

## Decodificador e Controle

### 1. Definições

#### 1.1. Decodificador de instruções

Um decodificador de instruções é um componente eletrônico que possui uma entrada (do tamanho de uma instrução) e algumas saídas, com os significados da instrução de entrada.

#### 1.2. Unidade de extensão de sinal

Uma unidade de extensão de sinal completa os bits mais significativos de um dado de entrada com o bit de sinal (positivo ou negativo) desse dado. A saída possui número de bits maior do que a entrada para que o sinal seja copiado.

#### 1.3. Unidade de Controle Principal

A unidade de controle ativa ou desativa *flags* de seleção de dados dentro do datapath. Essa unidade dita o caminho percorrido pelos dados dentro do processador, selecionando as saídas dos *muxes* do *datapath*.

#### 1.4. Unidade de Controle da ULA

A unidade de controle da ULA seleciona a operação a ser realizada na ULA. A operação na ULA é selecionada por essa unidade de controle em separado, que tem como entrada os campos FUNCT das instruções R e um dado de controle de 2 bits chamado ALUOp.

### 2. Exercício

O exercício consiste em implementar um decodificador de instruções do MIPS, uma unidade de extensão de sinal, uma unidade de controle principal e uma unidade de controle da ULA como componentes separados a serem utilizados em um caminho de dados monociclo. Componentes prontos do simulador (como distribuidores, muxes e memórias) podem ser utilizados, desde que justificados.

### 3. Especificação

#### 3.1. Decodificador de instruções

A Figura 1 mostra um diagrama das entradas e saídas do decodificador de instruções. Basicamente, o decodificador separa os campos de cada instrução de 32 bits nos formatos R, I e J do MIPS.

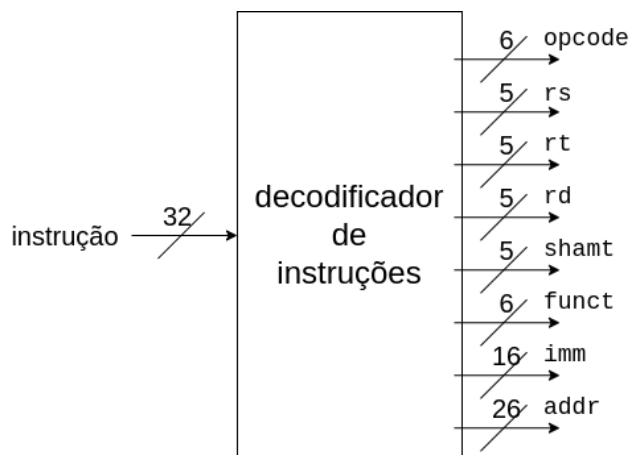


Figura 1 - Entradas e saídas do decodificador de instruções.

A entrada do decodificador de instruções deve ser uma instrução de 32 bits.

As saídas do decodificador devem ser como a descrição abaixo:

- opcode: Código da operação, especificado em instrução[26:31].
- rs: registrador de origem 1 (instruções R) ou de endereço relativo (instruções I), especificado em instrução[21:25].
- rt: registrador de origem 2 (instruções R) ou de destino de dado (instruções I), especificado em instrução[16:20].
- rd: registrador de destino para escrita (instruções R), especificado em instrução[11:15].
- shamt: quantidade de shift (instruções R), especificado em instrução[6:10].
- funct: código de função (instruções R), especificado em instrução[0:5].
- imm: valor do imediato (instruções I), especificado em instrução[0:15].
- addr: endereço absoluto (instruções J), especificado em instrução[0:25].

### 3.2. Unidade de Extensão de Sinal

A entrada e a saída da unidade de extensão de sinal devem ser como o diagrama da Figura 2, na qual a saída contém 16 bits a mais do que a entrada.

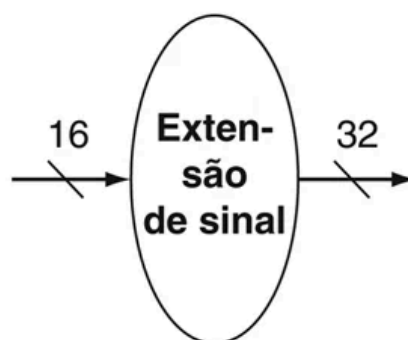


Figura 2 - Unidade de extensão de sinal [1].

No qual:

- A entrada é um dado de 16 bits.
- A saída é um dado de 32 bits, que copia o bit de sinal da entrada para os bits mais significativos da saída.

### 3.3. Unidade de controle principal

A Figura 3 mostra um esquema de entradas e saídas da unidade de controle principal do *datapath*. O campo OPCODE fornecido pelo decodificador de instruções é a única entrada da unidade de controle principal.

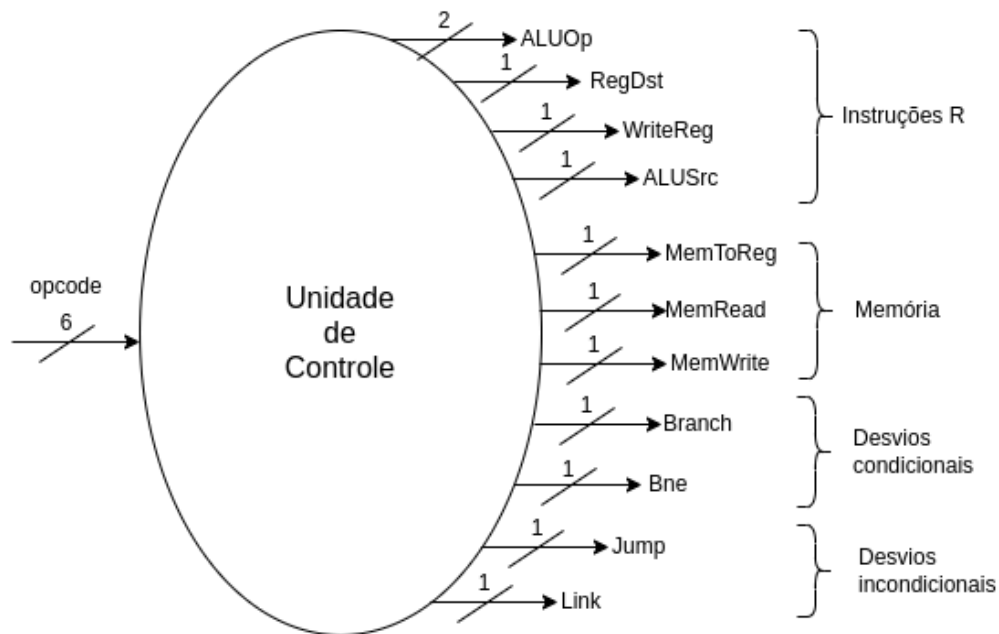


Figura 3 - Entradas e saídas da unidade de controle principal.

Todas as instruções com opcode 0x0 são do tipo R. Os sinais ALUSrc, RegDst, WriteReg e ALUOp servem para essas instruções, cujas saídas abaixo devem ser configuradas:

- RegDst: deve ter o valor 1, indicando que o registrador de destino é RD. Essa saída deve ser ligada ao bit de seleção de um mux na entrada WR do banco de registradores. Esse mux escolhe entre RT e RD para o registrador de escrita dos resultados.
- WriteReg: deve ter o valor 1, indicando que a operação escreve em um registrador. Esse bit deve ser ligado à entrada RegWrite do banco de registradores, que ativa a escrita no banco.
- ALUOp: deve ter o valor 00, indicando que a operação da ULA será selecionada pelo campo FUNCT.

O comportamento das instruções com opcode diferente de 0x0 deve ser como abaixo:

- **ADDI**: deve ativar os sinais WriteReg e ALUSrc. ALUOp deve selecionar a operação de soma na ULA. Outros sinais devem ser desativados.
- **LW**: deve setar os sinais MemToReg (escreve valor da memória em registrador), MemRead, WriteReg e ALUSrc (segunda entrada da ULA é o imediato). ALUOp deve selecionar a operação de soma para calcular o endereço da memória pelo imediato. Outros sinais devem ser desativados.
- **SW**: deve setar os sinais MemWrite (escreve valor do registrador na memória) e ALUSrc (segunda entrada da ULA é o imediato). ALUOp deve selecionar a operação

de soma para calcular o endereço da memória pelo imediato. Outros sinais devem estar desativados.

- **BEQ**: deve colocar como 1 o sinal Branch e informar em ALUOp que se trata de um branch para realizar subtração. Outros sinais devem estar como zero.
- **BNE**: os sinais Branch e Bne devem estar em 1, além disso, deve informar em ALUOp que se trata de um branch para realizar subtração. Outros sinais devem estar desativados.
- **J**: deve setar o sinal Jump. Outros sinais devem ser desativados.
- **JAL**: os sinais Link, Jump e WriteReg devem estar em 1, outros sinais devem estar desativados.
- **JR**: o sinal JumpRegister (saída da unidade de controle da ULA) deve estar como 1 e outros sinais devem estar desativados.

### 3.4. Unidade de Controle da ULA

A Figura 4 mostra um esquema de entradas e saídas para a unidade de controle da ULA. O campo FUNCT fornecido pelo decodificador de instruções deve selecionar qual operação será realizada na ULA, a qual terá seu seletor de instrução ligado à saída Op.

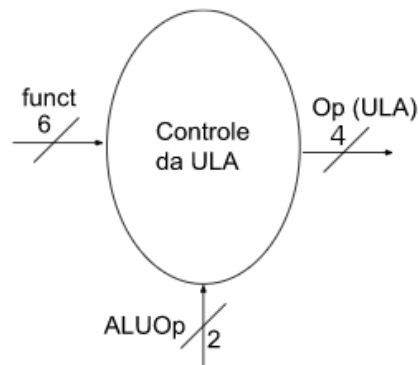


Figura 4 - Unidade de controle da ULA.

A saída Op depende da implementação da ULA feita pelo grupo. Dessa forma, o campo FUNCT da instrução deve servir para selecionar qual operação será ativada quando o valor de ALUOp for "00".

## 4. Entrega

Deverão ser entregues:

1. Um relatório, continuando o documento da entrega anterior, que descreva o comportamento e a implementação das unidades especificadas.
2. Os arquivos com o projeto do simulador Logisim.

O relatório da atividade deve estar no formato do modelo disponibilizado no moodle.

Os dois itens devem estar compactados em um mesmo arquivo .zip.

## Referências

[1] Patterson, David A. Hennessy, John L. Organização e Projeto de Computadores. Disponível em: Minha Biblioteca, (5a. edição). Grupo GEN, 2017.