INF01058

Circuitos Digitais

Síntese de FSM com FF D

Aula 21b

1. Introdução

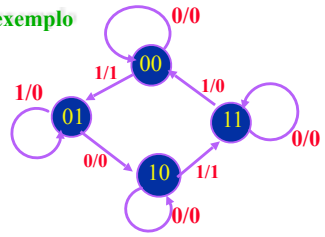
- processo de projeto
  - especificação (p.ex. FSM)
  - ⇓ tabela de estados
  - ⇓ equações de entrada (para FF's) e de saída
  - ⇓ síntese a partir das equações (problema de lógica combin.)
- número de flip-flops
  - com codificação: n FF's → 2^n estados
  - sem codificação: n FF's → n estados
- escolha do tipo dos FF's
  - influencia determinação das equações de entrada

2. Projeto com flip-flops tipo D

processo de projeto

1. Obter tabela de estados
2. Derivar equações de entrada a partir do "próximo estado" na tabela
3. Derivar equações de saída a partir da "saída" na tabela
4. Simplificar equações de entrada e saída
5. Desenhar circuito lógico com FF's D e portas lógicas de acordo com as equações.

exemplo



4 estados → 2 FF's: A, B

1 entrada X

1 saída Y

Tabela de Estados

Estado Atual A	Estado Atual B	Entrada X	Próx. Estado A	Próx. Estado B	Saída Y	MINTERMO
0	0	0	0	0	0	0
0	0	1	0	1	1	1
0	1	0	1	0	0	2
0	1	1	0	1	0	3
1	0	0	1	0	0	4
1	0	1	1	1	1	5
1	1	0	1	1	0	6
1	1	1	0	0	0	7

equações de entrada para FF's tipo D

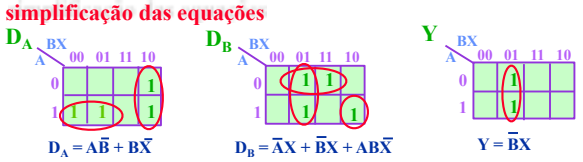
$A(t+1) = D_A(A,B,X) = \Sigma\, m\,(2,4,5,6)$

$B(t+1) = D_B(A,B,X) = \Sigma\, m\,(1,3,5,6)$

equação de saída

$Y(A,B,X) = \Sigma\, m\,(1,5)$

simplificação das equações

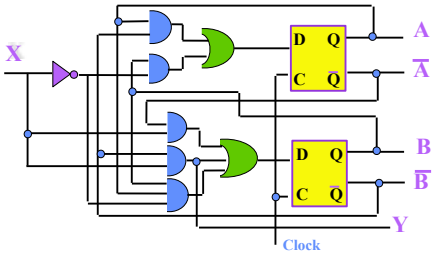


$D_A = A\bar{B} + B\bar{X}$

$D_B = \bar{A}X + \bar{B}X + AB\bar{X}$

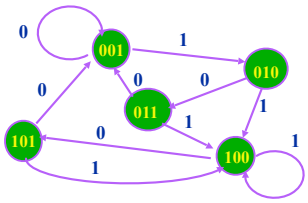
$Y = \bar{B}X$

"desenhar" circuito lógico



Reconhecer:
- entrada
- saída
- FF's
- realimentações
- lógica combinac.

3. Projeto com estados não utilizados



5 estados → 3 FF's

estados não utilizados:
- 000
- 110
- 111

saídas do circuito = saída dos FF's

Tabela de estados

Estado Atual A	B	C	Entrada X	Próx. Estado A	B	C	MINTERMO
0	0	1	0	0	0	1	2
0	0	1	1	0	1	0	3
0	1	0	0	0	1	1	4
0	1	0	1	1	0	0	5
0	1	1	0	0	0	1	6
0	1	1	1	1	0	0	7
1	0	0	0	1	0	1	8
1	0	0	1	1	0	0	9
1	0	1	0	0	0	1	10
1	0	1	1	1	0	0	11

equações de entrada

$A(t+1) = D_A = \Sigma\, m\,(5,7,8,9,11)$

$B(t+1) = D_B = \Sigma\, m\,(3,4)$

$C(t+1) = D_C = \Sigma\, m\,(2,4,6,8,10)$

simplificação

Mintermos don't care: 0,1,12,13,14,15

D_A (CX / AB)

AB \ CX	00	01	11	10
00	X	X		
01		1	1	
11	X	X	X	X
10	1	1	1	

$D_A = AX + BX + \bar{B}\bar{C}$

D_B (CX / AB)

AB \ CX	00	01	11	10
00	X	X	1	
01	1			
11	X	X	X	X
10				

$D_B = \bar{A}\bar{C}\bar{X} + \bar{A}\bar{B}X$

D_C (CX / AB)

AB \ CX	00	01	11	10
00	X	X		1
01	1			1
11	X	X	X	X
10	1			1

$D_C = \bar{X}$

4. Escolha dos flip-flops

T — As váriáveis de estado vão de 0 → 1 e de volta 1 → 0 (ex. contadores)

D — Quando a informação de entrada deve ser armazenada por um tempo

SR — Quando sinais diferentes podem dar SET ou RESET nos flip-flops

JK — Quando queremos combinar as vantagens de um FF T com SR

SR e JK — Tendem a reduzir o custo das equações de entrada, mas demandam até o dobro de conexões do que os FF's D e T

Como os FF's D e T requerem um número menor de conexões, são preferidos para implementações VLSI