

**Prof. Dr. Maximilian Luppe**

Figura 1. MIPS 32 bits multiciclo

Cada componente que foi desenvolvido deve ser adicionado ao esquemático para o correto funcionamento do processador, abaixo será detalhado como deve ocorrer a conexão de cada componente.

## ULA

À direita do diagrama, encontra-se o slot reservado para a ULA conforme mostra a figura 2. As linhas **ALU\_INPUT\_A** e **ALU\_INPUT\_B** são reservadas para as entradas A e B respectivamente. **ALUSel** é o terminal para conexão do seletor (chamado de F na prática 1). **ALUResult** é a saída (S) da ULA e **Zero** é a linha de conexão da flag de zero.

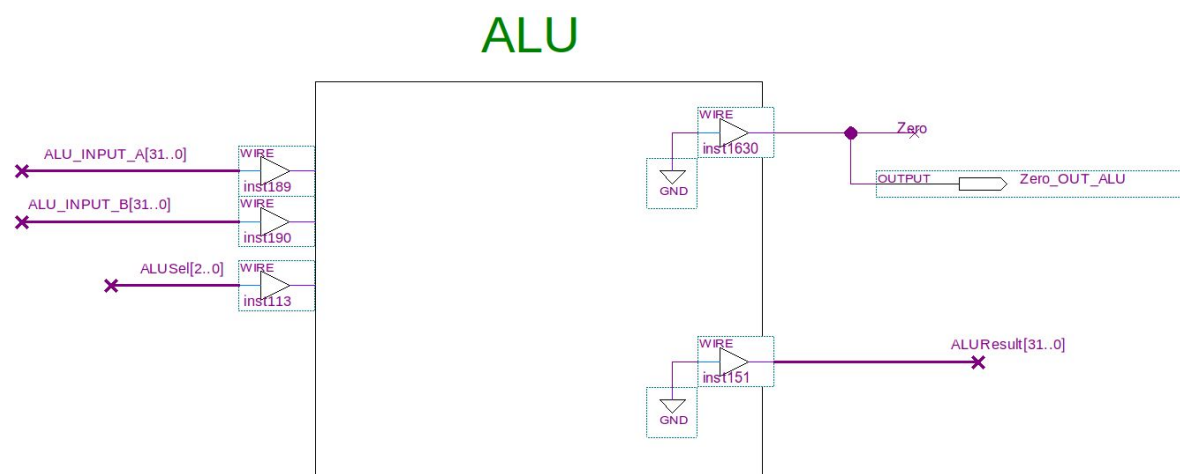


Figura 2.

## REGISTRADORES

Existem vários slots para conexão de registradores ao longo do esquemático. Começando da esquerda há o slot do PC (figura 3). O PC, como dito no relatório da prática 2 deve ser implementado como um contador com carga paralela.

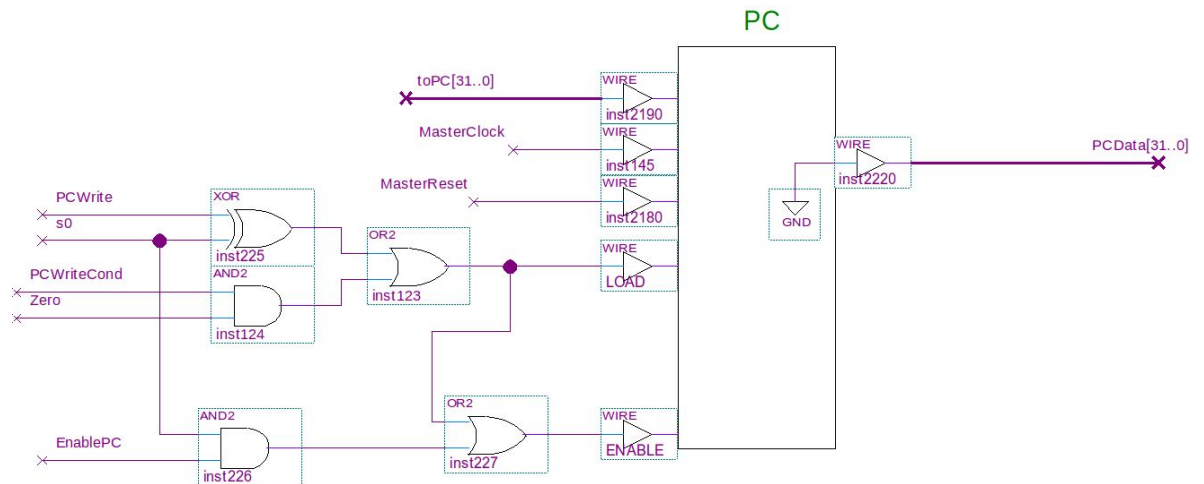


Figura 3.

A linha **toPC** deve ser utilizada para conectar os bits para carga paralela no contador. **MasterClock** e **MasterReset** são as linhas de clock e reset respectivamente. Abaixo, a conexão marcada como **LOAD** deve ser utilizada para habilitar escrita paralela dos dados em toPC (lembrando que a carga é síncrona) e em seguida a linha marcada como **ENABLE** é a habilitação do clock do contador. A contagem (saída) deve ser ligada em **PCData**.

O próximo registrador é o IR. O registrador IR (figura 4) deve ser ligado da seguinte maneira. A entrada de dados do registrador ligada em **Instruction**, saída de dados em **IRInstruction**, novamente o clock e o reset devem ser ligados em **MasterClock** e **MasterReset**. Além disso, o IR possui um sinal de controle **IRWrite**, que deverá ser ligado no enable do registrador.

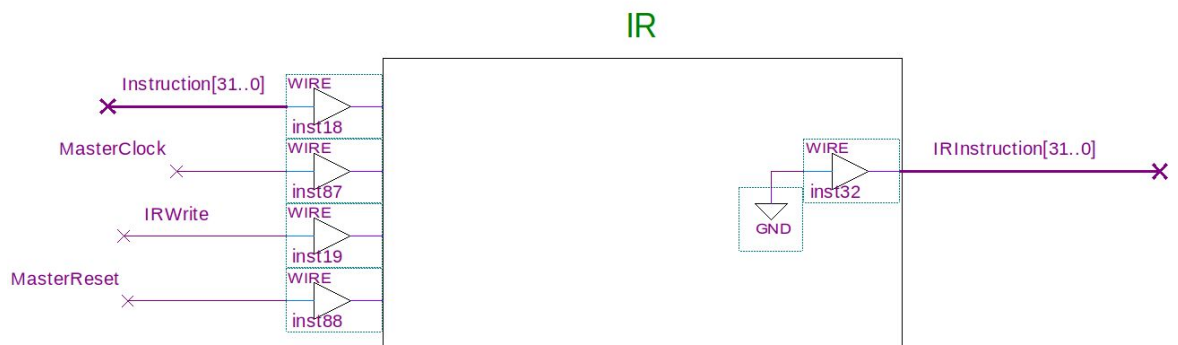


Figura 4.

Os registradores MDR, A, B e ALUOut são bastante parecidos. Possuem, como os demais, o clock e o reset ligados em **MasterClock** e **MasterReset**, entrada de dados à esquerda e saída de dados à direita. Não há entrada para enable para estes registradores.

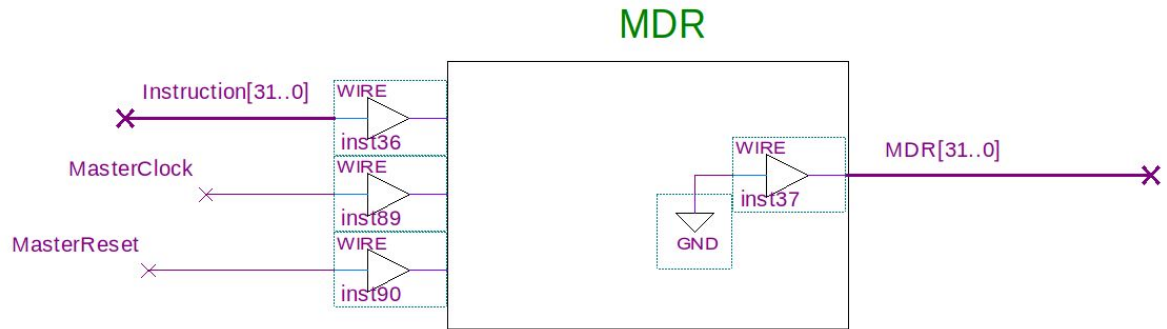


Figura 5.

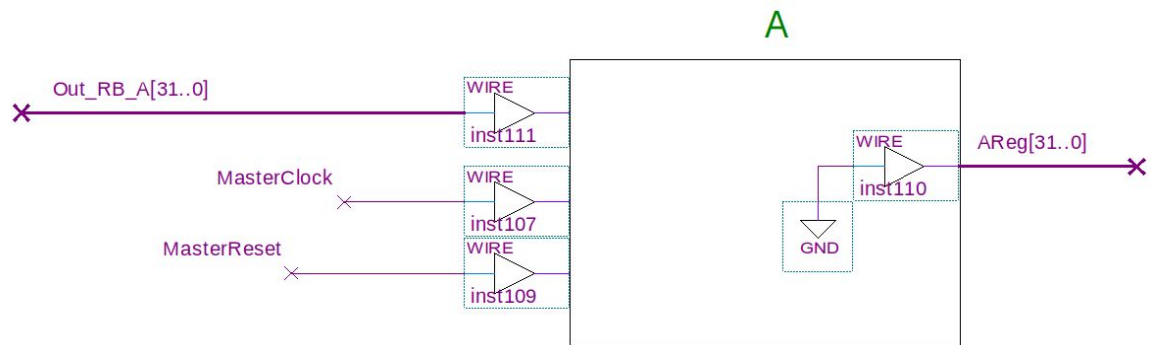


Figura 6.

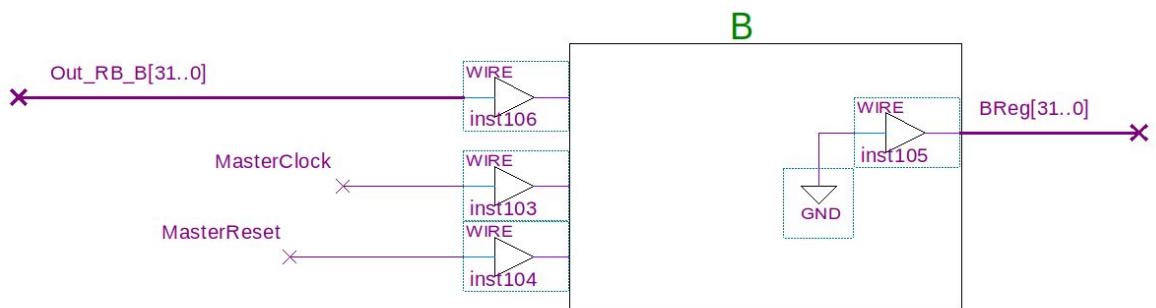


Figura 7.

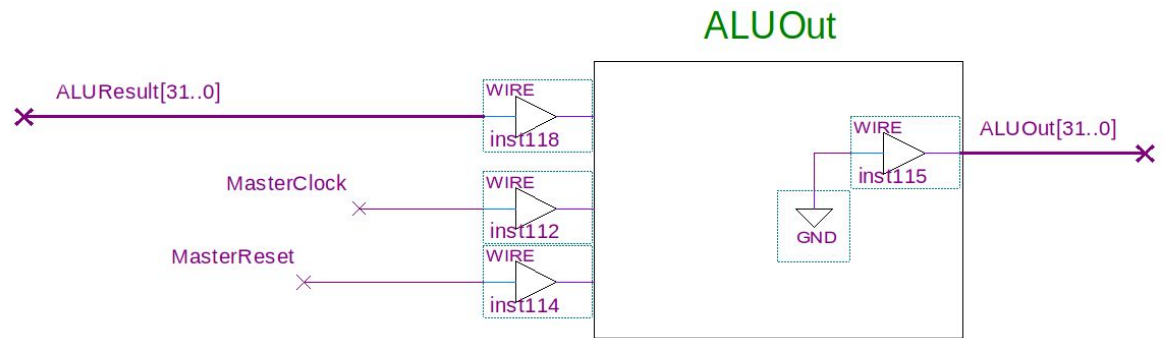


Figura 8.

Por fim, no centro do diagrama está localizado o slot do banco de registradores. As conexões de saída são **Out\_RB\_A** e **Out\_RB\_B**; entrada de dados deve ser ligada à **writeDataR**; **readA** e **readB** são os seletores de registrador para a leitura; **WriteSelect** o seletor de escrita no banco; **RegWrite** o enable para escrita e novamente as linhas **MasterClock** e **MasterReset**. Existe ainda uma conexão para o Set do banco de registradores, mas como esta função não será utilizada a entrada está aterrada (se não foi implementado um set, deixe a linha aberta).

## REGISTER BANK

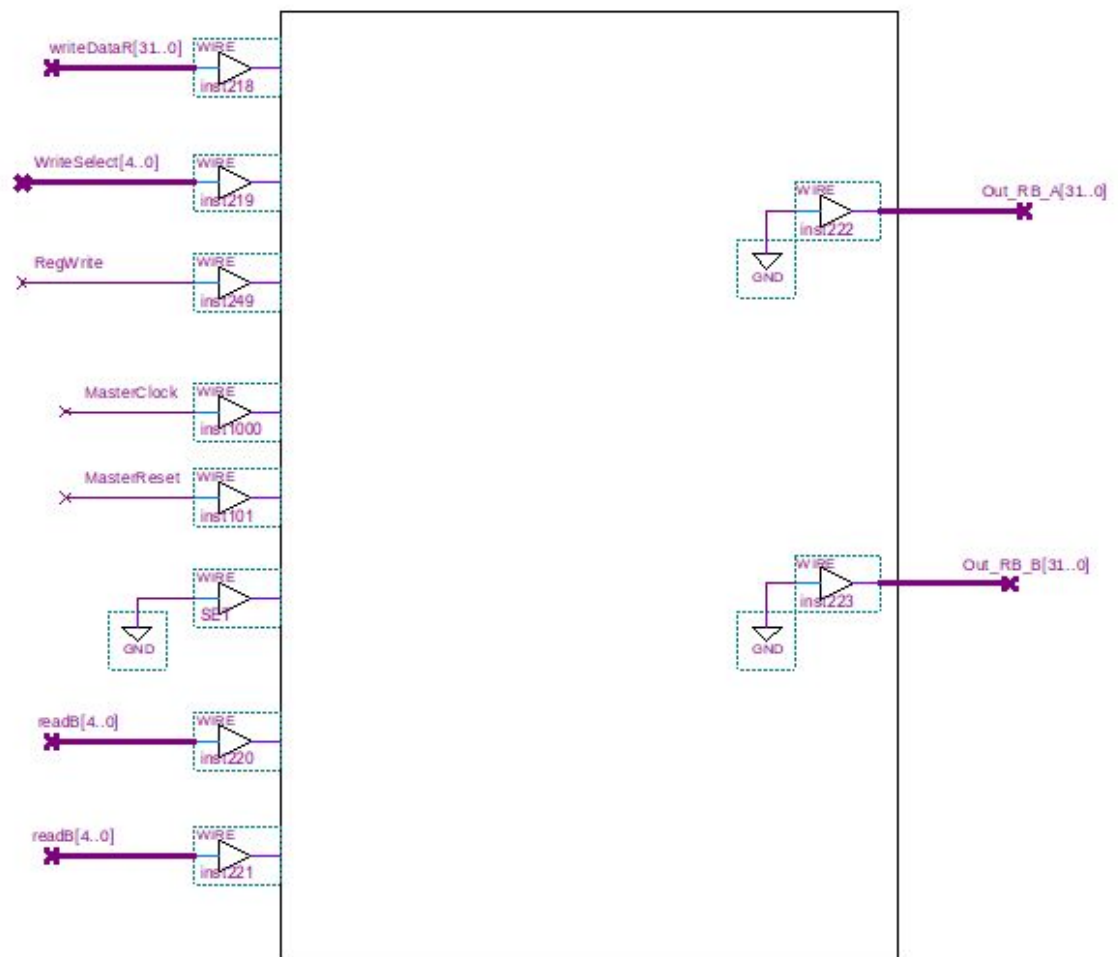


Figura 9.

### UNIDADE DE CONTROLE

O último componente a se conectar é a Unidade de Controle. A conexão da UC é também bastante simples. **SplitedInstruction** será a entrada, **MasterClock** e **MasterReset** novamente para clock e reset da UC e cada sinal de controle possui sua saída à direita.

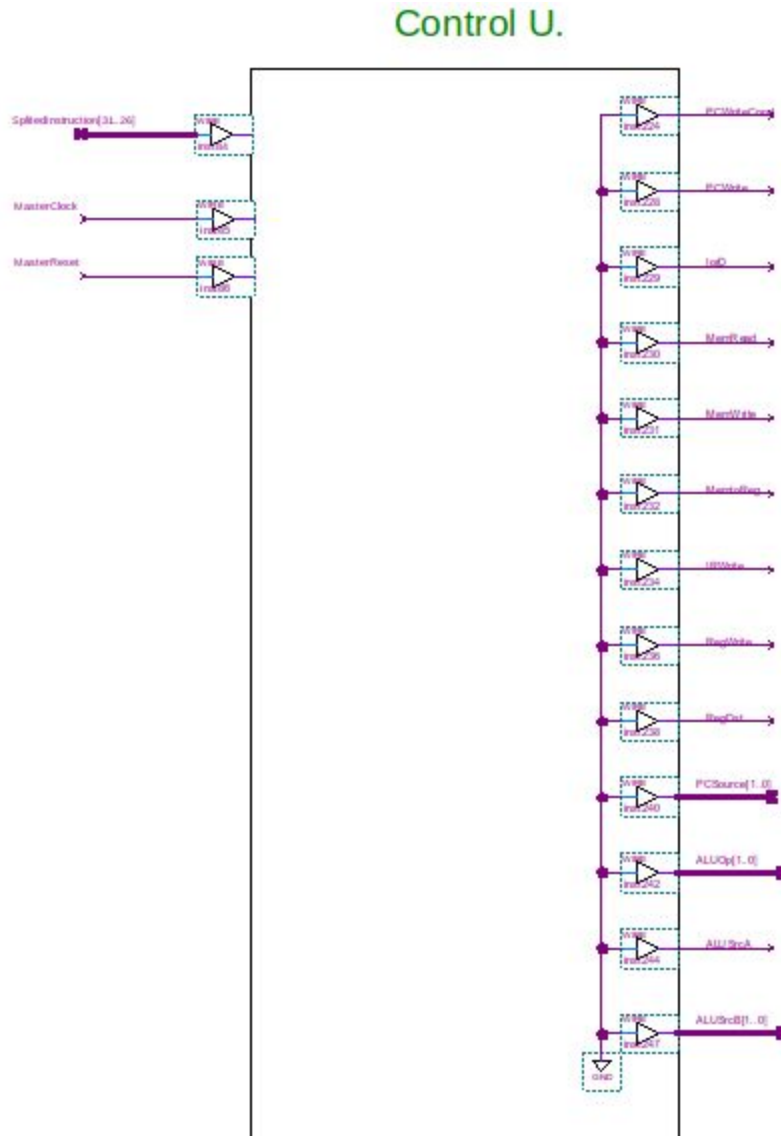


Figura 10.

### Problema

Para esta prática o aluno deverá completar o esquemático com os componentes montados nas práticas anteriores e testar o processador no kit da Altera.

Para poder testar o processador, será preciso adicionar uma memória no slot indicado pela figura 11.

# MEMORY

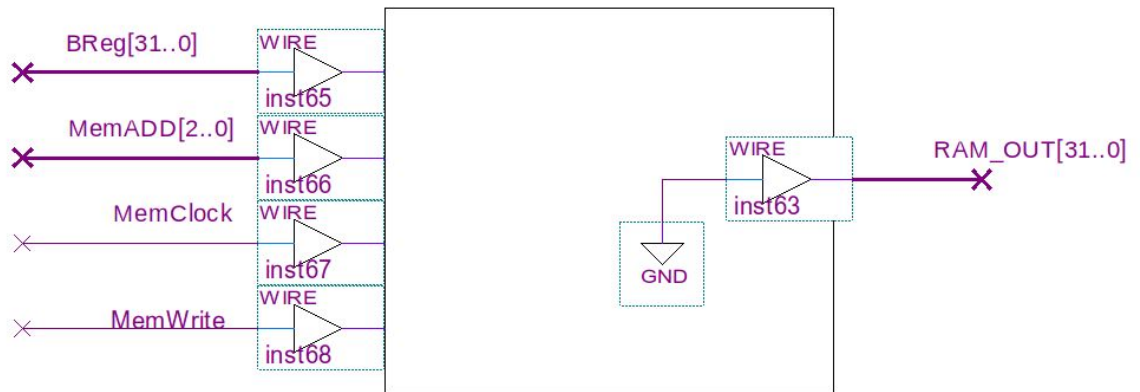


Figura 11.

Utilize o componente LPM\_RAM\_DQ da pasta *megafuncions* para criar a memória (figura 12). Altere os parâmetros do componente (*Properties -> Parameter*) e altere os seguintes campos:

Parameter	Value
LPM_ADDRESS_CONTROL	
LPM_FILE	
LPM_INDATA	
LPM_NUMWORDS	
LPM_OUTDATA	"UNREGISTERED"
LPM_WIDTH	
LPM_WIDTHHAD	

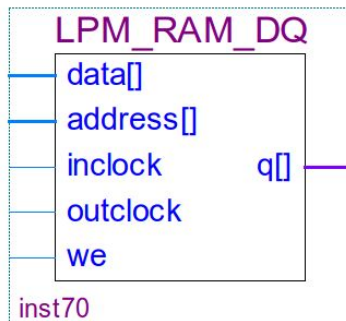


Figura 12.



LPM_ADDRESS_CONTROL	"REGISTERED"
LPM_FILE	mips.mif
LPM_INDATA	"REGISTERED"
LPM_NUMWORDS	8
LPM_OUTDATA	"UNREGISTERED"
LPM_WIDTH	32
LPM_WIDTHHAD	3

Tabela 1.

Também altere em *Properties* -> *Ports* o pino *outclock*. Certifique-se de marcá-lo como *unused* na coluna *status*. O componente deve ficar como mostra a figura 13.

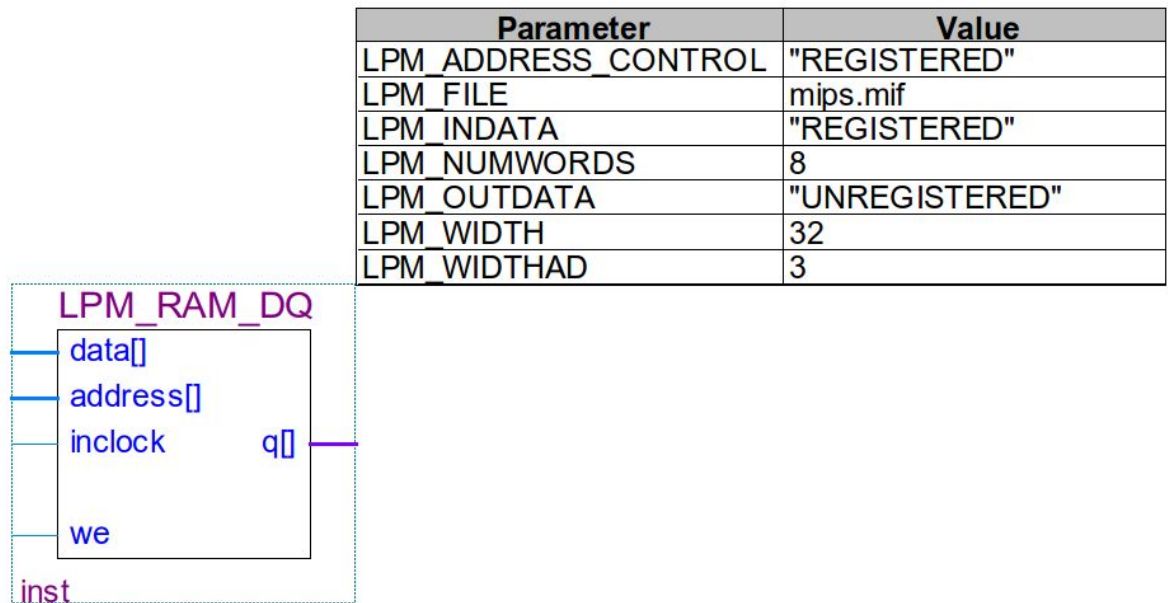


Figura 13.

O arquivo *mips.mif* está carregado com o programa mostrado na tabela 2.

0	LW \$R1,6(\$ZERO)	Carrega dado da posição 6 para registrador 1
1	LW \$R2,7(\$ZERO)	Carrega dado da posição 7 para registrador 2
2	ADD \$R3, \$R1, \$R2	Soma o conteúdo dos registradores 1 e 2 e armazena no registrador 3
3	BEQ \$R3,\$ZERO, 1	Salta para a posição 5 se o conteúdo de R3 for zero
4	SW \$R3, 5(\$ZERO)	Armazena conteúdo de R3 na posição 5
5	-	-
6	Dado	Dado 1
7	Dado	Dado 2

Tabela 2.

Este programa pode ser alterado com as instruções implementadas na prática 3 (Unidade de Controle), para isso altere o arquivo *mips.mif*. (Para expandir o programa, alterar campo LPM\_NUMWORDS da memória e criar outro arquivo “.mif” compatível com o tamanho da memória.)

A conexão da memória deve ser feita da seguinte maneira:

- **BReg[31..0]** ligado em **data[]**
- **MemADD[2..0]** ligado em **address[]**
- **MemClock** ligado em **inclock**
- **MemWrite** ligado em **we**
- **RAM\_OUT[31..0]** ligado em **q[]**

O esquemático montado já cria uma interface no kit para debug. A interface montada está dividida em 4 áreas e é apresentada na figura 14.

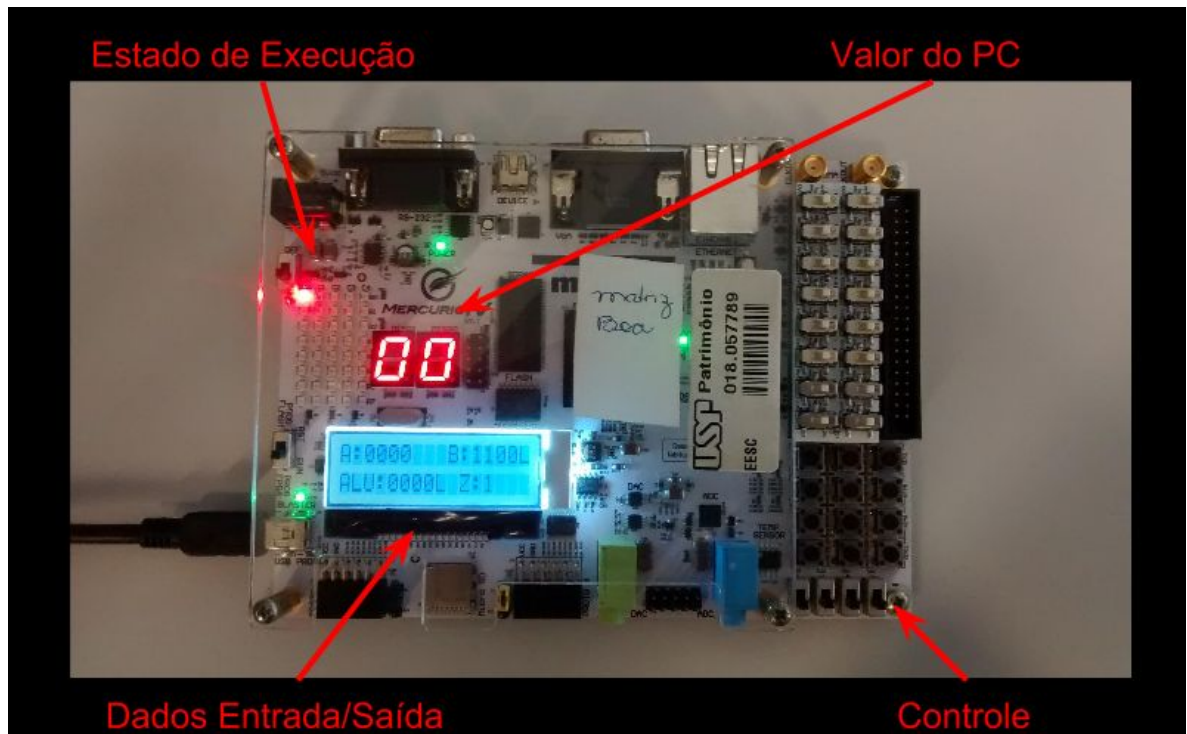


Figura 14.

- O display de 7 segmentos indica o valor atual do PC
- O display LCD mostra quais os operandos da ULA, o resultado da operação e a flag de zero. Os 32 bits foram representados em hexadecimal. A chave SW[1] pode ser utilizada para alternar entre os bits mais significativos (representado por "M" à direita do valor) ou menos significativos (representado por "L" à direita do valor) da ULA, RegA e RegB.
- Na parte de controle, encontram-se a chave SW[1] para controle dos bits mais significativos e o Reset (SW[0]). Além disso, o botão KEY[11] é o clock do sistema.
- A matriz de LED's mostra qual o estado de execução da UC conforme a tabela abaixo:

C0	C1	C2	C3	C4	
Estado 0	Estado 1	Estado 2	Estado 3	Estado 4	<b>R0</b>
Estado 5	Estado 6	Estado 7	Estado 8	Estado 9	<b>R1</b>

Tabela 3.

Alguns cuidados e dicas ao montar adicionar seus componentes:

- Antes de iniciar a simulação, faça o reset do sistema para garantir que os componentes estejam nas devidas condições iniciais.
- Os quadrados são apenas indicações de onde o componente deverá ser adicionado, os componente não precisa estar dentro deste espaço.
- Em cada saída de dados está conectado um GND apenas para que o Quartus compile sem que todos os componentes estejam interligados. Ao conectar uma saída, certifique-se de remover este aterramento.
- Em todas as conexões, o bit mais significativo está à esquerda ([31..0]), certifique-se que os bits dos seus componentes estejam nesta posição. Caso não esteja, poderá utilizar uma interface para fazer a rotação conforme mostra a figura 15.



Figura 15.

- O programa da memória pode ser alterado manipulando o arquivo mips.mif.
- O arquivo mips.mif deve ser incluído no projeto. Para isso vá em *Project -> Add/Remove Files in Project* e adicione o arquivo mips.mif
- Deixe desabilitadas funções adicionais implementadas (Ex.: set dos registradores)

**Departamento de Engenharia Elétrica e de Computação**

**SEL 606 – Laboratório de Sistemas Digitais**

**Prof. Dr. Maximilian Luppe**

NOTA:

**FOLHA DE RESPOSTAS: PRÁTICA Nº4**

**“Montagem do Processador”**

**Data:**

**Nomes**

**NºUSP**
