

Grupo I (5 valores)

1. [2] Considere o seguinte mapa de Karnaugh da função $F(A,B,C,D)$. Simplifique a função de modo a obter uma soma de produtos, e um produto de somas.

		Soma de Produtos						Produto de Somas			
CD		00	01	11	10	CD		00	01	11	10
AB						AB					
00		0	X	0	0	00	0	X	0	0	
01		0	X	X	1	01	0	X	X	1	
11		1	X	0	1	11	1	X	0	1	
10		1	1	0	1	10	1	1	0	1	

NOTA1: O valor x na tabela corresponde a uma indiferença (don't care).

NOTA2: Na sua resolução marque os laços utilizados no mapa acima, e faça corresponder cada termo da função resultante com o laço que lhe dá origem. Caso contrário a resposta não se considera justificada.

O primeiro passo é marcar os laços nos mapas, no caso da soma de produtos devem-se incluir todos os 1's em laços que contenham o maior número possível de casas (podem ser 1's ou X, indiferenças, que quando são incluídas nos laços passam a ser consideradas 1's).

No caso do produto de somas marcam-se os laços de forma a considerarem todos os 0's, devendo também os laços incluir o maior número de 0's possível.

Os laços são neste caso:

1. [2] Considere o seguinte mapa de Karnaugh da função $F(A,B,C,D)$. Simplifique a função de modo a obter uma soma de produtos, e um produto de somas.

		Soma de Produtos						Produto de Somas			
CD		00	01	11	10	CD		00	01	11	10
AB						AB					
00		0	X	0	0	00	0	X	0	0	
01		0	X	X	1	01	0	X	X	1	
11		1	X	0	1	11	1	X	0	1	
10		1	1	0	1	10	1	1	0	1	

(Handwritten annotations: Circles and lines connecting cells in both maps. In the first map, circles are drawn around (0,1), (1,1), (1,0), (0,1), (1,1), (1,0) and a larger circle around (0,1), (1,1), (1,0), (0,1). In the second map, circles are drawn around (0,0), (0,1), (1,0), (1,1), (1,0), (1,1) and a larger circle around (0,0), (0,1), (1,0), (1,1).)

NOTA1: O valor x na tabela corresponde a uma indiferença (don't care).

NOTA2: Na sua resolução marque os laços utilizados no mapa acima, e faça corresponder cada termo da função resultante com o laço que lhe dá origem. Caso contrário a resposta não se considera justificada.

No caso da soma de produtos para englobar todos os 1's marcamos três laços, 1 e 2 com 4 1's e o 3 com 2 1's. para o caso dos laços 2 e 3 há duas formas possíveis de juntar os 1's. As alternativas são identificadas como 2' e 3'.

As expressões para cada laço são:

$$\text{Laço 1} = \overline{AD}$$

$$\text{Laço 2} = \overline{AC}$$

$$\text{Laço 2'} = \overline{CD}$$

$$\text{Laço 3} = \overline{BCD}$$

$$\text{Laço 3'} = \overline{ABC}$$

De notar que cada grupo de 2 1's corresponde às duas combinações de uma das variáveis de entrada pelo que a expressão vai ser o número de variáveis menos 1. É o caso dos laços 3, em que uma das quatro variáveis de entrada A, B, C ou D vai desaparecer da expressão. No caso de termos laços de 4 1's estes consistem nas quatro combinações possíveis de duas das variáveis, pelo que o laço reduz em duas variáveis, neste caso de 4 passamos a 4-2= 2.

A expressão final será a soma dos três produtos acima, devendo conter o laço 1, uma das possibilidades de laço 2 e uma das possibilidades de laço 3 . Um exemplo é:

$$F(A,B,C,D) = \overline{AD} + \overline{AC} + \overline{ABC}$$

Em relação ao produto de somas os laços são os que se encontram marcados na figura, a que correspondem as expressões:

$$\text{Laço 1} = A + B$$

$$\text{Laço 2} = \overline{C} + \overline{D}$$

$$\text{Laço 3} = A + C$$

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

2. Efectue as seguintes conversões entre bases numéricas:

2. a) [0.5] Represente o número F8h em base 8:

F – 1111

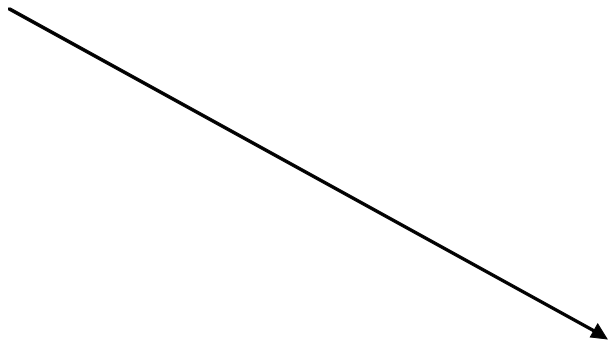
8 – 1000

$$F8h = 11\ 111\ 000 = 370 \text{ octal}$$

Agrupando os dígitos binários em grupos de três, começando pela direita e fazendo a identificação de cada grupo.

2. b) [0.5] Represente o número 903_{10} em base 2:

$$\begin{array}{r}
 903 \mid \underline{2} \\
 10 \quad 451 \mid \underline{2} \\
 03 \quad 05 \quad 225 \mid \underline{2} \\
 \underline{1} \quad 11 \quad 05 \quad 112 \mid \underline{2} \\
 \quad \quad \underline{1} \quad \underline{1} \quad 12 \quad 56 \mid \underline{2} \\
 \quad \quad \quad \quad \underline{0} \quad 16 \quad 28 \mid \underline{2} \\
 \quad \quad \quad \quad \quad \underline{0} \quad \underline{0} \quad 14 \mid \underline{2} \\
 \quad \quad \quad \quad \quad \quad \underline{0} \quad 7 \mid \underline{2} \\
 \quad \quad \quad \quad \quad \quad \quad \underline{1} \quad 3 \mid \underline{2} \\
 \quad \quad \quad \quad \quad \quad \quad \quad \underline{1} \quad \underline{1}
 \end{array}$$



O número em binário é dado pelos restos seguindo a ordem das setas:

$$\mathbf{1110000111}$$

←

2. c) [0.5] Represente o número 10110000_2 em base 10:

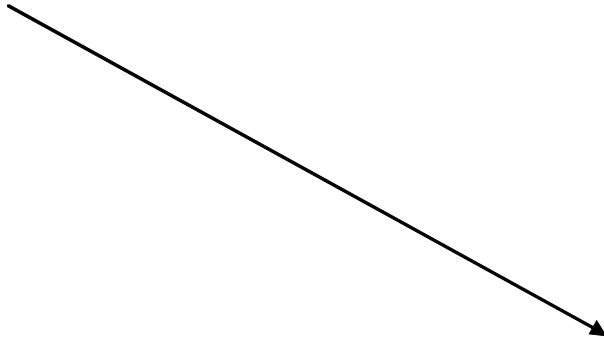
$$10110000_2 = 2^4 + 2^5 + 2^7 = 16 + 32 + 128 = 176$$

A ordem das potências começam a contar em zero da direita para a esquerda, correspondendo aos bits a 1.

3. Efectue as seguintes conversões tendo em atenção as considerações de cada alínea:
 3. a) [0.5] Represente o número -109 em binário com 8 bits, utilizando a técnica de complemento para 2.

Converter para binário:

$$\begin{array}{r}
 109 \mid \underline{2} \\
 09 \mid \underline{54} \mid \underline{2} \\
 \underline{1} \mid 14 \mid \underline{27} \mid \underline{2} \\
 \quad \underline{0} \mid 07 \mid 13 \mid \underline{2} \\
 \quad \quad \underline{1} \mid \underline{1} \mid \underline{6} \mid \underline{2} \\
 \quad \quad \quad \underline{0} \mid \underline{3} \mid \underline{2} \\
 \quad \quad \quad \quad \underline{1} \mid \underline{1} \mid \underline{2} \\
 \quad \quad \quad \quad \quad \underline{1} \mid \underline{0}
 \end{array}$$



O número em binário é dado pelos restos seguindo a ordem das setas:

1 1 0 1 1 0 1
 ←

Em oito bits:

0 1 1 0 1 1 0 1

Complementando e somando 1 obtém-se o número em complemento para 2:

$$\begin{array}{r}
 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \\
 + \quad \quad \quad \quad \quad 1 \\
 \hline
 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1
 \end{array}$$

3. b) [0.5] Represente o número 11000011 em notação decimal, considerando que tem seis dígitos inteiros e dois fraccionários.

Parte inteira: $32 + 16 = 48$

Parte fraccionária $2^{-1} + 2^{-2} = 0,5 + 0,25 = 0,75$

Valor decimal: 48,75

3. c) [0.5] 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: 1100011010000000

S = 1

E = 16 + 1 = 17

F = 1010000000

Nº em decimal = $-1 \times (2^0 + 2^{-1} + 2^{-3}) \times 2^{17-15} = - (1 + 0,5 + 0,125) \times 4 = -6,5$

Grupo II

Considere a seguinte função lógica f:

$$f(a,b,c) = a \cdot b + \bar{c} + (a \cdot b + \bar{a} \cdot c) \cdot \overline{(b \cdot c + c \cdot b)}$$

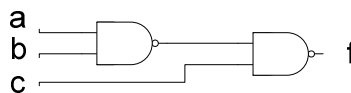
1. [1.5] Simplifique algebricamente a função f.

Dado que $\overline{(b \cdot c + c \cdot b)} = \bar{b}c + bc\bar{b} = 0$

$$f(a,b,c) = a \cdot b + \bar{c}$$

2. [1] Indique uma expressão lógica que implemente a função f utilizando apenas portas NAND

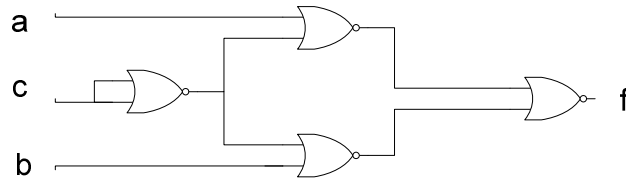
$$f(a,b,c) = a \cdot b + \bar{c} = \overline{\overline{a \cdot b + \bar{c}}} = \overline{\overline{a \cdot b} \cdot c} = \overline{\overline{a \cdot b} \cdot c}$$



3. [1] Indique uma expressão lógica que implemente a função f utilizando apenas portas NOR

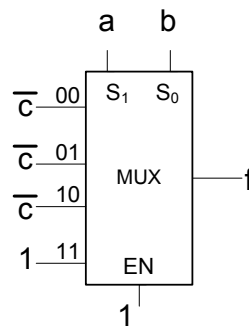
$$f(a,b,c) = a \cdot b + \bar{c} = (a + \bar{c}) \cdot (b + \bar{c}) = \overline{\overline{(a + \bar{c})} \cdot \overline{(b + \bar{c})}} = \overline{\overline{(a + \bar{c})} + \overline{(b + \bar{c})}}$$

(propriedade distributiva + leis de Morgan)



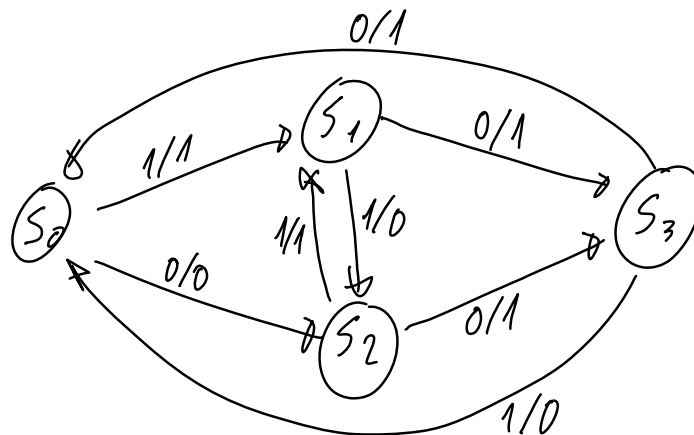
4. [1.5] Implemente a função recorrendo a um multiplexador de 2 variáveis de selecção.

$$f(a,b,c) = a \cdot b + \bar{c}$$



Grupo III

Considere o Diagrama de Estados seguinte:



Pretende-se construir um circuito digital síncrono que implemente este diagrama, utilizando básicas tipo D.

1. [2] Construa a tabela de transição de estados correspondente ao diagrama de estados.

4 estados => 2 variáveis de estado SV_1 SV_0

1 entrada I

1 saída O

Codificação dos Estados:

	SV_1	SV_0
S_0	0	0
S_1	0	1
S_2	1	0
S_3	1	1

Tabela de Transição de Estados:

Estado Actual			Estado Seguinte		
I	SV_1	SV_0	SV_1	SV_0	O
0	0	0	1	0	0
1	0	0	0	1	1
0	0	1	1	1	1
1	0	1	1	0	0
0	1	0	1	1	1
1	1	0	0	1	1
0	1	1	0	0	1
1	1	1	0	0	0

2. [2] Simplifique as variáveis de saída e de estado.

Apresentam-se os 3 mapas de Karnaugh, para SV_1 , SV_0 e para O construídos por inspecção da tabela anterior, e a simplificação das variáveis.

SV_1

$SV_1 \backslash SV_0$	00	01	11	10
I	0	1	0	1
1	0	1	0	0

②

- ① $\overline{SV_0} \cdot \overline{I}$
- ② $\overline{SV_1} \cdot SV_0$

SV_0

$SV_1 \backslash SV_0$	00	01	11	10
I	0	1	0	1
1	1	0	0	1

③

- ① $\overline{SV_0} I$
- ② $\overline{SV_1} \overline{SV_0}$
- ③ $\overline{SV_1} SV_0 \overline{I}$

$SV_1 \backslash SV_0$	00	01	11	10
I	0	1	1	1
1	1	0	0	1

- ① $\overline{SV_0} I$
- ② $SV_1 \overline{SV_0}$
- ③ $SV_0 \overline{I}$

3. [1] Desenhe o circuito digital pretendido.

Grupo IV

1. [1] Complete a tabela com as instruções em assembly do P3, que implementam a funcionalidade pretendida:

Funcionalidade	Instrução P3
Coloca na pilha o conteúdo de R1	PUSH R1
Chamada condicional à subrotina "rotina", se a última operação aritmética/lógica teve resultado não negativo (bit mais significativo a 0)	CALL.NN rotina
Coloca em R1 a soma de R1 com R2	ADD R1,R2
Coloca em R1 os seus bits deslocados quatro unidades para a esquerda	SHL R1,4
Coloca em R1 a disjunção dos bits de R1 com a constante "W"	OR R1,W

2. [2] Converta a seguinte função em C, em assembly do P3, assumindo que os argumentos são passados no Stack e o resultado é colocado no registo R1:

```
int SomaQuadrados(int n)
{
    int soma=0;
    for(int i=1;i<=n;i++)
        soma+=i*i;
    return soma;
}
```

```
SomaQuadrados: MOV R1,R0
                MOV R2, 1
ciclo:         CMP R2, M[SP+2]
                BR.P fim
                MOV R3,R2
                MOV R4,R2
                MUL R3,R4
                ADD R1,R4
                INC R2
                BR ciclo
fim:           RET
```

3. [2] Faça uma rotina em assembly do P3, assumindo que o Stack está inicializado e não tem problemas de limites. A rotina deve considerar que o registo R1 contem o início de uma string, terminada com 0. Pretende-se que a rotina coloque a string compactada na memória a começar no endereço no registo R2, com 2 letras em cada posição de memória, devendo para cada par de letras colocar a primeira letra nos 8 bits mais altos, e a segunda letra nos 8 bits mais baixos.

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

FIM