

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Engenharia de Eletrónica e Telecomunicações e de Computadores

e

Engenharia Informática e de Computadores



1.º Trabalho Prático de Arquitetura de Computadores

Estudo de um processador

26 de setembro de 2018

1 Objetivos

Este trabalho prático tem como principal objetivo o estudo do funcionamento de um processador. Neste contexto, será abordada a codificação do conjunto de instruções que compõe o seu ISA, o projeto do decodificador de instruções correspondente e a execução de programas escritos usando a linguagem máquina.

2 Descrição da arquitetura

O processador considerado neste trabalho, que é de ciclo único, implementa uma arquitetura de Harvard a 8 bits, em que as instruções são codificadas com 13 bits e as memórias de dados e de código contêm ambas 256 posições.

A arquitetura disponibiliza ao programador oito registos de uso geral (r_0 a r_7) e uma Unidade Lógica e Aritmética (ALU) com capacidade para realizar duas operações aritméticas (soma e subtração) e uma operação lógica (XOR), conforme é ilustrado na Figura 1 do Anexo A. A arquitetura inclui ainda um registo de estado do processador (PSW) que disponibiliza os indicadores zero (Z) e transporte (C).

A Tabela 1 apresenta o conjunto de instruções suportado pela arquitetura, em que:

- rx , ry e rz representam um dos oito registos de uso geral do processador (r_0 a r_7);
- $const3$ e $const6$ simbolizam constantes, sem sinal, codificadas com 3 e 6 bits, respetivamente;
- $direct6$ simboliza o valor de uma constante, sem sinal, codificada com 6 bits, que é usada como parte de menor peso na síntese de endereços de memória (os bits de maior peso dos endereços são colocados a zero);
- $offset6$ simboliza o valor de uma constante, com sinal, codificada com 6 bits, que é usada como parte de menor peso na síntese do endereço relativo de memória (os bits de maior peso são estendidos com o bit de sinal).

Instrução	Descrição	
<code>mov rx, #const6</code>	Carrega o valor da constante const6 no registo rx .	$rx \leftarrow const6$
<code>ld rx, [ry+rz]</code>	Transfere para rx o conteúdo da posição de memória cujo endereço é dado pela soma dos conteúdos de ry e rz .	$rx \leftarrow mem[ry+rz]$
<code>ld rx, direct6</code>	Transfere para rx o conteúdo da posição de memória cujo endereço é definido direct6 .	$rx \leftarrow mem[direct6]$
<code>st rx, [ry]</code>	Transfere o conteúdo de rx para a posição de memória cujo endereço está definido em ry .	$mem[ry] \leftarrow rx$
<code>add rx, ry, rz</code>	Adiciona os conteúdos de ry e de rz e coloca o resultado em rx .	$rx \leftarrow ry + rz$
<code>sub rx, ry, #const3</code>	Subtrai const3 ao conteúdo de ry e coloca o resultado em rx .	$rx \leftarrow ry - const3$
<code>xrl rx, ry, rz</code>	Realiza a operação lógica XOR entre ry e rz e coloca o resultado em rx .	$rx \leftarrow ry \text{ XOR } rz$
<code>jnz offset6</code>	Quando a <i>flag Z</i> apresenta o valor 0, muda a execução para o endereço resultante da adição ao PC do deslocamento representado por offset6 .	$PC \leftarrow PC + offset6$ se $Z == 0$
<code>jc offset6</code>	Quando a <i>flag C</i> apresenta o valor 1, muda a execução para o endereço resultante da adição ao PC do deslocamento representado por offset6 .	$PC \leftarrow PC + offset6$ se $C == 1$
<code>jmp rx</code>	Muda a execução para o endereço que se encontra em rx .	$PC \leftarrow rx$
<code>jmp offset6</code>	Muda a execução para o endereço resultante da adição ao PC do deslocamento representado por offset6 .	$PC \leftarrow PC + offset6$

Tabela 1 – Conjunto de instruções do processador.

Na Tabela 2 apresentam-se os *opcodes* incompletos das instruções do ISA.

Instrução	opcode
mov rx, #const6	0000
ld rx, [ry+rz]	00??
ld rx, direct6	0010
st rx, [ry]	00??
add rx, ry, rz	10??
sub rx, ry, #const3	10??
xrl rx, ry, rz	10??
jnz offset6	0100
jc offset6	0101
jmp rx	0110
jmp offset6	0111

Tabela 2 – Mapa de codificação das instruções.

3 Trabalho a realizar

Respeitando o ISA e a microarquitetura apresentados, pretende-se analisar o processador proposto e utilizá-lo para executar um programa. Para tal, devem ser realizadas três tarefas que se descrevem a seguir.

3.1 Codificação das instruções do ISA

- Complete o *opcode* das instruções apresentadas na Tabela 2, por forma a corresponderem ao circuito apresentado na Figura 3 do Anexo A para a realização da ALU.
- Complete o mapa de codificação das instruções apresentado na Tabela 2, de acordo com o diagrama de blocos do processador descrito na Figura 1 do Anexo A.

3.2 Projeto do decodificador de instruções

- Projete o decodificador de instruções do processador, apresentando, numa tabela, o valor lógico das saídas deste circuito para cada uma das instruções descritas na Tabela 1 (NOTA: deve indicar os casos de indiferença – *don't care*). Identifique os sinais da palavra de controlo obtidos diretamente do código da instrução.
- Determine o conteúdo da ROM utilizada para implementar o decodificador de instruções. Preencha a ROM do decodificador de instruções do CPU na aplicação Logisim.

3.3 Teste da arquitetura

Considere a seguinte sequência de instruções, que deverá utilizar para testar o funcionamento do processador utilizando a aplicação Logisim.

```
mov  r3, #1
mov  r0, #5
xrl  r1, r1, r1
ld   r2, [r3+r0]
add  r1, r1, r2
sub  r0, r0, #1
jnc  -3
ld   r2, 1
add  r1, r1, r2
st   r1, [r0]
jmp  0
```

O código apresentado acumula os conteúdos da memória entre os endereços 1 e 7, e guarda o resultado na posição 0 da memória.

- a) Codifique as instruções apresentadas e carregue-as na memória de código do processador no Logisim. Carregue também as primeiras 8 posições da memória de dados com números entre 0 e 9.
- b) Execute o código no Logisim e registe, para cada uma das instruções, as alterações ocorridas nos registos do processador (r0-r7, PC e PSW) e na memória de dados.

4 Avaliação

O trabalho deve ser realizado em grupo, conta para o processo de avaliação da unidade curricular e tem a duração de 2 semanas.

A apresentação da solução proposta por cada grupo decorre em sessão de laboratório, em data a combinar com o docente responsável pela lecionação das aulas da respetiva turma. Após esta apresentação, cada grupo deverá entregar o relatório do trabalho ao docente, no qual deve constar:

- Uma descrição sucinta da solução proposta, que deve ser acompanhada dos esquemas de todos os circuitos e subcircuitos desenvolvidos;
- As respostas às perguntas formuladas no enunciado;
- As conclusões.

Anexo A. Diagramas de blocos

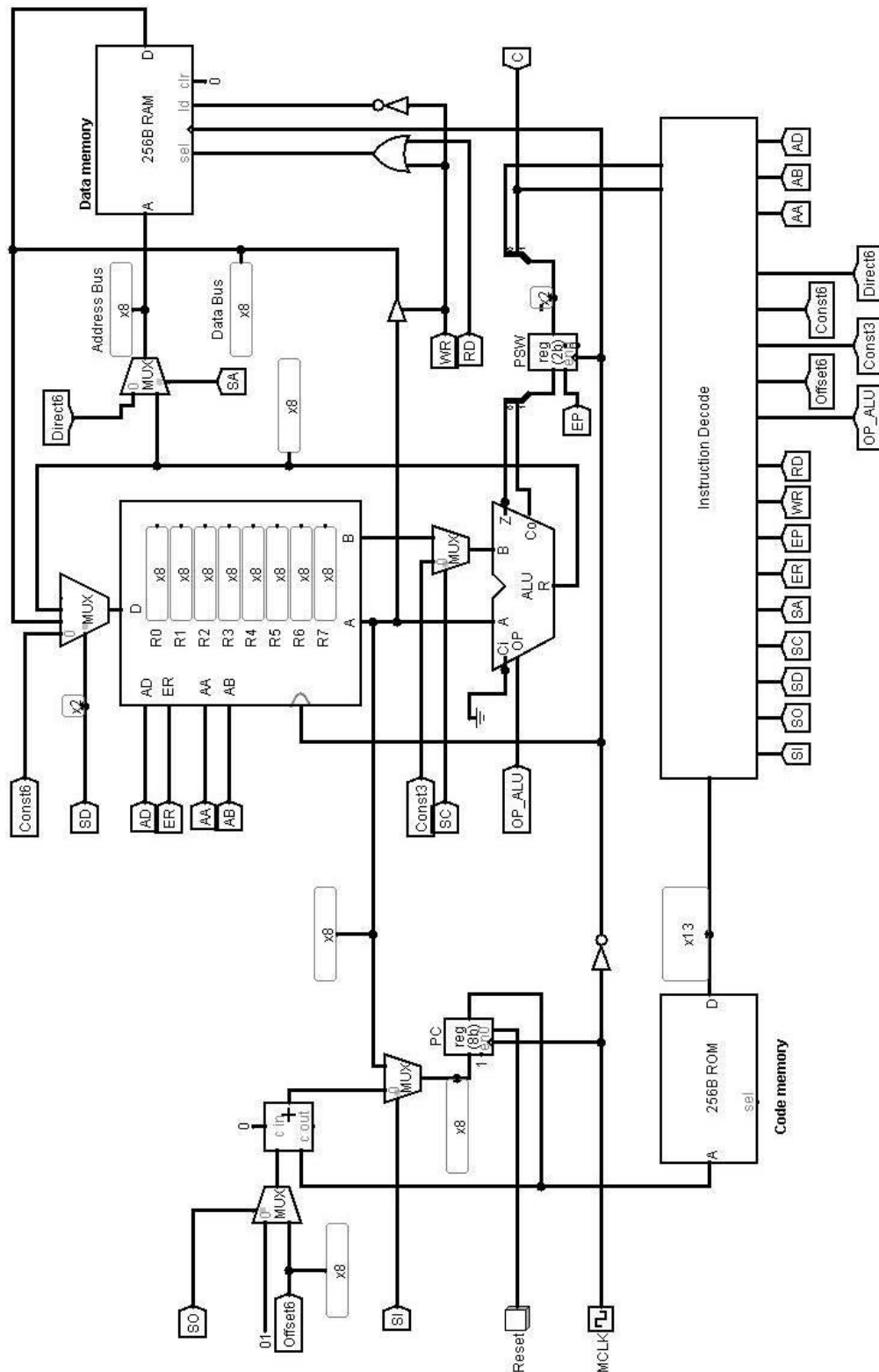


Figura 1 – Diagramas de blocos do processador.

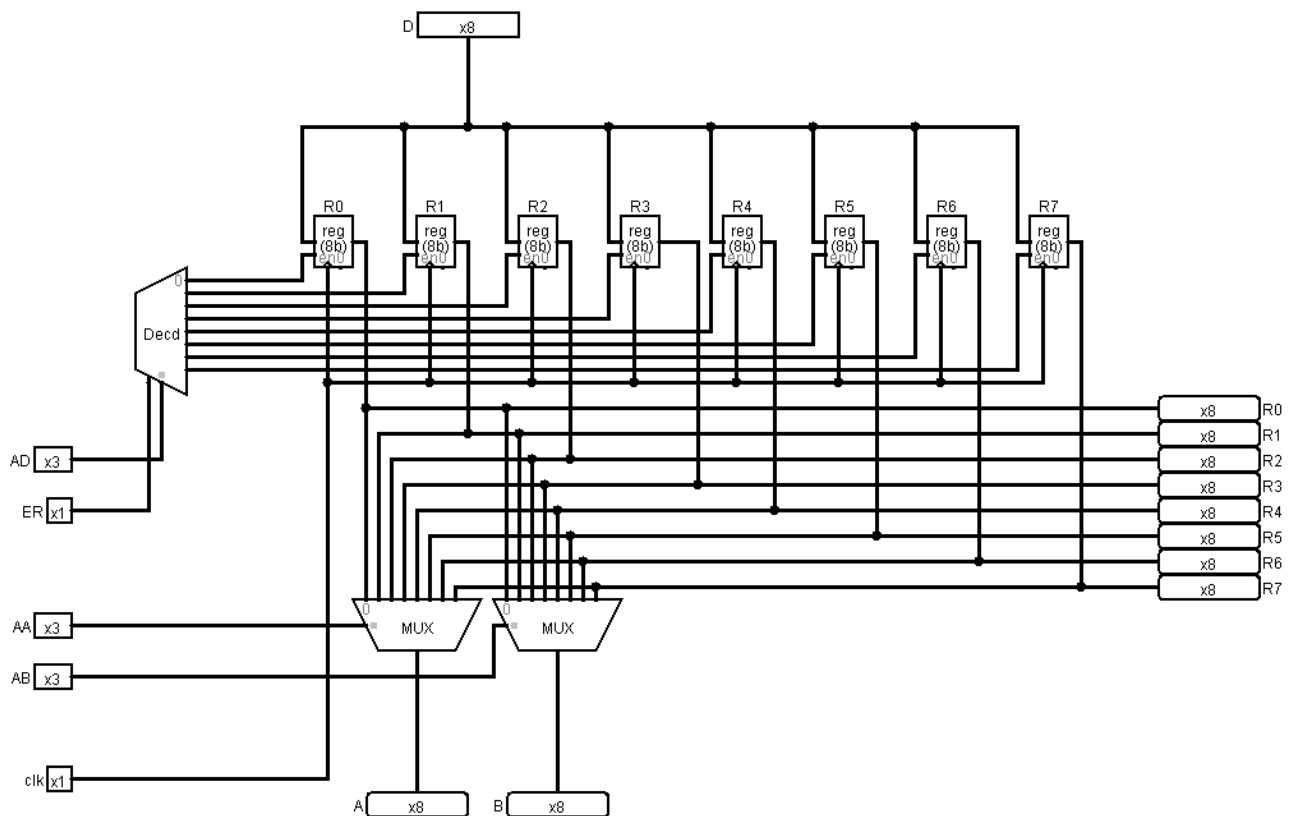


Figura 2 – Diagrama de blocos do banco de registos.

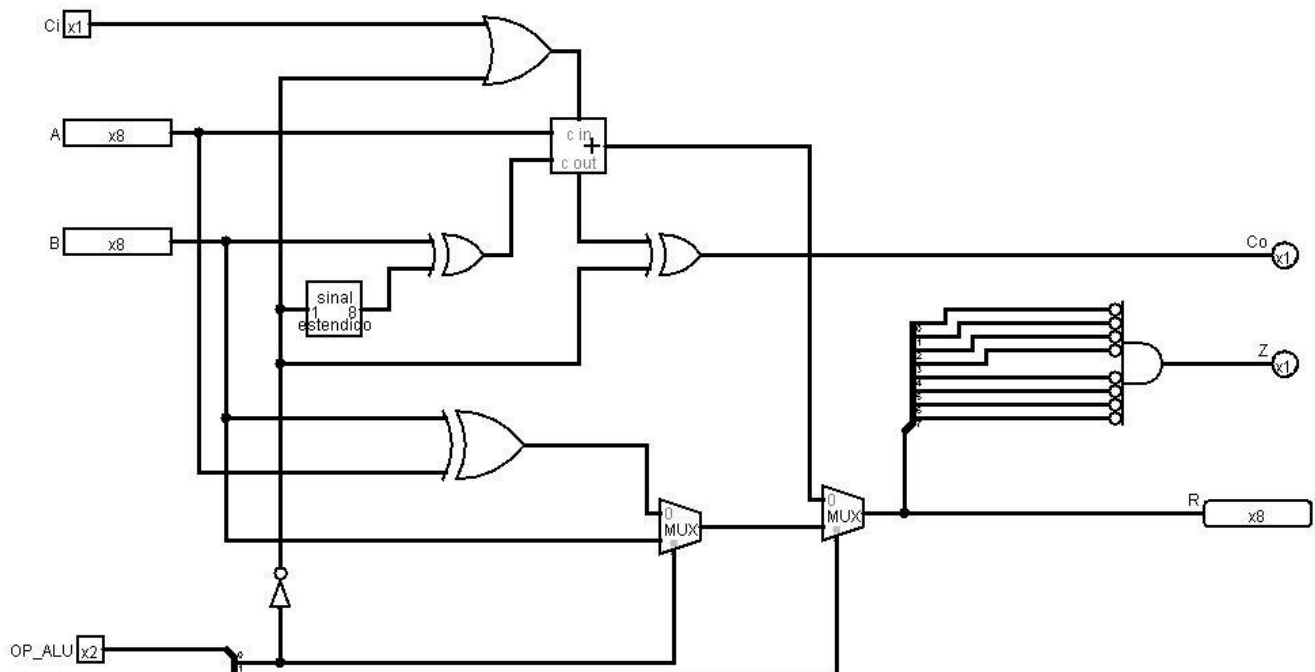


Figura 3 – Diagrama de blocos da ALU.