PMR-3303 Eletrônica Digital para Mecatrônica
Prof. Celso Massatoshi Furukawa - Lista de Exercícios 12 (P3)

Q1. A figura 1 mostra um flip-flop tipo D. A figura 2 mostra um contador de 3 bits com entradas para carga em paralelo e sua tabela de funcionamento. Construa o registrador da figura 2 usando flip-flops da figura 1 e circuitos lógicos adicionais (portas, decodificadores, multiplexadores, etc).

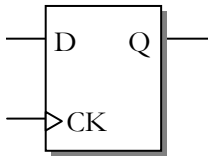

Figura 1

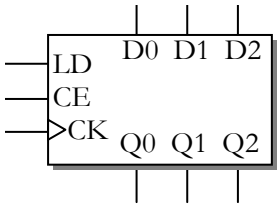

Figura 2

CK	CE	LD	Q[2:0]	Função
↑	0	X	Q[2:0]	Mantém
↑	1	1	D[2:0]	Carga paralela
↑	1	0	Q[2:0] + 1	Incremento

Q2. Descreva em Verilog o contador da figura 2 de forma procedural de alto nível (com bloco *always*, sem instanciar flip-flops) em um módulo de nome **CB3LD**

Q3. Seja uma máquina de estados finitos MEALY, de entrada Y e saída Z. Tem-se Z = 1 sempre que o circuito detectar na entrada Y sequências 10110 em bordas consecutivas de clock; caso contrário, Z deve permanecer em 0. Ao entrar em seu estado inicial, o circuito não deve assumir qualquer sequência prévia. Duas sequências podem ser superpostas, como mostra o exemplo abaixo.

Y	0	1	0	1	1	0	1	1	0	1	0
Z	0	0	0	0	0	**1**	0	0	**1**	0	0

a) Faça o diagrama de estados da máquina de estados MEALY deste circuito.
b) Transcreva o diagrama para uma tabela de transição e de saída (NÃO É NECESSÁRIO codificar os estados)

Q4. A tabela abaixo descreve as transições de uma máquina sequencial de duas entradas Y_1Y_0, onde Q_1 e Q_0 são os bits de saída de flip-flops tipo D que codificam os estados, e Z é a saída.

$Q_1Q_0^k$	$Q_1Q_0^{k+1}$ / Z			
	Y_1Y_0=00	Y_1Y_0=01	Y_1Y_0=11	Y_1Y_0=10
00	11 / 1	10 / 1	00 / 0	10 / 0
01	11 / 0	01 / 0	10 / 0	11 / 0
11	01 / 0	01 / 1	10 / 0	01 / 0
10	XX / X	XX / X	00 / 0	XX / X

a) Determine a função minimizada da saída Z por meio de um mapa de Veitch-Karnaugh.
b) Determine a expressão minimizada da entrada D **SOMENTE** do flip-flop de Q_1

Q5. A tabela 1 mostra os mneumônicos e opcodes (em hexadecimal) de algumas instruções de uma CPU hipotética de 8 bits, onde RA e RB são dois registradores de uso geral; *kk* representa uma constante de 8 bits; PC é *Program Counter*; e Z é um *flag* tal que Z=1 indica que o resultado da última operação executada foi zero. Mem[*kk*] indica a posição de memória de endereço *kk*. A coluna *Flags* mostra quais *flags* são afetados pela instrução. A coluna *Bytes* indica o número de bytes necessários para codificar a instrução.

A figura 3 mostra um trecho de programa em *assembly*, onde .ORG é uma diretiva que define o endereço da próxima instrução; e `LOOP` é um rótulo associado ao **endereço** da instrução que o segue. Todas as constantes são dadas em hexadecimal. Pede-se:

a) Transcreva o programa da figura 3 para o código de máquina a partir da posição 0 de memória.
b) Explique textualmente o que esse programa faz (NÃO o descreva instrução por instrução), e determine o valor dos registradores RA e RB ao final da execução do programa.

Tabela 1

Mneum.	Opcode	Descrição	Flags	Bytes
ADD RA, RB	41	RA ← RA + RB (em C2)	Z	1
CLR RA	0B	RA ← 0	-	1
DEC RB	F0	RB ← RB - 1 (em C2)	Z	1
JNZ *kk*	0D	se Z = 0, PC ← *kk*	-	2
LDI RB, *kk*	E0	RB ← *kk*	-	2
STS RA, *kk*	11	Mem[*kk*] ← RA	-	2

```
.ORG 0
       CLR RA
       LDI RB, 4
LOOP:  ADD RA, RB
       DEC RB
       JNZ LOOP
       STS RA, FF
```
Figura 3

Algumas respostas

Q1) Di = CE'.Qi + CE.LD.Di + CE.LD'.[Qi⊕(Qi-1..Q0)], i = 1, 2. Para i =0, o último termo é CE.LD'.Qi'.

Q3) 5 estados: $Q^k \rightarrow [Q^{k+1}/Z$ para Y=0; Q^{k+1}/Z para Y=1]: A → [A/0; B/0], B → [C/0; B/0], C → [A/0; D/0], D → [C/0; E/0], E → [C/1; B/0].

Q4a) Z = Y1'.Q0' + Y1'.Y0.Q1. b) Y1'.Q0' + Y0'.Q1' + Y1.Y0.Q0

Q5a) Endereço: byte(s). 0: 0B, 1: E0 04, 3: 41, 4: F0, 5: 0D 03, 7: 11 FF. b) Soma 4 a 1; ao final, RA = 10 e RB = 0.