



Curso:

Prova de Arquitectura de Computadores (21010)

Data: 12 de Fevereiro 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 (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	AB	00	01	11	10
CD	AB	00	0	1	1	0	00	0	1	1	0	
00	00	0	0	x	x	1	01	0	x	x	1	
01	01	0	x	x	1	1	11	x	0	0	x	
11	11	x	0	0	x	0	10	0	0	0	x	
10	10	0	0	0	x	0	0	0	0	0	x	

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.

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

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

2. Efectue as seguintes conversões entre bases numéricas:

2. a) [0.5] Represente o número 173_h em base 8:

$$\begin{array}{cccc}
 & 1 & 7 & 3 \\
 & \text{hexadecimal} & & \\
 \overbrace{\quad\quad\quad\quad}^{\text{binaário}} & 0001 & 0111 & 0011 \\
 & \text{binário} & & \\
 \overbrace{\quad\quad\quad\quad}^{\text{octal}} & 0 & 5 & 6 & 3 \\
 & & & & \text{octal}
 \end{array}$$

2. b) [0.5] Represente o número 1209₁₀ em base 2:

$$\begin{array}{c}
 1209 \mid 2 \\
 009 \quad | \quad 609 \quad | \quad 2 \\
 1 \quad | \quad 0 \quad | \quad 302 \quad | \quad 2 \\
 0 \quad | \quad 151 \quad | \quad 75 \quad | \quad 2 \\
 1 \quad | \quad 15 \quad | \quad 37 \quad | \quad 2 \\
 1 \quad | \quad 17 \quad | \quad 18 \quad | \quad 2 \\
 1 \quad | \quad 9 \quad | \quad 0 \quad | \quad 2 \\
 1 \quad | \quad 4 \quad | \quad 2 \quad | \quad 2 \\
 0 \quad | \quad 2 \quad | \quad 0 \quad | \quad 1
 \end{array}
 \qquad 100101110011$$

2. c) [0.5] Represente o número 100100010001₂ em base 10:

$$\begin{aligned}
 100100010001_2 &= 2^{11} + 2^8 + 2^4 + 2^0 = \\
 &= 1 + 16 + 256 + 2048 = 2321_{10} & \begin{array}{r}
 2048 \\
 256 \\
 16 \\
 \hline
 2321
 \end{array}
 \end{aligned}$$

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

3. a) [0.5] Represente o número -31 em binário com 8 bits, utilizando a técnica de complemento para 2.

$$-31 \rightarrow 11100001$$

$$\begin{array}{r} 31 \\ | \\ 15 \\ | \\ 7 \\ | \\ 3 \\ | \\ 1 \end{array} \quad \begin{array}{r} 2 \\ | \\ 2 \\ | \\ 2 \\ | \\ 2 \\ | \\ 1 \end{array}$$

$$\begin{aligned} 31_{10} &= 00011111 \\ &11100000 \quad (\text{complemento}) \\ &11100001 \quad (\text{incremento}) \end{aligned}$$

3. b) [0.5] Represente o número 01000110 em notação decimal, considerando que tem seis dígitos inteiros e dois fraccionários.

$$010001,10 = 2^4 + 2^0 + 2^{-1} = 16 + 1 + 0,5 = 17,5$$

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: 1011111010000000

$$\begin{aligned} S &= 1 & \text{Número} &= -1 \times 1,1010000000 \times 2^{15-15} = \\ E &= 01111 = 15 & & = -\left(2^0 + 2^{-1} + 2^{-3}\right) = -(1 + \frac{1}{2} + \frac{1}{8}) = \\ F &= 1010000000 & & = -\frac{8+4+1}{8} = -\frac{13}{8} = \underline{\underline{-1,625}} \end{aligned}$$

$$\begin{array}{r} 13 \\ | \\ 50 \\ | \\ 20 \\ | \\ 40 \\ | \\ 0 \end{array} \quad \begin{array}{r} 8 \\ | \\ 1,625 \end{array}$$

Grupo II

Considere a seguinte função lógica f :

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

1. [1.5] Simplifique algebricamente a função f .

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

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

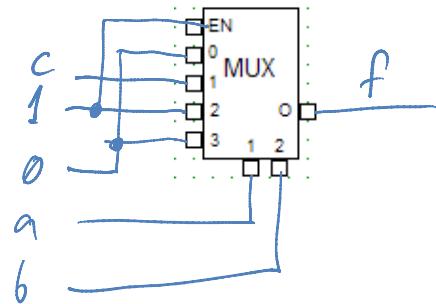
$$\begin{aligned} f &= \bar{a}b + a\bar{b}c = \overline{\bar{a}b + a\bar{b}c} = \overline{\bar{a}b} \cdot \overline{a\bar{b}c} = \\ &= \overline{\overline{\bar{a} \cdot a} \cdot b} \cdot \overline{\overline{a} \cdot \bar{b} \cdot b \cdot c} \end{aligned}$$

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

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

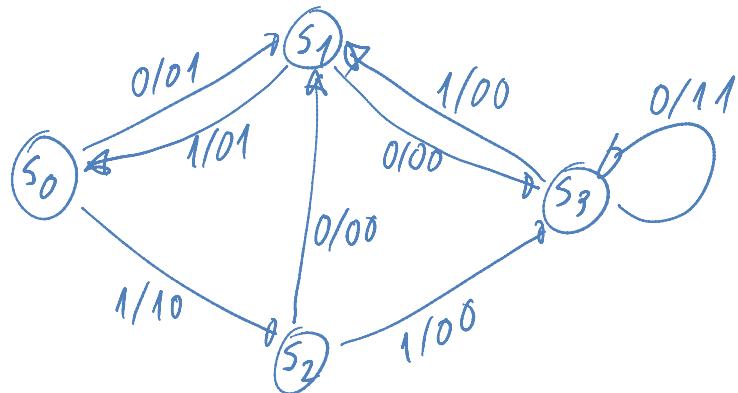
4. [1.5] Implemente a função recorrendo a um multiplexer de 2 variáveis de seleção.

$$f = \bar{a}b + ab\bar{c}$$



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 de estados correspondente ao diagrama de estados.

(entrada: A ; saídas: BC ; estados: e1; e2)

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

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

e_2e_1	00	01	11	10
A	0	0	1	1
0	0	1	1	0
1	1	0	0	1

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

e_2e_1	00	01	11	10
A	0	1	1	1
0	1	1	1	1
1	0	0	1	1

$$e_1 = \bar{A} + e_2$$

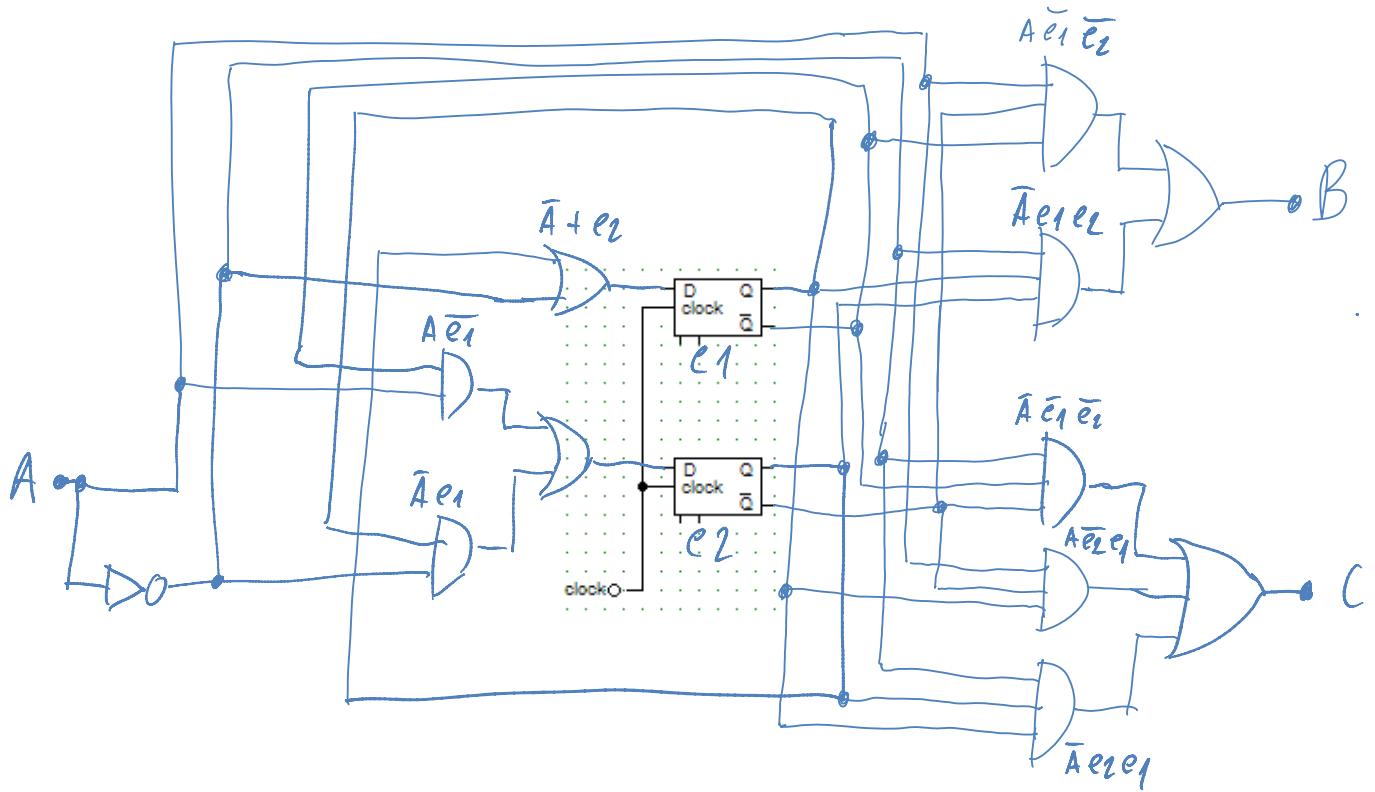
e_2e_1	00	01	11	10
A	0	0	1	1
0	0	0	1	0
1	1	0	0	0

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

e_2e_1	00	01	11	10
A	0	0	1	1
0	1	0	1	0
1	0	1	0	0

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

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 em R1 o topo da pilha, removendo o elemento da pilha	POP R1
Chamada condicional à subrotina "rotina", se a última operação aritmética/lógica teve resultado não negativo (bit mais significativo a 0)	CALL.NN rotina
Retorna de uma subrotina com 2 parâmetros no stack	RETN Z
Coloca em R1 a divisão inteira de R1 por R2 e em R2 coloca o resto dessa divisão	DIV R1, R2
Coloca em R1 a disjunção dos bits de R1 com os bits na posição de memória em "W"	OR R1, M[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:

Soma: MOV R1, R0
 MOV R2, M[SP+2]
 MOV R3, M[SP+3]
 CMP R3, R2
 BR.NN SomaOK
 XCH R3, R2
 SomaOK: ADD R1, R2
 CMP R3, R2
 BR.Z SomaFim
 INC R2
 BR SomaOK
 SomaFim: RET

```
int Soma(int a, int b)
{
    int resultado=0;
    if(a>b)
    {
        int aux=a;
        a=b;
        b=aux;
    }
    for(int i=a; i<=b; i++)
        resultado+=i;
    return resultado;
}
```

3. [2] Faça uma rotina em assembly do P3, assumindo que o Stack está inicializado e não tem problemas de limites. A rotina deve inverter os bits do registo R1, e manter o estado de todos os restantes registos.

InverterR1: PUSH R2
PUSH R3
MOV R2, R1
MOV R3, 16

InverterR1Loop: RORC R2
ROLC R1
DEC R3
BR.NZ InverterR1Loop
POP R3
POP R2
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