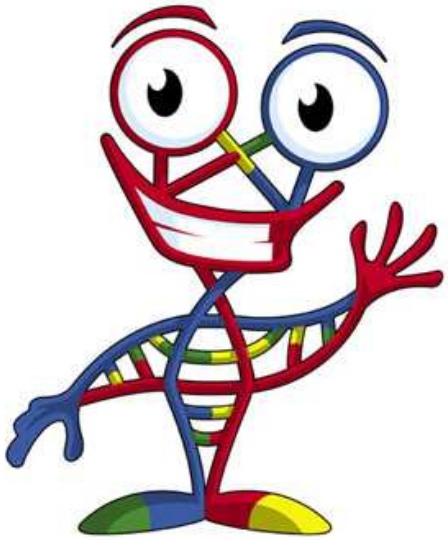
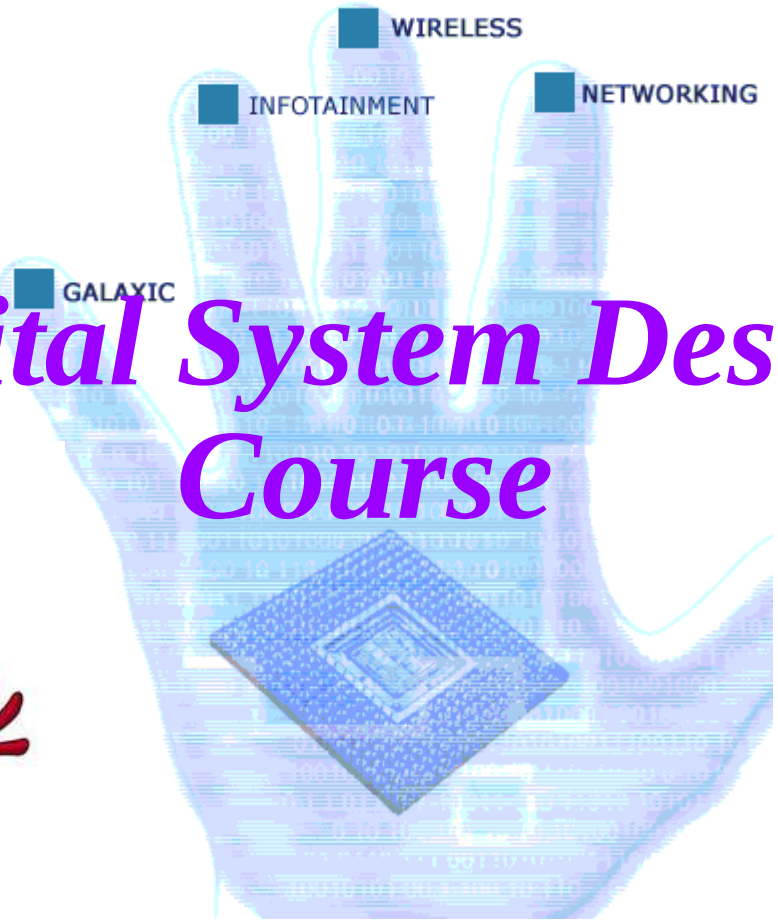
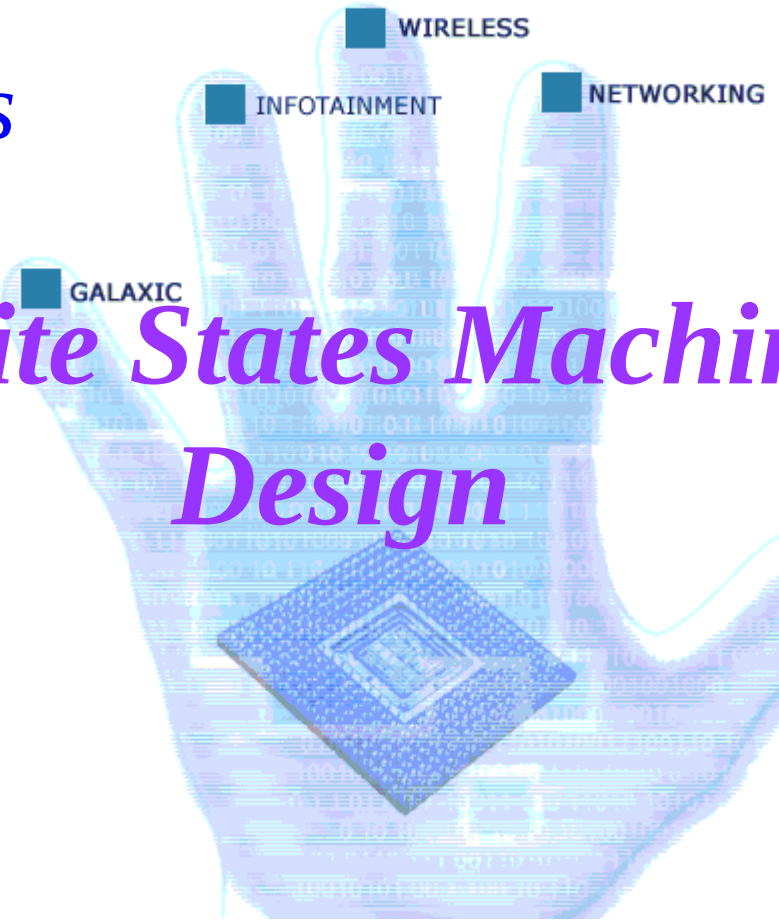




Digital System Design Course

creatures



Finite States Machine Design

FSM Design

Diseño de Máquinas de Estado

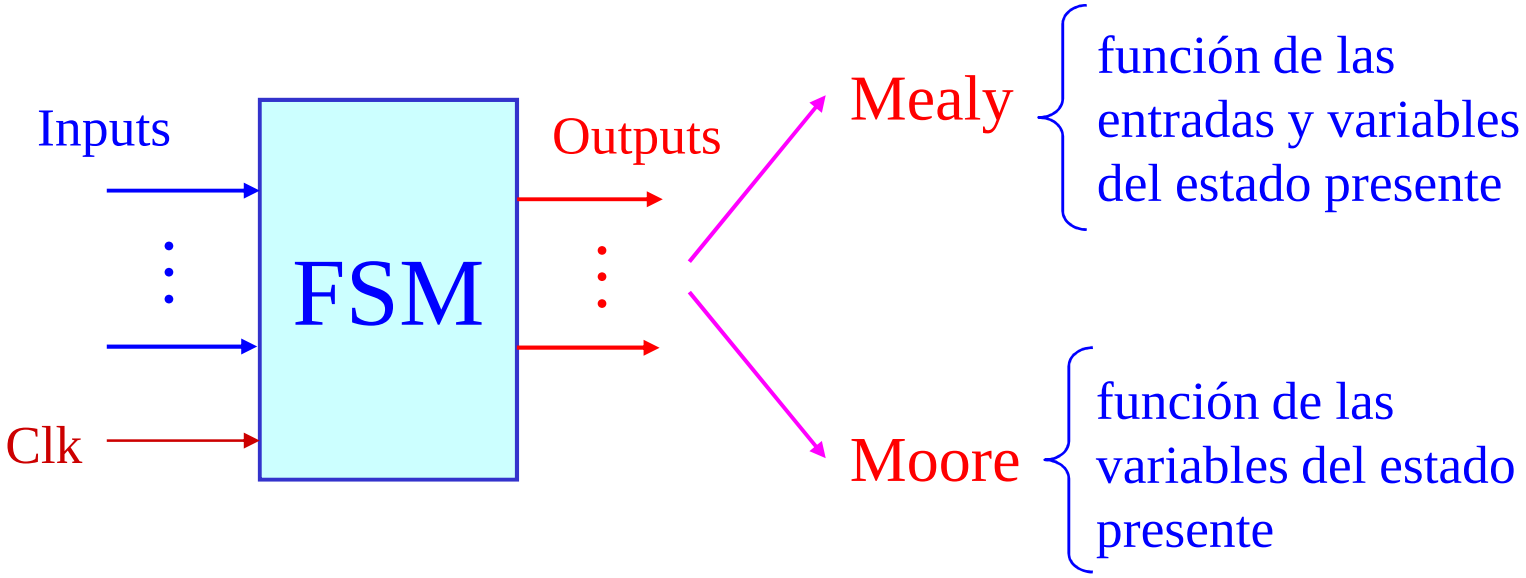
- Una *metodología manual* para diseñar *máquinas de estado* permite diseñar *circuitos secuénciales sencillos* o simples tales como: *contadores*, *detectores de secuencia*, etc
- El diseño de circuitos secuénciales sencillos requiere de un *procedimiento estructurado* que utiliza varias etapas

Diseño de Máquinas de Estado

- *Procedimiento estructurado*
  - Entender las *especificaciones* del problema
  - Analizar las *entradas/salidas* del circuito secuencial → determinar el tipo de *FSM: Mealy o Moore*
  - Usar una o varias *representaciones* para describir *FSMs*
  - *Reducir* y *codificar estados* → *tabla de estados codificada*
  - Seleccionar el tipo de *flip-flop*
  - Deducir la *lógica combinatoria para el próximo estado*
  - Deducir la *lógica combinatoria para la salida*
  - Implementar *el circuito secuencial o FSM*

Diseño de Circuitos Secuénciales

- Entender las *especificaciones* o consideraciones, es decir, entender la descripción del problema
- Analizar las *entradas/salidas* del circuito secuencial



Diseño de Circuitos Secuénciales

- Usar una o varias *representaciones* para describir las máquinas de estado:
  - *Diagrama de estados*
  - *Tabla de estados*
  - *Diagrama de flujo*
  - *Diagramas ASM: Algorithm State Machine*
- Reducir de estados

Diseño de Circuitos Secuénciales

- Codificar los estados
  - Seleccionar un método de codificación
  - Asignar el código binario a los estados y a las salidas:
    - diagrama de estados
    - tabla de estados: representar la tabla de estados en código binario
  - Generalmente, las diferentes codificaciones de los estados conduce a circuitos lógicos diferentes, algunos de los cuales son mejores (más simples o más económicos)

Diseño de Circuitos Secuénciales

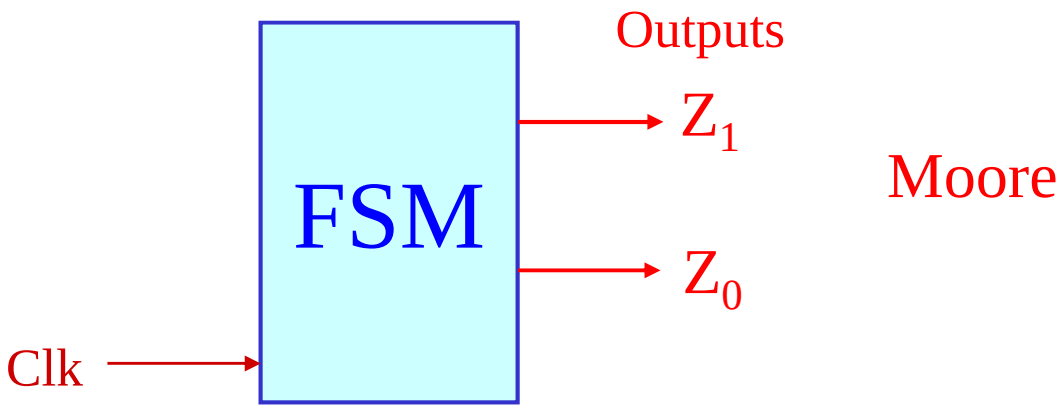
- Seleccionar el tipo de flip-flop, es decir, el flip-flop puede ser: D, JK, RS o T
- Deducir la lógica combinatoria para el próximo estado, es decir, la lógica para la excitación de los flip-flops. En este caso, usar mapas de Karnaugh
- Deducir la lógica combinatoria para la salida. En este caso usar mapas de Karnaugh
- Implementar el circuito secuencial o FSM

creatures

Binary counters design

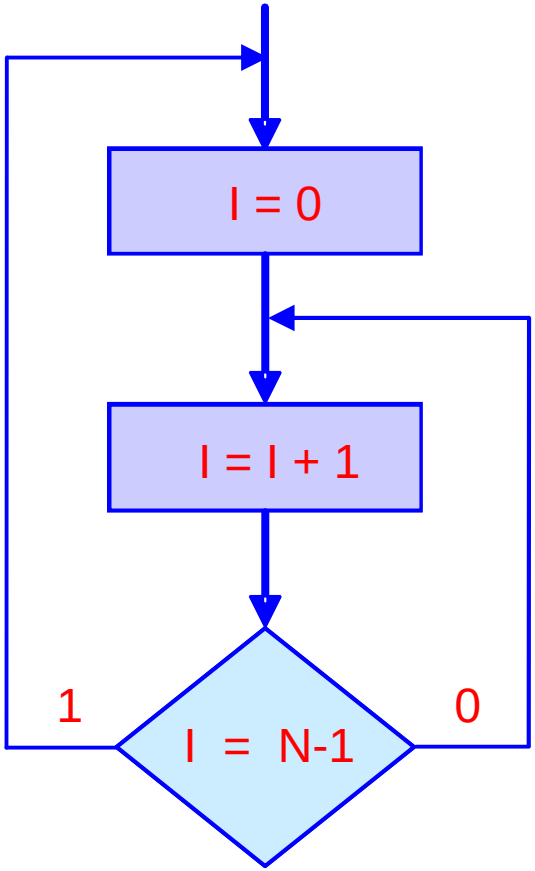
Ejemplo 1: Diseño de un Contador Binario

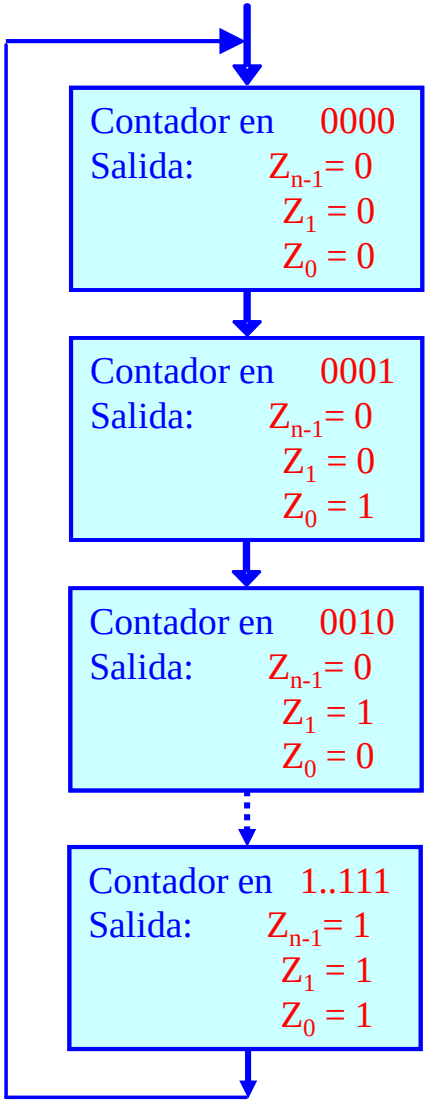
- Contador binario síncrono de 2 bits o módulo 4
  - Cuenta cuatro eventos, es decir: 0,1,2,3 ó 1,2,3,4
  - Requiere dos salidas: Z_0 y Z_1
  - La lectura en código binario es: 00, 01, 10, 11, 00, 01, ...
- Representación del circuito secuencial o FSM



Ejemplo 1: Diseño de un Contador Binario

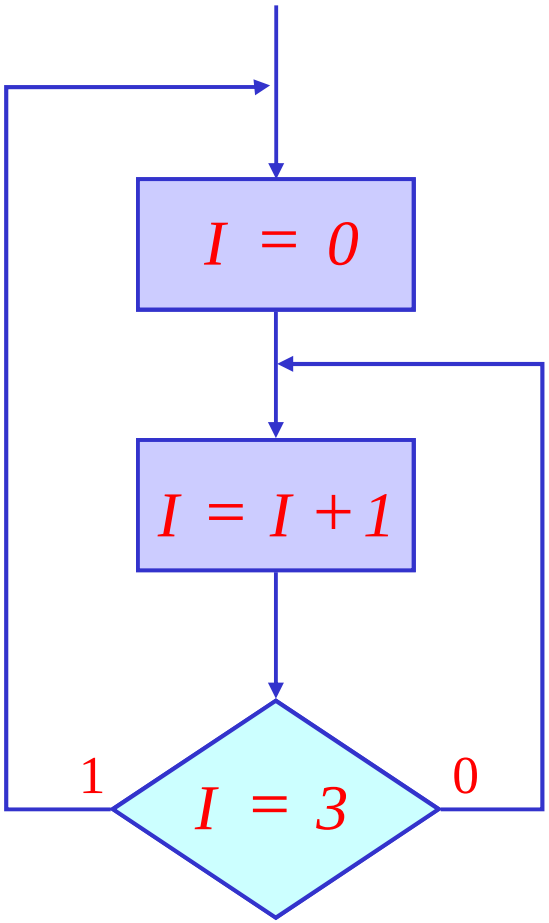
- Diagrama de flujo:

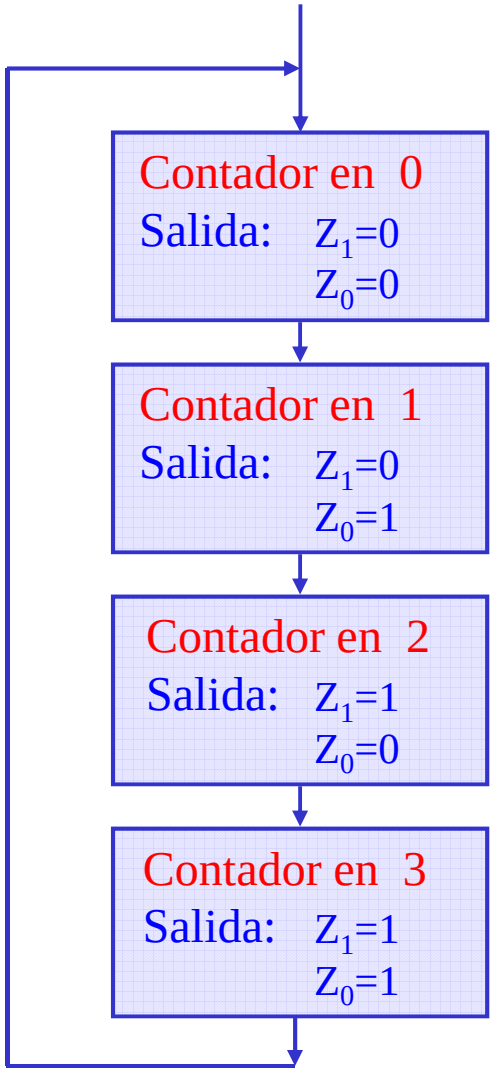




Ejemplo 1: Diseño de un Contador Binario

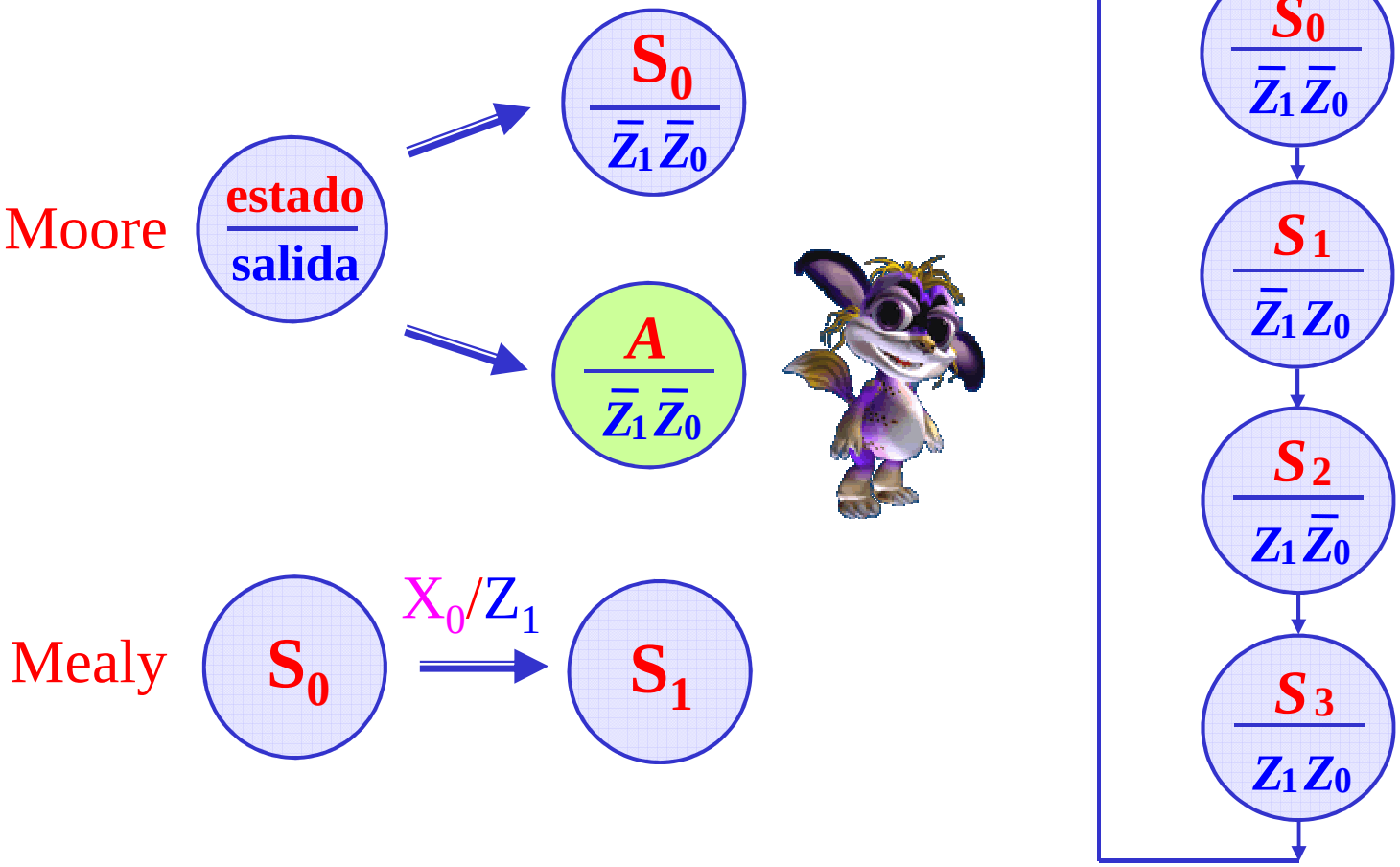
- Diagrama de flujo:





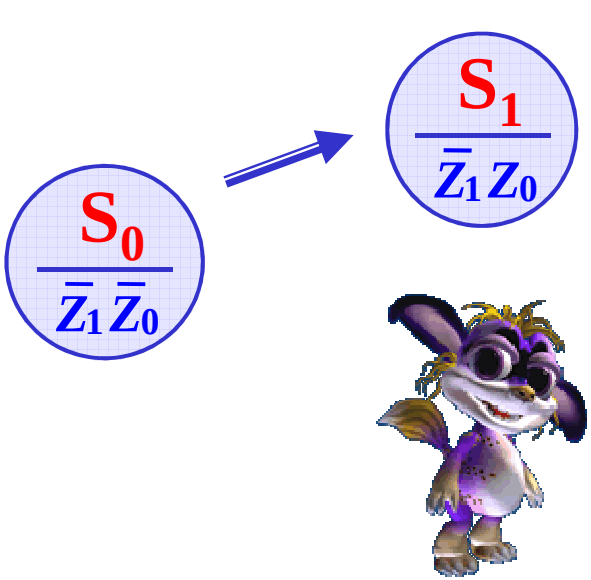
Ejemplo 1: Diseño de un Contador Binario

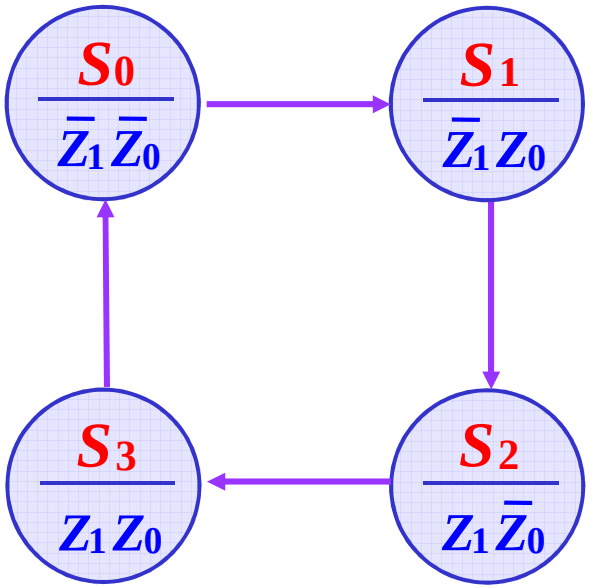
- Diagrama de estados:



Ejemplo 1: Diseño de un Contador Binario

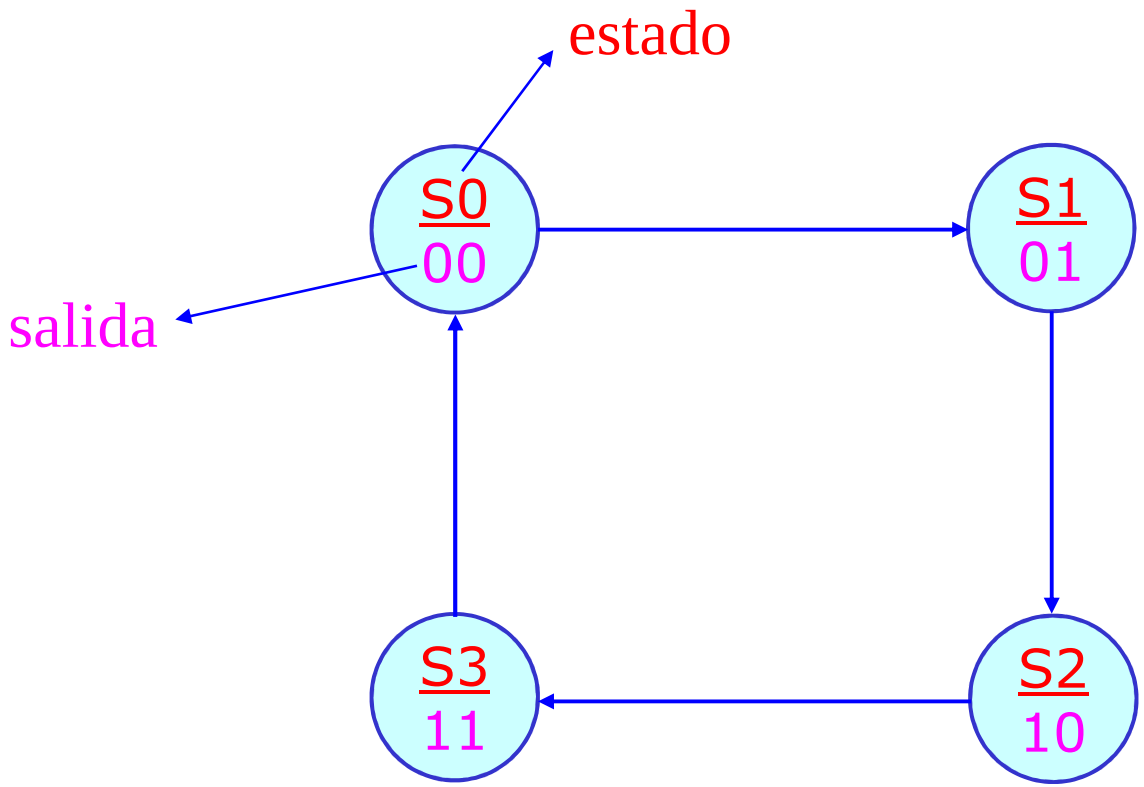
- Diagrama de estados:





Ejemplo 1: Diseño de un Contador Binario

- Diagrama de estados: *FSM tipo Moore*



Ejemplo 1: Diseño de un Contador Binario

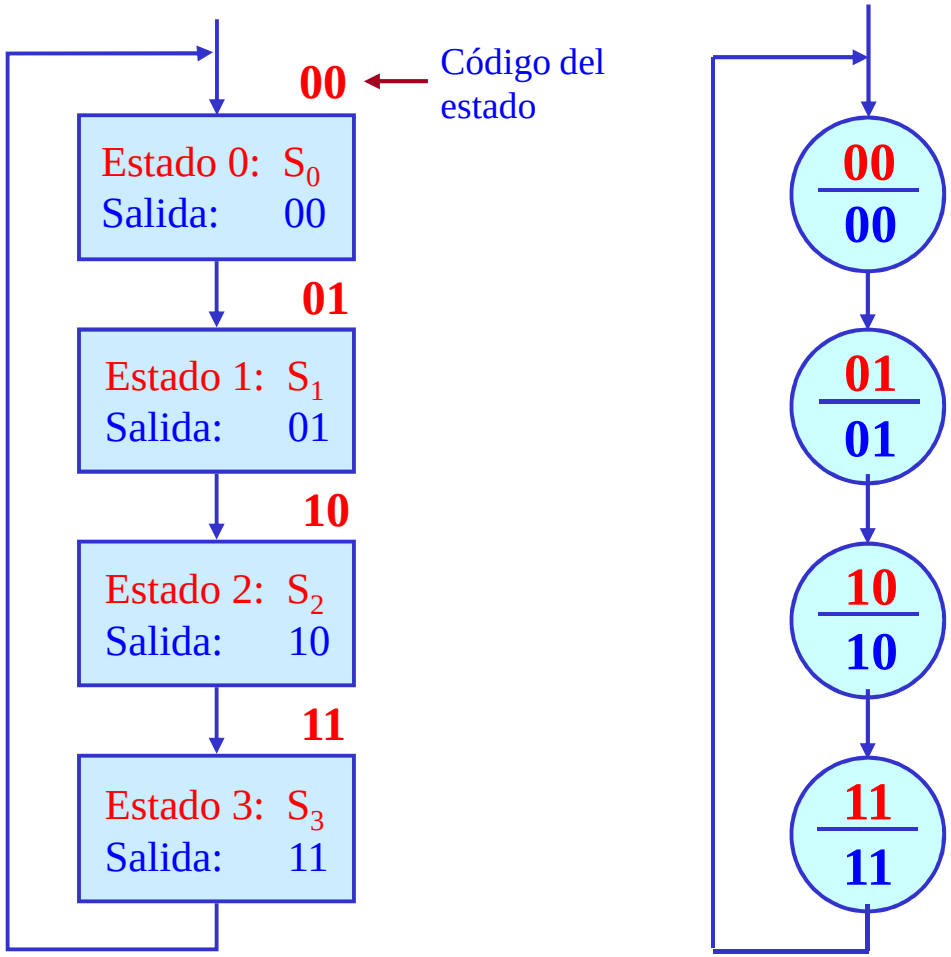
- Codificar estados: *codificación directa*

$S_0 = 00$

$S_1 = 01$

$S_2 = 10$

$S_3 = 11$



Ejemplo 1: Diseño de un Contador Binario

- Tabla de estados:
  - *La información de los diagramas de estado se puede colocar en forma tabular, es decir, en una tabla de estados*

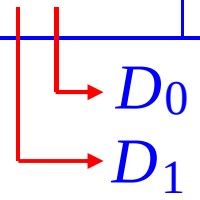
Entradas	EP	PE	**Salidas**
-	S_0	S_1	LL
-	S_1	S_2	LH
-	S_2	S_3	HL
-	S_3	S_0	HH

EP = Estado Presente
PE = Próximo Estado

Ejemplo 1: Diseño de un Contador Binario

- Tabla de estados *en código binario*

Entradas	EP	PE	**Salidas**
-	$Q_1^n\, Q_0^n$	$Q_1^{n+1}\, Q_0^{n+1}$	$Z_1 Z_0$
-	00	01	00
-	01	10	01
-	10	11	10
-	11	00	11



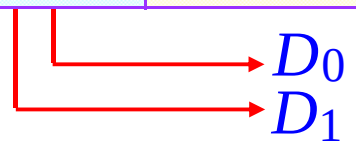
Ejemplo 1: Diseño de un Contador Binario

- Selección del flip-flop:
  - El flip-flop seleccionado es el D
  - El circuito emplea cuatro estados, entonces se requieren dos flip-flops D: FFD_1 y FFD_0
  - Las salidas de los flip-flops son Q_1 y Q_0. Entonces, las salidas son: $Q_1Q_0 \rightarrow 00, 01, 10, 11$
  - En este caso, el estado S_0 se representa por el estado de los flip-flops : $Q_1Q_0 = 00$

Ejemplo 1: Diseño de un Contador Binario

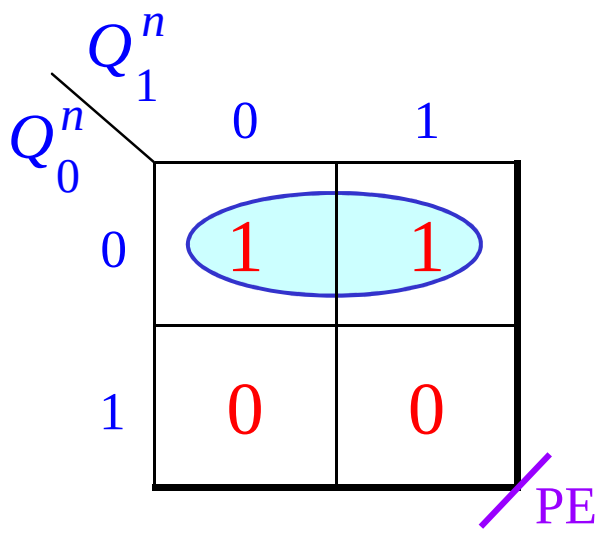
- Deducir la lógica del PE (lógica de excitación)
  - Para el flip-flop D, la tabla de transición es la misma del PE

EST	EP	IN	PE	OUT	D1	D0
-	Q_1Q_0	-	$Q_1^+Q_0^+$	Z_1Z_0	-	-
S0	00	-	01	00	0	1
S1	01	-	10	01	1	0
S2	10	-	11	10	1	1
S3	11	-	00	11	0	0

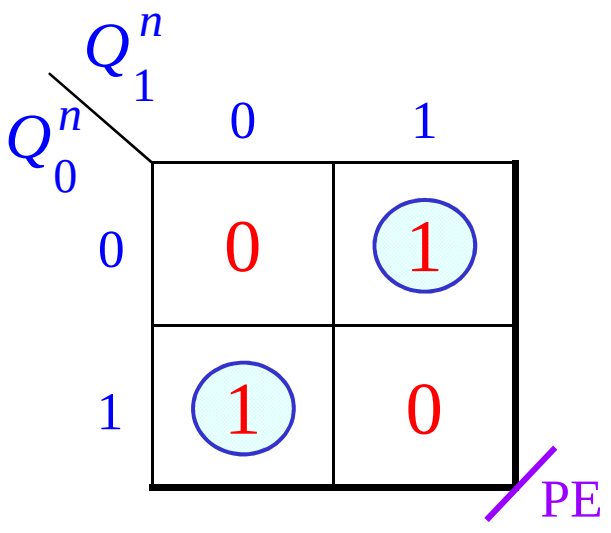


Ejemplo 1: Diseño de un Contador Binario

- Deducir la lógica del PE (lógica de excitación)
  - $Q_0^{n+1} \rightarrow$ PE para el flip-flop D_0: FFD_0
  - $Q_1^{n+1} \rightarrow$ PE para el flip-flop D_1: FFD_1



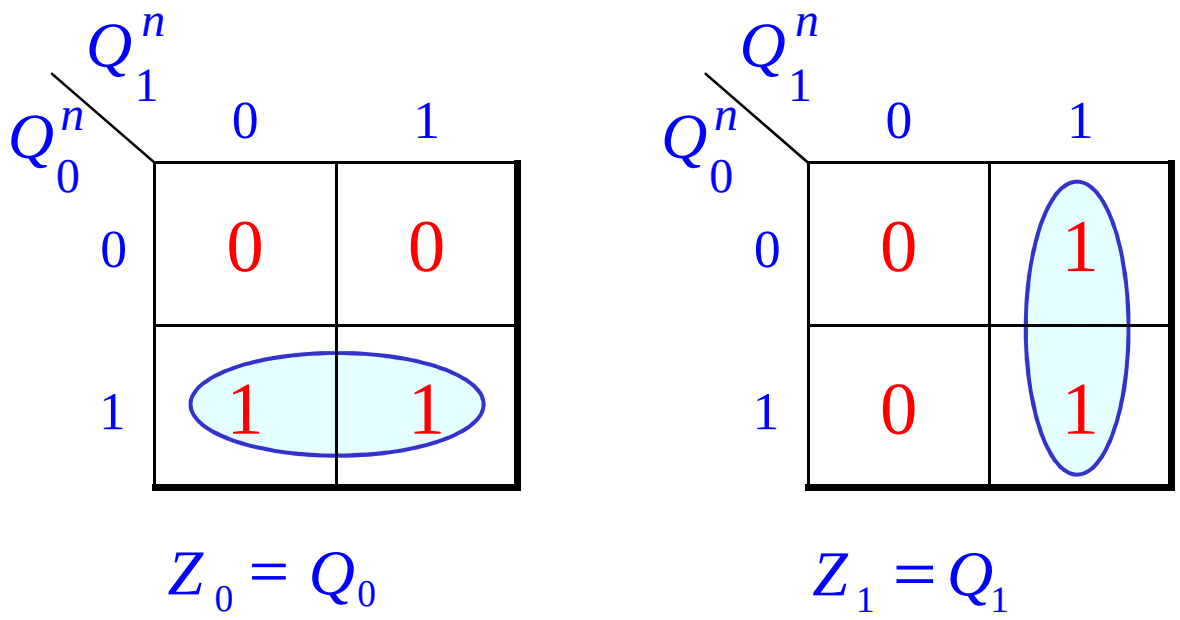
$$D_0 = \overline{Q}_0$$



$$D_1 = Q_1\overline{Q}_0 + \overline{Q}_1 Q_0$$

Ejemplo 1: Diseño de un Contador Binario

- Deducir la lógica de salida
  - $Z_0 \rightarrow$ Salida actual generada por FFD_0
  - $Z_1 \rightarrow$ Salida actual generada por FFD_1

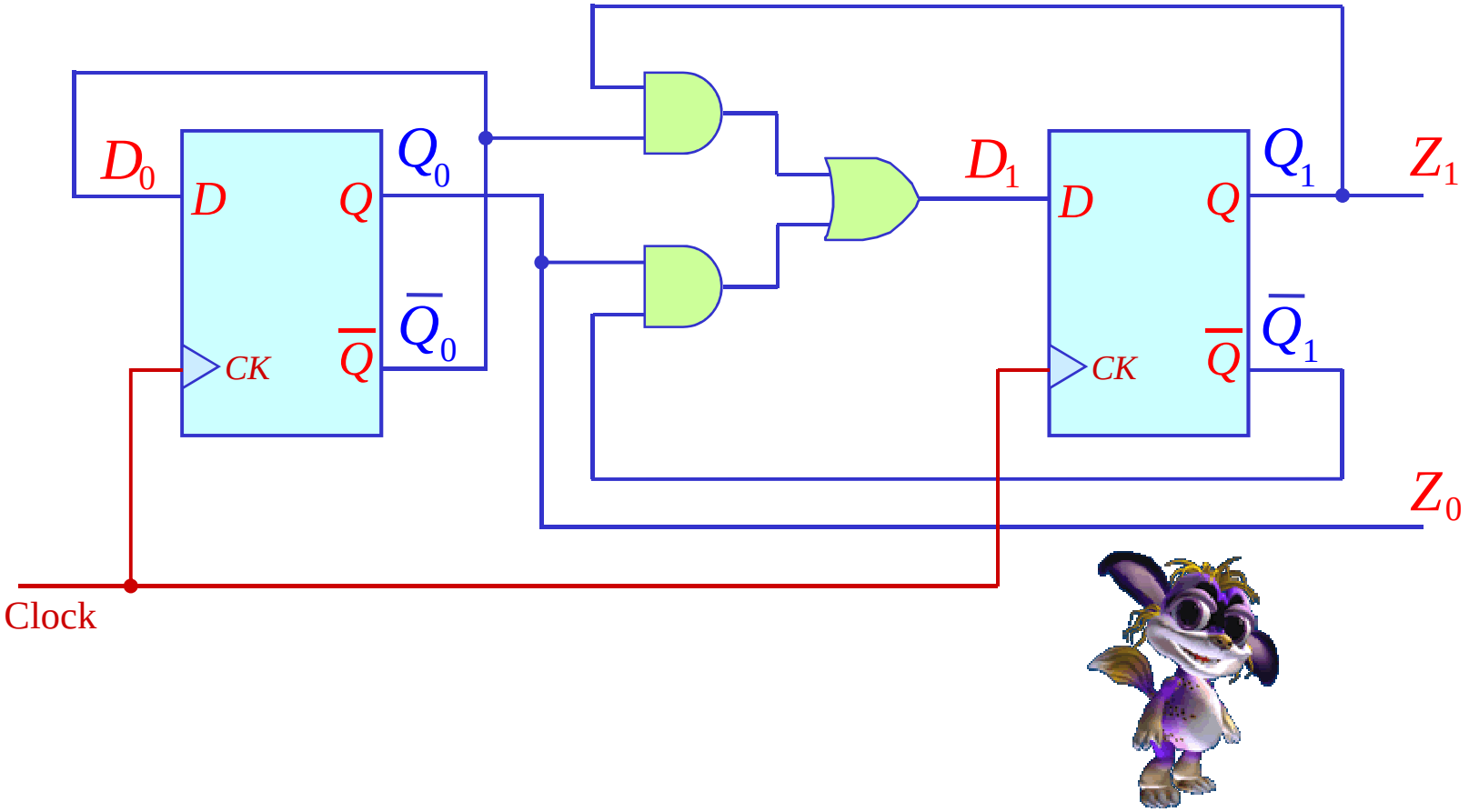


$Z_0 = Q_0$ $\qquad$ $Z_1 = Q_1$

NOTA: Cuando la salida actual es igual al estado actual, como en este diseño, no es necesario realizar los mapas de Karnaugh

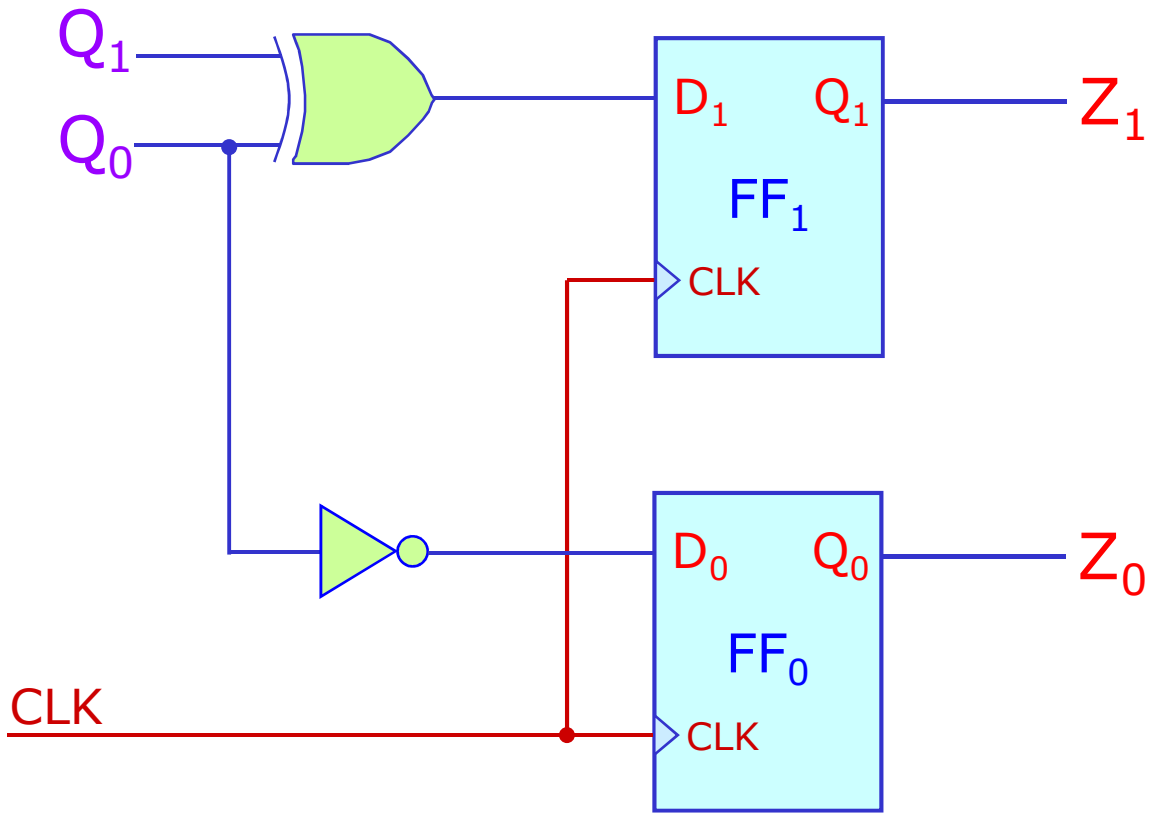
Ejemplo 1: Diseño de un Contador Binario

- Implementación: diagrama lógico general



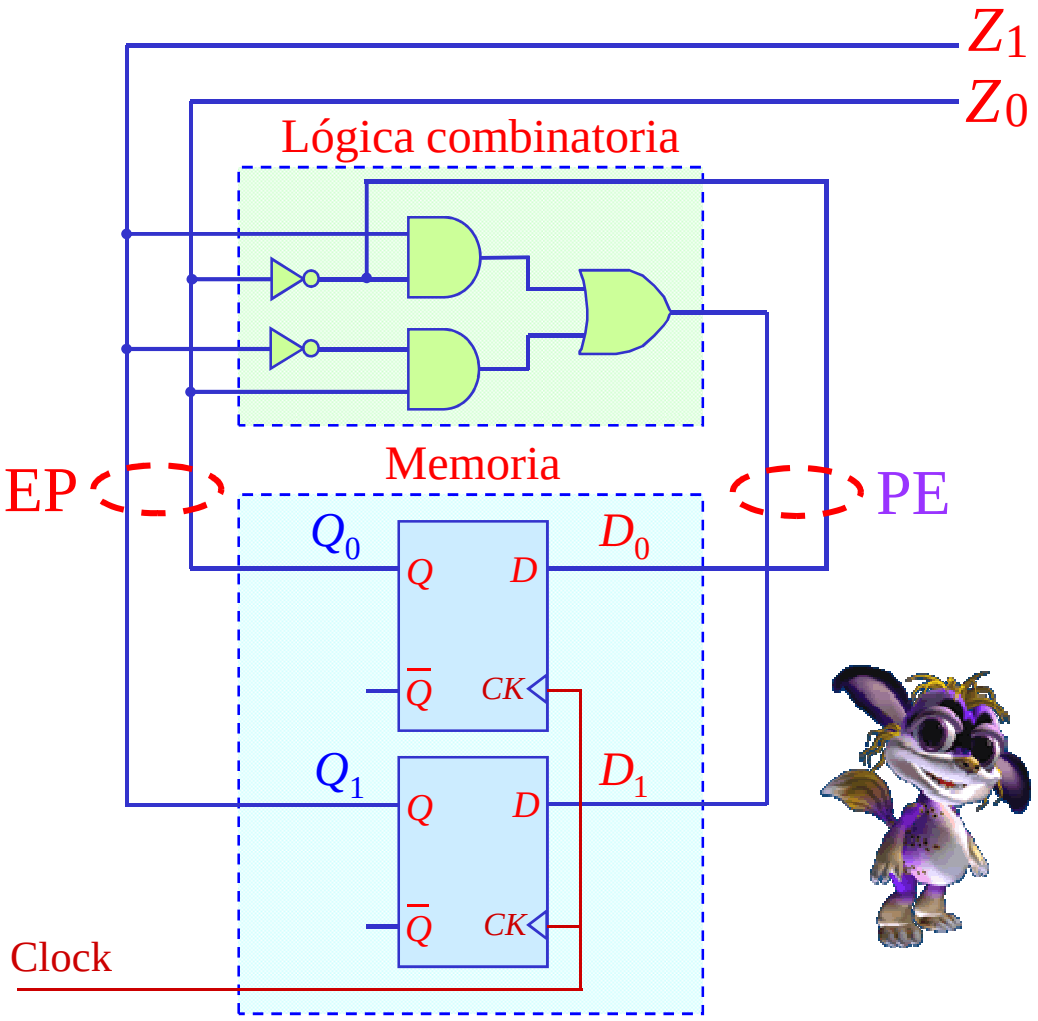
Ejemplo 1: Diseño de un Contador Binario

- Implementación: diagrama lógico general



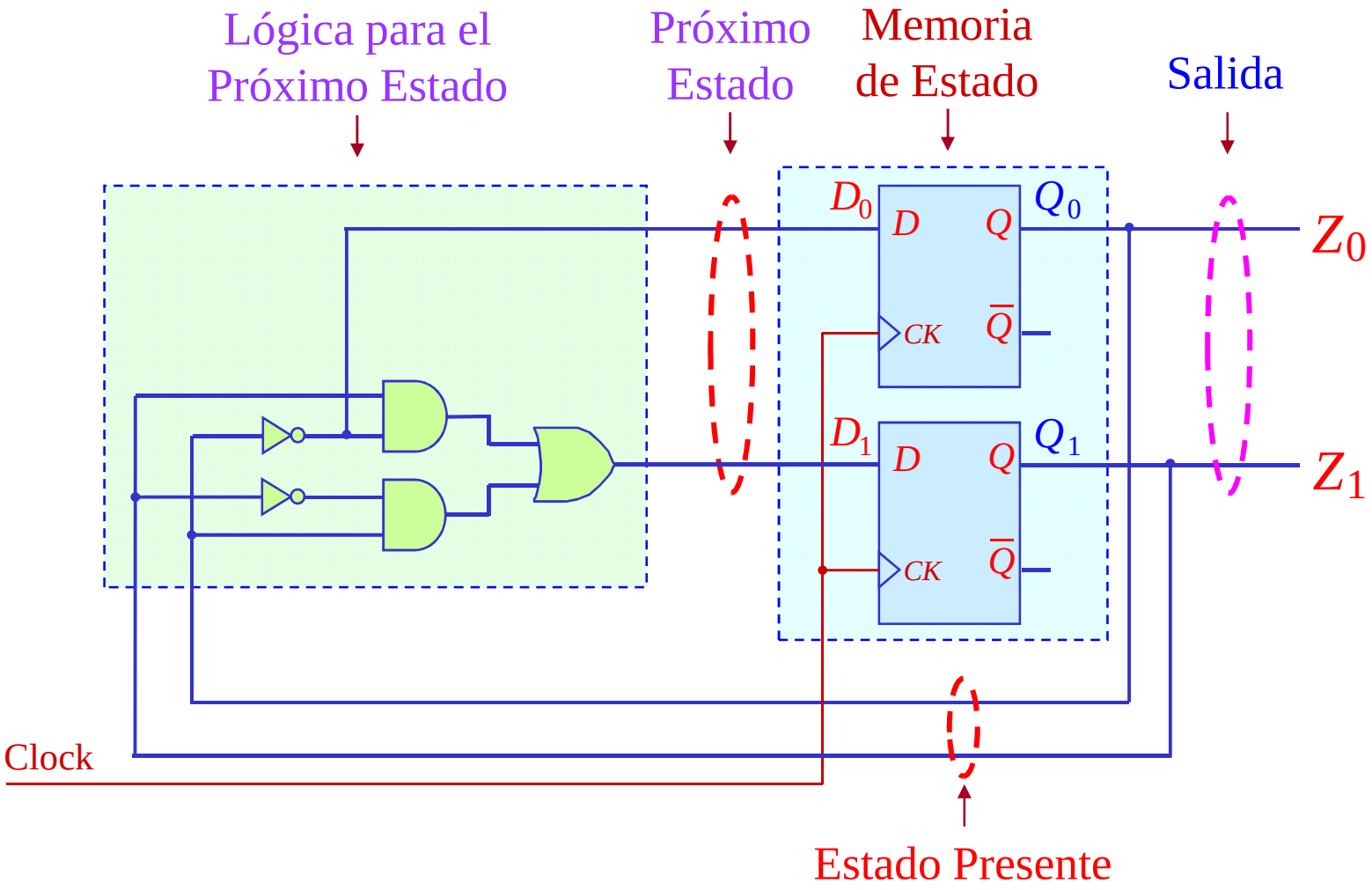
Ejemplo 1: Diseño de un Contador Binario

- Implementación: modelo de circuito secuencial



Ejemplo 1: Diseño de un Contador Binario

- Implementación: modelo FSM



creatures

Binary counters design

Ejemplo 2: Contador binario Ascendente-descendente

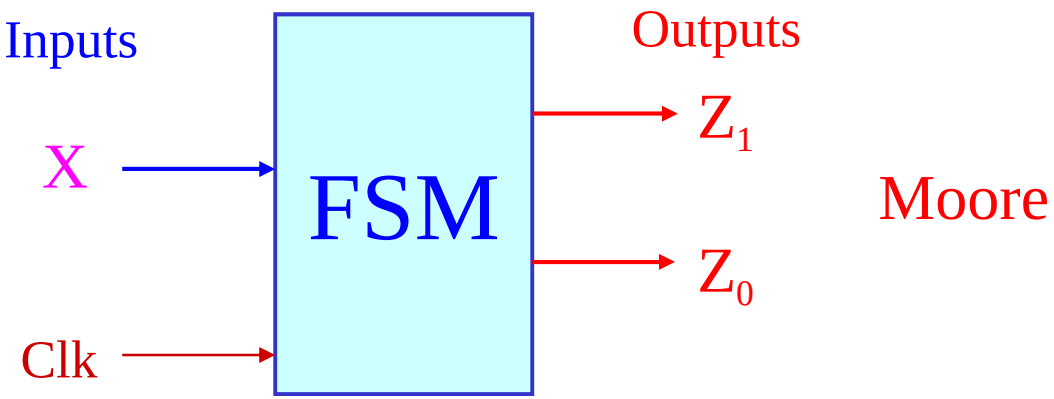
- Contador binario síncrono ascendente-descendente de 2 bits o módulo 4
  - Cuenta cuatro eventos, es decir: 0,1,2,3 ó 1,2,3,4
  - Requiere dos salidas: Z_0 y Z_1
  - La lectura en código binario es: 00, 01, 10, 11

    11, 10, 01, 00
- Representación del circuito secuencial o FSM

Ejemplo 2: Contador binario Ascendente-descendente

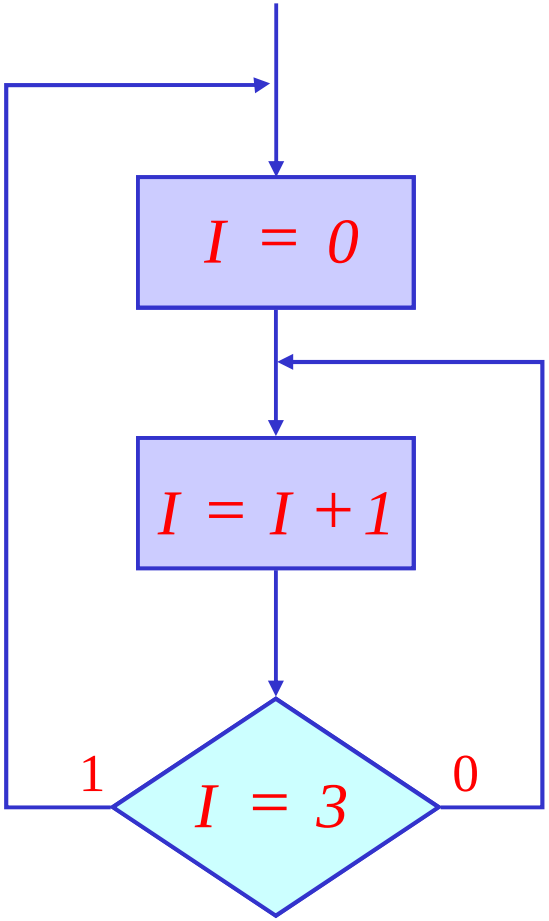
- Analizar las *entradas/salidas* del contador

X = 1 cuenta ascendente: 00, 01, 10, 11
X = 0 cuenta descendente: 11, 10, 01, 00



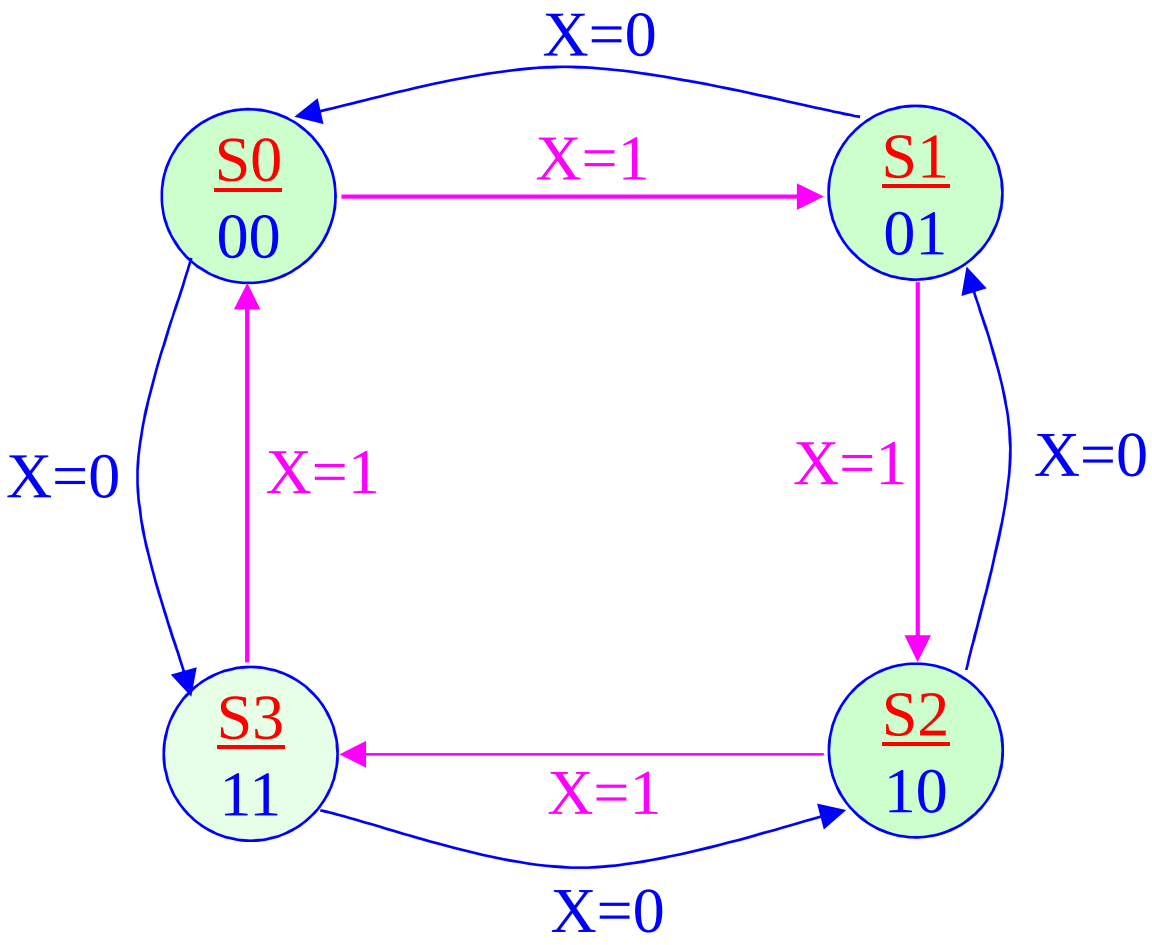
Ejemplo 2: Contador binario Ascendente-descendente

- Diagrama de flujo:



Ejemplo 2: Contador binario Ascendente-descendente

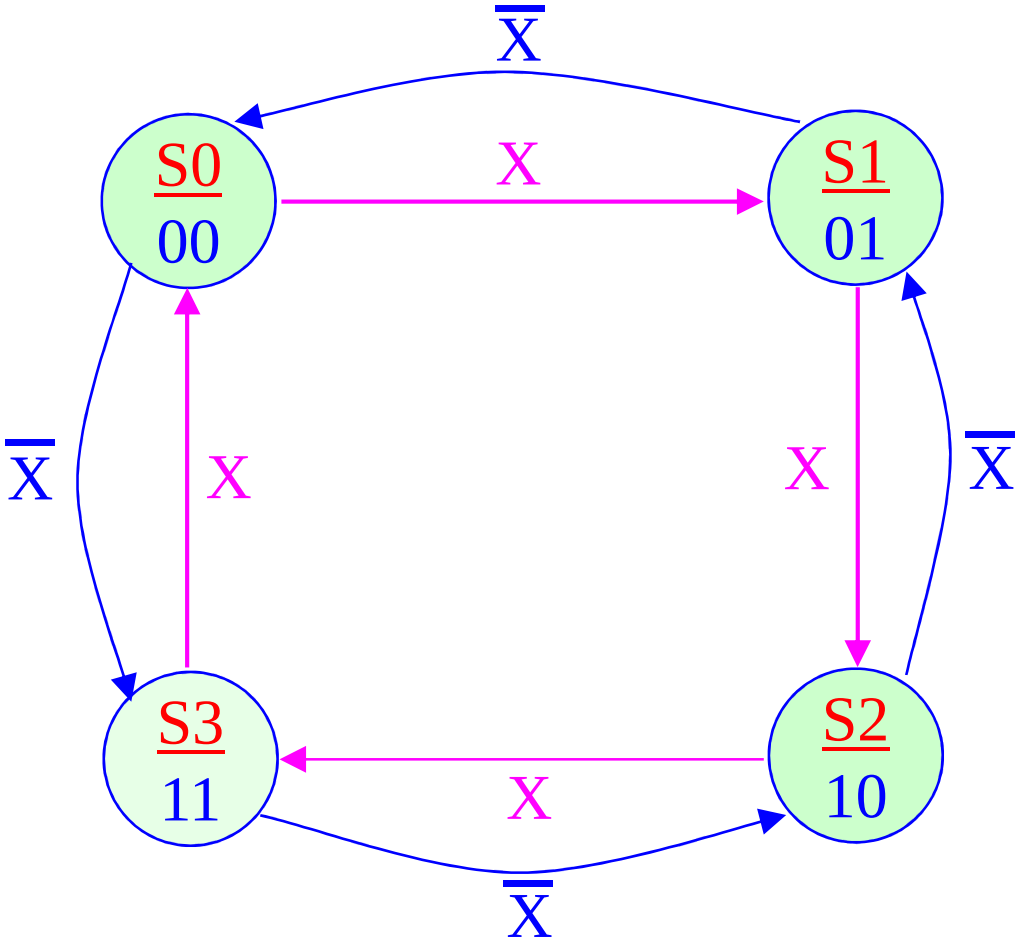
- Diagrama de estados: *FSM tipo Moore*



X	Count
0	B↓
1	B↑

Ejemplo 2: Contador binario Ascendente-descendente

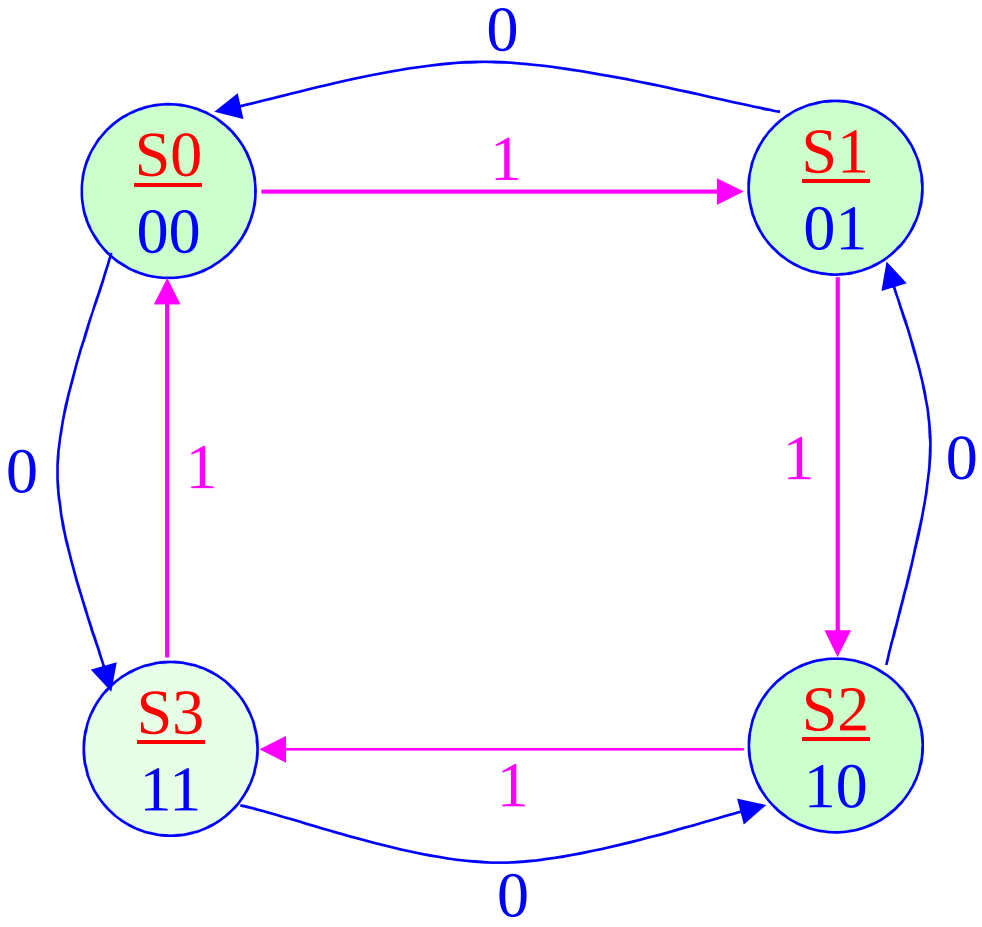
- Diagrama de estados: *FSM tipo Moore*



X	Count
0	B↓
1	B↑

Ejemplo 2: Contador binario Ascendente-descendente

- Diagrama de estados: *FSM tipo Moore*



X	Count
0	B↓
1	B↑

Ejemplo 2: Contador binario Ascendente-descendente

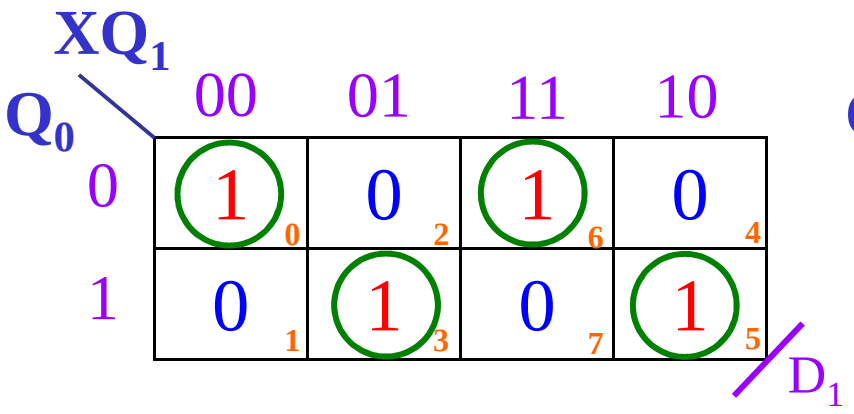
- Deducir la lógica del PE (lógica de excitación)

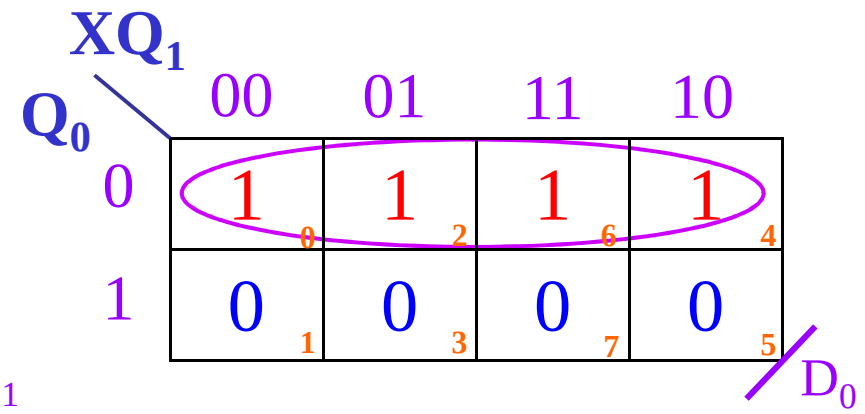
$S_0 = 00$
$S_1 = 01$
$S_2 = 10$
$S_3 = 11$

IN	EP	PE	OUT
X	Q_1Q_0	$Q_1^+Q_0^+$	Z_1Z_0
0	00	11	00
0	01	00	01
0	10	01	10
0	11	10	11
1	00	01	00
1	01	10	01
1	10	11	10
1	11	00	11

D_0
D_1

Ejemplo 2: Contador binario Ascendente-descendente





$$\mathbf{D_1} = \overline{X}\,\overline{Q_1}\,\overline{Q_0} + \overline{X}Q_1Q_0 + X\overline{Q_1}Q_0 + XQ_1\overline{Q_0}$$

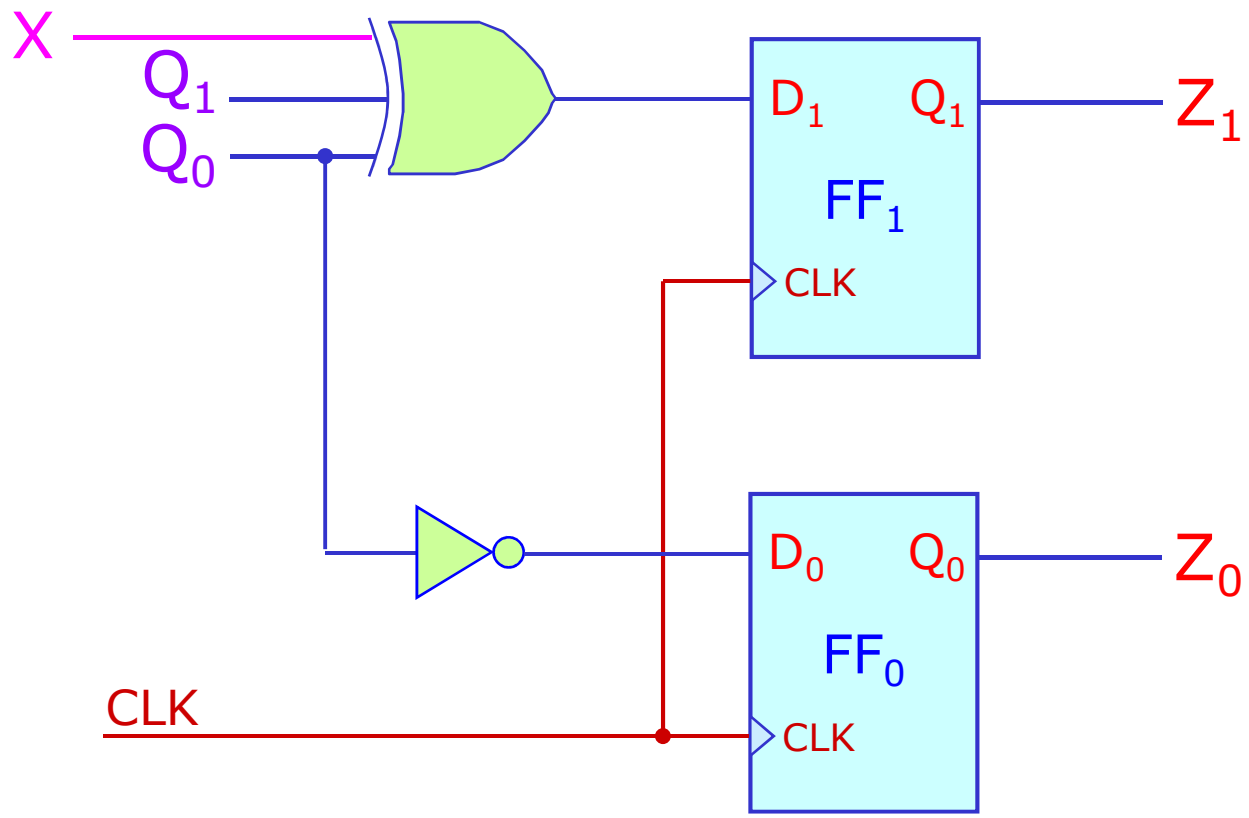
$$\mathbf{D_1} = X \oplus Q_1 \oplus Q_0$$

$$\mathbf{D_0} = \overline{Q_0}$$

$$\mathbf{Z_1} = Q_1\,, \quad \mathbf{Z_0} = Q_0$$

Ejemplo 2: Contador binario Ascendente-descendente

- Implementación: diagrama lógico general



creatures

Binary counters design

Ejemplo 3: Contador binario-gray:Ascendente-descendente

- Contador binario-gray síncrono ascendente-descendente de 2 bits o módulo 4
  - Cuenta cuatro eventos, es decir: 0,1,2,3 ó 1,2,3,4
  - Requiere dos salidas: Z_0 y Z_1
  - La lectura en código binario es: 00, 01, 10, 11

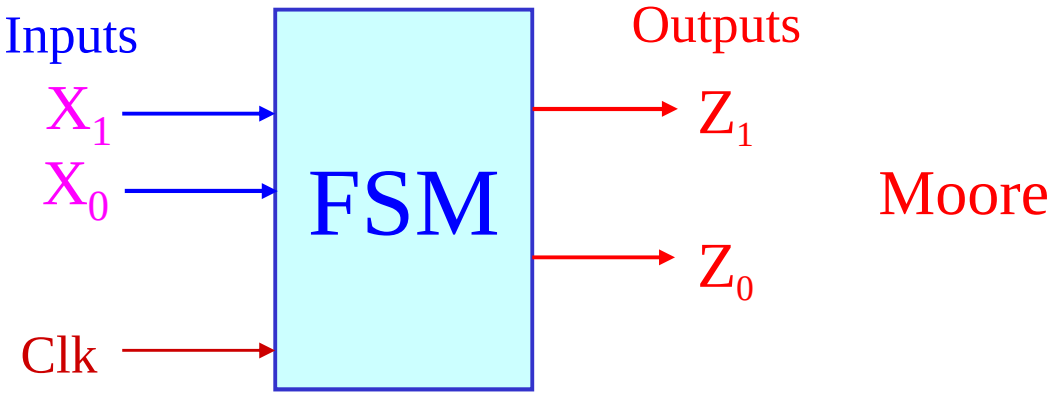
    11, 10, 01, 00
  - La lectura en código gray es: 00, 01, 11, 10

    10, 11, 01, 00
- Representación del circuito secuencial o FSM

Ejemplo 3: Contador binario-gray:Ascendente-descendente

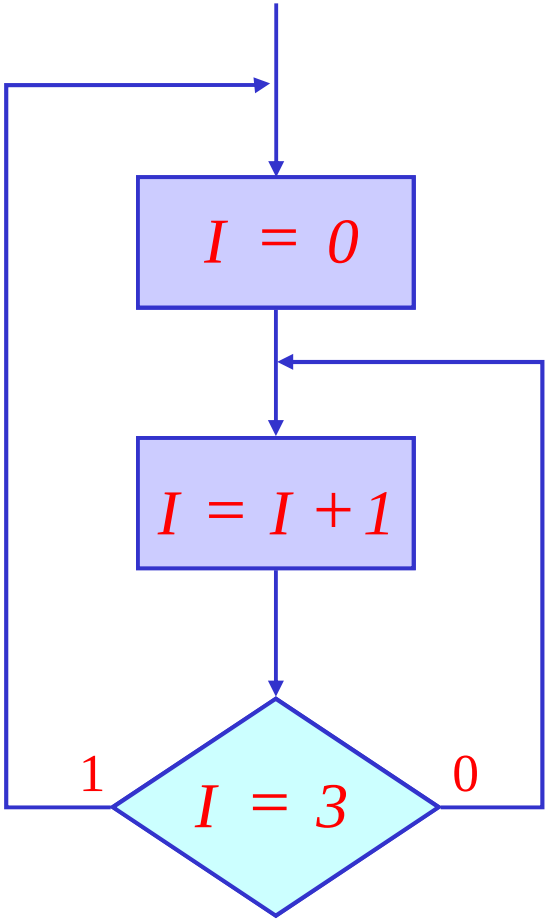
- Analizar las *entradas/salidas* del contador

$X_0 = 1$ binario cuenta ascendente: 00, 01, 10, 11
$X_0 = 0$ binario cuenta descendente: 11, 10, 01, 00
$X_1 = 1$ gray cuenta ascendente: 00, 01, 11, 10
$X_1 = 0$ gray cuenta descendente: 10, 11, 01, 00



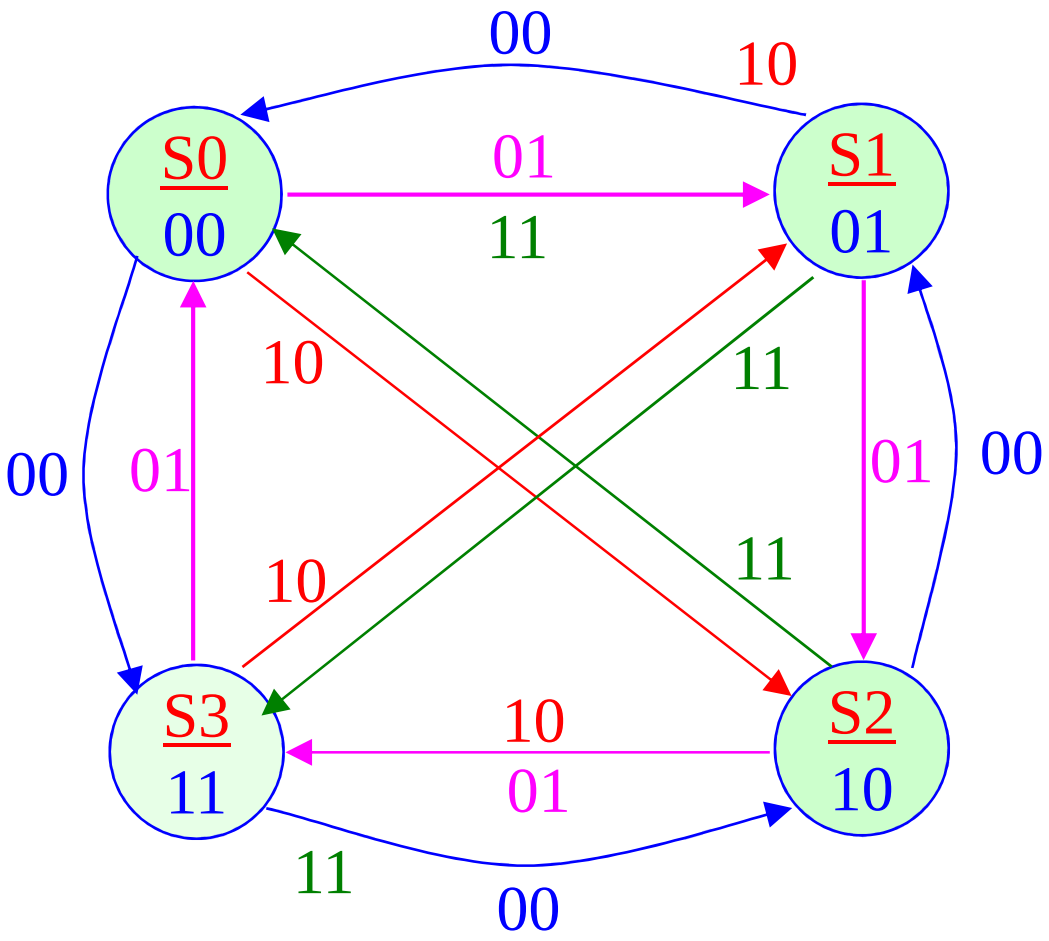
Ejemplo 3: Contador binario-gray:Ascendente-descendente

- Diagrama de flujo:



Ejemplo 3: Contador binario-gray:Ascendente-descendente

- Diagrama de estados: *FSM tipo Moore*



11, 10, 01, 00
00, 01, 10, 11
10, 11, 01, 00
00, 01, 11, 10

X_1	X_0	Count
0	0	B↓
0	1	B↑
1	0	G↓
1	1	G↑

Ejemplo 3: Contador binario-gray:Ascendente-descendente

- Deducir la lógica del PE (lógica de excitación)

11, 10, 01, 00
00, 01, 10, 11
10, 11, 01, 00
00, 01, 11, 10

X_1	X_0	Count
0	0	B↓
0	1	B↑
1	0	G↓
1	1	G↑

EP Q_1Q_0	PE: $Q_1^+Q_0^+$				OUT Z_1Z_0
	00	01	10	11	
00	11	01	10	01	00
01	00	10	00	11	01
10	01	11	11	00	10
11	10	00	01	10	11

D_0
D_1

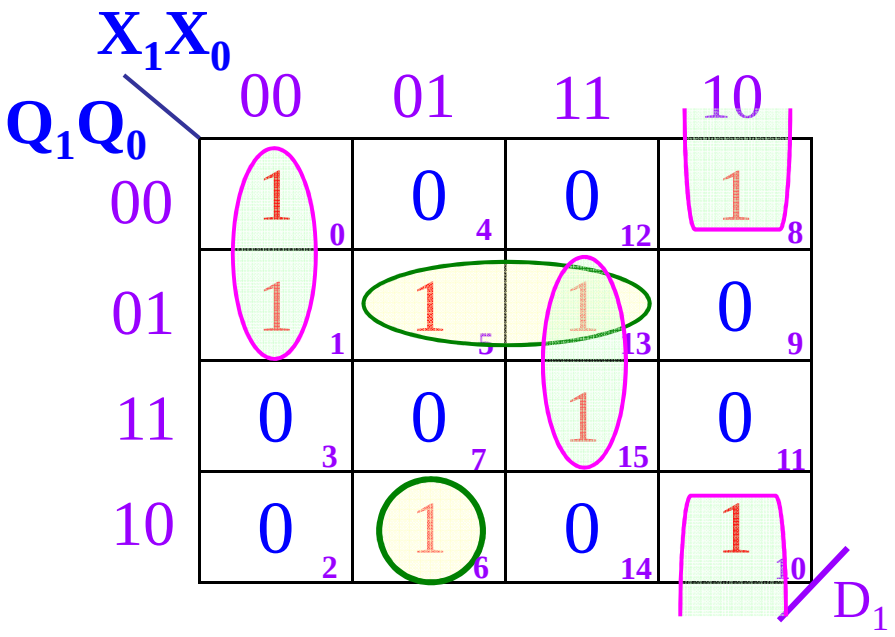
$S_0 = 00$
$S_1 = 01$
$S_2 = 10$
$S_3 = 11$

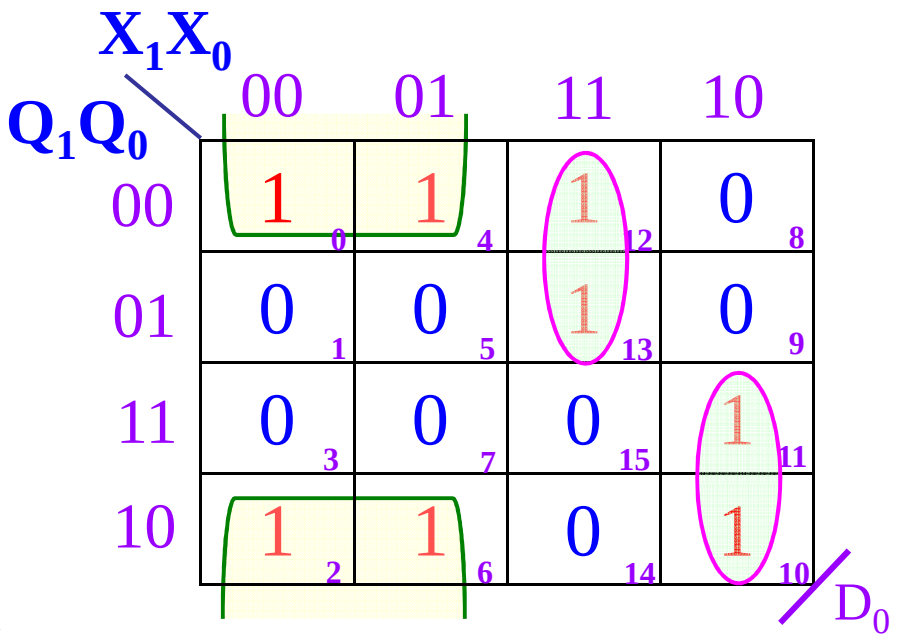
Ejemplo 3: Contador binario-gray:Ascendente-descendente

11, 10, 01, 00
00, 01, 10, 11
10, 11, 01, 00
00, 01, 11, 10

	IN		EP		PE		Z	
	X_1	X_0	Q_1^n	Q_0^n	Q_1^{n+1}	Q_0^{n+1}	Z_1	Z_0
M0	0	0	0	0	1	1	0	0
	0	0	0	1	1	0	0	1
	0	0	1	0	0	1	1	0
	0	0	1	1	0	0	1	1
	0	1	0	0	0	1	0	0
	0	1	0	1	1	0	0	1
	0	1	1	0	1	1	1	0
	1	1	1	1	0	0	1	1
	1	0	0	0	1	0	0	0
	1	0	0	1	0	0	0	1
	1	0	1	0	1	1	1	0
	1	0	1	1	0	1	1	1
	1	1	0	0	0	1	0	0
	1	1	0	1	1	1	0	1
	1	0	1	0	0	0	1	0
M15	1	1	1	1	1	0	1	1

Ejemplo 3: Contador binario-gray:Ascendente-descendente



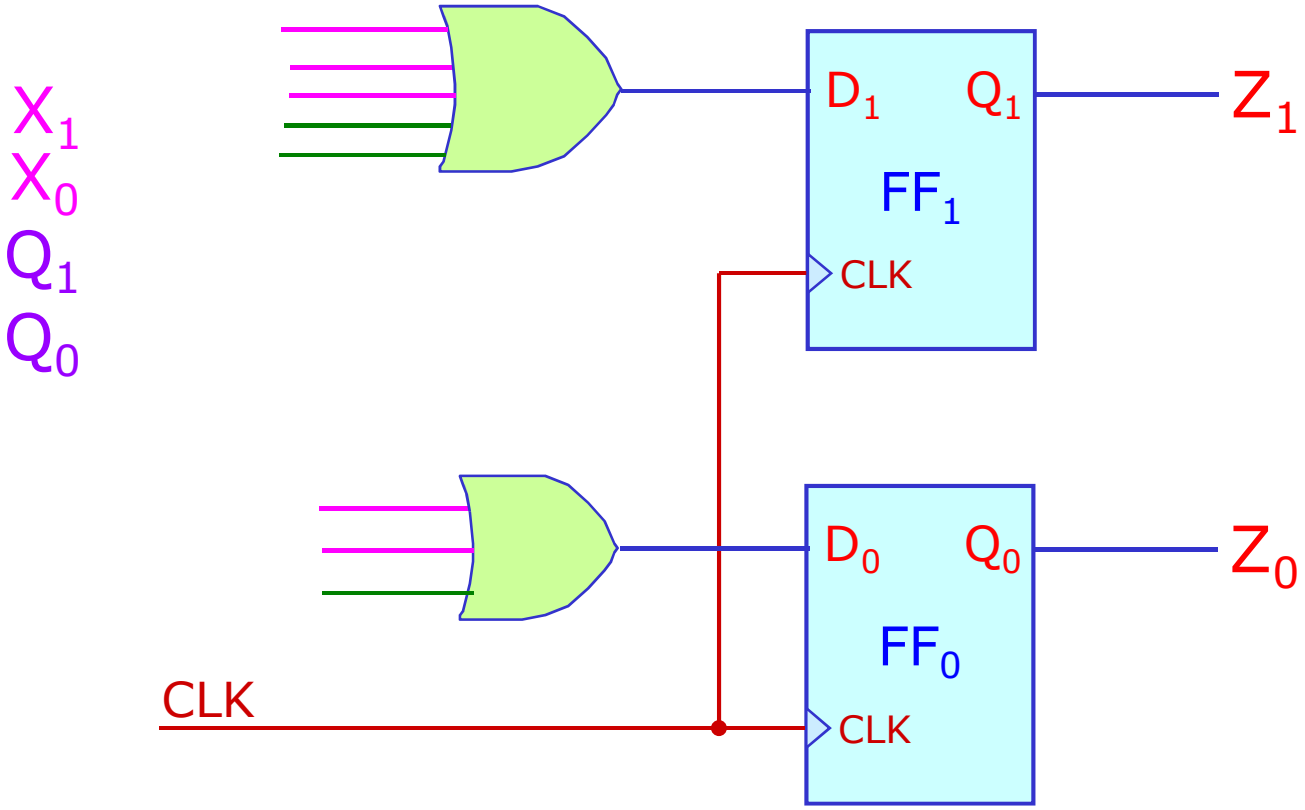


$$D_1 = X_1\overline{X}_0Q_1\overline{Q}_0 + X_0Q_1\overline{Q}_0 + \overline{X}_1\overline{X}_0\overline{Q}_1 + X_1X_0Q_0 + X_1\overline{X}_0\overline{Q}_0$$

$$D_0 = \overline{X}_1\overline{Q}_0 + X_1\overline{X}_0Q_1 + X_1\overline{X}_0Q_1$$

$$Z_1 = Q_1\,, \quad Z_0 = Q_0$$

Ejemplo 3: Contador binario-gray:Ascendente-descendente



Ejemplo 3: Contador binario-gray:Ascendente-descendente

- *Simulation using Quartus II*