

Introducción al Diseño Lógico (E0301)

Ingeniería en Computación

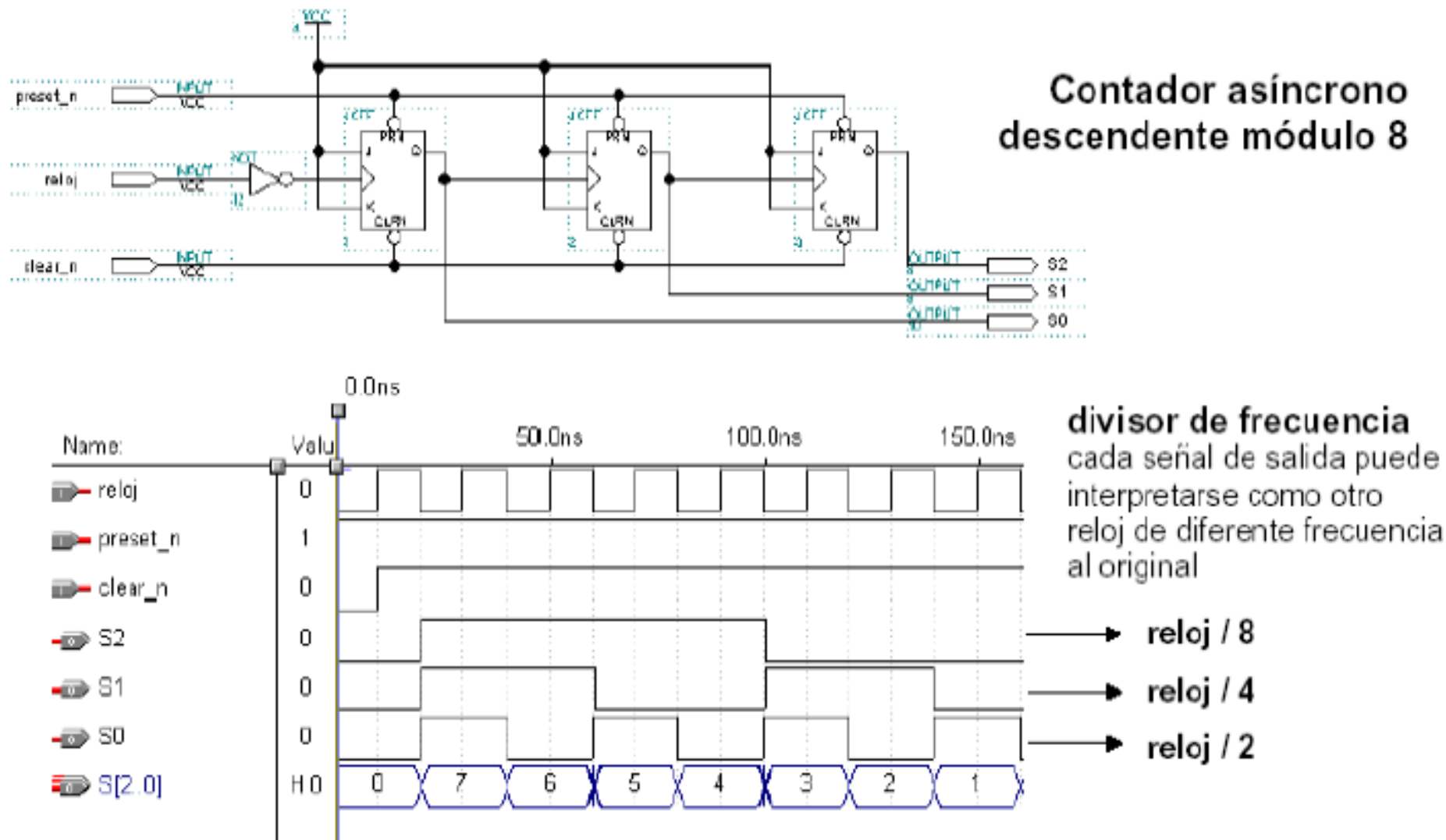
Gerardo E. Sager

Clase 7 curso 2025

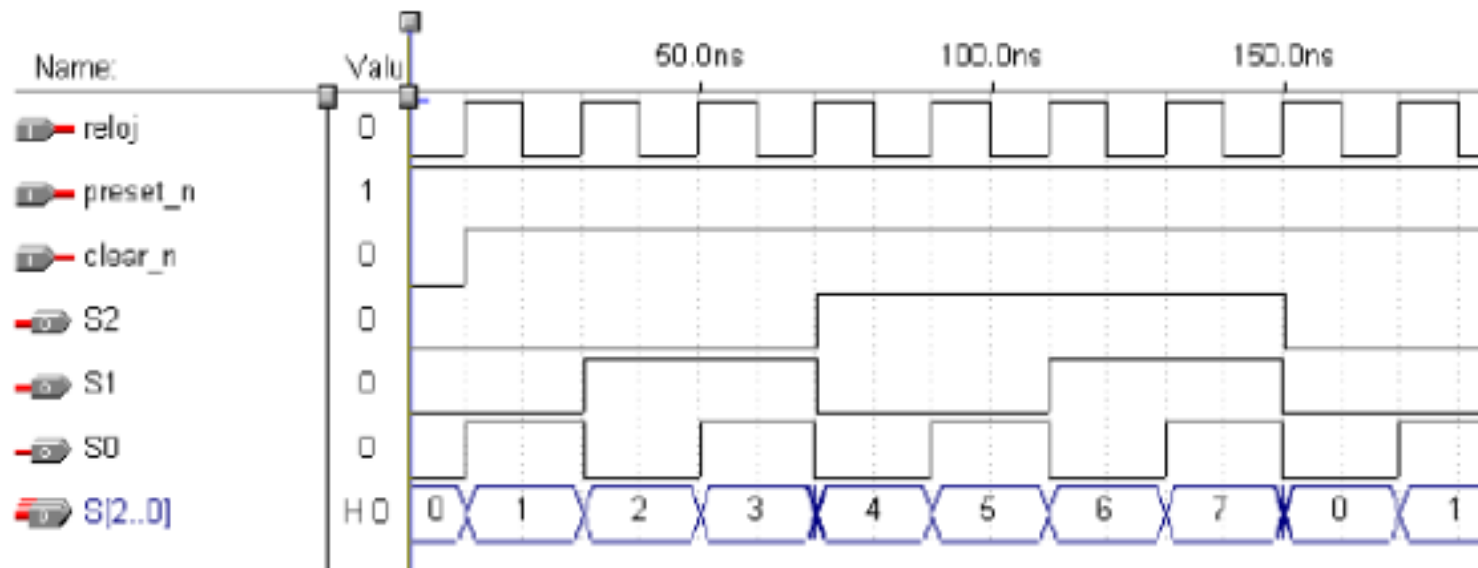
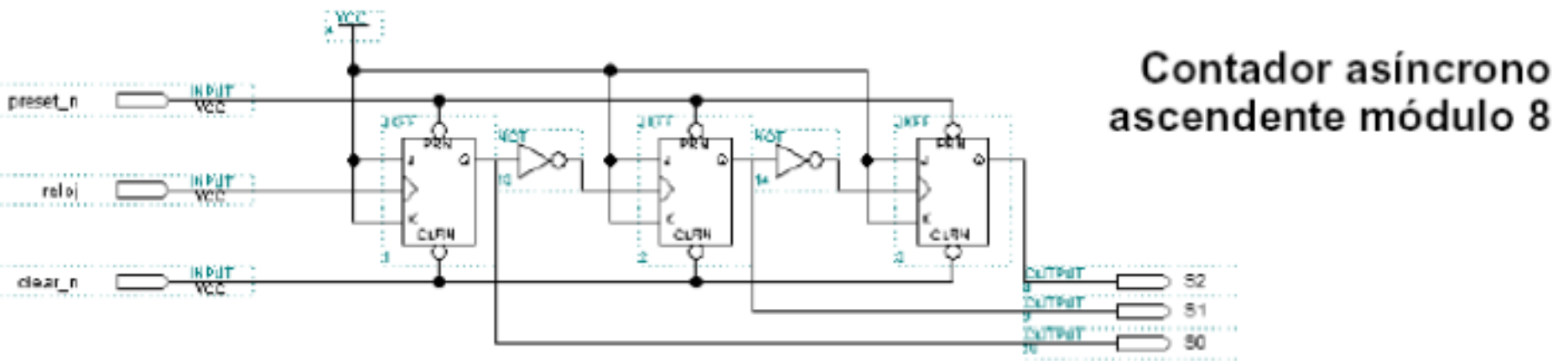
CONTADORES: Funcionamiento y utilidad

- Cuenta el número de impulsos que recibe en una de sus entradas, habitualmente la entrada de Reloj
- Aplicaciones
 - Divisores de frecuencia
 - Permiten obtener una frecuencia de salida $f_s = f_{in} / M$ donde f_{in} es la frecuencia de entrada y M es el número máximo hasta el que llega la cuenta
 - Control de tiempos
 - Permiten la medición o generación de intervalos de tiempo a partir de una entrada periódica llamada base de tiempo
 - Generación de Direcciones
 - Permiten generar números consecutivos que pueden utilizarse para acceder secuencialmente a direcciones de una memoria

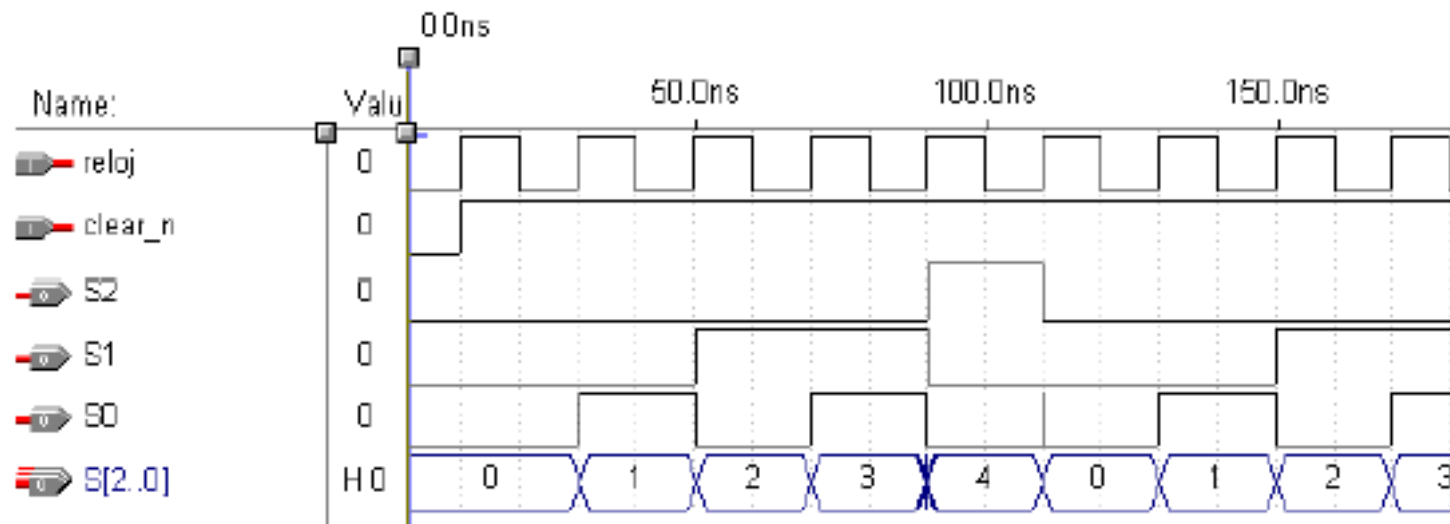
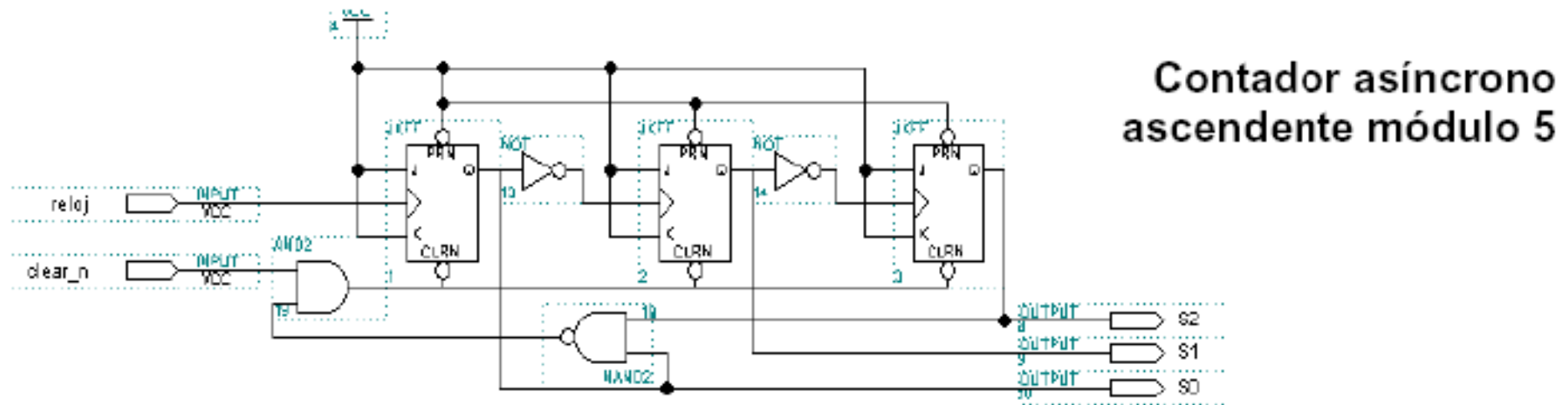
Contador Asincrónico I (Ripple Counter)



Contador Asincrónico II



Contador Asincrónico III

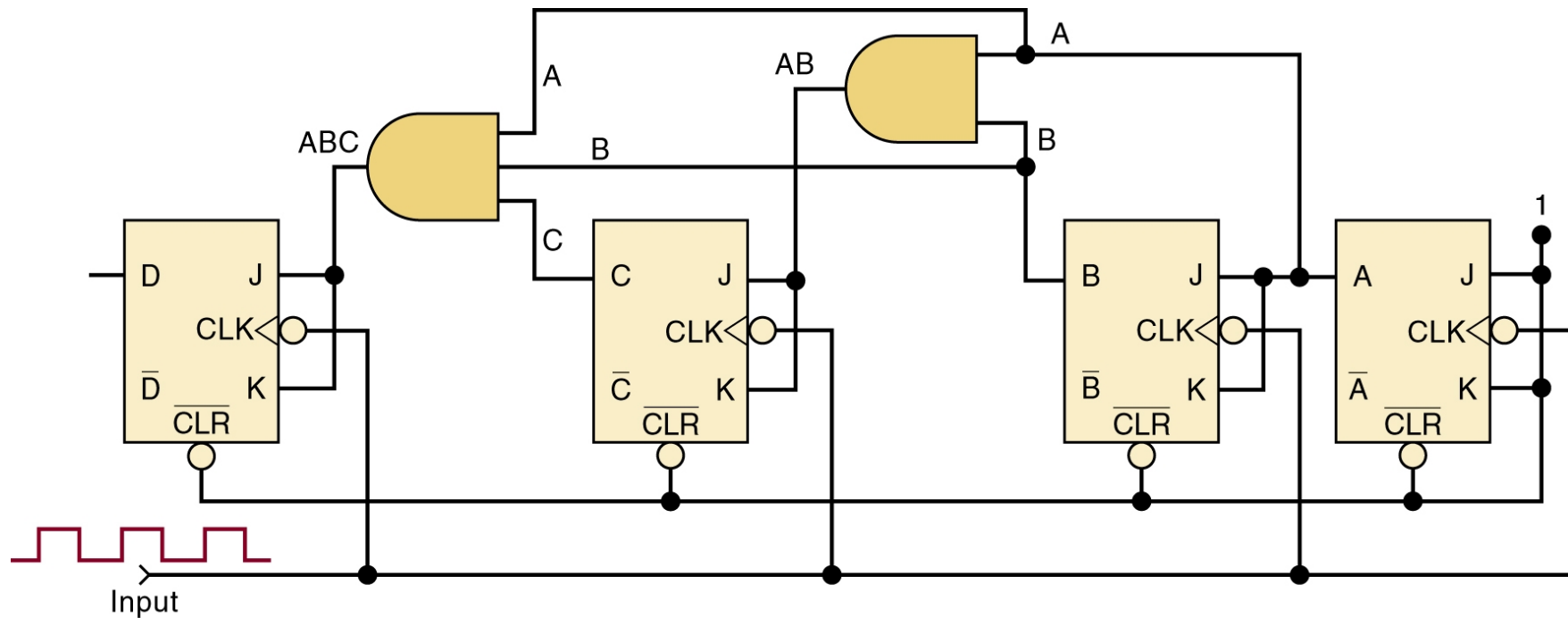


Contador Asincrónico IV

- Los Contadores tipo “Ripple” son sencillos y necesitan pocos componentes para producir una operación dada
 - Los retardos de propagación acumulados pueden crear problemas a altas frecuencias.
- Si el período entre pulsos de entrada es más largo que el tiempo de propagación total del contador, se evita este problema
 - Para que funcione correctamente: $T_{clock} \geq N t_{pd}$
 - Frecuencia Máxima: $f_{max} \leq 1/(N t_{pd})$

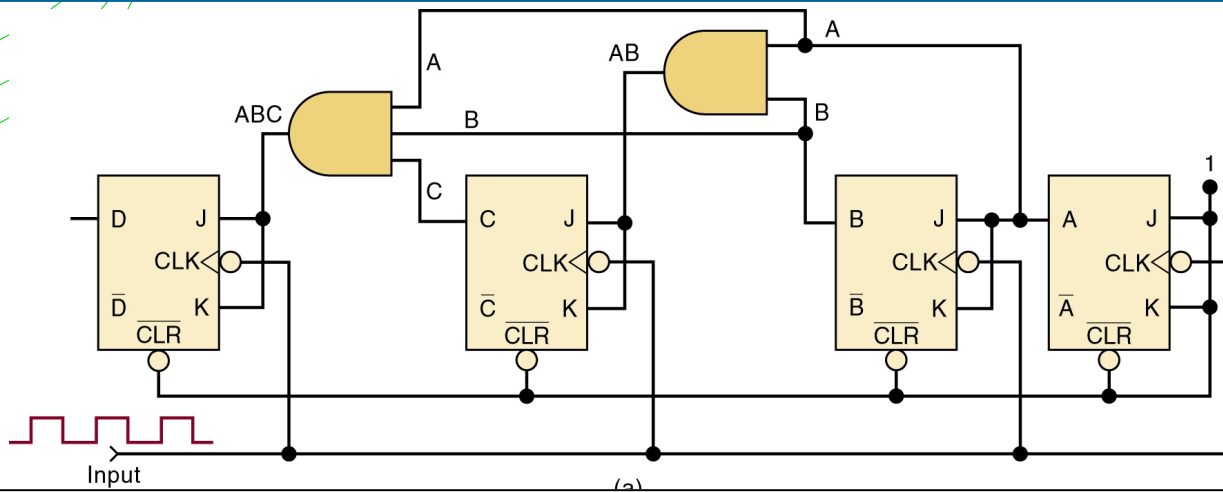
Contadores Sincrónicos I

- En contadores **síncronos** o **paralelos**, todos los FF son disparados simultáneamente por el reloj.



Los contadores síncronos pueden operar a frecuencias mucho más altas que los asíncronos.

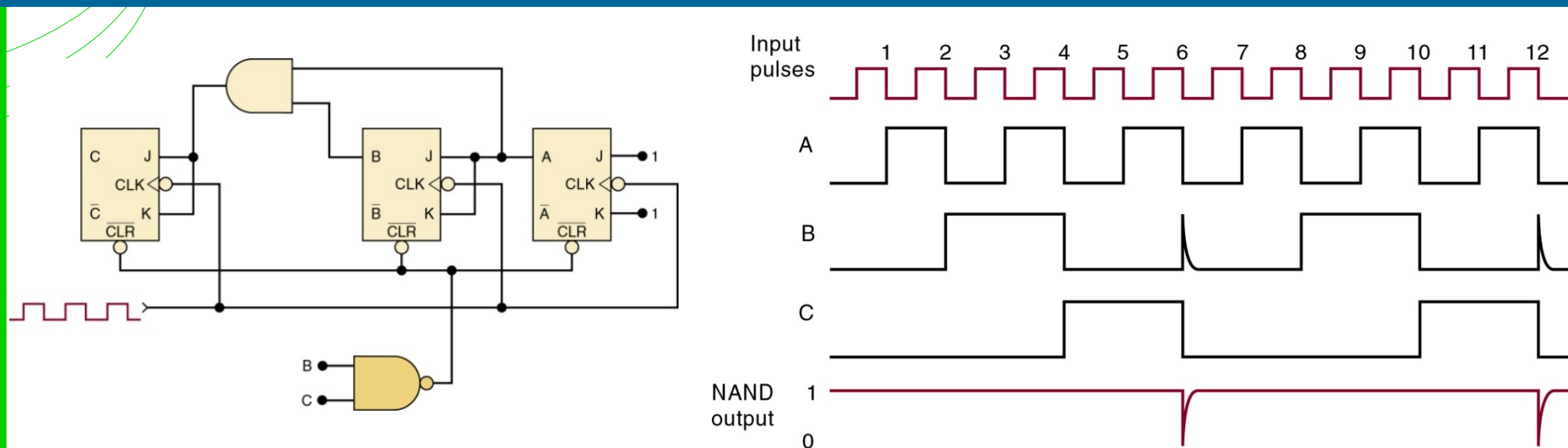
Contadores Sincrónicos II



- Cada FF tiene entradas J & K que se establecen en ALTO sólo cuando todas las salidas de orden inferior están en ALTO.
- Para que este circuito cuente correctamente, sólo aquellos FF que se supone que deben cambiar en una transición dada de reloj, deben tener un valor ALTO aplicado a sus entradas J y K
- El contador síncrono básico que se obtiene siguiendo las reglas vistas cuenta solamente MOD 2^N . N es el número de FFs.

Count		D	C	B	A
0		0	0	0	0
1		0	0	0	1
2		0	0	1	0
3		0	0	1	1
4		0	1	0	0
5		0	1	0	1
6		0	1	1	0
7		0	1	1	1
8		1	0	0	0
9		1	0	0	1
10		1	0	1	0
11		1	0	1	1
12		1	1	0	0
13		1	1	0	1
14		1	1	1	0
15		1	1	1	1
0		0	0	0	0
.	
.	
.		.	etc.	.	.

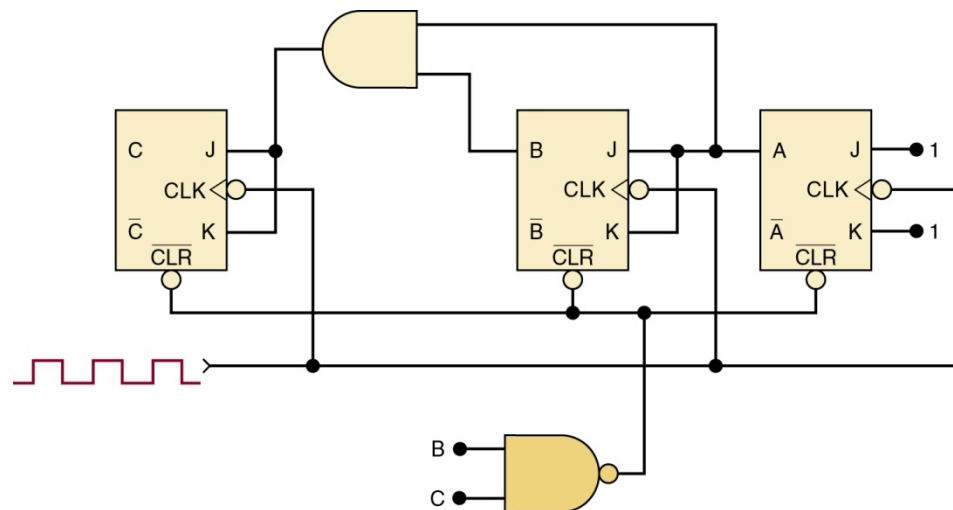
Contadores Sincrónicos III



- El contador síncrono básico que se ha visto, puede modificarse para contar MOD M donde M es menor que 2^N .
 - Esto se logra forzando a que el contador saltee estados, que serían normalmente parte de la secuencia de conteo.
 - En este ejemplo puede obtenerse un contador MOD-6 haciendo un clear de los FF de un contador MOD-8, cuando la cuenta llega a seis (110).

Contadores Sincrónicos IV

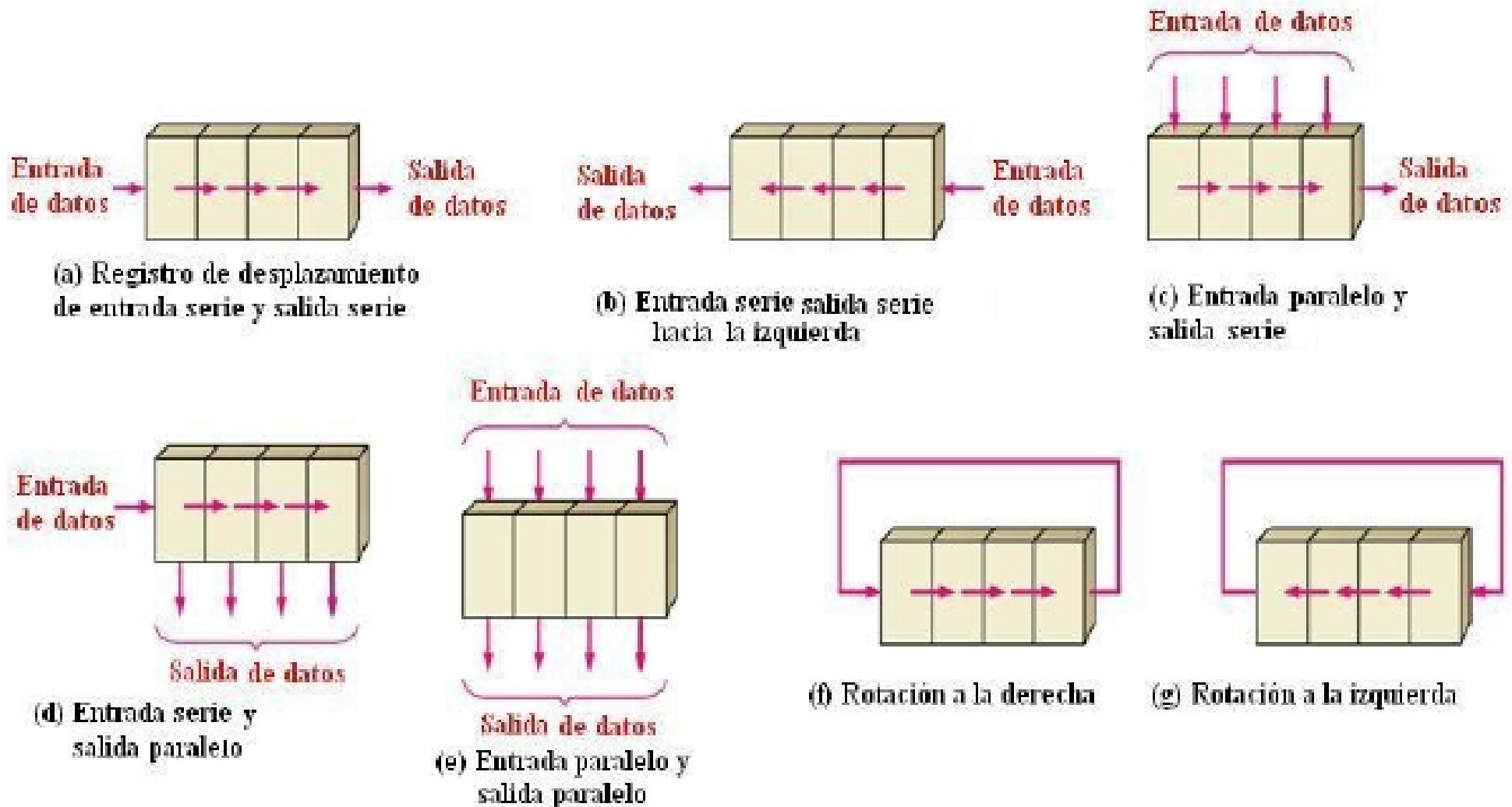
- Como cambiar el Módulo M del contador.
 - Encontrar el valor mínimo de N tal que 2^N sea mayor que el valor deseado M.
 - Conectar la salida de una compuerta NAND a la entrada Clear asincrónica de todos los FF.
 - Determinar cuáles FFs están en ALTO cuando se alcance la cuenta deseada y conectar las salidas (Q) de esos FF a las entradas de la compuerta NAND.



Registros

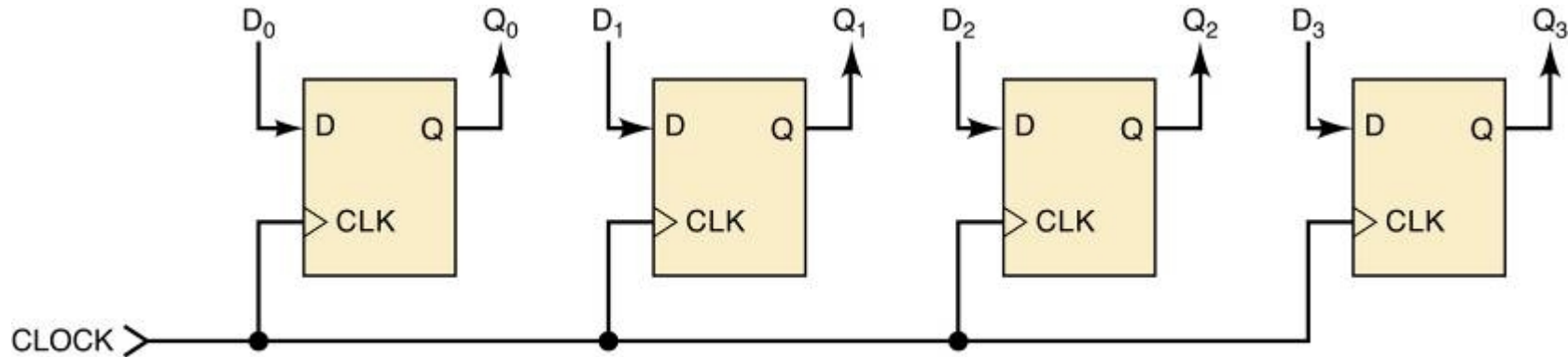
- Un registro es un circuito sincrónico que permite almacenar N bits utilizando N Flip Flops,
- Los Flip Flops que componen el registro comparten todas sus señales de control:
 - Preset, Clear, Clk, etc.
- Hay distintos tipos:
 - Segun el modo de entrada / salida: serie y paralelo
 - Registros de Desplazamiento: uni y bidireccionales.

Tipos de Registros

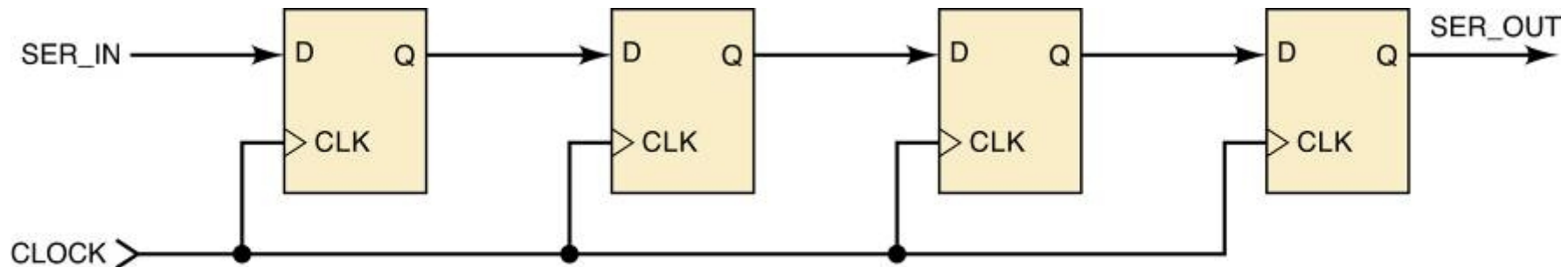


Tipos de Registros

Entrada Paralelo, Salida Paralelo (PIPO)

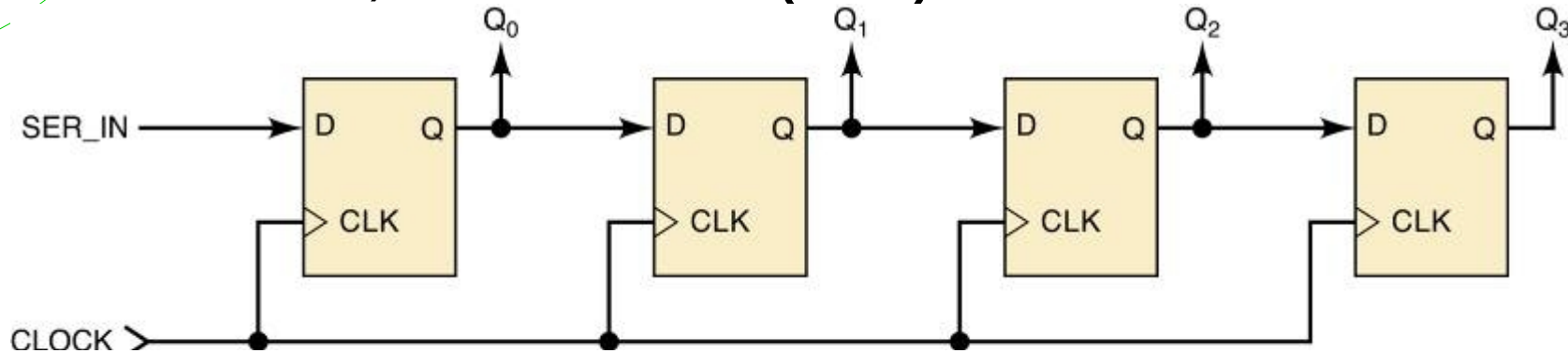


Entrada Serie, Salida Serie (SISO)

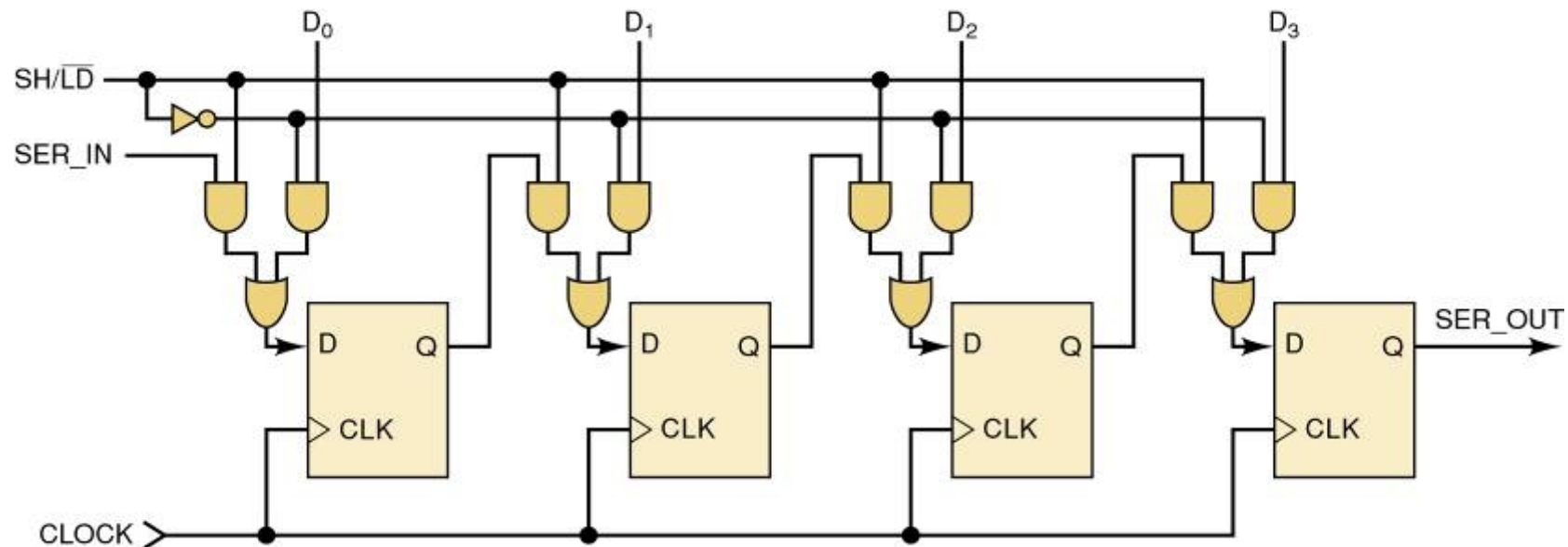


Tipos de Registros

Entrada Serie, Salida Paralelo (SIPO)



Entrada Paralelo / Serie, Salida Serie (PISO)

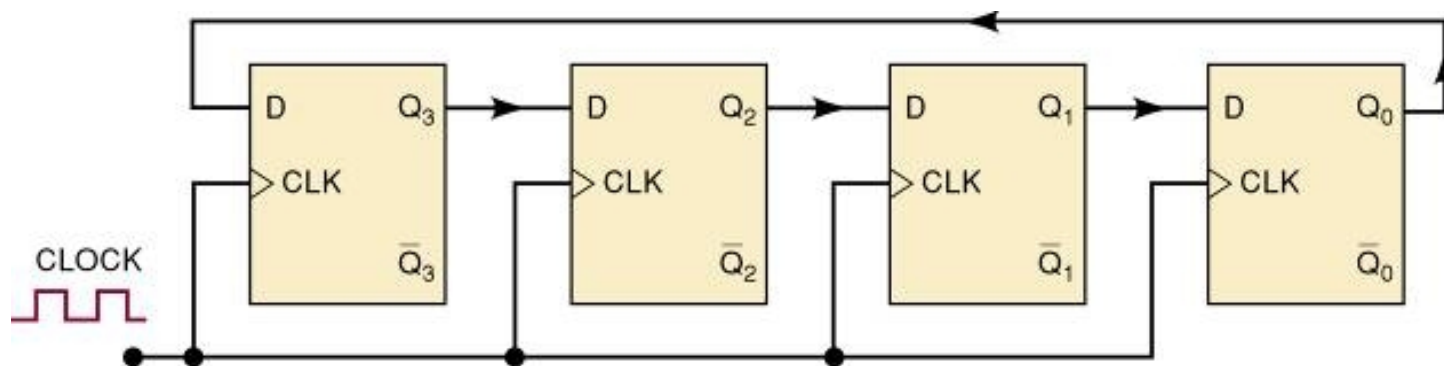


Contadores basados en registros

- Los contadores basados en registros de desplazamiento (Shift-Registers) usan *realimentación* o *feedback*, esto es, la salida del último FF en el registro está conectada de alguna manera, a la entrada del primer FF.

Contador de Anillo:

- Es un registro de desplazamiento circular, conectado de manera tal que la salida del último FF se conecta directamente a la entrada del primero.
- Necesita un circuito de arranque, que inyecte un 1 en un FF al inicio

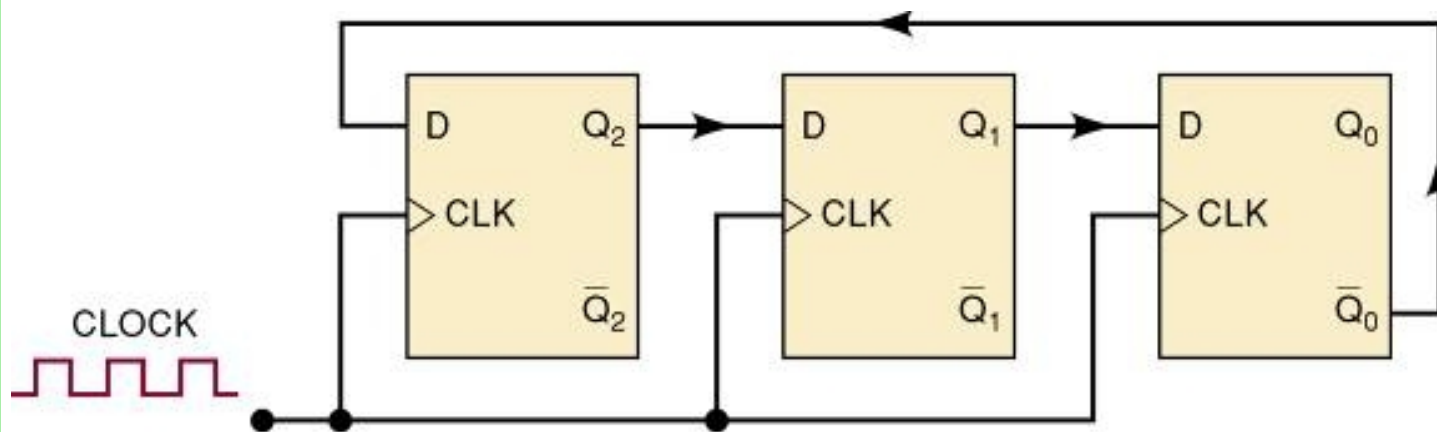


Q ₃	Q ₂	Q ₁	Q ₀	CLOCK pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
·	·	·	·	·
·	·	·	·	·

Contadores basados en registros

Contador Johnson o contador de doble anillo:

- Es un registro de desplazamiento circular, conectado de manera tal que la salida invertida (\bar{Q}) del último FF se conecta directamente a la entrada del primero.
- Si arranca en ciertos estados (010 o 101) se queda alternando entre estos dos valores.

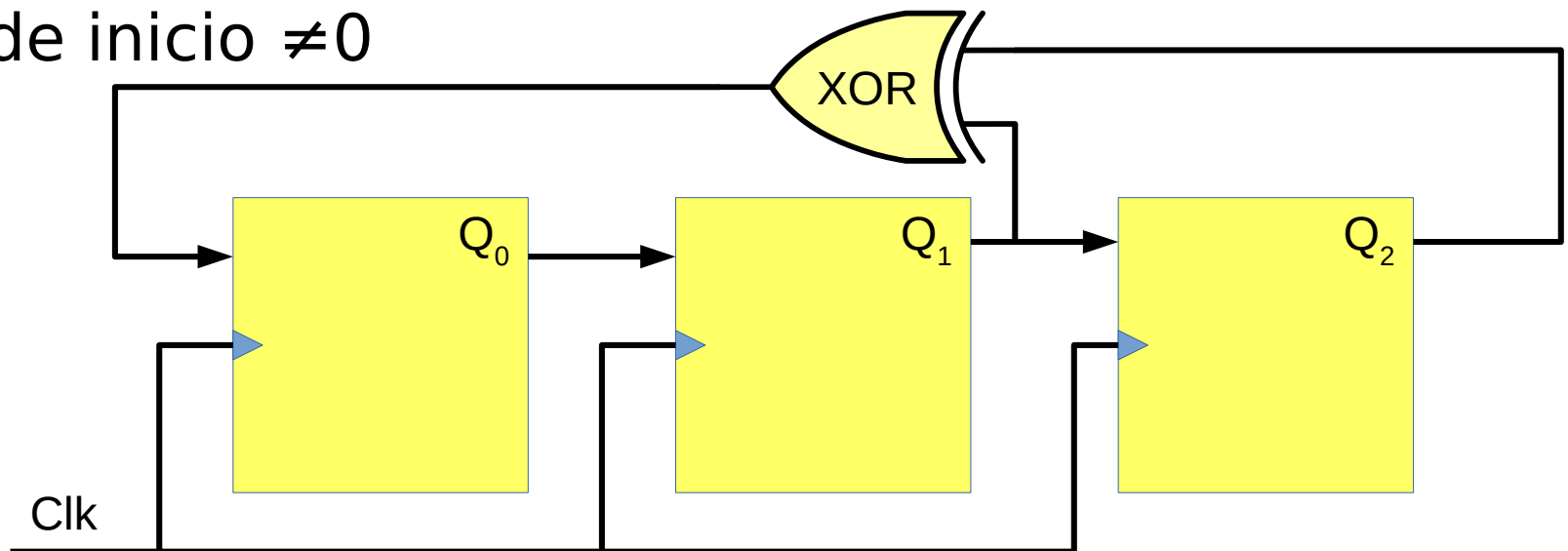


Q_2	Q_1	Q_0	CLOCK pulse
0	0	0	0
1	0	0	1
1	1	0	2
1	1	1	3
0	1	1	4
0	0	1	5
0	0	0	6
1	0	0	7
1	1	0	8

Contadores basados en registros

Generador de secuencias

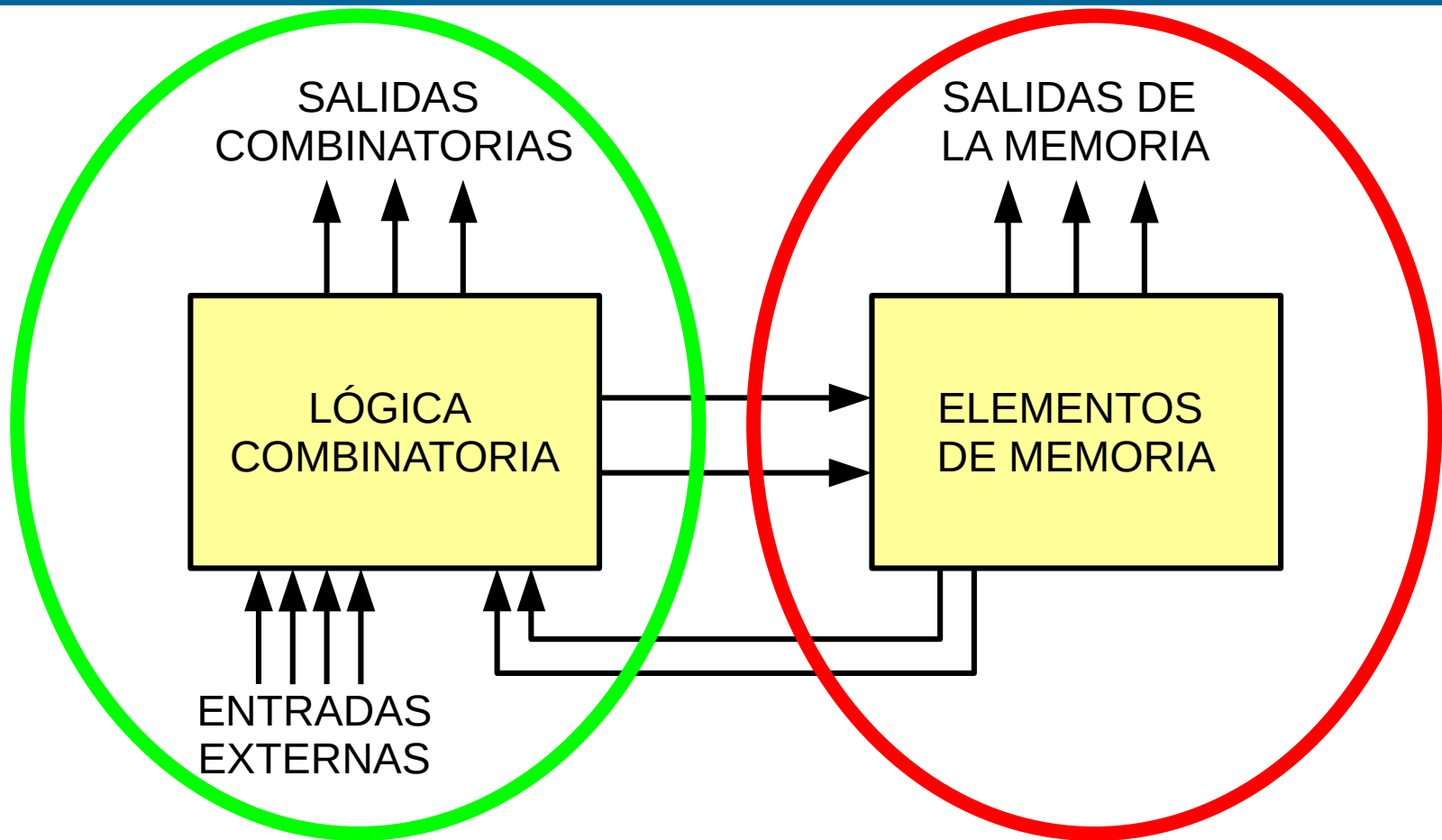
- Es un registro de desplazamiento circular, conectado de manera tal que la salida y al menos un FF intermedio, se combinen mediante una función lógica y el resultado se aplique a la entrada.
- Si cumplen ciertas condiciones, la secuencia generada se llama secuencia pseudo aleatoria. Se debe proveer un valor de inicio $\neq 0$



Circuitos Secuenciales Sincrónicos

- Temas a tratar
 - Introducción
 - Máquinas de Estado Finito
 - Modelo de Moore
 - Modelo de Mealy
 - Análisis de circuitos secuenciales sincrónicos
 - Síntesis de circuitos secuenciales sincrónicos
 - Ejemplos

Introducción

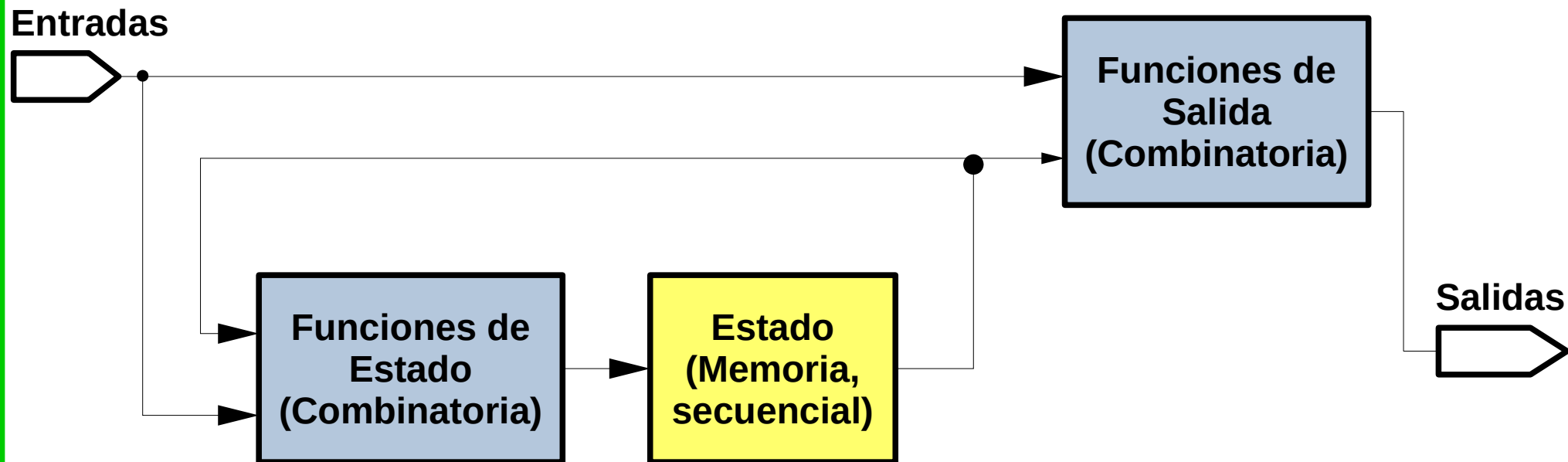


Hasta ahora vimos como analizar y diseñar lógica combinatoria

También vimos como analizar y diseñar elementos de memoria

Circuito secuencial síncrono

Esquema general de un circuito secuencial síncrono



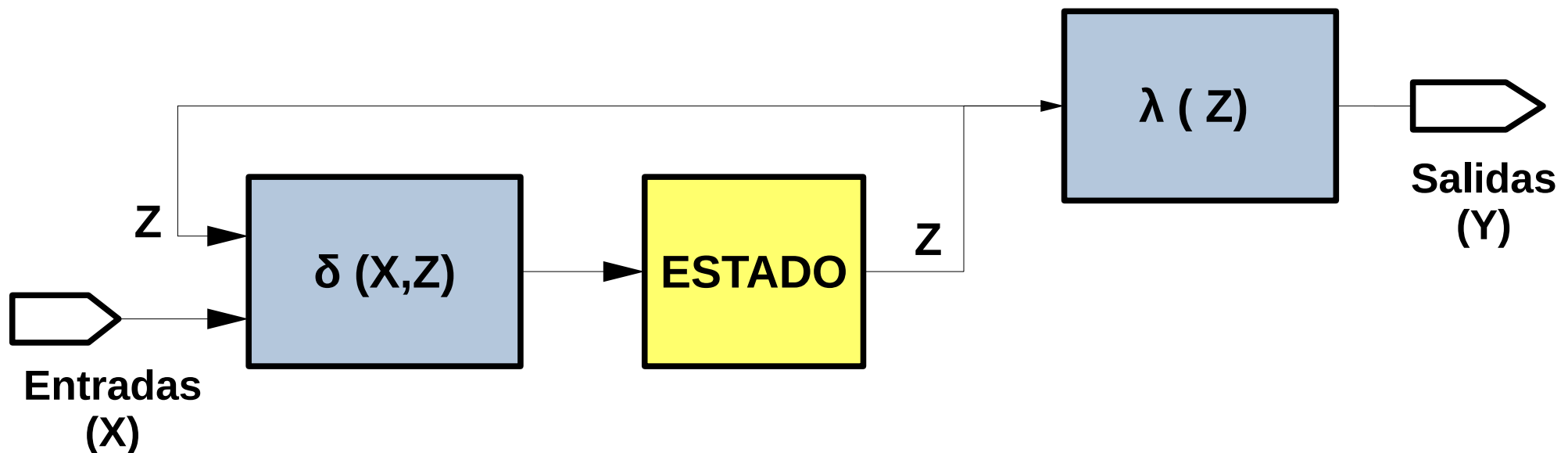
El bloque “ESTADO” está formado por biestables (Flip-Flops), todos ellos sincronizados por la misma señal de reloj.

Máquinas de Estados Finitos (FSM)

- El comportamiento de un circuito síncrono puede representarse mediante una Máquina de Estados Finitos (FSM o Finite State Machine)
- Una Máquina de Estados Finitos tiene los siguientes elementos
 - X : Entradas
 - Y : Salidas
 - Z : Estados (Valores de los FF, cambian con cada flanco de reloj)
 - δ : Funciones de Estado (Funciones combinatorias de entrada de los FF)
 - λ : Funciones de Salida (Combinacionales)
- Una FSM se define como una serie de eventos en tiempos discretos. El estado Z cambia en cada evento y el cambio está definido por δ

Modelo de Moore

- En el modelo de Moore las salidas dependen únicamente de los estados (NO de las entradas)
- Formalmente podemos describir una máquina de Moore como:
 - $Z = \delta (X, Z)$
 - $Y = \lambda (Z)$

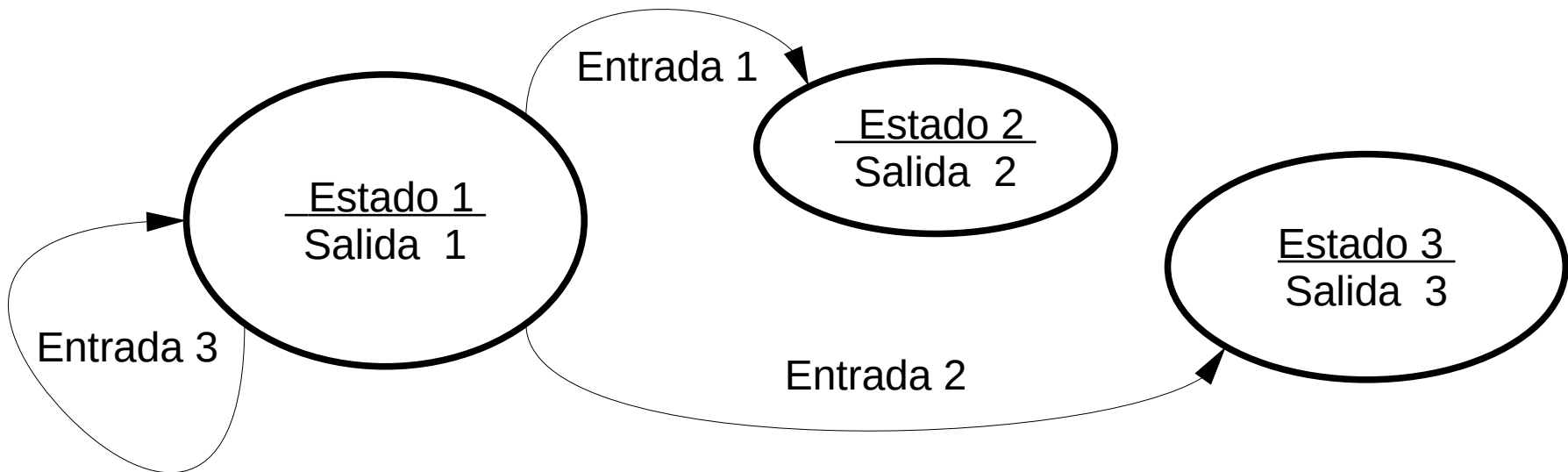


Modelo de Moore

- El reloj y el reset no aparecen en las máquinas de estados, la asociación entre estas señales en un circuito y la máquina de estados es:
 - En cada flanco del reloj, se produce una transición o cambio de estado
 - El reset se utiliza únicamente para establecer el estado inicial
- En las máquinas de estados de Moore las salidas cambian únicamente si hay un cambio de estado
 - Las salidas están sincronizadas con el reloj

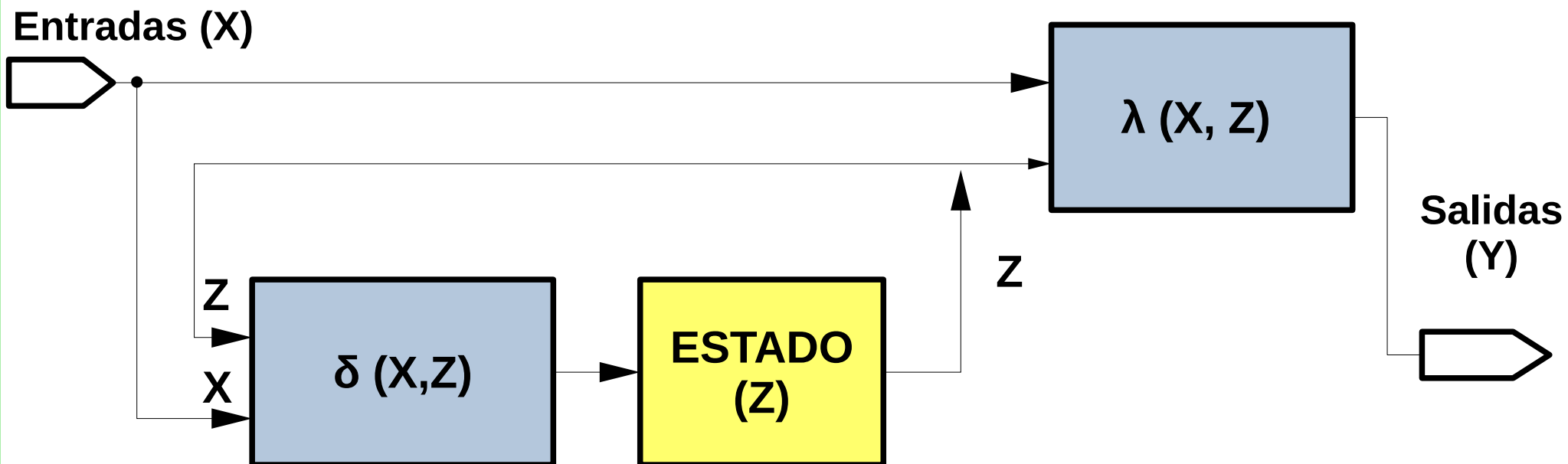
Modelo de Moore

- Una FSM se puede también representar mediante un diagrama de estados (STG o “State Transition Graph”)
 - Cada estado se representa con un círculo
 - Cada transición de estado se representa con una flecha
 - Los diferentes valores de las entradas se representan en las flechas
 - En el caso del modelo de Moore, las salidas se representan dentro de cada estado
- Diagrama de estados (Moore)



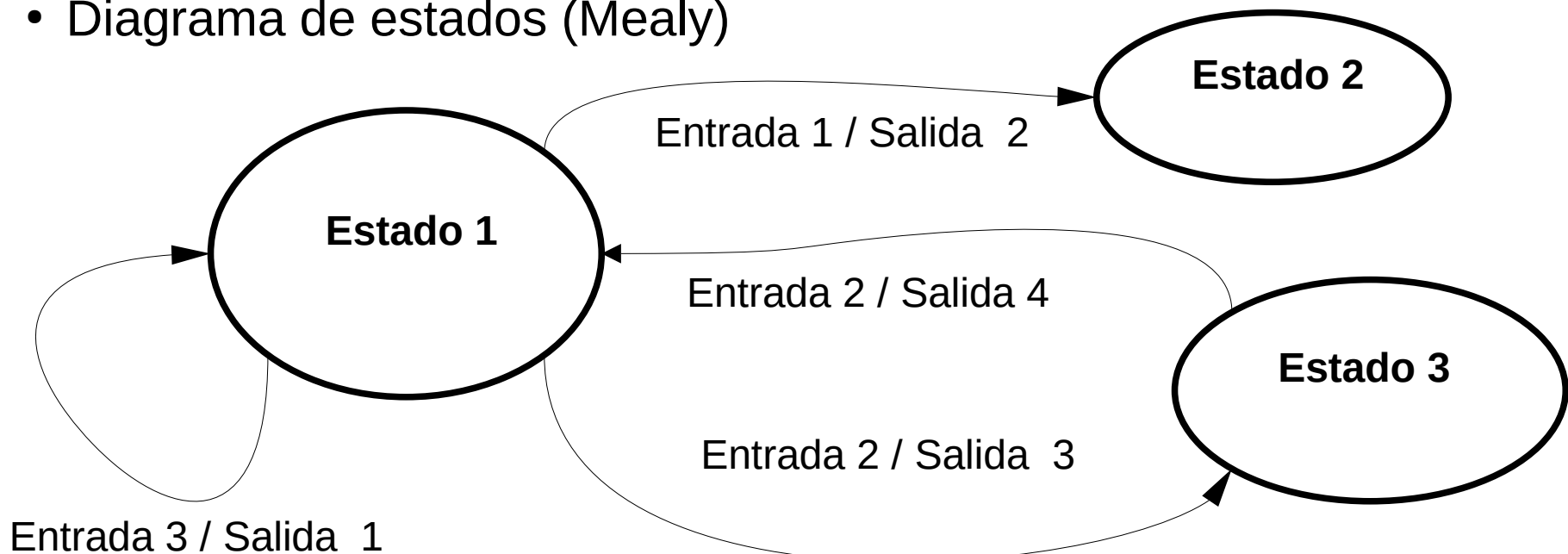
Modelo de Mealy

- En el modelo de Mealy las salidas dependen tanto de las entradas como de los estados (Caso general)
- Formalmente podemos describir una máquina de Mealy como:
 - $Z = \delta (X, Z)$
 - $Y = \lambda (X, Z)$



Modelo de Mealy

- Como vimos antes, una FSM se puede representar mediante un diagrama de estados (STG o “State Transition Graph”)
 - Cada estado se representa con un círculo y cada transición de estado se representa con una flecha como antes.
 - En el caso del modelo de Mealy, las entradas se representan en las flechas junto con las salidas, ya que cuando cambien las entradas, también cambiará la salida
- Diagrama de estados (Mealy)



Modelo de Mealy

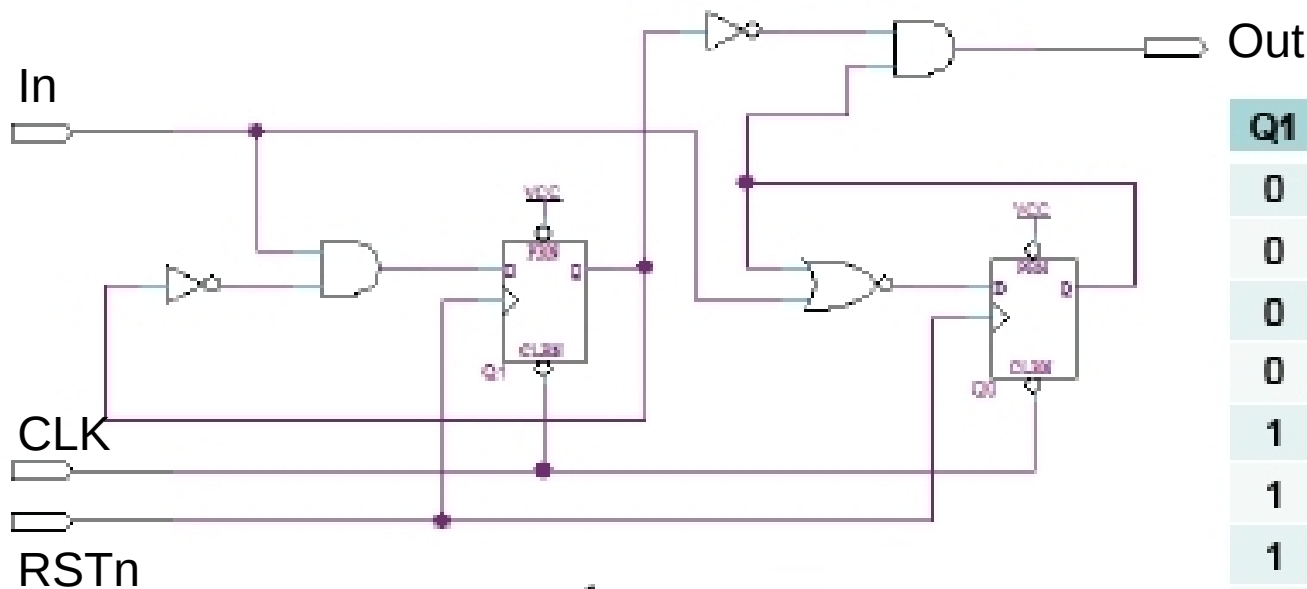
- Igual que en Moore, el reloj y el reset no aparecen en las máquinas de estados,
- En las máquinas de estados de Mealy las salidas pueden cambiar en cualquier momento, (alcanza con que cambie una de las entradas)
 - Las salidas NO están sincronizadas con el reloj
 - De todas maneras el circuito sigue siendo sincrónico, ya que los biestables estan todos sincronizados con el mismo reloj.
 - Los estados SI están sincronizados con el reloj

Análisis y Síntesis de Circuitos Secuenciales Sincrónicos

- **Análisis:** A partir de un circuito, obtener su funcionalidad
 - Circuitos Combinacionales:
 - Obtener tablas de verdad o funciones booleanas
 - Circuitos Secuenciales:
 - Obtener diagrama de estados, o funciones de estado y de salida (δ y λ)
- **Síntesis:** Dada una funcionalidad, obtener la implementación de un circuito
 - Circuitos Combinacionales:
 - Obtener expresiones booleanas, implementar con puertas lógicas, multiplexores, decodificadores, etc.
 - Circuitos Secuenciales:
 - Obtener diagrama de estados e implementar las funciones de estado y de salida (δ y λ) con puertas lógicas, multiplexores, decodificadores y Biestables.

Análisis de Circuitos Secuenciales Sincrónicos

- **Análisis:** Obtener tabla de transiciones, calcular $(\delta$ y $\lambda)$ y obtener diagrama de estados
- **Ejemplo:**



$$\bar{\mathbf{D}} \Rightarrow \begin{cases} D_0 = \overline{Q_0} + In \\ D_1 = \overline{Q_1} \cdot In \end{cases}$$

$$\lambda \rightarrow Q_{int} = \overline{Q}_1 \cdot Q_2$$

Tabla de transiciones:

Q1	Q0	In	D1	D0	Q1+	Q0+	Out
0	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	1	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Análisis de Circuitos Secuenciales Sincrónicos

Tabla de transiciones:

Q1	Q0	In	D1	D0	Q1+	Q0+	Out
0	0	0	0	1	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	0	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	1	0
1	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

Diagrama estados (Mealy):

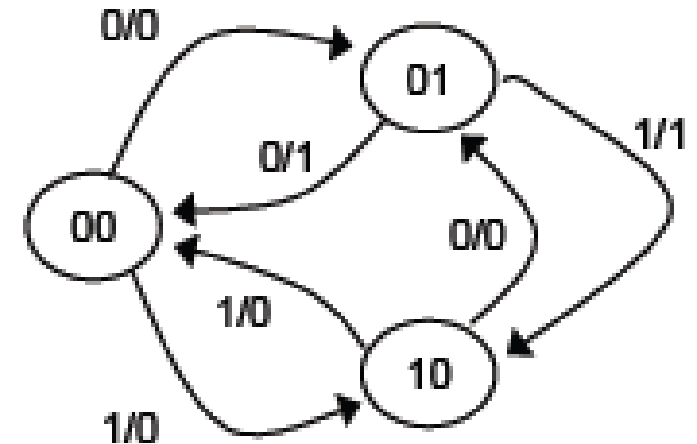
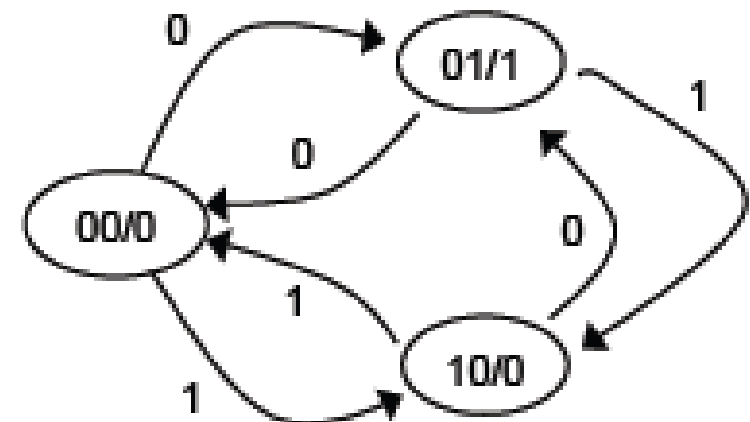


Diagrama estados (Moore):



Síntesis de Circuitos Secuenciales Sincrónicos

- A partir de la descripción de la funcionalidad de un circuito secuencial. los pasos a seguir para obtener una implementación son:
 - 1) Obtener diagrama de Estados
 - 2) Codificación de los Estados
 - 3) Obtener tablas de salidas y de transiciones de estados
 - 4) Obtener tabla de excitación de los Biestables, también se la llama tabla inversa
 - 5) Obtener funciones de salida
 - 6) Obtener funciones de estado
 - 7) Implementar con FF y Lógica/ MuX /Decodif.
- La diferencia entre Moore y Mealy está en las funciones de salida

Tablas de excitación o tablas inversas de los Biestables

- Tablas de excitación o Tablas inversas
 - Describen todas las posibles combinaciones de entradas que permiten pasar del estado actual Q al estado siguiente Q^+

R-S latch

Q	Q ⁺	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

J-K flip-flop

Q	Q ⁺	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

D flip-flop

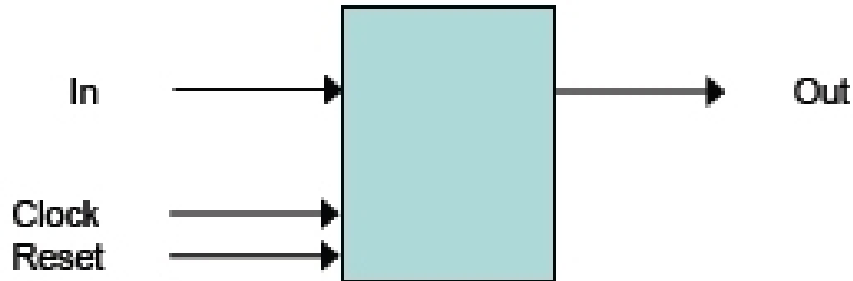
Q	Q ⁺	D
0	0	0
0	1	1
1	0	0
1	1	1

T flip-flop

Q	Q ⁺	T
0	0	0
0	1	1
1	0	1
1	1	0

Síntesis de Circuitos Secuenciales Sincrónicos

- Problema: Diseñar un circuito secuencial síncrono que permita detectar una secuencia de tres o más “unos” consecutivos a través de una entrada serie
 - La entrada se lee en cada flanco ascendente del reloj
 - La salida se activa cuando se detecta la secuencia

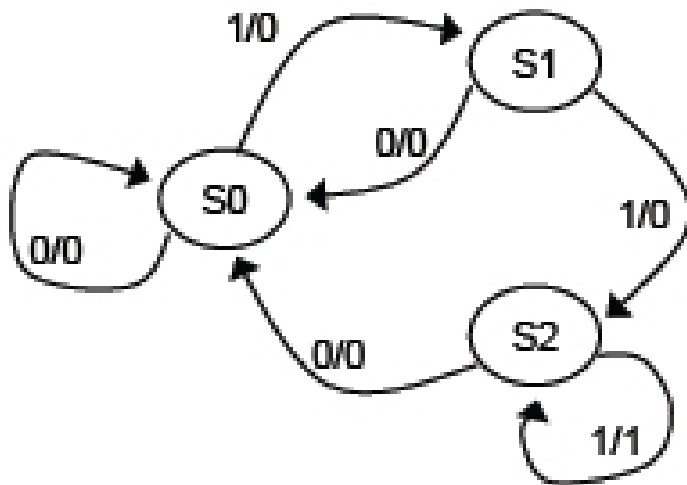


- Ejemplo de secuencia de entradas y de salidas
- X: 0 0 1 1 0 1 1 1 1 1 0 0 1 1 1
- Z: 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 1: Mealy con FF tipo D

1. Diagrama de estados:



2. Codificación de estados:

Estado	Q1	Q0
S0	0	0
S1	0	1
S2	1	1
	1	0

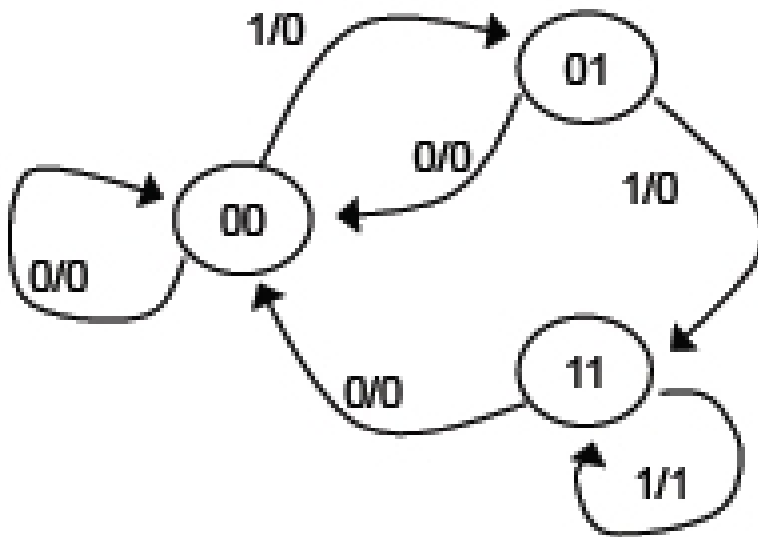


Estado no alcanzable

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 1: Mealy con FF tipo D

3. Tabla de transiciones y tabla de salidas (combinadas juntas):



In	Q1	Q0	Q1+	Q0+	Out
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	X	X	X
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	X	X	X
1	1	1	1	1	1

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 1: Mealy con FF tipo D

4. Tabla inversa de biestables (biestables D):

In	Q1	Q0	Q1+	Q0+	Out	D1	D0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	X	X	X	X	X
0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	1
1	1	0	X	X	X	X	X
1	1	1	1	1	1	1	1

5. Función de salida:

In	Q1 Q0			
	00	01	11	10
0				X
1			1	X

$$Out = Q_1 In$$

6. Funciones de estado

In	Q1 Q0			
	00	01	11	10
0				X
1		1	1	X

$$D_1 = Q_0 In$$

In	Q1 Q0			
	00	01	11	10
0				X
1	1	1	1	X

$$D_0 = In$$

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 2: Mealy con FF tipo JK

4. Tabla inversa de biestables (biestables J-K):

In	Q1	Q0	Q1+	Q0+	Out	J1	K1	J0	K0
0	0	0	0	0	0	0	X	0	X
0	0	1	0	0	0	0	X	X	1
0	1	0	X	X	X	X	X	X	X
0	1	1	0	0	0	X	1	X	1
1	0	0	0	1	0	0	X	1	X
1	0	1	1	1	0	1	X	X	0
1	1	0	X	X	X	X	X	X	X
1	1	1	1	1	1	X	0	X	0

5. Función de salida:

$$Out = Q_1 In$$

6. Funciones de estado

In	Q1 Q0			
	00	01	11	10
0			X	X
1		1	X	X

$$J_1 = Q_0 In$$

In	Q1 Q0			
	00	01	11	10
0		X	X	X
1	1	X	X	X

$$J_0 = In$$

In	Q1 Q0			
	00	01	11	10
0	X	X	1	X
1	X	X		X

$$K_1 = \overline{In}$$

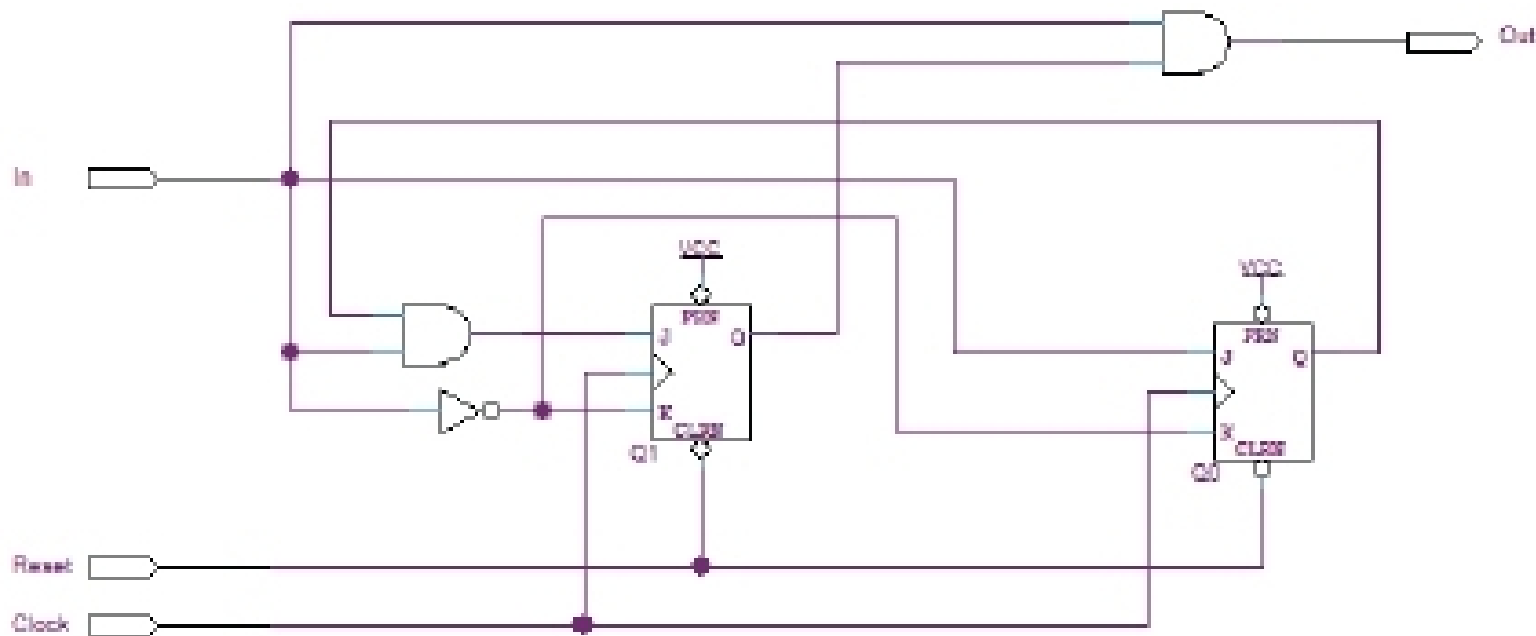
In	Q1 Q0			
	00	01	11	10
0	X	1	1	X
1	X			X

$$K_0 = \overline{In}$$

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 2: Mealy con FF tipo JK

7. Implementación



$$Out = Q_1 In$$

$$J_0 = In$$

$$K_0 = \overline{In}$$

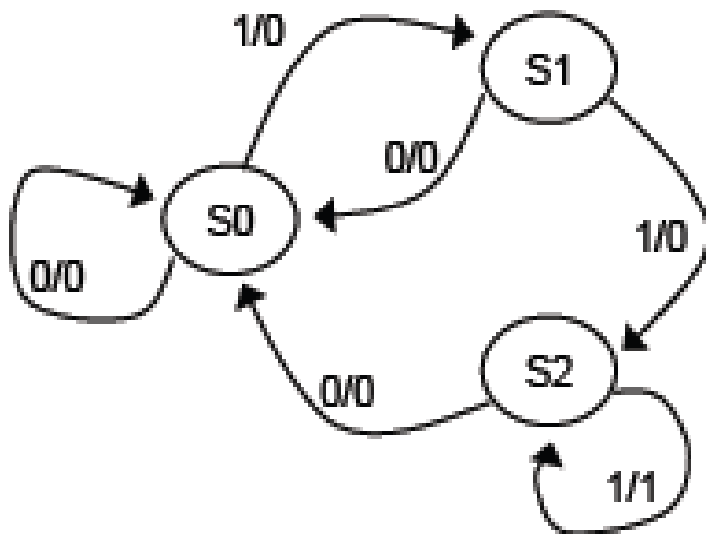
$$J_1 = Q_0 In$$

$$K_1 = \overline{In}$$

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 3: Mealy con una codificación diferente

1. Diagrama de estados:



2. Codificación de estados:

Estado	Q1	Q0
S0	0	0
S1	0	1
S2	1	0
	1	1

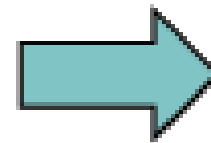
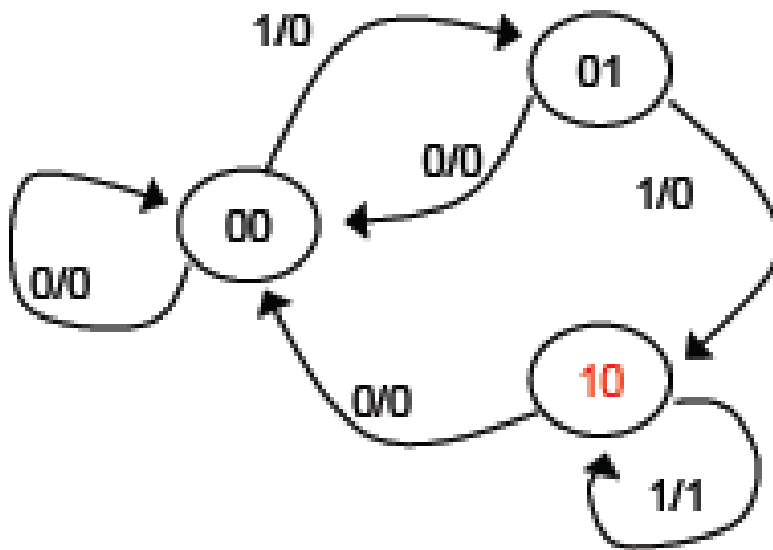


Ahora codificamos S2 de forma diferente

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 3: Mealy con una codificación diferente

3. Tablas de transiciones y salidas (combinadas en una sola):



In	Q1	Q0	Q1+	Q0+	Out
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	X	X	X
1	0	0	0	1	0
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	X	X	X

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 3: Mealy con una codificación diferente y FF tipo D

4. Tabla inversa de biestables (biestables D):

In	Q1	Q0	Q1+	Q0+	Out	D1	D0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0
0	1	1	X	X	X	X	X
1	0	0	0	1	0	0	1
1	0	1	1	0	0	1	0
1	1	0	1	0	1	1	0
1	1	1	X	X	X	X	X

5. Función de salida:

In	Q1 Q0			
	00	01	11	10
0			X	
1			X	1

$$Out = Q_1 In$$

6. Funciones de estado

In	Q1 Q0			
	00	01	11	10
0			X	
1		1	X	1

$$D_1 = Q_0 In + Q_1 In = In(Q_0 + Q_1)$$

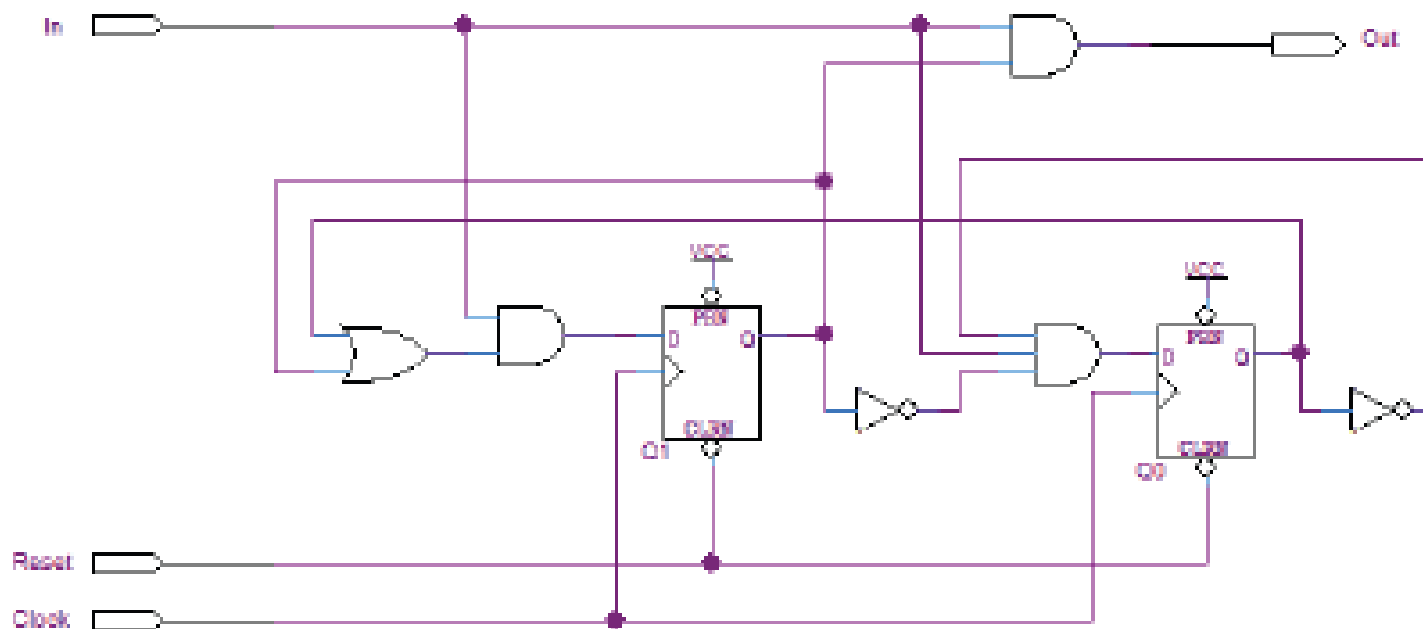
In	Q1 Q0			
	00	01	11	10
0			X	
1	1		X	

$$D_0 = \overline{Q_1} \overline{Q_0} In$$

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 3: Mealy con una codificación diferente y FF tipo D

7. Implementación

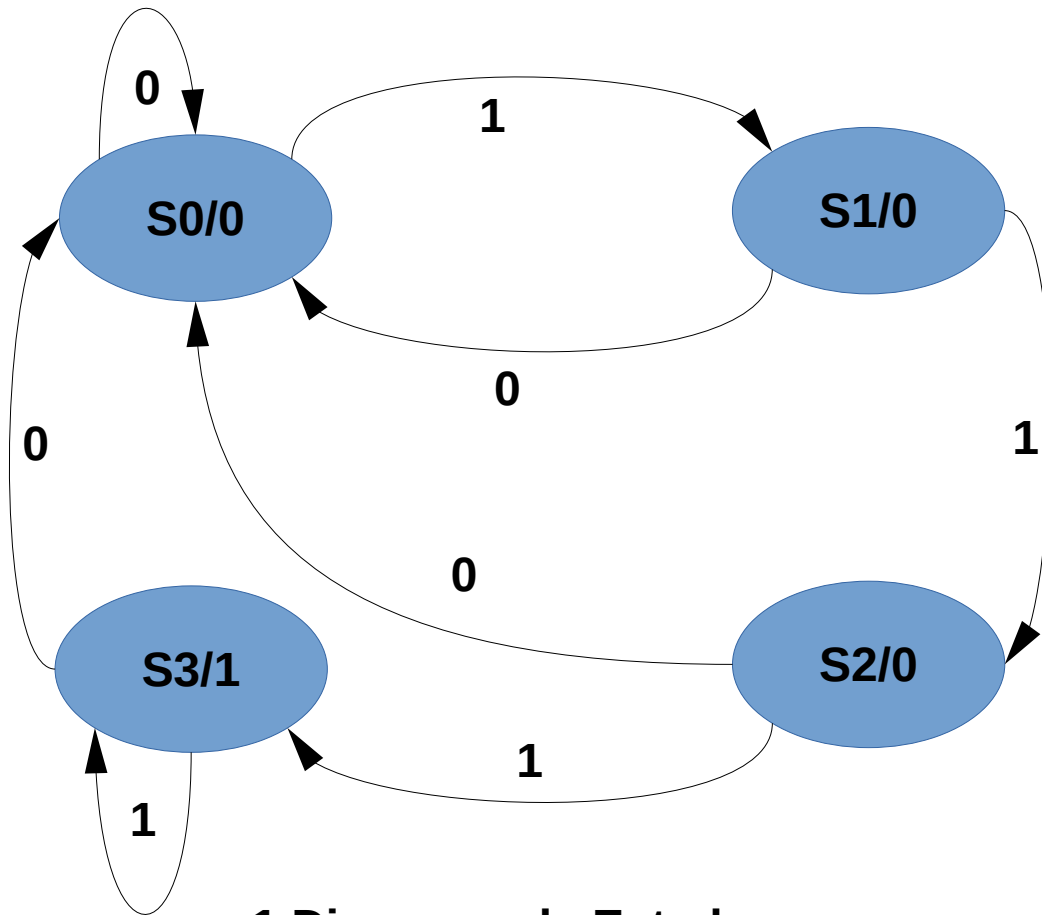


Con esta otra codificación sale más complejo y se requieren más puertas lógicas para la implementación

$$Out = Q_1 In$$
$$D_1 = In(Q_0 + Q_1)$$
$$D_0 = \overline{Q_1} Q_0 In$$

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 4: Moore con biestables tipo D



1. Diagrama de Estados

Q1	Q0	ESTADO
0	0	S0
0	1	S1
1	0	S3
1	1	S2

2. Codificación de Estados

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 4: Moore con biestables tipo D

In	Q1	Q0	Q1+	Q0+
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	1	0
1	1	1	1	0

3. Tabla de Transiciones

Q1	Q0	OUT
0	0	0
0	1	0
1	0	1
1	1	0

3. Tabla de Salidas

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 4: Moore con biestables tipo D

4. Tabla inversa de biestables (biestables D):

In	Q1	Q0	Q1+	Q0+	D1	D0
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	1	0	1
1	0	1	1	1	1	1
1	1	0	1	0	1	0
1	1	1	1	0	1	0



5. Función de salida:

Q1	Q0	Out
0	0	0
0	1	0
1	0	0
1	1	1

$\Rightarrow Out = Q_1 \bar{Q}_0$

6. Funciones de estado

	Q1 Q0			
	00	01	11	10
In				
0				
1		1	1	1

$$D_1 = Q_0 In + Q_1 In = (Q_0 + Q_1) In$$

	Q1 Q0			
	00	01	11	10
In				
0				
1	1	1		

$$D_0 = \bar{Q}_1 In$$

Síntesis de Circuitos Secuenciales Sincrónicos

- Ejemplo 4: Moore con biestables tipo D

7. Implementación

