

Curso:

Prova de Arquitectura de Computadores (21010)

Data: 18 de Junho de 2010

Nome:

Nº de Estudante: B. I. nº

Turma: 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		00	01	11	10
AB						AB					
00		x	1	0	0	00		x	1	0	0
01		x	1	x	1	01		x	1	x	1
11		1	0	0	0	11		1	0	0	0
10		x	0	1	0	10		x	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.

2. Efectue as seguintes conversões entre bases numéricas:

2. a) [0.5] Represente o número 201h em base 8:

2. b) [0.5] Represente o número 725_{10} em base 2:

2. c) [0.5] Represente o número 010100110000_2 em base 10:

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.
3. b) [0.5] Represente o número 11001010 em notação decimal, considerando que tem cinco dígitos inteiros e três fraccionários.
3. c) [0.5] Considere a seguinte norma, baseada na recomendação IEEE-754, mas adaptada para 16 bits: $S=1, E=5, F=10$; $\text{Número} = (-1)^S * 1,F * 2^{(E-15)}$
Represente em notação decimal, o número: 0110110011000000

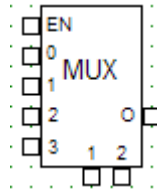
Grupo II

Considere a seguinte função lógica f :

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

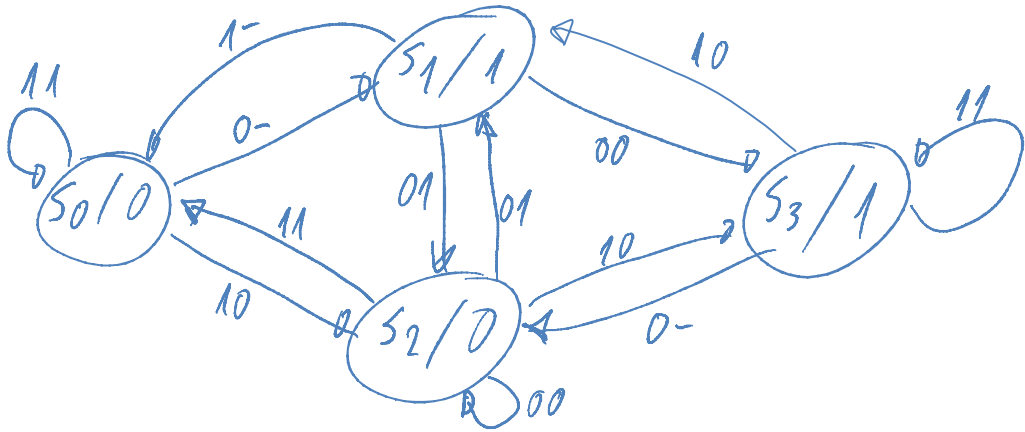
1. [1.5] Simplifique algebricamente a função f .
2. [1] Indique uma expressão lógica que implemente a função f utilizando apenas portas NAND
3. [1] Indique uma expressão lógica que implemente a função f utilizando apenas portas NOR

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



Grupo III

Considere o Diagrama de Estados seguinte:

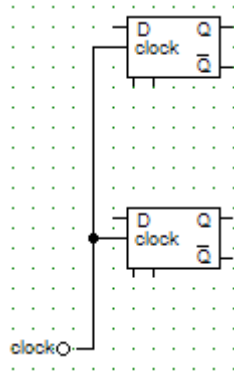


Pretende-se construir um circuito digital síncrono que implemente este diagrama, utilizando básicas tipo D.

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

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

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

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 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 registros 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 registros, desde que retorne em R1 o valor pretendido.

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

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