

Grupo I 1.a)

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

/BC+/AD+A/D

Grupo I 1.b)

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

(A+C+D)
(/A+C+/D)
(/B+/C)

Cr terios (positivos contam de 0 para cima, negativos s o penaliza es contando de cima para baixo):

- 0,3 - mapa corretamente constru do
- -0,3 - um ou mais la os n o m ximos
- -0,3 - utiliza mais la os que os necess rios
- -0,3 - um ou mais la os n o corretamente convertidos em express es l gicas
- -0,3 - n o existe rela  o identificada entre la os e a respetivo termo na express o l gica
- -0,3 - mapa incorretamente constru do (mas avaliado o resto considerando o mapa colocado)
- -0,3 - utiliza la os imposs veis (de 3, por exemplo)
- Na al nea b, caso o mapa esteja bem constru do,   contabilizado apenas um erro (ficando 0,2), com tudo certo fica 0,5.
- No mesmo mapa, la os com zeros e uns   valorizado em zero valores, excepto se o mapa estiver correto valendo nesse caso 0,3 na al nea A.

Grupo I.2a) 321h 0011.0010.0001b 001.100.100.001b 1441(8)

- 0,2 - convers o para bin rio
- 0,3 - convers o para octal

Grupo I.2b) 567(8) 101.110.111b 1+2+4+16+32+64+256 375

- 0,2 - convers o para bin rio
- 0,3 - convers o para decimal
- 0,5 - convers o com a f rmula do n mero (d gito multiplicado pela base elevado   posi  o)
- 0,1 - penaliza  o por pequeno erro de contas, no caso da convers o estar detalhada
- 0,2 - penaliza  o por erro de contas mais grave, no caso da convers o estar detalhada (exemplo, trocar os d gitos mais altos com os mais baixos)

Grupo I.3a) -27 27(10): 0001 1011b complemento: 1110 0100 b +1: 1110 0101 b

- 0,3 - convers o para bin rio
- 0,4 - complemento (8 bits)
- 0,3 - soma de uma unidade
- 0,1 - valor incorreto mas bin rio de 8 bits, sem qualquer outras contas

Grupo I.3b) 10001010 b 10001,010 b 16 + 1 + 0,25 17,25

- 0,5 convers o inteira
- 0,5 convers o decimal
- -0,1 erro de contas

Grupo II.1

/A.(B+C).(A+B)./C+/B

/A.(B+C).B./C+/B

/AB/C+/B

- -0,3 primeiro passo incorreto, existindo vários passos corretos
- -0,3 dois ou mais passos incorretos, existindo vários passos corretos
- 0,3 um passo correto
- -0,3 expressão final pode ser simplificada
- -0,3 passo correcto, mas sem regra existente (ou múltiplos passos realizados de uma só vez)
- 0 valores para tentativas sem um único passo correto

Grupo II.2 /AB/C+/B /(/(/(A.A).B./ (C.C)).B)

- 0,5 expressão lógica tem apenas portas NAND
- 0,5 circuito desenhado corretamente

Grupo II.3 //(/(A+/(B+B)+C)+/(B+B))

- 0,5 expressão lógica tem apenas portas NOR
- 0,5 circuito desenhado corretamente

Grupo II.4 00 - 1; 01 - /C 10 - 1; 11 - 0 --- pergunta avaliada de acordo com a resposta dada na alínea 1

- 0,5 - por uma ou mais das entradas corretas, e outras entradas incorretas
- 0 valores por todas as entradas incorretas

Grupo III.1

Entrada	Estado atual	Estado seguinte	Saída
00	00	00	0
01	00	00	0
10	00	01	1
11	00	10	0
00	01	01	1
01	01	01	1
10	01	10	1
11	01	10	1
00	10	10	0
01	10	11	1
10	10	10	0
11	10	11	1
00	11	11	1
01	11	11	1
10	11	00	0
11	11	00	0

Quem ordenou por entrada primeiro e depois por estado atual:

Entrada	Estado atual	Estado seguinte	Saída
00	00	00	0
00	01	01	1
00	10	10	0
00	11	11	1
01	00	00	0
01	01	01	1
01	10	11	1
01	11	11	1
10	00	01	1
10	01	10	1
10	10	10	0
10	11	00	0
11	00	10	0
11	01	10	1
11	10	11	1
11	11	00	0

- -0,5 um ou mais erros no estado seguinte
- -0,5 um ou mais erros numa das saídas
- -0,5 um ou mais erros na codificação dos estados

- -1 estados não codificados em variáveis binárias
- -0,5 um ou mais erros na entrada de dados ou no estado atual
- +0,5 tem uma tabela com as colunas corretas, e número de linhas correto

III.2 (exame 2 valores - simplificar variáveis de saída e de estado), (p-fólio 1 valor - simplificar apenas as variáveis de estado)

- 0,5 valores por cada mapa correto (3 variáveis no total do exame, o primeiro mapa vale 1 em vez de 0,5 valores, e 2 no p-fólio, cada a valer 0,5)
- 0,5 no caso de erro na conversão para os mapas de Karnaugh, mas correta simplificação (p-fólio)

Apresentam-se duas versões de mapas, para quem tenha trocado AB por CD (entradas nas linhas e variáveis de estado nas colunas, ou vice-versa)

E1									
Entrada AB \ Estado CD	00	01	11	10	CD\AD	00	01	11	10
00	0	0	1	1	00	0	0	1	0
01	0	0	1	1	01	0	0	1	1
11	1	1	0	1	11	1	1	0	0
10	0	1	0	1	10	1	1	1	1

/AC+C/D+AB/C+A/CD

E0									
Entrada AB \ Estado CD	00	01	11	10	CD\AD	00	01	11	10
00	0	1	1	0	00	0	0	0	1
01	0	1	1	1	01	1	1	0	0
11	0	0	0	1	11	1	1	0	0
10	1	0	0	0	10	0	1	1	0

/AD+BC/D+A/B/C/D

Saída									
Entrada AB \ Estado CD	00	01	11	10	CD\AD	00	01	11	10
00	0	1	1	0	00	0	0	0	1
01	0	1	1	1	01	1	1	1	1
11	0	1	0	1	11	1	1	0	0
10	1	1	0	0	10	0	1	1	0

/AD+/CD+BC/D+A/B/C

Grupo III.3

1 Esquema corretamente desenhado, com base na alínea anterior.

Grupo IV. 1a) SUB M[R1+W], R2

Grupo IV. 1b) MOV M[W], R2

Grupo IV. 1c) PUSH R1

Grupo IV. 1d) SHR R1, 3

- 0,5 por cada alínea correta
- 0,25 no caso de pequeno erro, inversão da ordem dos argumentos, e troca de push com pop

Grupo IV. 1/2

- Avaliado apenas a parte da função solicitada
- 0,5 por uma tentativa realizada com algum código com algum sentido, relacionado com o solicitado, mesmo contendo erros sintáticos
- -0,5 por cada erro de instrução, no caso do código ter uma estrutura correta na globalidade: multiplicação e divisão sem ter em conta que ambos os argumentos ficam com o resultado da operação; teste do ciclo; utilização de registos não inicializados; multiplicação com um dos argumentos uma constante, etc. Máximo de 3 erros, ficando com 1,5 valores, dado que a estrutura do código está correta na sua globalidade.

	MOV R2, 1	; res = 1
	MOV R3, 1	; num = 1
	MOV R4, 1	; den = 1
	MOV R5, 1	; i=1
For:	PUSH R5	; utilizar R5 numa multiplicação e recuperar o valor de seguida

MUL R5, R3	; num = num * i, num fica com a parte baixa, assume-se que os números não irão necessitar da parte alta
POP R5	; recuperar o valor de R5
MOV R6, R5	; R6 irá ficar com 2^{*i+1} , para já fica com i
SHL R6, 1	; R6 fica com 2^{*i}
INC R6	; R6 fica com 2^{*i+1}
MUL R6, R4	; den = den * (2^{*i+1}), R4 fica com a parte baixa, e assume-se que não é necessária a parte alta
MOV R6, R3	; preparar num para dividir por dem
PUSH R4	; utilizar R4 (dem) e depois manter o valor
DIV R6, R4	; R6 fica o resultado da divisão, em R4 será o resto para ignorar
POP R4	; recuperar o valor de R4
ADD R2, R6	; res = res + num/dem, assumindo-se que os 16 bits não serão atingidos
INC R5	; incrementar a variável iteradora
CMP R5, R1	; teste com o valor máximo, se R5 é menor ou igual, continua, caso contrário pára
BR.NN For	; continuar o ciclo
MOV R3, R2	; colocar na variável de saída res
SHL R2, 1	; multiplicar por 2
RET	; retornar