



Titulación: Ingeniería Informática
Asignatura: Fundamentos de Computadores

Bloque 3: Sistemas secuenciales

Tema 7: Máquinas finitas de estados

Pablo Huerta Pellitero
Luis Rincón Córcoles



ÍNDICE

- Bibliografía
- Introducción
- Tipos de máquinas finitas de estados
- Síntesis de máquinas finitas de estados
 - Síntesis de máquinas de Mealy
 - Síntesis de máquinas de Moore
- Análisis de máquinas finitas de estados
 - Análisis de máquinas de Mealy
 - Análisis de máquinas de Moore



BIBLIOGRAFÍA

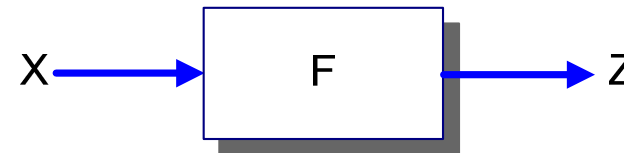
- Román Hermida, Ana M^o del Corral, Enric Pastor, Fermín Sánchez
“Fundamentos de Computadores” , cap 3
Editorial Síntesis
- Daniel D. Gajski
“Principios de Diseño Digital”, cap 5
Editorial Prentice Hall
- M. Morris Mano
“Diseño Digital”, cap 4,5
Editorial Prentice Hall



INTRODUCCIÓN

- En los sistemas **combinacionales** la salida Z en un determinado instante de tiempo t_i sólo depende de X en ese mismo instante de tiempo t_i , es decir que no tienen capacidad de memoria y que se puede obviar la variable de tiempo t

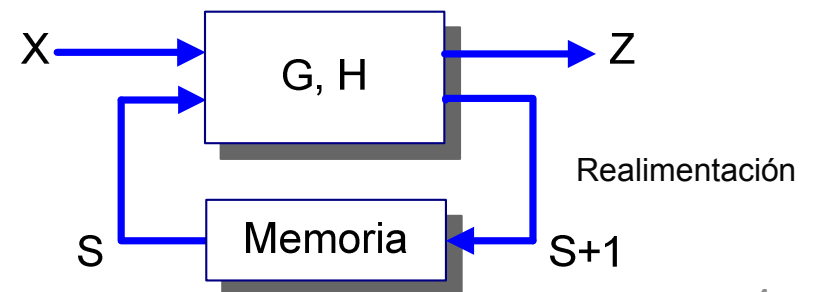
$$Z(t) = F(X(t)) \Rightarrow Z = F(X)$$



- En los sistemas **secuenciales** la salida Z en un determinado instante de tiempo t_i depende de X en ese mismo instante de tiempo t_i y en todos los instantes temporales anteriores ¿capacidad ∞ de memoria? No, todas las secuencias se resumen en un número finito de estados (**FSM: máquina finita de estados**)

$$Z(t) = G(X(t), S(t)) \quad - \text{Salida}$$

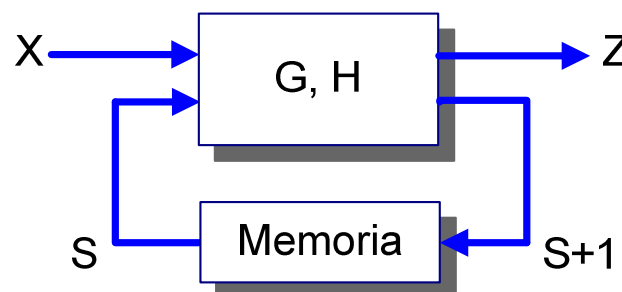
$$S(t+1) = H(X(t), S(t)) \quad - \text{Cambio de estado}$$





INTRODUCCIÓN

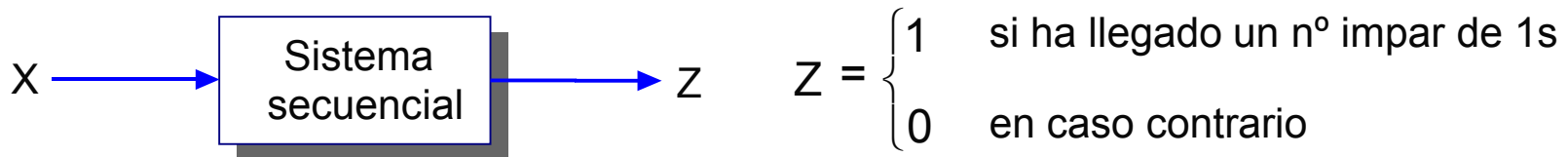
- Un sistema secuencial dispone de **elementos de memoria** cuyo contenido puede cambiar a lo largo del tiempo. Estos elementos de memoria determinan el **estado** del sistema.
- Los sistemas secuenciales suelen tener una señal que inicia los elementos de memoria con un valor determinado: **señal de inicio (reset)**.
 - La señal de inicio determina el estado del sistema en el momento del arranque (normalmente pone toda la memoria a cero).
- La salida en un instante concreto viene dada por la entrada y por el estado del sistema.
- El estado actual del sistema, junto con la entrada, determinará el estado en el instante siguiente \Rightarrow **realimentación**.





INTRODUCCIÓN

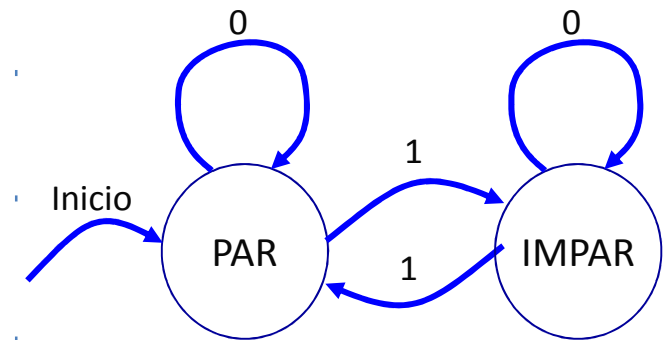
- Ejemplo: sistema secuencial que recibe datos a través de una entrada de 1 bit e indica si ha recibido un número impar de 1s.



Ejemplo de secuencia:

En t=0, condición inicial, hay 0 1s lo que significa un número par de 1s

t:	0	1	2	3	4	5	6	7	8	9	10	...
S:	P	P	P	I	P	I	I	I	I	I	P	...
X:	0	0	1	1	1	0	0	0	0	1	0	...
S+1:	P	P	I	P	I	I	I	I	I	P	P	...
Z:	0	0	1	0	1	1	1	1	1	0	0	...

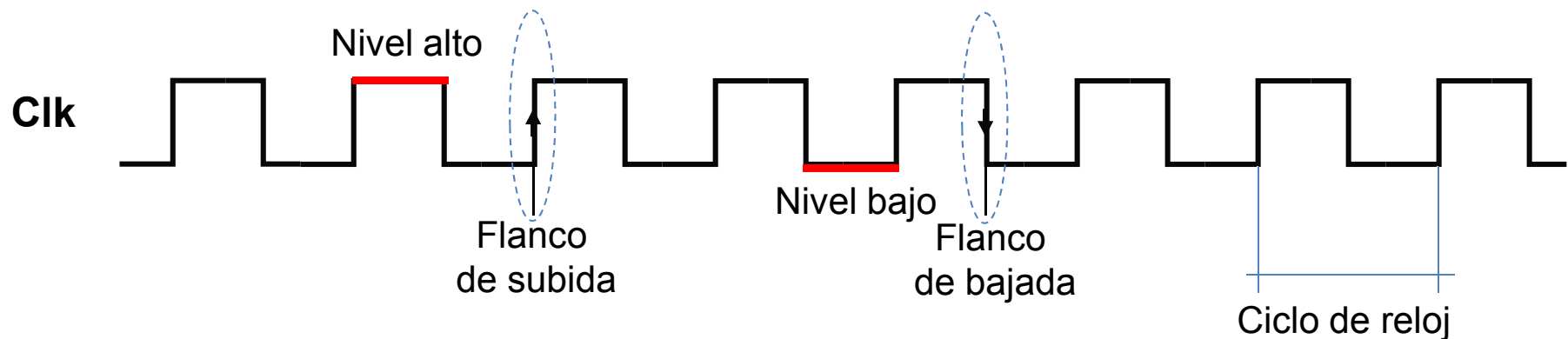


En cualquier instante de tiempo **sólo hay dos posibles clases de secuencias**, las que han recibido un n^o par o un n^o impar de 1s, esto significa que todos los datos recibidos se pueden clasificar en dos clases o estados (S): PAR e IMPAR. Por tanto, es una Máquina Finita de Estados (FSM).



SINCRONISMO EN MÁQUINAS DE ESTADOS

- Existen dos tipos de sistemas secuenciales: asíncronos y síncronos.
 - Los **asíncronos** son sistemas secuenciales que pueden cambiar de estado en cualquier instante de tiempo en función de cambios en las señales de entrada.
 - Los **síncronos** son sistemas secuenciales que sólo pueden cambiar de estado en determinados instantes de tiempos, es decir, están “sincronizados” con una señal que indica dicho instante y que se conoce como señal de reloj (Clk), sin importar si las señales de entrada han cambiado o no. Debido a su peso específico en el diseño sólo consideraremos los secuenciales síncronos.



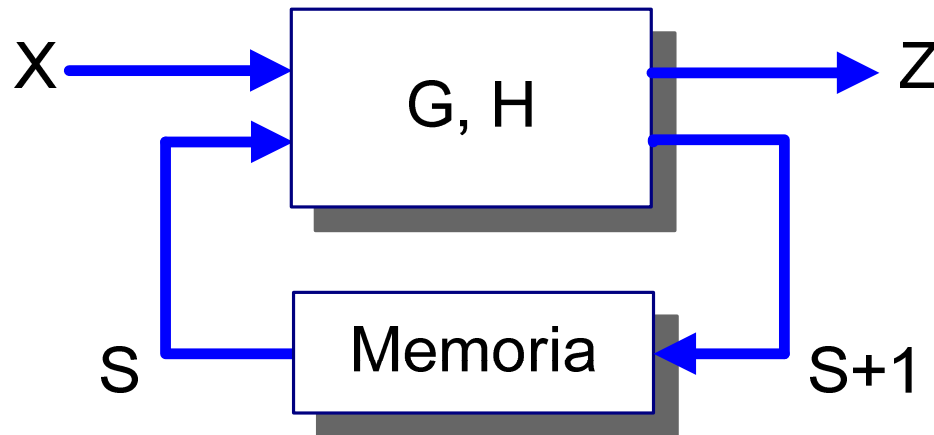


ÍNDICE

- Bibliografía
- Introducción
- **Tipos de máquinas finitas de estados**
- Síntesis de máquinas finitas de estados
 - Síntesis de máquinas de Mealy
 - Síntesis de máquinas de Moore
- Análisis de máquinas finitas de estados
 - Análisis de máquinas de Mealy
 - Análisis de máquinas de Moore



TIPOS DE MÁQUINAS FINITAS DE ESTADOS



$X(t)$: entrada actual

$Z(t)$: salida actual

$S(t)$: estado actual

$S(t+1)$: estado próximo

Las FSM constan de:

- Un conjunto de entradas $X \in \{X_0, X_1, \dots, X_{k-1}\}$
- Un conjunto de salidas $Z \in \{Z_0, Z_1, \dots, Z_{m-1}\}$
- Un conjunto de estados $S \in \{S_0, S_1, \dots, S_{n-1}\}$
- Una función de transición $S(t+1) = H(X(t), S(t))$
- Una función de salida $Z(t) = G(X(t), S(t))$

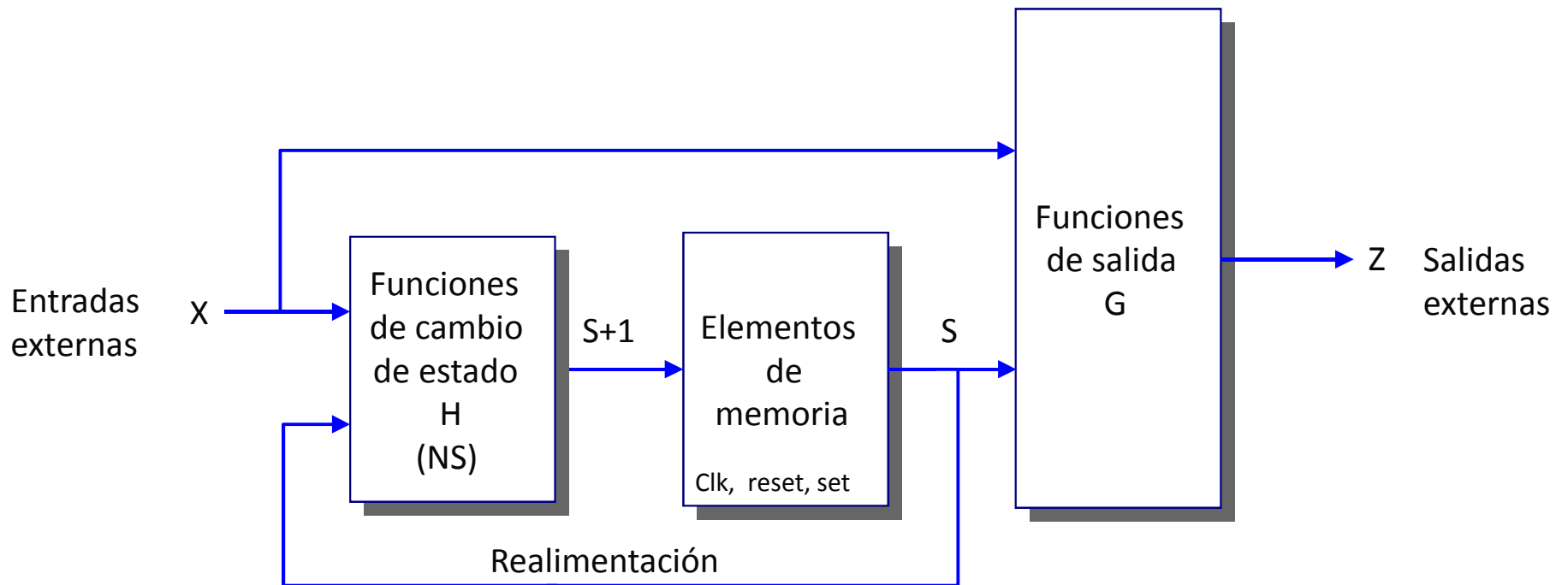
Tipos de FSM:

- **Mealy**
- **Moore**



MÁQUINAS FINITAS DE ESTADOS: MEALY

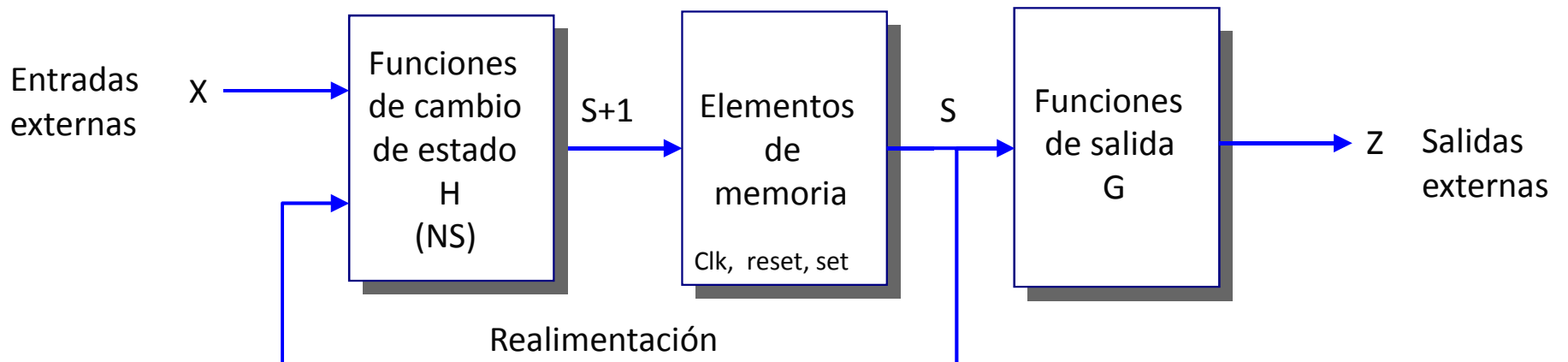
- FSM tipo Mealy:
 - El próximo estado del sistema se genera a través de la **función de transición de estados H** que genera el próximo estado (NS), y que actúa en función del estado actual del sistema (S) y de las entradas presentes (X).
 - La **función de salida (G)** se genera a partir del estado actual del sistema (S) y de los valores actuales de las entradas (X).





MÁQUINAS FINITAS DE ESTADOS: MOORE

- FSM tipo Moore (caso particular de Mealy):
 - El próximo estado del sistema se genera, como en las máquinas de Mealy, a través de la **función de transición de estados** H que genera el próximo estado (NS), y que actúa en función del estado del sistema (S) y de los valores de las entradas (X).
 - La **función de salida** (G) se genera, a diferencia de las máquinas de Mealy, exclusivamente en función del estado actual del sistema (S), sin importar el valor de las entradas.

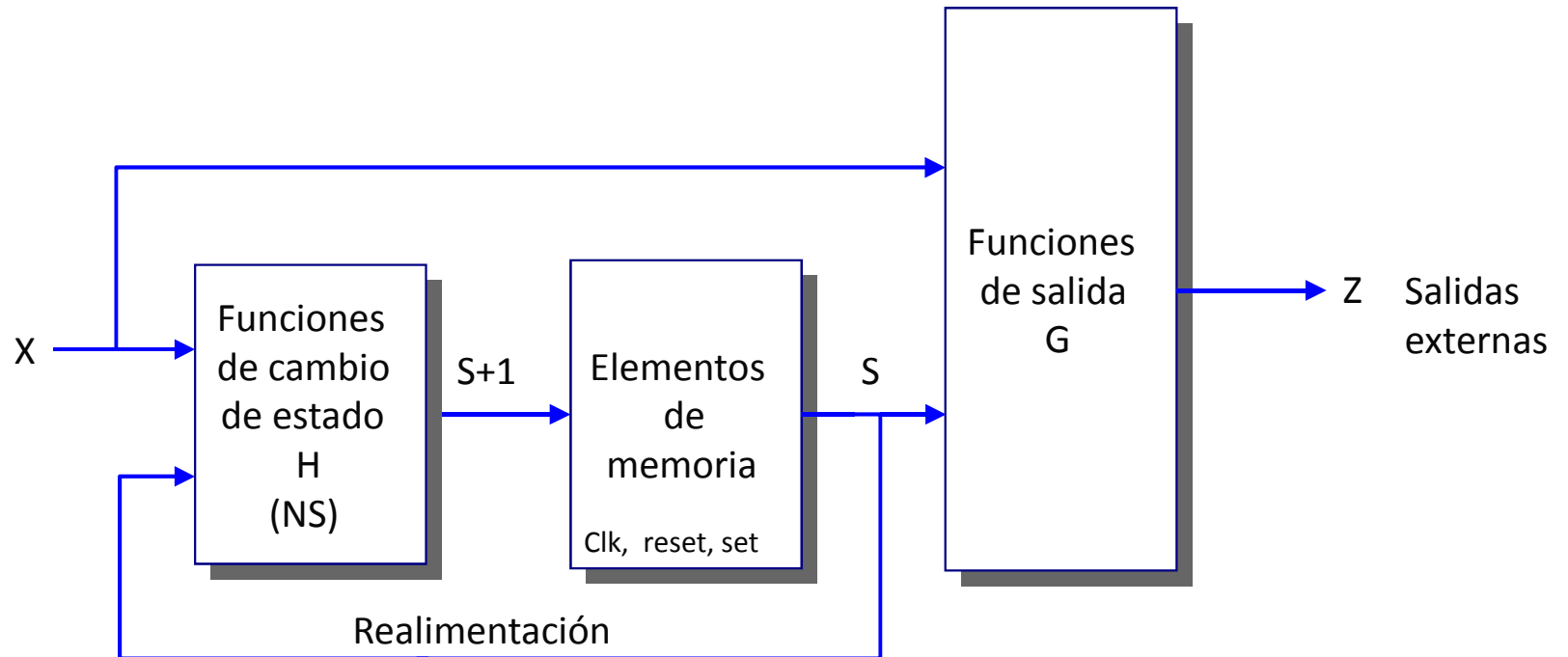




MÁQUINAS FINITAS DE ESTADOS: MEALY Y MOORE

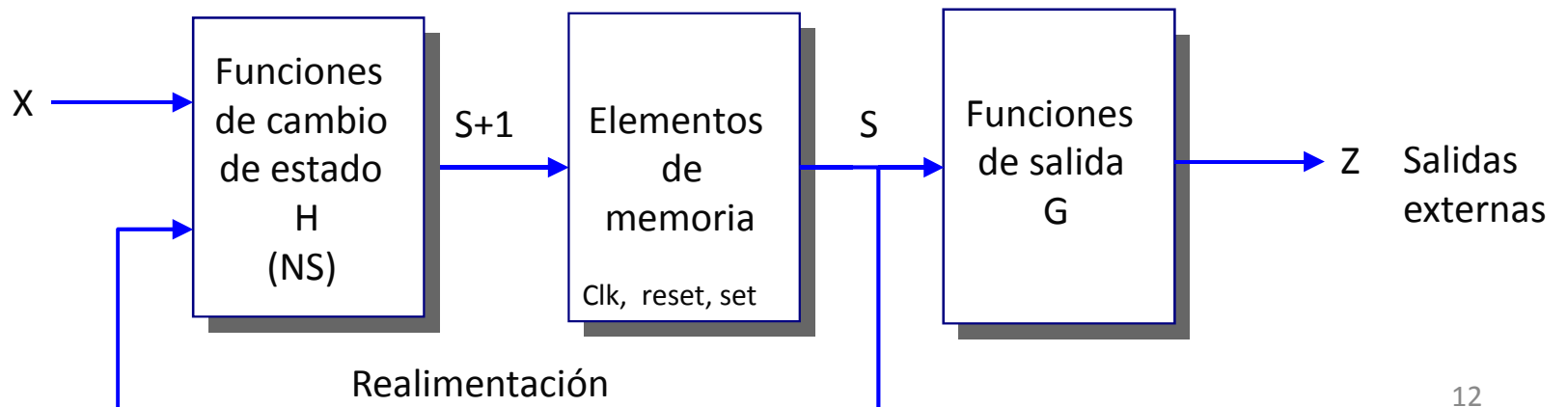
Máquina de Mealy

Entradas externas



Máquina de Moore

Entradas externas



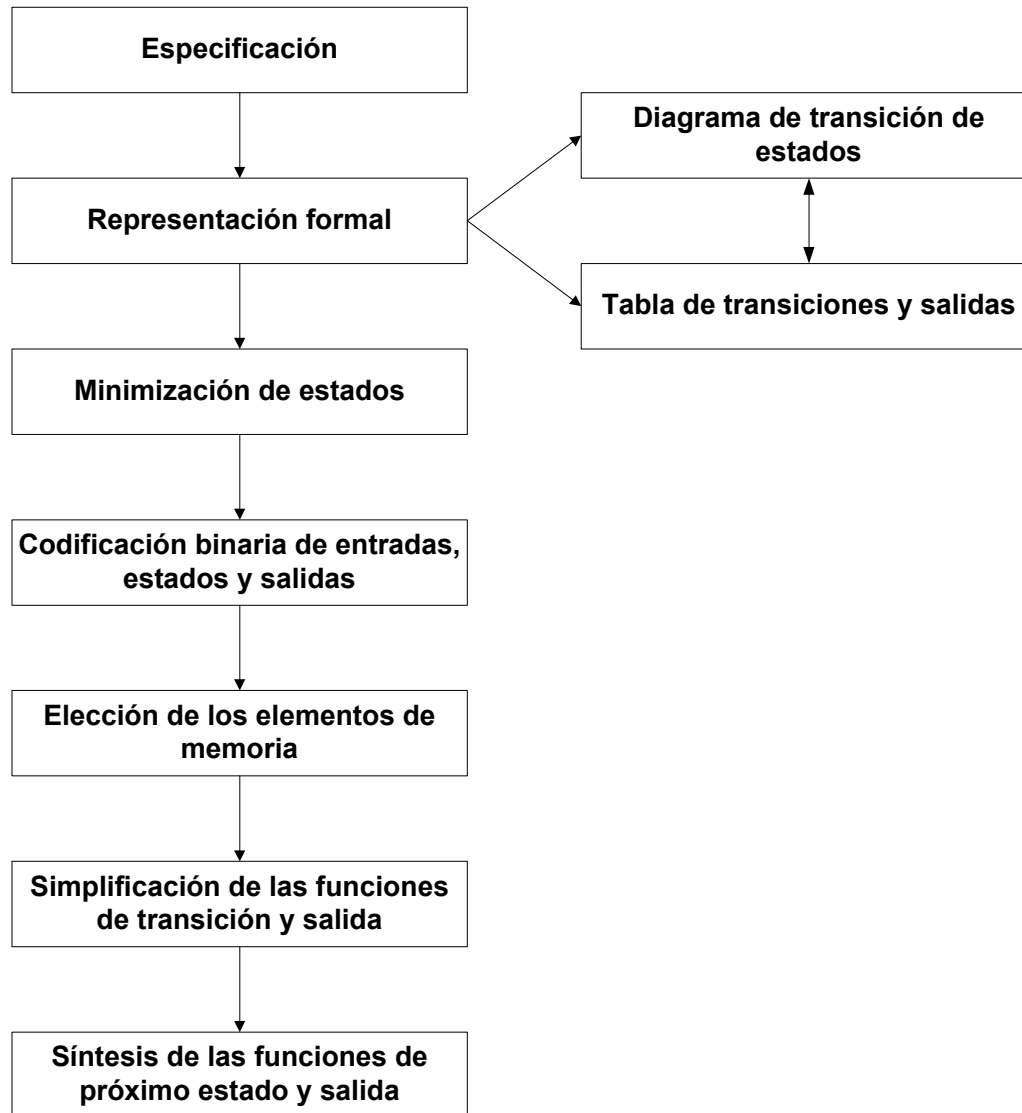


ÍNDICE

- Bibliografía
- Introducción
- Tipos de máquinas finitas de estados
- **Síntesis de máquinas finitas de estados**
 - Síntesis de máquinas de Mealy
 - Síntesis de máquinas de Moore
- Análisis de máquinas finitas de estados
 - Análisis de máquinas de Mealy
 - Análisis de máquinas de Moore



SÍNTESIS DE MÁQUINAS FINITAS DE ESTADOS



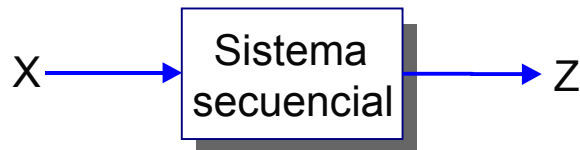


SÍNTESIS DE FSM: ESPECIFICACIÓN

Veremos los pasos de diseño a partir de un ejemplo.

Especificación de un sistema secuencial

Diseñar un sistema secuencial con una entrada serie que detecte si los tres últimos datos recibidos coinciden con la secuencia **abb**.



$$X \in \{a, b\}$$

$$Z \in \{p, q\}$$

$$Z = \begin{cases} q & \text{si los tres últimos datos recibidos son abb} \\ p & \text{en caso contrario} \end{cases}$$

Después de la especificación del sistema secuencial, el siguiente paso es representarlo formalmente.



SÍNTESIS DE FSM: REPRESENTACIÓN FORMAL - MEALY

- La representación formal de un sistema secuencial se suele hacer en forma de **tabla de estados y salidas** o **diagrama de estados**. Ambas son formas equivalentes.

Diagrama de estados

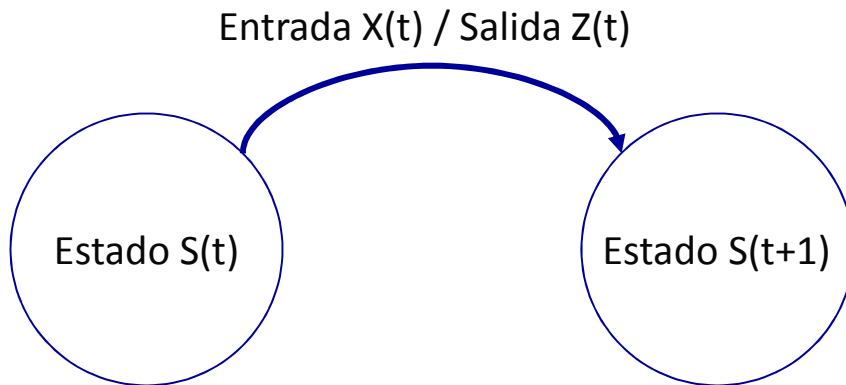


Tabla de estados y salidas

Estado actual	Entrada actual			
	X_0	X_1	...	X_m
S_0	$S_{0,0}/Z_{0,0}$	$S_{0,1}/Z_{0,1}$...	$S_{0,m}/Z_{0,m}$
S_1	$S_{1,0}/Z_{1,0}$	$S_{1,1}/Z_{1,1}$...	$S_{1,m}/Z_{1,m}$
...
S_n	$S_{n,0}/Z_{n,0}$	$S_{n,1}/Z_{n,1}$...	$S_{n,m}/Z_{n,m}$

Estado siguiente / Salida



SÍNTESIS DE FSM: REPRESENTACIÓN FORMAL - MEALY

- Ejemplo: Diagrama de estados del caso propuesto.

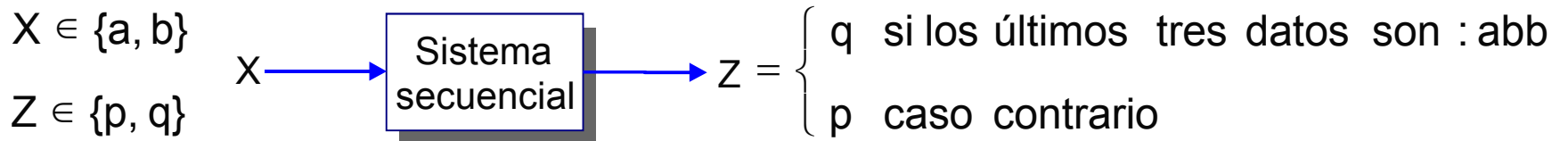


Diagrama de estados

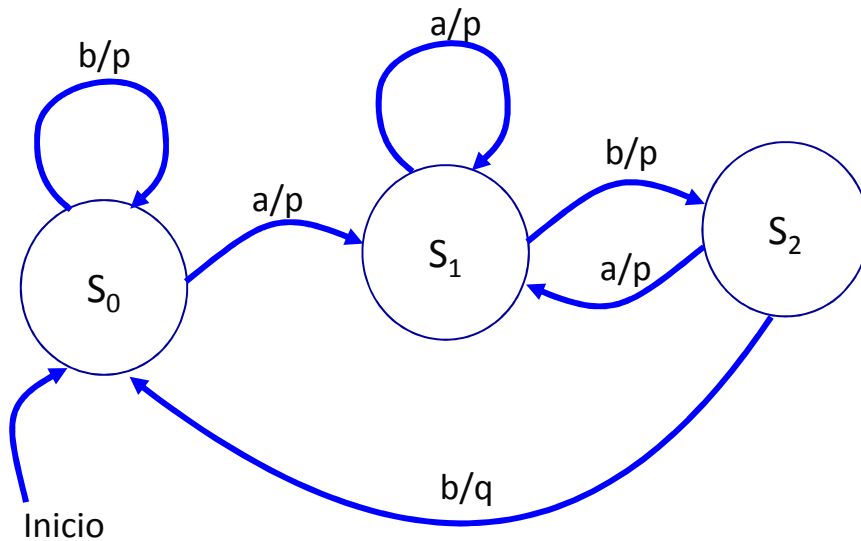


Tabla de estados y salidas

Estado actual	Entrada actual	
	a	b
S_0	S_1/p	S_0/p
S_1	S_1/p	S_2/p
S_2	S_1/p	S_0/q

Estado siguiente / Salida



SÍNTESIS DE FSM: CODIFICACIÓN BINARIA - MEALY

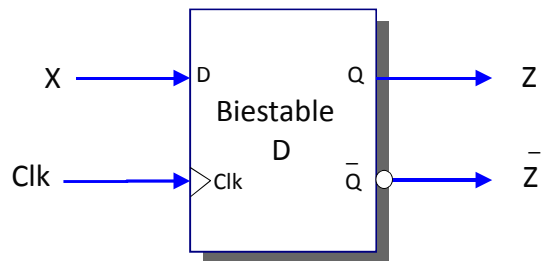
- A la hora de materializar el circuito hay que transformar la tabla de estados y salidas asignando valores binarios a cada estado y salida.
- Distintas asignaciones pueden conducir a materializaciones con prestaciones distintas aunque funcionalmente equivalentes.
- Ejemplo: codificación binaria del caso propuesto.

Entrada		Salida		Estado			Tabla de estados y salidas			
X(t)	X ₀	Z(t)	Z ₀	S(t)	Q ₁	Q ₀	S(t)		X ₀	
							Q ₁	Q ₀	0	1
a	0	p	0	S ₀	0	0	0	0	01/0	00/0
b	1	q	1	S ₁	0	1	0	1	01/0	10/0
				S ₂	1	0	1	0	01/0	00/1

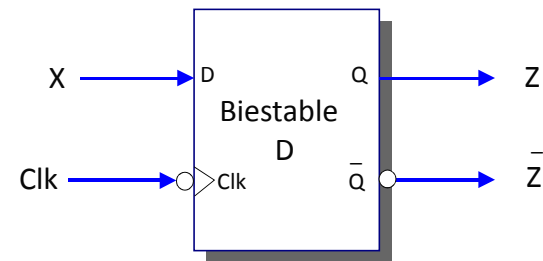
ELEMENTOS DE MEMORIA: BIESTABLES

- Un **biestable** es un dispositivo capaz de almacenar un bit (H ó L).
 - El biestable siempre ofrece a la salida el valor que tiene almacenado en su interior.
- Existen diferentes tipos de biestables, pero el más adecuado y sencillo en nuestro caso es el **biestable D (Delay) activo por flanco de reloj**:
 - El biestable D activo por flanco de subida (de bajada) captura el valor que tiene en su entrada de datos cuando se produce el flanco de subida (de bajada) del reloj.

Biestable D disparado por flanco de subida



Biestable D disparado por flanco de bajada



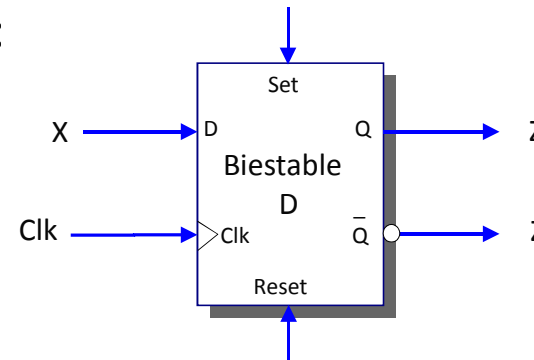
- A los biestables activos por flanco se les denomina también ***flip-flops***.



ELEMENTOS DE MEMORIA: BIESTABLES

- Los biestables suelen tener entradas asíncronas (independientes del reloj) que sirven para darle valor inicial:

- *Reset (o Clear)*: puesta a 0.
- *Set (o Preset)*: puesta a 1.



- Las entradas asíncronas tienen prioridad sobre las síncronas.
- Modo de operación (biestable D activo por flanco de subida):

Set	Reset	D	Clk	Q(t+1)	Not Q(t+1)	
1	0	X	X	1	0	Set
0	1	X	X	0	1	Reset
1	1	X	X	1	1	No permitido
0	0	0	↑	0	1	Flanco positivo
0	0	1	↑	1	0	Flanco positivo
0	0	X	0	Q(t)	Not Q(t)	Retención
0	0	X	1	Q(t)	Not Q(t)	Retención
0	0	X	↓	Q(t)	Not Q(t)	Retención

En modo síncrono, para poner un valor en un biestable D activo por flanco basta con colocar dicho valor en su entrada de datos antes de que llegue el flanco.

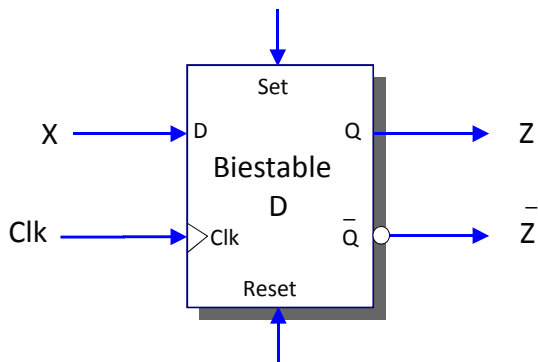


SÍNTESIS DE FSM: SIMPLIFICACIÓN - MEALY

- Una vez seleccionado el tipo de biestable, la codificación binaria sirve para crear la tabla de verdad de las funciones de transición y salida (**tabla de excitación y salida**).
- La síntesis se realiza de forma similar a la de los circuitos combinacionales.
- Ejemplo: tabla de excitación y salida del caso propuesto.

S(t)		X ₀	
Q ₁	Q ₀	0	1
0	0	01/0	00/0
0	1	01/0	10/0
1	0	01/0	00/1

S(t), X			S(t+1), Z		
Q ₁	Q ₀	X	Q' ₁	Q' ₀	Z
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	X	X	X
1	1	1	X	X	X





SÍNTESIS DE FSM: SIMPLIFICACIÓN - MEALY

S(t), X			S(t+1), Z		
Q ₁	Q ₀	X	Q' ₁	Q' ₀	Z
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	X	X	X
1	1	1	X	X	X



Función de transición

$$Q'_0(Q_1, Q_0, X_0) = \sum m(0, 2, 4) + \sum \Phi(6, 7) \Rightarrow$$
$$Q'_0 = D_0 = \overline{X_0}$$

$$Q'_1(Q_1, Q_0, X_0) = \sum m(3) + \sum \Phi(6, 7) \Rightarrow$$
$$Q'_1 = D_1 = X_0 \cdot Q_0$$

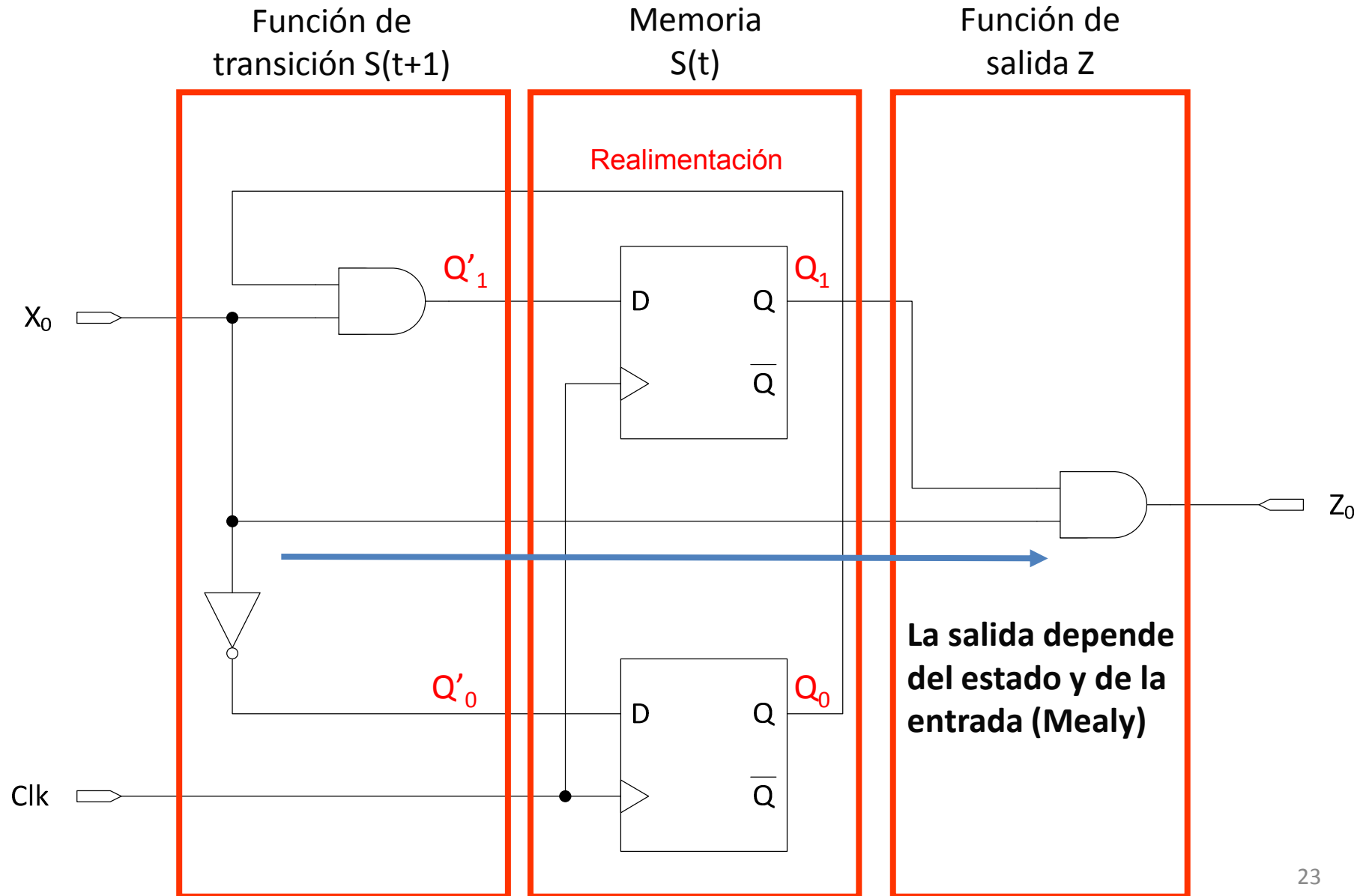
Función de salida

$$Z_0(Q_1, Q_0, X_0) = \sum m(5) + \sum \Phi(6, 7) \Rightarrow$$
$$Z_0 = X_0 \cdot Q_1$$

- La función de transición (par Q'_1 - Q'_0) y la función de salida (Z_0) se materializan mediante puertas lógicas o mediante los dispositivos combinatoriales que se indiquen, junto con los biestables seleccionados para almacenar el estado.



SÍNTESIS DE FSM: MEALY





SÍNTESIS DE FSM: REPRESENTACIÓN FORMAL - MOORE

- En la máquina de Moore, la salida no va ligada a la transición de un estado a otro, sino que depende únicamente del estado.

Diagrama de estados

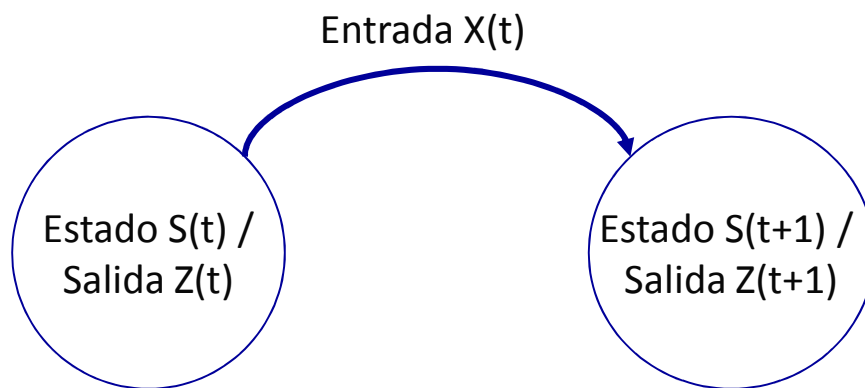


Tabla de estados y salidas

Estado actual	Entrada actual				Salida
	X_0	X_1	...	X_m	
S_0	$S_{0,0}$	$S_{0,1}$...	$S_{0,m}$	Z_0
S_1	$S_{1,0}$	$S_{1,1}$...	$S_{1,m}$	Z_1
...
S_n	$S_{n,0}$	$S_{n,1}$...	$S_{n,m}$	Z_n
	Estado siguiente				



SÍNTESIS DE FSM: REPRESENTACIÓN FORMAL - MOORE

- Ejemplo: Diagrama de estados del caso propuesto.

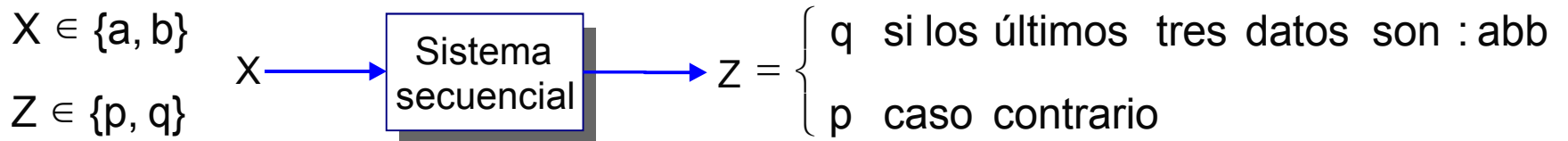


Diagrama de estados

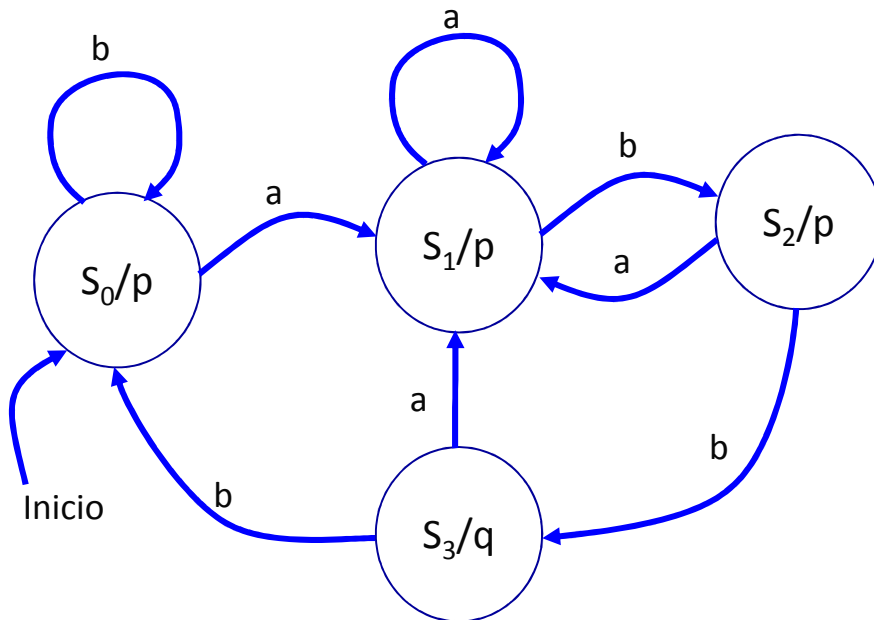


Tabla de estados y salidas

Estado actual	Entradas		Salida
	a	b	
S_0	S_1	S_0	p
S_1	S_1	S_2	p
S_2	S_1	S_3	p
S_3	S_1	S_0	q

Estado siguiente



SÍNTESIS DE FSM: CODIFICACIÓN BINARIA - MOORE

- A la hora de materializar el circuito hay que transformar la tabla de estados y salidas asignando valores binarios a cada estado y salida.
- Distintas asignaciones pueden conducir a materializaciones con prestaciones distintas aunque funcionalmente equivalentes.
- Ejemplo: codificación binaria del caso propuesto.

Entrada		Estado			Tabla de estados y salidas				
X(t)	X ₀	S(t)	Q ₁	Q ₀	S(t)		X ₀		Z
					Q ₁	Q ₀	0	1	Z ₀
a	0	S ₀	0	0	0	0	01	00	0
b	1	S ₁	0	1	0	1	01	10	0
		S ₂	1	0	1	0	01	11	0
		S ₃	1	1	1	1	01	00	1
Salida									
Z(t)	X ₀								
p	0								
q	1								

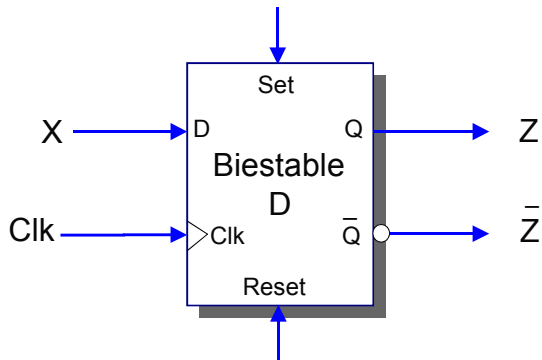


SÍNTESIS DE FSM: SIMPLIFICACIÓN - MOORE

- Una vez seleccionado el tipo de biestable, crearemos las tablas de verdad de las funciones de transición y salida (**tabla de excitación y tabla de salida**).
- La síntesis se realiza de forma similar a la de los circuitos combinacionales.
- Ejemplo: tabla de excitación y tabla de salida del caso propuesto.

S(t)		X ₀		Z
Q ₁	Q ₀	0	1	Z ₀
0	0	01	00	0
0	1	01	10	0
1	0	01	11	0
1	1	01	00	1

S(t), X			S(t+1)		S(t)		Z
Q ₁	Q ₀	X	Q' ₁	Q' ₀	Q ₁	Q ₀	Z ₀
0	0	0	0	1	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	1	1	0	0
0	1	1	1	0	1	1	1
1	0	0	0	1			
1	0	1	1	1			
1	1	0	0	1			
1	1	1	0	0			





SÍNTESIS DE FSM: SIMPLIFICACIÓN - MOORE

S(t), X			S(t+1)	
Q ₁	Q ₀	X	Q' ₁	Q' ₀
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0

$$Q'_0(Q_1, Q_0, X_0) = \sum m(0, 2, 4, 5, 6) \Rightarrow$$

$$Q'_0 = D_0 = Q_1 \cdot \overline{Q_0} + \overline{X_0}$$

$$Q'_1(Q_1, Q_0, X_0) = \sum m(3, 5) \Rightarrow$$

$$Q'_1 = D_1 = Q_1 \cdot \overline{Q_0} \cdot X_0 + \overline{Q_1} \cdot Q_0 \cdot X_0$$

S(t)		Z
Q ₁	Q ₀	Z ₀
0	0	0
0	1	0
1	0	0
1	1	1

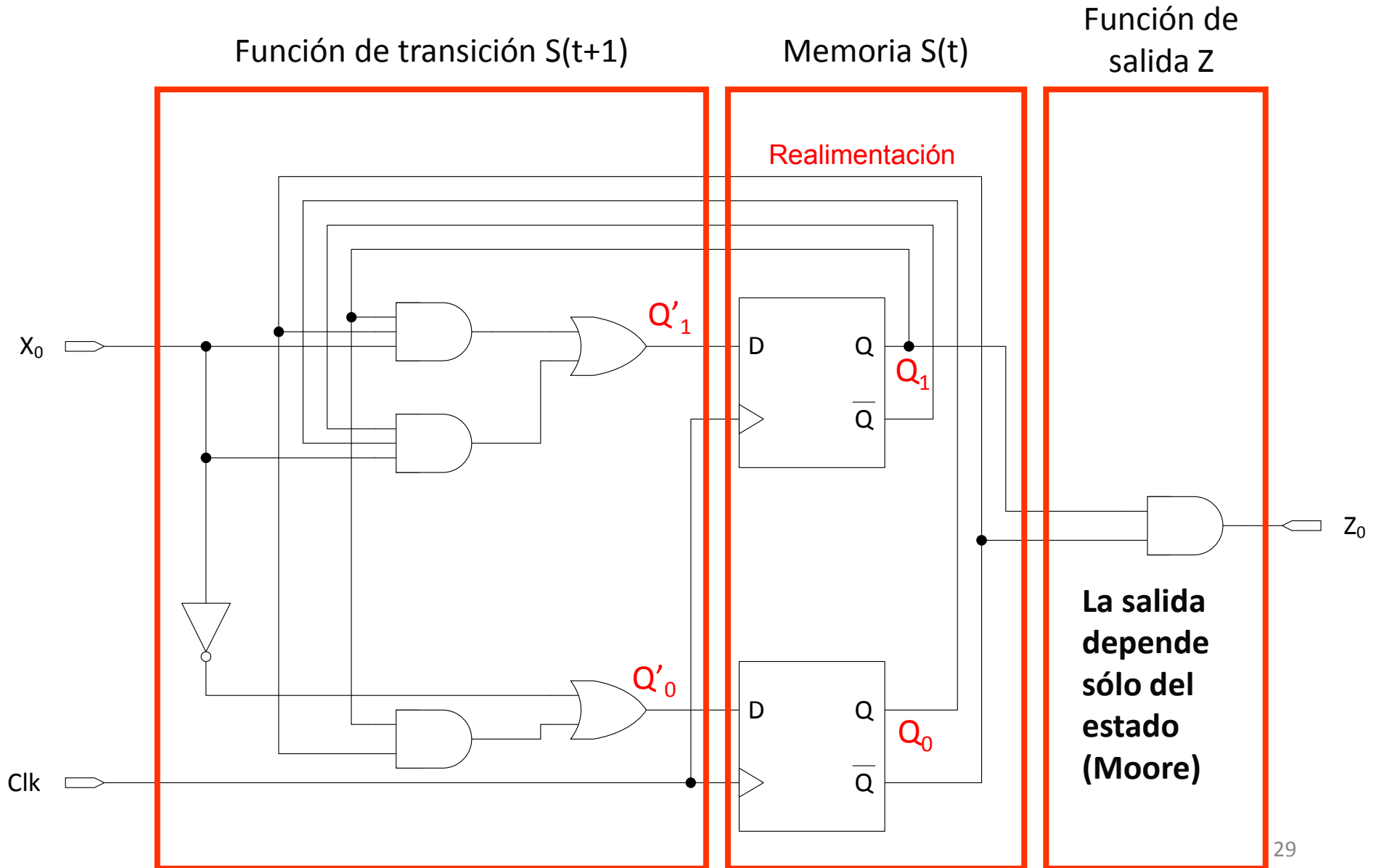
$$Z_0(Q_1, Q_0) = \sum m(3) \Rightarrow$$

$$Z_0 = Q_1 \cdot Q_0$$

- La función de transición (par Q'₁-Q'₀) y la función de salida (Z₀) se materializan mediante puertas lógicas o mediante los dispositivos combinatoriales que se indiquen, junto con los biestables seleccionados para almacenar el estado.



SÍNTESIS DE FSM: MOORE



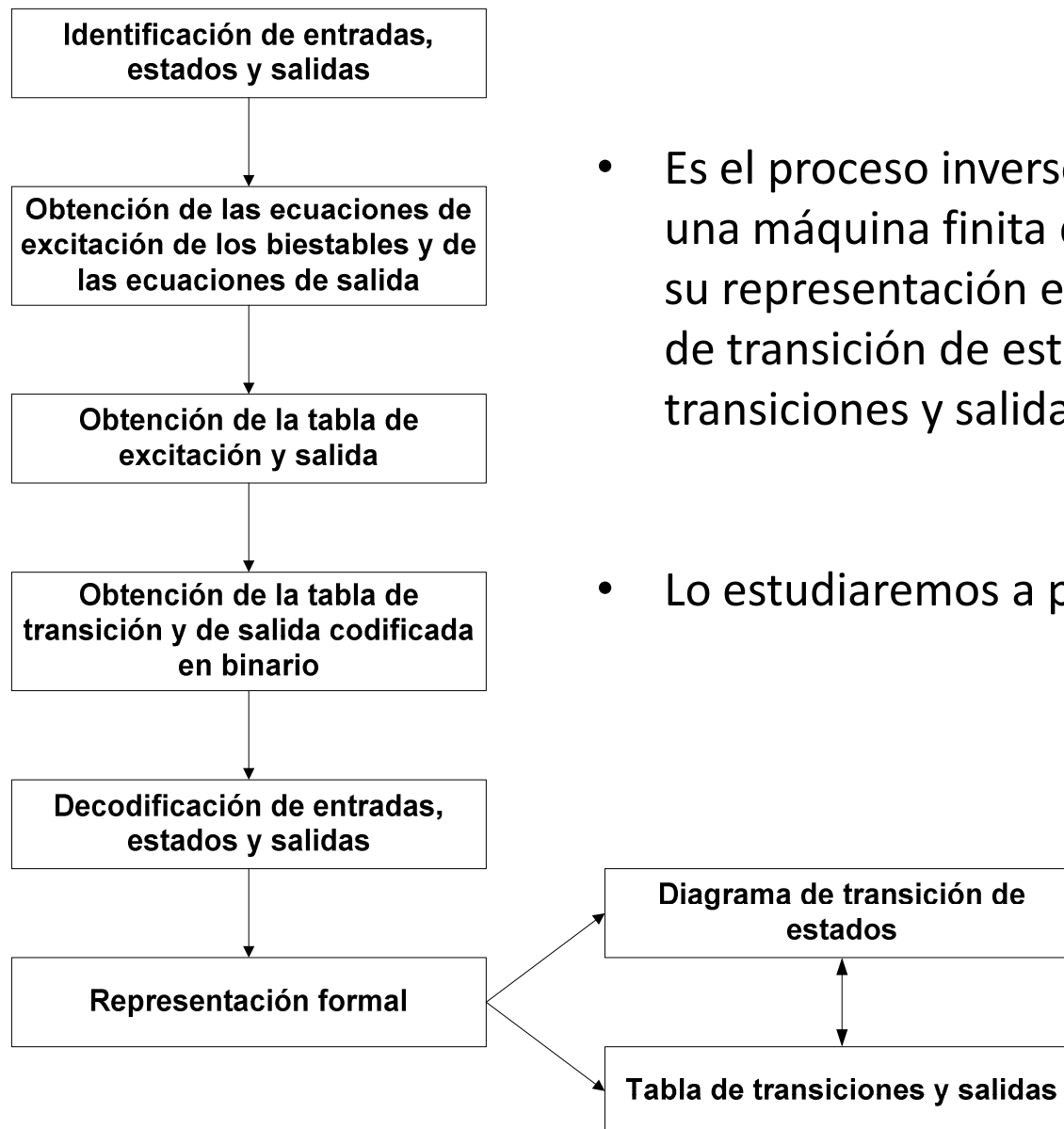


ÍNDICE

- Bibliografía
- Introducción
- Tipos de máquinas finitas de estados
- Síntesis de máquinas finitas de estados
 - Síntesis de máquinas de Mealy
 - Síntesis de máquinas de Moore
- **Análisis de máquinas finitas de estados**
 - **Análisis de máquinas de Mealy**
 - **Análisis de máquinas de Moore**



ANÁLISIS DE MÁQUINAS FINITAS DE ESTADOS

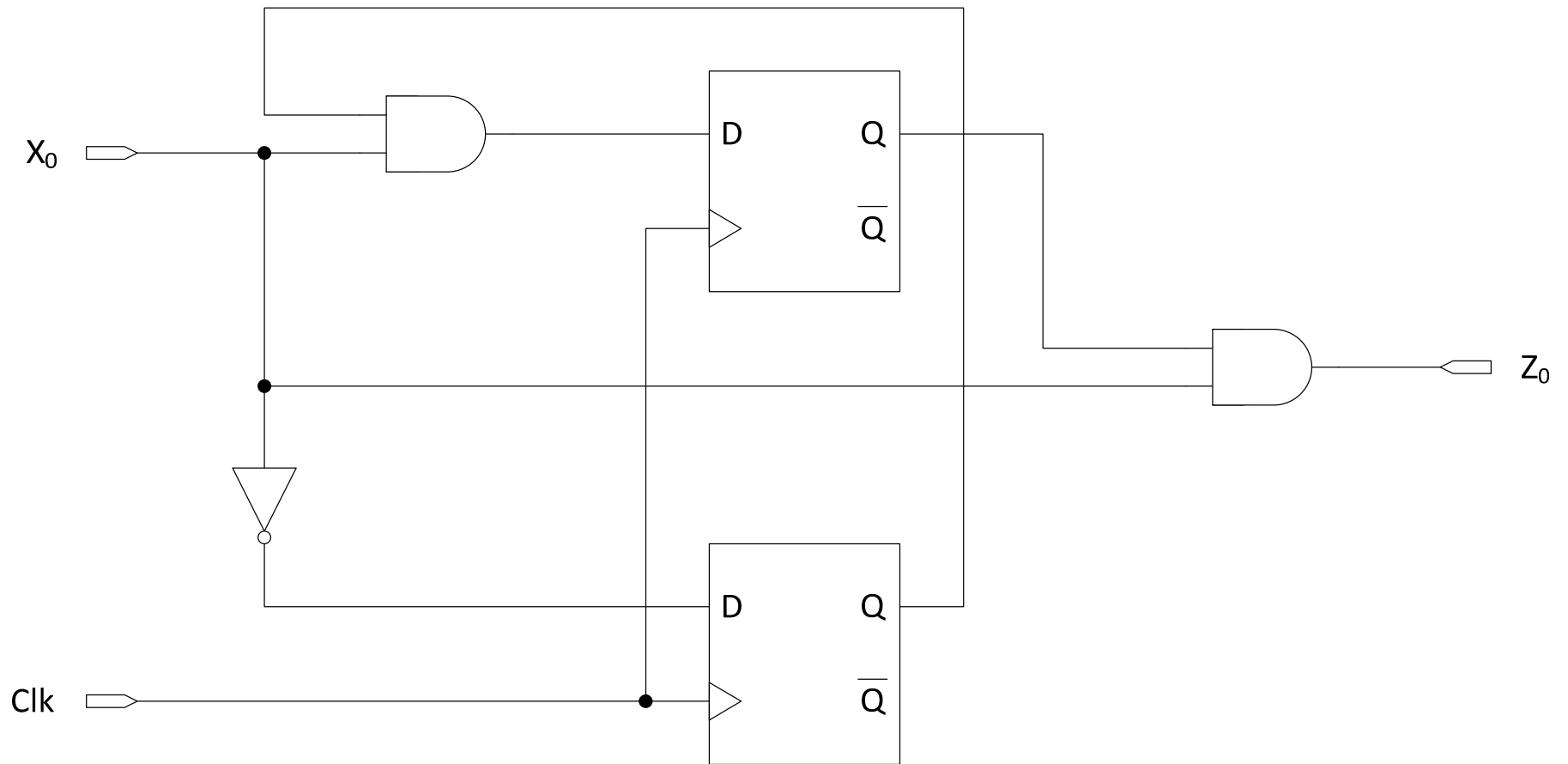


- Es el proceso inverso al de síntesis: dada una máquina finita de estados, obtener su representación en forma de diagrama de transición de estados y/o de tablas de transiciones y salidas.
- Lo estudiaremos a partir de ejemplos.



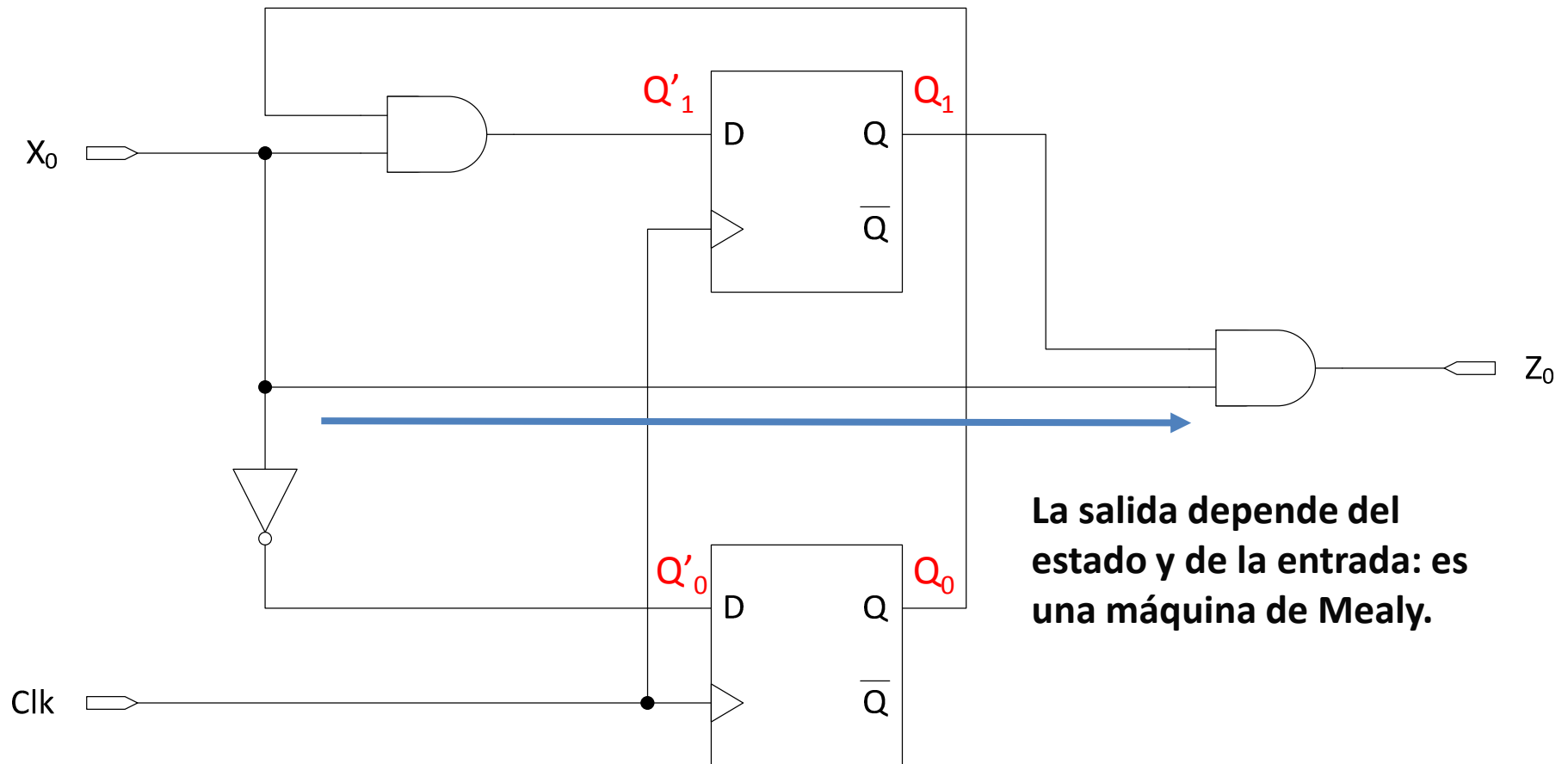
ANÁLISIS DE FSM: EJEMPLO 1

- Partimos de la siguiente FSM:





ANÁLISIS DE FSM: IDENTIFICACIÓN



- Entradas: $X_0 \in \{0,1\}$
- Salidas: $Z_0 \in \{0,1\}$
- Variables de estado: 2 $\{Q_1, Q_0\}$ darán lugar a 4 estados como máximo.



ANÁLISIS DE FSM: ECUACIONES Y TABLAS - MEALY


Función de transición

$$D_0 = Q'_0(Q_1, Q_0, X_0) = \overline{X_0}$$

$$D_1 = Q'_1(Q_1, Q_0, X_0) = X_0 \cdot Q_0$$




Función de salida

$$Z_0(Q_1, Q_0, X_0) = X_0 \cdot Q_1$$



S(t), X			S(t+1), Z		
Q ₁	Q ₀	X	Q' ₁	Q' ₀	Z
0	0	0	0	1	0
0	0	1	0	0	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	0	1	0
1	1	1	1	0	1

La salida depende del estado y de la entrada: es una máquina de Mealy.



S(t)		X ₀	
Q ₁	Q ₀	0	1
0	0	01/0	00/0
0	1	01/0	10/0
1	0	01/0	00/1
1	1	01/0	10/1

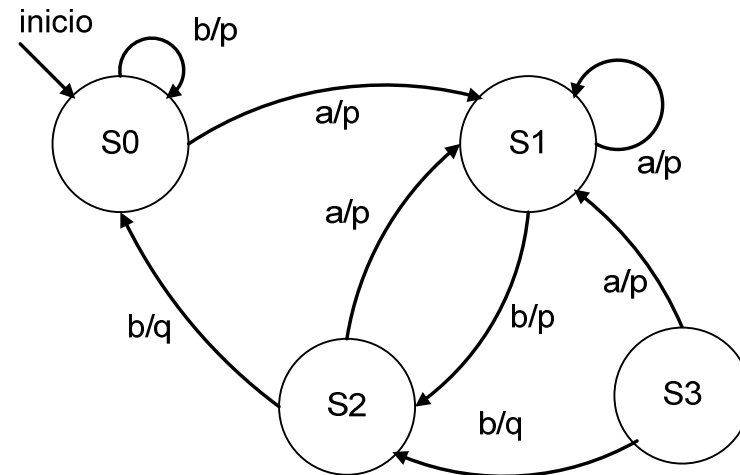


ANÁLISIS DE FSM: DECODIFICACIÓN Y REPRESENTACIÓN FORMAL - MEALY

- Asignamos nombres y valores a las entradas, los estados y la salida.

S(t)		X ₀		Entrada		Salida		Estado		
Q ₁	Q ₀	0	1	X ₀	X(t)	Z ₀	Z(t)	Q ₁	Q ₀	S(t)
0	0	01/0	00/0	0	a	0	p	0	0	S ₀
0	1	01/0	10/0	1	b	1	q	0	1	S ₁
1	0	01/0	00/1					1	0	S ₂
1	1	01/0	10/1					1	1	S ₃

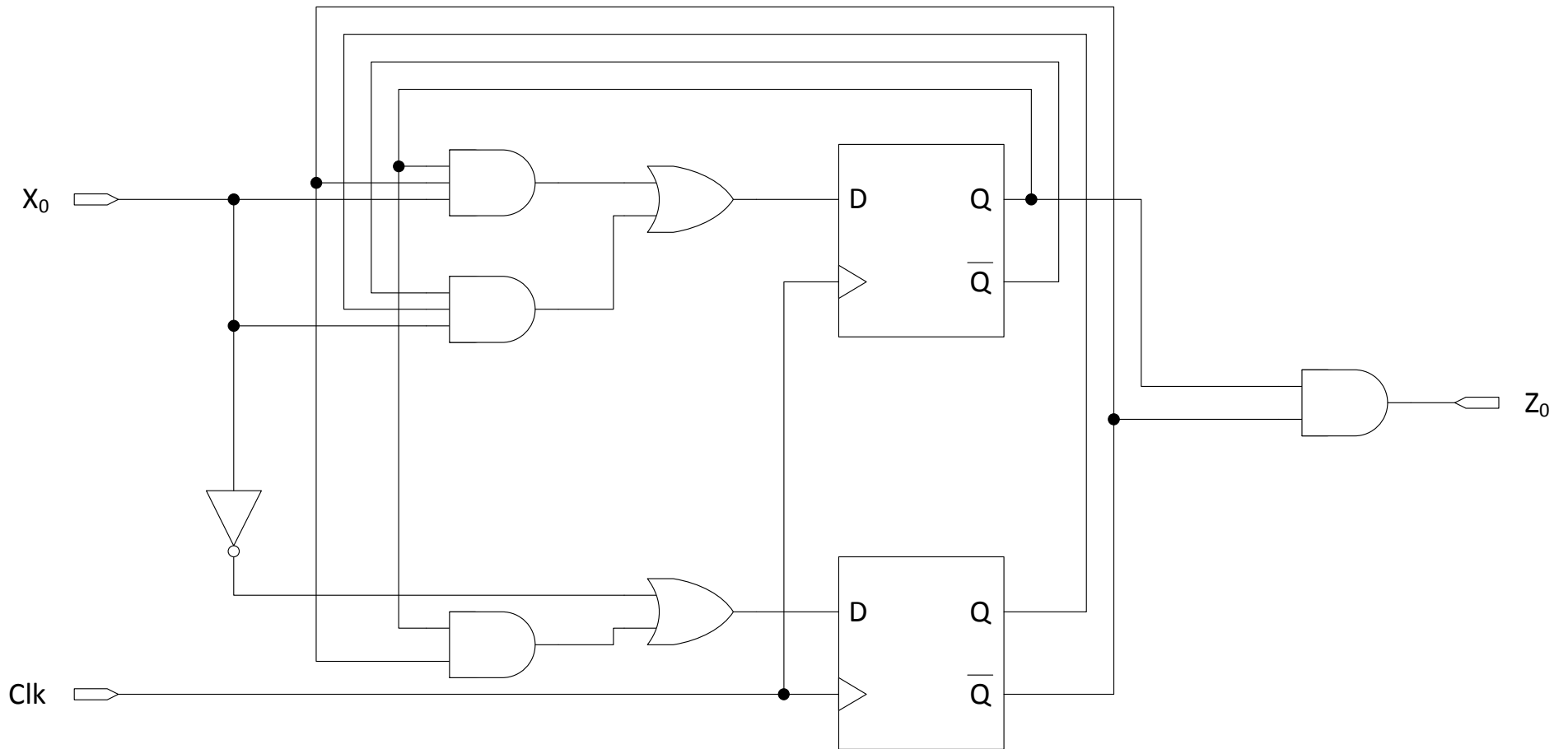
Estado actual	Entrada actual	
	a	b
S ₀	S ₁ /p	S ₀ /p
S ₁	S ₁ /p	S ₂ /p
S ₂	S ₁ /p	S ₀ /q
S ₃	S ₁ /p	S ₂ /q





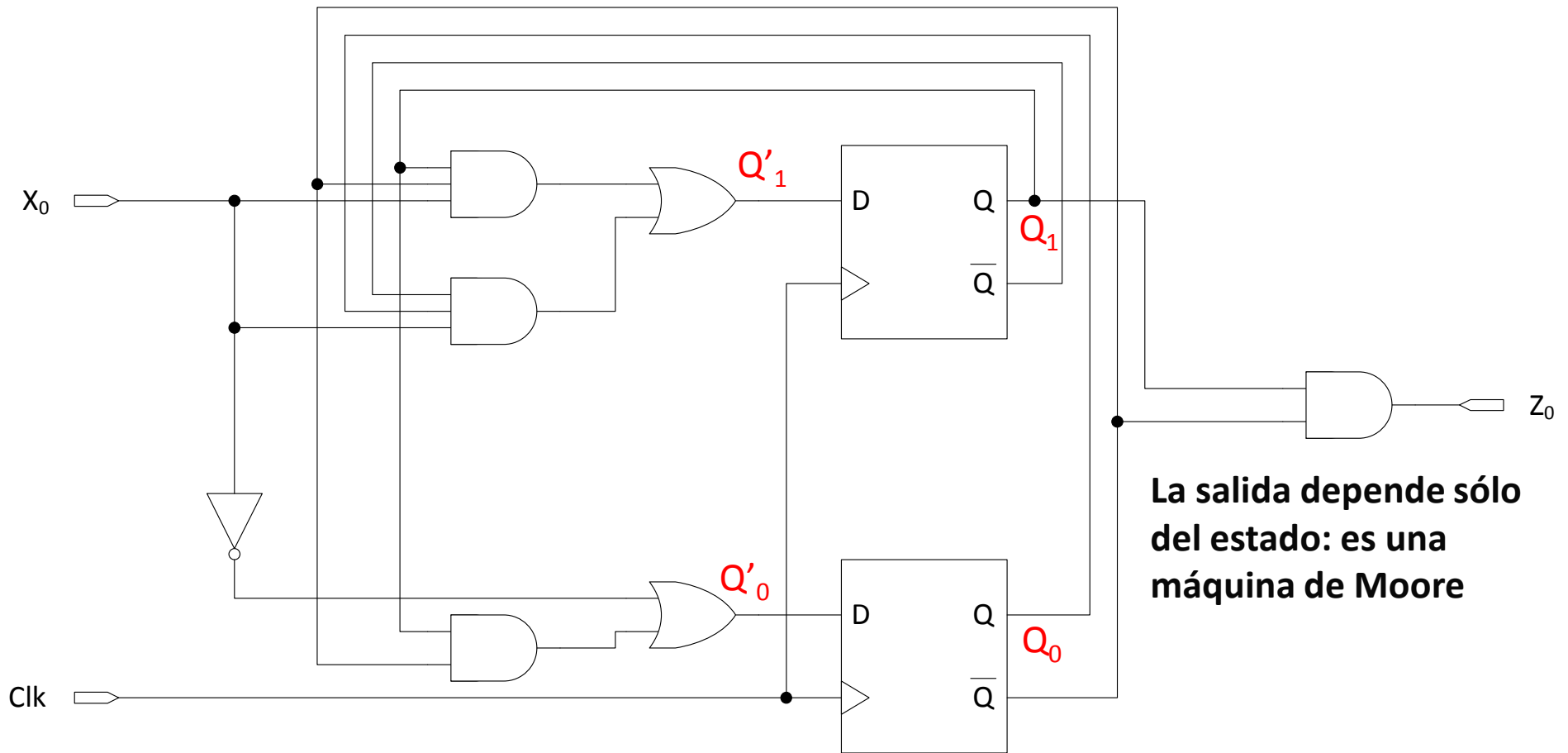
ANÁLISIS DE FSM: EJEMPLO 2

- Partimos de la siguiente FSM:





ANÁLISIS DE FSM: IDENTIFICACIÓN



- Entradas: $X_0 \in \{0,1\}$
- Salidas: $Z_0 \in \{0,1\}$
- Variables de estado: 2 $\{Q_1, Q_0\}$ darán lugar a 4 estados como máximo.



ANÁLISIS DE FSM: ECUACIONES Y TABLAS - MOORE

Función de transición

$$D_0 = Q'_0(Q_1, Q_0, X_0) = Q_1 \cdot \overline{Q_0} + \overline{X_0}$$

$$D_1 = Q'_1(Q_1, Q_0, X_0) = Q_1 \cdot \overline{Q_0} \cdot X_0 + \overline{Q_1} \cdot Q_0 \cdot X_0$$



S(t), X			S(t+1)	
Q ₁	Q ₀	X	Q' ₁	Q' ₀
0	0	0	0	1
0	0	1	0	0
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0



S(t)		X ₀	
Q ₁	Q ₀	0	1
0	0	01	00
0	1	01	10
1	0	01	11
1	1	01	00

Función de salida

$$Z_0(Q_1, Q_0) = Q_1 \cdot Q_0$$



La salida depende sólo del estado (Moore)



S(t)		Z
Q ₁	Q ₀	Z ₀
0	0	0
0	1	0
1	0	0
1	1	1



ANÁLISIS DE FSM: DECODIFICACIÓN Y REPRESENTACIÓN FORMAL - MOORE

- Asignamos nombres y valores a las entradas, los estados y la salida.

Tabla de estados

S(t)		X ₀	
Q ₁	Q ₀	0	1
0	0	01	00
0	1	01	10
1	0	01	11
1	1	01	00

Tabla de salida

S(t)		Z
Q ₁	Q ₀	Z ₀
0	0	0
0	1	0
1	0	0
1	1	1

Entrada

X ₀	X(t)
0	a
1	b

Salida

Z ₀	Z(t)
0	p
1	q

Estado

Q ₁	Q ₀	S(t)
0	0	S ₀
0	1	S ₁
1	0	S ₂
1	1	S ₃



Estado actual	Entradas	
	a	b
S ₀	S ₁	S ₀
S ₁	S ₁	S ₂
S ₂	S ₁	S ₃
S ₃	S ₁	S ₀



Estado	Salida
S ₀	p
S ₁	p
S ₂	p
S ₃	q



ANÁLISIS DE FSM: REPRESENTACIÓN FORMAL - MOORE

Estado actual	Entradas		Estado	Salida
	a	b		
S_0	S_1	S_0	S_0	p
S_1	S_1	S_2	S_1	p
S_2	S_1	S_3	S_2	p
S_3	S_1	S_0	S_3	q

