



**Universidade Federal De Roraima
Centro De Ciência e Tecnologia
Departamento de Ciência da Computação**



CURSO DE BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

PROCESSADOR RISC DE 8 BITS

Boa Vista/RR
2025

ACADÊMICO:

ABRAHÃO PICANÇO NERES DE OLIVEIRA

JONATHAN EMERSON BRAGA DA SILVA

PROCESSADOR RISC DE 8 BITS

Trabalho Avaliativo para matéria de
Arquitetura e Organização de
Computadores em Ciências da
Computação da Universidade de
Roraima - UFRR.

Orientador: Herbert Oliveira

Boa Vista/RR
2025

SUMÁRIO

1. INTRODUÇÃO.....	4
2. CONJUNTO DE INSTRUÇÕES (ISA).....	4
3. FORMATO DAS INSTRUÇÕES.....	4
3.1 TIPO R (REGISTRADOR).....	5
3.2 TIPO I (IMEDIATO).....	5
3.3. TIPO J (JUMP).....	6
4. REGISTRADORES.....	6
5. MODOS DE ENDEREÇAMENTO.....	6
6. TABELA DE INSTRUÇÕES.....	7
7. DATAPATH.....	7
8. UNIDADE DE CONTROLE.....	8
9. PROGRAMA DE EXEMPLO.....	8

1. INTRODUÇÃO

O projeto de um processador RISC de 8 bits exige um planejamento detalhado das instruções que serão suportadas, bem como a definição de sua arquitetura e funcionamento. Este documento descreve o planejamento das instruções, incluindo o conjunto de instruções, formatos, modos de endereçamento e a organização do datapath e da unidade de controle.

2. CONJUNTO DE INSTRUÇÕES (ISA)

O conjunto de instruções (ISA - *Instruction Set Architecture*) define as operações que o processador pode executar. Para este projeto, as instruções obrigatórias são:

- **Load (LD):** Carrega um valor da memória para um registrador.
- **Store (ST):** Armazena um valor de um registrador na memória.
- **Add (ADD):** Soma dois registradores.
- **Subtract (SUB):** Subtrai dois registradores.
- **Branch if Equal (BEQ):** Salta para um endereço se dois registradores forem iguais.
- **Jump(JMP):** Salto incondicional para um endereço.

3. FORMATO DAS INSTRUÇÕES

As instruções são codificadas em 8 bits e divididas em três formatos principais:

3.1 TIPO R (REGISTRADOR)

Utilizado para operações entre registradores.

Campo	Bits	Descrição
Opcode	2 bits	Código da operação (00 para tipo R)
rs	2 bits	Registrador fonte 1
rt	2 bits	Registrador fonte 2
rd	2 bits	Registrador destino.
funct	2 bits	Função específica (ex: 00 para ADD)

Exemplo: ADD R1, R2, R3 → 00 01 10 11 00

3.2 TIPO I (IMEDIATO)

Utilizado para operações com valores imediatos ou acesso à memória.

Campo	Bits	Descrição
Opcode	2 bits	Código da operação (01 para LD, 10 para BEQ)
rs	2 bits	Registrador fonte
rt	2 bits	Registrador destino
immediate	2 bits	Valor imediato ou deslocamento

Exemplo: LD R1, 0x05 → 01 01 00 05

3.3. TIPO J (JUMP)

Utilizado para saltos incondicionais.

Campo	Bits	Descrição
Opcode	2 bits	Código da operação (11 para JMP)
address	6 bits	Endereço de destino

4. REGISTRADORES

O processador utiliza os seguintes registradores:

- **Registradores de propósito geral:** R0, R1, R2, R3 (2 bits para endereçamento).
- **Program Counter (PC):** Aponta para a próxima instrução.
- **Registrador de status:** Armazena flags (ex: zero, carry).

5. MODOS DE ENDEREÇAMENTO

Os modos de endereçamento definem como os operandos são acessados:

- **Imediato:** O operando é um valor constante (ex: ADD R1, R2, 5).
- **Registrador:** O operando está em um registrador (ex: ADD R1, R2, R3).
- **Memória:** O operando está na memória (ex: LD R1, 0x05).

6. TABELA DE INSTRUÇÕES

A Tabela 1 descreve as instruções suportadas pelo processador.

Instrução	Formato	Opco de	rs	rt	rd	funct	immed iate	addre ss	Descri ção
ADD	R	00	rs	rt	rd	00	-	-	Soma rs e rt, armazena em rd
LD	I	01	rs	rt	-	-	immediate	-	Carrega memória[rs + immediate] em rt
BEQ	I	10	rs	rt	-	-	immediate	-	Salta se rs == rt
JMP	J	11	-	-	-	-	-	addresses	Salta para address

7. DATAPATH

O datapath do processador inclui os seguintes componentes:

- **Banco de registradores:** Armazena dados temporários.

- **ULA (Unidade Lógica e Aritmética):** Realiza operações como soma e subtração.
- **Memória de Instruções:** Armazena o programa.
- **Memória de dados:** Armazena dados.
- **PC (Program Counter):** Aponta para a próxima instrução.
- **Multiplexadores:** Selecionam entre diferentes entradas.

8. UNIDADE DE CONTROLE

A unidade de controle gera sinais para controlar o datapath com base na instrução atual. Os principais sinais são:

- **RegWrite:** Habilita escrita no banco de registradores.
- **ALUOp:** Define a operação da ULA.
- **MemRead/MemWrite:** Controla leitura/escrita na memória de dados.
- **Branch:** Ativa o salto condicional.
- **Jump:** Ativa o salto incondicional.

9. PROGRAMA DE EXEMPLO

Abaixo está um exemplo de programa em assembly:

```
LD R1, 0x05      ; Carrega o valor da memória[5] em R1
LD R2, 0x06      ; Carrega o valor da memória[6] em R2
ADD R3, R1, R2   ; Soma R1 e R2, armazena em R3
ST R3, 0x07      ; Armazena R3 na memória[7]
BEQ R3, R1, 0x04 ; Se R3 == R1, pula para a instrução 4
JMP 0x00         ; Volta para o início
```