

# ARQUITECTURA DE COMPUTADORES | 21010

Enunciado / resolução época normal 2023/24

## Grupo I (3 / 5 valores)

1. Considere uma função lógica  $F(A,B,C,D)$ , em que A é a variável de maior peso e D a variável de menor peso. A distribuição de mintermos (m) e indiferenças (md) da função  $F(A,B,C,D)$  é a seguinte:

$$\sum m(0,2,7,9,13) + \sum md(3,4,10,11)$$

AB\CD	00	01	11	10
00	1	0	X	1
01	X	0	1	0
11	0	1	0	0
10	0	1	X	X

1. a) [1.5] Construa o mapa de Karnaugh e simplifique a função de modo a obter uma soma de produtos.

AB\CD	00	01	11	10
00	1	0	X	1
01	X	0	1	0
11	0	1	0	0
10	0	1	X	X

$$\overline{A}B\overline{D} + \overline{A}C\overline{D} + \overline{A}C\overline{D}$$

1. b) [0.5, apenas exame] Duplique o mapa obtido na alínea anterior e simplifique a expressão de forma a obter um produto de somas.

AB\CD	00	01	11	10
00	1	0	X	1
01	X	0	1	0
11	0	1	0	0
10	0	1	X	X

$$(\overline{A} + \overline{C})(\overline{B} + D)(\overline{A} + D)(A + C + \overline{D})$$

**NOTA: Na sua resolução marque os laços utilizados no mapa, e faça corresponder cada termo da função resultante com o laço que lhe dá origem. Caso contrário a resposta não se considera justificada.**

2. Efetue as seguintes conversões entre bases numéricas:

2. a) **[0.5]** Represente o número A01h em base 8.

Hexadecimal: A01h

Binário: 1010.0000.0001b

Binário: 101.000.000.001b

Base 8: 5001<sub>8</sub>

2. b) **[0.5, apenas exame]** Represente o número 3524<sub>8</sub> em base 10.

Base 8: 3524

Fórmula:  $3 \cdot 8^3 + 5 \cdot 8^2 + 2 \cdot 8 + 4 = (24+5) \cdot 64 + 20 = 1876$

3. Efetue as seguintes conversões:

3. a) **[1]** Represente o número -17 em binário com 8 bits, utilizando a técnica de complemento para 2.

Conversão para binário:  $17 = 16 + 1 = 2^4 + 1 = 10001b$

Binário de 8 bits: 00010001b

Complemento: 11101110b

Mais 1: 11101111b

**3. b) [1, apenas exame]** Considere a seguinte norma, baseada na recomendação IEEE-754, mas adaptada para 16 bits: S=1, E=5, F=10; Número= $(-1)^S * 1,F * 2^{(E-15)}$ . Represente em notação decimal, o número: 0011100011100000

Número: 0011100011100000

S=0, E=01110, F=0011100000

Número =  $(-1)^0 * 1,00111 * 2^{(14-15)}$

Número =  $0,100111b = 1/2 + 1/16 + 1/32 + 1/64 = 0.609375$

### Grupo II (3 / 5 valores)

Considere a seguinte função lógica de três variáveis  $F(A,B,C)$ :

$$F(A, B, C) = \overline{\overline{B} + \overline{C} + \overline{A}} + \overline{\overline{C} + \overline{AB}} + \overline{\overline{A} + \overline{C}}$$

**1. [1.5]** Simplifique algebricamente a função  $F$ .

$$\overline{\overline{B} + \overline{C} + \overline{A}} + \overline{\overline{C} + \overline{AB}} + \overline{\overline{A} + \overline{C}} =$$

$$\overline{\overline{B} \cdot \overline{\overline{C} + \overline{A}}} + \overline{\overline{C} \cdot \overline{AB}} + \overline{\overline{A} \cdot \overline{C}} =$$

$$B(C + A) + CAB + AC =$$

$$AB + BC + AC$$

**2. [1, apenas exame]** Indique uma expressão lógica que implemente a função  $F$  utilizando apenas portas NAND, desenhando o circuito correspondente.

$$AB + BC + AC =$$

$$\overline{\overline{AB} + \overline{BC} + \overline{AC}} =$$

$$\overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}}$$

Desenho do circuito, com as ligações:

$\text{NAND}(\text{NAND}(A,B), \text{NAND}(B,C), \text{NAND}(A,C))$

**3. [1, apenas exame]** Indique uma expressão lógica que implemente a função  $F$  utilizando apenas portas NOR, desenhando o circuito correspondente.

$$AB + BC + AC =$$

$$A(B + C) + BC =$$

$$(A + BC)(B + C + BC) =$$

$$(A + BC)(B + C) =$$

$$(A + B)(A + C)(B + C) =$$

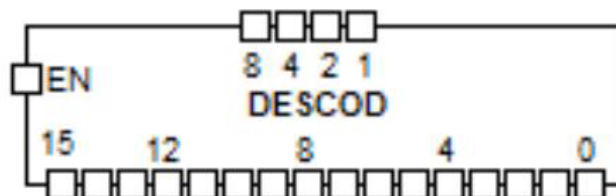
$$\overline{\overline{(A + B)(A + C)(B + C)}} =$$

$$\overline{\overline{A + B + A + C + B + C}}$$

Desenho do circuito, com as ligações:

$\text{NOR}(\text{NOR}(A,B), \text{NOR}(B,C), \text{NOR}(A,C))$

**4. [1.5]** Implemente a função recorrendo a um decodificador de 4 bits.



Desenho do circuito, com as ligações:

$A=4, B=2, C=1$  (ou inverso), ligando 8 à terra,  $\text{EN}=1$

$AB + BC + AC$  – extrair mintermos de CNF (poderia ser feito o mapa de Karnaugh)

$AB=4+2 = 6$ , (C indiferente  $6+1=7$ ),

$BC=2+1=3$  (A indiferente  $3+4=7$ ),

$AC=4+1=5$  (B indiferente  $5+2=7$ )

$OR(3,5,6,7)$

Nas resoluções em que a função é 1 devido má visualização da fórmula:

Entradas:  $A=4$ ,  $B=2$ ,  $C=1$ , 8 ligado à terra,  $EN=1$

Saídas:  $F=OR(0, \dots, 7)$

### Grupo III (3 / 5 valores)

Considere um sistema sequencial síncrono, com uma entrada e uma saída. A saída deverá ser 0 até que a entrada tenha ocorrido a sequência 110 ou 111. Após esse momento a saída deverá ser 1 até que a entrada ocorra a sequência 000.

Exemplo de funcionamento:

Entrada	1	0	0	<b>1</b>	<b>1</b>	<b>0</b>	0	1	<b>0</b>	<b>0</b>	<b>0</b>	0	1
Saída	0	0	0	0	0	1	1	1	1	1	0	0	0

A sequência 110 ocorreu uma vez, e nessa altura a saída passou a 1. A saída permaneceu com o valor 1 até que ocorreu a sequência 000.

**1. [2]** Determine o diagrama de estados.

Estados (máquina de Moore):

S0: nada detetado, saída a 0 --- 0>S0, 1>S1

S1: 1 detetado, saída a 0 --- 0>S0, 1>S2

S2: 11 detetado, saída a 0 --- x>S3

S3: 11x detetado, saída a 1 --- 0>S4, 1>S3

S4: 0 detetado, saída a 1 --- 0>S5, 1>S3

S5: 00 detetado, saída a 1 --- 0>S0, 1>S3

Estados (máquina de Mealy):

S0: nada detetado --- 0/0>S0, 1/0>S1

S1: 1 detetado --- 0/0>S0, 1/0>S2

S2: 11 detetado --- x/1>S3

S3: 11x detetado --- 0/1>S4, 1/1>S3

S4: 0 detetado --- 0/1>S5, 1/1>S3

S5: 00 detetado --- 0/0>S0, 1/1>S3

**2. [1] [e-fólio Global apenas]** Reproduza o exemplo de funcionamento para os valores da entrada, com o diagrama de estados da alínea 1. Assuma que o estado inicial é 0, e determine a saída e o número do estado a cada instante.

Entrada	1	0	0	1	1	0	0	1	0	0	0	0	1
Saída	0	0	0	0	0	1	1	1	1	1	0	0	0
Estado	S1	S0	S0	S1	S2	S3	S4	S3	S4	S5	S0	S0	S1

**2. [2, apenas exame]** Construa a tabela de transição de estados correspondente ao diagrama de estados.

Com 6 estados é preciso 3 variáveis de estado:

E	s2	s1	s0	S	s2	s1	s0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	1	1
0	0	1	1	1	1	0	0
0	1	0	0	1	1	0	1
0	1	0	1	1	0	0	0
0	1	1	0	X	X	X	X
0	1	1	1	X	X	X	X
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	0
1	0	1	0	0	0	1	1
1	0	1	1	1	0	1	1
1	1	0	0	1	0	1	1
1	1	0	1	1	0	1	1
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

**3. [1, apenas exame]** Simplifique as variáveis de saída.

$E, s_2 \setminus s_1, s_0$	00	01	11	10
00	0	0	1	0
01	1	1	X	X
11	1	1	X	X
10	0	0	1	0

$$s_2 \bar{s}_1 + s_1 s_0$$



### Grupo IV (3 / 5 valores)

1. [2, apenas exame] Indique as instruções, em assembly do P3, que implementam as seguintes funcionalidades:

1. a) Escreva em assembly do P3 uma instrução que: Coloca na posição de memória em "W" o conteúdo de R2

1. b) Escreva em assembly do P3 uma instrução que: Coloca em R1 o conteúdo da posição de memória em R2

1. c) Escreva em assembly do P3 uma instrução que: Salto condicional relativo para "label", se a última operação aritmética não gerou transporte

1. d) Escreva em assembly do P3 uma instrução que: Subtrai R2 a R1

a) MOV M[W], R2

b) MOV R1, M[R2]

c) JMP.NC label

d) CMP R1, R2

2. [3] Elabore uma sub-rotina no assembly do P3 que receba no registo R1 o valor de  $n$ , no registo R2 o valor de  $r$ , e retorne no registo R3 o resultado da função EP1A3:

```
Function EP1A3(N, R)
    res = 1
    For i = N - R + 1 To N
        res = res * i
    Next
    EP1A3 = res
End Function
```

O valor  $n$  e  $r$  é um inteiro positivo, tal como todas as variáveis na função. Caso o resultado não possa ser armazenado num registo, retorne FFFFh.

Exemplo:

R1 = 5

R2 = 3

Pretende-se que efetue o produto desde  $5-3+1=3$  até 5, ou seja,  $3*4*5$ . R3 deverá ficar com o valor 60. Caso este valor não pudesse ser representado num só registo, R3 ficaria com o valor FFFFh.

```
EP1A3:    MOV R3, 1      ; R3 fica logo com o resultado
          MOV R4, R1    ; R4 a variável iteradora i=N
          SUB R4, R2    ; i=N-R
Ciclo:    INC R4        ; próximo elemento a iterar
          PUSH R4      ; guardar para recuperar a variável
          MUL R4, R3   ; produto, se bem sucedido R4=0
          CMP R4, R0   ; teste se R4 é vazio
          JMP.NZ Falha ; falha, valor demasiado elevado
          POP R4       ; recuperar i
          CMP R4, R1   ; ver se i já chegou a N
          JMP.NZ Ciclo ; ainda não, continua
Fim:      RET ; retorna, está o resultado em R3 conforme solicitado
Falha:    MOV R3, FFFFh ; valor de erro a retornar
          JMP Fim
```

; Nota: em MUL O bit de estado Z é atualizado de acordo com o resultado, os restantes ficam a 0. O overflow tem de ser verificado no contexto do exercício, do número não caber num só registo, pelo que o primeiro operando é distinto de 0.



## Anexo

### Primeiras potências de 2:

1	2	4	8	16	32	64	128
256	512	1024	2048	4096	8192	16384	32768

### Conjunto de Instruções do Processador P3:

Aritméticas	Lógicas	Deslocamento	Controle de Fluxo	Transferência de Dados	Diversas
NEG INC DEC ADD	COM AND OR XOR	SHR SHL SHRA SHLA	BR BR.cond JMP JMP.cond	MOV MVBH MVBL XCH	NOP ENI DSI STC
ADDC SUB	TEST	ROR ROL	CALL CALL.cond	PUSH POP	CLC CMC
SUBB CMP MUL DIV		RORC ROLC	RET RETN RTI INT		

### Conjunto de Condições de Salto:

Condição	Mnemónica
Zero	Z
Não Zero	NZ
Transporte ( Carry )	C
Não Transporte	NC
Negativo	N
Não Negativo	NN
Excesso ( Overflow )	O
Não Excesso	NO
Positivo	P
Não Positivo	NP
Interrupção	I
Não Interrupção	NI