# INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

LEIC, LEETC

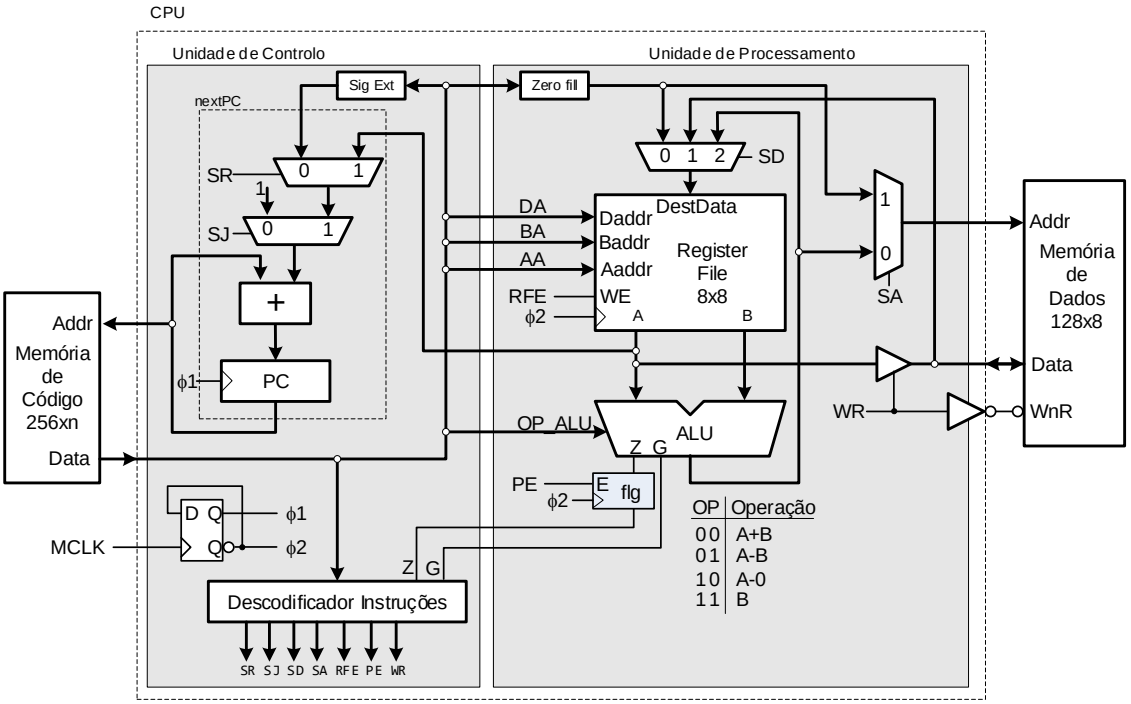
Arquitetura de Computadores

## 1º Teste (19/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 a constante `const4` representa um número natural e a constante `offset` representa um número relativo:

| N.º | Instrução | Codificação | | | | | | | | | | Descrição |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **$b_9$** | **$b_8$** | **$b_7$** | **$b_6$** | **$b_5$** | **$b_4$** | **$b_3$** | **$b_2$** | **$b_1$** | **$b_0$** | |
| 1 | `ldi rx,#const4` | 0 | 1 | 0 | $c_3$ | $c_2$ | $c_1$ | $c_0$ | $rx_2$ | $rx_1$ | $rx_0$ | `rx = const4` |
| 2 | `ld rx,[ry,rx]` | A definir | | | | | | | | | | `rx = M[ry+rx]` |
| 3 | `st rx,[ry]` | 0 | 1 | 1 | 0 | $ry_2$ | $ry_1$ | $ry_0$ | $rx_2$ | $rx_1$ | $rx_0$ | `M[ry] = rx` |
| 4 | `add rx,ry` | 0 | 0 | 0 | 0 | $ry_2$ | $ry_1$ | $ry_0$ | $rx_2$ | $rx_1$ | $rx_0$ | `rx = rx + ry` |
| 5 | `sub rx,ry` | 0 | 0 | 1 | 0 | $ry_2$ | $ry_1$ | $ry_0$ | $rx_2$ | $rx_1$ | $rx_0$ | `rx = rx – ry` |
| 6 | `jnz offset` | A definir | | | | | | | | | | `(Z == 0) ? PC = PC + offset : PC = PC + 1` |
| 7 | `jmp rx` | A definir | | | | | | | | | | `PC = PC + rx` |

a) Codifique as instruções `ld`, `jnz` e `jmp` utilizando uma codificação linear a 3 bits. Explicite os bits do código de instrução que correspondem aos sinais AA, BA, DA, OP_ALU e OPCODE. [2 val.]

b) Considere que o PC = 0x80. Indique a gama de endereços possíveis de alcançar com a instrução JNZ. [0,5 val.]

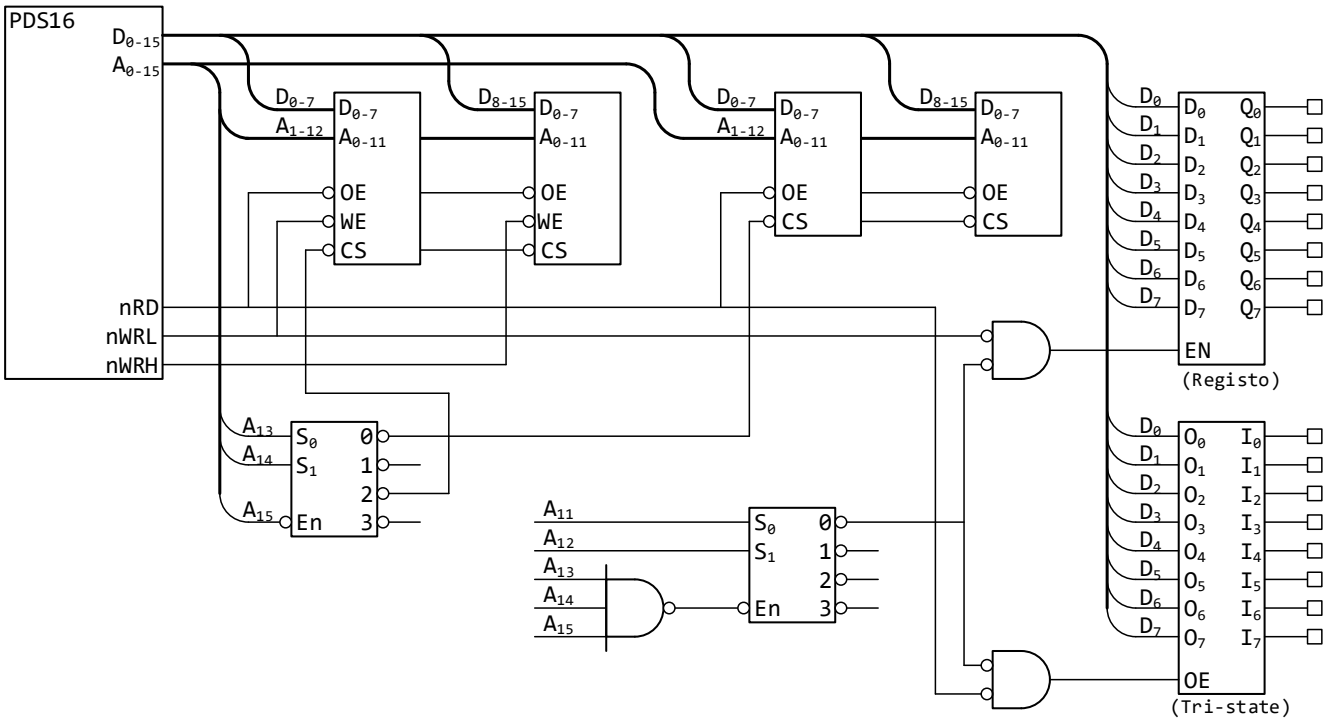
c) Considerando que o módulo `Descodificador Instruções` é implementado usando exclusivamente uma ROM, indique a programação da mesma. [1,5 val.]

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 `signExt`. [0,5 val.]

[Escreva aqui]

**[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 entradas e saída ocupam no espaço de endereçamento? [1,5 val.]

b) Desenhe o esquema de um módulo de RAM adicional com 8Kbyte de dimensão e ocupando a gama de endereços 0x2000-0x3FFF, usando módulos RAM de 4Kx8. [2 val.]

c) Qual o número mínimo de acessos à memória, do tipo escrita, necessários para preencher a memória com o valor zero na gama de endereços 0x2020-0x207F? Justifique. [0,5 val.]

d) Escreva um troço de programa que coloque o valor 0x53 no porto de saída. [1 val.]

[Escreva aqui]

**[3] Considerando as convenções definidas para a passagem de parâmetros, retorno de valores e preservação de registos e que os tipos uint8 e uint16 representam inteiros sem sinal a 8 bits e a 16 bits, respetivamente, considere as definições seguintes:**

```
uint8 pwr2_data[] = { 0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80 };

uint16 power2( uint8 pos ){
   uint16 res = pwr2_data[ pos & 0x07 ];
   if( pos >= 8 ){
       res <<= 8;
   }
   return res;
}

uint8 count_ones( uint16 value ){
   uint8 res = 0;
   for( uint8 i = 0; i != 16; ++i ){
       if( ( value & power2( i ) ) != 0 ){
             ++res;
       }
   }
   return res;
}
```

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

a) Escreva a definição do array `pwr2_data`. Traduza para assembly do PDS16 a função `power2` e defina as respetivas variáveis, se necessário [2,5 val.]
b) Traduza para *assembly* do PDS16 a função `count_ones` e defina as respetivas variáveis, se necessário. [2,5 val.]

[Escreva aqui]

**[4] Tendo como base o sistema SDP16, pretende-se implementar o sistema de controlo de semáforo de uma passagem de nível de via-férrea de sentido único, conforme ilustra a figura, com a seguinte especificação:**

O semáforo deve estar verde, até que o sensor S1 seja ativado, indicando o início do comboio. Neste momento, acende o semáforo vermelho e apaga o verde.

O semáforo deve manter-se vermelho até que o sensor S2 seja desativado, indicando o fim do comboio. Neste momento, o semáforo vermelho apaga e acende o verde, voltando o sistema ao estado inicial.

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