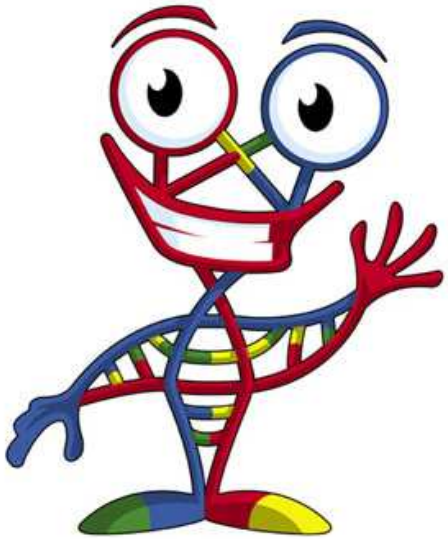
Digital System Design Course



FSM Design

creatures

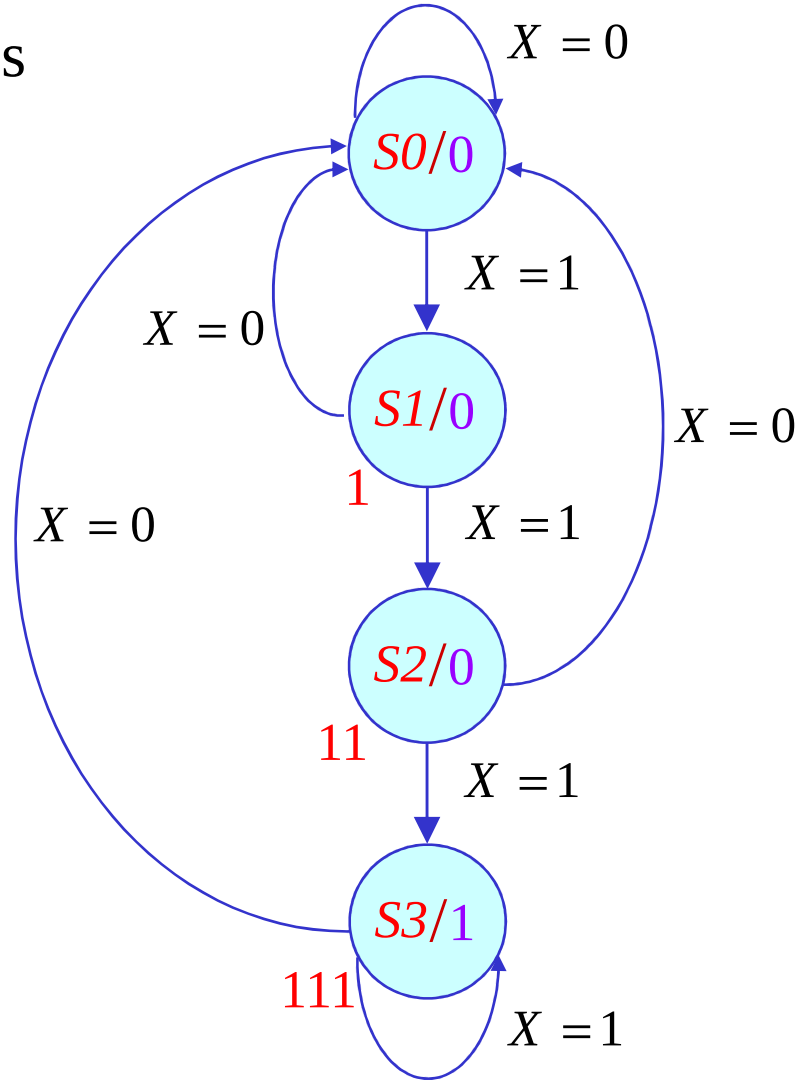
Sequencing detectors design

Ejemplo 1: Diseño de un Detector de Secuencia

- Diseñar una maquina de estados que permita *detectar una secuencia dada*
- Consideraciones:
  - Detectar la secuencia X = 111
  - La secuencia se puede superponer X= 0111101
  - Implementar el circuito usando flipflop JK
  - Diseñar el circuito usando una FSM tipo Moore

Ejemplo 1: Diseño de un Detector de Secuencia

- Diagrama de estados



Ejemplo 1: Diseño de un Detector de Secuencia

- Tabla de estados

X	$Q^n{}_1\ Q^n{}_0$	$Q^{n+1}{}_1\ Q^{n+1}{}_0$	Z
IN	EP	PE	OUT
0	S_0	S_0	0
0	S_1	S_0	0
0	S_2	S_0	0
0	S_3	S_0	1
1	S_0	S_1	0
1	S_1	S_2	0
1	S_2	S_3	0
1	S_3	S_3	1

Ejemplo 1: Diseño de un Detector de Secuencia

- Tabla de verdad: flipflop JK

J	**K**	**CK**	Q	$\overline{Q}$
X	X	0	Q_0	$\overline{Q}_0$
X	X	1	Q_0	$\overline{Q}_0$
0	0	↑	Q_0	$\overline{Q}_0$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	$\overline{Q}_0$	Q_0

Ejemplo 1: Diseño de un Detector de Secuencia

- Tabla de estados: flipflop JK

J	K	CK	EP	PE	
X	X	0	Q_0	Q_0	
X	X	1	Q_0	Q_0	
0	0	↑	0	0	1
0	0	↑	1	1	4
0	1	↑	0	0	1
0	1	↑	1	0	3
1	0	↑	0	1	2
1	0	↑	1	1	4
1	1	↑	0	1	2
1	1	↑	1	0	3

- Tabla de transición de estados

	EP	PE	J	K
1	0	0	0	X
2	0	1	1	X
3	1	0	X	1
4	1	1	X	0

Ejemplo 1: Diseño de un Detector de Secuencia

	Entrada	EP		PE		Flipflops JK				Salida
	X	$Q^n{}_1$	$Q^n{}_0$	$Q^{n+1}{}_1$	$Q^{n+1}{}_0$	J_1	K_1	J_0	K_0	Z
M0	0	0	0	0	0	0	X	0	X	0
	0	0	1	0	0	0	X	X	1	0
	0	1	0	0	0	X	1	0	X	0
	0	1	1	0	0	X	1	X	1	1
	1	0	0	0	1	0	X	1	X	0
	1	0	1	1	0	1	X	X	1	0
	1	1	0	1	1	X	0	1	X	0
M7	1	1	1	1	1	X	0	X	0	1

J_0K_0

J_1K_1

Ejemplo 1: Diseño de un Detector de Secuencia

- Mapa de Karnaugh

J_1 :

X \ Q_1Q_0	00	01	11	10
0	**0**	**0**	**X**	**X**
1	**0**	**1**	**X**	**X**

J_1

$$J_1 = Q_0 X$$

K_1 :

X \ Q_1Q_0	00	01	11	10
0	**X**	**X**	**1**	**1**
1	**X**	**X**	**0**	**0**

K_1

$$K_1 = \overline{X}$$

Ejemplo 1: Diseño de un Detector de Secuencia

- Mapa de Karnaugh

J_0 :

X \ Q_1Q_0	00	01	11	10
0	**0**	**X**	**X**	**0**
1	**1**	**X**	**X**	**1**

J_0

$$J_0 = X$$

K_0 :

X \ Q_1Q_0	00	01	11	10
0	**X**	**1**	**1**	**X**
1	**X**	**1**	**0**	**X**

K_0

$$K_0 = \overline{X} + \overline{Q_1}$$

Ejemplo 1: Diseño de un Detector de Secuencia

- Mapa de Karnaugh

$Z:$

$X \backslash Q_1Q_0$	00	01	11	10
0	**0**	**0**	**1**	**0**
1	**0**	**0**	**1**	**0**

Z

$$Z = Q_1Q_0$$

Ejemplo 1: Diseño de un Detector de Secuencia

- Implementación: diagrama lógico general

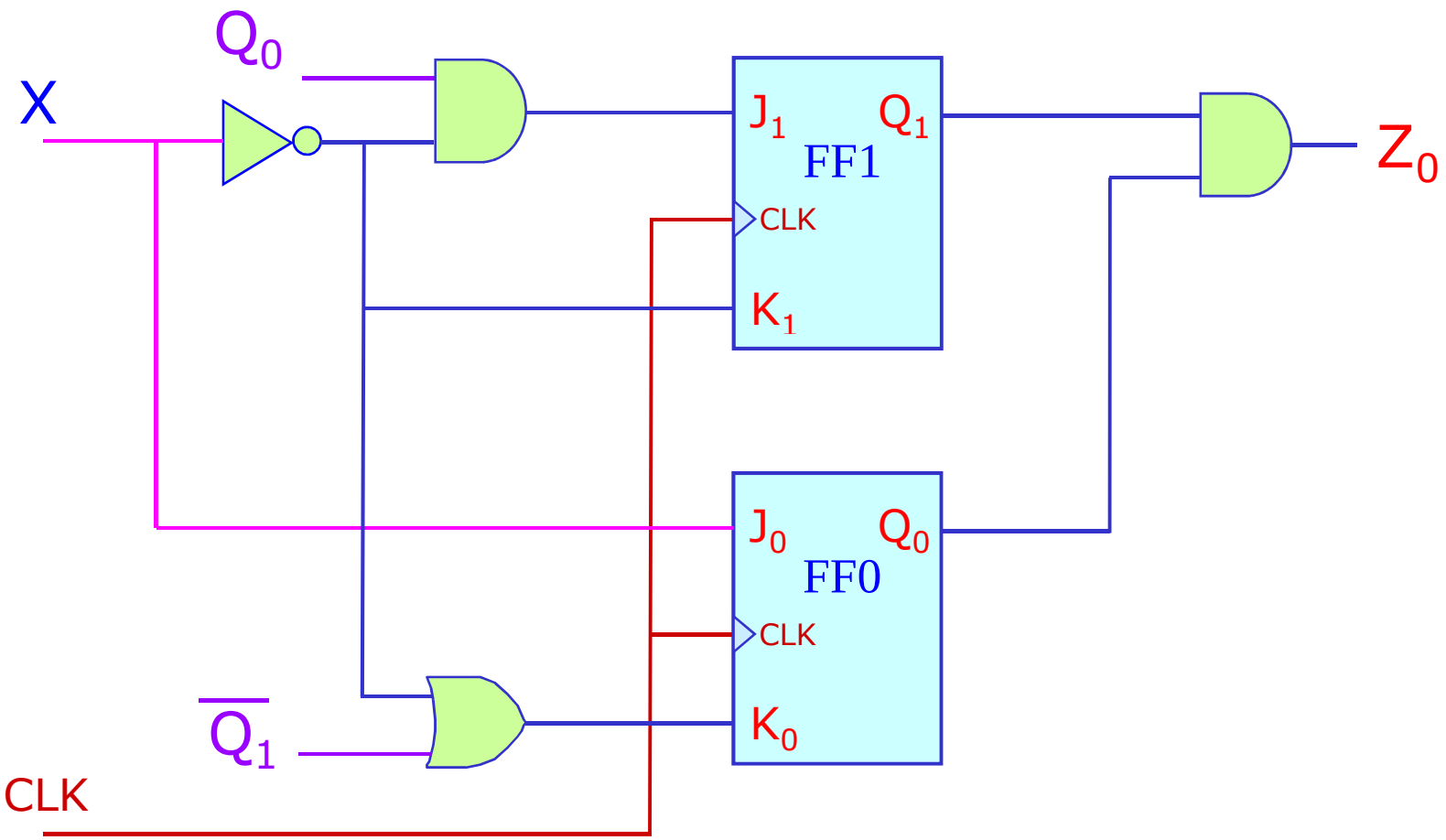


Tabla de Transición para el Flipflop D

- Tabla de verdad: flipflop D

D	**CK**	Q	$\overline{Q}$
X	0	Q_0	$\overline{Q}_0$
X	1	Q_0	$\overline{Q}_0$
0	↑	0	1
1	↑	1	0

- Tabla de estados

D	CK	EP	PE
X	0	Q_0	Q_0
X	1	Q_0	Q_0
0	↑	0	0
0	↑	1	0
1	↑	0	1
1	↑	1	1

Tabla de Transición para el Flipflop D

- Tabla de estados

D	CK	EP	PE	
X	0	Q_0	Q_0	
X	1	Q_0	Q_0	
0	↑	0	0	1
0	↑	1	0	2
1	↑	0	1	3
1	↑	1	1	4

- Tabla de estados

	EP	D	
		0	1
1, 3	$Q = 0$	0	1
2, 4	$Q = 1$	0	1

PE

Tabla de Transición para el Flipflop D

- Tabla de estados

D	CK	EP	PE	
X	0	Q_0	Q_0	
X	1	Q_0	Q_0	
0	↑	0	0	1
0	↑	1	0	2
1	↑	0	1	3
1	↑	1	1	4

- Tabla de transición de estados

	EP	PE	D
1	0	0	0
3	0	1	1
2	1	0	0
4	1	1	1

Tabla de Transición para el Flipflop D

- Tabla de estados

EP	D	
	0	1
$Q = 0$	0	1
$Q = 1$	0	1

PE

- Ecuación de transición

$$Q^{n+1} = D$$

Tabla de Transición para el Flipflop JK

❖ Tabla de transición de estados

	EP	PE	J	K
1	0	0	0	X
2	0	1	1	X
3	1	0	X	1
4	1	1	X	0

	EP	J K			
		00	01	11	10
1, 2	Q = 0	0	0	1	1
3, 4	Q = 1	1	0	0	1

PE

Tabla de Transición para el Flipflop JK

- Tabla de transición de estados
- Ecuación de transición

	EP	J K			
		00	01	11	10
1, 2	Q = 0	0	0	1	1
3, 4	Q = 1	1	0	0	1

PE

$$Q^{n+1} = \overline{Q}J + Q\overline{K}$$

creatures

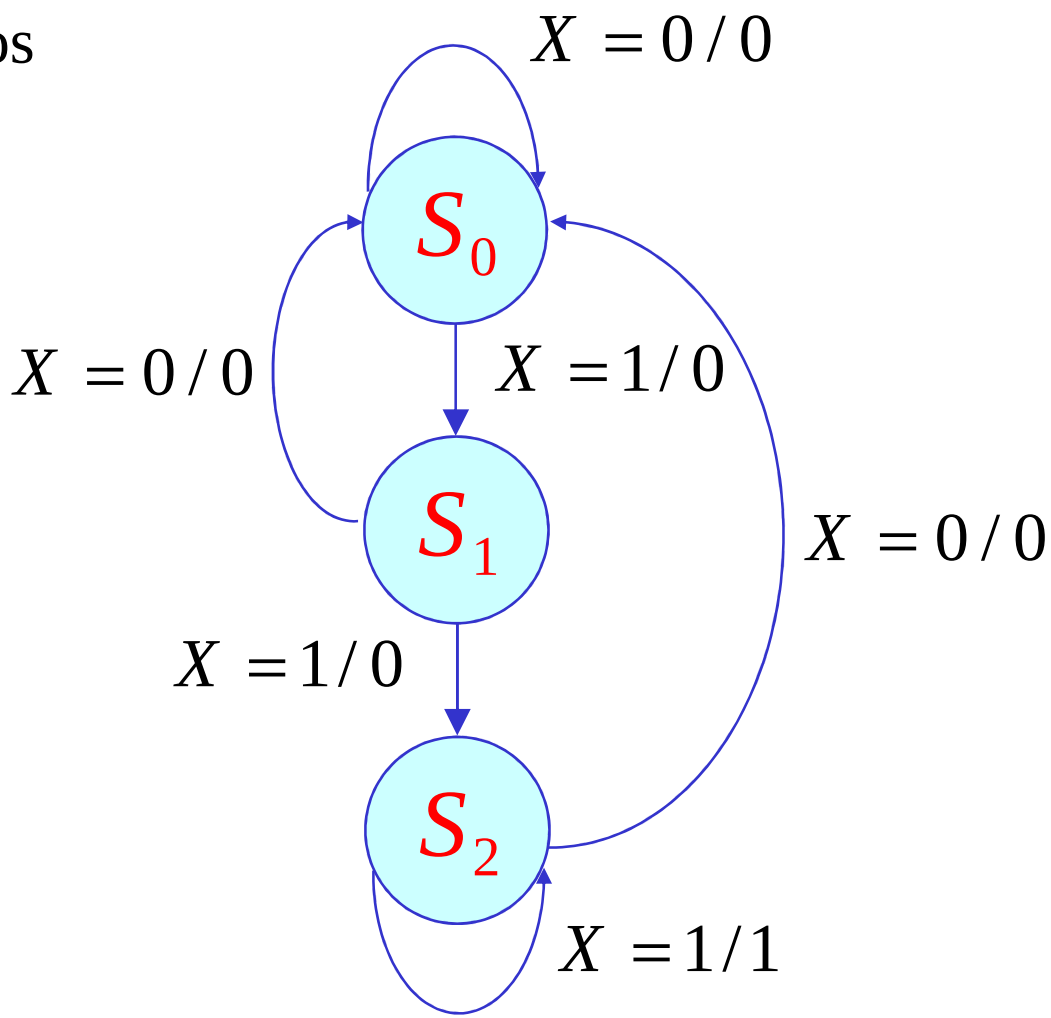
Sequencing detectors design

Ejemplo 2: Diseño de un Detector de Secuencia

- Diseñar una maquina de estados que permita *detectar una secuencia dada*
- Consideraciones:
  - Detectar la secuencia X = 111
  - La secuencia se puede superponer X= 0111101
  - Implementar el circuito usando flipflop JK
  - Diseñar el circuito usando una FSM tipo Mealy

Ejemplo 2: Diseño de un Detector de Secuencia

- Diagrama de estados



Ejemplo 2: Diseño de un Detector de Secuencia

- Tabla de estados

X	EP	PE
0	S_0	S_0 / 0
0	S_1	S_0 / 0
0	S_2	S_0 / 0
1	S_0	S_1/ 0
1	S_1	S_2 / 0
1	S_2	S_2 / 1

Ejemplo 2: Diseño de un Detector de Secuencia

Entrada	EP		PE / Z		Flipflops JK			
X	$Q^n{}_1$	$Q^n{}_0$	$Q^{n+1}{}_1$	$Q^{n+1}{}_0$	J_1	K_1	J_0	K_0
0	0	0	0	0/0	0	X	0	X
0	0	1	0	0/0	0	X	X	1
0	1	0	0	0/0	X	1	0	X
1	0	0	0	1/0	0	X	1	X
1	0	1	1	0/0	1	X	X	1
1	1	0	1	0/1	X	0	0	X

J_0K_0

J_1K_1

Ejemplo 2: Diseño de un Detector de Secuencia

- Mapa de Karnaugh

J_1 :

X \ Q_1Q_0	00	01	11	10
0	**0**	**0**	**X**	**X**
1	**0**	**1**	**X**	**X**

J_1

$$J_1 = Q_0 X$$

K_1 :

X \ Q_1Q_0	00	01	11	10
0	**X**	**X**	**X**	**1**
1	**X**	**X**	**X**	**0**

K_1

$$K_1 = \overline{X}$$

Ejemplo 2: Diseño de un Detector de Secuencia

- Mapa de Karnaugh

J_0 :

X \ Q_1Q_0	00	01	11	10
0	**0**	**0**	**X**	**0**
1	**0**	**0**	**X**	**0**

J_0

$J_0 = \overline{Q_1}X$ $\qquad$ $J_0 = X$

K_0 :

X \ Q_1Q_0	00	01	11	10
0	**X**	**1**	**1**	**X**
1	**X**	**1**	**0**	**X**

K_0

$K_0 = \overline{X} + \overline{Q_1}$ $\qquad$ $K_0 = \overline{X} + \overline{Q_1}$

Ejemplo 2: Diseño de un Detector de Secuencia

- Mapa de Karnaugh

Z :

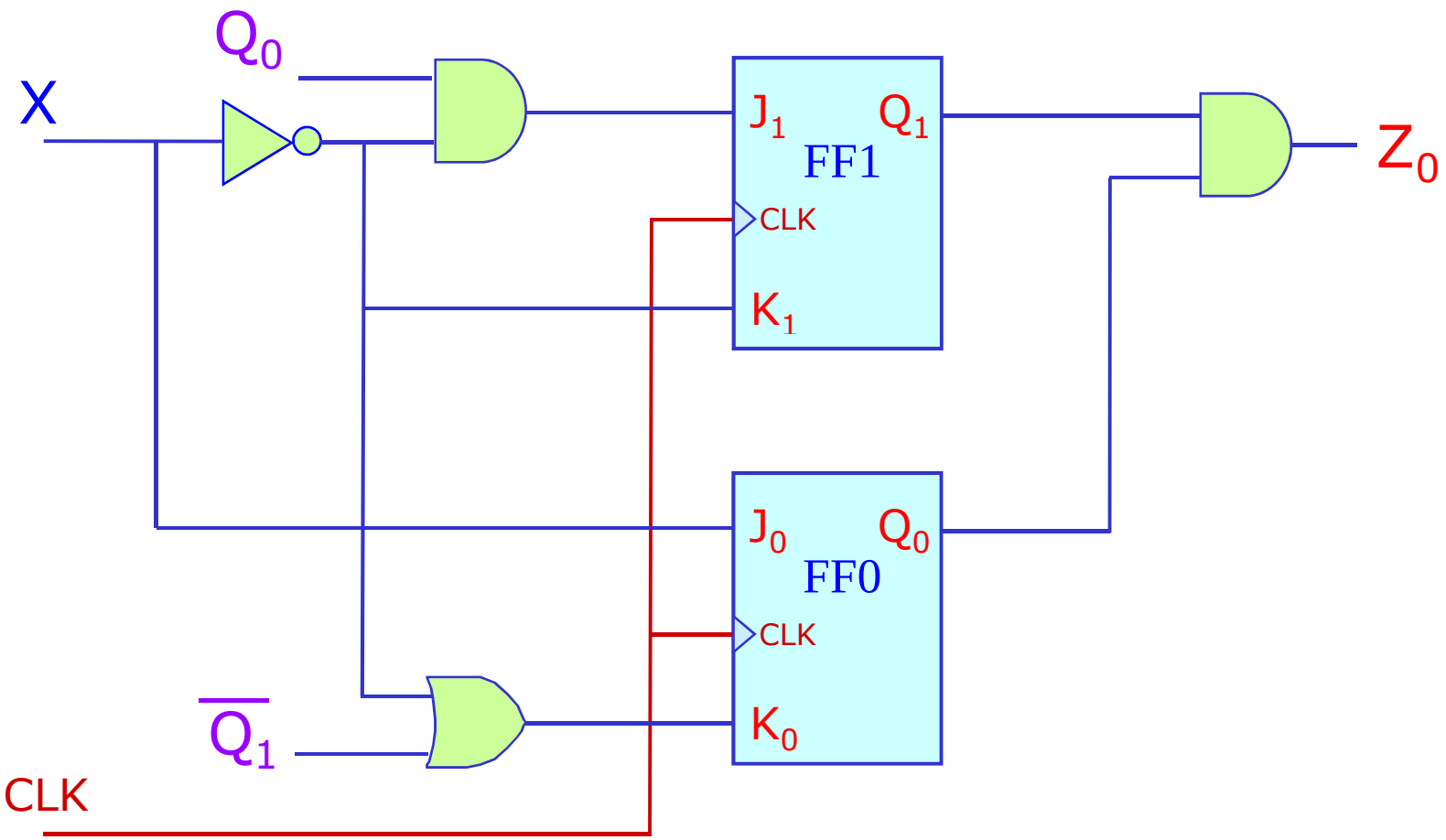
X \ Q_1Q_0	00	01	11	1 0
0	**0**	**0**	**X**	**0**
1	**0**	**0**	**X**	**1**

Z

$$Z = Q_1 Q_0$$

Ejemplo 2: Diseño de un Detector de Secuencia

- Implementación: diagrama lógico general



creatures

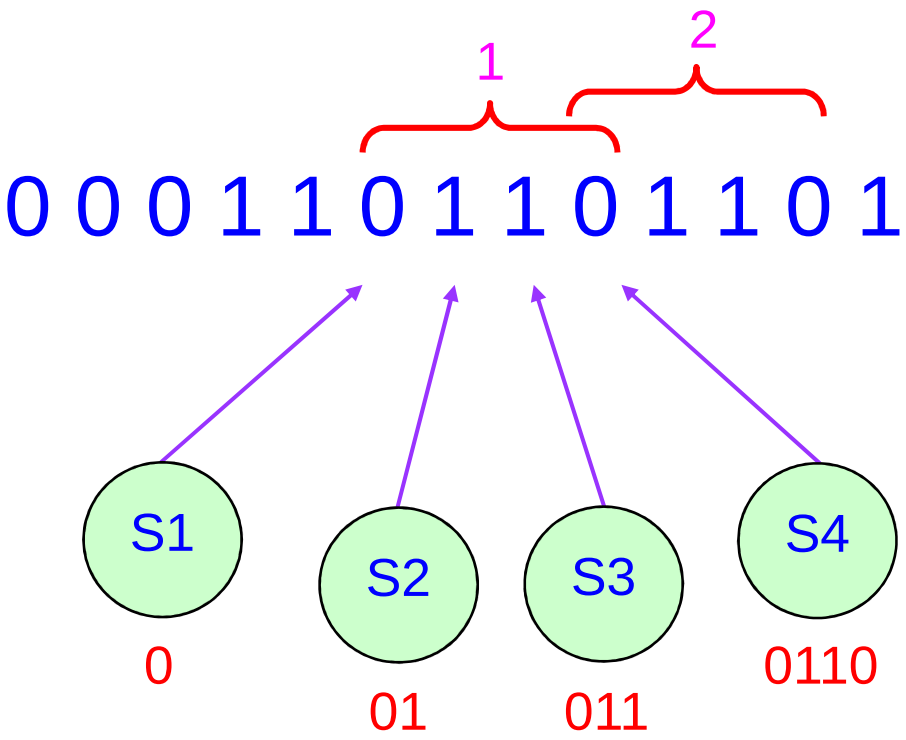
Sequencing detectors design

Ejemplo 3: Diseño de un Detector de Secuencia

- Diseñar una maquina de estados que permita *detectar una secuencia dada*
- Especificaciones del circuito detector de secuencias:
  - Una *entrada X* y una *salida Z*
  - Cuando la máquina de estados detecta la secuencia de entrada *X = 0110*, la salida Z se activa, es decir *Z=1*, en caso contrario *Z=0*
  - Existe superposicion
  - No existe superposicion cuando la secuencia se detecta solo cuando se presenta de manera completa

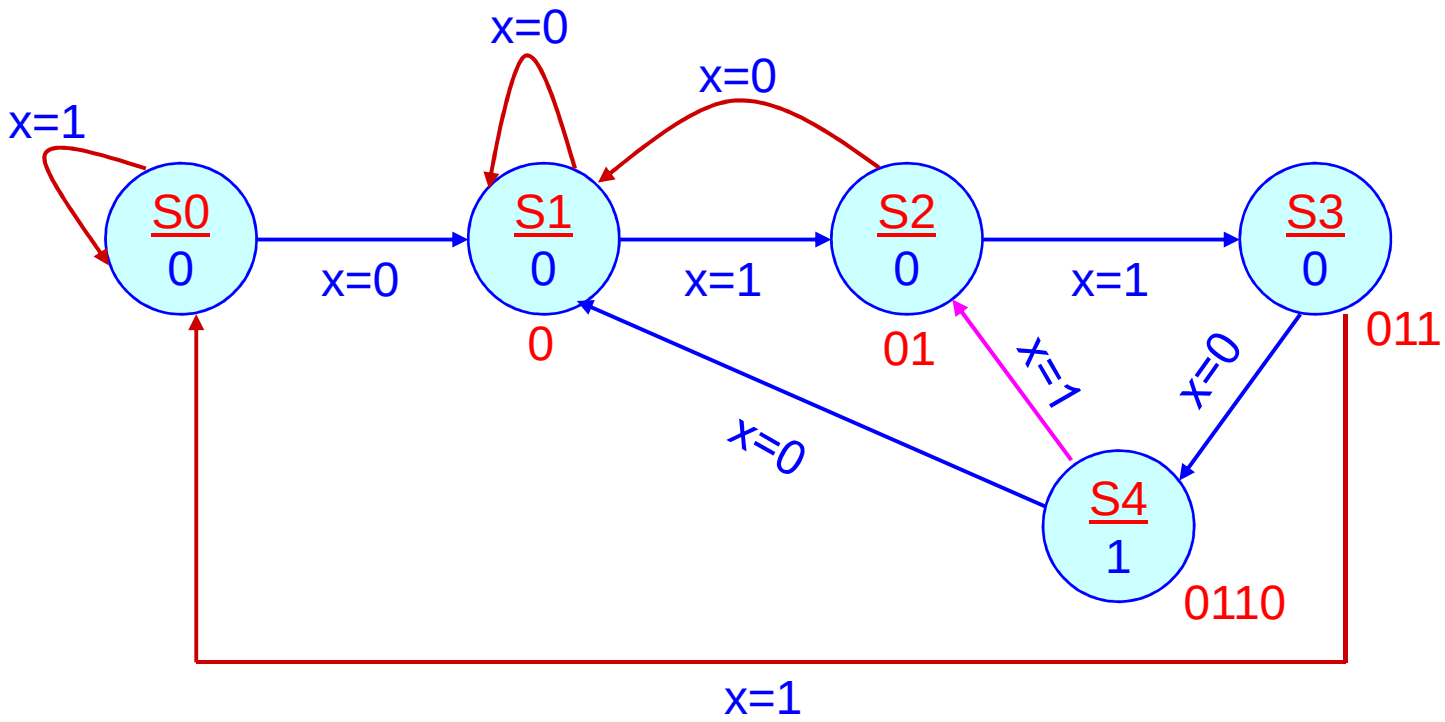
Ejemplo 3: Diseño de un Detector de Secuencia

- Superposicion de secuencias
  - Las secuencias se pueden superponer, es decir, una parte de la secuencia detectada puede ser usada para detectar una nueva



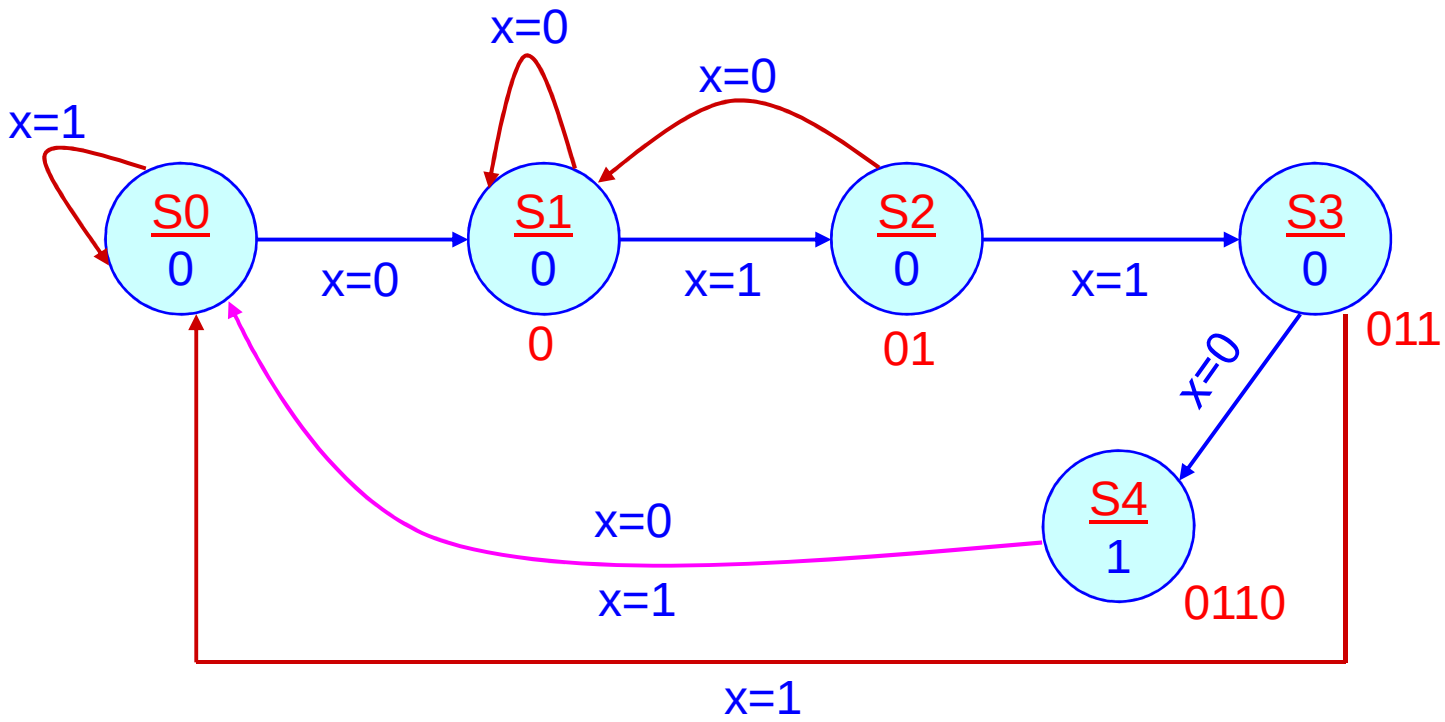
Ejemplo 3: Diseño de un Detector de Secuencia

❖ Diagrama de estados considerando superposición



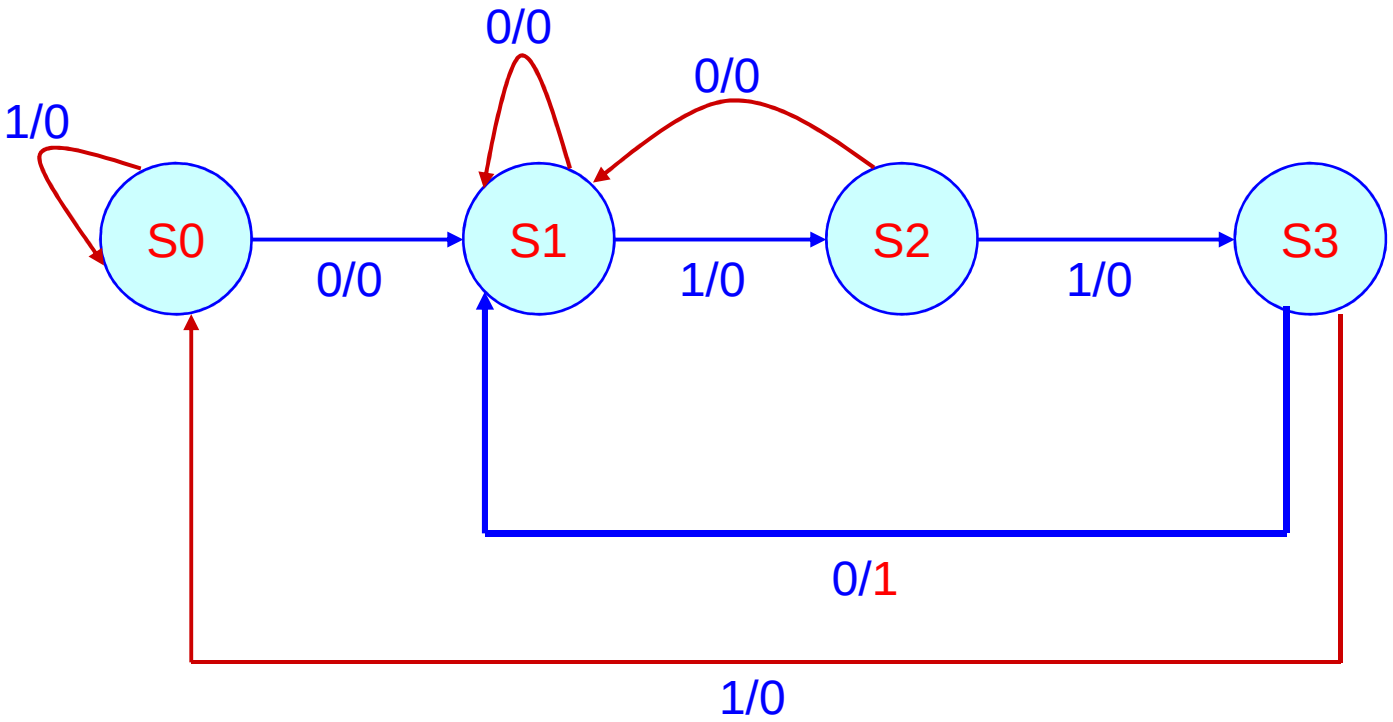
Ejemplo 3: Diseño de un Detector de Secuencia

❖ Diagrama de estados sin considerar superposición



Ejemplo 3: Diseño de un Detector de Secuencia

- Diagrama de estados tipo mealy considerando superposición



Ejemplo 3: Diseño de un Detector de Secuencia

- Deducir la lógica del PE (lógica de excitación)

X	EP	PE	OUT
-	Q_1Q_0	$Q_1^+Q_0^+$	/Z
0	00	01	0
0	01	01	0
0	10	01	0
0	11	01	1
1	00	00	0
1	01	10	0
1	10	11	0
1	11	00	0

D1	D0
-	-
0	1
0	1
0	1
0	1
0	0
1	0
1	1
0	0

Ejemplo 3: Diseño de un Detector de Secuencia

- Deducir la lógica del PE (lógica de excitación)

Q_0^n \ XQ_1^n	00	01	11	10
0	0	0	1	0
1	0	0	0	1

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

Q_0^n \ XQ_1^n	00	01	11	10
0	1	1	1	0
1	1	1	0	0

$$D_0 = \overline{X} + Q_1\,\overline{Q}_0$$

Ejemplo 3: Diseño de un Detector de Secuencia

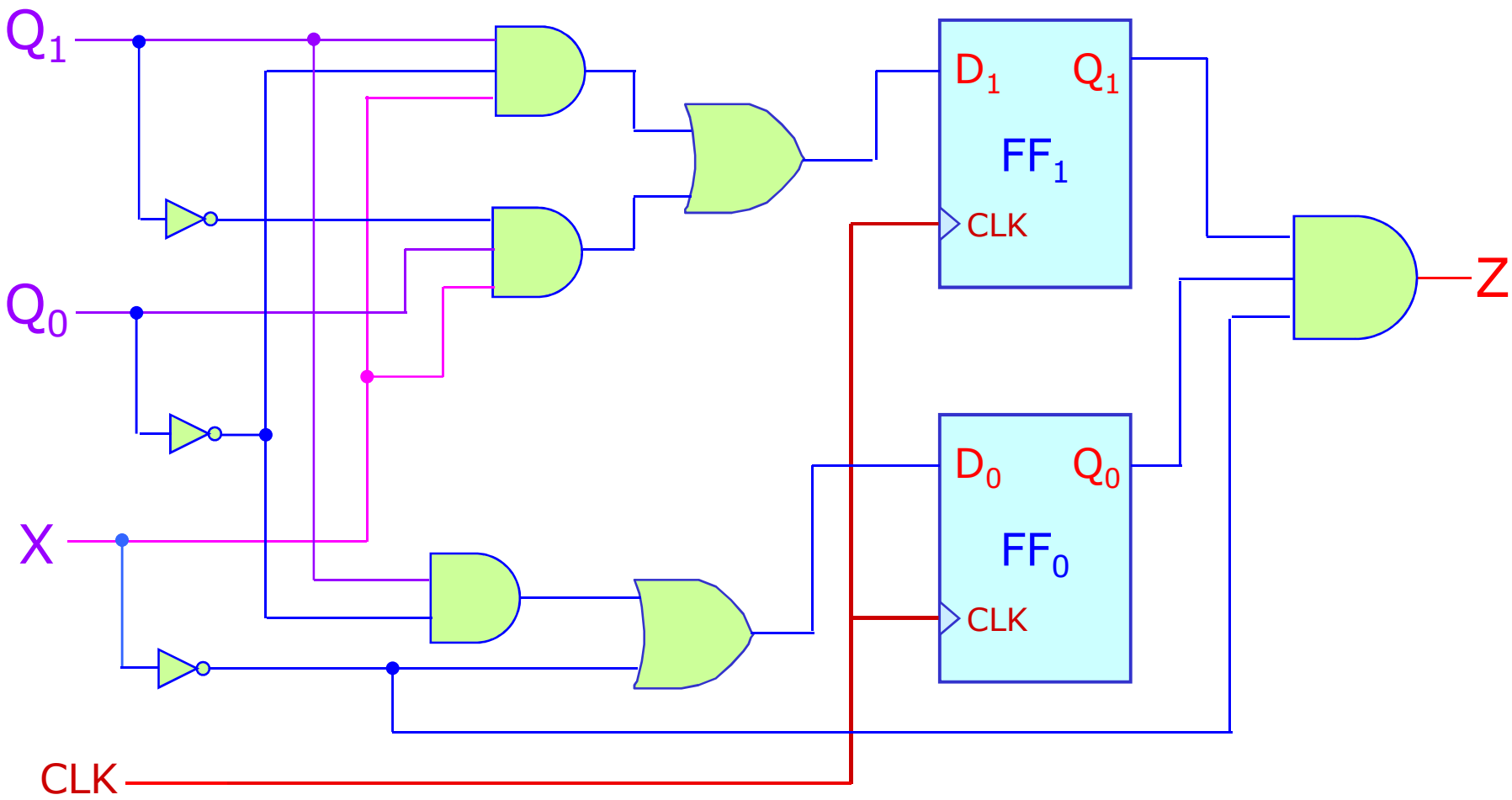
- Deducir la lógica de salida Z

Q_0^n \ XQ_1^n	00	01	11	10
0	0	0	0	0
1	0	1	0	0

$$Z = \overline{X}\, Q_1 Q_0$$

Ejemplo 3: Diseño de un Detector de Secuencia

- Implementación: diagrama lógico general



creatures

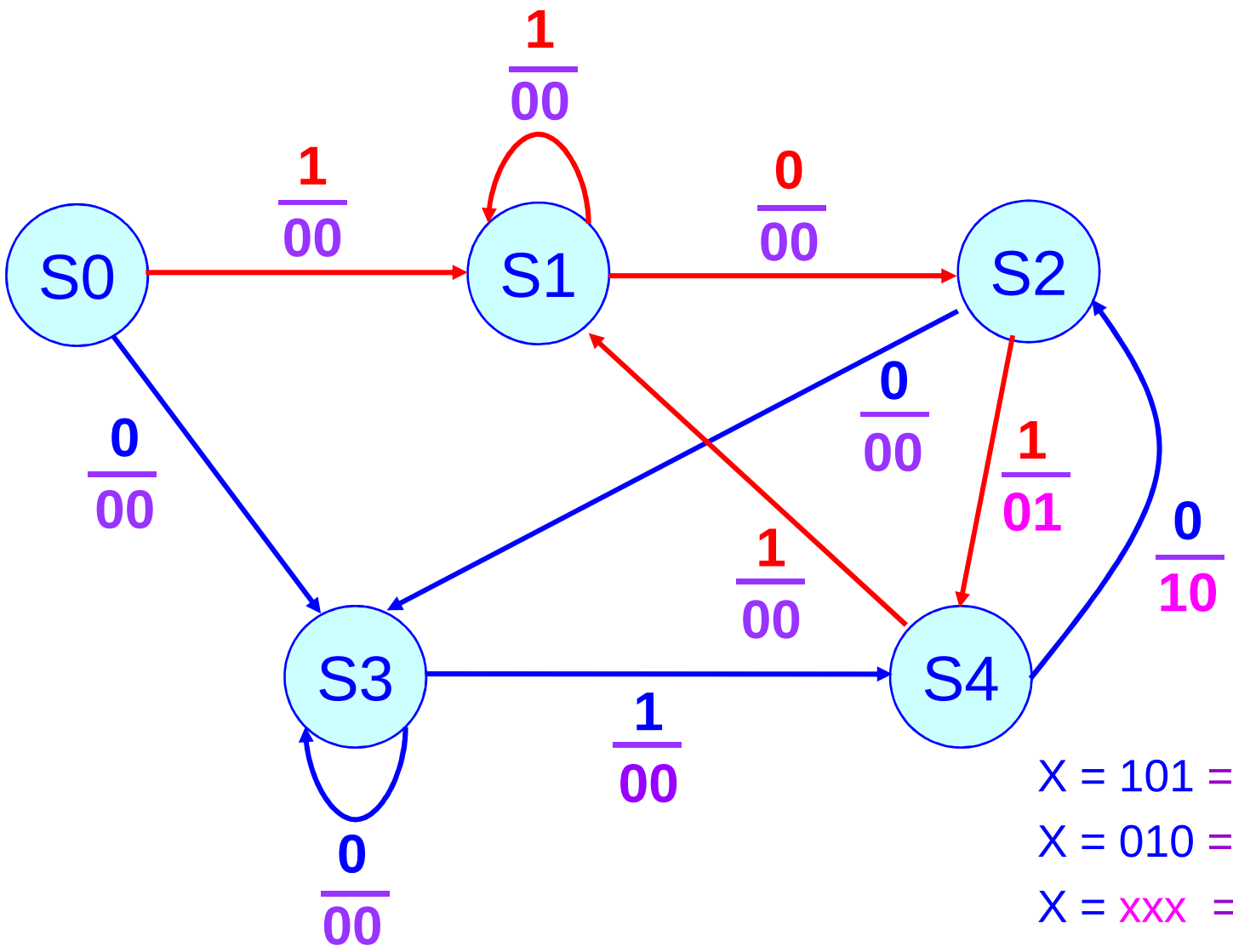
Sequencing detectors design

Ejemplo 4: Diseño de un Detector de Secuencia

Diseñar una FSM síncrona, la cual tiene una entrada X y dos salidas Z_1 y Z_0. Debe cumplir con las siguientes condiciones:

- Solo si se detecta la secuencia **101**, la salida Z_1Z_0 debe ser 01
- Solo si se detecta la secuencia de entrada **010**, la salida Z_1Z_0 debe ser 10
- Para otras secuencias, la salida Z_1Z_0 debe ser 00
- Existe superposición, diseñar usando "flipflop D", usar codificación Gray, la FSM síncrona debe ser Mealy

Ejemplo 4: Diseño de un Detector de Secuencia



X = 101 => Z_1Z_2=01

X = 010 => Z_1Z_2=10

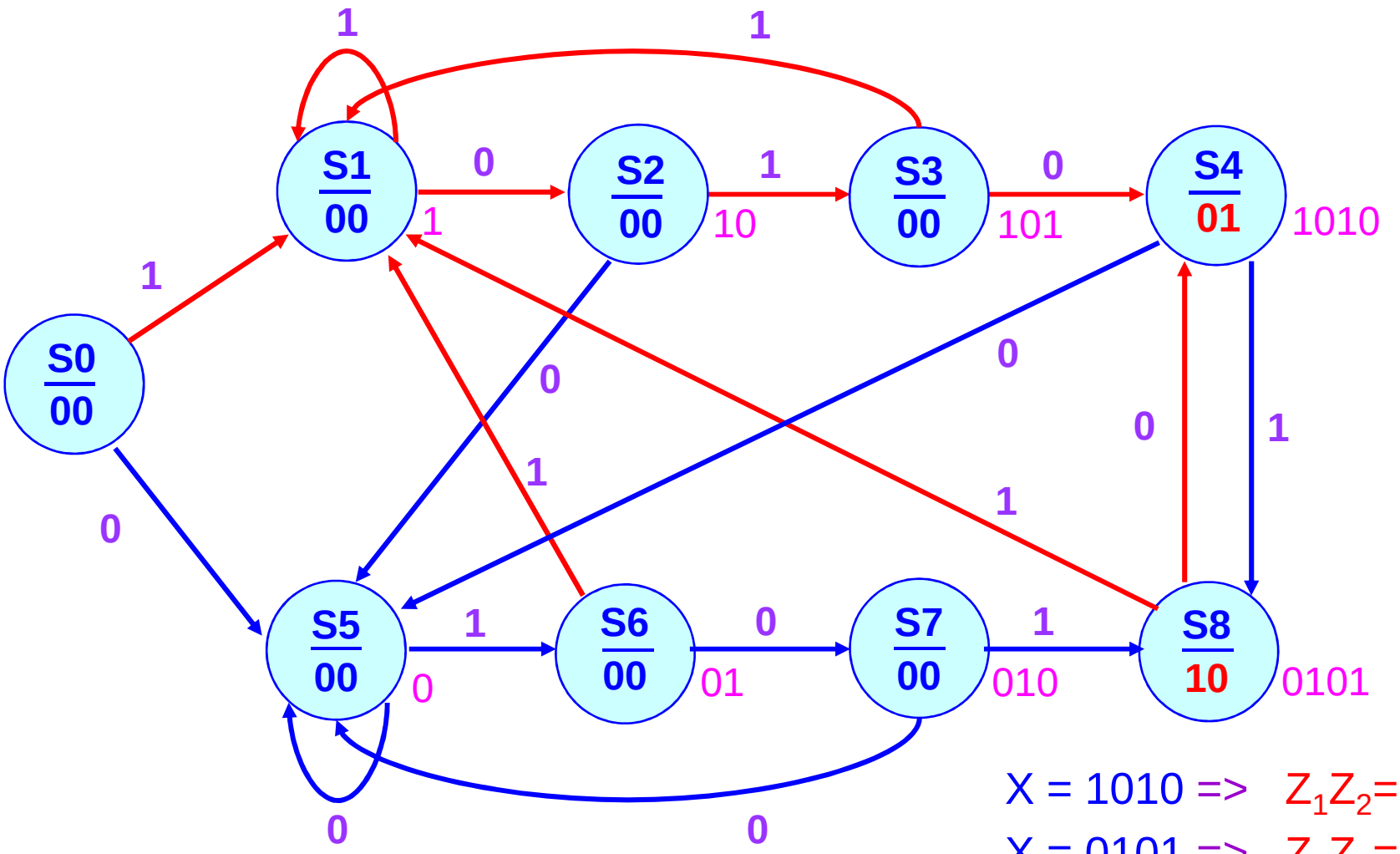
X = xxx => Z_1Z_2=00

Ejemplo 5: Diseño de un Detector de Secuencia

Diseñar una FSM síncrona, la cual tiene una entrada X y dos salidas Z_1 y Z_0. Debe cumplir con las siguientes condiciones:

- Solo si se detecta la secuencia **1010**, la salida Z_1Z_0 debe ser 01
- Solo si se detecta la secuencia de entrada **0101**, la salida Z_1Z_0 debe ser 10
- Para otras secuencias, la salida Z_1Z_0 debe ser 00
- Existe superposición, diseñar usando "flipflop D", usar codificación Gray, la FSM síncrona debe ser Moore

Ejemplo 5: Diseño de un Detector de Secuencia



X = 1010 => Z_1Z_2=01

X = 0101 => Z_1Z_2=10

X = xxxx => Z_1Z_2=00