

INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Licenciatura em Engenharia de Eletrónica e Telecomunicações e de Computadores

e

Licenciatura em Engenharia Informática e de Computadores



3.º Trabalho Prático de Arquitetura de Computadores

Memória e Portos

Grupo 2

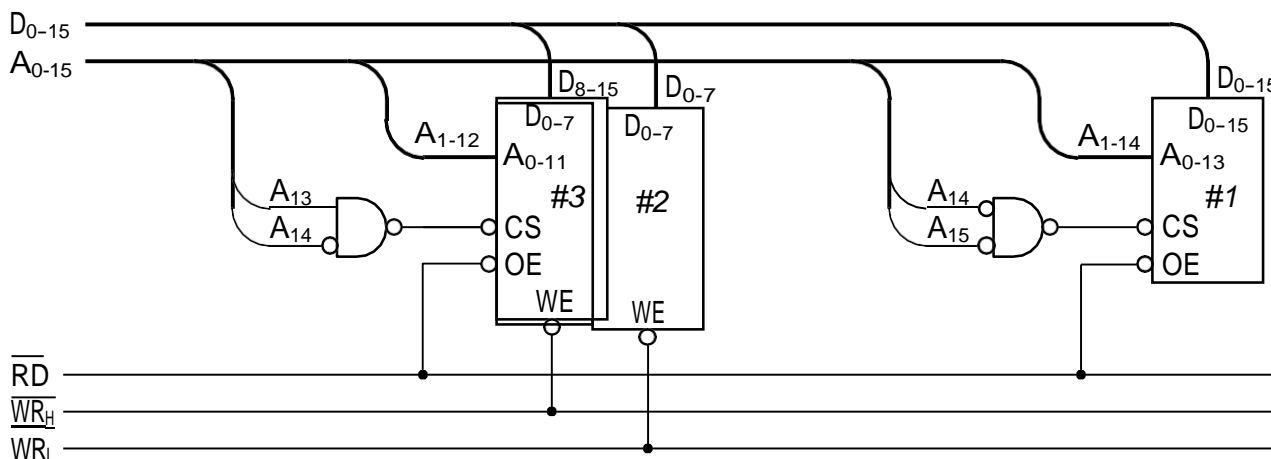
Paulo Rosa – 44873
Ricardo Pinto - 44808

1. Objetivos

Este trabalho tem como principal objetivo o estudo dos mecanismos de endereçamento usados pelos processadores no acesso aos dispositivos de memória e aos periféricos. Tomando como exemplo sistemas baseados no processador P16, aborda-se a utilização dos respectivos sinais de controlo e a geração dos sinais de seleção de endereços atribuídos aos dispositivos envolvidos, com destaque para a sua representação nos mapas de endereçamento, também chamados mapas de memória.

2. Trabalho a realizar

Considere o diagrama apresentado na figura, que descreve um exemplo de descodificação de endereços em torno de um processador P16, com alguns problemas no mapeamento em memória dos dispositivos.



- 2.1 Indique os tipos e as dimensões dos dispositivos #1 a #3, individualmente tomados, e as capacidades (em bytes) dos grupos que formam.

Dispositivo	Tipo	Dimensões(Lin e Col)	Capacidade(em bytes)
#1	ROM	$2^{14}=16K*16$	$(16K*16)/8 = 32KB$
#2, #3	RAM	$2^{12}=4K*8$	4KB(cada parte, 1 indv)

- 2.2 Desenhe o mapa de endereçamento do conjunto, indicando os tipos, as dimensões e os endereços de início e de fim do espaço atribuído a cada grupo de dispositivos, inscrevendo igualmente, se for o caso, a ocorrência de subaproveitamento ou de *fold-back* e a localização de eventuais zonas livres e de zonas interditas (também designadas por “conflito”).

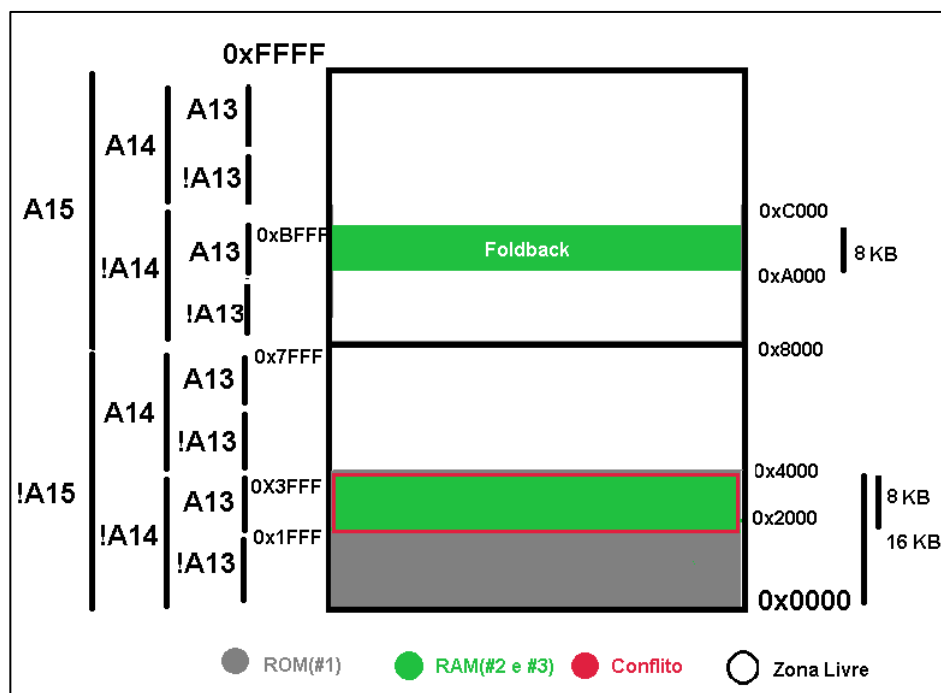
O CS do dispositivo ROM(#1) é ativado quando $A_{15} = 0$ e $A_{14} = 0$. $CS = \overline{A_{15}} \cdot \overline{A_{14}}$

O CS do dispositivo RAM(#2 e #3) é ativado quando $A_{14} = 0$ e $A_{13} = 1$. $CS = \overline{A_{14}} \cdot A_{13}$

Logo, vai haver uma zona interdita em $\overline{A_{15}} \cdot \overline{A_{14}} \cdot A_{13}$

ROM(#1) vai de 0x0000 até 0x3FFF, a RAM(#2 e #3) vai de 0x2000 até 0x3FFF(que é também onde se encontra a zona de conflito com a ROM) e o seu foldback vai de 0xA000 até 0xBFFF.

Existe subaproveitamento da parte da ROM. Esta tem a capacidade de 32KB, mas por causa da limitação do endereçamento, só podemos usar até 16KB.



Mapa de endereçamento

- 2.3 Seguindo o exemplo apresentado na tabela, registre a atividade dos barramentos e dos sinais em referência, quando observados passo-a-passo (*single step*), para a execução do seguinte troço de código sobre o sistema apresentado e considerando os seguintes valores iniciais: $r0 = 0x5070$; $r1 = 0x1040$; $r2 = 0x0204$; $r3 = 0xFEDC$; $r4 = 0x2030$; $pc = 0x1234$.

do_this:

```
ldr    r0, [r1, r2]
mov    r3, r0
ldr    r0, #var1_addr
str    r2, [r1, r2]
b      and_that
mov    r1, #0x70
movt   r1, #0x50
```

and_that:

```
mov    r2, #0x08
movt   r2, #0x06
strb   r2, [r4]
str    r2, [r4]
ldr    r0, [r4]
b      -
```

var1_addr:

```
.word 0x3020
```

CTRL			ADDR	DATA	instruction
nWRH	nWRL	nRD	A15 ... A0	D15 ... D0	
H	H	L	1234	1110	ldr r0, [r1, r2]
H	H	L	1244	7062	
H	H	L	1236	B003	mov r3, r0
			1238		

Tabela 1 – Atividade dos barramentos para observação passo-a-passo.

Nota – genericamente, no barramento de dados pode ocorrer: um valor concreto; alta impedância – ZH; ou conflito – conf.

do_this:
0x0000 - ldr r0, [r1, r2]
0x0002 - mov r3, r0
0x0004 - ldr r0,
#var1_addr
0x0006 - str r2, [r1, r2]
0x0008 - b and_that
0x000A - mov r1, #0x70
0x000C - movt r1, #0x50
and_that:
0x000E - mov r2, #0x08
0x0010 - movt r2, #0x06
0x0012 - strb r2, [r4]
0x0014 - str r2, [r4]
0x0016 - ldr r0, [r4]
0x0018 - b .
var1_addr:
0x001A - .word 0x3020

CTRL			ADDR	DATA	instruction
nWRH	nWRL	nRD	A15 ... A0	D15 ... D0	
H	H	L	0000 ₁₂₃₄	1110	ldr r0, [r1, r2]
H	H	L	1244	7062	
H	H	L	0002	B003	mov r3, r0
H	H	L	0004	0D00	ldr r0, #var1_addr
H	H	L	001A	3020	
H	H	L	0006	3112	str r2, [r1, r2]
L	L	H	1244	0204	
H	H	L	0008	580E	b and_that
H	H	L	000E	6082	mov r2, #0x08
H	H	L	0010	7062	movt r2, #0x06
H	H	L	0012	2842	strb r2, [r4]
H	L	H	2030	0608	
H	H	L	0014	2042	str r2, [r4]
L	L	L	2030	0608	
H	H	L	0016	0040	ldr r0, [r4]
H	H	L	2030	0608	
H	H	L	0018	5818	b .

R0 = 0x7062

r3 = 0x7062

[pc, <imm6>] = 0x001A
0x004 + X = 0x001A
4 + X = 26
X = 22 = 0x0016; r0=0x3020

R2 = 0x0608

R4 = 0x2030

RO=0x0608(=R2)

2.4 Sabendo que se pretende redesenhar o sistema apresentado para passar a ser funcional, desenhe o novo mapa de endereçamento e apresente as novas expressões lógicas dos sinais de *chip select*, cumprindo os seguintes critérios:

- Manter o mapeamento da memória ROM;
- Acrescentar 8 KB de memória do tipo RAM;
- Acrescentar um porto de entrada e um porto de saída, ambos a 16 bits e com acesso *word-wise* e *byte-wise*;
- Garantir que a dimensão do espaço atribuído a cada memória é coincidente com a sua capacidade;
- Assegurar que os módulos de memória do mesmo tipo ficam em endereços contíguos entre si;
- Impedir a existência de zonas interditas (de “conflito”).

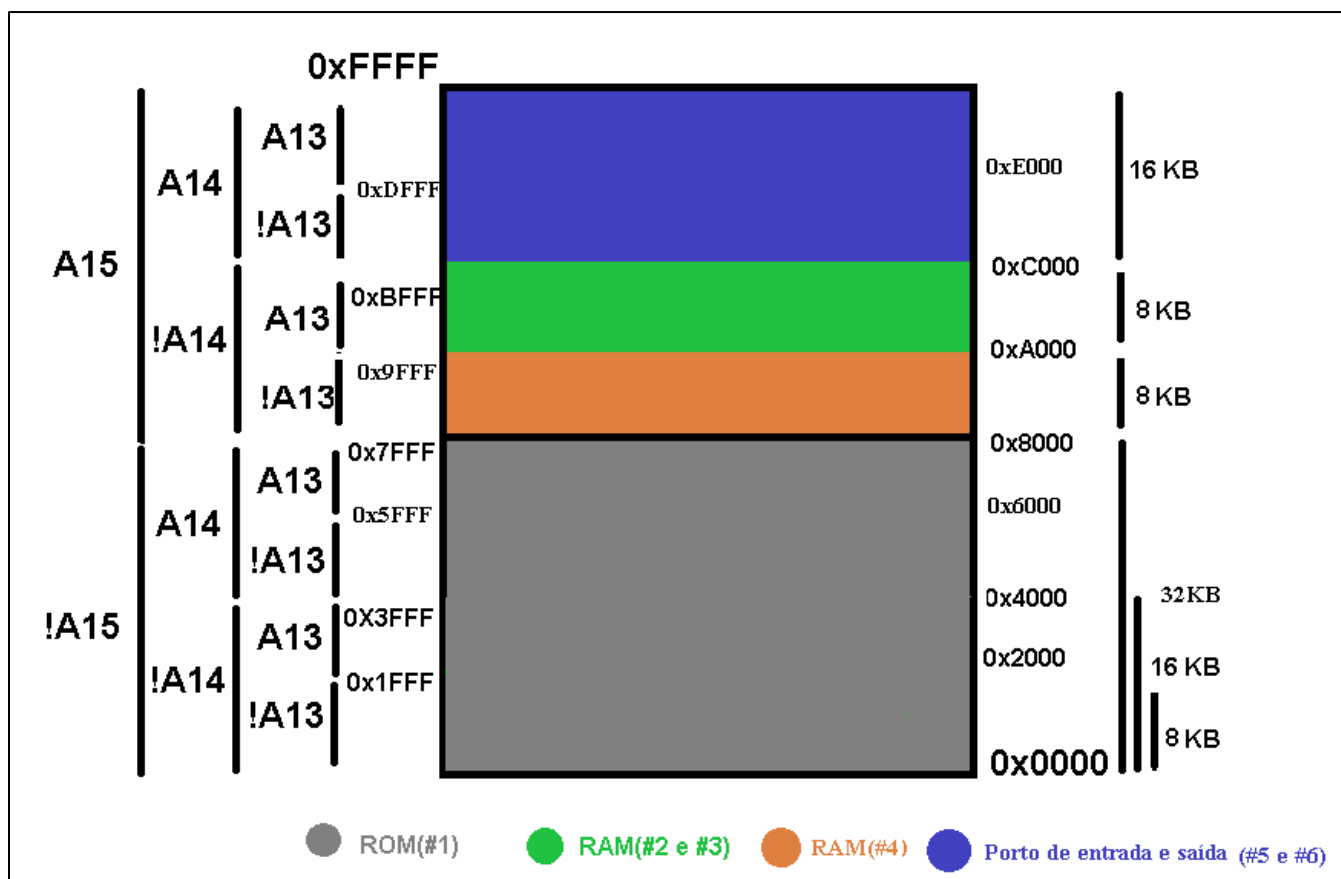
O dispositivo ROM(#1) é ativado quando: $CS = \overline{A15}$

O dispositivo RAM(#2 e #3) é ativado quando: $CS = A15 \bullet \overline{A14} \bullet A13$

O dispositivo RAM(#4) é ativado quando: $CS = A15 \bullet \overline{A14} \bullet \overline{A13}$

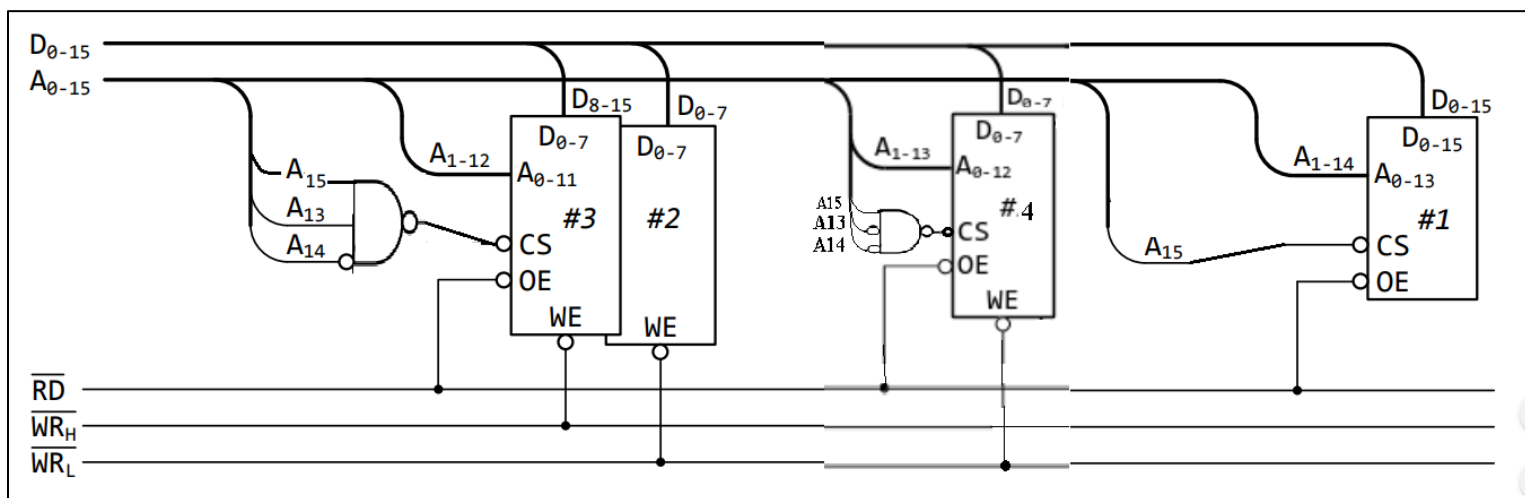
O dispositivo I/O(#5 e #6) é ativado quando: $CS = A15 \bullet A14$

Dispositivo	Tipo	Dimensões(Lin e Col)	Capacidade(em bytes)
#4	RAM	$2^{13}=8K*8$	8KB
#5,#6	I/O	$2^{12}=4K*16$	$4K*16/8 = 8KB$ (cada bloco, 1 indv)

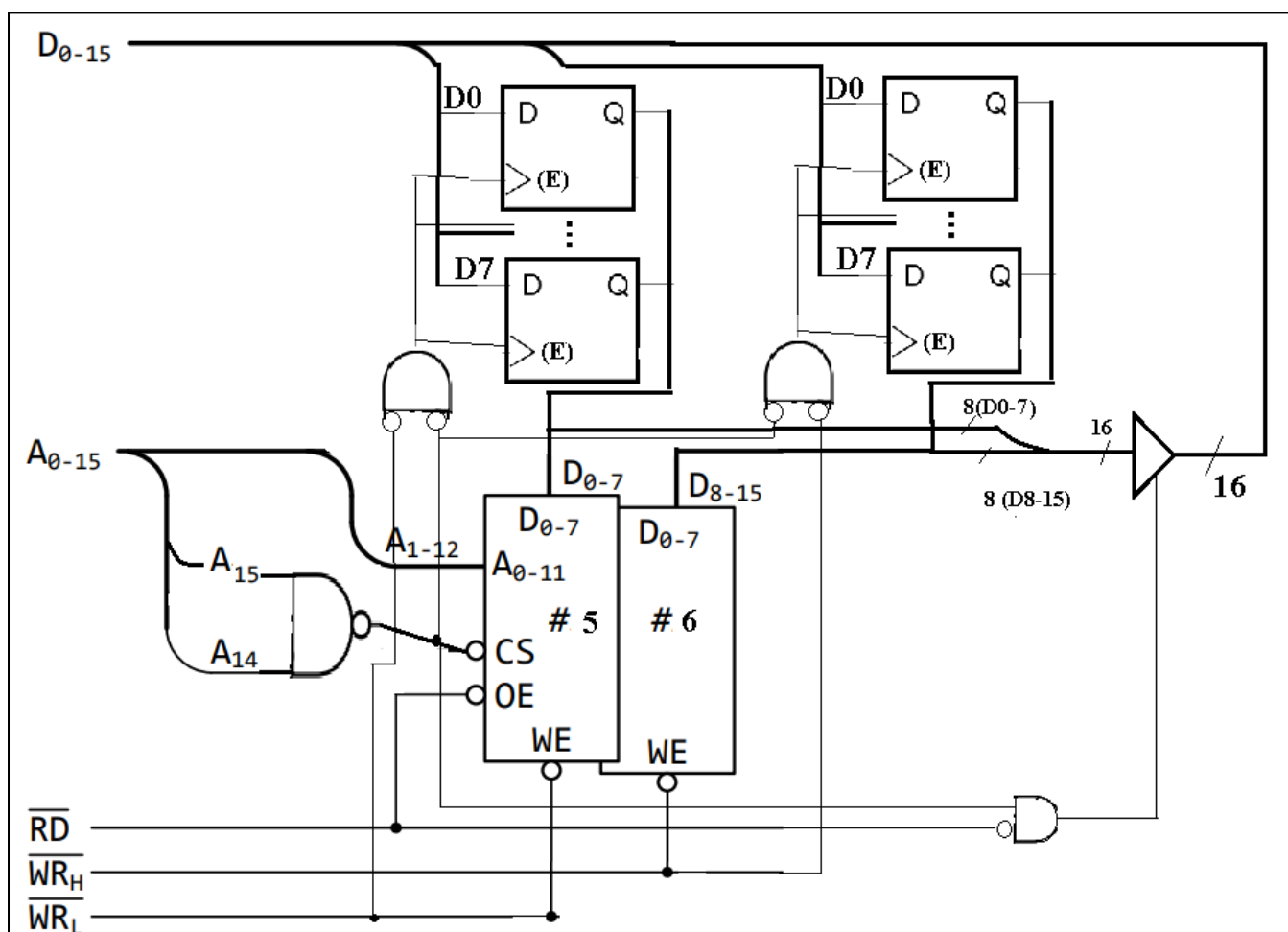


- 2.5 Desenhe o diagrama correspondente à instalação da nova memória, conforme a solução realizada no ponto anterior, escolhendo os dispositivos RAM que considerar mais adequados, de entre os seguintes:
 $4\text{ K} * 8$ | $8\text{ K} * 8$ | $8\text{ K} * 16$ | $16\text{ K} * 8$ | $16\text{ K} * 16$ | $32\text{ K} * 8$ | $32\text{ K} * 16$.

Escolhi dispositivo $8\text{ K} * 8$.



- 2.6 Desenhe o diagrama relativo aos portos a acrescentar, conforme a solução realizada no ponto 2.4, recorrendo aos circuitos que considerar mais adequados: registos do tipo *edge-triggered* ou *latch*, com 8 bits e 16 bits, e *tri-state buffers*, também com 8 bits e 16 bits.



Conclusão: Neste trabalho explorámos e aplicámos os conhecimentos dados em aula de acesso à memória, como ela interage com o CPU, que dados o CPU recebe e precisa da memória, como a podemos gerir e como a podemos usar e controlar. E também vimos a implementação de periféricos, para este, também, interagir com o CPU e as interfaces de controlo para que o sistema funcione sem provocar curto circuito.

3. Avaliação

O trabalho é realizado em grupo, conta para o processo de avaliação da unidade curricular e tem a duração de duas semanas.

A apresentação da solução proposta por cada grupo decorre em sessão de laboratório, em data a combinar com o docente responsável pela leção das aulas da respetiva turma.

Após esta apresentação, cada grupo deverá entregar o relatório do trabalho ao docente, no qual deve constar:

- A descrição sucinta dos raciocínios e dos cálculos efetuados;
- Os mapas de endereçamento projetados e os esquemas dos circuitos resultantes.