



# Visão geral

## Especificações do Projeto

Para a implementação do projeto, foi fornecido um circuito de um processador base, conforme ilustrado na Figura 01. O processador fornecido é capaz de realizar apenas duas instruções: subtração (sub) e adição (add), ambas com instruções de 32 bits e seguindo o formato descrito abaixo:

OPCODE (31-28)	RD (27-24)	RS (23-20)	RT (19-16)	N/A (15-0)
-------------------	---------------	---------------	---------------	---------------

1. OPCODE: Indica a instrução a ser executada; o valor "0" representa a operação de adição, enquanto o valor "1" representa a operação de subtração.
2. RD: Indica o registrador de destino, onde o resultado da operação será armazenado.
3. RS: Indica um dos registradores de origem, sendo o minuendo no caso da subtração e a parcela no caso da adição.
4. RT: Indica outro registrador de origem, sendo o subtraendo no caso da subtração e a segunda parcela no caso da adição.
5. N/A: Não é utilizado.

Cada equipe é responsável por incluir novas instruções, especificar o formato delas e modificar o processador para que seja capaz de executar as novas instruções. Essas instruções adicionais devem ser escolhidas de forma a serem suficientes para implementar o algoritmo de multiplicação de matrizes.

## 1. BANCO DE REGISTRADORES.

O Banco de Registradores do Processador Base é composto por 4 registradores, como mostrado na Figura 02, mas cada equipe pode incrementar essa quantidade de registradores.

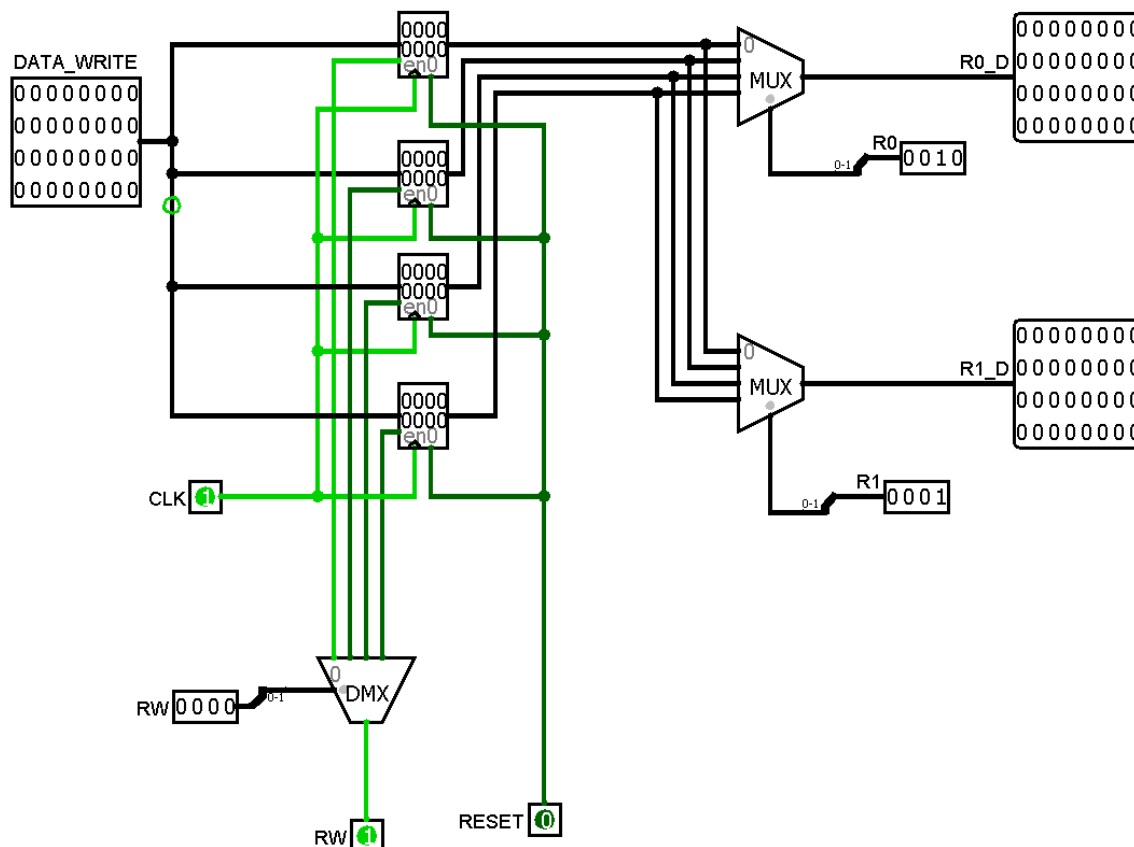


Figura 02: Banco de Registradores do Processador Base

## 2. UNIDADE DE CONTROLE

A Unidade de Controle desempenha um papel crucial na execução das instruções do processador. Ela recebe os primeiros 4 bits da instrução, conhecidos como campo "opcode", e utiliza essas informações para gerar os sinais de controle necessários para a execução da instrução. Na Figura 03, é apresentada a Unidade de Controle do processador base.

É importante ressaltar que o processador base atualmente implementa apenas as operações de adição (código 0000) e subtração (código 0001). Portanto, a Unidade de Controle existente só precisa gerar os sinais de controle `ula_op`, que indicam à unidade lógico-aritmética (ULA) qual operação realizar, e `write_reg`, que especifica se haverá ou não escrita no banco de registradores.

DICA: A Unidade de Controle é um circuito combinacional, o que significa que sua funcionalidade pode ser implementada automaticamente no Logisim. Isso pode ser realizado acessando a funcionalidade de análise de circuitos no Logisim através do caminho: Projeto -> Analisar Circuito -> Tabela. Isso permitirá uma compreensão mais aprofundada do funcionamento da Unidade de Controle.

### 3. ULA.

A ULA recebe dois valores de entrada, `DATA_0` e `DATA_01`, e um sinal de controle, `ULA_OP`, e produz uma saída, `ULA_OUT`.

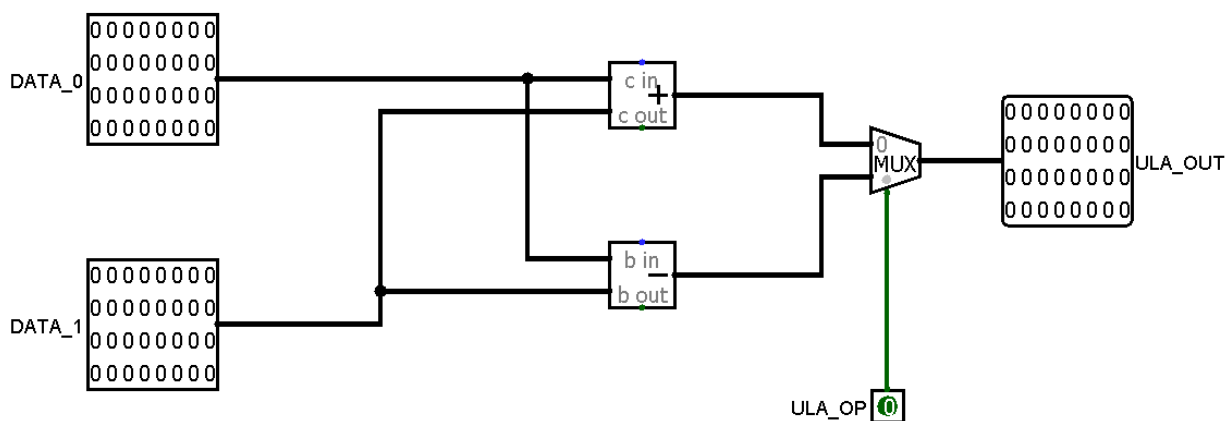


Figura 03: ULA Processador Base

### 4. MEMÓRIAS

As memórias, tanto as de instrução quanto as de dados, podem ser implementadas utilizando a memória RAM do Logisim. Estas memórias podem ser inicializadas por meio de um arquivo de texto, conforme exemplificado na Figura 04.

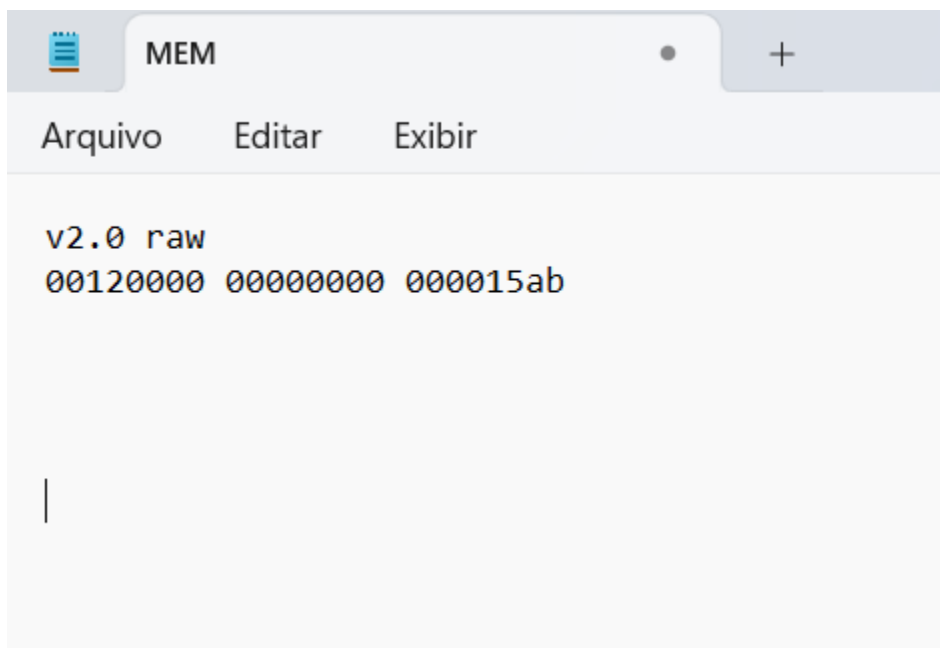


Figura 04: Arquivo de inicialização de memória

A primeira linha, "v2.0 raw", é um cabeçalho obrigatório. Na linha seguinte, cada instrução é separada por um espaço. Cada instrução consiste em 32 bits, e no arquivo de inicialização, cada instrução é representada por 8 números em hexadecimal, totalizando os 32 bits. Como exemplificado na Figura x, a primeira instrução "00120000" em binário seria:

**0000 0000 0001 0010 0000 0000 0000 0000**, que significa: *add R0, R1, R2*

Para inicializar a memória, basta clicar com o botão direito na memória e selecionar a opção "Carregar Imagem". Isso permitirá o carregamento das instruções e dos dados do programa a ser executado.

## Dicas de Implementação

- DIVIDA OS CAMPOS DAS INSTRUÇÕES DE 4 EM 4 BITS, ISSO FACILITARÁ NA HORA DE ESCREVER A INSTRUÇÃO EM HEXADECIMAL.
- VOCÊ PODE UTILIZAR TODOS OS RECURSOS DO LOGISIM.
- NÃO DEIXE PARA ÚLTIMA HORA, O TRABALHO É DESAFIADOR.

## Entrega e Avaliação

- O trabalho poderá ser feito em dupla, mas a avaliação é **INDIVIDUAL**, ou seja, **TODOS** os integrantes **DEVEM** saber de todo o trabalho;
- Os que optarem por fazer o trabalho, a prova valerá 6 e o trabalho 6.
- A nota da prova **interfere** na nota do trabalho. Não faz sentido errar coisas básicas na prova, se tiver feito o trabalho.
- O grupo deverá apresentar o trabalho ao professor da disciplina semanalmente que, no ato, fará perguntas a todos os integrantes.
- Além da apresentação, o grupo deverá entregar um relatório explicando o funcionamento do circuito e os arquivos do projeto. Além disso, os integrantes deverão, de forma sigilosa, mandar uma mensagem privada ao professor informando o percentual de participação de cada componente do grupo.
- A data da entrega do projeto será divulgada no google sala de aula.
- **EM CASO DE PLÁGIO OU QUALQUER FRAUDE, TODOS OS GRUPOS ENVOLVIDOS TERÃO A NOTA ZERADA!!**