



Centro de Competências das Ciências Exatas e da Engenharia

Processador P-8

Componente de avaliação P1 (15%) de Arquitetura de Computadores

Ano letivo: 2014/2015

Data de entrega: 06-04-2015

Data de discussão: 08-04-2015

1- Descrição do processador P-8

Este trabalho prático de avaliação pretende que seja realizada a análise, simulação (dos módulos em VHDL, no programa ISE da *Xilinx*), e implementação (em FPGAs, SPARTAN 3E ou 6) do processador P-8, cujo diagrama de blocos encontra-se representado na Figura 1.

Neste processador cada intrução é de 13 *bits*, sendo 5 *bits* reservados para o código de instrução (*Opcode*) e 8 *bits* para os dados ou endereço, seguindo a estrutura presente na tabela 1. São implementadas 5 instruções de leitura/escrita, 10 de lógica/aritmética, 8 de salto e a instrução NOP (*No operation*).

Tabela 1: Estrutura das instruções.

12 - 8 <i>Opcode</i>	7 - 0 Constante
-------------------------	--------------------

Tanto a RAM (Memória de Dados) como a Memória de Instruções são endereçadas por 8 *bits*, tendo a primeira uma capacidade de armazenamento de 8 *bits* e a segunda de 13 *bits*.

A comunicação com o exterior é administrada pelo gestor de periféricos, sendo usado 1 periférico de entrada e 1 de saída, ambos de 8 *bits*. As instruções são realizadas num único período de relógio na transição ascendente e o processador engloba os blocos ALU, Gestor de Periféricos, MuxA, MuxB, MuxPC, PC, ROM, Registo A e Registo das *Flags*. Já a Motherboard inclui o processador, a RAM e a Memória de Instruções.

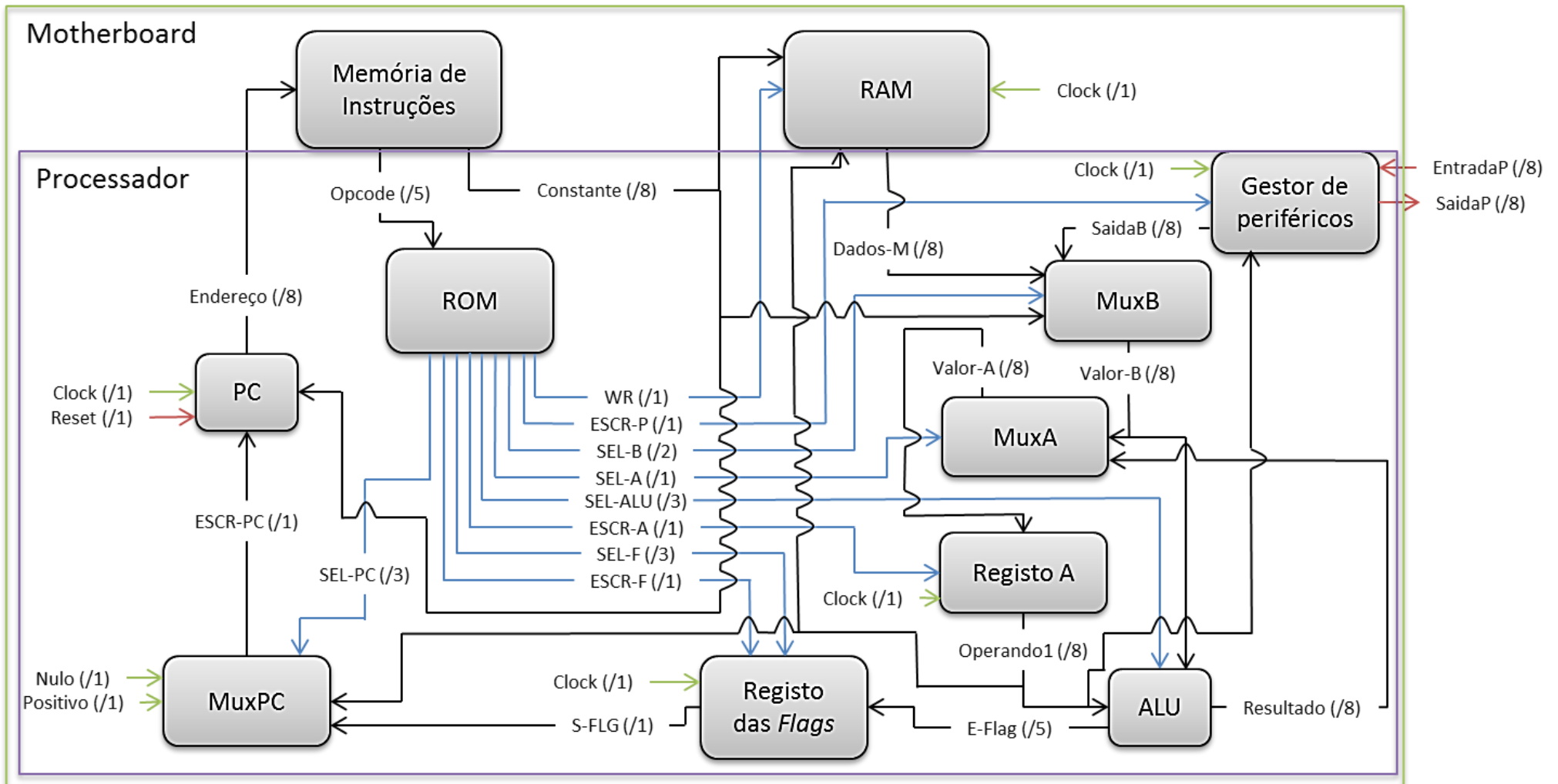


Figura 1: Diagrama de blocos do Processador P-8. Preto: Informação; Azul: Sinais de controlo; Verde: Sinal de relógio; Vermelho: Sinais externos.

2- Descrição dos componentes

2.1- Program Counter (PC)

Assinala o Endereço (8 *bits*) à Memória de Instruções. A entrada ESCR-PC (1 *bit*) indica se deve ser realizado um salto para a posição de memória especificada pela Constante (8 *bits*), caso esta esteja a “1” ou se apenas deve incrementar o contador, se esta estiver a “0”. O Reset (1 *bit*) permite, quando está a “1”, voltar à instrução inicial colocando o contador a 0. O novo ciclo só se deve executar se a entrada Clock (1 *bit*) estiver na transição ascendente.

2.2- Multiplexer do Program Counter (MuxPC)

Indica se o PC deve realizar um salto ou incrementar o contador, através do sinal ESCR-PC (1 *bit*). Este é controlado pelo comando SEL-PC (3 *bits*) que indica qual das fontes de entrada devem passar para a saída, consoante a tabela 2.

Tabela 2: Controlo do Mux PC.

<u>SEL-PC</u>	<u>Entrada</u>
000	S-Flag
001	Positivo
010	Nulo
011	Operando1
100	Operando1
Outros	X

Nota: “X” indica um valor não especificado.

Quando é realizado um salto condicional (JL, JLE, JE, JGE ou JG) o seletor deverá escolher a entrada S-Flag (1 *bit*). Para o salto incondicional (JMP) é utilizada a entrada Positivo (1 *bit* a “1”). A entrada Nulo (1 *bit* a “0”) é usada quando é pretendido apenas incrementar o contador.

Quando é analisado o salto zero (JZ) o seletor indica “011” sendo verificado se o valor de Operando1 (8 *bits*) é 0, se for então ESCR-PC é “1 *bit* a “1”, caso contrário é “1 *bit* a “0”. Já quando é examinado o salto negativo (JN) o seletor indica “100” sendo verificado se o valor de Operando1 é negativo, se for então ESCR-PC é “1 *bit* a “1”, caso contrário é “1 *bit* a “0”.

2.3- Registo das Flags

Armazena o resultado de uma comparação efetuada na ALU, trocando o valor previamente existente se a entrada Clock (1 *bit*) estiver na transição ascendente. Esta informação é fornecida ao MuxPC, através da saída S-Flag

(1 *bit*). O resultado da comparação é indicado através da entrada E-Flag (5 *bits*). O sinal de controlo ESCR-F (1 *bit*) indica se deve, quando é “1”, ou não, quando é “0”, ser guardado o valor da entrada e SEL-F (3 *bits*) designa qual dos bits armazenados é que deve passar para a saída; o seu procedimento encontra-se na tabela 3.

Tabela 3: Controlo do Registo das *Flags*.

<u>SEL-F</u>	Saída
000	Registo “0”, correspondente ao “<”
001	Registo “1”, correspondente ao “<=”
010	Registo “2”, correspondente ao “=”
011	Registo “3”, correspondente ao “>=”
100	Registo “4”, correspondente ao “>”
Outros	X

2.4- Multiplexer B (MuxB)

Seleciona de entre as entradas Dados-M (8 *bits*) provinda da Memória de dados, Dados-P (8 *bits*) provinda do gestor de periféricos ou Constante (8 *bits*) provinda da memória de instruções, qual é o valor que deverá passar para Valor-B (8 *bits*), de acordo com o sinal de seleção SEL-B (2 *bit*) estando o seu procedimento indicado na tabela 4.

Tabela 4: Controlo do MuxB.

<u>SEL-B</u>	Saída
00	<u>Constante</u>
01	<u>Dados-M</u>
10	<u>Dados-P</u>
Outros	X

2.5- ROM de descodificação (ROM)

Recebe o Opcode (5 *bits*) e fornece os seguintes sinais de controlo:

- SEL-ALU (3 *bits*) seleciona qual a operação a ser efetuada na ALU;
- ESCR-P (1 *bit*) indica ao Gestor de Periféricos se deve realizar uma leitura ou uma escrita no periférico;
- SEL-A (1 *bit*) seleciona qual é a entrada do MuxA que deve passar para a saída;
- SEL-B (2 *bits*) escolhe qual é a entrada do MuxB que deve passar para a saída;
- WR (1 *bit*) assinala se está a ser efetuada uma escrita ou leitura na RAM;
- SEL-PC (3 *bits*) seleciona a origem do sinal de controlo do PC;
- ESCR-A (1 *bit*) fornece autorização de escrita no Registo A;
- ESCR-F (1 *bit*) indica se deve ser escrito o valor provindo da comparação da ALU no Registo das *Flags*;
- SEL-F (3 *bits*) assinala qual dos bits guardados no Registo *Flags* é que deve passar para a saída.

Na tabela 5 encontra-se a relação entre as operações *assembly*, o código máquina e os sinais de controlo.

Tabela 5: Relação entre as operações *assembly*, o código máquina e os sinais de controlo.

<u>Opcode</u>	<u>Assembly</u>	<u>Descrição</u>	<u>SEL- ALU</u>	<u>ESCR-P</u>	<u>SEL-A</u>	<u>SEL-B</u>	<u>WR</u>	<u>SEL-PC</u>	<u>ESCR-A</u>	<u>ESCR-F</u>	<u>SEL-F</u>
Leitura e Escrita											
00000	LD valor	Escreve no Registo A a constante indicada por valor	XXX	1	0	00	0	010	1	0	XXX
00001	LD [endereço]	Escreve no Registo A uma cópia da célula da RAM indicada por endereço	XXX	1	0	01	0	010	1	0	XXX
00010	ST [endereço]	Escreve na célula da RAM indicada por endereço uma cópia do Registo A	XXX	1	X	XX	1	010	0	0	XXX
Lógica e Aritmética											
00011	ADD valor	Soma o Registo A com a constante indicada por valor e escreve o resultado no Registo A	000	1	1	00	0	010	1	0	XXX
00100	ADD [endereço]	Soma o Registo A com a célula da RAM indicada por endereço e escreve o resultado no Registo A	000	1	1	01	0	010	1	0	XXX
00101	SUB valor	Subtrai do Registo A a constante indicada por valor e escreve o resultado no Registo A	001	1	1	00	0	010	1	0	XXX
00110	SUB [endereço]	Subtrai do Registo A a célula da RAM indicada por endereço e escreve o resultado no Registo A	001	1	1	01	0	010	1	0	XXX
00111	AND valor	Efetua a conjugação lógica <i>bit a bit</i> do Registo A com a constante indicada por valor e escreve o resultado no Registo A	010	1	1	00	0	010	1	0	XXX
01000	AND [endereço]	Efetua a conjugação lógica <i>bit a bit</i> do Registo A com a célula da RAM indicada por endereço e escreve o resultado no Registo A	010	1	1	01	0	010	1	0	XXX
01001	OR valor	Efetua a disjunção lógica <i>bit a bit</i> do Registo A com a constante indicada por valor e escreve o resultado no Registo A	011	1	1	00	0	010	1	0	XXX
01010	OR [endereço]	Efetua a disjunção lógica <i>bit a bit</i> do Registo A com a célula da RAM indicada por endereço e escreve o resultado no Registo A	011	1	1	01	0	010	1	0	XXX
01011	CMP valor	Compara o Registo A com a constante indicada por valor e escreve o resultado no Registo das <i>Flags</i>	100	1	X	00	0	010	0	1	XXX
01100	CMP [endereço]	Compara o Registo A com a célula da RAM indicada por endereço e escreve o resultado no Registo das <i>Flags</i>	100	1	X	01	0	010	0	1	XXX

Salto											
01101	JL endereço	Salta para a instrução indicada por endereço se o valor do Registo das <i>Flags</i> referente à comparação "<" for "1"	XXX	1	X	XX	0	000	0	0	000
01110	JLE endereço	Salta para a instrução indicada por endereço se o valor do Registo das <i>Flags</i> referente à comparação "<=" for "1"	XXX	1	X	XX	0	000	0	0	001
01111	JE endereço	Salta para a instrução indicada por endereço se o valor do Registo das <i>Flags</i> referente à comparação "=" for "1"	XXX	1	X	XX	0	000	0	0	010
10000	JGE endereço	Salta para a instrução indicada por endereço se o valor do Registo das <i>Flags</i> referente à comparação ">=" for "1"	XXX	1	X	XX	0	000	0	0	011
10001	JG endereço	Salta para a instrução indicada por endereço se o valor do Registo das <i>Flags</i> referente à comparação ">" for "1"	XXX	1	X	XX	0	000	0	0	100
10010	JMP endereço	Salta para a instrução indicada por endereço	XXX	1	X	XX	0	001	0	0	XXX
10011	JZ endereço	Salta para a instrução indicada por endereço se o valor do Registo A for "0"	XXX	1	X	XX	0	011	0	0	XXX
10100	JN endereço	Salta para a instrução indicada por endereço se o valor do Registo A for negativo	XXX	1	X	XX	0	100	0	0	XXX
Periféricos											
10101	LD periférico	Escreve no Registo A uma cópia do valor do Periférico	XXX	1	0	10	0	010	1	0	XXX
10110	ST periférico	Escreve no Periférico uma cópia do Registo A	XXX	0	X	XX	0	010	0	0	XXX
Sem Operação											
Outros	NOP	Não executa nenhuma operação e apenas incrementa o contador do PC	XXX	1	X	XX	0	010	0	0	XXX

Nota: "X" indica um valor não especificado.

2.6- Memória de Instruções

Armazena as informações sobre o programa que irá ser executado numa memória de 13 *bits*, endereçada pelo Endereço (8 *bits*), disponibilizando o Opcode (5 *bits*) e a Constante (8 *bits*).

2.7- Memória de Dados (RAM)

Permite guardar os dados numa memória de 8 *bits* endereçada por 8 *bits* se a entrada Clock (1 *bit*) estiver na transição ascendente. Quando o sinal WR (1 *bit*) está a “1” indica que a informação presente em Entrada (8 *bits*) deve ser guardada na célula indicada por Constante (8 *bits*); caso contrário, é colocado na saída Dados-M (8 *bits*), o valor guardado na célula de memória referenciada por Constante.

2.8- Arithmetic Logic Unit (ALU)

Responsável por gerar as operações aritméticas e lógicas. São recebidos 2 sinais na entrada, Operando1 (8 *bits*) e Operando2 (8 *bits*), do tipo **signed** e a operação efetuada é selecionada por SEL-ALU (3 *bits*), estando as funções indicadas na tabela 6.

Tabela 6: Funções da ALU.

<u>SEL-ALU</u>	<u>Descrição</u>
000	Soma do <u>Operando1</u> com o <u>Operando2</u>
001	Subtração do <u>Operando1</u> com o <u>Operando2</u>
010	AND do <u>Operando1</u> com o <u>Operando2</u>
011	OR do <u>Operando1</u> com o <u>Operando2</u>
100	Comparação do <u>Operando1</u> com o <u>Operando2</u>
Outros	X

Nas operações de soma, subtração, AND ou OR o resultado é colocado na saída Resultado (8 *bits*). Já se for efetuada uma comparação então a saída E-Flag (5 *bits*) é atualizada, sendo executada uma comparação individual para cada bit.

2.9- Gestor de Periféricos

Realiza a gestão da comunicação do processador com o exterior. Quando o sinal de controlo ESCR-P (1 *bit*) está a “1” indica que é efetuada a leitura do periférico pelo que o valor presente em EntradaP (8 *bits*) é passado para a saída SaidaB (8 *bits*) que está ligada ao MuxB.

Quando ESCR-P está a “0” designa uma escrita no periférico sendo o valor EntradaA (8 *bits*), que está ligado à saída do Registo A, passado para a saída SaídaP (8 *bits*) se a entrada Clock (1 *bit*) estiver na transição ascendente.

2.10- Multiplexer A (MuxA)

Seleciona de entre as entradas Valor-B (8 *bits*) provinda do MuxB e Resultado (8 *bits*) provinda da ALU, qual é o valor que deverá passar para Valor-A (8 *bits*), de acordo com o sinal de seleção SEL-A (1 *bit*) estando o seu procedimento indicado na tabela 7.

Tabela 7: Controlo do MuxB.

<u>SEL-A</u>	<u>Saída</u>
0	<u>Valor-B</u>
1	<u>Resultado</u>
Outros	X

2.11- Registo A

Possui um único registo de 8 *bits* que é controlado por ESCR-A (1 *bit*) que quando é “1” o valor presente na entrada Valor-A (8 *bits*) é guardado no registo se a entrada Clock (1 *bit*) estiver na transição ascendente. O valor armazenado é transmitido para os restantes blocos pela saída Operando1 (8 *bits*).

3- Teste ao processador P-8

Na tabela 8 encontra-se o programa de teste que deve ser implementado para testar o projeto.

Tabela 8: Instruções de teste do projeto.

<u>Endereço</u>	<u>Instrução (assembly)</u>	<u>Instrução (código máquina)</u>
00000000	LD 2	
00000001	ADD 1	
00000010	ST periférico	
00000011	ST [129]	
00000100	SUB 1	
00000101	AND [129]	
00000110	ST periférico	
00000111	CMP 3	
00001000	JE 11	
00001001	LD 4	
00001010	JMP 4	
00001011	LD [129]	
00001100	ADD [129]	

00001101	ST periférico
00001110	LD periférico
00001111	CMP 170
00010000	JE 18
00010001	JMP 14
00010010	ST periférico
00010011	JMP 18

4- Avaliação e informações relevantes

O projeto deve ser realizado individualmente ou em grupo de 2 alunos, representando 15% na nota final e tem nota mínima de 8 valores.

O relatório poderá ter no máximo 5 páginas (sem contar com os anexos, capa e índice), contendo a descrição do trabalho realizado. A sua estrutura deverá incluir os seguintes tópicos: 1. Introdução, 2. Objetivos, 3. Discussão de resultados, 4. Conclusão e 5. Bibliografia, Anexo A, Anexo B e Anexo C. No primeiro anexo (A) deverá ser apresentada a tabela das instruções de teste (tabela 8) preenchida e a respetiva simulação (com os sinais de entrada e saída em decimal e legíveis). No segundo (B) deverão ser representados os fluxogramas de todos os módulos que o processador contém, da Memória de Instruções (genérico) e da RAM (genérico). O anexo C deverá conter a listagem de código dos módulos em VHDL.

O processador deverá funcionar corretamente no simulador (ISim) e na FPGA usando o sinal de relógio da mesma. Apenas o relatório deverá ser entregue ao Gabinete de Apoio ao Estudante ("trabalhos@uma.pt"). A cópia do trabalho implica a anulação do mesmo.

A discussão do trabalho é individual, sendo necessário apresentar o funcionamento do mesmo no simulador e na FPGA (o aluno deverá iniciar a discussão já com o programa a correr no computador e uma cópia do enunciado). Caso não possua computador pessoal poderá usar um dos computadores da Universidade, devendo avisar previamente os docentes.