INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

LEIC, LEETC

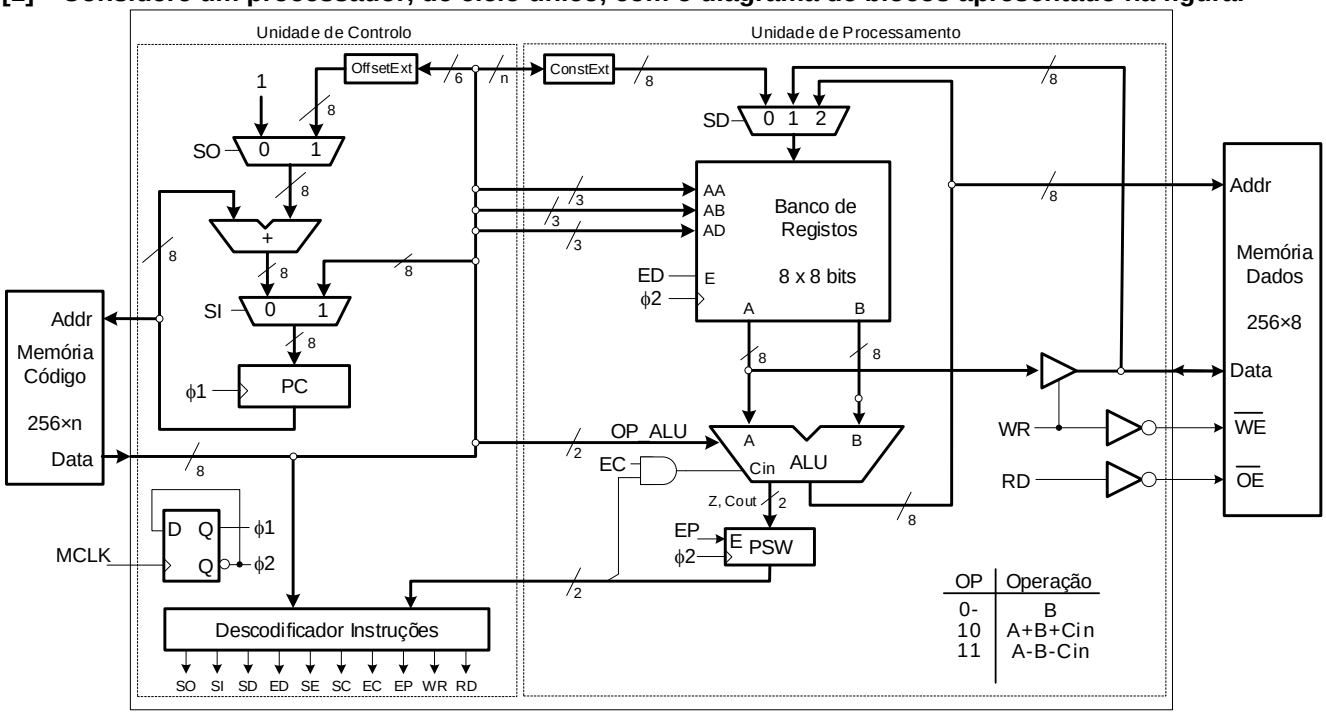
Arquitetura de Computadores

2º Teste (30/jan/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 as constantes `const5` e `address8` representam números naturais e a constante `offset6` representa um número relativo:

N.º	Instrução	Codificação												Descrição
		b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
1	`ldi rx,#const5`	A definir												`rx = const5`
2	`ld rx,[ry]`	A definir												`rx = M[ry]`
3	`st rx,[ry]`	0	0	0	rx_2	rx_1	rx_0	ry_2	ry_1	ry_0	0	0	0	`M[ry] = rx`
4	`adc rx,ry,rz`	0	1	0	ry_2	ry_1	ry_0	rz_2	rz_1	rz_0	rx_2	rx_1	rx_0	`rx = ry + rz + cin`
5	`sbb rx,ry,rz`	0	1	1	ry_2	ry_1	ry_0	rz_2	rz_1	rz_0	rx_2	rx_1	rx_0	`rx = ry - rz - cin`
6	`jz address8`	A definir												`(Z==1) ? PC=address8 : PC = PC + 1`
7	`jmp offset6`	A definir												`PC = PC + offset6`

a) Codifique as instruções `ldi`, `ld`, `jz` e `jmp` 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.]

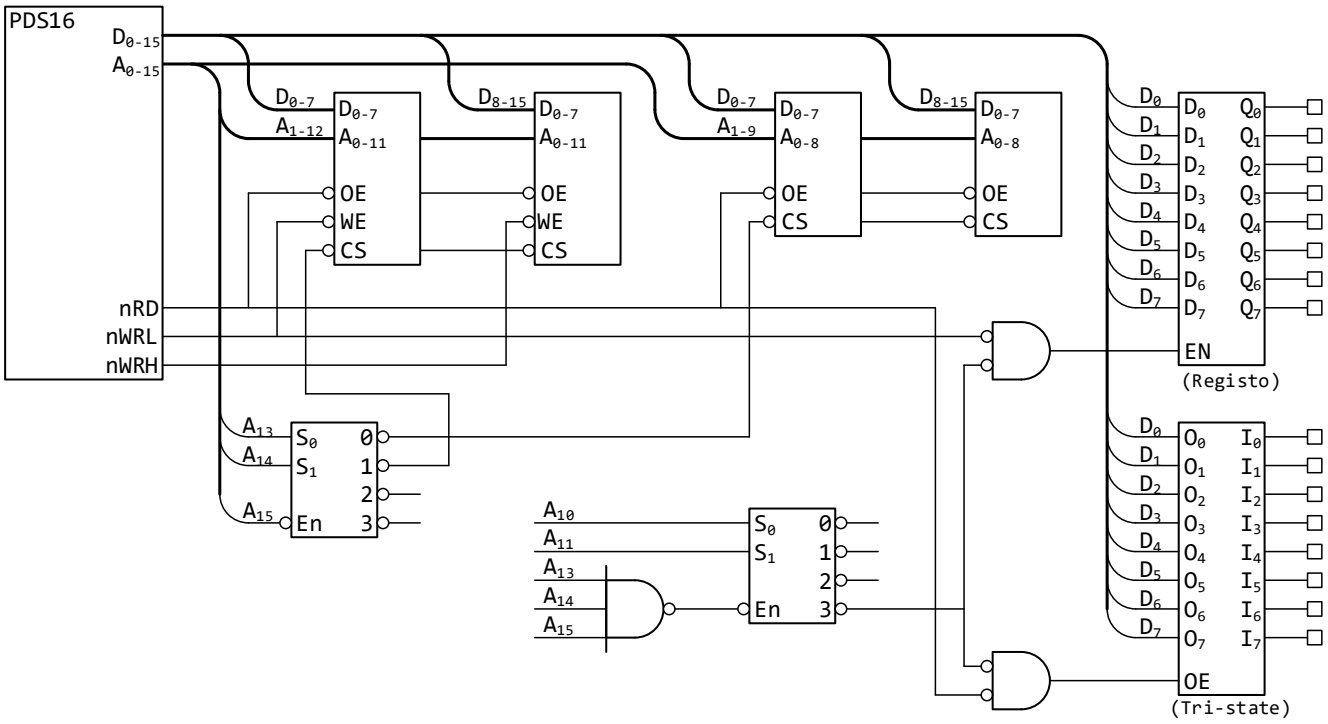
b) Considere que o PC = 0x40. Indique a gama de endereços possíveis de alcançar com a instrução JMP. [0,5 val.]

c) Considerando que o módulo `Descodificador Instruções` é implementado usando exclusivamente uma ROM, indique a programação da mesma. [1,5 val.]

d) Indique a dimensão em bits da memória de código e da ROM do módulo `Descodificador Instruções`, apresentando os cálculos realizados. [0,5 val.]

e) Proponha, justificando, um diagrama lógico para o módulo `constExt`. [0,5 val.]

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



a) Quais os endereços base e dimensões que os módulos ROM, RAM e portos de entrada e saída ocupam no espaço de endereçamento? Indique eventuais zonas em *foldback*. [1,5 val.]

b) Desenhe o esquema de um módulo de RAM adicional com 16Kbyte de dimensão e ocupando a gama de endereços 0x6000-0x9FFF, usando módulos RAM de 4Kx8. [2 val.]

c) Desenhe o esquema de um porto de saída adicional com 8 bits de modo a que possa formar, junto com o existente, um porto de saída acessível a 8 ou a 16 bits [1 val.]

d) Porque é que os portos de entrada/saída representados na figura não podem ser acedidos pelas instruções que utilizam endereçamento direto? [0.5 val.]

[Escreva aqui]

[3] Considerando as convenções definidas para a passagem de parâmetros, retorno de valores e preservação de registos e que os tipos int16 e uint16 representam inteiros a 16 bits com e sem sinal, respetivamente, considere as definições seguintes:

```
uint16 valabs( int16 val ){
  if( val < 0 ){
   return -val;
  }
  return val;
}

void copyabs( uint16 d[], int16 s[], uint16 n ){
  while( n != 0 ){
   --n;
   d[n] = valabs( s[n] );
  }
}
```

Com vista ao alojamento de variáveis, assuma que a secção ".data" está localizada na gama de memória com endereçamento direto.

Escreva, em *assembly* do PDS16, o alojamento das variáveis em memória, se necessárias, e a tradução do código:

a) Da função valabs. [2,5 val.]
b) Da função copyabs. [2,5 val.]

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

- O sistema inicia-se com os dois sensores, S1 e S2, inativos e o valor 0x00 afixado no porto de saída.
- Enquanto o sensor S1 estiver ativo, o sistema conta as transições ascendentes no sensor S2;
- Quando o sensor S1 ficar inativo, o sistema deve fixar no porto de saída o valor da contagem. Este valor deve ser mantido até à próxima contagem de objetos.
- Os sensores S1 e S2 estão ligados aos bits de índice 0 e 7, respetivamente, do porto de entrada disponível no SDP16.

a) Desenhe um fluxograma da máquina de estados do sistema. [1 val.]
b) Programe em *assembly* do PDS16 o sistema de contagem enunciado. [4 val.]