



Curso:

Prova de Arquitectura de Computadores (21010)

Data: 18 de Junho de 2010

Nome:

Nº de Estudante: B. I. nº

Assinatura do Vigilante:

RESERVADO PARA A *Universidade Aberta*

Classificação: ()

Prof. que classificou a prova:

LEIA ATENTAMENTE as instruções para a resolução do exame:

1. O tempo de resolução do exame é de duas horas, mais trinta minutos de tolerância.
2. **Não é permitida a utilização de calculadora** durante a execução do exame.
3. O exame é constituído por quatro Grupos e termina com a palavra **FIM**.
4. A cotação total de cada Grupo é de 5 valores, sendo a cotação de cada uma das questões indicada junto do enunciado da mesma, entre [].
5. As suas respostas devem ser claras, **indicando todos os passos seguidos na resolução de cada questão**. Resultados apresentados sem justificação poderão incorrer num desconto de $\frac{1}{2}$ da cotação total da questão.
6. A resposta a cada questão deve ser dada ocupando apenas o espaço destinado para o efeito.
7. Se o seu exemplar não estiver completo ou nele se verificar qualquer outra anomalia, por favor dirija-se ao professor vigilante.

Grupo I

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	AB	00	01	11	10
CD	AB	00	x	1	0	0	00	x	1	0	0	
00	00	x	1	0	0	0	01	x	1	x	1	
01	01	x	1	x	1	0	11	1	0	0	0	
11	11	1	0	0	0	0	10	x	0	1	0	
10	10	x	0	1	0	0	0	0	1	0		

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.

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

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

2. Efectue as seguintes conversões entre bases numéricas:

2. a) [0.5] Represente o número 201₁₆ em base 8:

$$\begin{array}{r}
 & 2 & 0 & 1 \\
 & 0010 & 0000 & 0001 \\
 + & 1 & 0 & 0 & 1 \\
 \hline
 & 1 & 0 & 0 & 1
 \end{array}
 \quad \begin{array}{l}
 \text{hexadecimal} \\
 \text{binário} \\
 \text{octal}
 \end{array}$$

$$201_{16} = 1001_8$$

2. b) [0.5] Represente o número 725₁₀ em base 2:

$$\begin{array}{r}
 725 \quad | 2 \\
 12 \quad | 362 \quad | 2 \\
 0 \quad | 181 \quad | 2 \\
 5 \quad | 0 \quad | 181 \quad | 2 \\
 1 \quad | 90 \quad | 2 \\
 0 \quad | 45 \quad | 2 \\
 1 \quad | 22 \quad | 2 \\
 0 \quad | 11 \quad | 2 \\
 1 \quad | 5 \quad | 2 \\
 1 \quad | 2 \quad | 2 \\
 0 \quad | 1
 \end{array}
 \quad 1011010101_2$$

2. c) [0.5] Represente o número 010100110000₂ em base 10:

$$\begin{aligned}
 010100110000_2 &= 2^4 + 2^5 + 2^8 + 2^{10} = \\
 &= 16 + 32 + 256 + 1024 = 1328
 \end{aligned}$$

$$\begin{array}{r}
 1024 \\
 256 \\
 32 \\
 + 16 \\
 \hline
 1328
 \end{array}$$

3. Efectue as seguintes conversões tendo em atenção as considerações de cada alínea:

3. a) [0.5] Represente o número -16 em binário com 8 bits, utilizando a técnica de complemento para 2.

$$\begin{array}{r}
 16 \rightarrow 00010000 \\
 + 11101111 \quad (\text{complemento}) \\
 \hline
 \underline{11110000} \quad (-16)
 \end{array}$$

$$\begin{array}{r}
 16 \mid 2 \\
 8 \mid 2 \\
 0 \mid 9 \mid 2 \\
 0 \mid 2 \mid 1
 \end{array}$$

3. b) [0.5] Represente o número 11001010 em notação decimal, considerando que tem cinco dígitos inteiros e três fraccionários.

$$\begin{aligned}
 11001,010_2 &= 2^4 + 2^3 + 2^{-2} = 16 + 8 + 0,25 = \\
 &= 24,25
 \end{aligned}$$

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: 0110110011000000

$$S=0 \quad E=11011 = 16+8+2+1 = 27$$

$$F=0011000000$$

$$\begin{aligned}
 \text{Número} &= -1^0 \cdot 1,0011_2 \cdot 2^{27-15} = 1,0011_2 \cdot 2^{12} = \\
 &= 2^{12} + 2^9 + 2^8 = 4096 + 512 + 256 = \underline{\underline{4864}}
 \end{aligned}$$

$$\begin{array}{r}
 4096 \\
 512 \\
 256 \\
 \hline
 4864
 \end{array}$$

Grupo II

Considere a seguinte função lógica f :

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

1. [1.5] Simplifique algebricamente a função f .

$$\begin{aligned} f &= a \left(a + \overline{b} + \overline{\overline{a}+c} \right) + \overline{b}\overline{c} + \overline{\overline{a}+\overline{b}} = \\ &= a + \cancel{a\overline{b}} + \cancel{a\overline{\overline{a}+c}} + b\overline{c} + \cancel{a\cdot\overline{b}} = a + b\overline{c} \end{aligned}$$

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

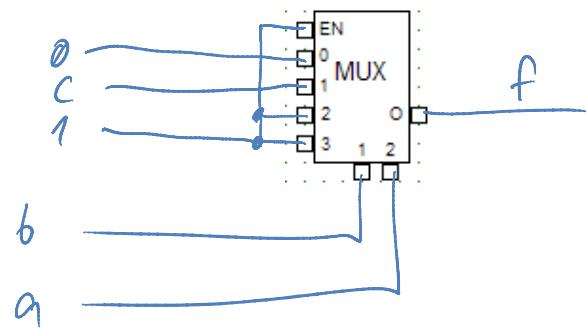
$$f = a + b\overline{c} = \overline{\overline{a+b\overline{c}}} = \overline{\overline{a} \cdot \overline{b\overline{c}}} = \overline{\overline{a} \cdot \overline{b} \cdot \overline{\overline{c}}} = \overline{\overline{a} \cdot \overline{b} \cdot \overline{c}}$$

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

$$\begin{aligned} f &= a + b\overline{c} = \overline{\overline{a+b\overline{c}}} = \overline{\overline{a} + \overline{b} + \overline{c}} = \\ &= \overline{\overline{\overline{a} + \overline{b} + \overline{c}}} + \overline{\overline{a} + \overline{b} + \overline{c}} \end{aligned}$$

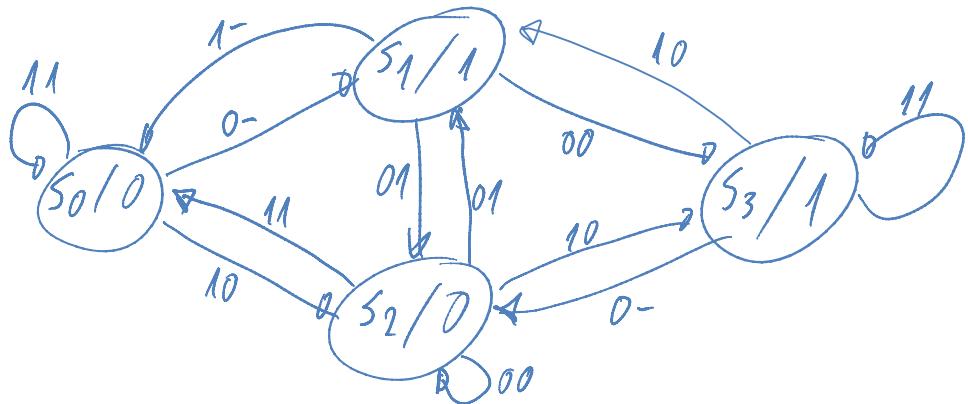
4. [1.5] Implemente a função recorrendo a um multiplexer de 2 variáveis de seleção.

$$f = a + b c$$



Grupo III

Considere o Diagrama de Estados seguinte:



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

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

(entradas: A;B estados: e2'; e1 saída: C)

A	B	e2	e1	e2	e1
0	0	0	0	0	1
0	0	0	1	1	1
0	0	1	0	1	0
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

e2	e1	C
0	0	0
0	1	1
1	0	0
1	1	1

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

$e_1 \setminus e_2$	0	1
0	0	0
1	1	1

$$C = e_1$$

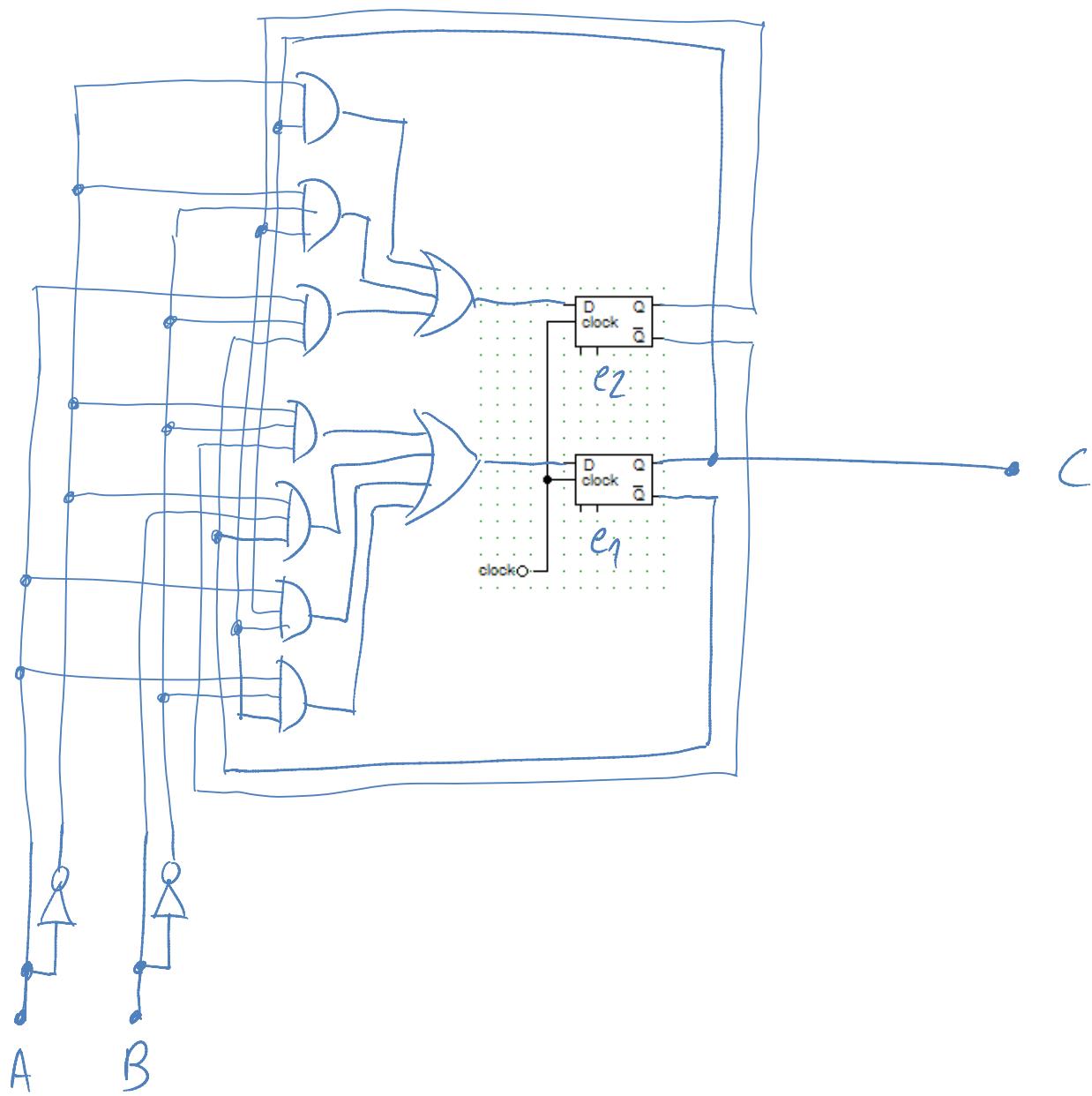
$e_2 \setminus e_1$	00	01	11	10
AB	00	0	1	1
00	0	1	1	0
01	0	1	1	0
11	0	0	1	0
10	D	0	0	1

$$e_2 = \bar{A} e_1 + \bar{A} \bar{B} e_2 + A \bar{B} \bar{e}_1$$

$e_2 \setminus e_1$	00	01	11	10
AB	00	1	1	0
00	1	1	0	0
01	D	0	0	1
11	0	0	1	0
10	0	0	1	1

$$e_1 = \bar{A} \bar{B} \bar{e}_2 + \bar{A} B \bar{e}_1 + A e_2 e_1 + A \bar{B} e_2$$

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 posição de memória em "W" o conteúdo de R2	MOV M[W], R2
Coloca na pilha o conteúdo da posição de memória em R1	PUSH M[R1]
Salto condicional relativo para "label", se a última operação aritmética/lógica não teve resultado positivo (portanto, ou negativo ou nulo)	BR.NP label
Coloca em R1 a disjunção exclusiva dos bits de R1 com os bits na posição de memória em R2	XOR R1, M[R2]
Coloca em R1 os seus bits deslocados uma unidade para a direita, sendo o bit perdido colocado no bit mais significativo	ROR R1

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:

len: MOV R1, R0
 MOV R2, M[SP+2]

len1: CMP M[R2], R0
 BR.Z lenFim

INC R1

INC R2

BR len1

lenFim: RET

```
int len(char *str)
{
    int resultado=0;
    while((*str) != 0)
    {
        resultado++;
        str++;
    }
    return resultado;
}
```

3. [2] Faça uma rotina em assembly do P3, que retorne em R1 o número de bits que os registos R2 e R3 têm iguais, ou seja, 0 significa que R2 é o complemento binário de R3, e 16 significa que R2 é igual a R3. A rotina pode destruir o valor dos diferentes registos, desde que retorne em R1 o valor pretendido.

Nbits: MOV R1, 16

XOR R2, R3 ; a 1 os bits diferentes

MOV Rh, 16

NbitsLoop: RORC R2

BR.WC NbitsProxbit

DEC R1 ; descontar um bit

NbitsProxbit: DEC R4

BR.NZ NbitsLoop

RET

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