

An aerial photograph of a beach. The top half of the image shows the deep blue ocean with white-capped waves breaking onto the shore. The bottom half shows the golden-brown sand of the beach. The text is overlaid on the ocean and sand.

Curso de Engenharia Elétrica
Laboratório de Circuitos Digitais II

Roteiros

Laboratório de Digitais II

Prof. Fernando Passold

Sumário

6 Divisor de frequência	5
7 Aplicação de <i>Ripple Counter</i>	7
8 Máquina síncrona “Exótica”	11
9 Projeto de Contador Síncrono	15
10 Aplicação de Contador Síncrono 74190	19

9 | Projeto de Contador Síncrono

Exemplo – Semestre 2019.1

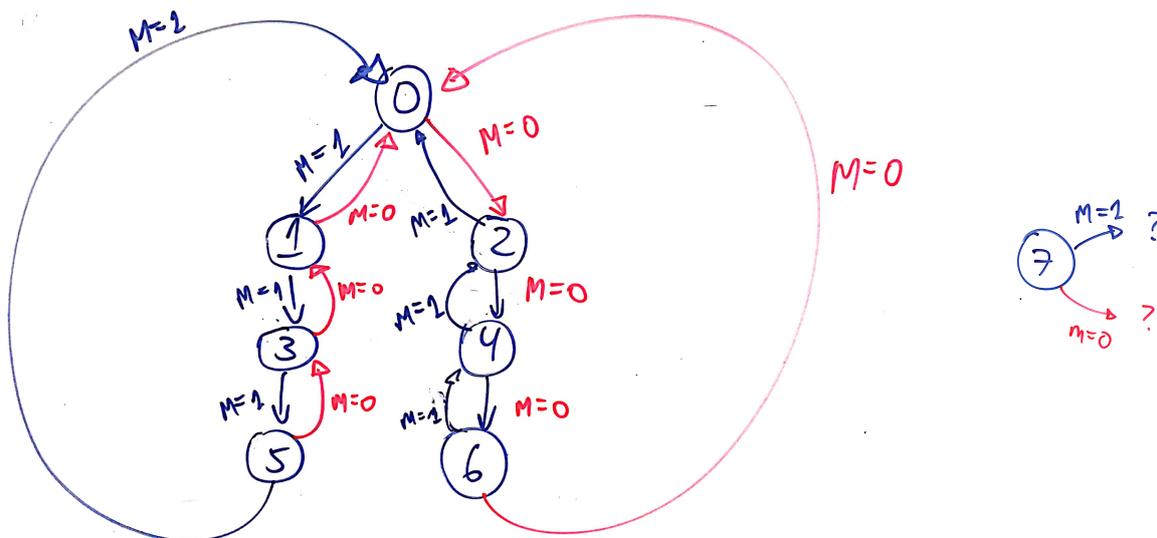


Figura 9.1: Sequencia de estados referente ao contador à ser projetado.

9.1 Introdução

O objetivo deste projeto é levantar as equações, circuito e realizar em laboratório toda a parte de um circuito necessário para gerar um contador síncrono capaz de realizar a sequência de contagem (ou estados) mostrada na figura 9.1.

Notar que para este contador existe uma entrada extra de controle, a entrada M ou "Mode", cujo nível lógico define a forma como este circuito deve executar a sua sequência de contagem.

Nota-se que o maior valor decimal alcançado por este circuito é 7, o que significa que serão necessários 3 Flips-Flops para sintetizar este contador ($6_{(10)} = 110_{(2)}$). Também percebemos que a combinação de estados $7_{(10)} = 111_{(2)}$ não está prevista. Mas de todas as formas, fica a questão de como o circuito deve evoluir se por acaso for inicializado nesta condição – canto direito da fig. 9.1.

Continuando o projeto, já percebemos que serão necessários 3 flip-flops. Pode-

se adotar FF's do tipo D ou do tipo J:

- Opções
- FF's-JK: 3 × FF's:
3 × 2 entradas (J, K):
6 × Mapas K
 - FF's-D: 3 × FF's:
3 × 1 entrada (J, K):
3 × Mapas K

O uso de FF's do tipo D vai levar ao desenvolvimento de 3 Mapas de Karnaugh necessários para definir o circuito de controle (equação) de entrada de cada FF. Se forem adotados FF's do tipo JK, serão necessários $3 \times 2 = 6$ mapas de Karnaugh para levantar as equações (circuitos) necessários para controlar as entradas destes FF's.

Este documento está optando pela adoção de FF's do tipo D. Seguimos então com o levantamento da tabela de transição de estados do circuito completou ou deste contador. Em seguida, devemos levantar os mapas de Karnaugh e equações (circuitos) que definem as entradas D de cada FF. Em seguida podemos proceder ao desenho do diagrama elétrico final deste

circuito e ao final, analisar o que acontece se os FF's deste circuito forem inicializados na combinação $7_{(10)} = 111_{(2)}$.

9.2 Tabela de Transição de Estados

A figura ?? mostra a tabela de transição de estados levantada para este contador.

9.3 Mapas de Karnaugh e equações iniciais

De posse das condições levantadas na tabela 9.1 podemos levantar os mapas de Karnaugh que definem as equações iniciais para as entradas D dos FF's adotados. A figura 9.2 mostra os mapas resultantes.

Ref	M	q2 q1 q0	Q2 Q1 Q0	D2 D1 D0	Transição
0	0	000	010	010	(M=0) 0 → 2
1	0	001	000	000	(M=0) 1 → 0
2	0	010	100	100	(M=0) 2 → 4
3	0	011	001	001	(M=0) 3 → 1
4	0	100	110	110	(M=0) 4 → 6
5	0	101	011	011	(M=0) 5 → 3
6	0	110	000	000	(M=0) 6 → 0
7	0	111	XXX	XXX	(M=0) 7 → X
8	1	000	001	001	(M=1) 0 → 1
9	1	001	011	011	(M=1) 1 → 3
10	1	010	000	000	(M=1) 2 → 0
11	1	011	101	101	(M=1) 3 → 5
12	1	100	010	010	(M=1) 4 → 2
13	1	101	000	000	(M=1) 5 → 0
14	1	110	100	100	(M=1) 6 → 4
15	1	111	XXX	XXX	(M=1) 7 → X

Tabela 9.1: Tabela de transição de estados referente a este contador.

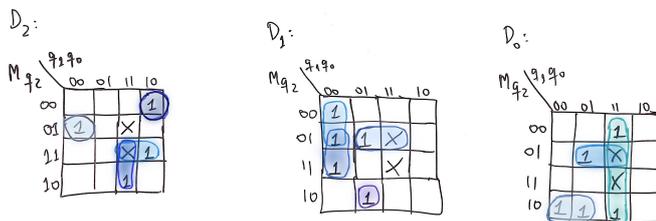


Figura 9.2: Mapas de Karnaugh.

$$\begin{aligned}
 D_2 &= \overline{M} \cdot q_2 \cdot \overline{q_1} \cdot \overline{q_0} + \overline{M} \cdot \overline{q_2} \cdot q_2 \cdot \overline{q_0} + \\
 &\quad + M \cdot q_2 \cdot q_1 + M \cdot q_1 \cdot q_0 \\
 &= \overline{M} \cdot \overline{q_0} (q_2 \cdot \overline{q_1} + \overline{q_2} \cdot q_1) + M \cdot q_1 (q_2 + q_0) \\
 &= \overline{M} \cdot \overline{q_0} (q_2 \odot q_1) + M \cdot q_1 \cdot (q_2 + q_0)
 \end{aligned}$$

$$\begin{aligned}
 D_1 &= \overline{M} \cdot \overline{q_1} \cdot \overline{q_0} + q_2 \cdot \overline{q_1} \cdot \overline{q_0} + \overline{M} \cdot q_2 \cdot q_0 + \\
 &\quad + M \cdot \overline{q_2} \cdot \overline{q_1} \cdot q_0 \\
 &= \overline{q_1} \cdot \overline{q_0} (\overline{M} + q_2) + q_0 (\overline{M} \cdot q_2 + M \cdot \overline{q_2} \cdot \overline{q_1})
 \end{aligned}$$

$$D_0 = q_1 \cdot q_0 + \overline{M} \cdot q_2 \cdot q_0 + M \cdot \overline{q_2} \cdot \overline{q_1}$$

Percebe-se pelas equações anteriores, que se fará necessário o uso de uma elevada quantidade de portas lógicas básicas e diferentes, implicando no uso de muitas pastilhas lógicas digitais diferentes.

Outra opção é sintetizar as expressões para D_2 , D_1 e D_0 através de um circuito lógico mais simples usando DEC ou MUX.

9.4 Opções de Síntese

O circuito combinacional necessário para colocar em marcha os 3 FF's necessários neste contador pode ser sintetizado usando portas DEC ou MUX. Neste caso, as equações se modificam para:

$$D_2 = \sum_m \{2, 4, 11, 14\}$$

$$D_2 = \sum_m \{0, 4, 5, 9, 12\}$$

$$D_0 = \sum_m \{3, 5, 8, 9, 11\}$$

Solução usando DEC

Se fosse optado por um DEC de 4 para 16 linhas teríamos que realizar:

$$\begin{aligned}
 D_2 &= \overline{O_2} + \overline{O_4} + \overline{O_{11}} + \overline{O_{14}} \\
 &= \overline{O_2 + O_4 + O_{11} + O_{14}} \\
 &= \overline{O_2 \cdot O_4 + O_{11} + O_{14}} \\
 &= \text{NAND}(4)
 \end{aligned}$$

$$\begin{aligned}
 D_1 &= \overline{O_0 \cdot O_4 \cdot O_5 \cdot O_9 \cdot O_{12}} \\
 &= \text{NAND}(5)
 \end{aligned}$$

$$\begin{aligned}
 D_0 &= \overline{O_3 \cdot O_5 \cdot O_8 \cdot O_9 \cdot O_{11}} \\
 &= \text{NAND}(5)
 \end{aligned}$$

Resumindo:

- 1x CI DEC 4/16: 74LS154 (24 pinos!).
- 1x porta NAND(4): 74LS20 (2x NAND(4)) (50% seria usado).
- 2x portas NAND(5): não existente comercialmente! Teríamos que optar pelo CI 74LS30 (1x NAND(8)) e conectar à +Vcc as entradas não usadas de cada NAND(8), ou seja, usaríamos: 2x 74LS30.

total de pastilhas = 4 pastilhas, sendo que uma delas é de 24 pinos.

Solução usando MUX

Outra opção é optar pelo uso de MUX para síntese de uma função lógica. A princípio seria necessário adotar MUX de 1 para 16 linhas de saída (CI de 24 pinos). Mas podemos acomodar a mesma expressão lógica num MUX mais reduzido, de 1 para 8 linhas: CI 74LS151, separando a variável de entrada "mais significativa" M . Porém seriam necessários 3 deles, um para D_2 , outro para D_1 e por último um para D_0 . Este CI apresenta uma saídas Z e \overline{Z} . No caso do MUX(8), teríamos que "programá-lo" como demonstrado na tabela 9.2

Esta solução exigiria:

- 3x MUX(8) = 3x CI 74LS151;
- 1 porta NOT = CI 74LS04 (usada apenas 1/6)

total = 4 pastilhas.

Eventualmente a vantagem em se optar por usar MUX ao invés de DEC seja, que a "programação" da lógica de controle do circuito fica mais fácil. E fácil de ser modificada ou mesmo adaptada para qualquer outro contador síncrono que exija uma entrada externa de controle do tipo M e seja passível de assumir até 8 estados distintos.

Ref	ref'	M	q2 q1 q0	Q2 Q1 Q0	D2 D1 D0	D2	MUX	D1	MUX	D0	MUX
0	0	0	000	010	010	0	0	1	\overline{M}	0	M
1	1	0	001	000	000	0	0	0	M	0	M
2	2	0	010	100	100	1	\overline{M}	0	0	0	0
3	3	0	011	001	001	0	M	0	0	1	1
4	4	0	100	110	110	1	\overline{M}	1	1	0	0
5	5	0	101	011	011	0	0	1	\overline{M}	1	\overline{M}
6	6	0	110	000	000	0	M	0	0	0	0
7	7	0	111	XXX	XXX	X	0	X	0	X	0
8	0	1	000	001	001	0	-	0	-	1	-
9	1	1	001	011	011	0	-	1	-	1	-
10	2	1	010	000	000	0	-	0	-	0	-
11	3	1	011	101	101	1	-	0	-	1	-
12	4	1	100	010	010	0	-	1	-	0	-
13	5	1	101	000	000	0	-	0	-	0	-
14	6	1	110	100	100	1	-	0	-	0	-
15	7	1	111	XXX	XXX	X	-	X	-	X	-

Tabela 9.2: Programação à ser adotada para os MUX'es associados com D_2 , D_1 e D_0 .

9.5 Circuito final

A fim de simplificar a montagem, a ideia é optar por alguma pastilha que já concentre vários FF's do tipo D. Reparamos que o CI 74LS175 possui 4 x FF's-D encapsulados no seu interior, seus FF's já se encontram com seus sinais internos de Clock interligados num único ponto comum (Clock sendo ativado por borda de subida) e ainda uma entrada assíncrona de Master Reset (pino \overline{MR}) – a figura 9.3 mostra sua pinagem, e a figura 9.4 seu diagrama lógico interno.

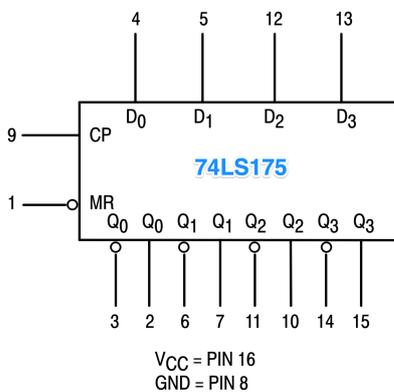


Figura 9.3: Pinagem do FF-D quádruplo 74LS175.

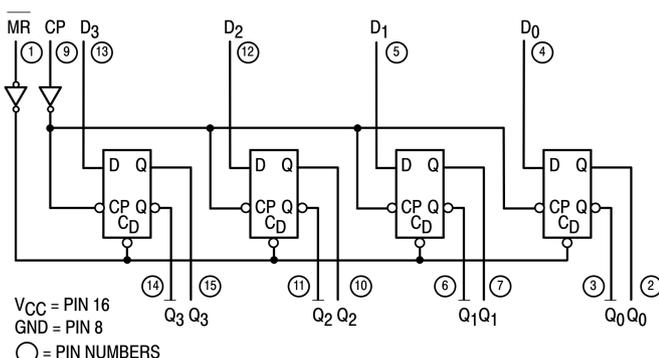


Figura 9.4: Diagrama lógico interno no CI 74LS175.

O digrama elétrico final aparece na figura 9.5.

9.6 Análise da evolução para estado 7

Se por acaso o circuito da figura 9.5 for inicializado em $7_{(10)}$, de acordo com a tabela 9.2 usada para “programar” os MUX'es, percebe-se que o próximo estado de evolução será 0.

9.7 Uso de FPGA

Note que este circuito poderia ter sido implementado numa pastilha FPGA (Field-Programmable Gate Array). Este tipo de pastilha permite programar o acionamento de Flip-Flop's através de uma “lookup table”. A lógica combinacional definida para acinar a(s) entrada(s) de controle do FF fica programada dentro de uma “lookup table”, ou “LUT”. A figura 9.6 mostra o diagrama lógico interno de uma LUT.

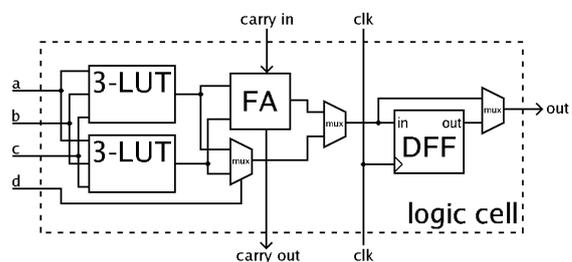


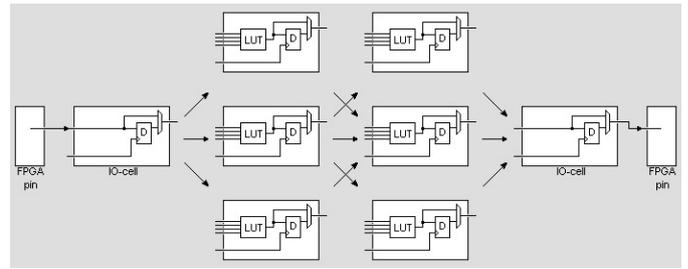
Figura 9.6: Exemplo de célula lógica básica numa pastilha FPGA.

Na figura 9.6, note a presença de um FF tipo D, um somador completo (FA = Full Adder) e 2 conjuntos de LUT para até 3 variáveis de entrada. Note que é comum que o circuito lógico correspondente a cada LUT seja formado usando um Multiplexador, no caso da figura mostrada, seria um MUX para até 8 linhas de entrada, os pontos a, b e c correspondem aos pinos de Select do MUX.

Note que o hardware associado com cada célula básica de um FPGA é similar à solução empregando MUX adotada para resolver este projeto. Neste caso, as equações de controle para cada uma das entradas *D* de nossos FF's são programas do MUX (LUT) associado com a célula que vai acionar um Flip-Flop em particular.

Alguns FPGA's não incluem na sua célula básica um somador completo. Neste caso, a saída do MUX é normalmente conectada à entrada de uma porta AND de 2 entradas, onde a segunda entrada da porta AND ficaria conectada ao ponto *d* (da figura 9.6). Note que este ponto *d* permite que o mesmo atue como um sinal de *Enable*.

Uma pastilha FPGA é formada por centenas ou milhares destas células lógicas básicas. Cada célula lógica pode ser conectada a outras células através de recursos de interconexão ("fusíveis") programados por software. Cada célula isoladamente pode fazer pouco, mas com muitas delas conectados juntas, funções lógicas complexas podem ser criadas através de células de Entrada/Saída (*IO-cells*) – ver figura 9.7.

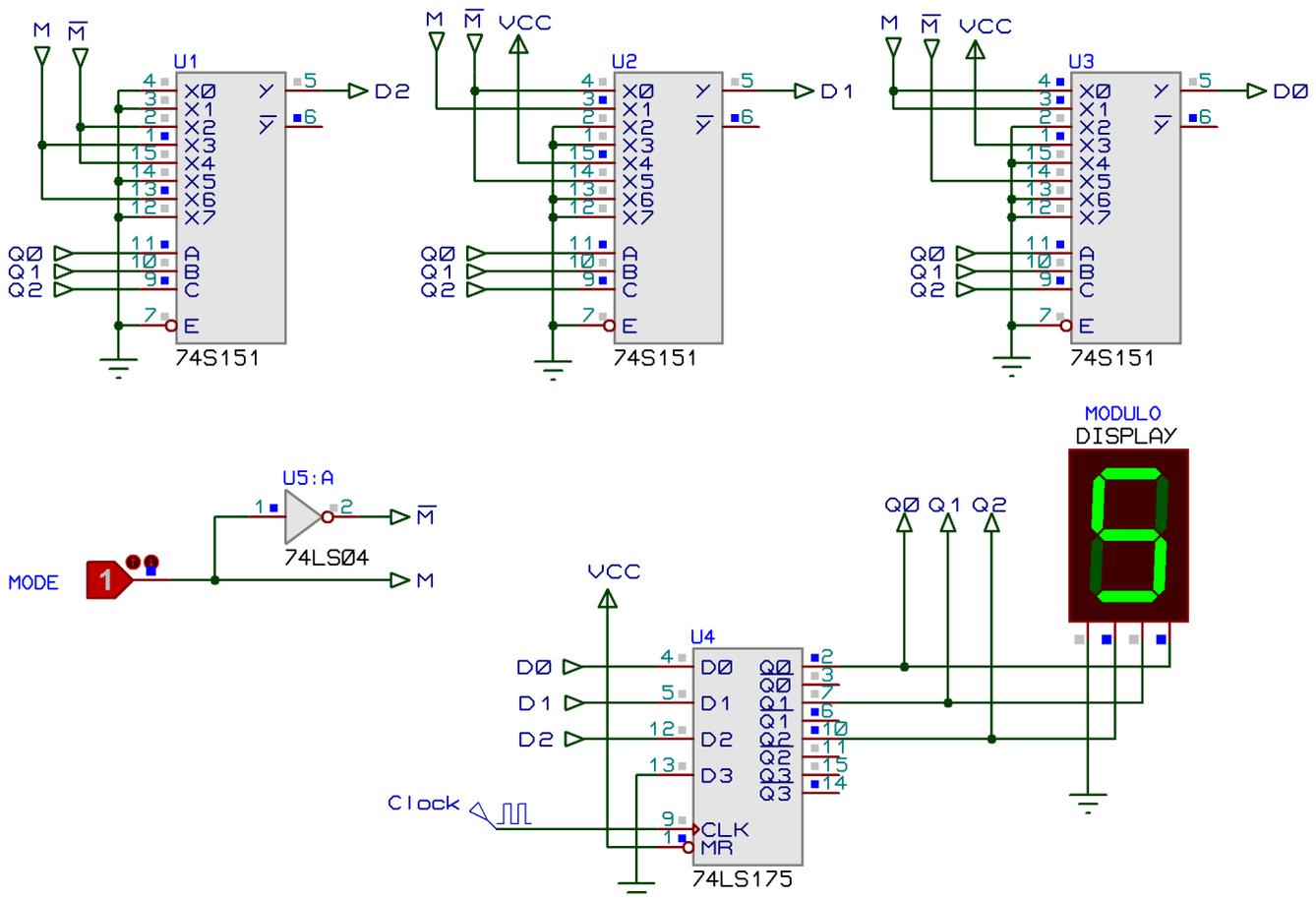


Fonte: <https://www.fpga4fun.com/FPGAinfo2.html> (27 April 2019)

Figura 9.7: Interconexão entre células básicas usando células de I/O

Lista de Componentes

- U1, U2, U3 = CI 74LS151 (3 pastilhas);
- U4 = CI 74LS175 (1 pastilha);
- U5 = CI 74LS04 (1 pastilha);
- 1x módulo display de 7-Segmentos.
- Fonte de alimentação compatível TTL;
- Gerador de Sinais (ou de funções), onda quadrada até 30 Hz.



	U1, U2, U3 (74LS151)	U4 (74LS175)	U5 (74LS04)
Vcc	Pin 16	Pin 16	Pin 14
GND	Pin 8	Pin 8	Pin 7

Figura 9.5: Circuito final do contador.