

Aluno:

1. (1,5) Simplifique, usando Álgebra de Boole: $S = [(AC)' + B + D]' + C(ACD)'$ 0,2
 2. (1,0) Ache a expressão na forma simbólica que define a função booleana dada pela tabela-verdade abaixo. (b) Simplifique a expressão e obtenha uma outra equivalente e mais curta. (c) Desenhe um circuito lógico que implemente a função. 0,4

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

3. (1,5) Um computador possui um conjunto de 128 instruções de um operando. Supondo que sua memória tenha capacidade para armazenar 512 palavras de 16 bits e que cada instrução tem o tamanho de uma palavra, responda as perguntas a seguir: a) Qual o tamanho em bits do REM, RDM, RI e PC? b) Qual a capacidade da memória, em bytes? c) Se alterarmos o tamanho das instruções para 18 bits, mantendo inalterado o tamanho do REM, quantas novas instruções poderiam ser criadas? 0,5
 4. (1,0) Um computador tem um REM de 16 bits e de mesmo tamanho da palavra do processador. Ele possui instruções de 1 operando, todas de tamanho de uma palavra de memória e de mesmo tamanho da palavra do processador. Ele foi adquirido com uma placa de 8K de memória. a) Qual o tamanho, em bits, do RDM e PC? b) É possível aumentar a capacidade de armazenamento desta memória? c) Quanto? c) Qual a quantidade máxima de instruções de máquina que poderia existir neste computador? 0,2
 5. (1,0) Considere uma célula de uma MP cujo endereço é, em hexadecimal, 812C e que tem armazenado em seu conteúdo um valor igual a, em hexadecimal, A5F. Sabe-se que nesse sistema as células têm o mesmo tamanho das palavras e que em cada acesso é lido o valor de uma célula. Pergunta-se: qual deve ser o tamanho do REM e do RDM nesse sistema? 0,5
 6. (2,0) Suponha que as posições de memória de endereços A4 a B1 do computador descrito no "Apêndice C" do livro-texto (a linguagem de máquina é parcialmente reproduzida abaixo) contenham os valores hexadecimais de bits listados na Tabela 1. Relembrando, no computador do "Apêndice C", cada instrução de máquina possui dois bytes. Os primeiros 4 bits contêm o código de operação e os últimos 12 bits o campo de operando. Cada registrador tem 8 bits de tamanho e cada célula de memória também é de 8 bits. Na descrição da linguagem de máquina, as letras R, S e T representam registradores e as letras X e Y representam variáveis (que não sejam os registradores). Todas essas letras são usadas no lugar de dígitos hexadecimais. Se o Program Counter (PC) contém o valor A4, pergunta-se: (a) O que haverá no registrador 0 na primeira vez em que a instrução de endereço AA for executada? (b) O que haverá no registrador 0 na segunda vez em que a instrução de endereço AA for executada? (c) Quantas vezes a instrução de endereço AA será executada antes do programa terminar? Explique todas as suas respostas.

Tabela 1

Endereço	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1
Conteúdo	20	00	21	03	22	01	B1	B0	50	02	B0	AA	C0	00

7. (2,0) Suponha que as posições de memória de endereços D8 a F5 do computador descrito na questão anterior contenham os valores hexadecimais de bits listados na Tabela 2. Pergunta-se: (a) se o Program Counter (PC) contém o valor D8, descreva o que se executou a cada instrução e apresente o valor final dos registradores manipulados pelo programa. Explique todas as suas respostas.

Tabela 2

Endereço	D8	D9	DA	DB	DC	DD	DE	DF	F0	F1	F2	F3	F4	F5
Conteúdo	21	A3	20	C0	30	F4	22	00	32	F5	93	01	FF	FF

***** "Apêndice C" *****

- Y LOAD (carrega) o registrador R com o padrão de bits encontrado na posição de memória de endereço XY. Ex: 14A3 carrega o conteúdo da posição de memória de endereço A3 no registrador 4.
 Y LOAD (Carrega) o registrador R com o valor XY. Exemplo: 20A3 carrega o valor A3 no registrador 0.
 Y STORE (armazena) o padrão de bits encontrado no registrador R na posição de memória de endereço XY. Ex: 35B1 armazena o conteúdo do registrador 5 na posição de memória de endereço B1.
 S MOVE (copia) o padrão de bits encontrado no registrador R para o registrador S. Ex: 40A4 copia o conteúdo do registrador 4 para o registrador 4.
 T ADD (soma) os padrões de bits dos registradores S e T, em complemento a dois, e coloca o resultado no registrador R. Ex: 5 soma os valores binários dos registradores 2 e 6 e coloca no registrador 7 esse resultado.
 T OR (ou) executa a operação lógica OR sobre os padrões de bits dos registradores S e T e coloca o resultado no registrador R. 7CB4 coloca no registrador C o resultado da operação OR com os conteúdos dos registradores B e 4.
 T AND (e) executa a operação lógica AND sobre os padrões de bits dos registradores S e T e coloca o resultado no registrador R. 8045 coloca no registrador 0 o resultado da operação AND com os conteúdos dos registradores 4 e 5.
 T EXCLUSIVE OR (Ou-Exclusivo) executa a operação de Ou-Exclusivo sobre os padrões de bits dos registradores S e T e coloca o resultado no registrador R. Ex: 9523 coloca no registrador 5 o resultado da operação de Ou-Exclusivo entre os conteúdos dos registradores 2 e 3.
 X ROTATE (gira) o padrão de bits do registrador R, de X bits para a direita e, em todas as vezes, colocando o bit que está na extremidade inferior na extremidade superior. Ex: A403 gira de 3 bits para a direita o conteúdo do registrador 4, de forma circular.
 Y JUMP (salta) para a instrução localizada na posição de memória de endereço XY se o padrão de bits do registrador R coincide com o padrão de bits do registrador 0. Caso contrário, prossegue na sequência normal de execução.
 0 HALT (para) finaliza a execução do programa corrente.

Guilherme Tatagiba Brandão

Parabéns!!

100!!

1.0

$$\begin{aligned}
 1. & \quad (\overline{A}C) + B + D + C(\overline{A}C\overline{D}) \\
 & = [(\overline{A} + \overline{C}) + B + D] + C(\overline{A} + \overline{C} + \overline{D}) \quad \text{TEOREMA DE MORGAN} \\
 & = [A.C.B.D] + AC + C\overline{D} \quad \text{TEO. DE MORGAN/DISTRIBUTIVIDADE} \\
 & = C\overline{D}(AB + 1) + \overline{A}C \quad \text{ADIÇÃO / DISTRIBUTIVIDADE} \\
 S & = C\overline{D} + \overline{A}C \quad \text{CIRC. SIMPLIFICADO.}
 \end{aligned}$$

1.0

4. REM $\Rightarrow$ 16 bits, BD $\Rightarrow$ 20 bits
8K memória total.

a) RDM possui o mesmo tamanho do barramento de dados, ou seja, 80 bits.

1) PC possui o mesmo tamanho do REM, ou seja, 16 bits.

$$\begin{aligned}
 b) & \quad \text{Pode ser aumentada para } 20 \cdot 2^{16} = 5 \cdot 2^2 \cdot 2^{16} = 5 \cdot 2^{18} = \\
 & = 5 \cdot 2^{10} \cdot 2^8 = 160 \text{ Kbytes}
 \end{aligned}$$

$\uparrow$ N° DE BITS DE CADA CÉLULA $\uparrow$ N° DE ENDEREÇOS

c)

4 bits	16 bits
CÓDIGO DE OPERAÇÃO	OPERANDO

 $\rightarrow$ POIS O REM POSSUI 16 bits, HÁ DE SE ENDEREÇAR QUALQUER POSIÇÃO DE MEMÓRIA.

N° de instruções possíveis $\Rightarrow 2^4 = 16$ instruções.

1.5

3.

a)

REM $\Rightarrow$ Armazena 512 palavras $\Rightarrow 2^9$, ou seja, 9 bits de REM.

RDM $\Rightarrow$ 16 bits $\Rightarrow$ TAMANHO DE CADA PALAVRA

RI $\Rightarrow$ 16 bits $\Rightarrow$ TAMANHO DE CADA PALAVRA

PC $\Rightarrow$ 9 bits $\Rightarrow$ mesmo valor do REM.

$$b) \quad 16 \cdot 512 = 2^4 \cdot 2^9 = 2^{13} = 2^{10} \cdot 2^3 = 8 \text{ Kbytes} = 1024 \text{ bytes.}$$

c) $\log_2 128 = 7$. Ou seja, o código de operação possui 7 bits.

Se aumentar para 18 bits o tamanho da instrução, sem alterar o REM, o operando permanece do mesmo tamanho e o código de operação passa a ter 9 bits $(7 + (18 - 16))$.

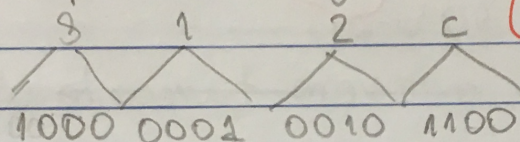
Logo, o nº de instruções possíveis será:

$2^9 = 512$ Antes havia $2^7 = 128$, ou seja, poderão ser criadas $512 - 128$ novas instruções.

$$\begin{array}{r} 512 \\ - 128 \\ \hline 384 \text{ novas instruções.} \end{array}$$

1.º

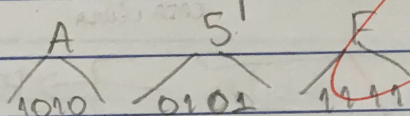
5. Como o endereço é 812C, o número de bits necessário para armazenar esse valor é:



$$4 \times 4 = 16 \text{ bits.}$$

Logo, o REM terá 16 bits.

O número de bits para armazenar ASF é:



$$4 \times 3 = 12 \text{ bits.}$$

Logo, o RDM terá 12 bits.

2.º

6. 2000 $R_0 = 00$

2103 $R_1 = 03$

* ESQUEMA ILUSTRATIVO DO 1º CICLO

2201 $R_2 = 01$

BB0

5002 $R_0 + R_2 = 00 + 01 = 01 \Rightarrow R_0 = 01$

B0AA Volta para AA.

C000

- a) No registrador 0 terá o valor 00, pois [2000] carrega 00 no registrador 0.
- b) Haverá o valor 01, pois as instruções estão somando uma unidade ao registrador zero a cada ciclo, até que o valor que esteja no registrador 0 seja igual ao que está no registrador 1.

c) Quatro vezes. Assim como explicamos no item anterior, inicialmente o R_0 (registrador zero) terá zero, na primeira passagem por AA. Na segunda, terá o valor 1, na terceira, 2, e na quarta 3. Como o R_0 tem o valor 3, o R_1 também, a instrução pulará para B0, onde há a instrução de parada C000.

2.º 7. $21A3 \Rightarrow R_1 = A3$ (neg. 1 recebe o valor A3).

$2000 \Rightarrow R_0 = 0$ (arg. 0 recebe o valor 0).

30F4 $\Rightarrow$ Posição de memória F4 recebe o valor CO de Ro.

$2200 \Rightarrow R_2 = 00$ (reg. 2 recebe o valor 00).

- 32 PS $\Rightarrow$ Posição de memória PS recebe o valor 00 do R2

$$9301 \Rightarrow R_0 \oplus R_1 = R_3$$

~~FF~~ FF

$$\overset{11}{CO} \overset{11}{+} \overset{11}{A3} = \overset{C}{1100} \overset{O}{0000}$$

④ A 3
10100011

$$R_3 = \begin{array}{cccccc} 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \hline & 6 & & & & 3 & \end{array}$$

COOD $\Rightarrow$ Para a expressão.

$$R_0 = C0; R_1 = A3; R_2 = 00; R_3 = 63$$

2.

a) $\bar{x}\bar{y}z + x\bar{y}z + xy\bar{z} = S$ ✓

$$b) = z(\bar{x}\bar{y} + x\bar{y} + xy) \quad \text{DISTRIBUTIVIDADE}$$
$$= z(\bar{y}(\bar{x} + x) + xy) \quad \text{DISTRIBUTIVIDADE / ADIÇÃO}$$
$$= Z(\bar{Y}(1) + XY) \quad \text{Seja } \bar{Y} = y \quad \text{SUBSTITUIÇÃO CONVENIENTE}$$
$$= Z(y + \bar{y}X) \quad \text{IDENTIDADE AUXILIAR (A + \bar{A}B = A + B)}$$
$$= z(y + X) \quad y = \bar{y}$$
$$= z(\bar{y} + x)$$

DISTRIBUTIVIDADE

$$= 7\bar{y} + 7x.$$

c)

