

Licenciatura em Engenharia Informática e de Computadores

Relatório do 3º Trabalho

*

Memória e Portos

Trabalho realizado por:

Romário Dias N° 50083

Iliano Santos N° 50096

Turma: 26D

Docente: Tiago Dias

Arquitetura de Computadores

2022 / 2023 verão

14 de maio de 2023

ÍNDICE

1	INTRODUÇÃO	3
2	DEFINIÇÃO DO MAPA DE ENDEREÇAMENTO.....	3
3	CARACTERIZAÇÃO DA ATIVIDADE DOS BARRAMENTOS	4
4	EVOLUÇÃO DA ARQUITETURA.....	5
5	TESTE DO SISTEMA	7
6	CONCLUSÕES.....	9

1 Introdução

O trabalho tem como objetivo principal o estudo dos diversos mecanismos de endereçamento usados pelos processadores no acesso aos dispositivos de memória e periféricos. Irão ser tratados conceitos como:

- Projeto de módulos de memória e de portos paralelos de entrada e de saída.
- Utilização dos respetivos sinais de controlo e a geração dos sinais de seleção de endereços atribuídos aos dispositivos envolvidos.
- Desenho de mapas de endereçamento / memória, demonstrando a sua devida utilização no processo, por exemplo, de deteção de falhas do sistema.

2 Definição do mapa de endereçamento

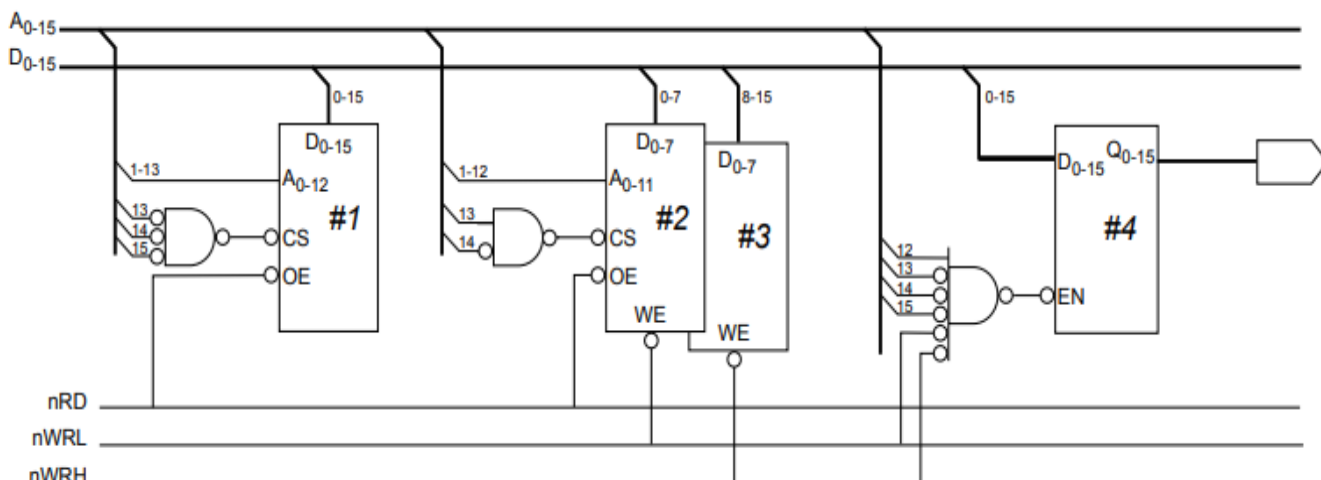


Figura 1: Subsistema de memória do sistema computacional em estudo

a) módulo #1: -tipo: ROM;

-Organização: $2^{13} = 2^3 * 2^{10} = 8 * 1K * 16 \text{ bits} = 8K * 16 \text{ bits}$

-Capacidade: $8K * 16 \text{ bits} = 8K * \frac{16}{8} \text{ B} = 16 \text{ KB}$

módulo #2, #3: -tipo: RAM;

-Organização: $2^{12} = 2^2 * 2^{10} = 4 * 1K = 4K * 8 \text{ bits}$

-Capacidade individual: $4K * 8 \text{ bits} = 4K * \frac{8}{8} \text{ B} = 4 \text{ KB}$

-Capacidade total: $= 4 \text{ KB} + 4 \text{ KB} = 8 \text{ KB}$

b) Na figura I, temos um porto paralelo de saída representado no módulo #4, porque temos um registo ligado à saída externa. Possui uma dimensão de 2 bytes e suporta os modos de acesso word-wise e byte-wise.

c) O diagrama deste circuito presente no enunciado apresenta uma zona de conflito, pois o porto de saída aparece no endereço onde se encontra o da memória ROM. Existe *foldback* na utilização da memória RAM dos dispositivos 2 e 3.

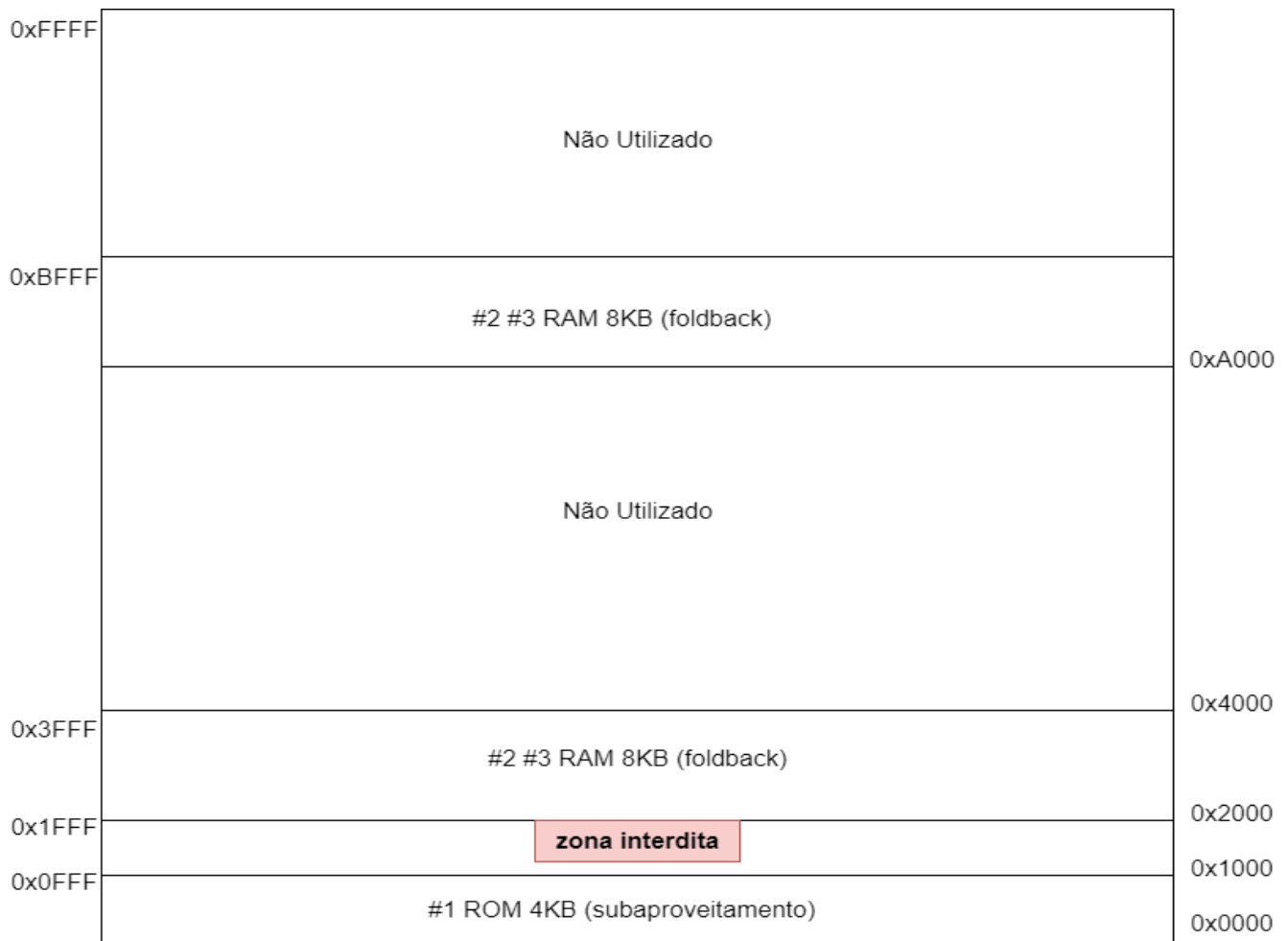


Figura 2: Mapa de memória da figura 2

d) "A capacidade de memória instalada no sistema é plenamente acessível."

Não, pois no valor *Chip Select* da ROM do sistema, devemos ter o bit 13 do barramento a nível lógico. A partir do momento que nós estamos o bit 13 em específico, para identificar o endereço a selecionar, este só será válido se for zero, e nunca a nível lógico 1. Também existe a ocorrência de subaproveitamento da memória ROM, dispositivo 1, que usa 4KB sendo que a sua capacidade é de 16KB. Portanto, devido a estas consideramos que este sistema não é acessível em toda a capacidade disponível, bem como no seu barramento de endereços.

3 Caracterização da atividade dos barramentos

Considerando valores iniciais para os registos do processador: R1=0x1000, R2=0x0155, R4=0x0003, SP=0xA002 e PC=0x0000, pretendemos demonstrar a atividade desses barramentos.

```

ldr    r0, sym
strb   r2, [r1, r4]
push   r1
push   r2
mov    r0, r15
ldr    r5, [r0, #0]
pop    r3
sym:
.word  val
val:
.word  0x4321

```

Figura 3: Código assembly em estudo

Instrução	Controlo			Endereço	Dados
	nRD	nWRH	nWRL	A15...0	D15...0
ldr r0, sym	L	H	H	0000	0C60
	L	H	H	000E	0010
strb r2, [r1, r4]	L	H	H	0002	3A12
	H	L	H	1003	5555
push r1	L	H	H	0004	2401
	H	L	L	A000	1000
push r2	L	H	H	0006	2402
	H	L	L	9FFE	0155
mov r0, r15	L	H	H	0008	B780
ldr r5, [r0, #0]	L	H	H	000A	0005
	L	H	H	000A	0005
pop r3	L	H	H	000C	0403
	L	H	H	9FFE	Z

*Z – Representa alta impedância

4 Evolução da arquitetura

a) Baseando-se nas debilidades do sistema anterior, procuramos, através de uma nova representação do mapa de memória, implementar um sistema melhor que seja uma solução possível a responder aos problemas identificados anteriormente, nomeadamente, as zonas de conflito, subaproveitamento e *foldback*, bem como garantir que a dimensão do espaço atribuído a cada módulo de memória é coincidente com a sua capacidade. Além disso introduziremos um porto paralelo de entrada acessível a 8 bits, na gama de endereços 0xF000 a 0xF7FF.

0xFFFF	#4 PORTO_SAÍDA 2KB	0xF800
0xF7FF	#5 PORTO_ENTRADA 2KB	0xF000
0xEFFF	Não Utilizado	0xC000
0xBFFF	#2 #3 RAM 8KB	0xA000
0x9FFF	Não Utilizado	0x8000
0x7FFF	#1 ROM 16KB	0x0000

Figura 4: Desenvolvimento de um mapa de endereços mais funcional

b) Tendo em conta o mapa da figura 4, as expressões lógicas dos sinais chip select são as seguintes:

Dispositivo #1 – CS – $\overline{A15}$

Dispositivo #2 / #3 – CS – $A15 \cdot \overline{A14} \cdot A13$

Dispositivo #4 – CS – $A15 \cdot A14 \cdot A13 \cdot A12 \cdot A11$

Dispositivo #5 – CS – $A15 \cdot A14 \cdot A13 \cdot A12 \cdot \overline{A11}$

$A_{0..15}$
 $D_{0..15}$

 nRD
 $nWAL$
 $nWRH$

Figura 5: Logigrama do porto paralelo de entrada

Conforme o que procuramos criar um código *assembly* que verificasse o valor do bit 0 continuamente após leitura do porto de entrada e caso for 1, faz uma uma extensão a 16 bits dos valores dos bits de 1 a 7, sabendo que estes definem um número inteiro com sinal.

Página 7

```
.word  main

.text

main:
    ldr r0, inport_addr
loop:
    bl    inport_read
    lsl r2, r0, #8
    asr r2, r2, #9
    bcs bit0_isSet
    b     loop

bit0_isSet:
    mov r0, r2
    bl outport_write
    b loop

inport_read:
    ldr    r1, inport_addr
    ldrb r0, [r1, #0]
    mov    pc, lr

inport_addr:
    .word  INPORT_ADDRESS

outport_write:
    ldr    r1, outport_addr
    strb r0, [r1, #0]
    mov    pc, lr

outport_addr:
    .word  OUTPORT_ADDRESS

.stack
.space  STACK_SIZE

tos:
```


6 Conclusões

Após realização do trabalho acreditamos ter alcançado os objetivos pretendidos, no entanto da nossa parte aprendemos mais e consolidamos nossos conhecimentos na matéria em causa.

Em suma, podemos concluir que o modelo de memória que aplicamos nos nossos processadores é de alta relevância e num mundo moderno, onde a cada dia se procura cada vez mais, por exemplo, computadores com um bom/alto desempenho, revela-se que a gestão das memórias é parte fundamental do processo. Pretendemos evitar problemas como o caso de subaproveitamento e de zonas de conflito que causem uma má funcionalidade do sistema.