**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 ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

OPCODE Código de operação

ULA Unidade Lógica aritmética

SUMÁRIO

[1. Especificações 7](#_Toc1)

[1.1. Plataforma de desenvolvimento 7](#_Toc2)

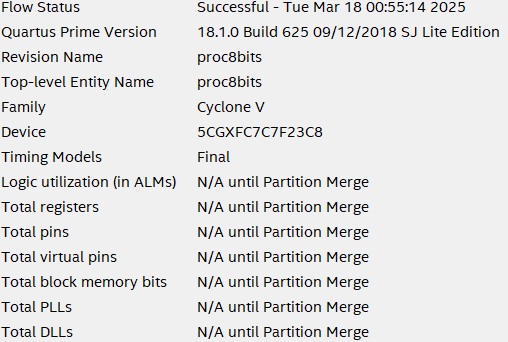
[1.2. Conjunto de instruções 7](#_Toc3)

[2. Conjunto de Instruções 7](#_Toc4)

# Especificações

## Plataforma de desenvolvimento

Para implementação do processador foi usado IDE Quartus Prime Lite Edition, versão 18.1.0 e ModelSin.



## 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 numero ali colocado.
* Endereço: é o endereço de destino.

Tipo R

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

Tipo I

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

Tipo J

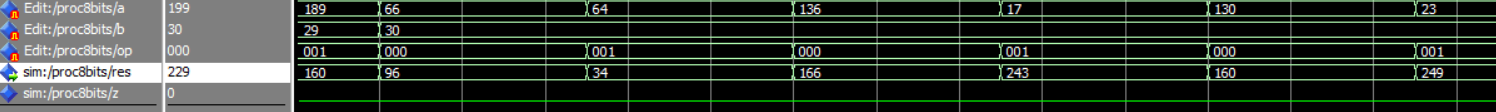
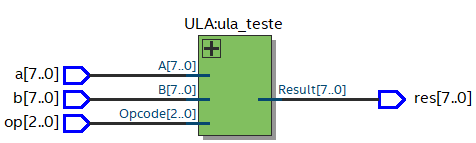
|  |  |  |
| --- | --- | --- |
| **Opcode** | **rs** | **rd** |
| 7 - 5 | 4 - 3 | 2 - 1 |
| 3 bits | 2 bits | 2 bits |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Instrução | *Opcode* | Tipo | Breve explicação | Exemplo |
| ADD | 000 | R | Soma | ADD R1, R2 (R2 <= R1 + R2) |
| SUB | 001 | R | Subtrai | SUB R3, R2 (R2 <= R3 - R2) |
| LW | 010 | I | Carrega palavra | LW R2, 011 (R2 <= Mem[3] ) |
| SW | 011 | I | Guarda palavra | SW R1, 101 (Mem[5] <= R1 |
| BEQ | 100 | I | Salto condicional | BEQ R3, 111 (R3 == R0, Pula para endereço 7) |
| JUMP | 101 | J | Salto incondicional | JUMP 1010 (Pula para instrução 10) |

## **Descrição de hardware**

### ULA

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



**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.**