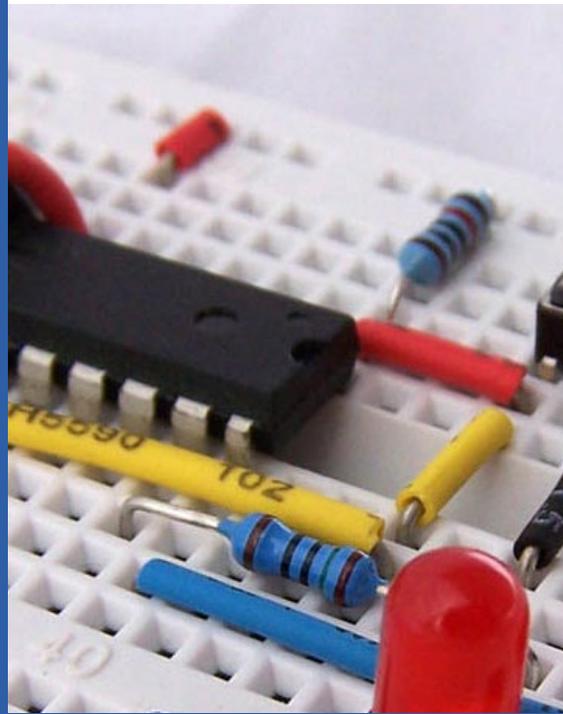
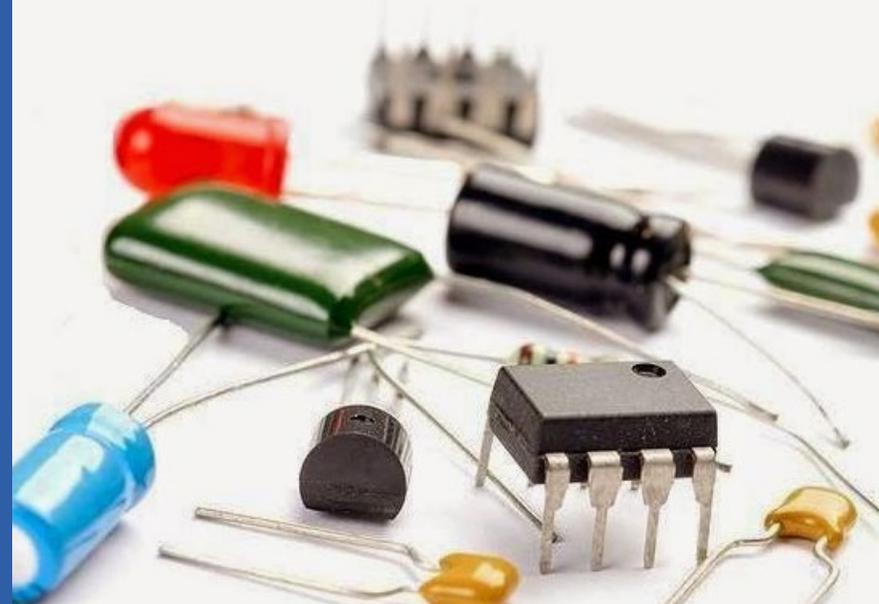
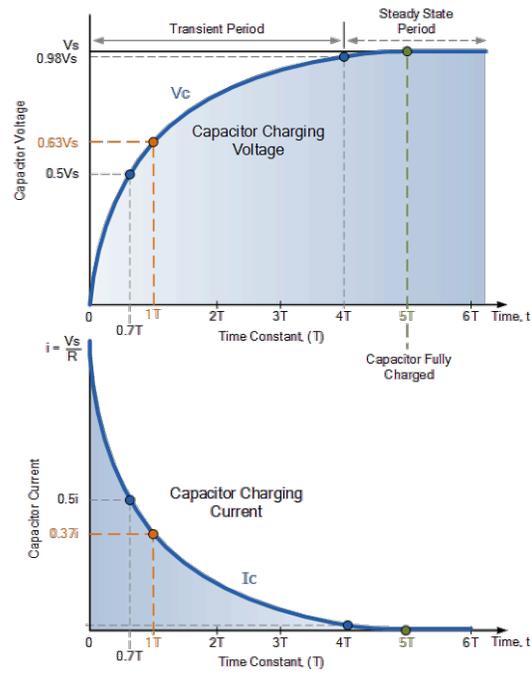


Electrónica y Tecnología de Computadores



TEMA 2 Sistemas Secuenciales



Francisco.sanchez@ufv.es



- Conocer y comprender el comportamiento y estructura de los circuitos secuenciales básicos que forman parte de cualquier sistema de computación.
- Especificar y diseñar circuitos secuenciales sencillos.
- Analizar circuitos secuenciales sencillos.

Objetivos específicos



- Distinguir entre sistemas digitales combinaciones y secuenciales
- Comprender el concepto de estado e identificar los estados de un sistema secuencial sencillo
- Distinguir entre sistemas secuenciales síncronos y asíncronos
- Utilizar diagramas de estado para especificar el comportamiento de un sistema secuencial
- Distinguir entre sistemas modelados como máquina de Mealy o Moore
- Utilizar tablas de verdad para especificar el comportamiento de un sistema secuencial
- Describir el comportamiento de un biestable tipo D, con entradas de set y clear y de otros biestables típicos.
- Distinguir entre biestables D activados por nivel, flanco de subida y bajada
- Obtener la implementación canónica de un sistema secuencial utilizando biestables tipo D y otros componentes combinacionales, así como las técnicas de diseño de sistemas combinacionales ya aprendidas.
- Analizar, Construir y verificar circuitos combinacionales sencillos utilizando Circuitos integrados.

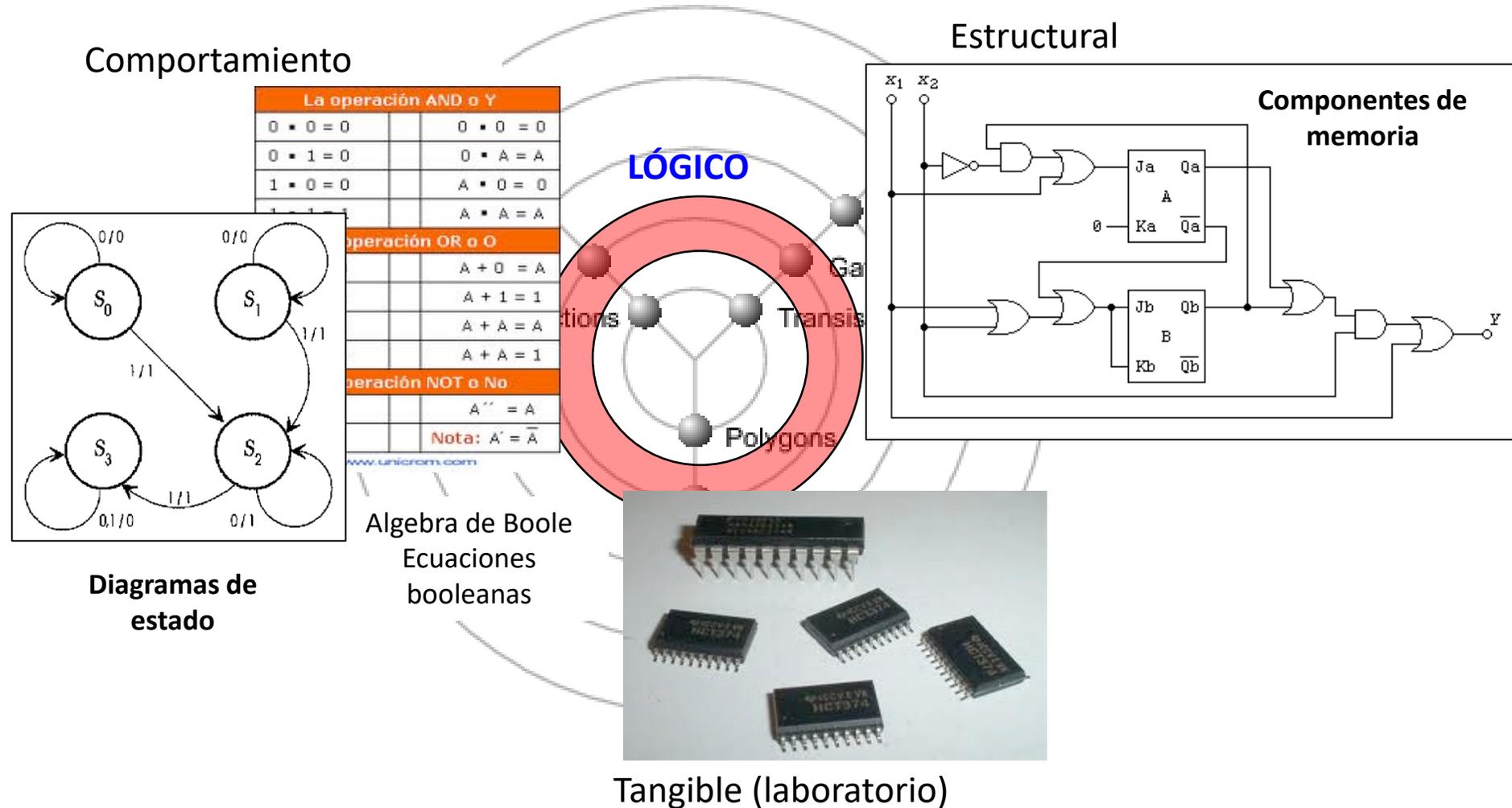


- Sistemas secuenciales vs. Combinacionales
- Especificación de sistemas secuenciales mediante diagramas de estado
- Modelado como Máquina de Mealy y Máquina de Moore
- Componentes secuenciales básicos: biestables
- Implementación canónica de un sistema secuencial

Bibliografía básica para este tema

- Fundamentos de Computadores, de R. Herminada
- Principios del diseño digital, de Daniel D. Gajski
- Diseño digital, M. Morris Mano

Niveles y dominios de este tema





- **¿Qué es un sistema secuencial?**

Sistema digital binario en el que el valor de sus señales de salida en un momento dado depende de los valores de sus señales de entrada en ese momento **y en todos los anteriores**

$$Z(t) = F(X(t), X(t-\Delta t), \dots, X(t_0))$$

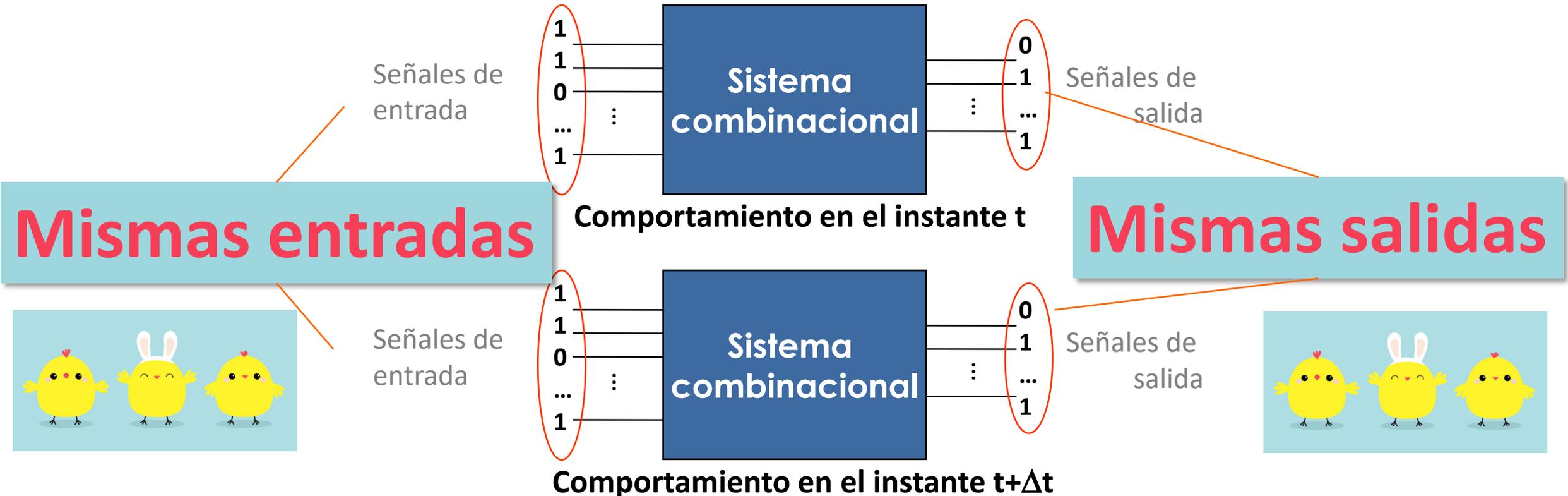
- En cambio, en un **sistema combinacional**, el valor de sus señales de salida en un momento dado depende **únicamente** de los valores de sus señales de entrada en ese momento

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

Sistemas secuenciales vs combinacionales



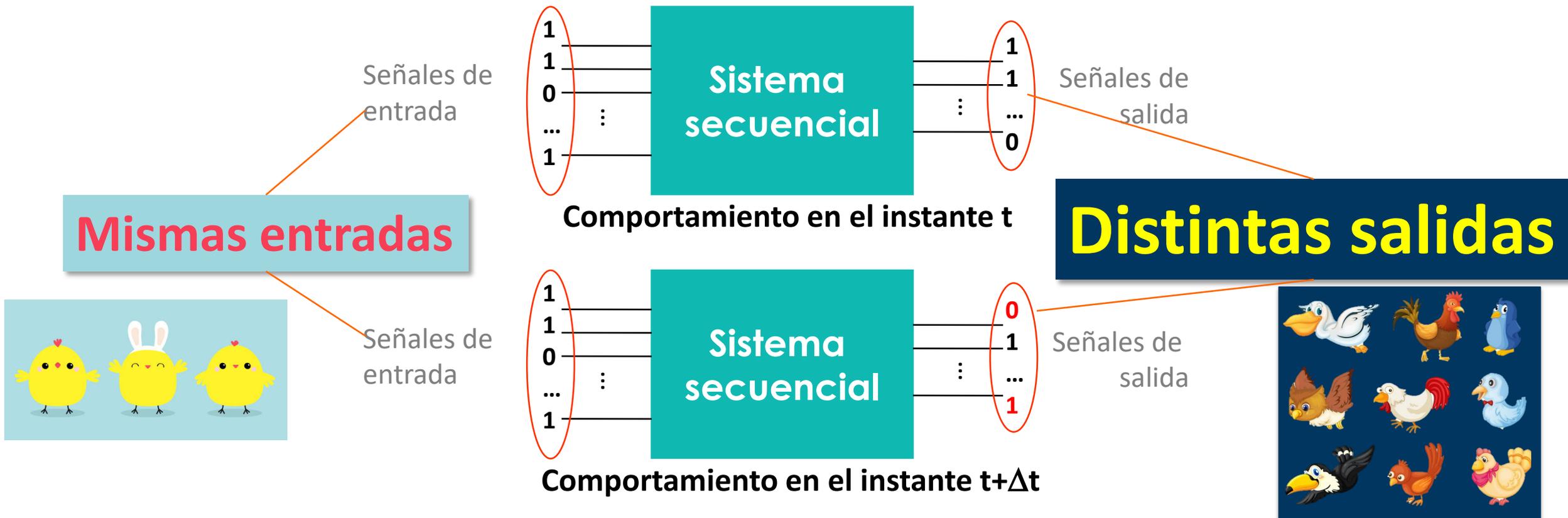
- En dos instantes de tiempo diferentes t y $t+\Delta t$, se reciben los mismos valores por las entradas del sistema.
 - El **sistema combinacional** generará los mismos valores de salida en t y $t+\Delta t$



Sistemas secuenciales vs combinacionales

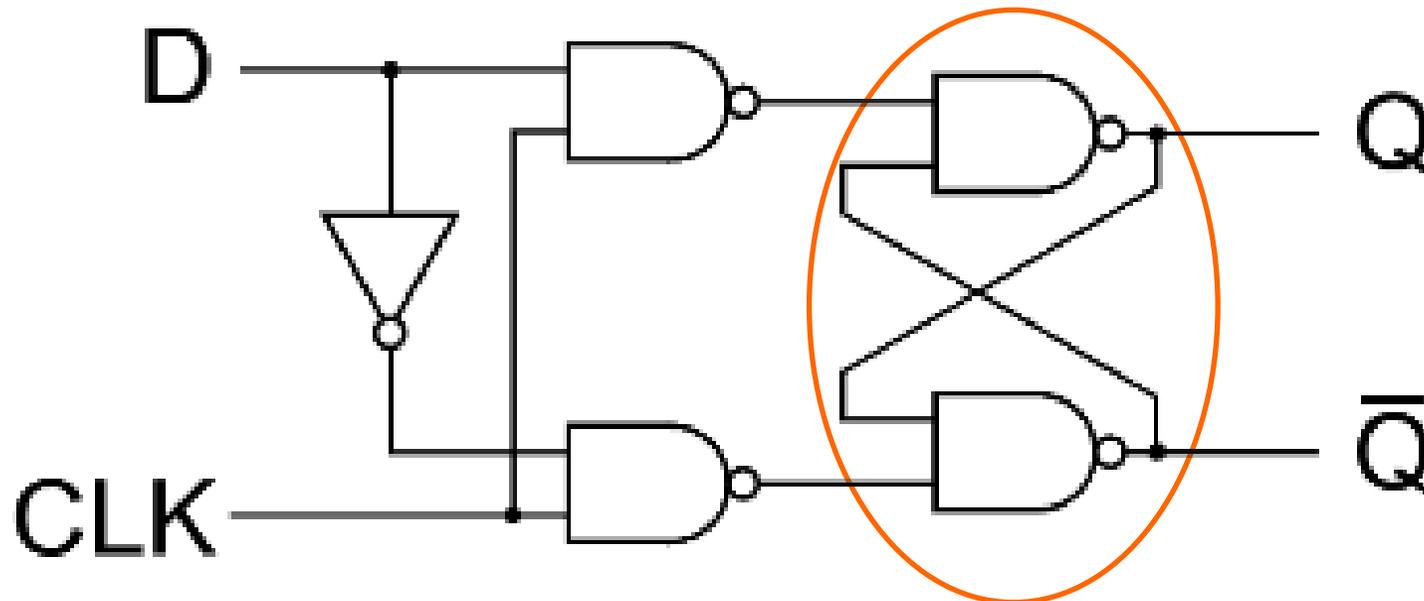


- En dos instantes de tiempo diferentes t y $t+\Delta t$, se reciben los mismos valores por las señales entradas del sistema.
 - El sistema secuencial puede no generar los mismos valores de salida en t y $t+\Delta t$





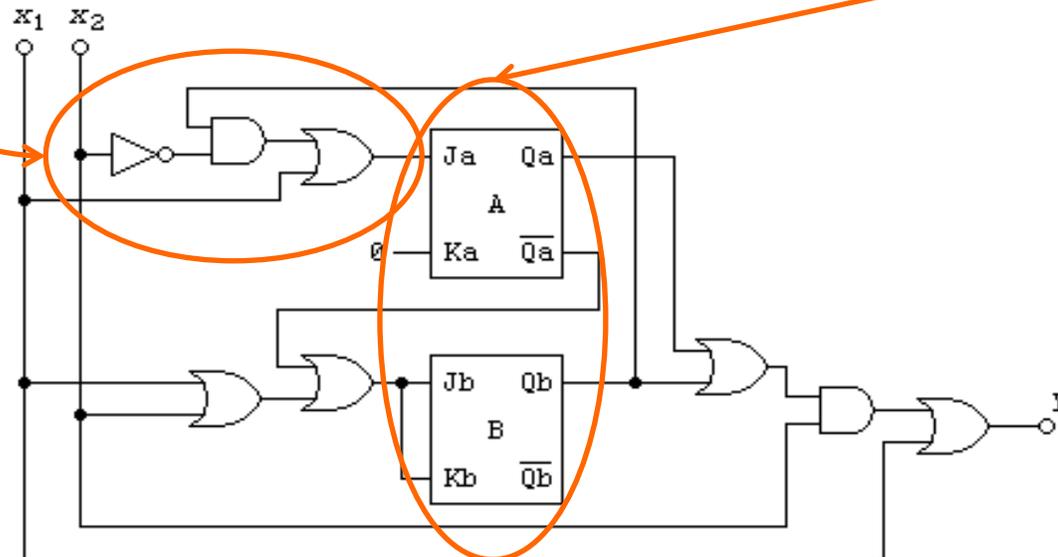
- Otras diferencias entre los circuitos secuenciales y combinacionales
 - Los circuitos secuenciales tienen uno o más caminos de “**realimentación**” (señales de salida o internas cuyo valor es calculado por el sistema, y que a su vez, vuelven a introducirse como entradas).



Sistemas secuenciales vs combinacionales



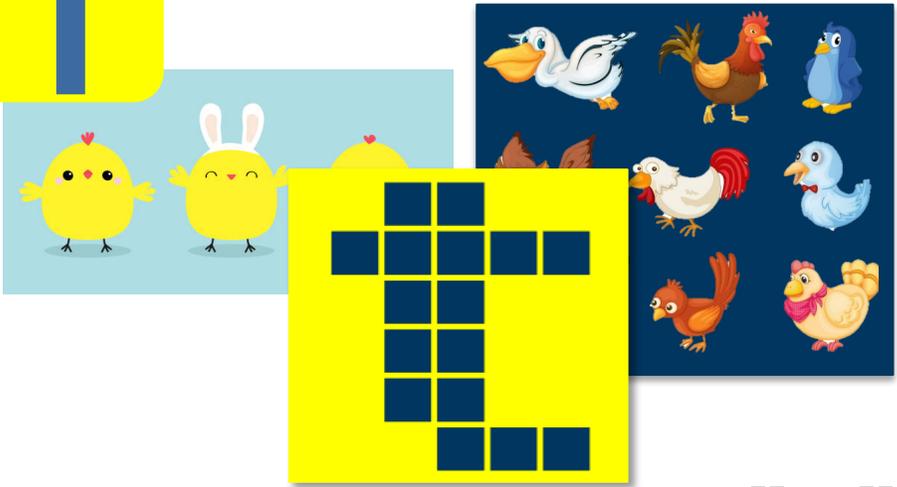
- Otras diferencias entre los circuitos secuenciales y combinacionales
 - Los circuitos secuenciales se construyen utilizando **componentes combinacionales** básicos (puertas lógicas, decodificadores, multiplexores,...) y **elementos capaces de almacenar información** (biestables y registros)



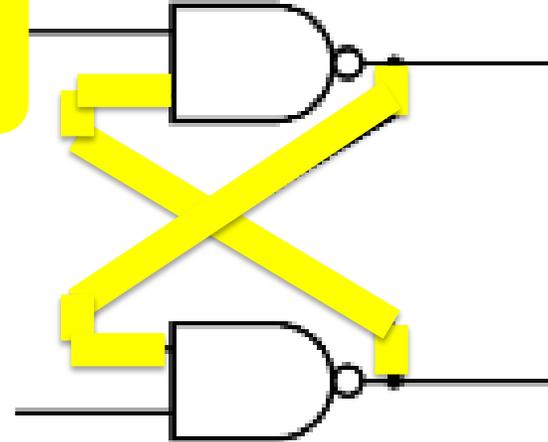
Sistemas secuenciales vs combinacionales



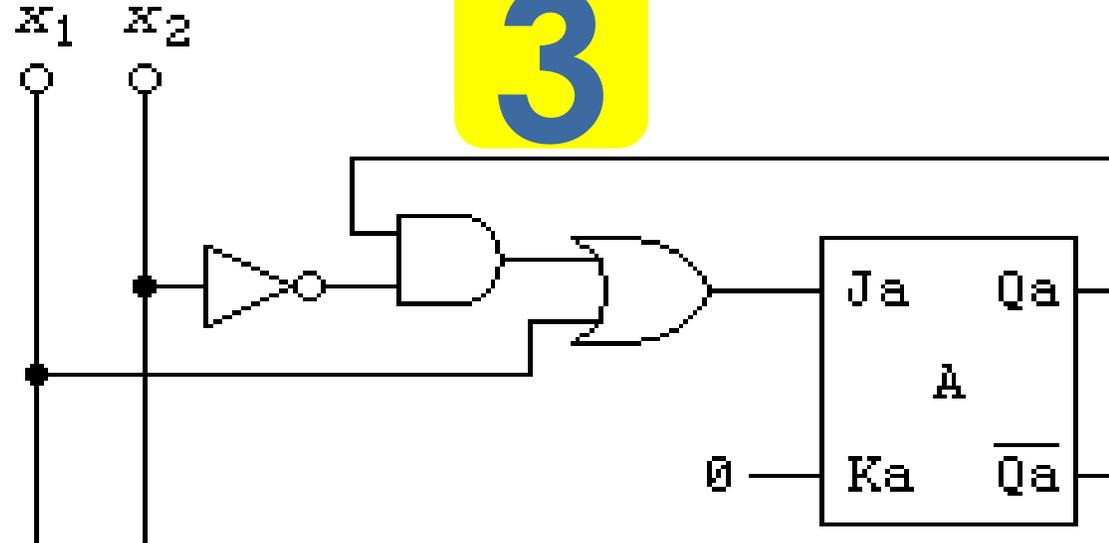
1



2

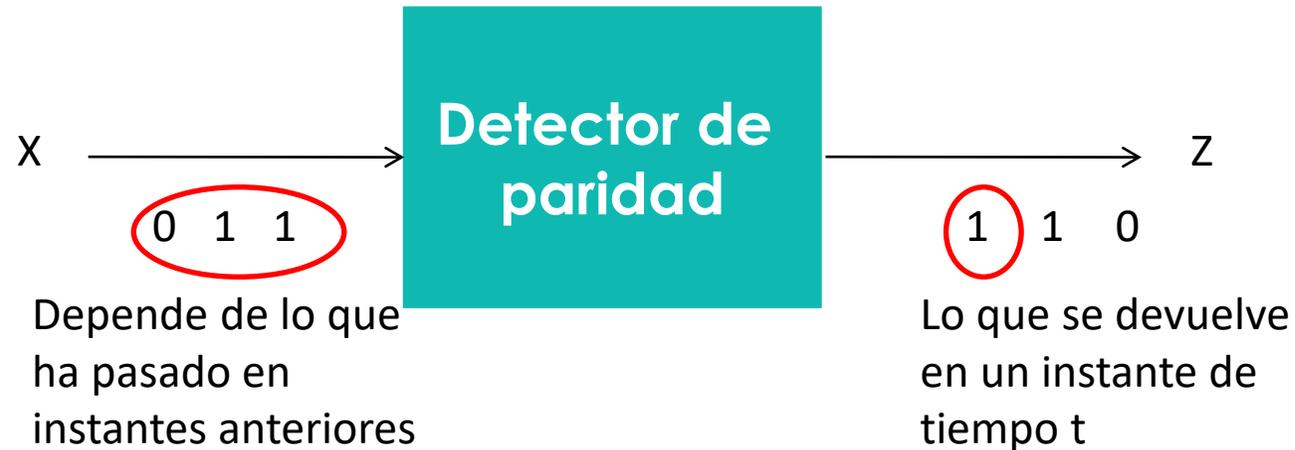


3





- **Ejemplo de sistema secuencial:** Z toma el valor 1 si el número total de unos recibidos por la entrada X es par.



Ante la entrada **1**, el sistema una vez devuelve **0** y otra **1**

Esta historia pasada tiene que ser almacenada



- **Ejemplo de sistema secuencial:** Z toma el valor 1 si el número total de unos recibidos por la entrada X es par.

Ante estas secuencias de entrada, **¿qué devuelve el sistema?**

111010110
10101010
0000110

Detector de
paridad



- **Ejemplo de sistema secuencial:** Z toma el valor 1 si el número total de unos recibidos por la entrada X es par.

Ante estas secuencias de entrada, **¿qué devuelve el sistema?**

111010110

10101010

0000110

Devuelve 1 en todas ellas,
pues el número total de unos recibidos en la
secuencia es par en todos los casos

¿Qué necesita almacenar este sistema?

- **No es necesario almacenar todas las entradas pasadas**, porque existen numerosas secuencias de entrada que producen el mismo efecto en el sistema. **Basta con identificar y almacenar la “clase” a la que pertenece la secuencia producida.**
- En el ejemplo, sólo hay dos clases de secuencias: las que tienen un número **PAR** de unos, y las que tienen un número **IMPAR**



- **Los sistemas secuenciales** tienen **estado**.

El conjunto de todas las secuencias distintas de valores de entrada que provocan el mismo efecto en el sistema se denomina **ESTADO**

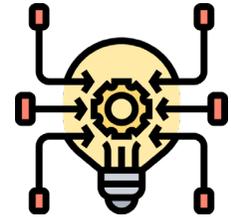


Los estados se definen mediante dos funciones de conmutación

- **Función de salida:** define el valor de la salida a partir de los valores de la entrada y el estado

$$Z(t) = F(X(t), X(t-\Delta t), \dots) \Rightarrow Z(t) = F(X(t), S(t))$$

Ejemplo: $Z(t) = F(0000110, par) \rightarrow \text{¿}Z(t)\text{?}$



- **Función de transición de estado:** define el valor siguiente del estado a partir de los valores actuales de la entrada y el estado

$$S(t+\Delta t) = G(X(t), S(t))$$

Ejemplo: $S(t+\Delta t) = G(0000110, par) \rightarrow \text{¿}S(t+\Delta t)\text{?}$





- Los sistemas secuenciales pueden ser
 - **Asíncronos**: los cambios de estado pueden producirse en cualquier momento

$$Z(t) = F(X(t), S(t))$$

$$S(t + \Delta t) = G(X(t), S(t))$$

- **Síncronos**: los cambios de estado sólo pueden producirse en instantes de tiempo concretos, marcados por una señal externa de sincronización llamada señal de **reloj**

$$Z(t) = F(X(t), S(t))$$

$$S(t + 1) = G(X(t), S(t))$$

El tiempo se mide en unidades discretas: **los pulsos de reloj**

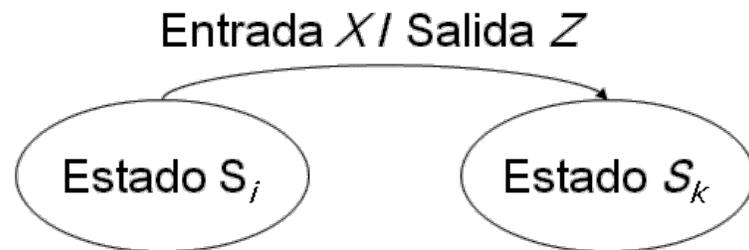
Especificación de sistemas secuenciales



Para poder diseñar un circuito secuencial es necesario partir de una **descripción formal** del sistema, igual que ocurría con los sistemas combinacionales.

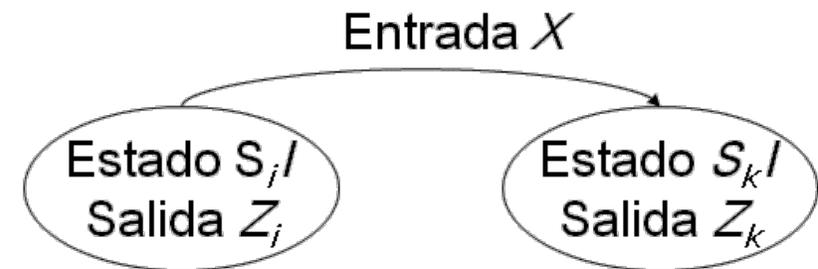


- **Diagrama de estados**: es un grafo dirigido, donde cada nodo representa un estado del sistema y cada arista representa una transición de estado. Muestra de forma gráfica el comportamiento del sistema ante diferentes secuencias de entrada.
- Existen dos formas de modelar un sistema secuencial:



$$Z(t) = F(X(t), S(t))$$
$$S(t+1) = G(X(t), S(t))$$

Máquina de Mealy



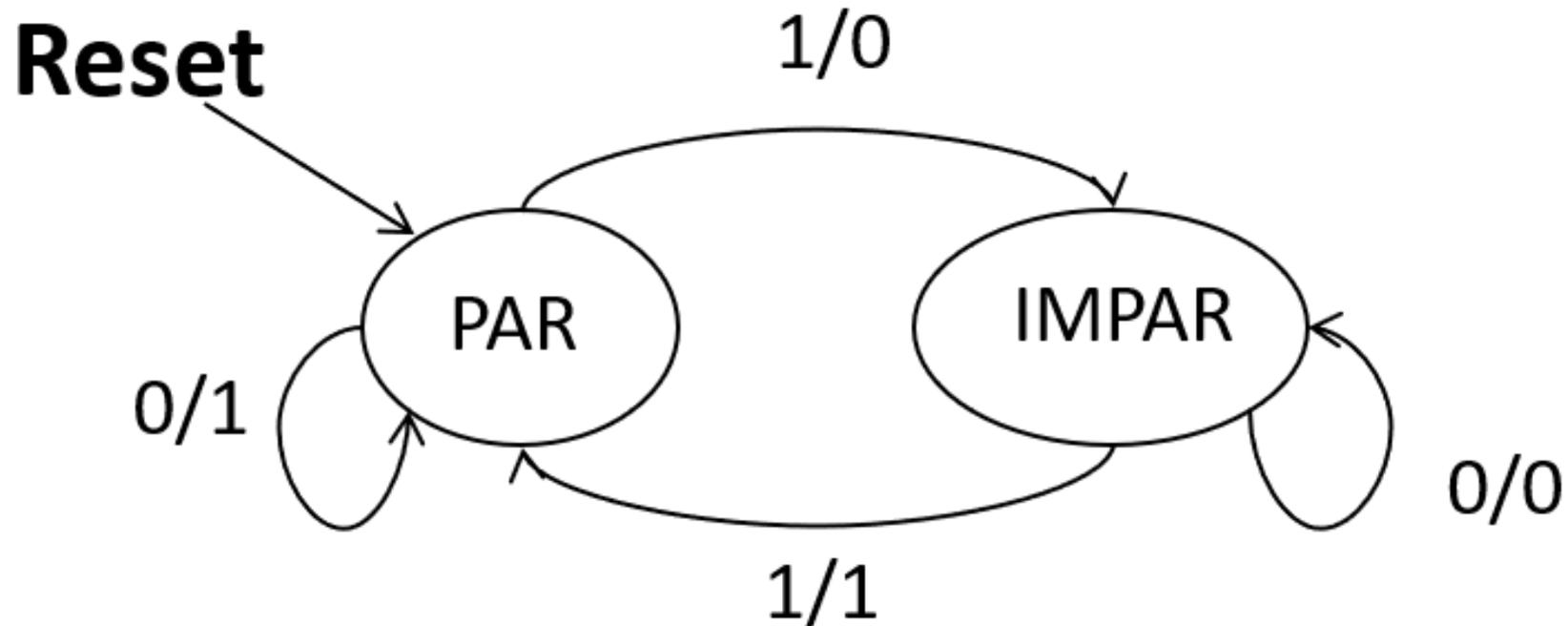
$$Z(t) = F(S(t))$$
$$S(t+1) = G(X(t), S(t))$$

Máquina de Moore



detector de paridad

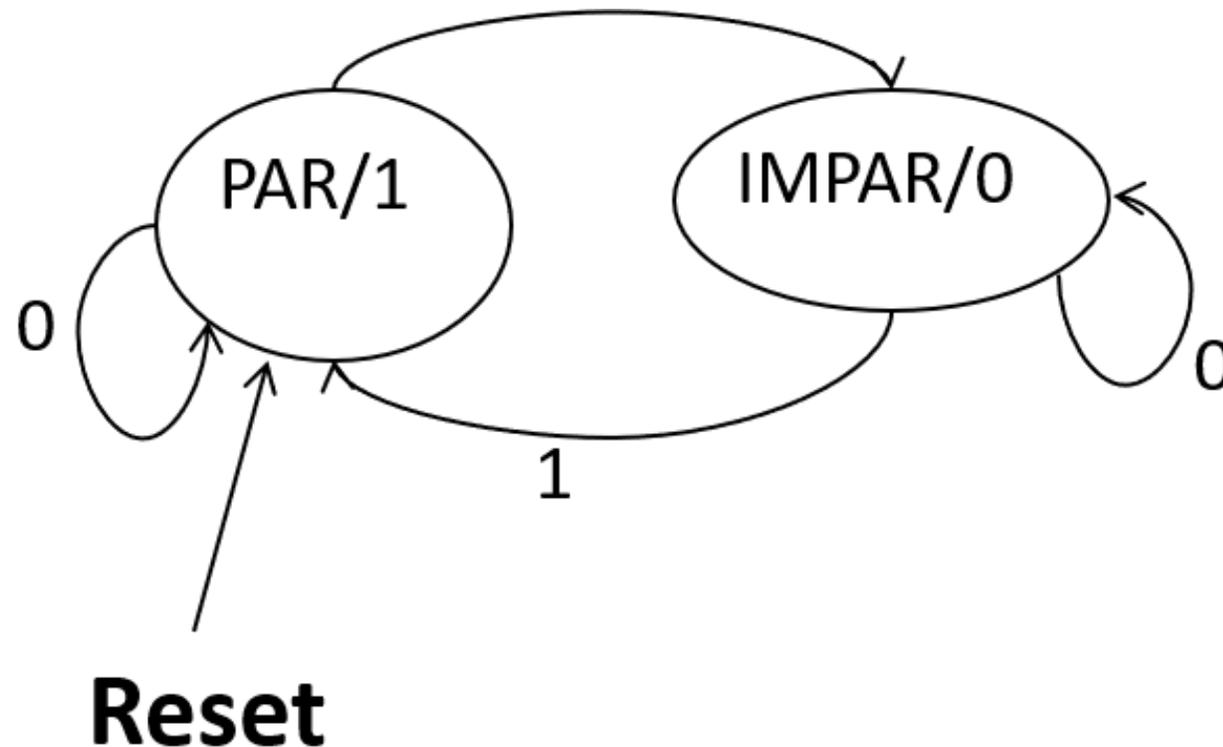
Máquina de Mealy





detector de paridad

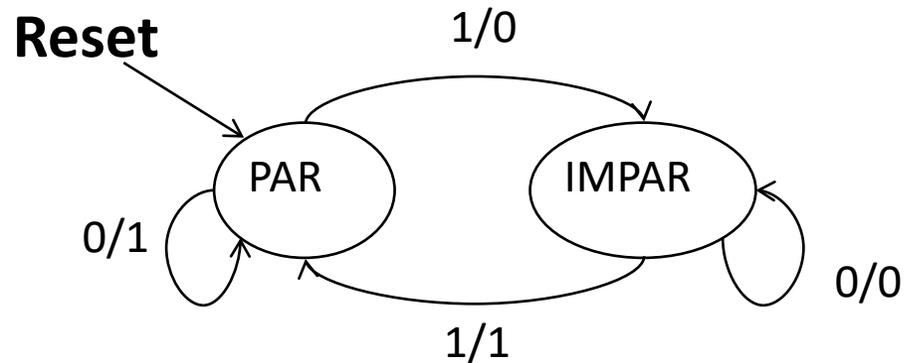
Máquina de Moore





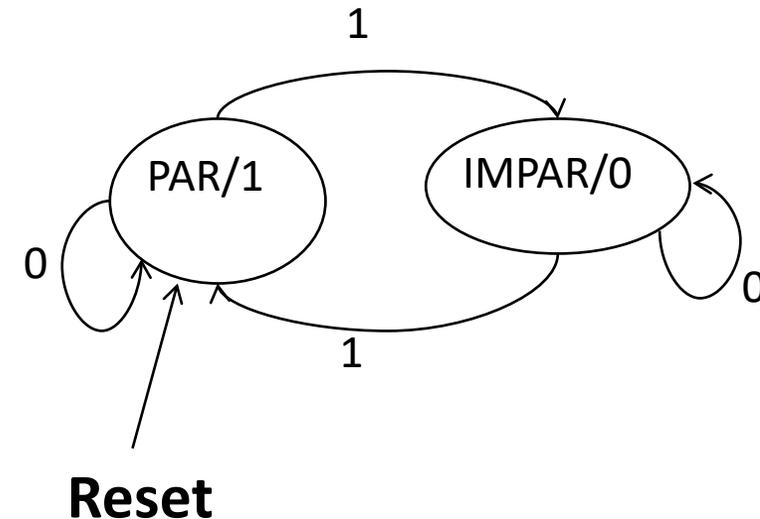
- Ejemplo: detector de paridad

Máquina de Mealy



Los sistemas secuenciales funcionan siguiendo el comportamiento definido en su diagrama de estados, a partir de un **ESTADO INICIAL**. Se utiliza una señal externa (**Reset**) para llevar al sistema a su estado inicial en cualquier momento

Máquina de Moore





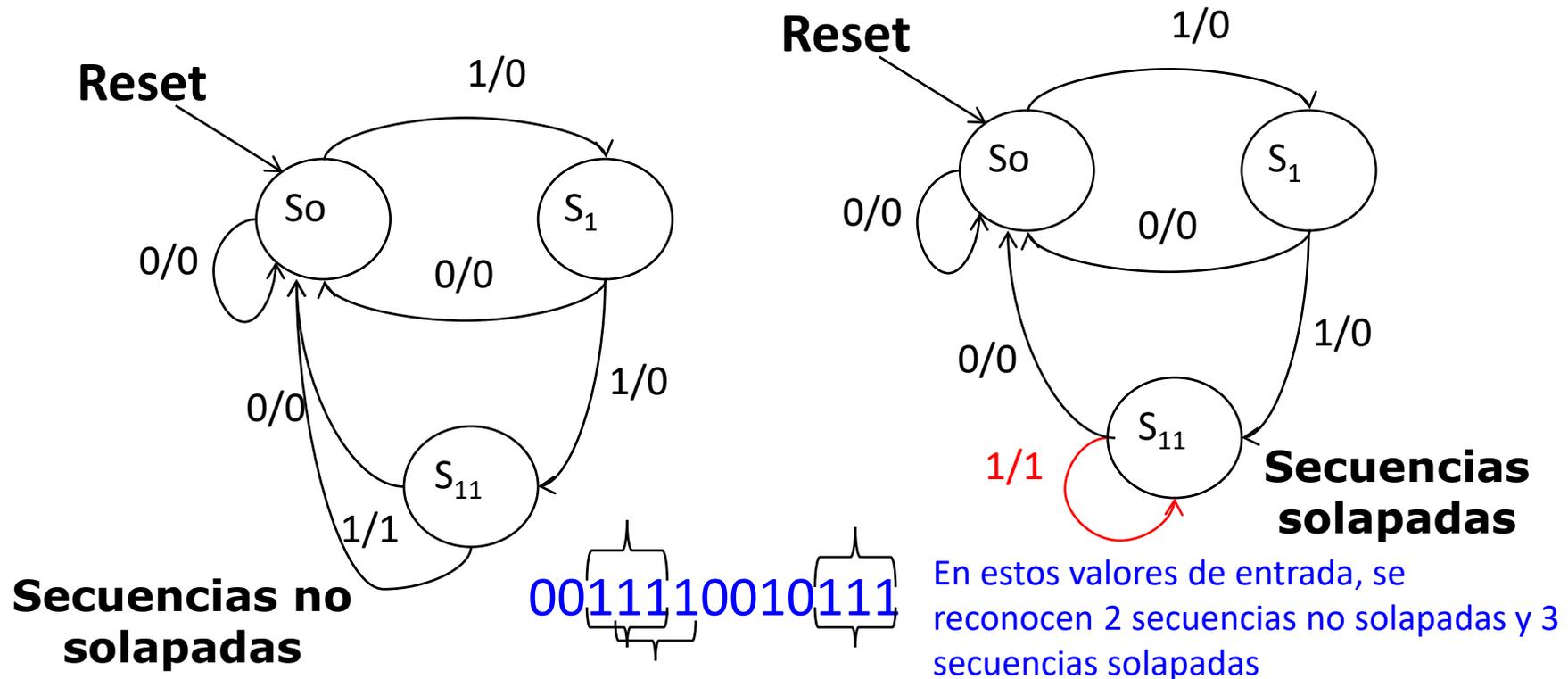
Sistemas típicos: reconocedor de secuencias

Especifica mediante un diagrama de estados como máquina de Mealy un sistema con una entrada y un salida de 1 bit que devuelva un '1' cuando por la entrada se reciba una secuencia de tres '1' seguidos, y '0' en caso contrario.

- En estos sistemas, existe un estado por cada trozo parcial de la secuencia ya reconocida
 - **S₀** (estado inicial): no se ha recibido nada aún, o se han recibido 0.
 - **S1**: se ha recibido un 1 (primer bit de la secuencia que se quiere reconocer)
 - **S11**: se han recibido dos 1 seguidos (dos primeros bits de la secuencia que se quiere reconocer)
- Pueden reconocer secuencias que se solapan (tienen bits en común) o no



- **Sistemas típicos: reconocedor de secuencias**





Sistemas típicos: contadores módulo p

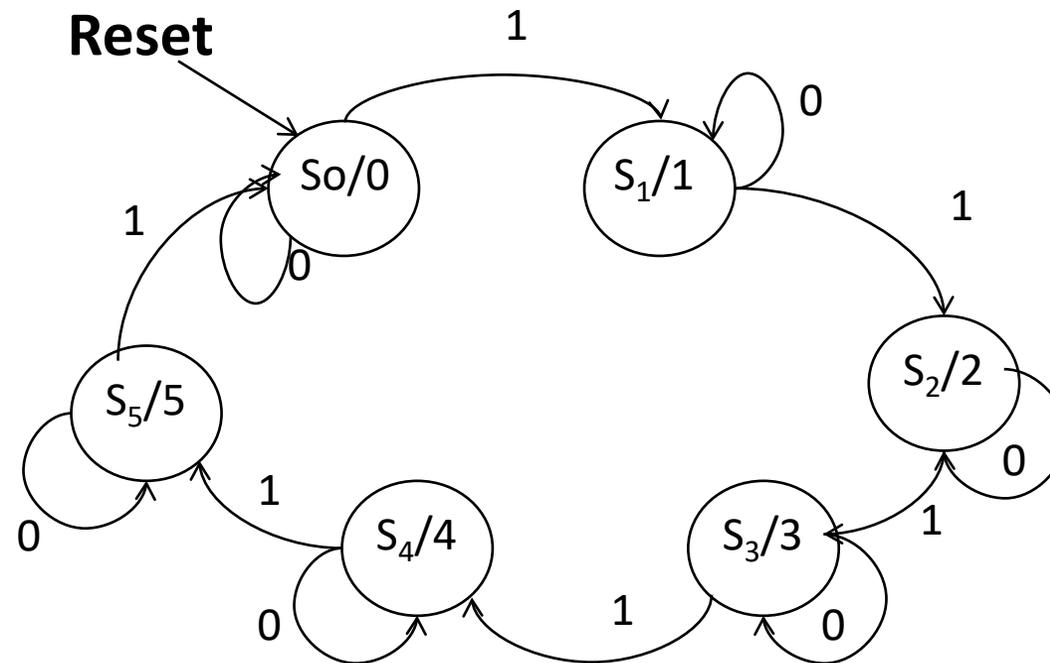
Un contador módulo p ascendente, es un sistema secuencial capaz de contar en sentido ascendente, desde 0 hasta $p-1$, volviendo nuevamente al 0. Especificar mediante un diagrama de estados como máquina de Mealy un contador módulo 6 ascendente.

- Estos sistemas tienen un estado por cada número que pueden contar (0, 1, 2, ..., $p-1$)
- Se implementan más fácilmente como máquina de Moore, porque la salida del sistema coincide con el estado.



Sistemas típicos: contadores módulo p

Un contador módulo p ascendente, es un sistema secuencial capaz de contar en sentido ascendente, desde 0 hasta $p-1$, volviendo nuevamente al 0. Especificar mediante un diagrama de estados como máquina de Moore un contador módulo 6 ascendente.



x	y	z=x⊕y	w=x⊕z
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

↑ y = w ↑

$$z = x \oplus y = \bar{x}y + x\bar{y}$$

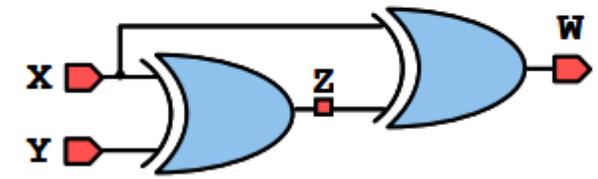
$$w = x \oplus z = \bar{x}z + x\bar{z} = \bar{x}(\bar{x}y + x\bar{y}) + x(\overline{\bar{x}y + x\bar{y}})$$

$$= \underbrace{\bar{x}\bar{x}}_0 y + \underbrace{x\bar{y}}_0 + x(\bar{x}\bar{y} + \overline{x\bar{y}}) = \bar{x}y + x(x + \bar{y}) \cdot (\bar{x} + y)$$

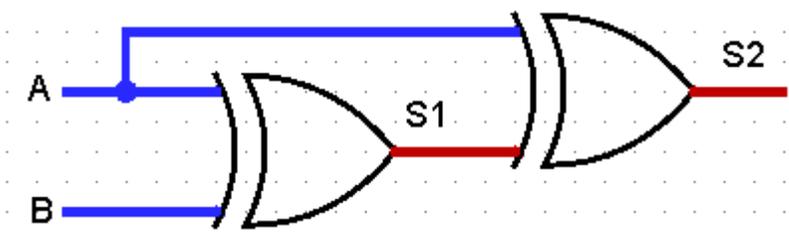
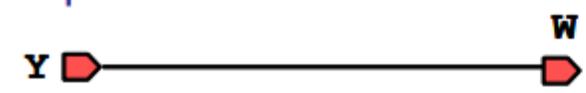
$$= \bar{x}y + x(\underbrace{x\bar{x}}_0 + \underbrace{x\bar{y}}_0 + \underbrace{\bar{x}y}_x + \underbrace{y\bar{y}}_0) = \bar{x}y + x\bar{y} + xy = (\bar{x} + x)y = y$$

$$= \bar{x}y + xy = (\bar{x} + x)y = y$$

Por tanto



Es equivalente a:



$$\overline{B} \cdot \overline{C} + \overline{B} \cdot \overline{D} + A \cdot \overline{C}$$

$$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} + AB\bar{C}\bar{D} + ABCD$$

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

$\bar{A}\bar{B}\bar{C}\bar{D}$
 $\bar{A}\bar{B}\bar{C}D$
 $\bar{A}\bar{B}C\bar{D}$
 $A\bar{B}\bar{C}\bar{D}$
 $A\bar{B}\bar{C}D$
 $A\bar{B}C\bar{D}$
 $A\bar{B}C\bar{D}$

A	B	C	D	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

$A+B+\bar{C}+\bar{D}$
 $A+\bar{B}+C+D$
 $A+\bar{B}+C+\bar{D}$
 $A+\bar{B}+\bar{C}+D$
 $A+\bar{B}+\bar{C}+\bar{D}$
 $\bar{A}+B+\bar{C}+\bar{D}$
 $\bar{A}+\bar{B}+\bar{C}+D$
 $\bar{A}+\bar{B}+\bar{C}+\bar{D}$

$$\overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot D + \overline{A} \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot \overline{D} + A \cdot \overline{B} \cdot \overline{C} \cdot D + A \cdot \overline{B} \cdot C \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot D$$

SOP

CD

		00	01	11	10
AB	00	1	1	0	1
	01	0	0	0	0
	11	1	1	0	0
	10	1	1	0	1

$$\overline{B} \cdot \overline{C} + \overline{B} \cdot \overline{D} + A \cdot \overline{C}$$

Revisitando conceptos → FSM



Sistemas secuenciales: Concepto de estado

- ❑ 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. Para ello es necesario que el sistema disponga de elementos de memoria que le permitan recordar la situación en que se encuentra (estado).



$X(t)$: Entrada actual.
 $Z(t)$: Salida actual.
 $S(t)$: Estado actual.
 $S(t+1)$: Estado próximo.

- ❑ Como un sistema secuencial es finito, tiene una capacidad de memoria finita y un conjunto finito de estados posibles:
Máquina Finita de Estados (FSM: Finite State Machine).



Sincronismo

❑ Existen dos (2) tipos de sistemas secuenciales:

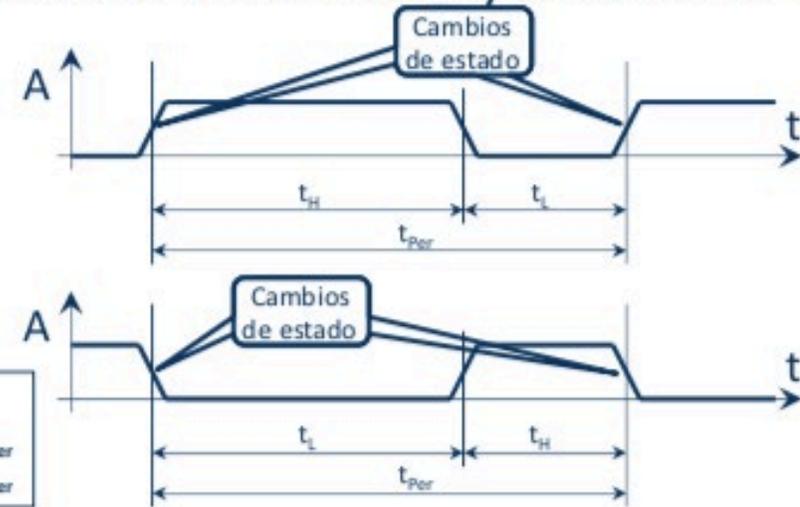
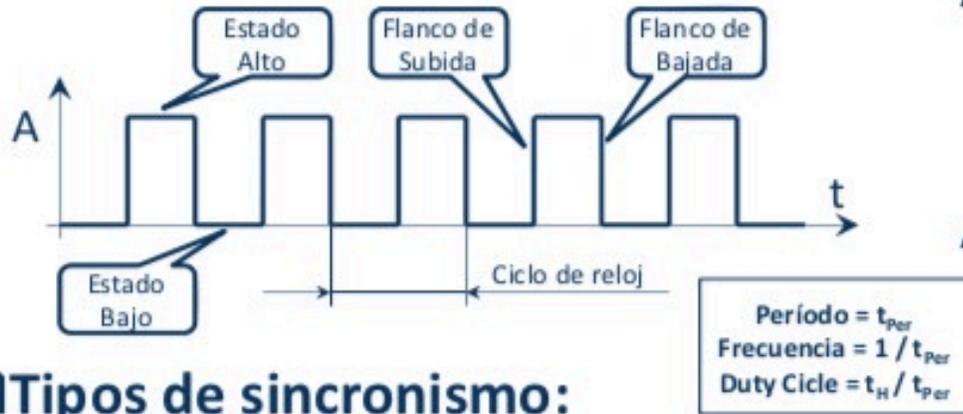
- Los asíncronos son sistemas secuenciales que pueden cambiar de estado en cualquier instante de tiempo en función de cambios en las señales de entrada.
 - Son más frecuentes en la vida real. ¡Pero no en sistemas computacionales!
 - Existen métodos específicos para diseñar sistemas asíncronos
- Los síncronos son sistemas secuenciales que sólo pueden cambiar de estado en determinados instantes de tiempo, es decir, están “sincronizados” con una señal que marca dichos instantes y que se conoce como señal de reloj (Clk).
 - El sistema sólo hace caso de las entradas en los instantes de sincronismo.
 - Son más fáciles de diseñar.



Relojes y sincronismo

❑ Propiedades de la señal de reloj.

- Normalmente se suelen utilizar dispositivos síncronos cuyos cambios de estado se rigen por la señal de reloj.



❑ Tipos de sincronismo:

- Sincronismo por nivel (alto ó bajo): el sistema hace caso de las entradas mientras el reloj esté en el nivel activo (alto ó bajo).
- Sincronismo por flanco (de subida o de bajada): el sistema hace caso de las entradas y evoluciona justo cuando se produce el flanco activo (de subida ó de bajada).

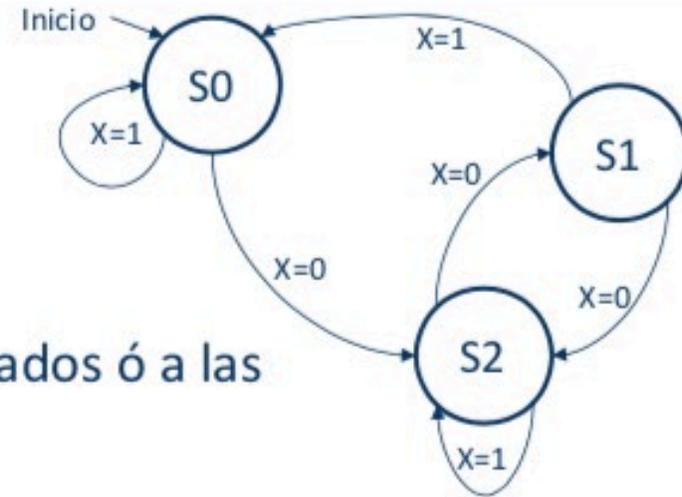
Para especificar sistemas secuenciales...



Técnicas de representación de sistemas secuenciales

❑ Diagrama de Transición de Estados.

- Gráficos.
 - Estados: Círculos.
 - Transiciones: Arcos.
- Las salidas pueden estar asociadas a los estados ó a las transiciones.
- El nombre de las entradas se puede omitir.



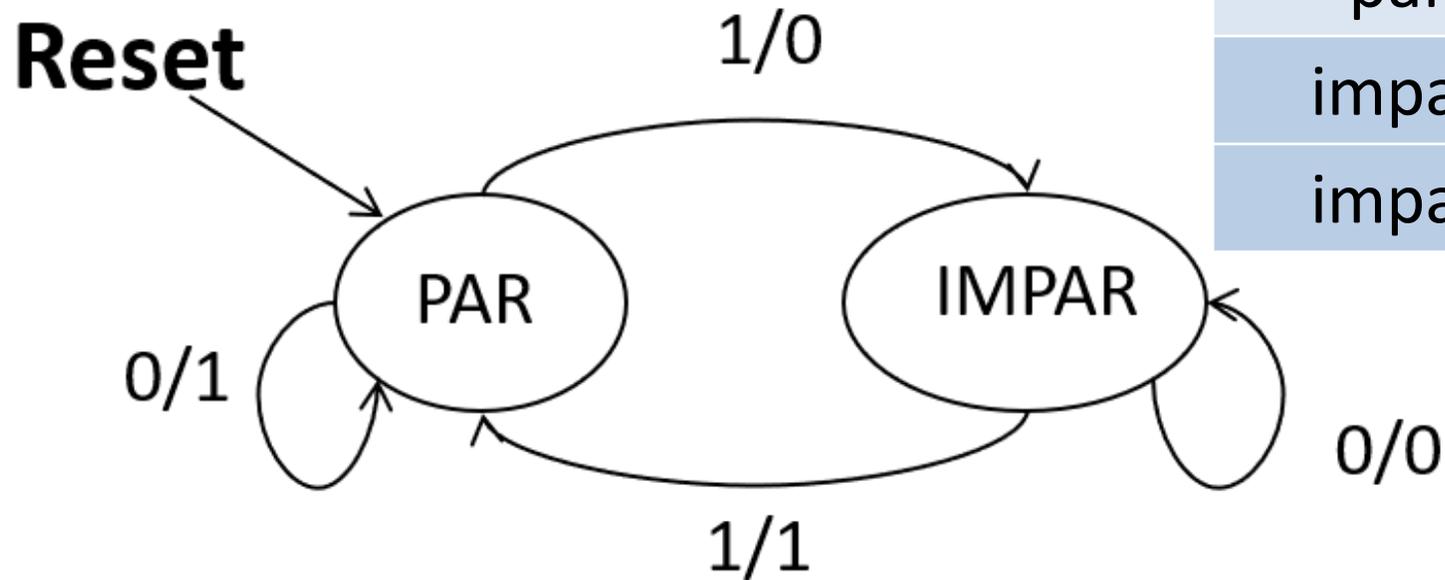
❑ Tablas de transición de estados.

- Son tablas de verdad en las que aparece descrito el comportamiento del sistema.
- El estado presente aparece como una de las entradas (a veces el reloj también).

Estado Actual	Entrada (X)	Estado siguiente
S0	0	S2
S0	1	S0
S1	0	S2
S1	1	S0
S2	0	S1
S2	1	S2



detector de paridad



estado actual	entrada	estado siguiente
par	0	par
par	1	impar
impar	0	impar
impar	1	par

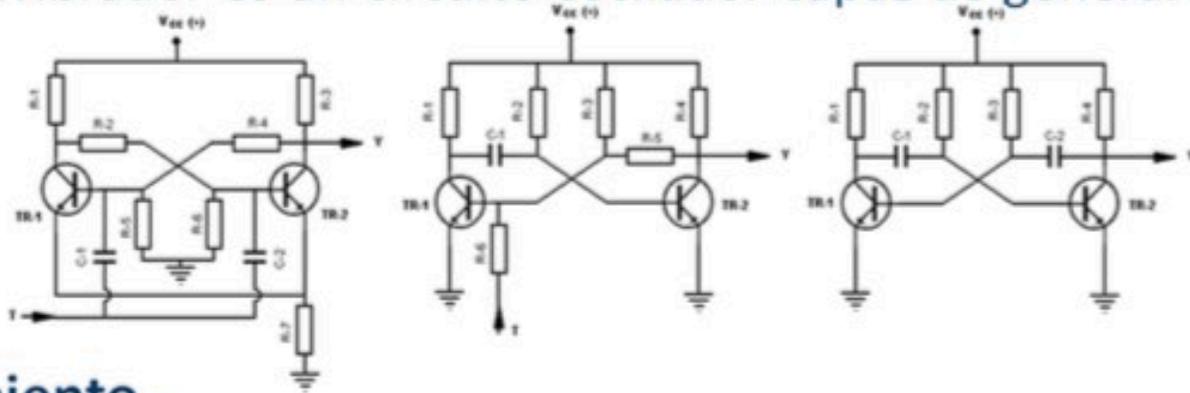
Inciso: Concepto de multivibrador



Multivibradores

Definición.

- En electrónica, un multivibrador es un circuito oscilador capaz de generar una onda cuadrada.



Tipos por su funcionamiento.

- Funcionamiento continuo, astable ó de oscilación libre: genera ondas a partir de la propia fuente de alimentación.
- Funcionamiento por disparo, a partir de una señal de disparo ó impulso sale de su estado de reposo. Éstos a su vez se pueden diferenciar en dos (2) subtipos.
 - Con dos (2) estados estables, se denomina biestable.
 - Con un (1) único estado estable, se denomina monoestable.

Latch SR asíncrono



Latch S-R (Set-Reset)

Definición.

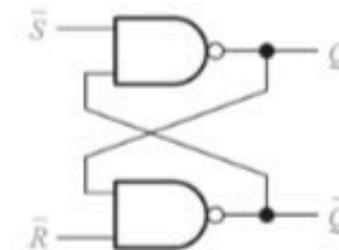
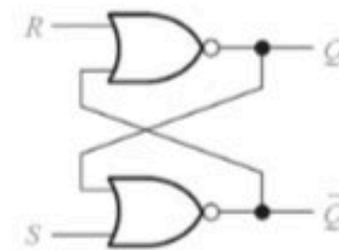
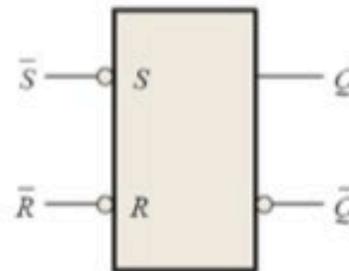
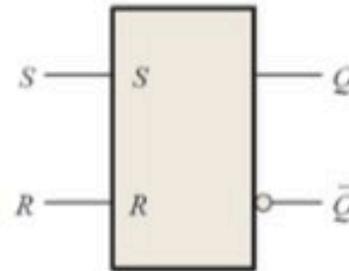
Una báscula "RS" es un tipo de dispositivo lógico biestable ó multivibrador, que dispone de dos (2) entradas y dos (2) salidas complementarias*.

Tabla.

S	R	Q	\bar{Q}	Descripción
0	0	NC	NC	No Cambia. Se mantiene en el estado anterior.
1	0	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
0	1	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.
1	1	x	x	Oscilación. Condición NO válida.

\bar{S}	\bar{R}	Q	\bar{Q}	Descripción
1	1	NC	NC	No Cambia. Se mantiene en el estado anterior.
0	1	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
1	0	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.
0	0	x	x	Oscilación. Condición NO válida.

Circuito, representación gráfica.

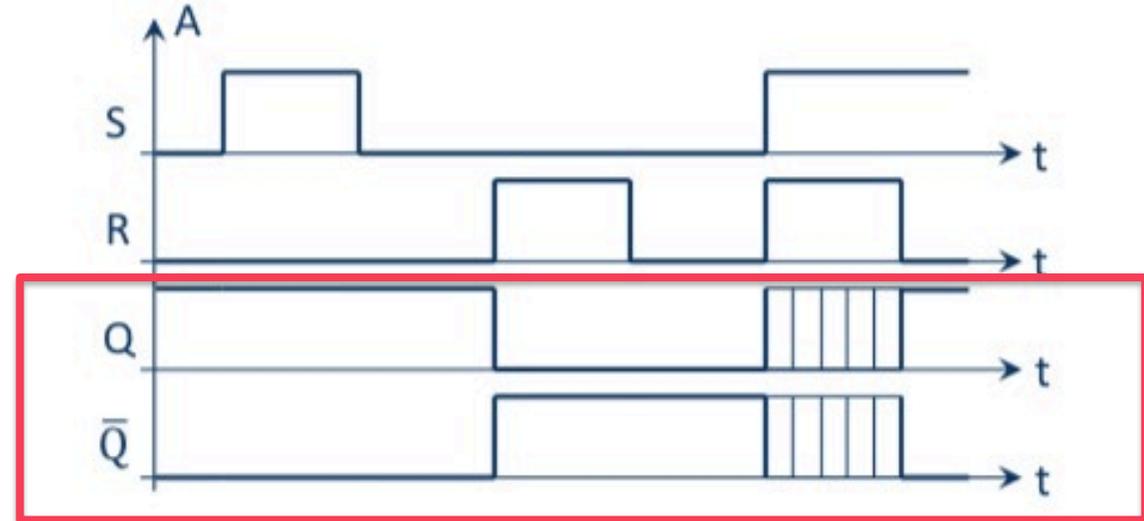
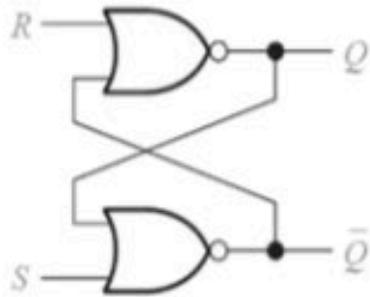


Funcionamiento Latch SR asíncrono

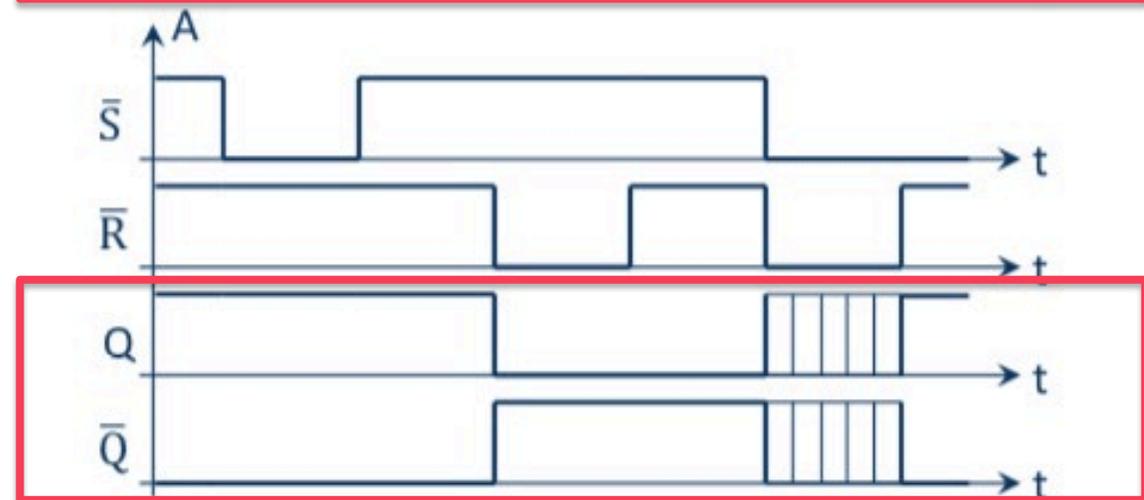
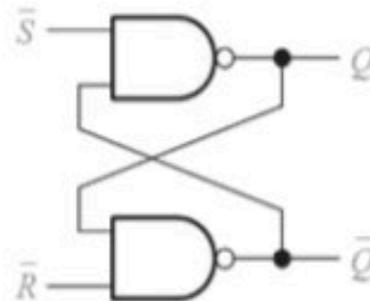


Latch S-R (Set-Reset): Análisis de funcionamiento

Con puertas "NOR".



Con puertas "NAND".



Latch SR con entrada "Enable"



Latch S-R (Set-Reset), con entrada de habilitación

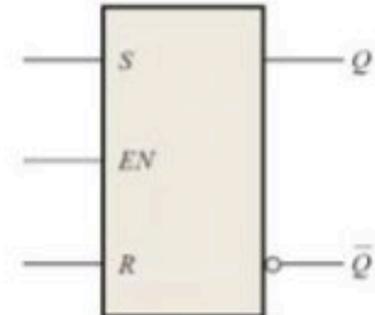
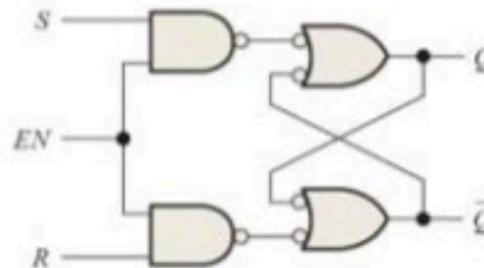
□ Definición.

Una báscula "RS" es un tipo de dispositivo lógico biestable ó multivibrador, que dispone de tres (3) entradas y dos (2) salidas complementarias*. Las entradas "R" y "S" controlan el estado de la báscula cuando la entrada "EN" está en ALTO, pero la báscula no cambia de estado si la entrada "EN" está en BAJO.

□ Tabla.

E	S	R	Q	\bar{Q}	Descripción
0	X	X	NC	NC	No Cambia.
1	0	0	NC	NC	No Cambia. Se mantiene en el estado anterior.
1	1	0	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
1	0	1	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.
1	1	1	x	x	Oscilación. Condición NO válida.

□ Circuito, representación gráfica.



Latch D con entrada "Enable"



Latch D (Data), con entrada de habilitación

Definición.

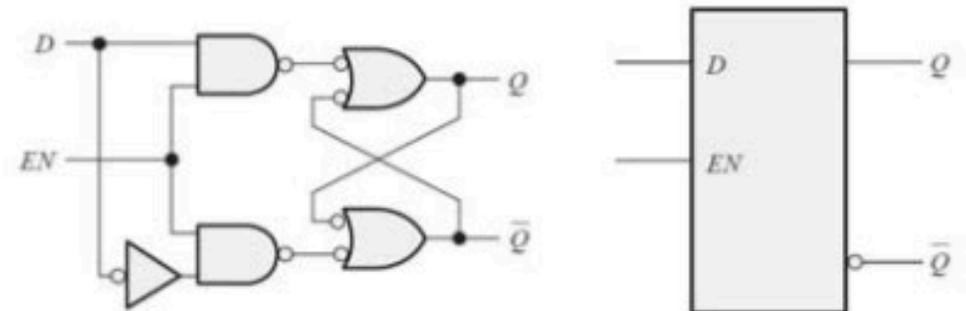
Partiendo de una báscula "RS" con "Enable", si se unen las entradas "R" y "S", se denomina báscula "D" (Dato). Cuando la entrada "D" está a nivel ALTO y la entrada "EN" también, la báscula se pone en estado SET. Cuando la entrada "D" está a nivel BAJO y la entrada "EN" está a nivel ALTO, la báscula se pone en estado RESET.

Con una puerta NOT

Tabla.

E	D	Q	\bar{Q}	Descripción
0	X	NC	NC	No Cambia.
1	1	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
1	0	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.

Circuito, representación gráfica.





❑ Hasta ahora:

- El funcionamiento de los circuitos dependía de los estados de las señales de entrada y salida.
- No existía una señal que “sincronizara”.
- Circuitos Asíncronos.

❑ “A partir de ahora”:

- Se van utilizar señales que “marquen” el instante.
- En cada circuito, en distintos ó en todos los circuitos.
- Circuitos Síncronos.

❑ La idea:

- Añadir un “detector de flanco” y añadirlo a los flip-flops estudiados.

RS Disparado por flanco



Flip-flops disparados por flanco

Latch RS

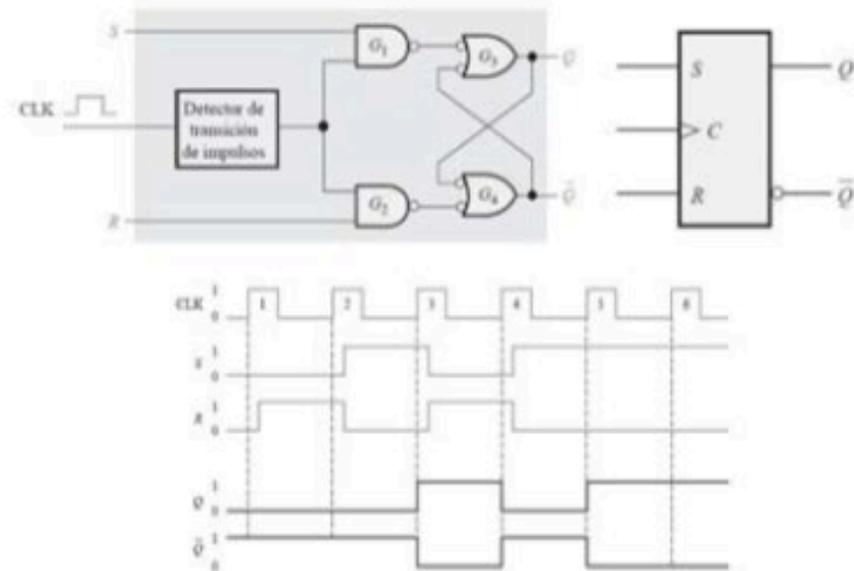
❑ Tabla.

!!! Con flancos de subida !!!

Clk	S	R	Q	\bar{Q}	Descripción
0	X	X	NC	NC	No Cambia.
↑	0	0	NC	NC	No Cambia. Se mantiene en el estado anterior.
↑	1	0	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
↑	0	1	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.
↑	1	1	x	x	Indeterminado. Condición NO válida.

¿ Qué ocurría cuando no funcionaba por flanco ?

❑ Circuito, representación gráfica y cronograma.



RS Disparado por flanco



Flip-flops disparados por flanco

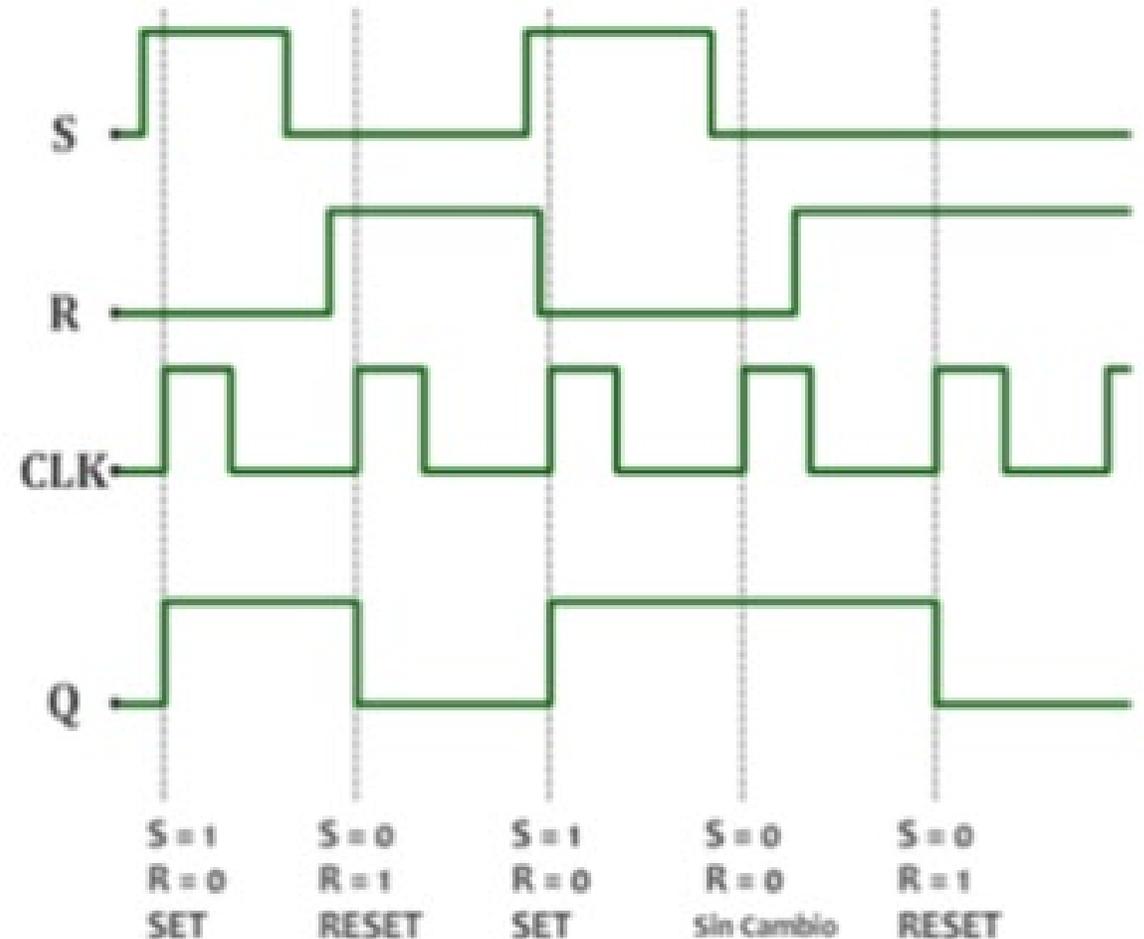
Latch RS

□ Tabla.

!!! Con flancos de subida !!!

Clk	S	R	Q	\bar{Q}	Descripción
0	X	X	NC	NC	No Cambia.
↑	0	0	NC	NC	No Cambia. Se mantiene en el estado anterior.
↑	1	0	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
↑	0	1	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.
↑	1	1	x	x	Indeterminado. Condición NO válida.

¿ Qué ocurría cuando no funcionaba por flanco ?



D Disparado por flanco



Flip-flops disparados por flanco

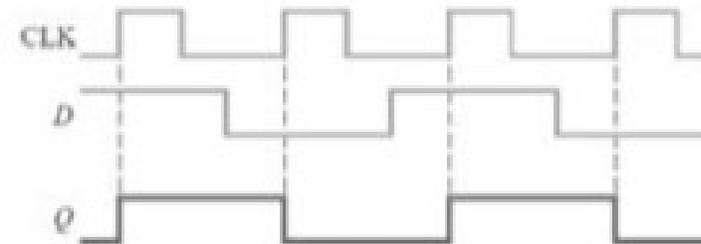
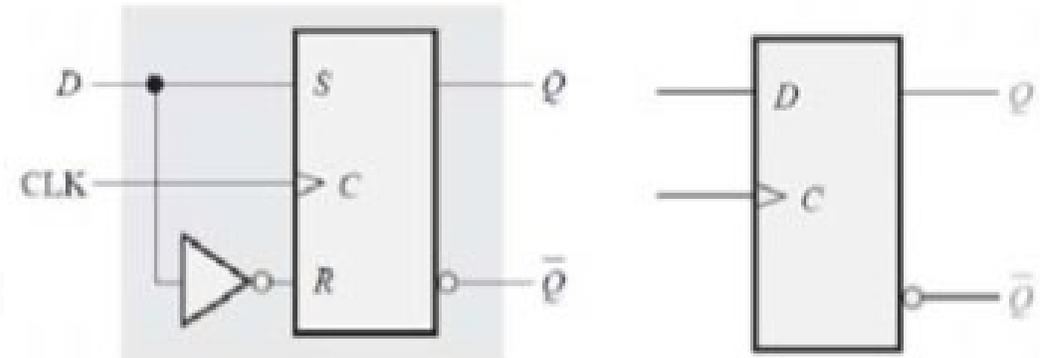
Latch D

❑ Tabla.

!!! Con flancos de subida !!!

Clk	D	Q	\bar{Q}	Descripción
0	X	NC	NC	No Cambia.
↑	1	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
↑	0	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.

❑ Circuito, representación gráfica y cronograma.



JK Disparado por flanco



Latch JK

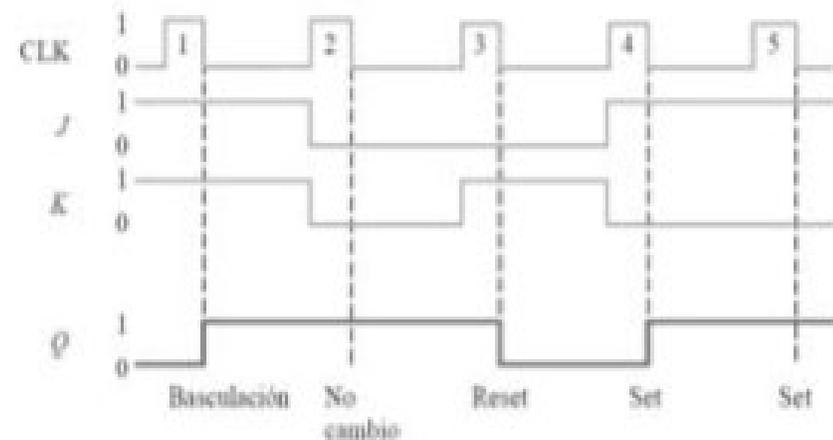
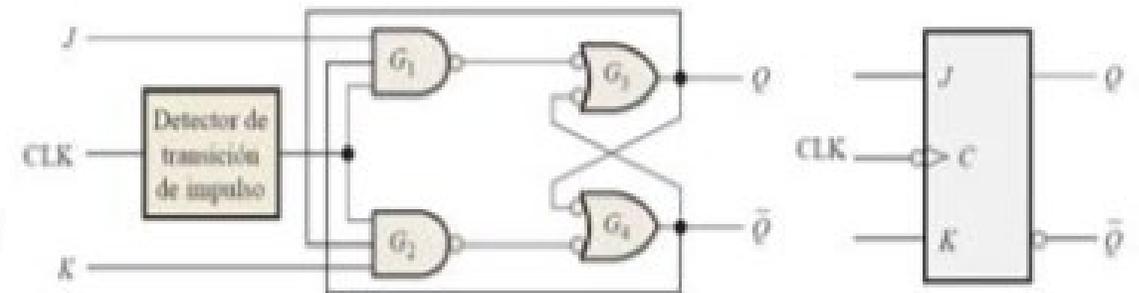
Tabla.

!!! Con flancos de bajada !!!

Clk	J	K	Q	\bar{Q}	Descripción
0	X	X	NC	NC	No Cambia.
↓	0	0	NC	NC	No Cambia. Se mantiene en el estado anterior.
↓	1	0	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
↓	0	1	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.
↓	1	1	x	x	Toggle. Basculan las salidas.

¿ Y esto tendrá alguna ventaja ?

Circuito, representación gráfica y cronograma.



T Disparado por flanco



Flip-flops disparados por flanco

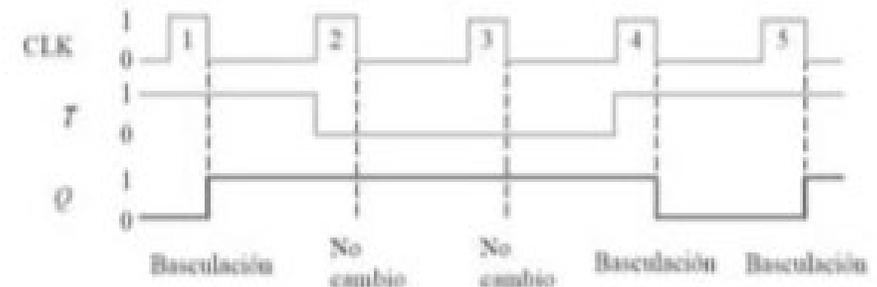
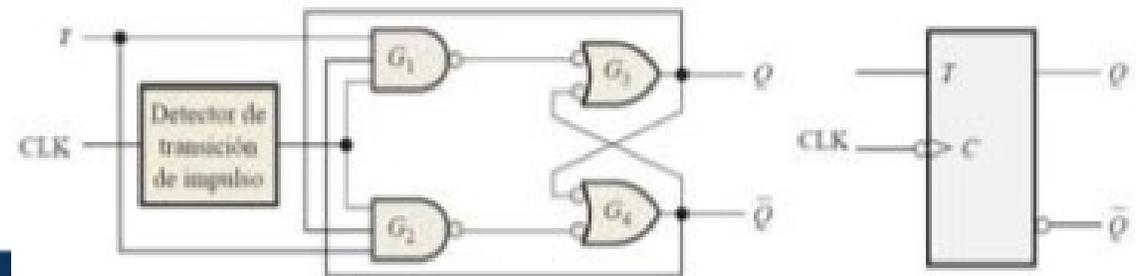
Latch T

Tabla.

!!! Con flancos de bajada !!!

Clk	T	Q	\bar{Q}	Descripción
0	X	NC	NC	No Cambia.
↓	0	NC	NC	No cambia. Se mantiene en el estado anterior.
↓	1	\bar{Q}_0	Q_0	Toggle. Basculan las salidas.

Circuito, representación gráfica y cronograma.

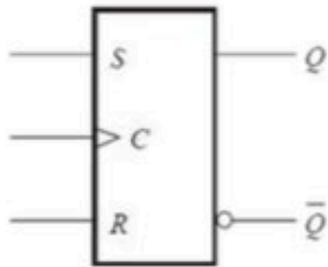


Resumen – Flancos positivos y negativos

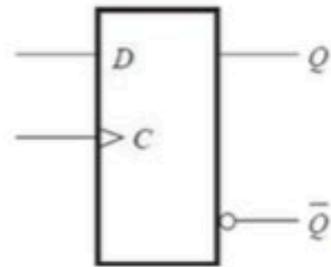


Flip-flops disparados por flanco

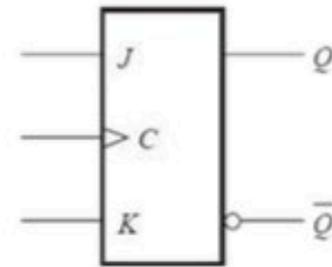
Resumen tipos de Latches



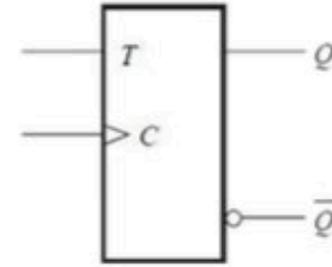
R-S



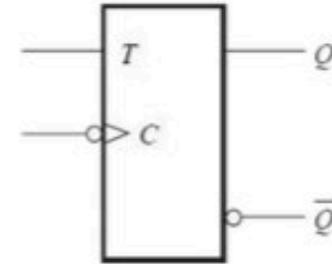
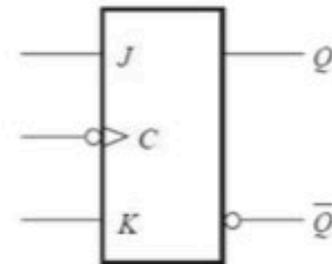
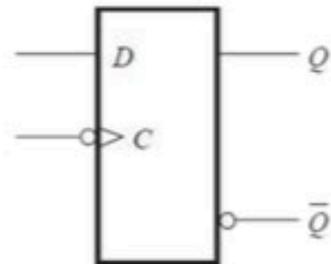
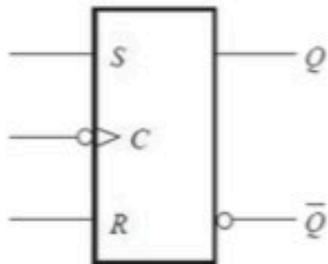
D



J-K



T





Flip-flops disparados por flanco

Entradas asíncronas de inicialización y borrado

- ❑ Los flip-flops con disparo por transición, se dice que sus entradas son entradas síncronas, ya que los datos de estas entradas condicionan la salida sólo durante el flanco de disparo del impulso de reloj; esto significa que los datos se transfieren sincronizados con la señal de reloj.

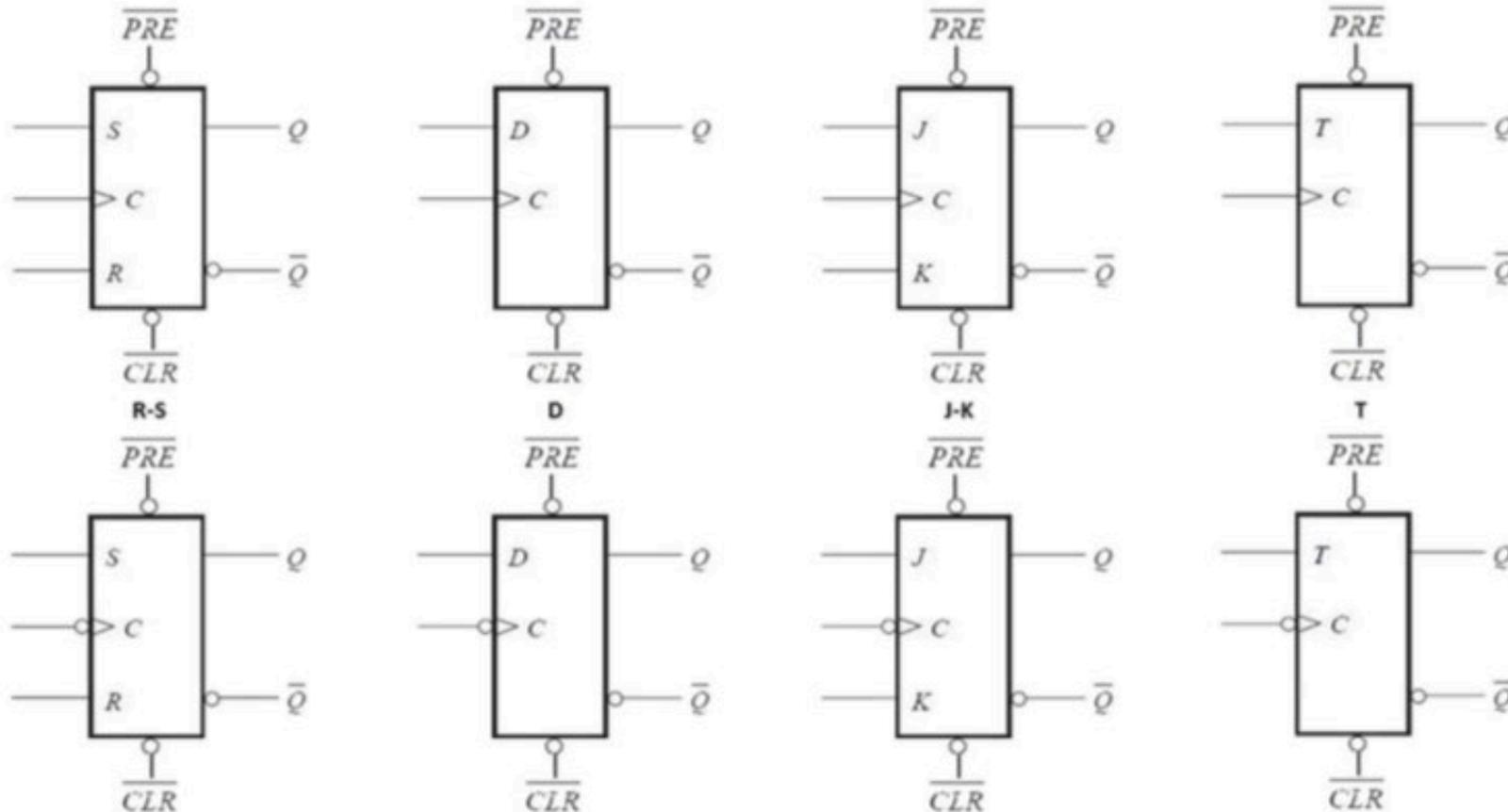
- ❑ La mayoría de los circuitos integrados flip-flops tienen también entradas asíncronas. Estas son entradas que pueden variar el estado del flip-flop independientemente del reloj. Generalmente, los fabricantes las denominan de inicialización:
 - Entrada de activación directa “preset” (PRE).
 - Entrada de desactivación directa ó borrado, “clear/reset” (CLR).

Resumen con inicialización asíncrona



Flip-flops disparados por flanco

Resumen tipos de Latches con entradas asíncronas (Preset y Reset)



Ecuaciones características



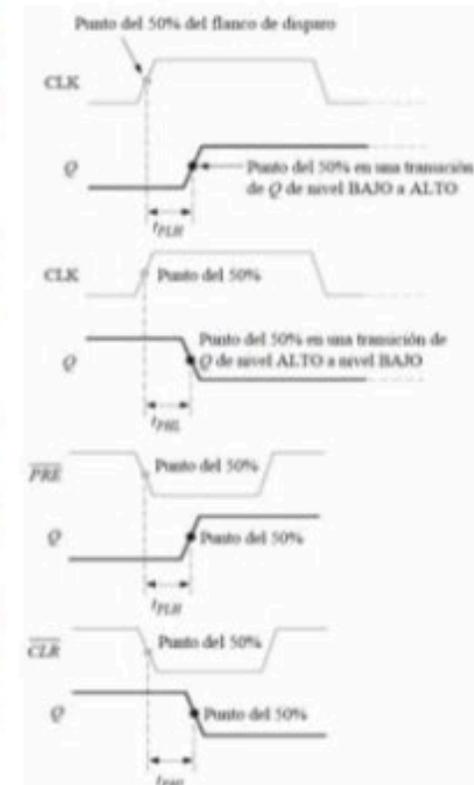
<i>Device Type</i>	<i>Characteristic Equation</i>
S-R latch	$Q^* = S + R' \cdot Q$
D latch	$Q^* = D$
Edge-triggered D flip-flop	$Q^* = D$
D flip-flop with enable	$Q^* = EN \cdot D + EN' \cdot Q$
Master/slave S-R flip-flop	$Q^* = S + R' \cdot Q$
Master/slave J-K flip-flop	$Q^* = J \cdot Q' + K' \cdot Q$
Edge-triggered J-K flip-flop	$Q^* = J \cdot Q' + K' \cdot Q$
T flip-flop	$Q^* = Q'$
T flip-flop with enable	$Q^* = EN \cdot Q' + EN' \cdot Q$



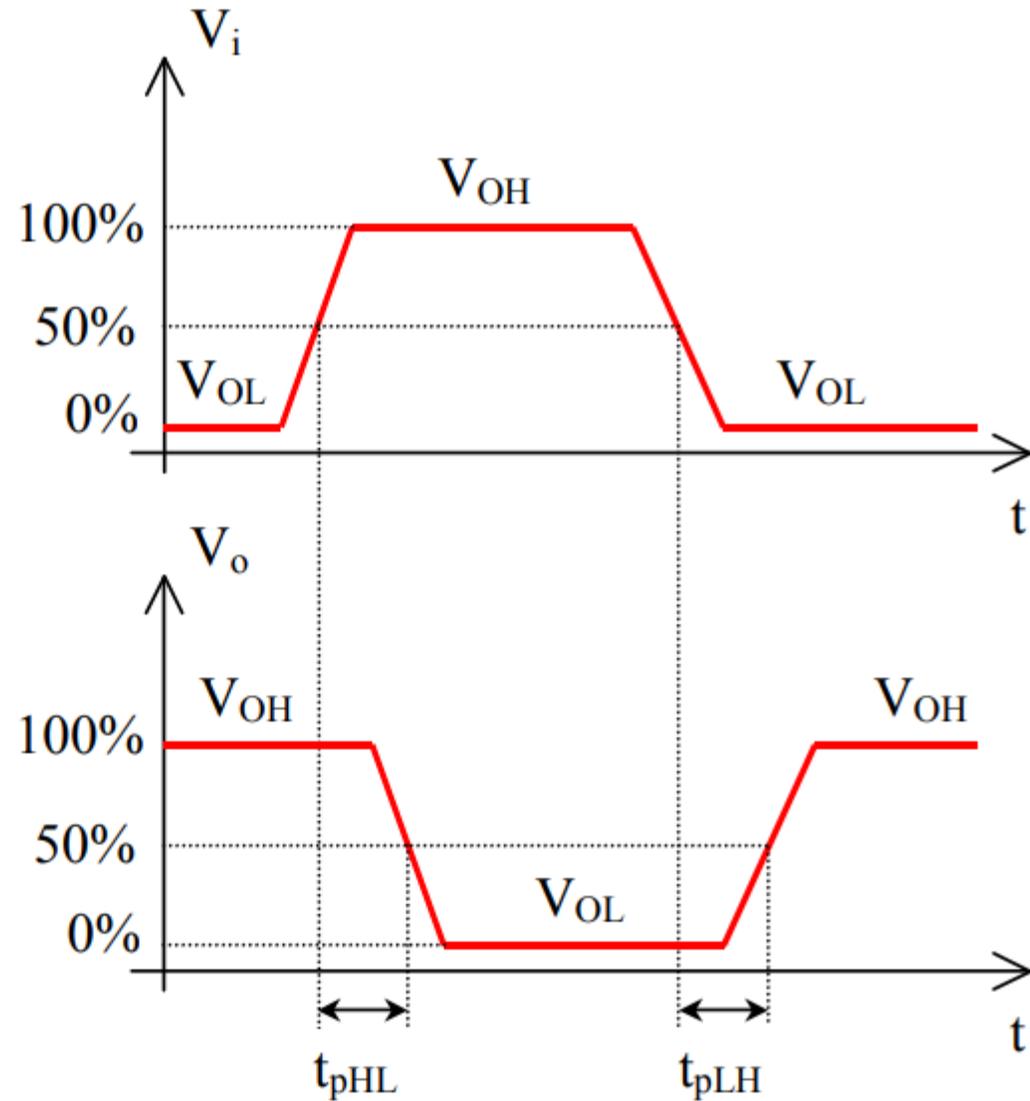
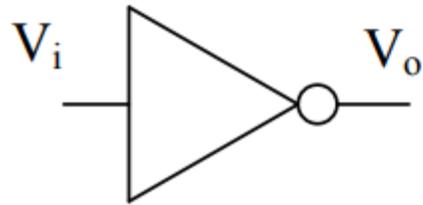
❑ Retardos de propagación.

Es el intervalo de tiempo requerido para que se produzca un cambio en la salida una vez que se ha aplicado una señal en la entrada.

- t_{PLH} se mide desde el flanco de disparo del impulso de reloj hasta la transición de nivel BAJO a nivel ALTO de la salida.
- t_{PHL} se mide desde el flanco de disparo de impulso del reloj hasta la transición de nivel ALTO a nivel BAJO de la salida.
- t_{PLH} medido desde la entrada de inicialización (*preset*) hasta la transición de nivel BAJO a nivel ALTO de la salida.
- t_{PHL} medido desde la entrada de borrado (*clear*) hasta la transición de nivel ALTO a nivel BAJO de la salida.



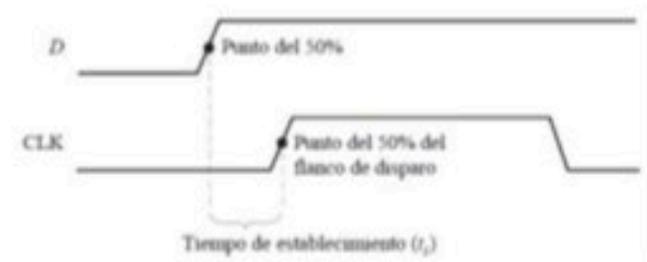
Propagación : Retardos





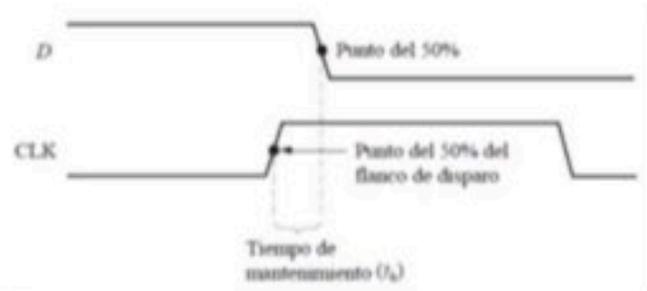
❑ Tiempo de establecimiento.

Setup time (t_s) es el intervalo mínimo que los niveles lógicos deben mantener constantes en las entradas (J y K , S y R o D) antes de que llegue el flanco de disparo del impulso de reloj, de modo que dichos niveles sincronicen correctamente en el flip-flop.



❑ Tiempo de mantenimiento.

Hold time (t_h) es el intervalo mínimo que los niveles lógicos deben mantenerse constantes en las entradas después de que haya pasado el flanco de disparo del impulso de reloj, de modo que dichos niveles se sincronicen correctamente en el flip-flop.





❑ Frecuencia máxima de reloj.

$f_{máx}$ es la mayor velocidad a la que se puede disparar el flip-flop de manera fiable. Para frecuencias del reloj por encima de la máxima, el flip-flop puede ser incapaz de responder lo suficientemente rápido y su funcionamiento se vería deteriorado.

❑ Anchura de impulsos.

Los fabricantes especifican la anchura mínima de los impulsos (t_w) para un funcionamiento adecuado de las entradas de reloj, inicialización y borrado. Típicamente, el reloj se especifica mediante sus intervalos de tiempo mínimo para los niveles alto y bajo.

❑ Disipación de potencia.

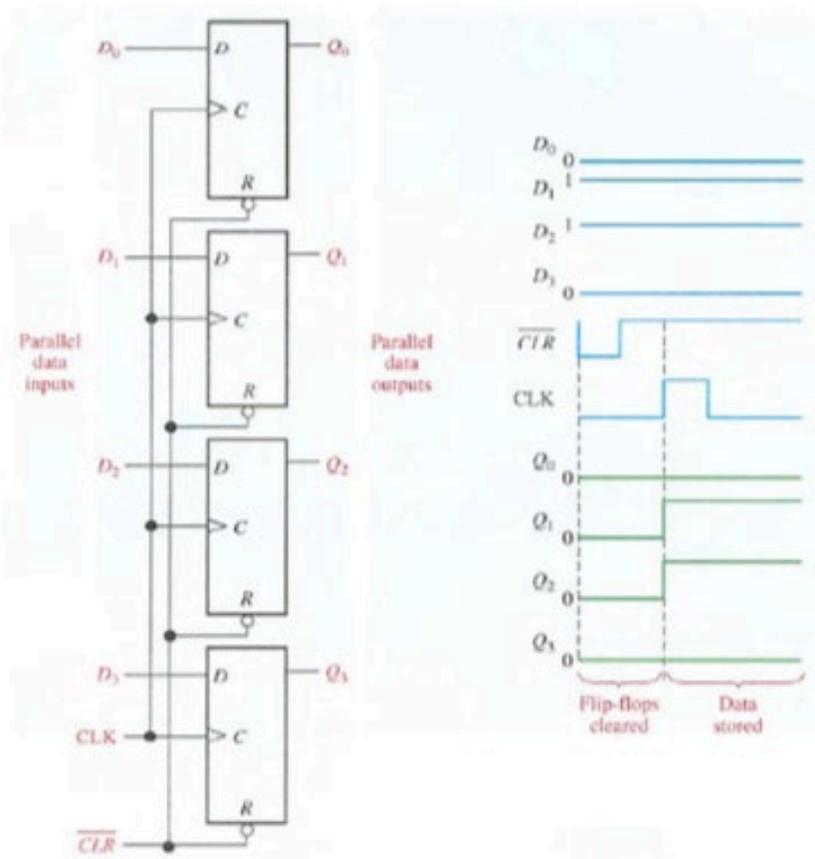
De cualquier circuito digital se define como la potencia total consumida por el dispositivo. Ejemplo:

$$P = V_{CC} \times I_{CC} = 5 \text{ V} \times 5 \text{ mA} = 25 \text{ mW}$$

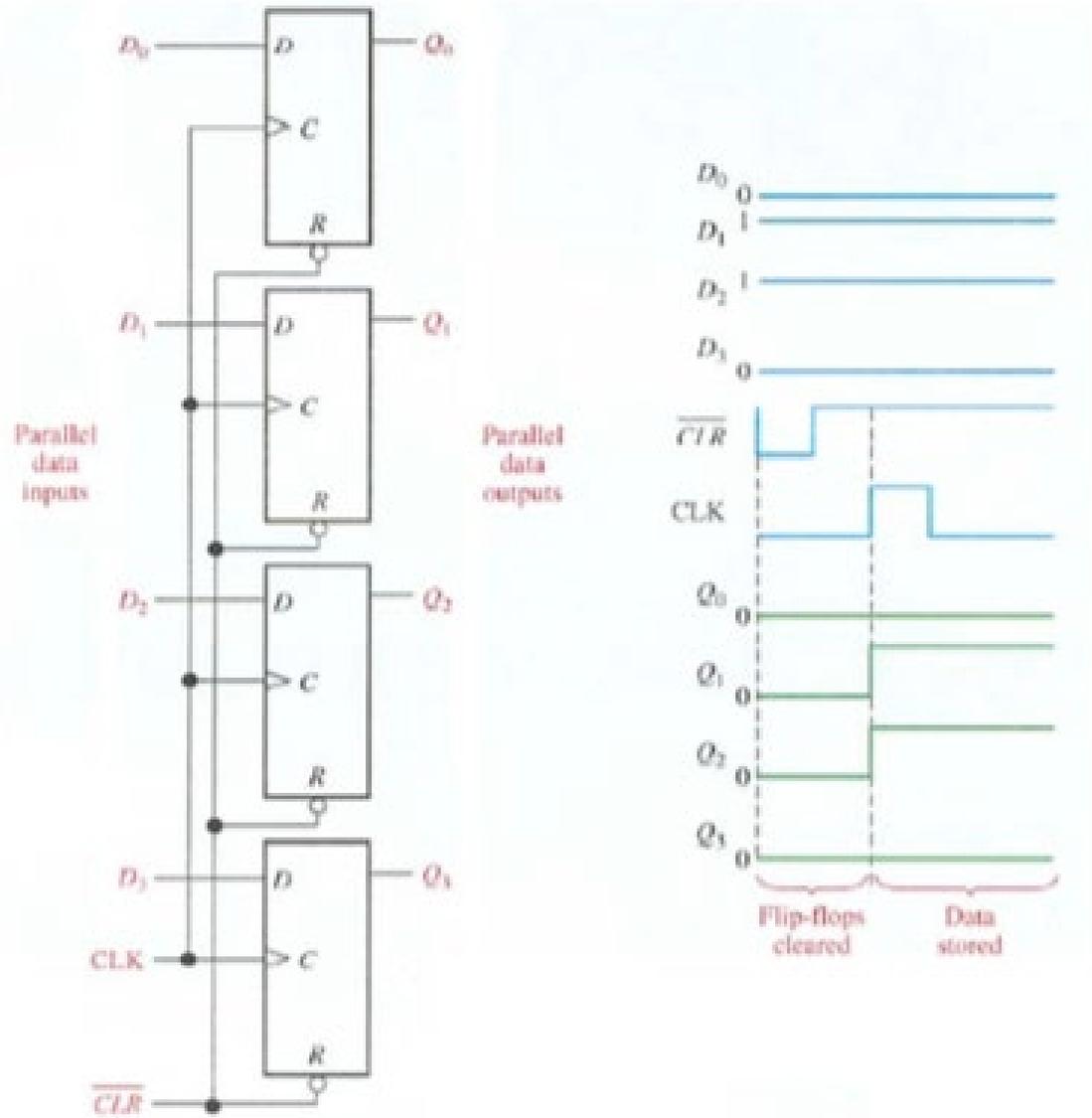


Almacenamiento de datos paralelo

- Uno de los requisitos más comunes de los sistemas digitales consiste en almacenar de forma simultánea una serie de bits de datos, procedentes de varias líneas paralelas, en un grupo de flip-flops.
- En los sistemas digitales, los datos se almacenan normalmente en grupos de bits (usualmente ocho ó múltiplos de ocho), que representa números, códigos u otras informaciones.



Aplicaciones típicas





Divisor de frecuencia

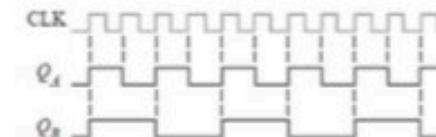
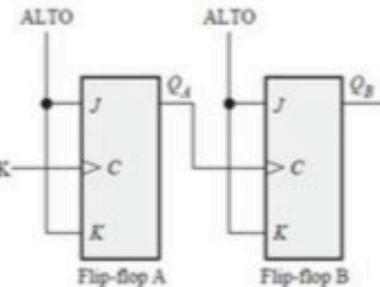
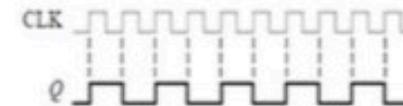
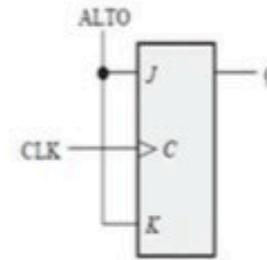
❑ División de frecuencia de una señal periódica.

- Cuando se aplica un tren de impulsos a la entrada de reloj de un flip-flop J-K (conectado en modo de basculación), la salida Q es una señal cuadrada que tiene una frecuencia igual a la que tiene la señal de reloj.

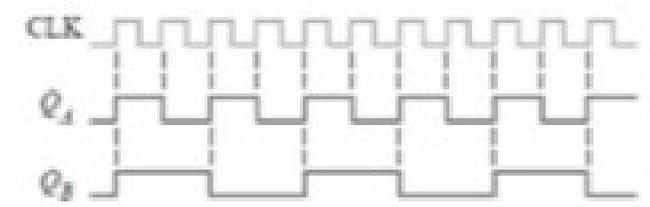
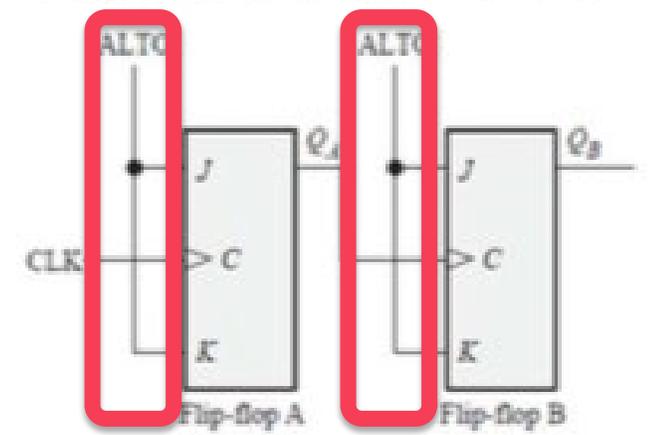
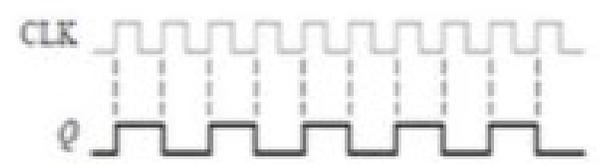
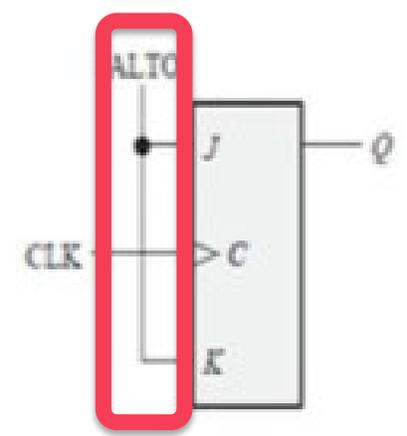


porque basculará todo el rato

- Se puede utilizar:
 - Un (1) único flip-flop como un divisor por dos (2).
 - Dos (2) flip-flops como un divisor por cuatro (4).

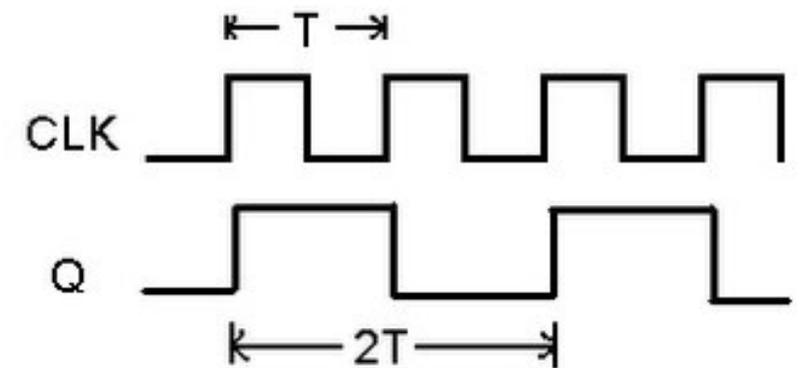
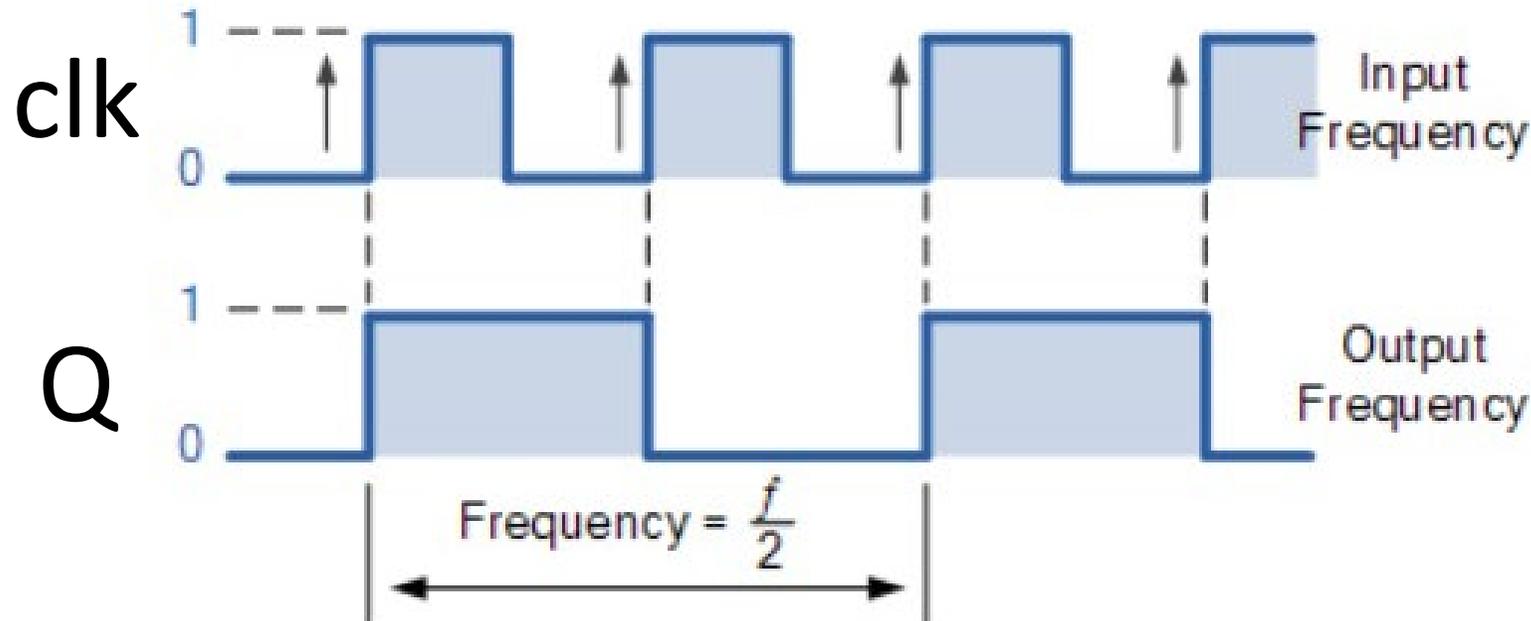


Clk	J	K	Q	\bar{Q}	Descripción
0	X	X	NC	NC	No Cambia.
↓	0	0	NC	NC	No Cambia. Se mantiene en el estado anterior.
↓	1	0	1	0	Salida "Q" en alto. Salida " \bar{Q} " en bajo.
↓	0	1	0	1	Salida "Q" en bajo. Salida " \bar{Q} " en alto.
↓	1	1	x	x	Toggle. Basculan las salidas.





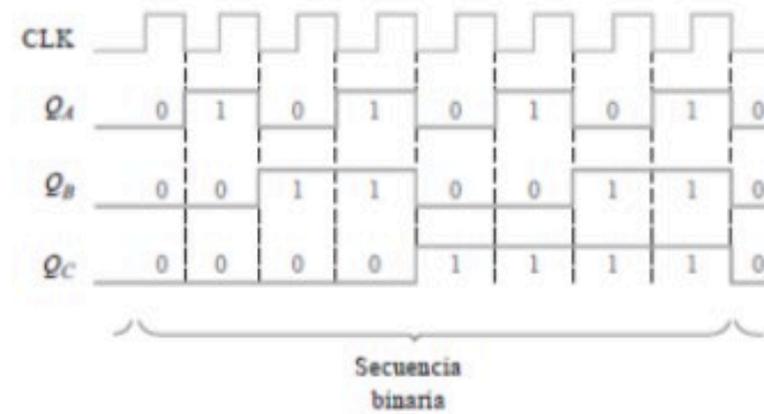
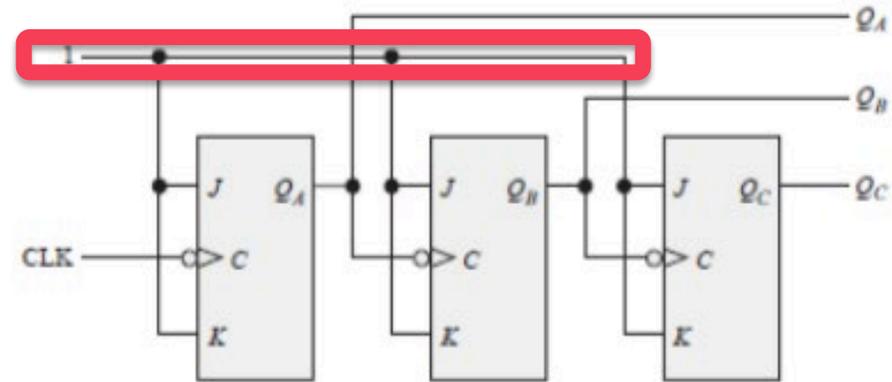
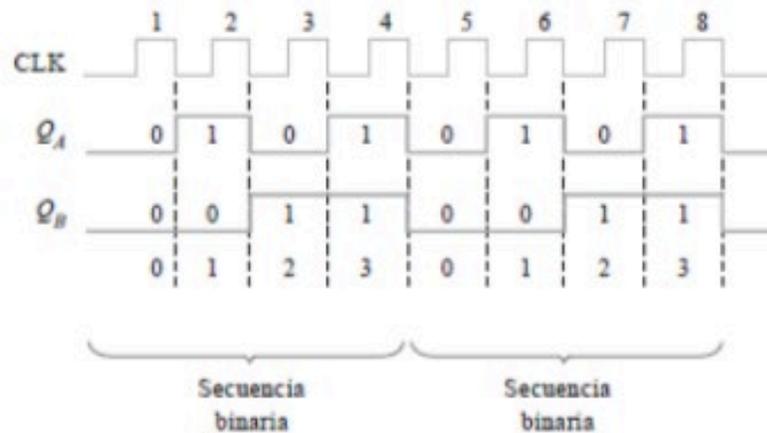
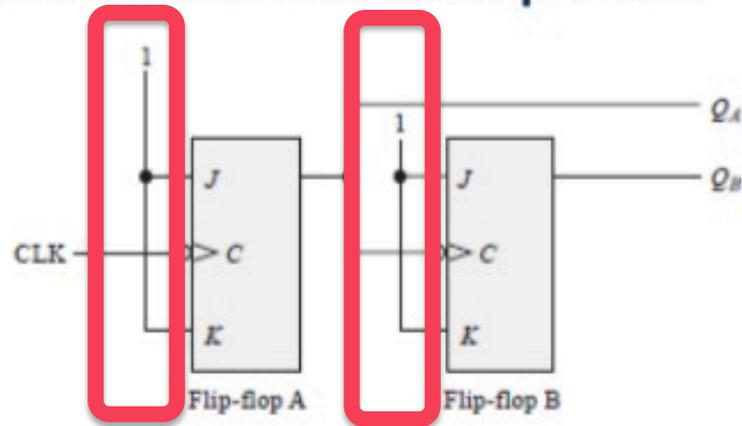
Cuando se aplica un tren de impulsos a la entrada de reloj de un J-K, la salida es una **señal cuadrada** cuya **frecuencia es la mitad que la de ese reloj**



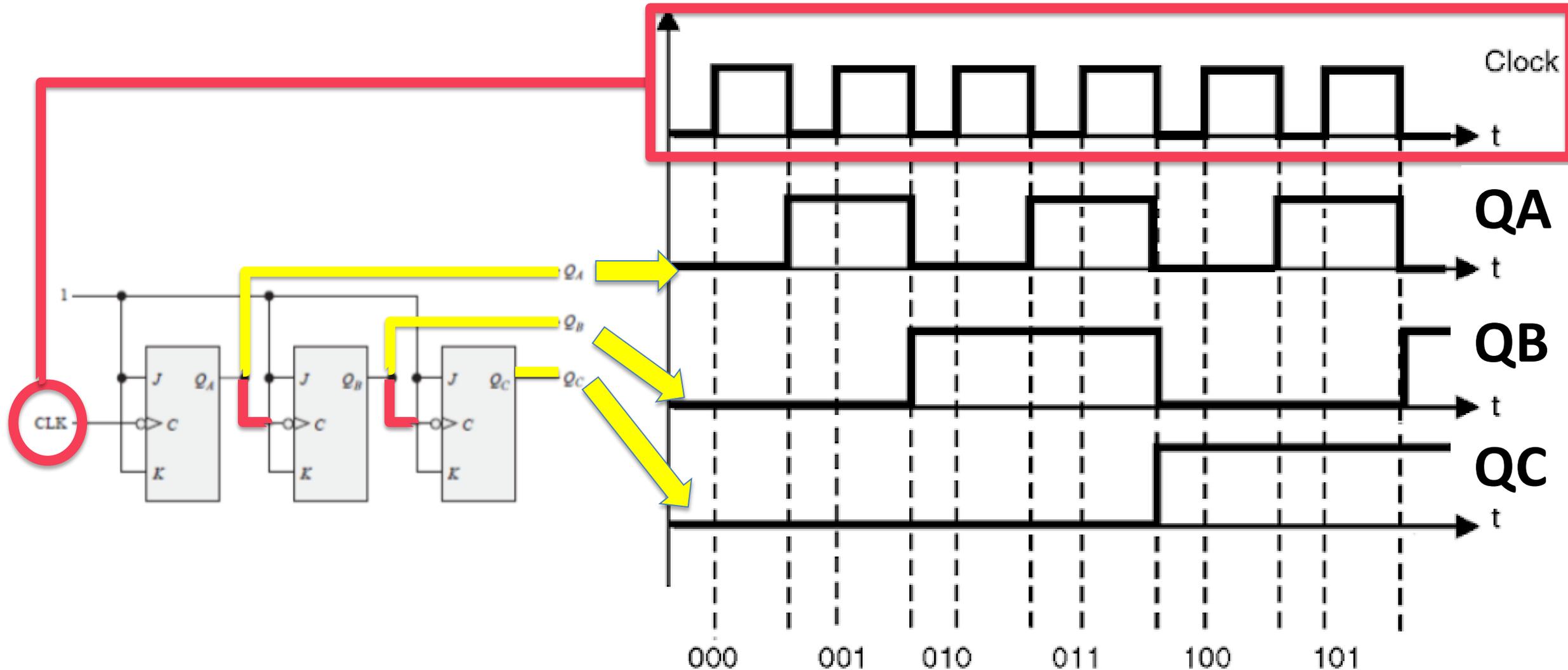


Contadores

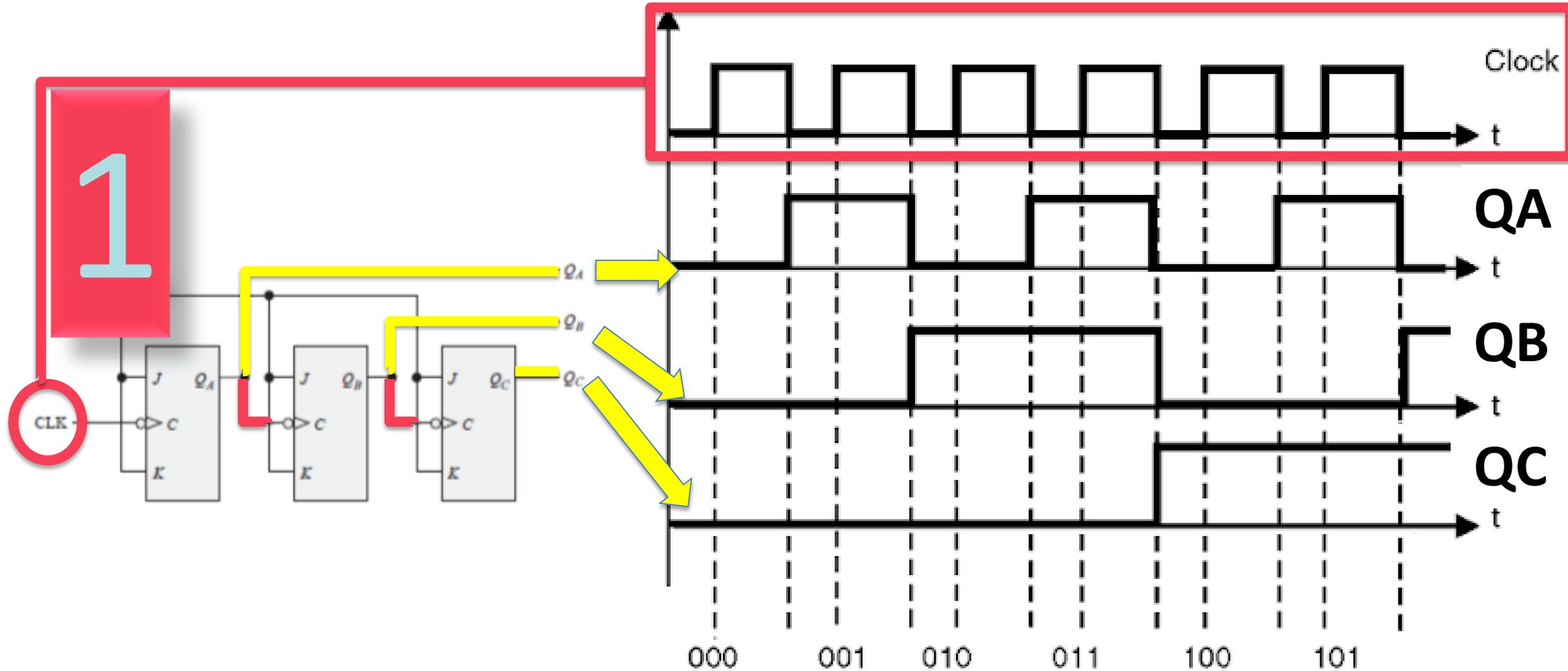
□ Conteo de eventos ó pulsos.



Aplicaciones típicas

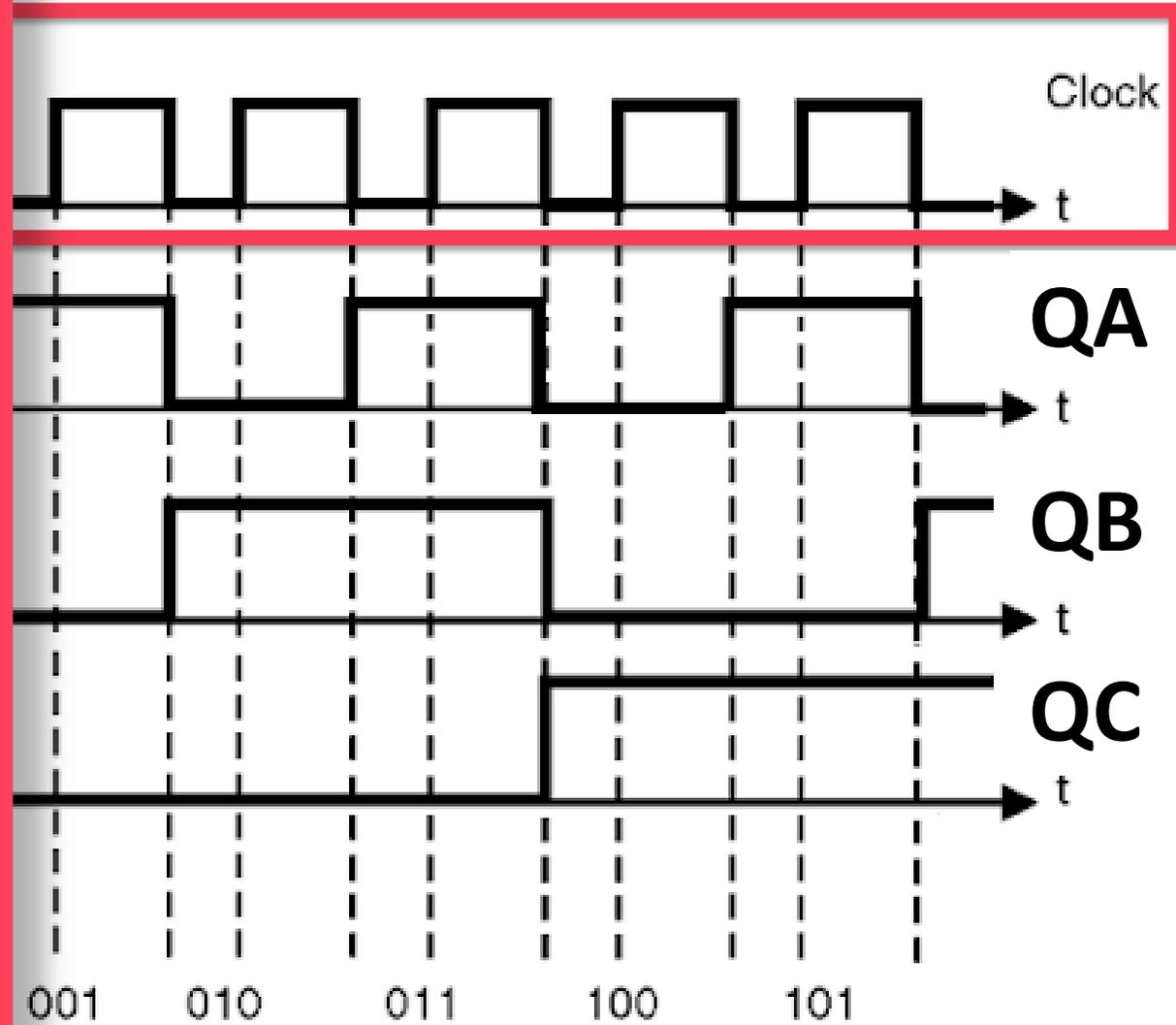


Aplicaciones típicas

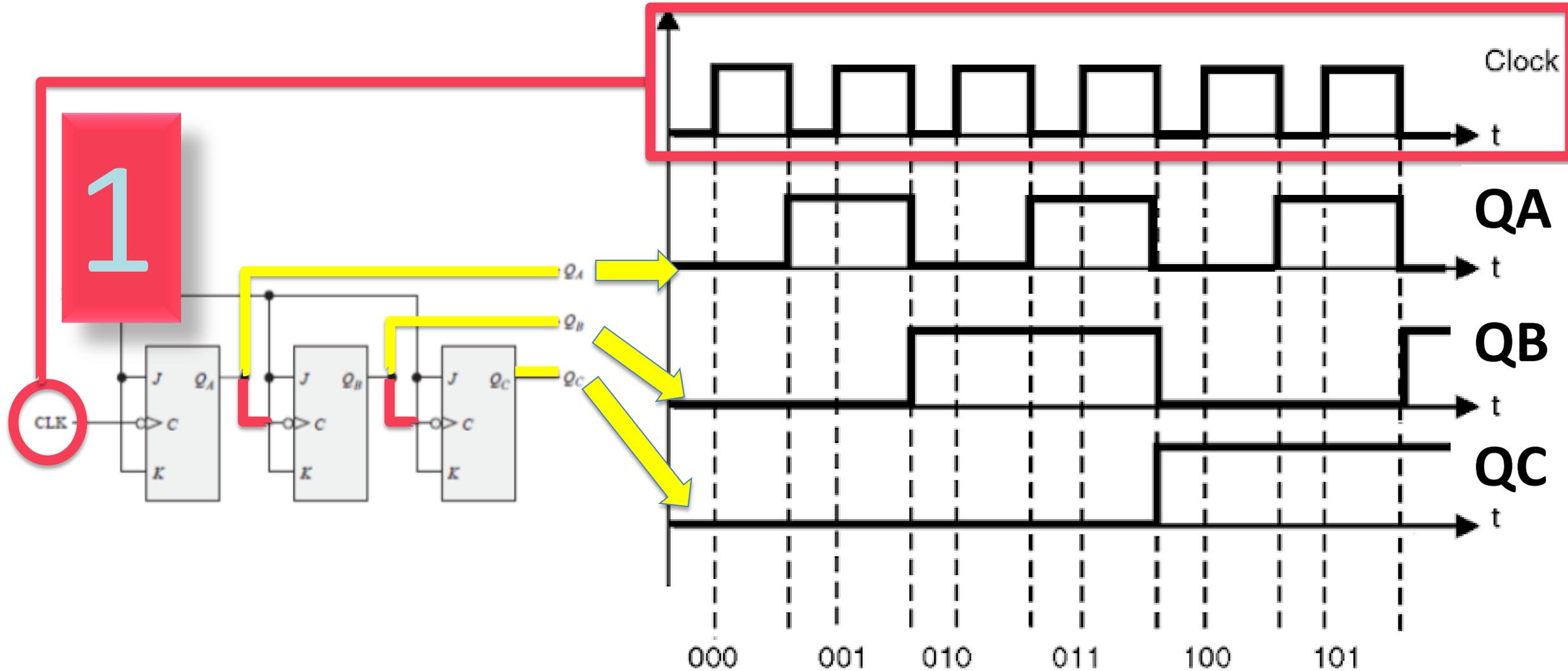




1

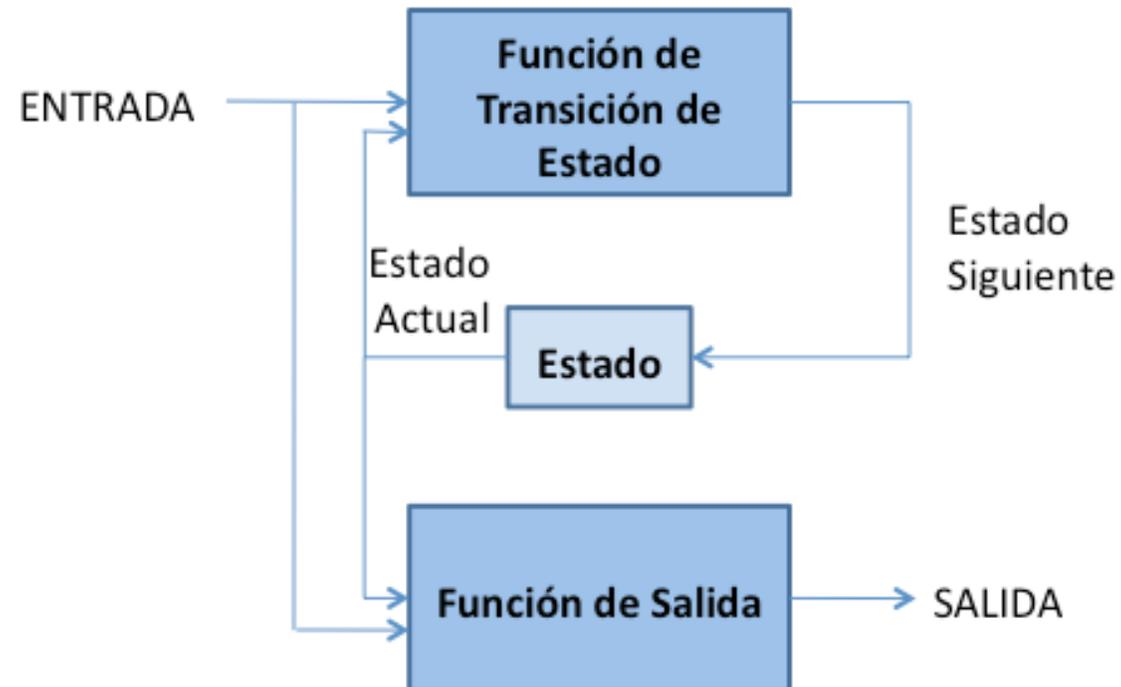


Aplicaciones típicas





- Esta formada por
 - **Dos bloques combinacionales**: uno implementa la función de transición de estado y el otro la función de salida
 - **Un registro** que almacena el estado del sistema.





- **Pasos para obtenerla**

1. Especificar el sistema mediante un diagrama de estados
2. Codificar toda la información en binario (PAR 0, IMPAR 1)
3. Obtener la tabla de verdad de las funciones de salida y transición de estados a partir del diagrama
4. Obtener la tabla inversa (o de excitación) de Biestables
5. Obtener funciones de salida
6. Obtener funciones de estado
7. Implementar el circuito

¡SIMPLIFICAR TODAS LAS FUNCIONES COMBINACIONALES!



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

R-S latch

Clk	S	R	Q	Q	Q+	S	R
				0	0	0	X
↑	1	0	1	0	1	1	0
↑	0	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

Clk	J	K	Q
0	X	X	NC
↓	0	0	NC
↓	1	0	1
↓	0	1	0
↓	1	1	x

D flip-flop

Q	Q+	D
0	0	0
0	1	1
1	0	0
1	1	1

Si D=1 -> S=1
Si D=0 -> S=0

T flip-flop

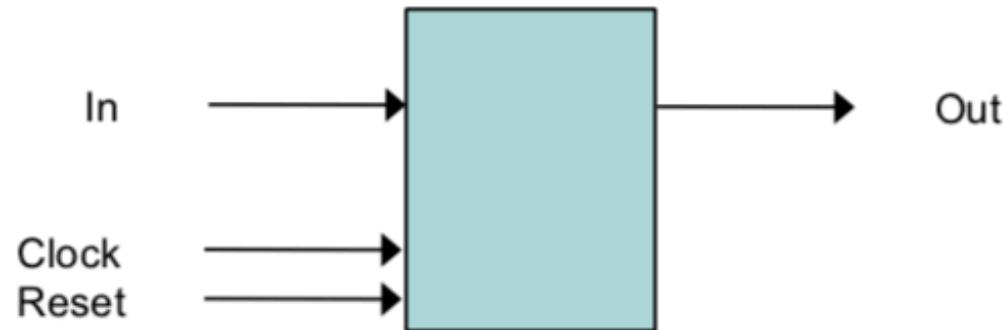
Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

Si T=1 -> S=CAMBIA

Ejercicio con ejemplos



- 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 de reloj
 - La salida se activa cuando se detecta la secuencia

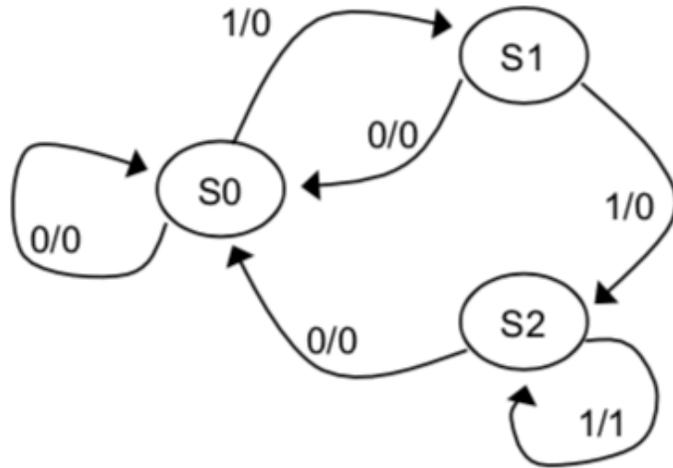


- Ejemplo de secuencia de entradas y salidas:
 - X : 0 0 1 1 0 1 1 1 1 0 0 1 1 1
 - Z : 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1



- **Ejemplo 1: Mealy con biestables D:**

1. Diagrama de estados:



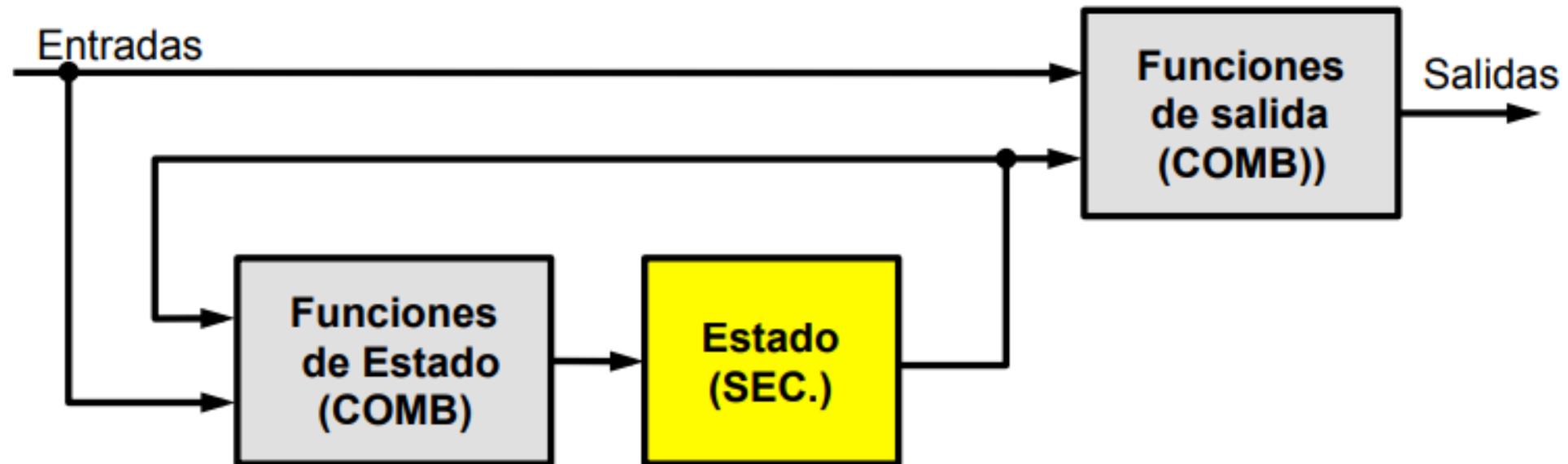
2. Codificación de estados:

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



Estado no alcanzable

- En el modelo de Mealy las salidas dependen tanto de los estados como de las entradas (caso general)
- Máquina de estados de Mealy:
$$Z(t) = F(X(t), S(t))$$
$$S(t+1) = G(X(t), S(t))$$
- Estructura de un circuito asociado a un modelo de Mealy:

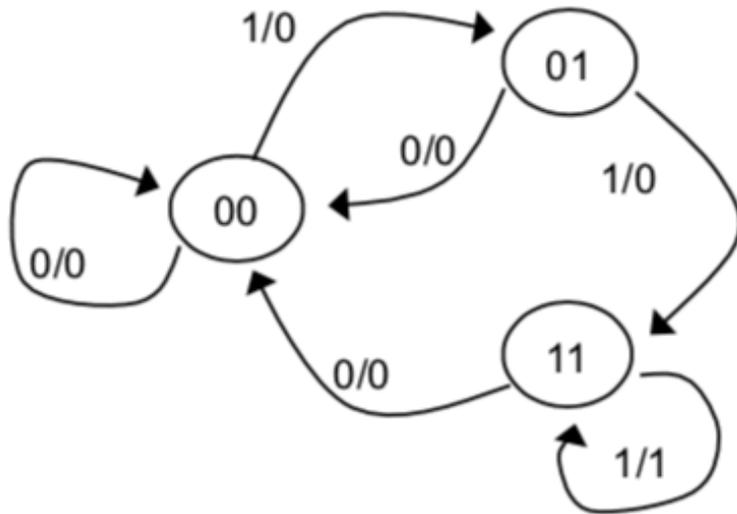


- En las máquinas de estados de Mealy las salidas pueden cambiar en cualquier momento (basta con que cambie una entrada del circuito):
 - Las salidas **no** están sincronizadas con el reloj
 - NOTA: Aunque las salidas no estén sincronizadas con el reloj, el circuito sigue siendo síncrono (todos los biestables están sincronizados con el mismo reloj)



- Mealy con biestables D :

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



3 ENTRADAS:

2 para codificar estados y la entrada del sistema

3 SALIDAS:

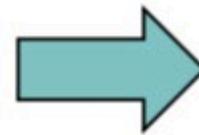
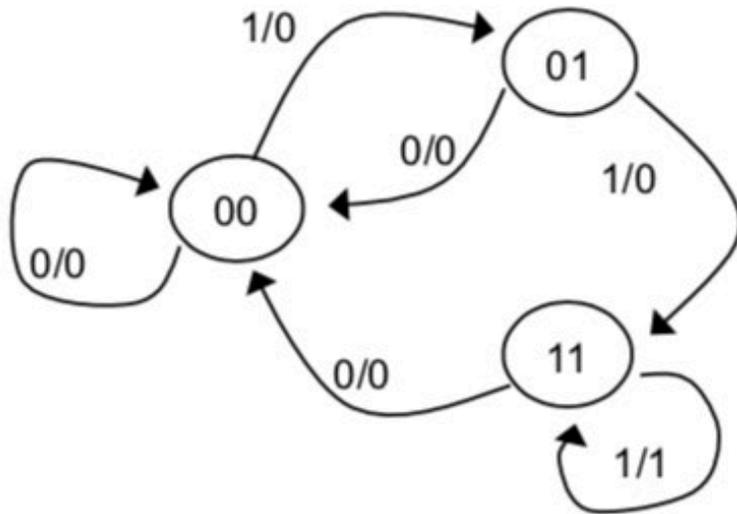
2 para estado siguiente y la salida del sistema

Ejercicio (3)



- Mealy con biestables D :

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



transición de estados

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

No hay estado 10

Entra 1 -> S1 (01)

Entra 1 -> S2 (11)

No hay estado 10

Entra 1 -> S2 (11)

Y salida 1 (tres 1s)

Ejercicio (4)



- Mealy con biestables 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

D flip-flop

Q	Q+	D
0	0	0
0	1	1
1	0	0
1	1	1

Si D=1 -> S=1

Si D=0 -> S=0

5. Función de salida:

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

$$Out = Q_1 In$$

6. Funciones de estado

In	Q1	Q0	D1		
	00	01	11	10	
0				X	
1			1	1	X

$$D_1 = Q_0 In$$

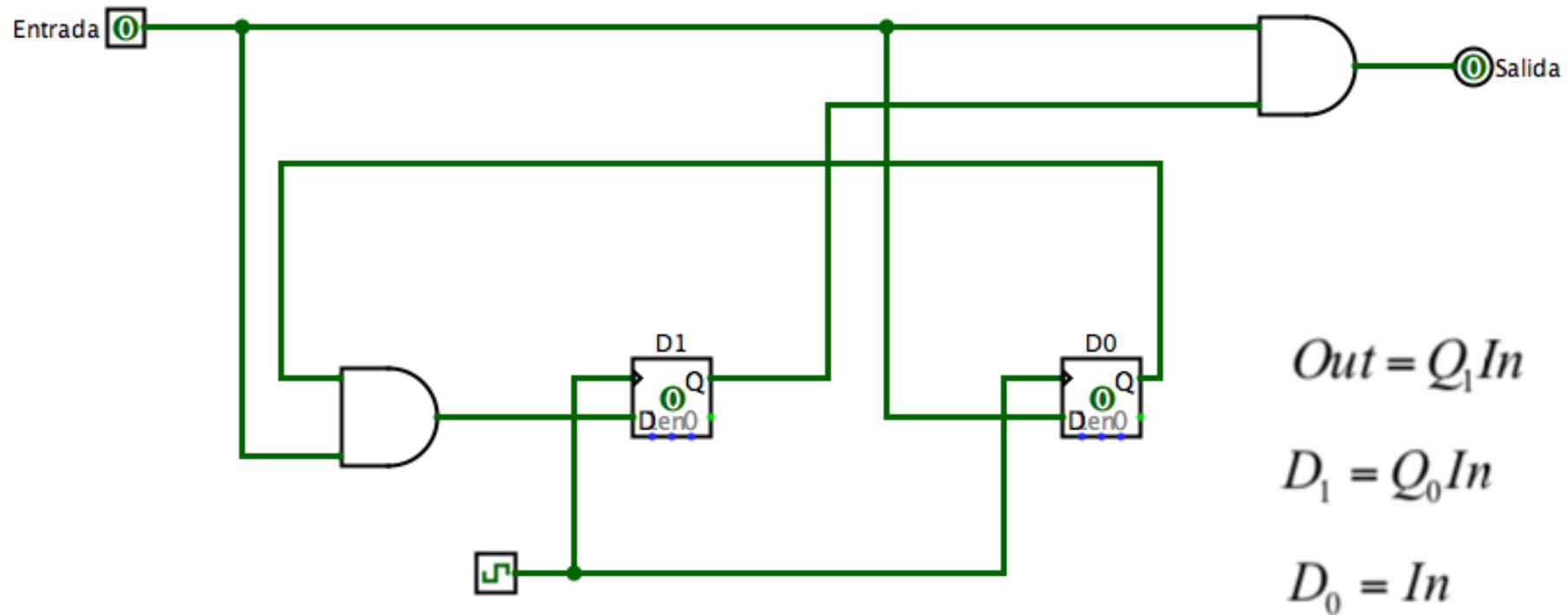
In	Q1	Q0	D0			
	00	01	11	10		
0				X		
1			1	1	1	X

$$D_0 = In$$

¡SIMPLIFICAR TODAS LAS FUNCIONES COMBINACIONALES!



7. Implementación



Ejercicio (6)



● Ejemplo 2. Mealy con biestables J-K :

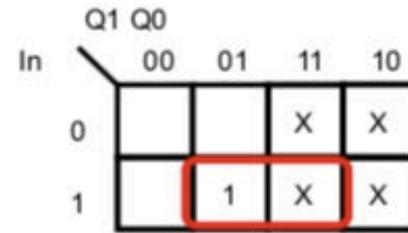
4. Tabla inversa de biestables (biestables J-K):

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

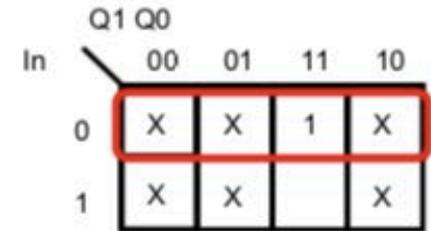
5. Función de salida:

$$Out = Q_1 In$$

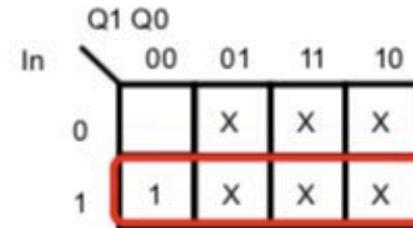
6. Funciones de estado



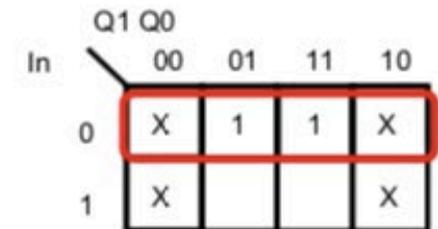
$$J_1 = Q_0 In$$



$$K_1 = \overline{In}$$



$$J_0 = In$$



$$K_0 = \overline{In}$$

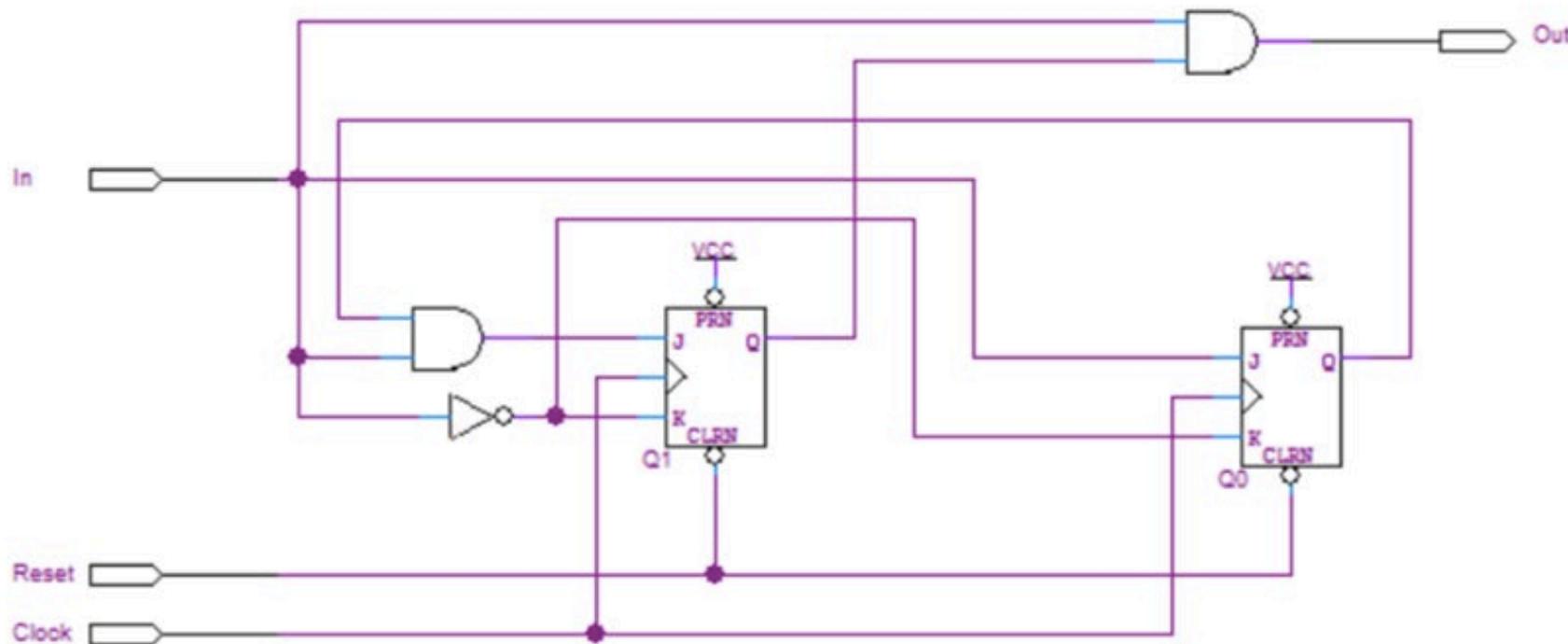
¡SIMPLIFICAR TODAS LAS FUNCIONES COMBINACIONALES!

Ejercicio (7)



- Mealy con biestables J-K:

7. Implementación



$$\begin{aligned} Out &= Q_1 In \\ J_0 &= In \\ K_0 &= \overline{In} \\ J_1 &= Q_0 In \\ K_1 &= \overline{In} \end{aligned}$$

Ejercicio (8)



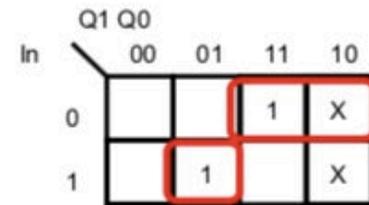
● Ejemplo 3. Mealy con biestables T :

4. Tabla inversa de biestables (biestables T):

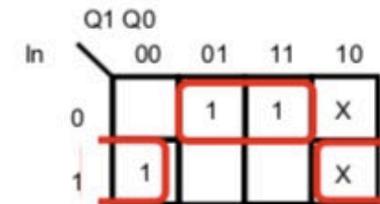
In	Q1	Q0	Q1+	Q0+	Out	T1	T0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	X	X	X	X	X
0	1	1	0	0	0	1	1
1	0	0	0	1	0	0	1
1	0	1	1	1	0	1	0
1	1	0	X	X	X	X	X
1	1	1	1	1	1	0	0

5. Función de salida: $Out = Q_1 In$

6. Funciones de estado



$$T_1 = Q_1 \bar{In} + \bar{Q}_1 Q_0 In$$



$$T_0 = \bar{In} Q_0 + In \bar{Q}_0 = In \oplus Q_0$$

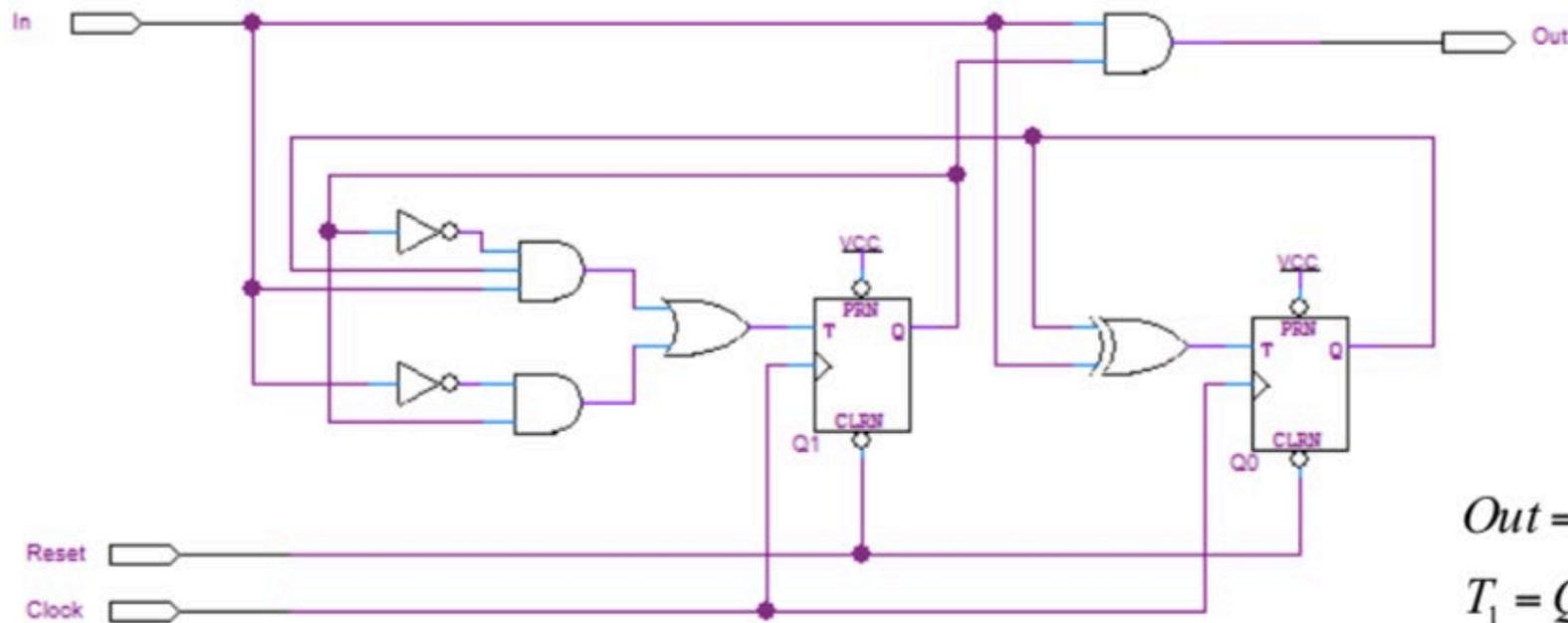
¡SIMPLIFICAR TODAS LAS FUNCIONES COMBINACIONALES!

Ejercicio (9)



- Mealy con biestables T:

7. Implementación



$$Out = Q_1 In$$

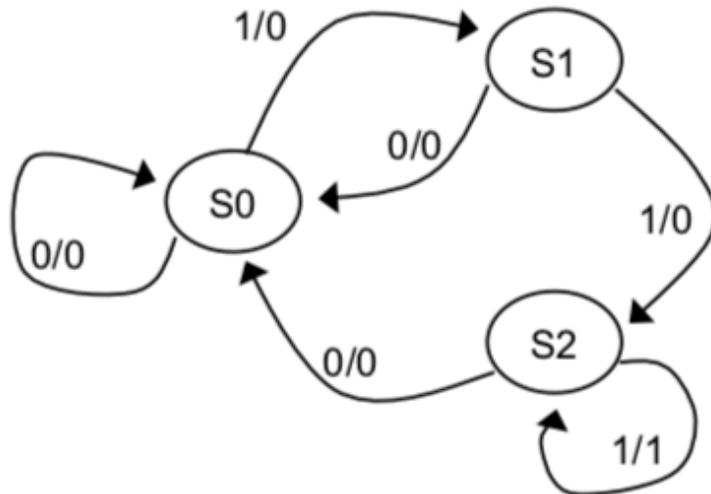
$$T_1 = Q_1 \overline{In} + \overline{Q_1} Q_0 In$$

$$T_0 = \overline{In} Q_0 + In \overline{Q_0} = In \oplus Q_0$$



- **Ejemplo 4:** Mealy, otra codificación diferente:

1. Diagrama de estados:



2. Codificación de estados:

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

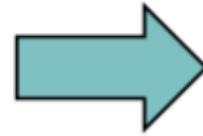
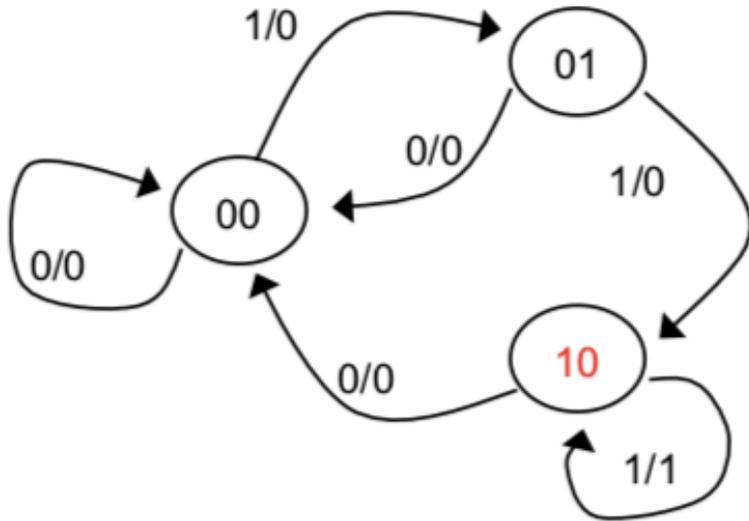


Ahora codificamos S2 de forma diferente

Ejercicio (11)



- Mealy con biestables D (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

Ejercicio (12)



- Mealy con biestables 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:

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

$$Out = Q_1 In$$

6. Funciones de estado

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

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

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

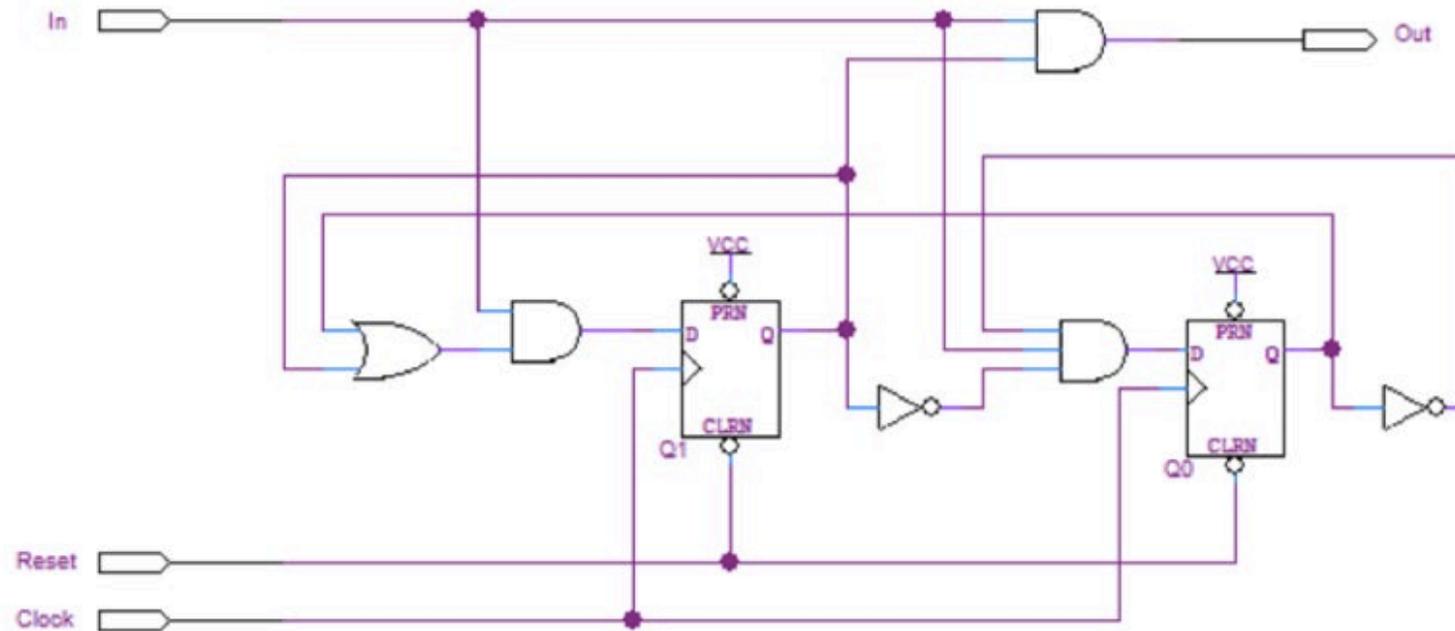
$$D_0 = \overline{Q_1} \overline{Q_0} In$$

Ejercicio (13)



- Mealy con biestables D (codificación diferente):

7. Implementación



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

$$Out = Q_1 In$$

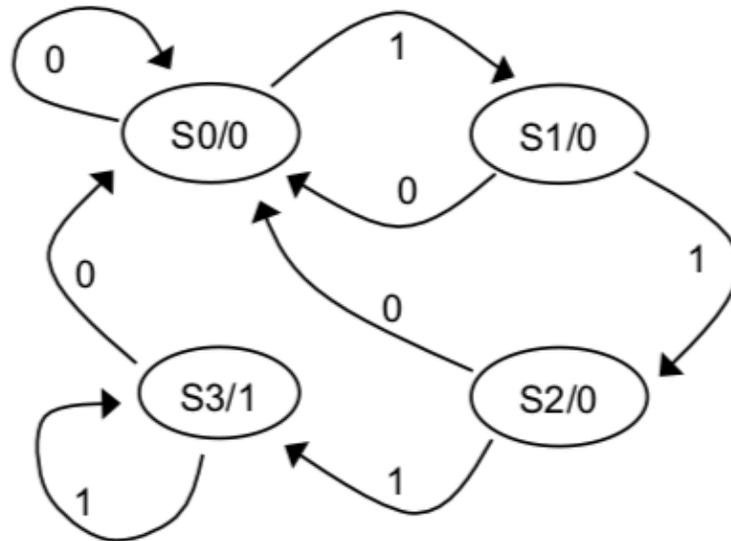
$$D_1 = In(Q_0 + Q_1)$$

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



- **Ejemplo 5: Moore con biestables D:**

1. Diagrama de estados:



2. Codificación de estados:

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

Ejercicio (14)

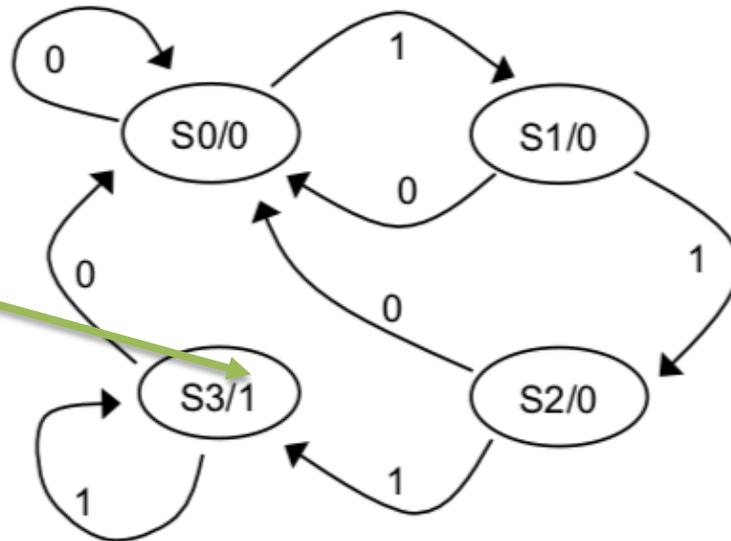


- **Ejemplo 5: Moore con biestables D:**

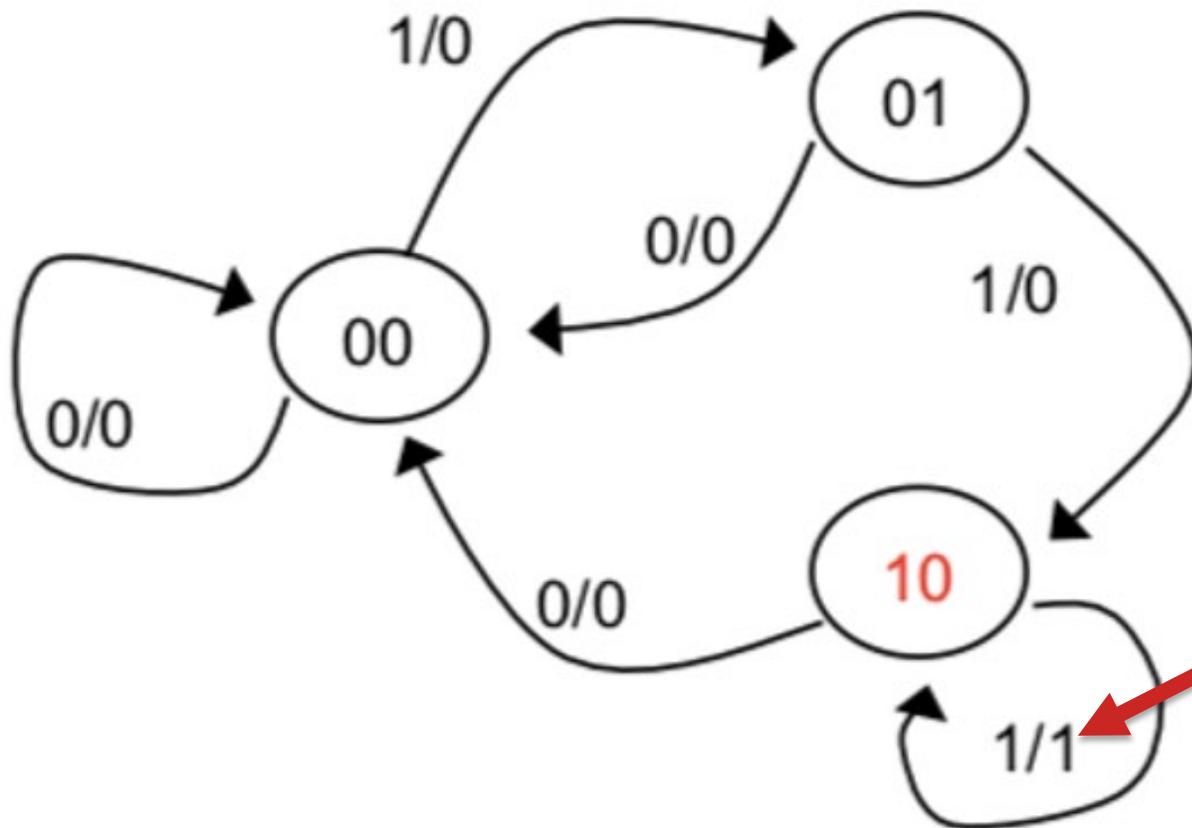
1. Diagrama de estados:

2. Codificación de estados:

La salida a 1, con tres 1s, la tenemos que meter en un estado. Por eso necesitamos un estado más.

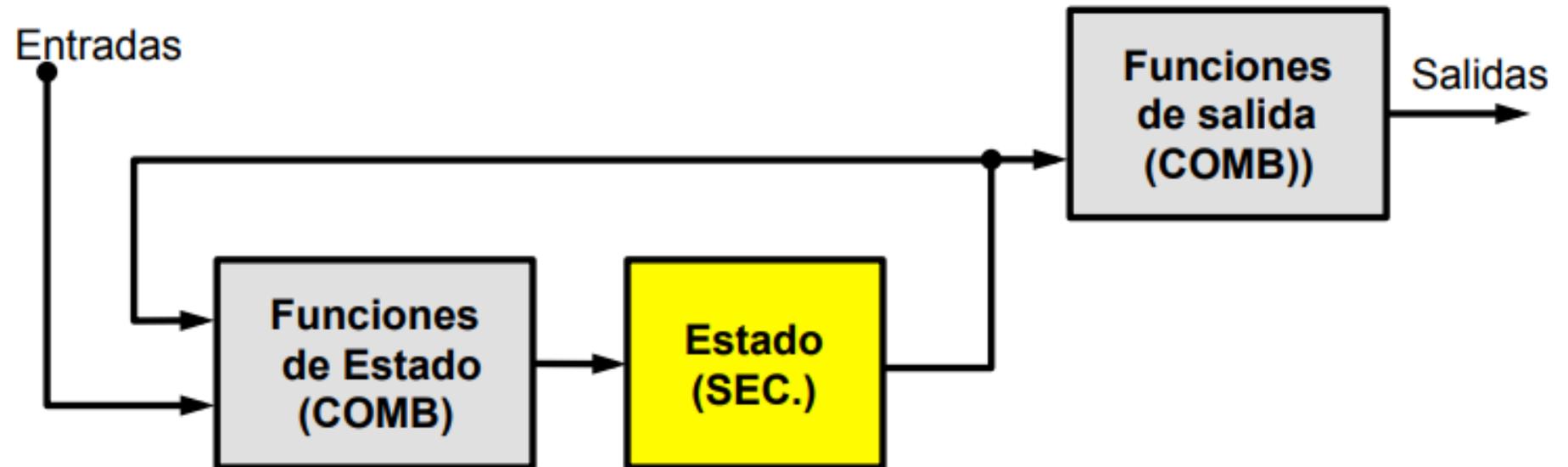


Estado	Q1	Q0
S0	0	0
S1	0	1
S2	1	1
S3	1	0

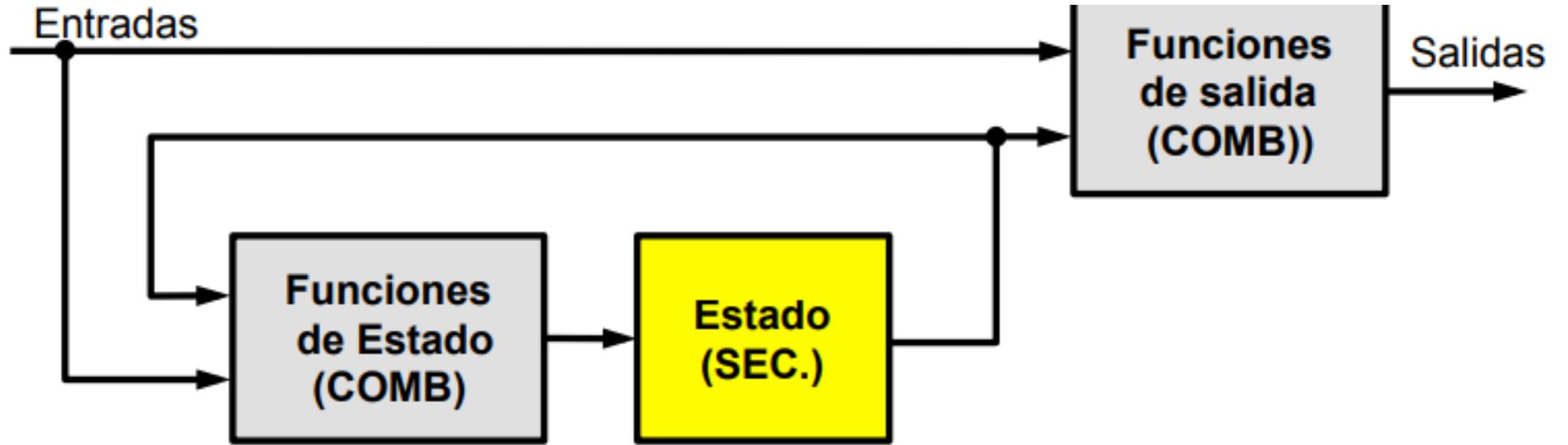


**En Mealy ya la
tenemos en la
transición**

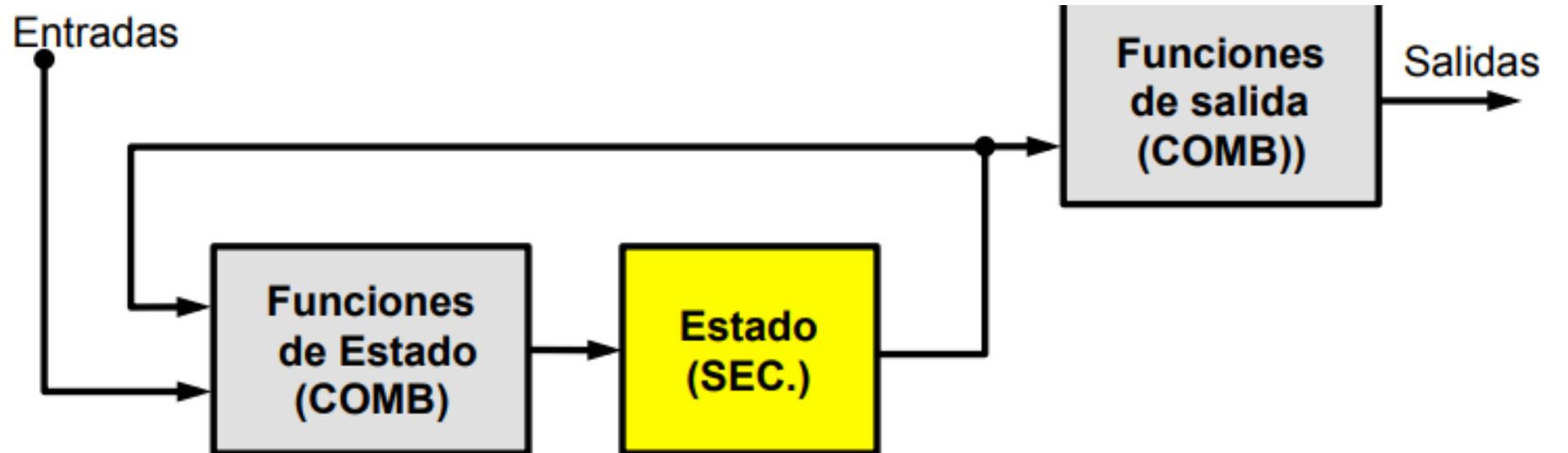
- En el modelo de Moore las salidas dependen únicamente de los estados (no de las entradas)
 - En las máquinas de estados de Moore las salidas cambian únicamente si hay un cambio de estado:
 - Las salidas están sincronizadas con el reloj
- $$Z(t) = F(S(t))$$
- $$S(t+1) = G(X(t), S(t))$$
- Estructura de un circuito asociado a un modelo de Moore:



Mealy



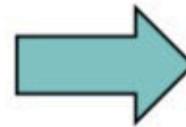
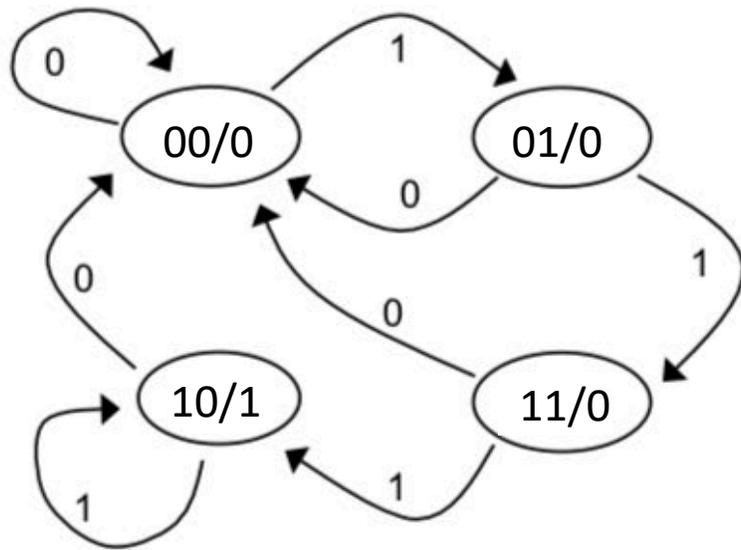
Moore



Ejercicio (15)



- Moore con biestables D :
 3. Tablas de transiciones y salidas:



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

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

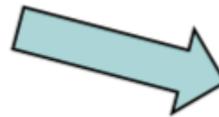
Ejercicio (16)



- Moore con biestables D :

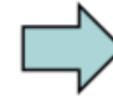
4. Tabla inversa de biestables (biestables D):

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	1
1	1	0



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

6. Funciones de estado

Q1 Q0

In	00	01	11	10
0				
1		1	1	1

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

Q1 Q0

In	00	01	11	10
0				
1	1	1		

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

¡SIMPLIFICAR TODAS LAS FUNCIONES COMBINACIONALES!

Ejercicio (17)



Moore con biestable D : 7. Implementación (Corregida)

