

IEE 2712 Sistemas Digitales

Clase 12

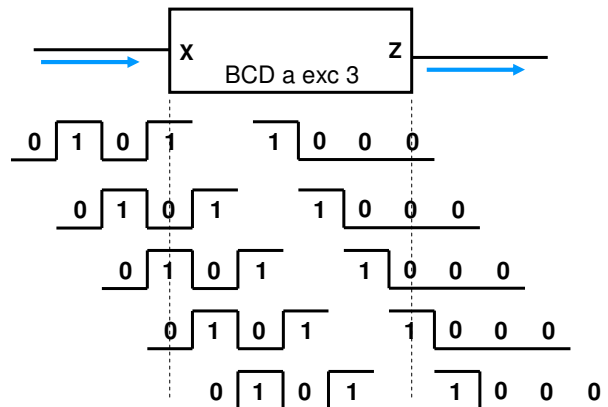
Objetivos educacionales:

1. Encontrar el diagrama de estado desde la descripción verbal.
2. Diseñar circuitos combinacionales con flip-flops D y JK

4.5 Diseño de circuitos secuenciales

Ejemplo: Convertidor serial de BCD a exceso de 3

Diseñar un circuito que tenga una entrada y una salida. La salida debe entregar el código exceso de 3 (el bit menos significativo primero) de la entrada que es un código BCD (el bit menos significativo primero también).



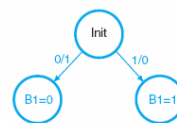
4.5 Diseño de circuitos secuenciales

Sequences in Order of Digits Represented								Sequences in Order of Common Prefixes							
BCD Input				Excess-3 Output				BCD Input				Excess-3 Output			
1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0
1	0	0	0	0	0	1	0	0	0	0	1	1	1	0	1
0	1	0	0	1	0	1	0	0	0	1	0	1	1	1	0
1	1	0	0	0	1	1	0	0	1	0	0	1	0	1	0
0	0	1	0	1	1	1	0	0	1	1	0	1	0	0	1
1	0	1	0	0	0	0	1	1	0	0	0	0	0	1	0
0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1
1	1	1	0	0	1	0	1	1	0	1	0	0	0	0	1
0	0	0	1	1	1	0	1	1	1	0	0	0	1	1	0
1	0	0	1	0	0	1	1	1	1	1	0	0	1	0	1

4.5 Diseño de circuitos secuenciales

BCD Input				Excess-3 Output			
1	2	3	4	1	2	3	4
0	0	0	0	1	1	0	0
0	0	0	1	1	1	0	1
0	0	1	0	1	1	1	0
0	1	0	0	1	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	0	0	0	1
1	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1

Primer dígito:



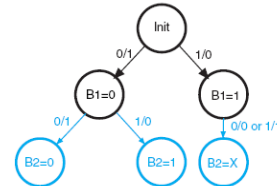
Indica que se necesita almacenar el estado

BCD Input				Excess-3 Output			
1	2	3	4	1	2	3	4
0	0	0	0	1	1	0	0
0	0	0	1	1	1	1	0
0	0	1	0	1	1	1	0
0	1	0	0	1	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	0	0	0	1
1	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1

4.5 Diseño de circuitos secuenciales

BCD Input				Excess-3 Output			
1	2	3	4	1	2	3	4
0	0	0	0	1	1	0	0
0	0	0	1	1	1	0	1
0	0	1	0	1	1	1	0
0	1	0	0	1	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	0	0	0	1
1	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1

Segundo dígito:



BCD Input				Excess-3 Output			
1	2	3	4	1	2	3	4
0	0	0	0	1	1	0	0
0	0	0	1	1	1	0	1
0	0	1	0	1	1	1	0
0	1	0	0	1	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	0	0	0	1
1	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1

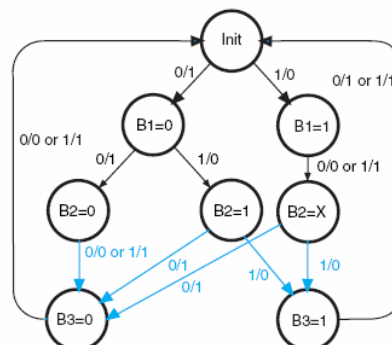
Indica que NO se necesita almacenar el estado



4.5 Diseño de circuitos secuenciales

BCD Input				Excess-3 Output			
1	2	3	4	1	2	3	4
0	0	0	0	1	1	0	0
0	0	0	1	1	1	0	1
0	0	1	0	1	1	1	0
0	1	0	0	1	0	1	0
0	1	1	0	1	0	0	1
1	0	0	0	0	0	1	0
1	0	0	1	0	0	1	1
1	0	1	0	0	0	0	1
1	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1

Diagrama de estado completo:



4.6 Diseño empleando flip-flops D

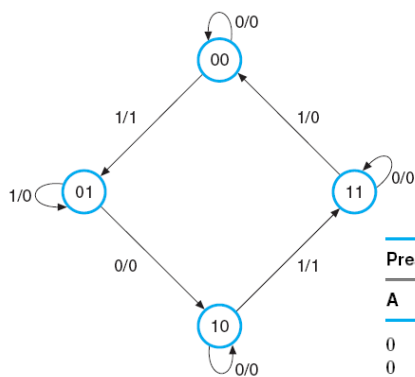
Empleando flip-flops D

D	$Q(t + 1)$	Operation
0	0	Reset
1	1	Set

El estado siguiente es el mismo que la entrada actual al flip-flop, por lo que hay que encontrar las entradas al flip-flop (estado siguiente) según la tabla de estados.

4.6 Diseño empleando flip-flops D

Ejemplo: empleando flip-flops D



Present State		Input	Next State		Output
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	0	0

4.6 Diseño empleando flip-flops D

Present State		Input X	Next State		Output Y
A	B		A	B	
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	1	1	0
1	1	1	0	0	0

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

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

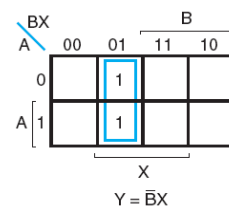
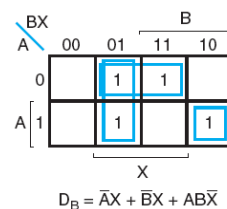
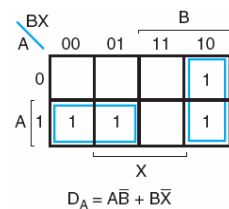
$$Y(A, B, X) = \sum m(1, 5)$$

4.6 Diseño empleando flip-flops D

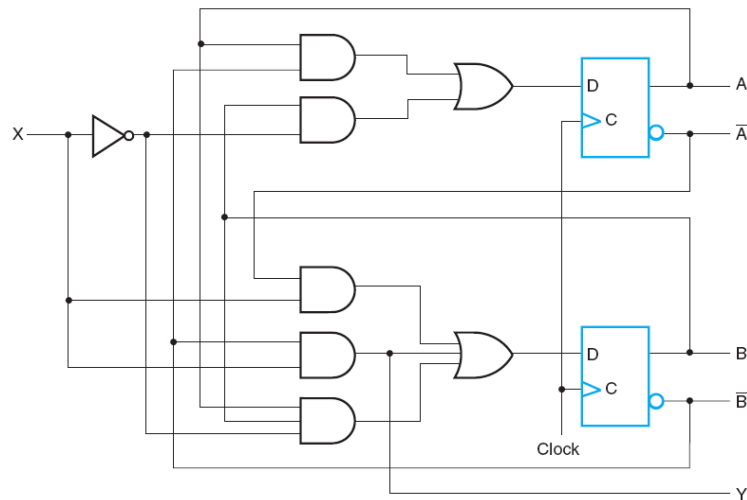
$$A(t+1) = D_A(A, B, X) = \sum m(2, 4, 5, 6)$$

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

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



4.6 Diseño empleando flip-flops D



4.6 Diseño empleando flip-flops D

Estados no usados (el número de estados que se necesitan no es una potencia de dos): se pueden usar para simplificar las expresiones.

Peligro: si por alguna razón se cae en uno de esos estados, el comportamiento del circuito es impredecible.

Present State			Input X	Next State		
A	B	C		A	B	C
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0

4.6 Diseño empleando flip-flops D

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

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

$D_C = \bar{X}$

Present State			Input	Next State		
A	B	C	X	A	B	C
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	1	1
0	1	0	1	1	0	0
0	1	1	0	0	0	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	1	0	0

4.7 Diseño empleando flip-flops JK

Ejemplo: empleando flip-flops JK

El diseño se realiza igual que para flip-flops D con la diferencia que ahora la entrada no es directamente el próximo estado sino una transición dada por el flip-flop.

Tablas de excitación:

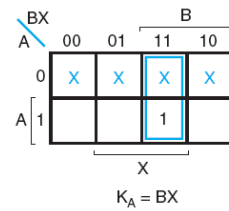
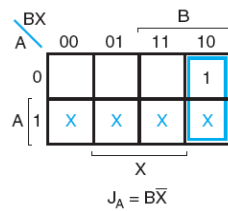
(a) JK Flip-Flop				(b) SR Flip-Flop			
$Q(t)$	$Q(t+1)$	J	K	$Q(t)$	$Q(t+1)$	S	R
0	0	0	X	0	0	0	X
0	1	1	X	0	1	1	0
1	0	X	1	1	0	0	1
1	1	X	0	1	1	X	0

(c) D Flip-Flop			(d) T Flip-Flop		
$Q(t)$	$Q(t+1)$	D	$Q(t)$	$Q(t+1)$	T
0	0	0	0	0	0
0	1	1	0	1	1
1	0	0	1	0	1
1	1	1	1	1	0

4.7 Diseño empleando flip-flops JK

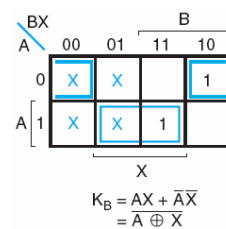
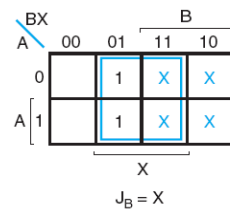
Ejemplo

Present State		Input	Next State		Flip-Flop Inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



4.7 Diseño empleando flip-flops JK

Present State		Input	Next State		Flip-Flop Inputs			
A	B	X	A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



4.7 Diseño empleando flip-flops JK

