

**PODER EXECUTIVO**

**MINISTÉRIO DA EDUCAÇÃO**

**UNIVERSIDADE FEDERAL DE RORAIMA**

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

**ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES**

**RELATÓRIO DO PROJETO: PROCESSADOR Fluminense**

**ALUNOS:**

**Luiz Gustavo Dall’Agnol Cavalcante – 2021000632**

**Felipe de Sousa Borges – 20200020120**

**Yan Alberto Siqueira - 2020005518**

**Novembro de 202****3**

**Boa Vista/Roraima**



**PODER EXECUTIVO**

**MINISTÉRIO DA EDUCAÇÃO**

**UNIVERSIDADE FEDERAL DE RORAIMA**

**DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO**

**ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES**

**RELATÓRIO DO PROJETO: PROCESSADOR Fluminense**

**Novembro de 202****3**

**Boa Vista/Roraima**

**Resumo**

Este trabalho aborda o projeto e implementação do processador Fluminense, que possui 4 registradores de 16 bits, construído no Logisim, sendo composto de quatro componentes principais: ULA, Unidade de Controle, Banco de Registradores e CPU. Fluminense suporta 15 instruções, sendo elas: OR, NOT, AND, XOR, ADD, SUB, MULT, DIV, MOV, MOVImm, LOAD, STORE, JMPZ, JMPN, JMPP e HALT. O processador suporta apenas instruções com dois registradores por vez, com o opcode de instruções sendo introduzidas apenas por input hexadecimal.

**Conteúdo**

[1 Especificação 6](#_Toc444681789)

[1.1 Plataforma de desenvolvimento 6](#_Toc444681790)

[1.2 Conjunto de instruções 6](#_Toc444681791)

[1.3 Descrição do Hardware 7](#_Toc444681792)

[1.3.1 ULA 7](#_Toc444681793)

[1.3.2 Banco de registradores 7](#_Toc444681794)

[1.3.3 Unidade de Controle 8](#_Toc444681795)

[1.4 Datapath 9](#_Toc444681804)

[2 Simulações e Testes 10](#_Toc444681805)

[3 Considerações finais 11](#_Toc444681806)

# Especificação

## Plataforma de desenvolvimento

Para a implementação do processador Fluminense foi utilizado o aplicativo: Logisim

## Conjunto de instruções

O processador Fluminense possui 4 registradores: RegAX, RegBX, RegCX e RegDX. Assim como formato hexadecimal de instrução de 4 bits, Instruções do **tipo R, I e J** seguem algumas considerações sobre as estruturas contidas nas instruções:

* **Opcode**: a operação básica a ser executada pelo processador, tradicionalmente chamado de código de operação, em fluminense ele existe em 4 bits no formato hexadecimal.
* **R1**: o registrador contendo o primeiro operando fonte e adicionalmente para alguns tipos de instruções (ex. instruções do tipo R) é o registrador de destino;
* **R2**: o registrador contendo o segundo operando fonte;
* **Endereço**: Um enderenço onde é armazenado os dados das operações;

Tipos de Instruções:

**Fluminense** só trabalha com dois registradores por vez, com a ordem de suas instruções sendo: Instrução (ex: ADD), R1, R2 e instrução imediata, no caso o valor a ser somado (neste relatório damos que esse valor é seis como exemplo).

**Visão geral das instruções do Processador Fluminense:**

Tabela 1 – Tabela que mostra a lista de Opcodes utilizadas pelo processador Fluminense.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ordem** | **Nome** | **Opcode em binário** | **Instrução completa em Hexadecimal** |
| 0 | OR | 0000 | 0000 00 01 00000110 |
| 1 | NOT | 0001 | 0001 00 01 00000110 |
| 2 | AND | 0010 | 0010 00 01 00000110 |
| 3 | XOR | 0011 | 0011 00 01 00000110 |
| 4 | ADD | 0100 | 0100 00 01 00000110 |
| 5 | SUB | 0101 | 0101 00 01 00000110 |
| 6 | MULT | 0110 | 0110 00 01 00000110 |
| 7 | DIV | 0111 | 0111 00 01 00000110 |
| 8 | MOV | 1000 | 1000 00 01 00000110 |
| 9 | MOVIm | 1001 | 1001 00 01 00000110 |
| 10 | LOAD | 1010 | 1010 00 01 00000110 |
| 11 | STORE | 1011 | 1011 00 01 00000110 |
| 12 | JMPZ | 1100 | 1100 00 01 00000110 |
| 13 | JMPN | 1101 | 1101 00 01 00000110 |
| 14 | JMPP | 1110 | 1110 00 01 00000110 |
| 15 | HALT | 1111 | 1111 00 01 00000110 |

## Descrição do Hardware

### ULA

O componente ULA (Unidade Lógica Aritmética) tem como principal objetivo efetuar as principais operações aritméticas, dentre elas: soma e subtração. A ULA em Fluminense recebe dois OPcodes de 16 bits, que são então processados dependendo do opcode inserido, para então passar pelo multiplexador e acabar com o output armazenado.

Diagrama, Esquemático

Descrição gerada automaticamente

### Banco de registradores

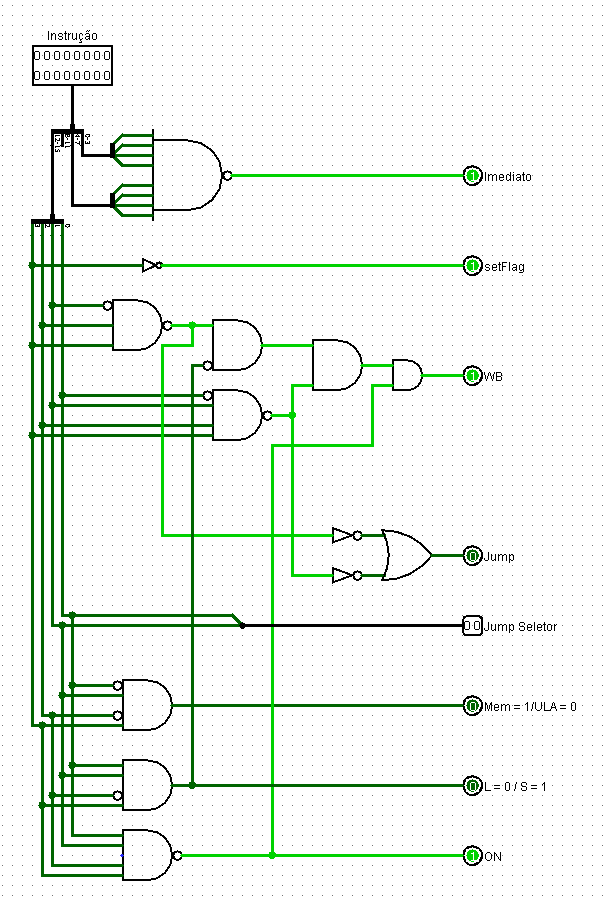
O componente Banco de Registradores tem como principal objetivo armazenar valores para uso imediato. Neste componente os valores adicionados aos registradores são armazenados e então são postos a mostra na CPU.

Diagrama, Esquemático

Descrição gerada automaticamente

### Unidade de Controle

O componente Unidade de Controle tem como objetivo realizar o controle de todos os componentes do processador de acordo com o opcode.



## Datapath

É a conexão entre as unidades funcionais formando um único caminho de dados e acrescentando uma unidade de controle responsável pelo gerenciamento das ações que serão realizadas para diferentes classes de instruções...

(Processador principal)

Diagrama, Esquemático

Descrição gerada automaticamente

# Simulações e Testes

Objetivando analisar e verificar o funcionamento do processador, efetuamos alguns testes analisando cada componente do processador em específico, em seguida efetuamos testes de cada instrução que o processador implementa. Para demonstrar o funcionamento do processador Fluminense utilizaremos como exemplo o código que guarda valores nos registradores, insere-os na memória RAM.

Neste exemplo tentamos realizar uma soma (ADD), onde adicionamos o valor de 6 presentes no RegBX com o valor presente no RegAX.

Formato para escrita em código binário:

|  |  |  |  |
| --- | --- | --- | --- |
| 4 bits | 2 bits | 2 bits | 8 bits |
| 0100 | 00 | 01 | 00000110 |
| ADD | RegAX | RegBX | 6 |

Formato para inserção no OPCode hexadecimal:

0100000100000110 -> 4106

# Considerações finais

Este trabalho apresentou o projeto e implementação do processador de 16 bits denominado de Fluminense. O grupo chegou à conclusão que o objetivo de aprendizado íntimo sobre a construção e funcionamento de um processador, fora alcançando plenamente. Graças a metodologia prática do projeto e avisos prévios da complexidade do trabalho, podemos realizar a atividade com paciência e aprender organicamente.

Apesar da dificuldade inicial de conectar os conteúdos aprendidos em sala de aula com a realidade prática do trabalho, concluímos que o tempo alocado fora suficiente para realizá-lo sem pressa ou aperto. Porém foram altas as dificuldades de reunião do grupo, por agendas conflituosas, resultando em um projeto com menos tempo de trabalho que o esperado.