

RELATÓRIO

Arquitetura de Computadores

Everton Lohan Pereira Ferreira
Arthur Ricardo Macêdo Pereira

Processador **MATIPS-BCL**

(Microprocessor with Absurd Tricks without Interlocked Pipeline Stages By Computer Legends)

Inicialmente, tivemos como base o processador MIPS e fomos estudar seus componentes. Nesse projeto não dividimos partes para cada um, mas sim fizemos tudo juntos.

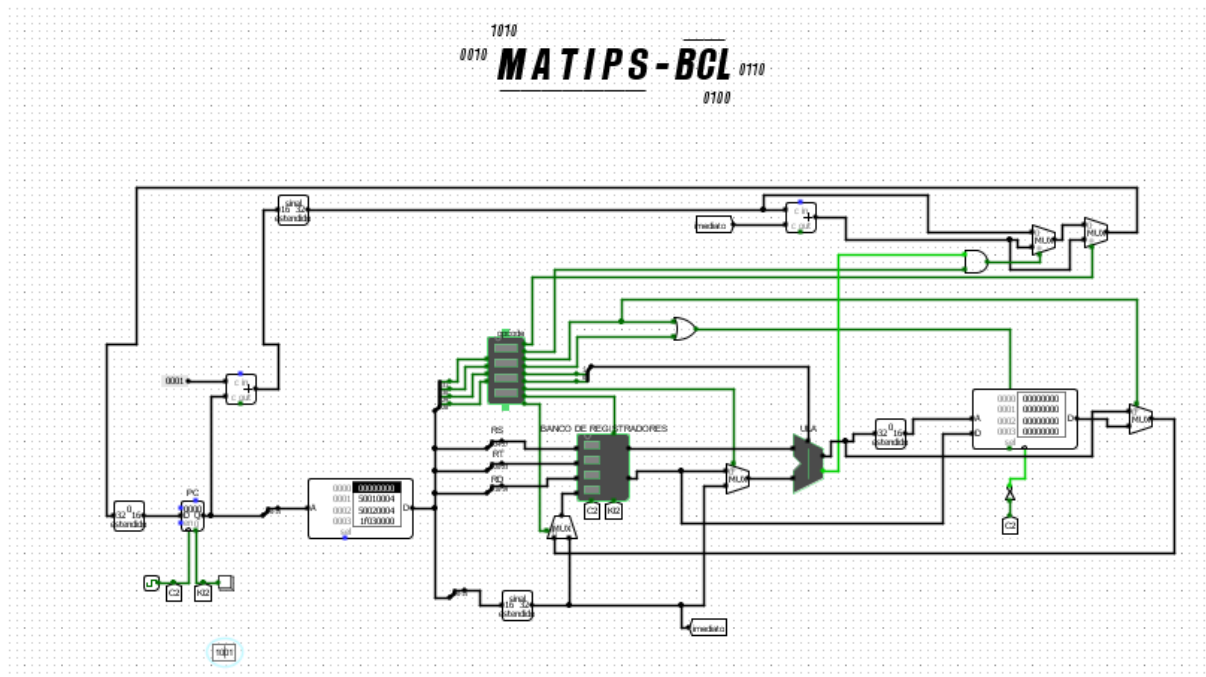
Primeiramente, tivemos que fazer o código de multiplicação de matrizes em assembly MIPS, para saber quais funções nós iríamos usar no código de multiplicação de matrizes. Como o professor pediu para fazermos uma multiplicação de matrizes 4X4 foram necessárias as seguintes funções:

- LI
- LW
- ADD
- ADDI
- SW
- MULT
- BEQ
- SUB
- JUMP

Após sabermos quais instruções vamos usar, tivemos que definir os tipos de instruções para nos diferenciar do MIPS, assim definimos o seguinte tipo:

OPCODE	RS	RT	RD	IMEDIATO
4 BITS	4 BITS	4 BITS	4BITS	16BITS

Definido isso, fomos montar o circuito no logisim, abaixo está o circuito



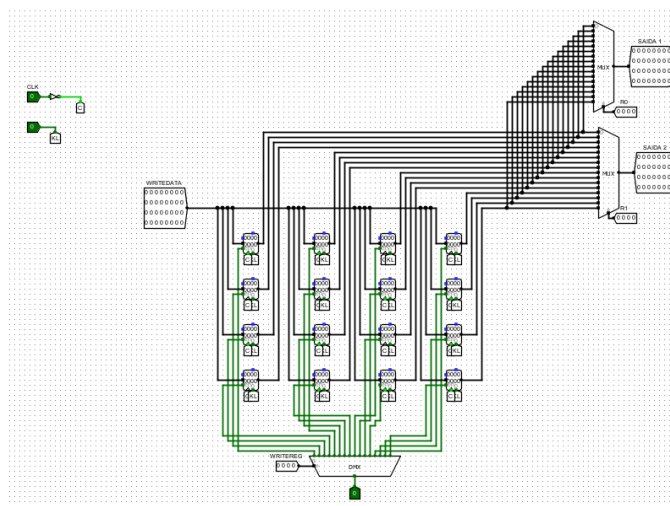
O circuito é composto por um PC (program counter), por uma memória de instruções, um banco de registradores, uma ULA, uma unidade de controle e por uma memória de dados.

- **Memória de Instrução:**

- A memória de instrução nós usamos a memória ROM disponibilizada pelo próprio Logisim.

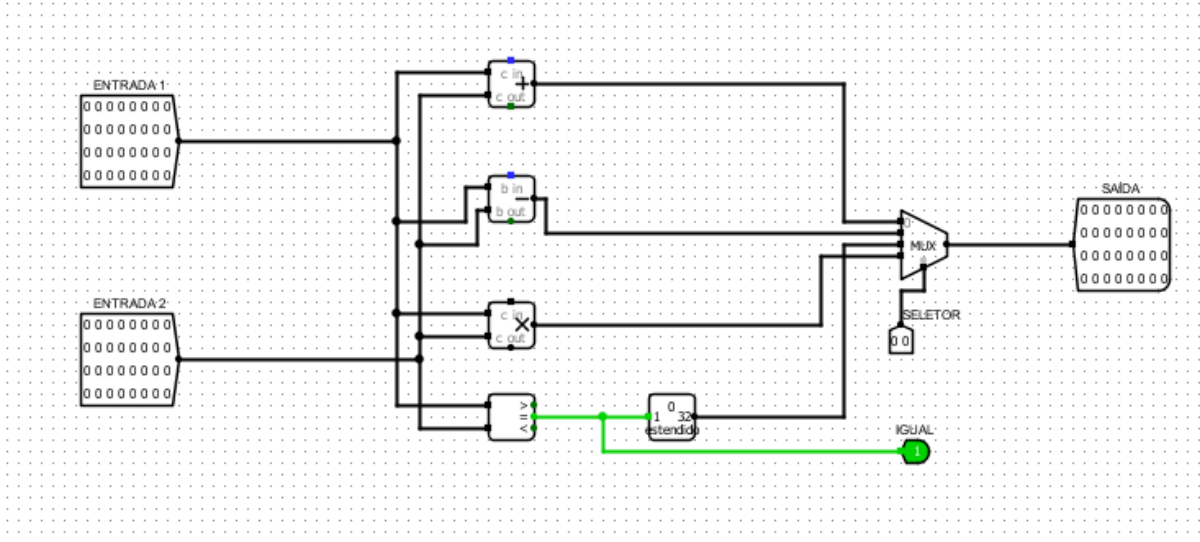
- **Banco de Registradores:**

- O banco de registradores foi feito com 16 registradores (0-15) com 32 bits de entrada em cada um, existem 4 entradas RS(primeiro registrador), RT(segundo registrador), RD(registrador destino) e write data (valor que queremos escrever em algum registrador).



- **ULA (Unidade Lógica Aritmética):**

- A ula possui 4 operações básicas, soma, subtração, multiplicação e comparação.
- Tivemos também de fazer os códigos das operações: 00 - adição, 01 - subtração, 10 - comparação, 11- multiplicação.



- **Unidade de Controle:**

- A unidade de controle é uma tabela verdade na qual divide todas as funções, não vamos mostrar o circuito combinacional pois o próprio logisim faz para nós, vamos somente mostrar a tabela verdade:

	INST R	JUM P	BRAN CH	MEM READ /REG	MEM WRIT E	ALUO P0	ALUO P1	ALUS RC	REG WRIT E	DATA SRC
ADD	0001	0	0	0	0	0	0	0	1	0
SUB	0010	0	0	0	0	0	1	0	1	0
ADDI	0011	0	0	0	0	0	0	1	1	0
LW	0100	0	0	1	0	0	0	1	1	0
LI	0101	0	0	0	1	0	0	1	1	1
MULT	0110	0	0	0	0	1	1	0	1	0
SW	0111	0	0	0	1	0	0	1	0	0
BEQ	1000	0	1	0	0	0	0	0	0	0
JUMP	1001	1	0	0	0	0	0	1	0	0

- **Memória de Dados**

- Para a memória de dados nós usamos a memória RAM disponibilizada pelo próprio logisim.

Feito tudo isso, nosso próximo trabalho foi transformar o código em Assembly MIPS para o nosso processador, ou seja, em Assembly MATIPS-BCL. Para isso, fizemos algumas coisas que nos auxiliam, como nomear os registradores para facilitar a conversão.

Nome do Registrador

Número que ele está no circuito

ZERO	0
S0	1
S1	2
S2	3
S3	4
S4	5
S5	6
S6	7
T0	8
T1	9
T2	A (11)
T3	B (12)
T4	C (13)
T5	D (14)
T6	E (15)
SP	F (16)