

Exercício das aulas 22 e 23

A Figura 1 apresenta o subsistema de memória de um sistema computacional baseado no processador P16 em que se verificou existirem alguns problemas na sua utilização. Para corrigir esta situação, pretende-se alterar o subsistema para torná-lo completamente funcional e aumentar a capacidade de memória disponível.

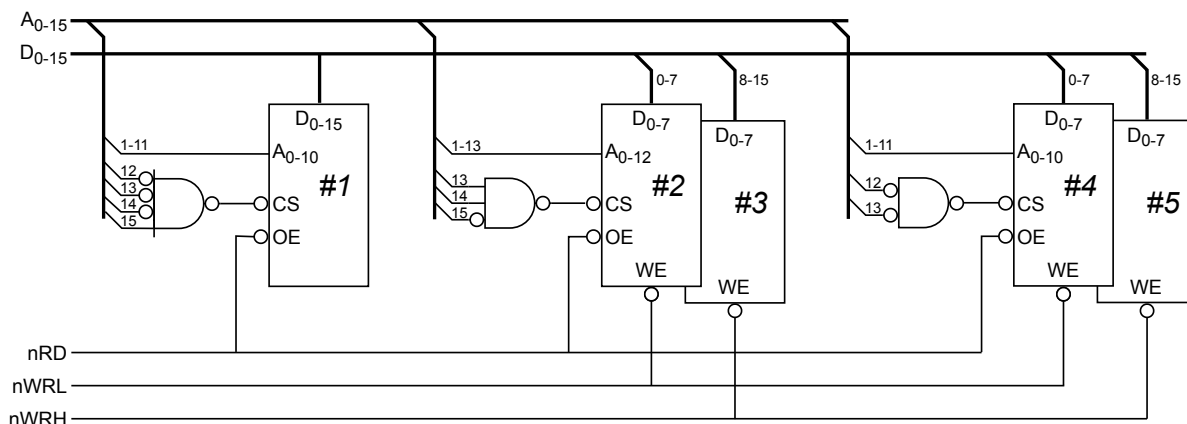


Figura 1: Diagrama lógico do subsistema de memória.

Caracterização do subsistema de memória

1. Indique o tipo, a organização e a capacidade, em bytes, dos módulos de memória identificados como #1, #2, #3, #4 e #5. Justifique a sua resposta.
2. Indique a capacidade, em bytes, dos pares de módulos de memória identificados como #2 e #3 e #4 e #5. Justifique a sua resposta.
3. Desenhe o mapa de endereçamento do conjunto, nele inscrevendo os tipos, as capacidades e os endereços de início e de fim do espaço atribuído a cada um dos módulos de memória. 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").

Análise do funcionamento do subsistema de memória

4. Numa tabela com o formato indicado na Tabela 1, indique a atividade dos barramentos e dos sinais em referência, quando observados passo-a-passo, para a execução do troço de código apresentado na Listagem 1 sobre o sistema apresentado Figura 1, onde o símbolo `var2.addr` tem o valor 32.

Considere que o troço de código apresentado é localizado em memória a partir do endereço 0x6100 e os seguintes valores iniciais para os registos do processador: R5=0x8020, R7=0x1234, R8=0x5678, SP=0x4000 e PC=0x6100.

```

        .text
        push pc
        ldr r0, var1_addr
        ldrb r1, [r0, #0]
        ldrb r2, [r0, #1]
        mov r3, #var2_addr
        ldr r4, [r3, #0]
        ldr r6, [r5, #0]
        strb r7, [r5, #0]
        strb r8, [r5, #1]
        ldr r9, [r3, #0]
        pop r10
        b .
var1_addr:
        .word      0x4020

        .data
var1:
        .word      0xABCD

```

Listagem 1: Excerto de ficheiro `.lst` com o troço de código objeto de estudo.

Instrução	Controlo			Endereço	Dados
	nRD	nWRH	nWRL	A15 ... A0	D15 ... D0
<code>push pc</code>					

Tabela 1: Tabela exemplo para o registo da atividade nos barramentos do processador. Genericamente, no barramento de dados pode ocorrer um valor concreto, alta impedância (**z**) ou conflito (**conf**).

Evolução do subsistema de memória

5. Apresente o mapa de endereçamento da nova versão do subsistema de memória, que deverá cumprir os seguintes critérios:
 - Utilizar todos os módulos de memória já existentes;
 - Acrescentar 8 KB de memória do tipo *Read Only Memory* (ROM);
 - Garantir que a dimensão do espaço atribuído a cada módulo de memória é coincidente com a sua capacidade;
 - Utilizar gamas de endereços contíguas para localizar os módulos de memória do mesmo tipo;
 - Impedir a existência de zonas interditas;
 - Assegurar a execução do programa imediatamente após a ligação da energia elétrica.
6. Indique as novas expressões lógicas dos sinais *chip select* (CS) dos módulos de memória identificados como #1, #2, #3, #4 e #5 e da nova ROM.

7. Apresente o diagrama correspondente à instalação da nova ROM, escolhendo os módulos de memória com a organização que considerar mais adequada, de entre os seguintes: $4\text{ K} \times 4$, $4\text{ K} \times 8$, $8\text{ K} \times 8$, $8\text{ K} \times 16$. Justifique a sua resposta.

Verificação do funcionamento do subsistema de memória usando a ferramenta P16Simulator

8. Descreva o mapa de endereçamento da configuração original do subsistema de memória, obtido no ponto 3, usando a sintaxe da ferramenta `p16sim`. Guarde a descrição num ficheiro de texto com o nome `"ap04_p16sim_config.txt"`

9. Execute a ferramenta `p16sim` com a configuração definida no ficheiro `"ap04_p16sim_config.txt"`, escrevendo o seguinte comando numa "Prompt de Comando" do Windows:

```
p16sim.exe -c ap04_p16sim_config.txt
```

10. Usando a ferramenta `p16as`, proceda à geração de um ficheiro binário correspondente ao troço de código apresentado na Listagem 1, garantindo a localização da secção `text` a partir do endereço `0x6100` e da secção `data` no par de módulos de memória identificados como #2 e #3. Para tal, escreva o seguinte comando numa "Prompt de Comando" do Windows:

```
p16as.exe -s .text=0x6100 -s .data=0x4020 ap04.S
```

onde `"ap04.S"` é o ficheiro assembly com o troço de código apresentado na Listagem 1.

11. Utilize a ferramenta `p16dbg` para validar o conteúdo da tabela elaborada no ponto 4. Para tal, execute o programa passo-a-passo a partir do endereço `0x6100` e verifique a alteração do conteúdo das vistas "Registers", "CPSR register" e, especialmente, "Memory content". Tenha em atenção as mensagens de aviso que surgem durante a execução do programa.