**UNIVERSIDADE FEDERAL DE RORAIMA**

**CENTRO DE CIÊNCIA E TECNOLOGIA**

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

**DCC301– ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES– 2024**

**PROF. DR. HEBERT OLIVEIRA ROCHA**

**CLEILLYSON OSMAR SOUZA DINIZ DE ALMEIDA**

**KAYLON GUTIERRE PERES GONÇALVES**

**O PROCESSADOR DE 8 BITS**

**BOA VISTA, RR**

**2025**

**CLEILLYSON OSMAR SOUZA DINIZ DE ALMEIDA**

**KAYLON GUTIERRE PERES GONÇALVES**

**O PROCESSADOR DE 8 BITS**

Trabalho da disciplina de Arquitetura e Organização de Computadores do ano de 2024.2 apresentado à Universidade Federal de Roraima do curso de Bacharelado em ciência da computação.

Docente: Prof. Dr. Hebert O. Rocha

**BOA VISTA, RR**

**2025**

LISTA DE ABREVIATURAS E SIGLAS

OPCODE Código de operação

ULA Unidade Lógica aritmética

SUMÁRIO

[1. Especificações 7](#_Toc444681824)

[1.1. Plataforma de desenvolvimento 7](#_Toc444681825)

[1.2. Conjunto de instruções 7](#_Toc444681826)

[1.3. **Descrição de hardware** 8](#_Toc444681827)

[1.3.1. ULA 8](#_Toc444681828)

[1.1.1 BDRegister 8](#_Toc444681829)

[1.1.2 Clock 8](#_Toc444681830)

[1.1.3 Controle 8](#_Toc444681831)

[1.1.4 Memória de dados 9](#_Toc444681832)

[1.1.5 Memória de Instruções 9](#_Toc444681833)

[1.1.6 Somador 9](#_Toc444681834)

[1.1.7 And 10](#_Toc444681835)

[1.1.8 Mux\_2x1 10](#_Toc444681836)

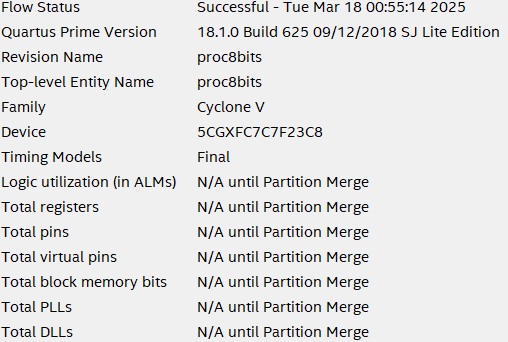
[1.1.9 PC 10](#_Toc444681837)

[1.1.10 ZERO 10](#_Toc444681838)

# Especificações

## Plataforma de desenvolvimento

Para implementação do processador foi usado IDE Quartus Prime Lite Edition, versão 18.1.0 e para simulação de ondas foi usado o modelsin, GTKwave e o GHDL .



## Conjunto de instruções

O processador possui 4 registradores: R0, R1, R2, R3. O registrador R0 é apenas para fazer operações BEQ, apresenta 3 tipos de instruções de 8 bits cada, tipo R que faz operações aritméticas, tipo I faz operações com valores da memoria e valores imediatos e J que são saltos incondicionais.

* Opcode: operação básica a ser executada.
* rs: O registrador contendo o primeiro operando-fonte.
* rd: O registrador contendo o segundo operando-fonte e também é o operador de destino.
* Imediato: é o endereço de memória.
* Endereço: é o endereço de destino.

Tipo R

|  |  |  |  |
| --- | --- | --- | --- |
| **Opcode** | **rs** | **rd** | **Funct** |
| 7 - 5 | 4 - 3 | 2 - 1 | 0 |
| 3 bits | 2 bits | 2 bits | 1 bit |

Tipo I

|  |  |  |
| --- | --- | --- |
| **Opcode** | **rs** | **Imediato** |
| 7 - 5 | 4 - 3 | 2 - 0 |
| 3 bits | 2 bits | 3 bits |

Tipo J

|  |  |  |
| --- | --- | --- |
| **Opcode** | **Endereço** | |
| 7 - 5 | 4 - 0 | |
| 3 bits | 5 bits | |

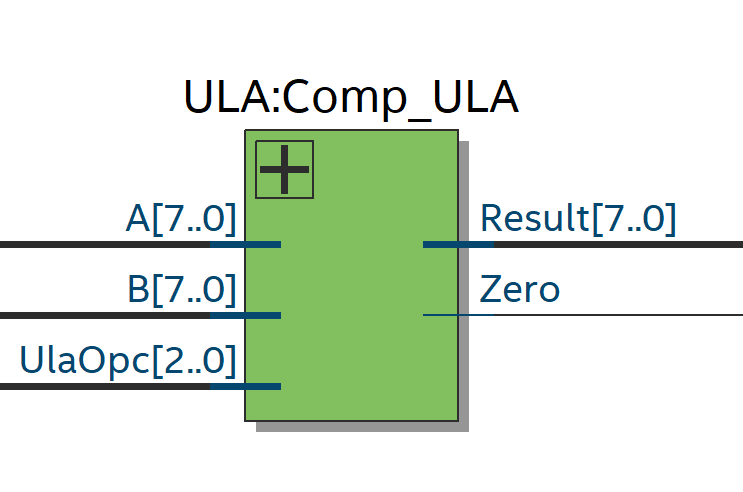
Lista de instruções

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instrução | *Opcode* | Tipo | Breve explicação | Exemplo |
| ADD | 000 | R | Soma | ADD R1, R2 (R2 <= R1 + R2) |
| ADDi | 001 | R | Adiciona imediato | ADD R1, 111 (R1 <= R1 + 7) |
| SUB | 010 | R | Subtrai | SUB R3, R2 (R2 <= R3 - R2) |
| LW | 011 | I | Carrega de endereço | LW R2, 011 (R2 <= Mem[3] ) |
| SW | 100 | I | Guarda no endereço | SW R1, 101 (Mem[5] <= R1) |
| BEQ | 101 | I | Salto condicional | BEQ R3, 111 (R3 == R0, Soma posição atual com -1) |
| JUMP | 110 | J | Salto incondicional | JUMP 01010 (Vai para o endereço 5) |

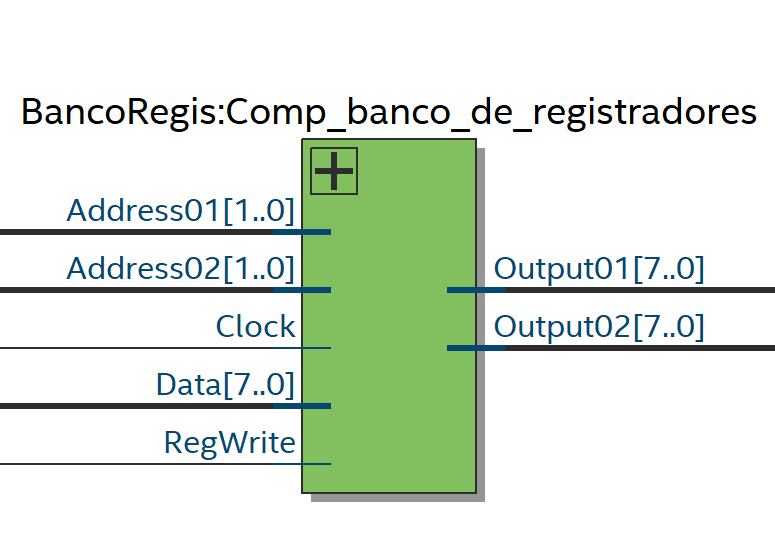
## **Descrição de hardware**

### ULA

A ULA tem como objetivo efetuar as operações aritméticas sendo elas soma, soma com imediato e subtração. Também usada na hora de fazer comparações com maior ou igual.



### BDRegister



### Unidade de Controle

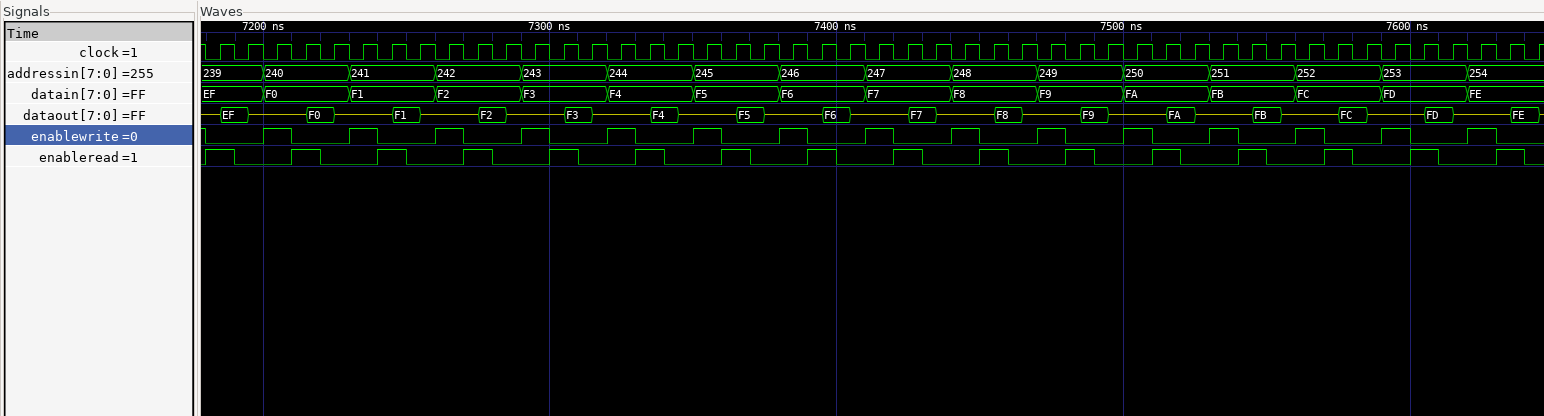
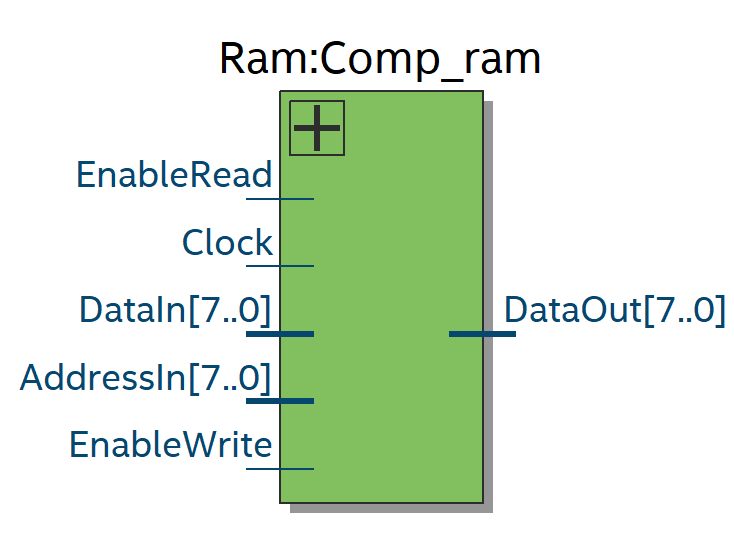
* **MemRead**: Leitura da memoria.
* **MemWrite**: Escrita na memoria.
* **MemToReg**: Memoria para registrador.
* **ResWrite**: Se vai ser escrito em registrador
* **UlaOpc**: Codigo de operação da ULA.
* **UlaSrc**: Soma imediata.
* **Jump**: Pulo sem condição.
* **Branch**: Comparação.

Abaixo segue a tabela, onde é feita a associação entre os opcodes e as flags de controle:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Comando | MemRead | MemWrite | MemToReg | ResWrite | UlaOpc | UlaSrc | Jump | Branch |
| ADD | 0 | 0 | 0 | 1 | 00 | 0 | 0 | 0 |
| ADDi | 0 | 0 | 0 | 1 | 01 | 1 | 0 | 0 |
| SUB | 0 | 0 | 0 | 1 | 11 | 0 | 0 | 0 |
| LW | 1 | 0 | 1 | 1 | 00 | 0 | 0 | 0 |
| SW | 0 | 1 | 0 | 0 | 00 | 0 | 0 | 0 |
| BEQ | 0 | 0 | 0 | 0 | 01 | 0 | 0 | 1 |
| JUMP | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |

Tabela 2 - Detalhes das flags de controle do processador.

### Memória de RAM



### PC

**[Todo] Descrição**

**CONCLUSÃO**

**REFERÊNCIAS**

**SOBRENOME, Nome do autor (se houver). Título da página. Nome do site, data de publicação (ou atualização). Disponível em: <URL>. Acesso em: dia mês ano.**

**SOBRENOME, Nome do autor. Título do livro: subtítulo (se houver). Edição (se não for a primeira). Local de publicação: Editora, ano de publicação.**