

Exercício das aulas 25 e 26

Usando o sistema ilustrado na Figura 1, baseado no processador P16, pretende-se implementar um programa que adquira o valor apresentado na entrada BCD, correspondente a um número inteiro representado com 4 bits em código *Binary Coded Decimal* (BCD), produza o código necessário à sua representação num mostrador de 7 segmentos de cátodo comum e afixe esse valor na saída SEG, onde o bit 0 corresponde ao segmento *a*, o bit 1 corresponde ao segmento *b* e assim sucessivamente.

A entrada TRG é utilizada para espoletar uma nova conversão, o que deve acontecer sempre que o seu valor lógico troca de zero para um.

O bit 7 da saída SEG deve apresentar o valor da entrada DOT a todo o momento, independentemente da ocorrência de transições $0 \rightarrow 1$ no valor da entrada TRG.

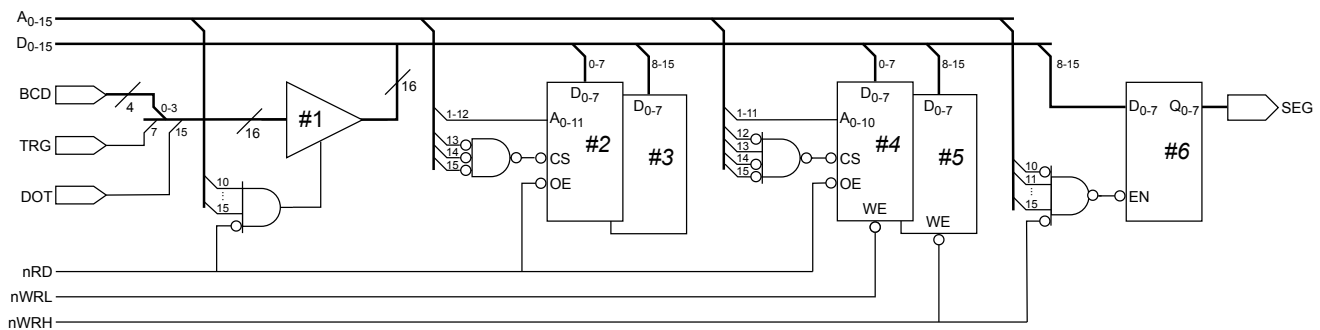


Figura 1: Diagrama lógico equivalente do sistema.

Caracterização do sistema

1. Caracterize os dispositivos de memória apresentados na Figura 1 quanto ao seu tipo, organização e capacidade, em bytes. Indique, ainda, as capacidades dos módulos que formam, também em bytes.
2. Caracterize os portos apresentados na Figura 1 quanto ao seu tipo, dimensão, em bytes, e modos de acesso suportados, i.e. *word-wise* e/ou *byte-wise*.
3. Desenhe o mapa de endereçamento do sistema, nele inscrevendo os tipos, as capacidades e os endereços de início e de fim do espaço atribuído a cada dispositivo. Se for o caso, indique também a ocorrência de subaproveitamento ou de *fold-back* e a localização de eventuais zonas interditas (também designadas por "conflito").

Desenvolvimento do gestor de periférico do porto de entrada ¹

4. Implemente, em linguagem *assembly* do P16, a função

```
uint16_t inport_read( );
```

que devolve o valor atual do estado dos bits do porto de entrada.

¹Defina todos os símbolos e variáveis que entender necessários para a realização do ponto 4.

Desenvolvimento do gestor de periférico do porto de saída ²

5. Implemente, em linguagem *assembly* do P16, a função

```
void outport_set_bits( uint8_t pins_mask );
```

que atribui o valor lógico um aos pinos do porto de saída identificados com o valor um em `pins_mask`. O valor dos restantes bits não é alterado.

6. Implemente, em linguagem *assembly* do P16, a função

```
void outport_clear_bits( uint8_t pins_mask );
```

que atribui o valor lógico zero aos pinos do porto de saída identificados com o valor um em `pins_mask`. O valor dos restantes bits não é alterado.

7. Implemente, em linguagem *assembly* do P16, a função

```
void outport_write_bits( uint8_t pins_mask, uint8_t value );
```

que atribui aos bits do porto de saída identificados com o valor lógico um em `pins_mask` o valor dos bits correspondentes de `value`. O valor dos restantes bits não é alterado.

8. Implemente, em linguagem *assembly* do P16, a função

```
void outport_init( uint8_t value );
```

que faz a iniciação do porto, atribuindo o valor `value` aos seus bits.

Desenvolvimento do programa ²

9. Implemente, em linguagem *assembly* do P16, o programa pedido.

Verificação da funcionalidade do programa

10. Descreva o mapa de endereçamento do sistema usando a sintaxe da ferramenta `p16sim`. Guarde a descrição num ficheiro de texto com o nome "ap05_p16sim_config.txt"
11. Teste o funcionamento do programa utilizando a ferramenta `p16sim`, com a configuração definida no ficheiro "ap05_p16sim_config.txt", e a ferramenta `p16dbg`.

²Defina todos os símbolos e variáveis que entender necessários para a realização dos pontos ⁵ a ⁹.