

# INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

Licenciatura em Engenharia Informática e de Computadores



## Arquitetura de Computadores

### Relatório Trabalho 3

Trabalho realizado por:

Nome: André Monteiro

Nº 43842

Nome: Constança Costa

Nº 50541

Turma: LEIC24D

Docente: Rui Duarte

2022 / 2023 Semestre Verão

13 de maio de 2023

# Índice

## **1. Introdução**

## **2. Desenvolvimento do trabalho**

- 2.1.** Definição do mapa de endereços
- 2.2.** Caracterização da atividade dos barramentos
- 2.3.** Evolução da arquitetura
- 2.4.** Teste do sistema

## **3. Conclusão**

## **1 Introdução**

Neste relatório, iremos analisar o projeto de módulos de memória e de portas paralelos de entrada e de saída num sistema com o processador P16. Os mapas de endereçamento são importantes para que o processador possa aceder corretamente aos dispositivos de memória e periféricos, permitindo que as informações sejam armazenadas e recuperadas de maneira eficiente.

## 2 Desenvolvimento do trabalho

### 2.1. Definição do mapa de endereços

a)

- #1 – ROM porque não tem sinal de escrita. Capacidade  $= 2^{13} * 2 = 16 \text{ KB}$
- #2 & #3 – RAM pois tem sinal de escrita. Capacidade  $= 2^{12} * 1 + 2^{12} * 1 = 8 \text{ KB}$

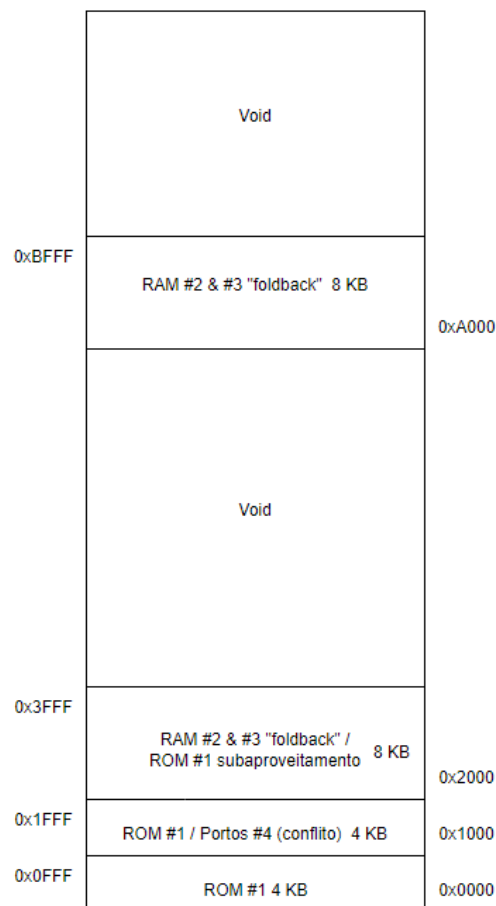
b)

#4 – porto de saída pois só escreve e não lê.

A sua dimensão é igual à capacidade, por isso 0001 bbbb bbbb bbbb, logo max = 0x1FFF e min = 0x1000, sendo assim max – min + 1 = 0x1000 que são 4 KB.

Modo de acesso é word-wise.

c)



d)

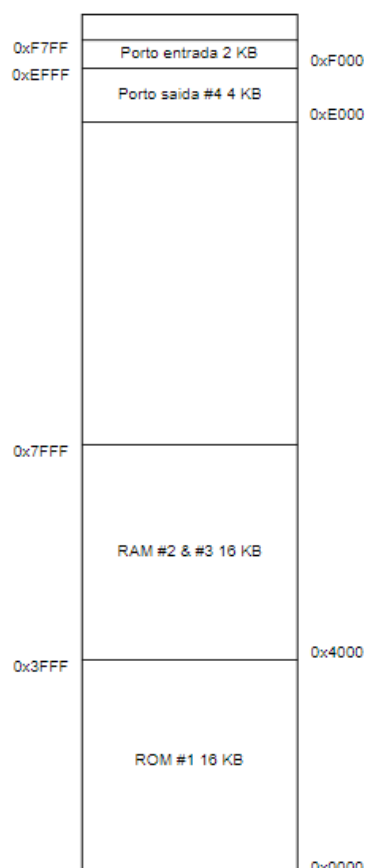
A afirmação é falsa pois a memória apresenta mais endereços do que aqueles a que temos “acesso”, isto é, subaproveitamento.

## 2.2 Caracterização da atividade dos barramentos

Instrução	Controlo			Endereço	Dados
	nRD	nWRH	nWRL		
ldr r0, sym	0	1	1	0x0000	0x0C60
	0	1	1	0x000E	0x0010
strb r2, [r1, r4]	0	1	1	0x0002	0x3A12
	1	0	1	0x1002	0x0155
push r1	0	1	1	0x0004	0x2401
	1	0	0	0xA000	0x1000
push r2	0	1	1	0x0006	0x2402
	1	0	0	z	0x0155
mov r0, r15	1	1	1	0x0008	0xB780
ldr r5, [r0, #0]	0	1	1	0x000A	0x0005
	0	1	1	0x000A	0x0005
pop r3	0	1	1	0x000C	0x0403
	1	0	0	z	xxxx

## 2.3 Evolução da arquitetura

a)



b)

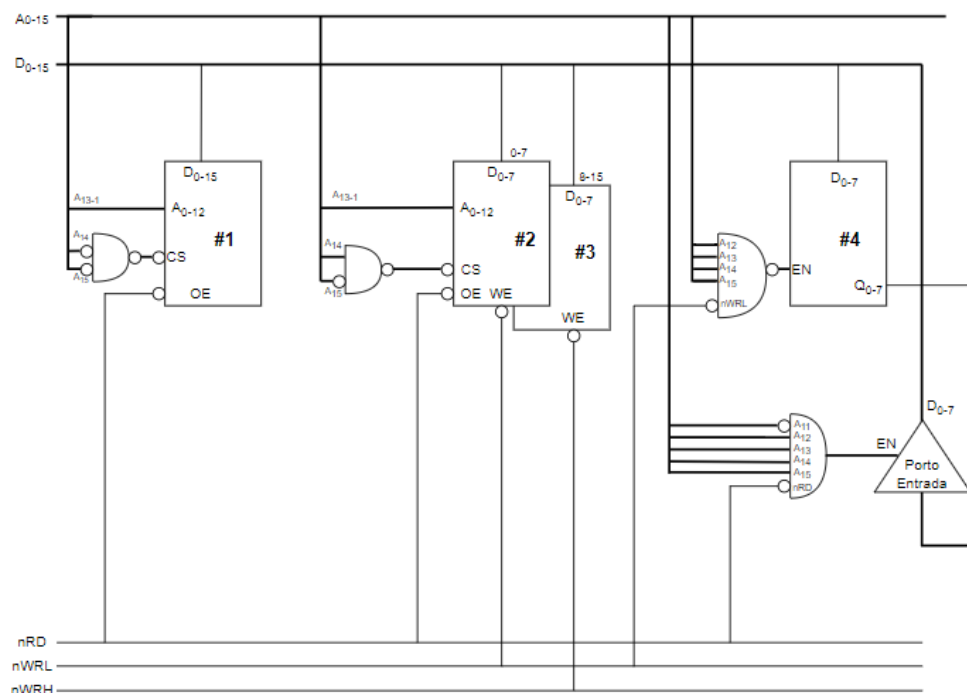
CS ROM #1 = not A15 and not A14

CS RAM #2 & #3 = not A15 and A14

CS porto entrada = A15 and A14 and A13 and A12 and not A11

CS porto saida #4 = A15 and A14 and A13 and not A12

c)



## 2.4 Teste do sistema

Este foi o código produzido pelo grupo de forma a testar o sistema:

```
.equ STACK_SIZE, 64
.equ INPORT_ADDRESS, 0xF000
.equ OUTPORT_ADDRESS, 0xE000

.section .startup
    b _start
    b .

_start:
```

```
ldr sp, tos_addr
mov r0, pc
add lr, r0, #4
ldr pc, main_addr
b .
```

```
tos_addr:
    .word tos
```

```
main_addr:
    .word main
```

```
main:
    ldr r0, in_port_addr
    ldrb r1, [r0, #0]
    mov r2, #0x01
    and r0, r1, r2
    bne change_output_port
    b main
```

```
change_output_port:
    lsr r1, r1, #1
    mov r0, #0x40
    and r2, r0, r1
    bne zero
    mov r1, #0x80
    add r0, r0, r1
    ldr r1, out_port_addr
    str r0, [r1, #0]
    b main
```

```
zero:
    mov r1, #0
    add r0, r0, r1
    b main
```

```
in_port_addr:
    .word INPORT_ADDRESS
```

```
out_port_addr:
    .word OUTPORT_ADDRESS
.data
```

```
.section .bss
```

```
.section .stack  
.space STACK_SIZE  
tos:  
    .word 0x8000 ; o stack começa no fim da RAM
```

### 3 Conclusão

Ao estudar os mecanismos de endereçamento, é possível entender como o processador P16 gere o acesso à memória e aos periféricos, tornando-se uma ferramenta fundamental para quem trabalha com sistemas embarcados e microcontroladores.