PMR3333 – Eletrônica para Engenharia Mecânica
Prof. Celso M. Furukawa – Lista de exercícios 15 (P3)

1. Dadas as funções lógicas *f* e *g* abaixo, manipule-as algebricamente e determine suas expressões equivalentes na forma soma de produtos simplificada. Indique os passos utilizados. Não utilize mapas de Karnaugh ou tabelas da verdade:

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

b) $g\left(a,b,c,d\right) = \left(a.c.\bar{d} + \bar{b} \oplus d + \bar{d}\right).\left(\bar{a} + \bar{b} + a.b\right)$

2. Dada a função lógica *h*(*a*, *b*, *c*, *d*) definida pela tabela mostrada na figura 2, determine a expressão lógica dessa função na forma soma de produtos simplificada usando mapa de Karnaugh.

3. A figura 3a mostra o símbolo de um decodificador 2:4 com entrada de *enable E* ativa em 1, onde *A*[1:0] são os bits de endereço e *Y*[3:0] são as saídas. O símbolo da figura 3b é representa um multiplexador 4:1, onde *S*[1:0] são os bits de seleção, *I*[3:0] são as entradas de dados, *E* é a entrada de *enable* ativa em 1 e *Y* é a saída.

a) Faça o diagrama lógico do decodificador e do multiplexador usando portas lógicas AND, OR e NOT genéricas.

b) Mostre como construir o multiplexador usando o decodificar e portas lógicas AND, OR e NOT genéricas.

a b c d	*h*	*a b c d*	*h*	*a b c d*	*h*	*a b c d*	*h*
0 0 0 0	1	0 1 0 0	1	1 0 0 0	1	1 1 0 0	0
0 0 0 1	1	0 1 0 1	0	1 0 0 1	1	1 1 0 1	1
0 0 1 0	1	0 1 1 0	1	1 0 1 0	1	1 1 1 0	0
0 0 1 1	1	0 1 1 1	0	1 0 1 1	1	1 1 1 1	1

Figura 2

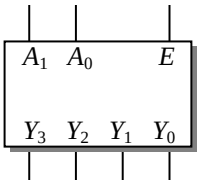

Figura 3a

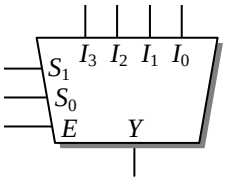

Figura 3b

4. Determine a forma de onda das saídas Q_0 e Q_1 dos flip-flops da figura 4a, considerando os sinais de entrada mostrados na figura 4b. As entradas CLR zeram os flip-flops de forma **assíncrona**.

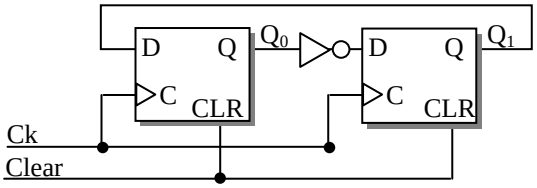

Figura 4a

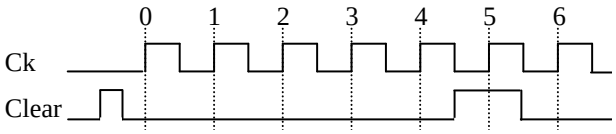

Figura 4b

5. A tabela da figura 5 mostra algumas instruções de uma CPU de 8 bits, onde: R0 e R1 são registradores; *kk* representa uma constante de 8 bits; PC é *Program Counter*; e Z é um *flag* tal que Z=1 indica que a última operação executada pela CPU resultou em zero. Operações aritméticas são efetuadas na notação Complemento de 2 (C2). Mem[*kk*] indica a posição de memória de endereço *kk*. A coluna *Flags* mostra quais *flags* são afetados pela instrução. A coluna *Bytes* mostra o número de bytes usados para codificar a instrução.

A figura 5 mostra também um trecho de programa em *assembly*, onde todas as constantes são dadas em hexadecimal.

a) Suponha que o programa seja transcrito em código de máquina para ser carregado na memória a partir do endereço 0. Liste o conteúdo de cada posição de memória ocupada pelo programa, indicando endereços e conteúdos em hexadecimal.

b) Determine o valor final dos registradores R0 e R1 e da posição de memória FFh após a execução do programa. Justifique.

Mneum.	Opcode	Descrição	Flags	Bytes
ADD R0, R1	41	R0 ← R0 + R1 (em C2)	Z	1
CLR R0	30	R0 ← 0	-	1
DEC R1	F1	R1 ← R1 - 1 (em C2)	Z	1
JNZ *kk*	04	se Z = 0, PC ← *kk*	-	2
LDI R1, *kk*	E1	R1 ← *kk*	-	2
STS *kk*, R0	A0	Mem[*kk*] ← R0	-	2

```
        CLR R0
        LDI R1, 3
LABEL:  ADD R0, R1
        DEC R1
        JNZ LABEL
        STS FF, R0
```

Figura 5

Algumas respostas
1.a) a'.b + a'.c'.d. 1.b) b + d'.
2) b' + a.d + a'.d'
3.a) Decoder: Y0 = E.A1'.A0', Y1 = E.A1'.A0, Y2 = E.A1.A0', Y3 = E.A1.A0. Mux: Y = E.(I0.S1'.S0' + I1.S1'.S0 + I2.S1.S0' + I3.S1.S0). 3.b) Entradas do decoder: E ← E externo, A1 ← S1, A0 ← S0. Usar as saídas do decoder para fazer a saída do mux Y = Y0.I0 + Y1.I1 + Y2.I2 + Y3.I3
4)Q0 indefinido até o primeiro Clear=1; Q0 ← 0 após Clear=1 até Ck 1; Q0 ← 1 entre Ck 1 e Ck 3.
Q1 indefinido até o primeiro Clear =1; Q1 ← 0 após Clear=1 até Ck 0, entre C2 e Ck 4, e assincronamente após o segundo Clear=1 até Ck 6; Q1 ← 1 entre Ck 0 e Ck 2, após Ck 4 até o segundo Clear=1, e após Ck 6.
5.a) (Endereço: byte) 0: 30; 1: E1; 2: 03; 3: 41; 4: F1; 5: 04; 6: 03; 7: A0; 8: FF. 5.b) R0=6; R1 = 0; Mem[FF]=6. Soma em R0 os números decrescentes de 3 a 1.