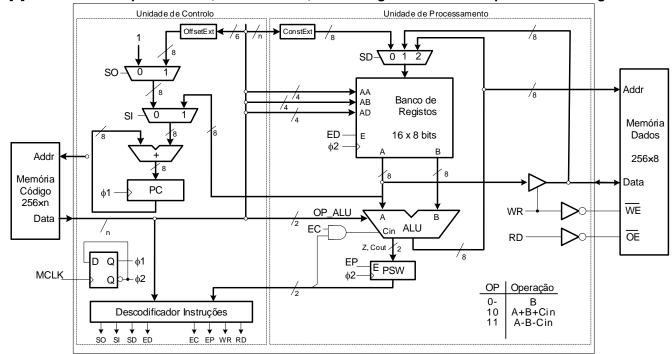
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

LEIC, LEETC, LEIRT Arquitetura de Computadores 1º Teste (19/jun/2018)

Duração do Teste: 2 horas e 30 minutos

[1] Considere um processador, de ciclo único, com o diagrama de blocos apresentado na figura.

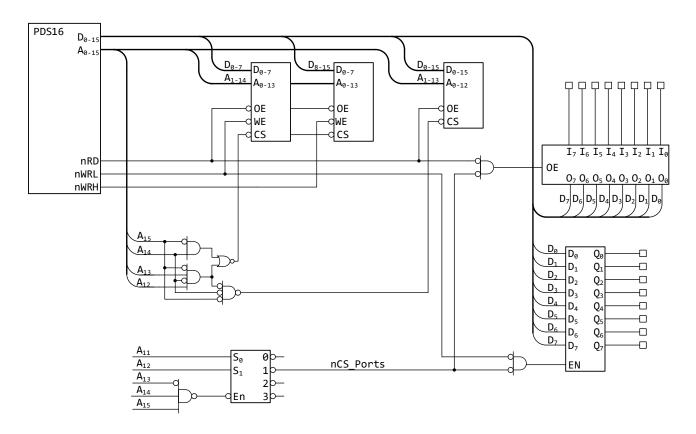


O processador suporta a execução do seguinte conjunto de instruções, em que a constante consts representa números naturais e a constante offset₆ representa números relativos:

N.º	Instrução	Codificação												Descrição
		b ₁₁	b ₁₀	b ₉	b ₈	b ₇	\mathbf{b}_6	\mathbf{b}_5	b_4	b ₃	b ₂	b_1	\mathbf{b}_0	
1	ldi rx,#const5	A definir												rx = const ₅
2	ld ry,[rx]	A definir											ry = M[rx]	
3	st ry,[rx]	A definir											M[rx] = ry	
4	adc ry,rx	0	1	0	0	rx3	rx2	rx1	rx ₀	rуз	ry2	ry1	ry0	ry = ry + rx + cin
5	sbb ry,rx	0	1	1	1	rx3	rx2	rx ₁	rx ₀	ry3	ry2	ry ₁	ry ₀	ry = ry - rx - cin
6	jnz rx		A definir											(Z==0) ? PC = PC + rx : PC = PC + 1
7	jc offset6	1	1	0	0	0	0	O ₅	O ₄	О3	O ₂	O ₁	O ₀	(C==1) ? PC = PC + offset6 : PC = PC + 1

- a) Codifique as instruções ldi, ld, st e jnz utilizando uma codificação linear a 3 bits. Explicite os bits do código de instrução que correspondem aos sinais AA, AB, AD, OP_ALU e OPCODE. [2 val.]
- b) Indique, justificando, a gama de endereços de memória de dados a que é possível aceder com as instruções 1d e st. [0,5 val.]
- c) Considerando que o módulo Descodificador Instruções é implementado usando exclusivamente uma ROM, indique o conteúdo desta memória. [1,5 val.]
- d) Indique a dimensão em bits das ROM utilizadas na implementação da memória de código e do módulo Descodificador Instruções. Apresente os cálculos realizados. [0,5 val.]
- e) Indique, justificando, quantos ciclos de relógio são necessários para executar as instruções de acesso à memória. [0,5 val.]

[2] Considere o sistema computacional baseado no PDS16 representado na figura.



- a) Desenhe os mapas de endereçamento do sistema inicial e do sistema modificado, indicando a funcionalidade, as dimensões, os endereços de início e de fim do espaço atribuído a cada dispositivo, a eventual existência de *fold-back* e as gamas de endereços livres. [2 val.]
- b) Desenhe o esquema de um módulo de RAM adicional, com apenas dois circuitos RAM de 8Kx8, acessível a *byte* e *word*, mas que ocupe todo o espaço de endereçamento livre. [2 val.]
- c) Admitindo os seguintes valores para os registos R0=0x9000, R1=0xD000, R2=1 e R3=2, qual o valor do registo R4 após as seguintes instruções. Justifique. [1 val.]

ST R2,[R1,#0]

ST R3,[R0,#0]

LD R4,[R1,#0]

[3] Considere as seguintes funções expressas em linguagem C:

```
int16 signal_extend(uint8 value, uint8 signal_mask) {
   uint8 value_mask = signal_mask - 1;
   if ((value & signal_mask) != 0)
        return ~value_mask | value;
   else
        return value_mask & value;
}

void array_signal_extend(uint8 input[], int16 output[], uint8 size) {
   for (uint8 i = 0; i < size; ++i)
        output[i] = signal_extend(input[i], 4);
}</pre>
```

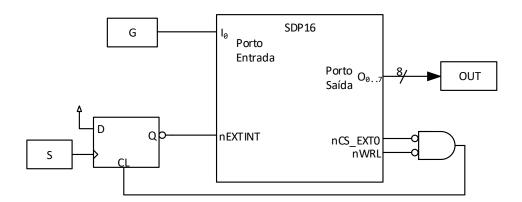
- a) Traduza para linguagem assembly do PDS16 a função signal_extend e defina as respetivas variáveis, se necessário. [2,5 val.]
- b) Traduza para linguagem assembly do PDS16 a função array_signal_extend e defina as respetivas variáveis, se necessário. [2,5 val.]

Notas:

- 1. Com vista ao alojamento de variáveis, assuma que a secção ".data" está localizada na zona de memória acessível com endereçamento direto.
- 2. Na programação em assembly deve usar as seguintes convenções: passagem de parâmetros em r0, r1, r2 e r3; valor de retorno em r0; int8 e int16 significam valores inteiros com sinal representados a 8 e a 16 bit, respetivamente; uint8 e uint16 significam valores inteiros sem sinal representados a 8 e a 16 bit. A função preserva os registos que utiliza para além dos usados para parâmetros.

[4] Tendo como base o sistema SDP16, pretende-se implementar um sistema com a seguinte especificação:

- Enquanto o sinal G estiver ativo o sistema conta as transições ascendentes do sinal S;
- Durante a contagem o sistema coloca zero na saída OUT;
- Estando a entrada G não ativa o sistema afixa em OUT o número de transições ascendentes que contou durante o tempo em que o sinal G esteve ativo.



Sabendo que o sistema inicia-se com os sinais G e S desativos,

- a) Escreva em assembly do PDS16 a rotina de atendimento da interrupção. [2 val]
- b) Tendo como base a rotina de atendimento da interrupção, escreva em *assembly* do PDS16 o sistema. [3 val]