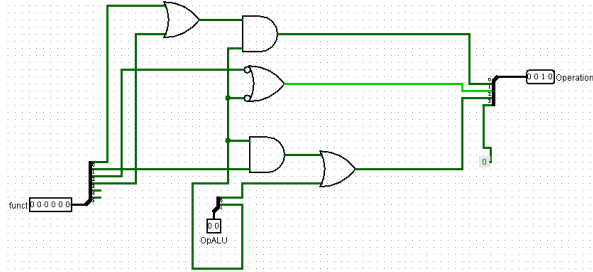


Figura 7: Sub circuito do controle da ULA

## 5. Conclusões

Neste trabalho, pôde-se entender o funcionamento e forma de implementação do datapath, controle principal e controle da ULA. Além disso, foi possível entender como programar o funcionamento de uma instrução a partir da implementação das funções jump e addi.

## 7. Referências

1. [Apêndice D - Mapeando o Controle no Hardware](#)  
Arquivo
2. <http://producao.virtual.ufpb.br/books/edusantana/introducao-a-arquiteturade-computadoreslivro/livro/livro.chunked/index.html>
3. [MAC0344 Arquitetura de computadores Uma visão geral \(usp.br\)](#)
4. [Aula02\\_03DPMIPS.PDF \(uminho.pt\)](#)
5. [\\_arquitet\\_computadores.indb \(ufsm.br\)](#)
6. [Introdução à Arquitetura de Computadores/O que é o MIPS? - Wikilivros \(wikibooks.org\)](#)