Electrónica Digital

Tema 7: **Circuitos Digitales.**

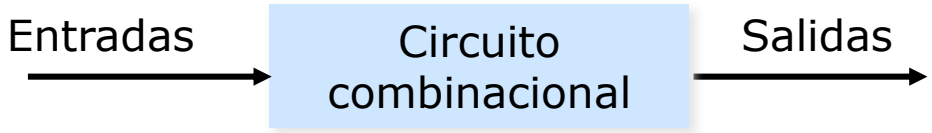
Sistemas Secuenciales: Biestables, Registros y Contadores. (3/3)

Sistemas Secuenciales
Índice

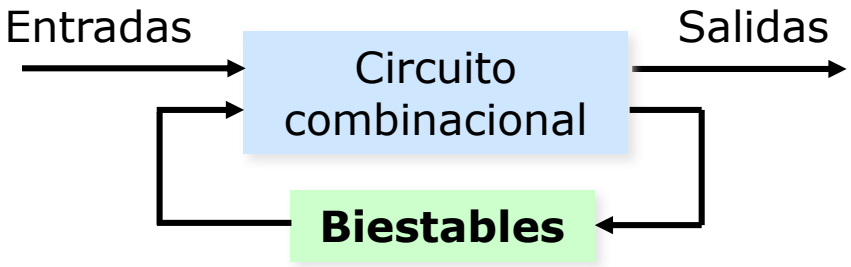
- Introducción
  - Circuitos digitales con Memoria
- Biestables
  - Elementos de Memoria elementales. Nociones y algunos tipos.
- Registros
  - Elementos de memoria compuestos.
- Contadores
  - Circuitos de cuenta y temporización
- Sistemas secuenciales
  - Máquinas de Estado

Biestables
Introducción

- Tipos de sistemas digitales:
  - **Combinacionales:** aquellos circuitos cuyas salidas, en un determinado instante, son función exclusivamente del valor de las entradas en ese instante. Entradas iguales dan lugar a las mismas salidas.

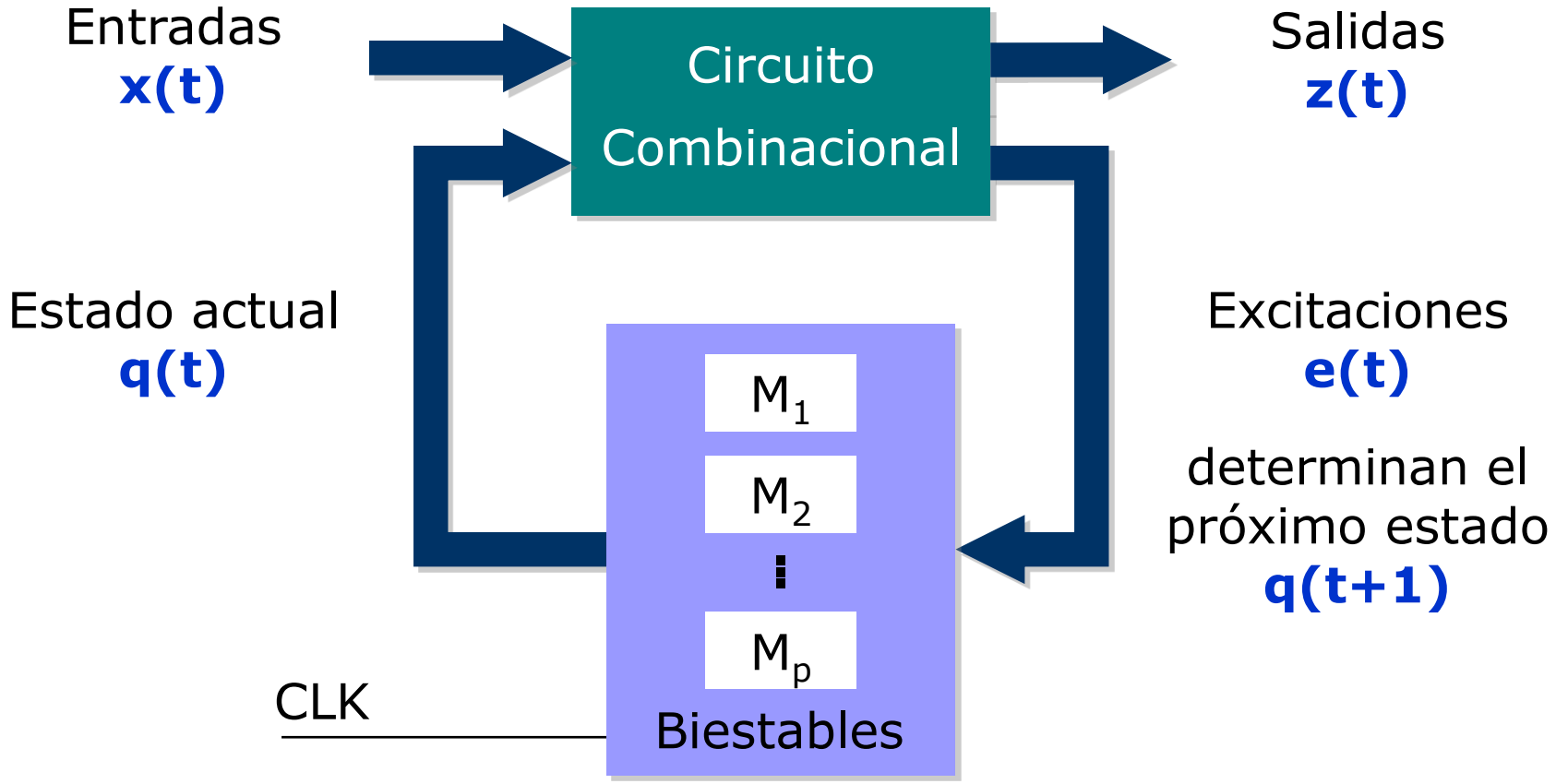


  - **Secuenciales:** aquellos circuitos en los que las salidas dependen de las entradas en el instante actual y en los anteriores. **Tienen memoria**, organizada mediante elementos básicos **biestables**. Entradas iguales pueden generar salidas distintas.



Autómatas de estados finitos

Definición y Nomenclatura



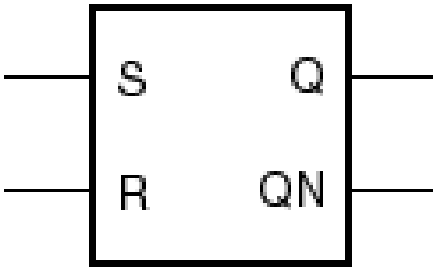
El estado a "grosso modo" será cierta información guardada a través de una determinada combinación de salida de los biestables

p elementos de memoria (biestables) ➡ **2^p** estados (máximo)

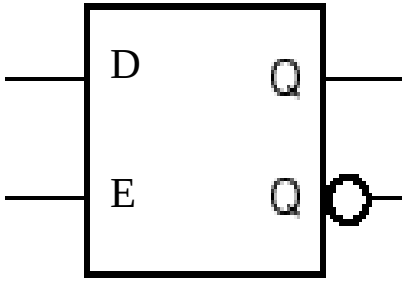
Biestables
Introducción

- Circuito lógico que:
  - Puede mantener indefinidamente un estado lógico (siempre y cuando el dispositivo esté alimentado)
  - Permite modificar su estado externamente
- Clasificación según sincronía:
  - Biestables asíncronos: Los cambios en la salida se producen siempre que hay un cambio en la entrada
  - Biestables síncronos: Los cambios en la salida se producen cuando se active la señal de sincronismo (reloj → suele ser una señal cuadrada):
    - Por nivel (latch), reloj=*enable*
    - Por flanco (flip-flops)
- Clasificación por funcionalidad:
  - R-S
  - D
  - J-K

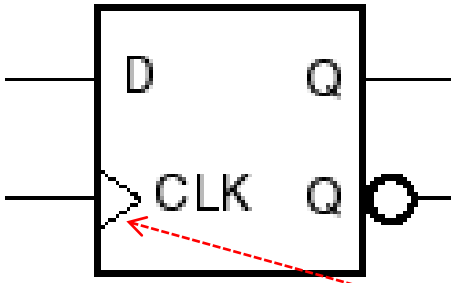
- Simbología general y nomenclatura:



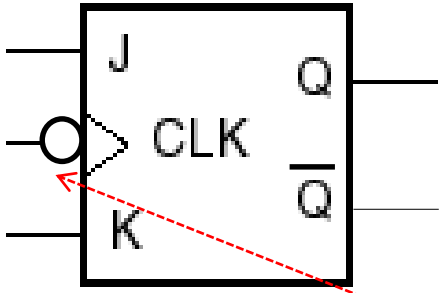
R-S asíncrono



D síncrono,
activo por nivel (alto)



D síncrono,
activo por flanco ascendente: ⌐↑



J-K síncrono,
activo por flanco descendente: ↓¬

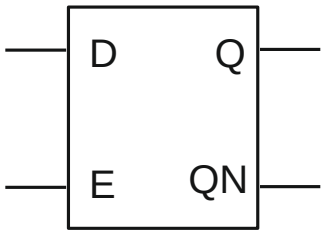
Nota.- Distintas representaciones de la salida Q invertida (QN, $\overline{Q}$, Q⊳–)

Biestables
Introducción

- Para determinar el próximo nivel a la salida (Q) de un biestable, será necesario:
  - Saber el valor de TODAS sus entradas, que se extrae de los datos del circuito a partir de:
    - cronogramas (o tablas) que describen la evolución temporal de las señales de entradas,
    - expresiones algebraicas (J=1, K=A⊕B …)
  - Conocer cuál era el estado previo (Q) del biestable
- En un biestable: entradas iguales podrán generar salidas distintas, dependiendo del estado previo (Q) del mismo

Biestables
Biestable D activo por nivel (D LATCH)

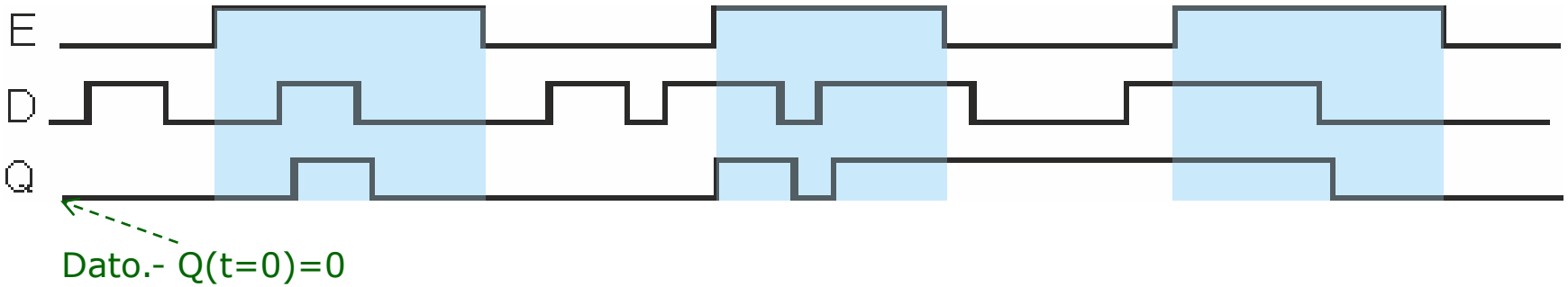
- La salida Q sigue a la entrada D validada con E



E	D	Q_{t+1}	QN_{t+1}
1	0	0	1
1	1	1	0
0	X	Q_t	QN_t

Código VHDL

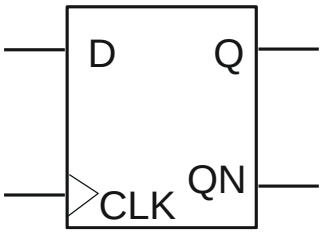
```
process (D, E)
 begin
      if E = '1' then
        Q  <= D;
        QN <= not D;
      end if;
 end process;
```



¿Variaciones si $\overline{E}$ (señal E fuera activa a nivel bajo)?

Biestables
Biestable D activo por flanco

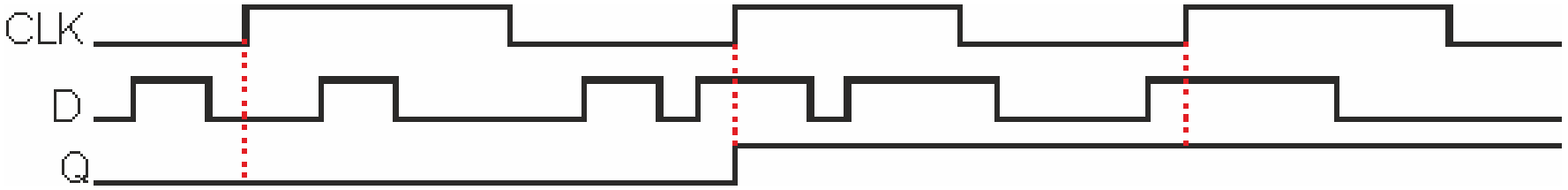
- La salida Q sigue a D en flancos activos de CLK



CLK	D	Q_{t+1}	QN_{t+1}
↓	0	0	1
↓	1	1	0
0	X	Q_t	QN_t
1	X	Q_t	QN_t

Código VHDL

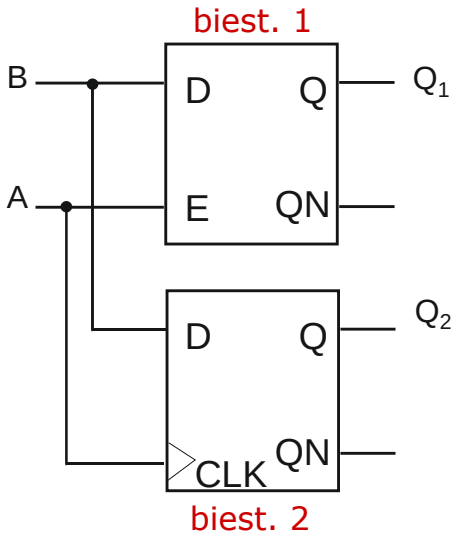
```
process (CLK)
  begin
   if CLK'event and CLK = '1' then
     Q  <= D;
     QN <= not D;
   end if;
  end process;
```

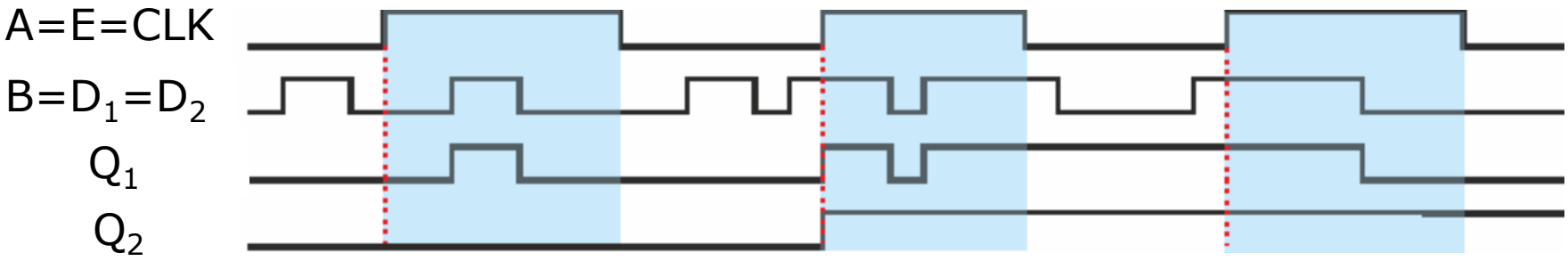


Dato.- Q(t=0)=0

¿Variaciones si CLK fuese activa en los flancos descendentes?

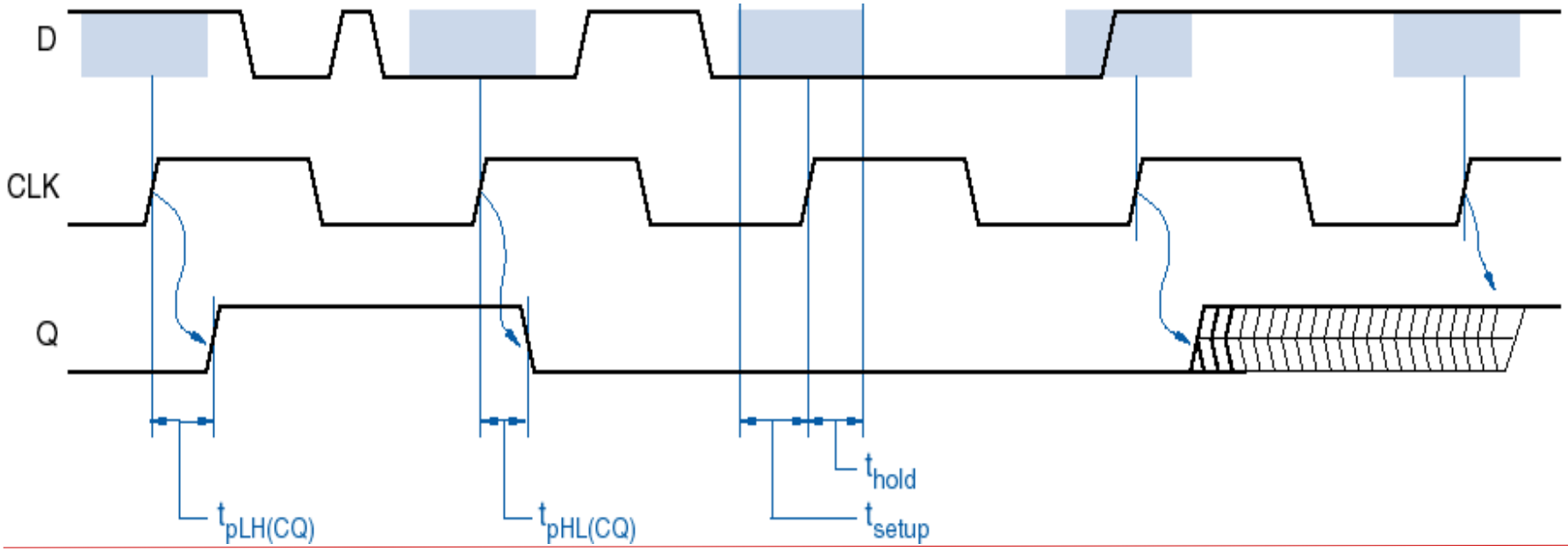
- Comparación de biestable activo por nivel y por flanco





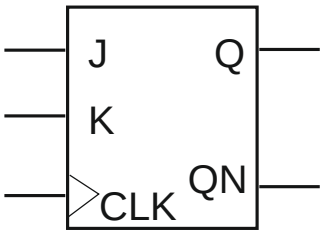
Dato.- $Q_1(t=0)=Q_2(t=0)=0$

- Parámetros temporales (ej: tipo D activo por ↓):
  - Retardos de propagación (en la salida, Q o QN, desde cambio en CLK)
  - Tiempo de "setup" (entr. D antes del flanco de CLK)
  - Tiempo de "hold" (entr. D después del flanco de CLK)



Biestable J-K activo por flanco

J-K activo por flanco:

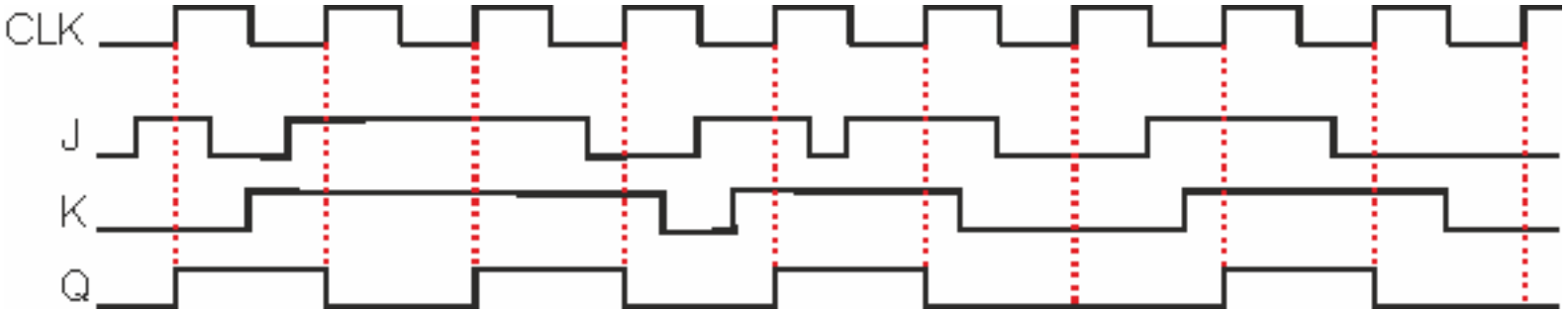


J	K	CLK	Q_{t+1}	QN_{t+1}
0	0	↓	Q_t	QN_t
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	QN_t	Q_t
X	X	0	Q_t	QN_t
X	X	1	Q_t	QN_t

Con J y K ambas activadas: biestable conmuta al estado contrario (Q) del que tenía

Código VHDL

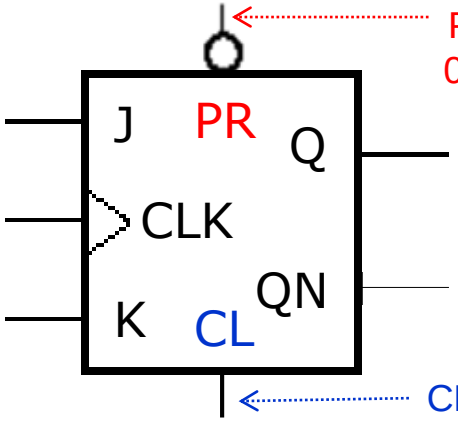
```
process (CLK)
  begin
   if CLK'event and CLK = '1' then
    if J = '1' and K = '0' then
         Q   <= '1';
         QN  <= '0';
    elsif J = '0' and K = '1' then
         Q   <= '0';
         QN  <= '1';
    elsif J = '1' and K = '1' then
          Q  <= QN;
          QN <= Q;
    end if;
   end if;
  end process;
```



Dato.- Q(t=0)=0

Biestables
Entradas asíncronas

- Entradas asíncronas (Preset y/o Clear):
  - **Son prioritarias**: Ignoran señales de sincronismo (reloj)
  - **Entradas síncronas sólo funcionarán cuando las asíncronas estén inactivas**
  - Entre las asíncronas también habrá prioridad



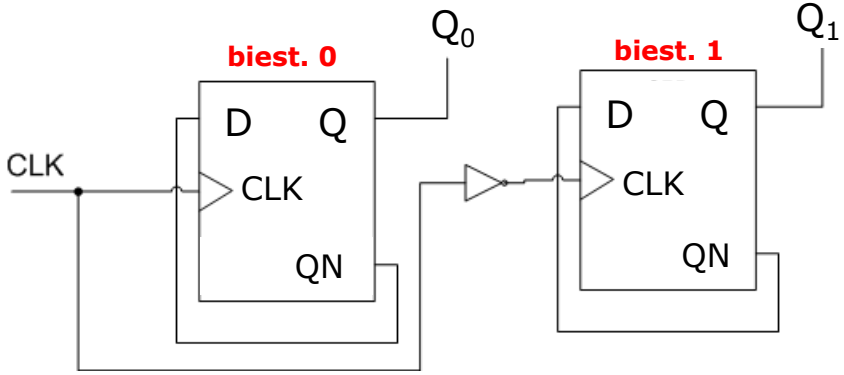
PRESET: En cuanto se activa (a 0 y sin importar CLK) pone Q a 1

CLEAR: En cuanto se activa (a 1 y sin importar CLK) pone Q a 0

Código VHDL

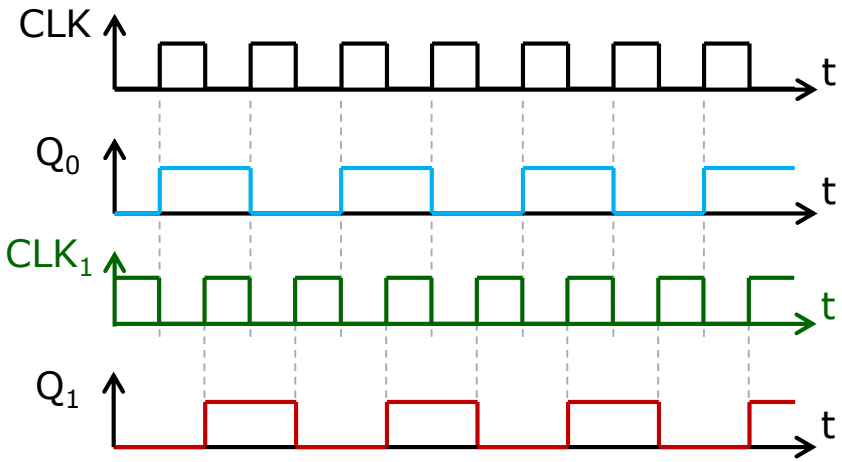
```
process (CLK,CLEAR,PRESET)
 begin
  if CLEAR = '1' then
    Q <= '0';
    QN  <= '1';
  elsif PRESET = '0' then
    Q   <= '1';
    QN  <= '0';
  elsif CLK'event and CLK = '1' then
   if J = '1' and K = '0' then
    Q   <= '1';
    QN  <= '0';
   elsif J = '0' and K = '1' then
    Q   <= '0';
    QN  <= '1';
   elsif J = '1' and K = '1' then
     Q  <= QN;
     QN <= Q;
   end if;
  end if;
 end process;
```

Biestables
Ejemplos de circuitos con biestables



$$D_0 = QN_0 \;\; ; \;\; D_1 = QN_1 \;\; ; \;\; CLK_1 = \overline{CLK}$$

Útil pintar CLK_1 (D_0 y D_1 no, porque conocemos cómo funcionan estos biestables cuando $D=QN$)



Datos.- Salidas de todos los biestables a nivel bajo (L) en t=0

Registros y contadores
Introducción. Definiciones

- En sistemas secuenciales:
  - Elementos básicos de memoria (biestables)
  - Capacidad de almacenamiento limitada
    - cada biestable almacena un bit
  - Capacidad de control

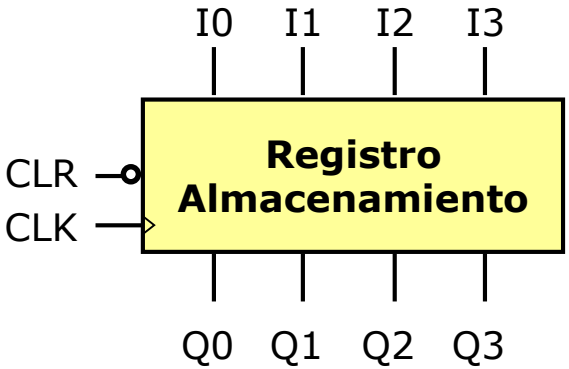
- Buscaremos aplicaciones que extiendan la potencia y la capacidad de esos elementos básicos:
  - incrementando la capacidad de almacenamiento: **registros**
  - realizando tareas relacionadas con el *conteo* de eventos: **contadores**

Registros
Introducción

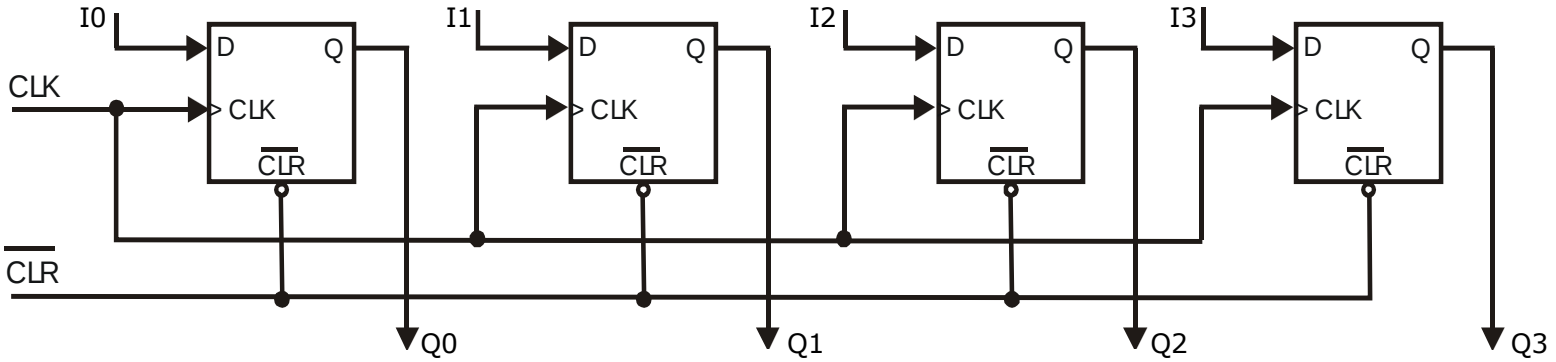
- Definición:
  - Circuito lógico capaz de almacenar (*registrar*) una cantidad limitada de información durante un determinado tiempo
  - Están compuestos por biestables tipo D agrupados
  - También tendrán mecanismos de *control*
    - pueden tener señales de: Clear, Shift/Load, Enable…
- Clasificación:
  - Registros de almacenamiento:
    - Almacenan información
  - Registros de desplazamiento:
    - Almacenan y permiten el *desplazamiento* de su contenido

Registros de almacenamiento:

- Basados en flip-flops tipo D
  - Con entradas y salidas de datos, señal de reloj y señal/es de control

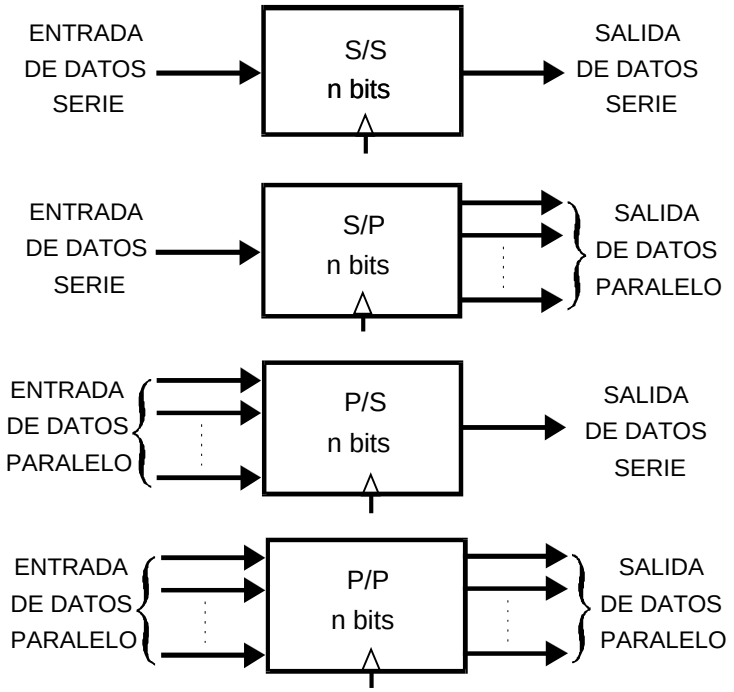


Modos de operación	Entradas			Salidas
	CLR	CLK	Ii	Qi_{t+1}
Reset (clear)	0	X	X	0
Load "1"	1	↑	1	1
Load "0"	1	↑	0	0

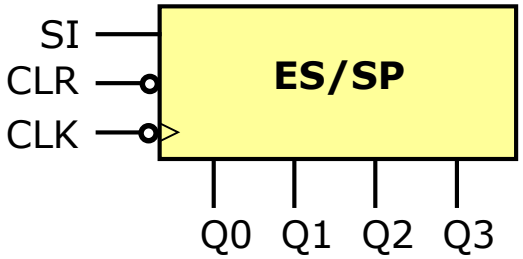


- Registros de desplazamiento:
  - Basados en flip-flops tipo D
    - La información se irá pasando de biestable en biestable

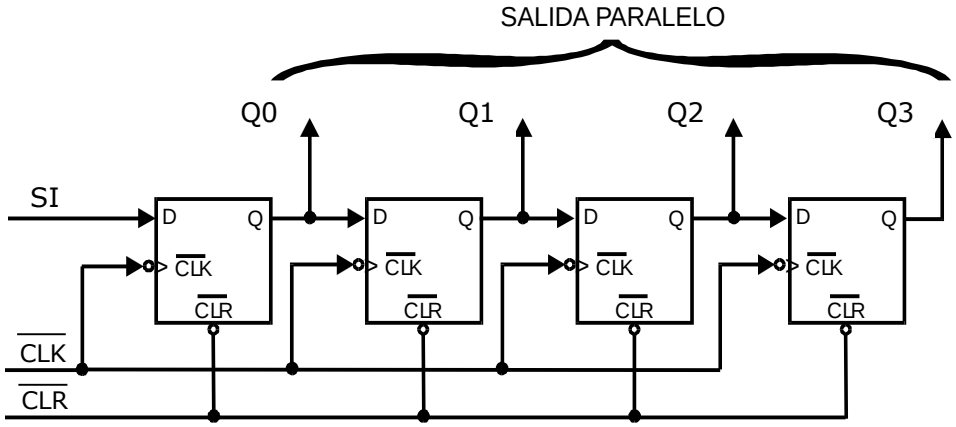
- Clasificados en función de su comportamiento E/S:
  - Entrada serie–salida serie
  - Entrada serie–salida paralelo
  - Entrada paralelo–salida serie
  - Entrada paralelo–salida paralelo

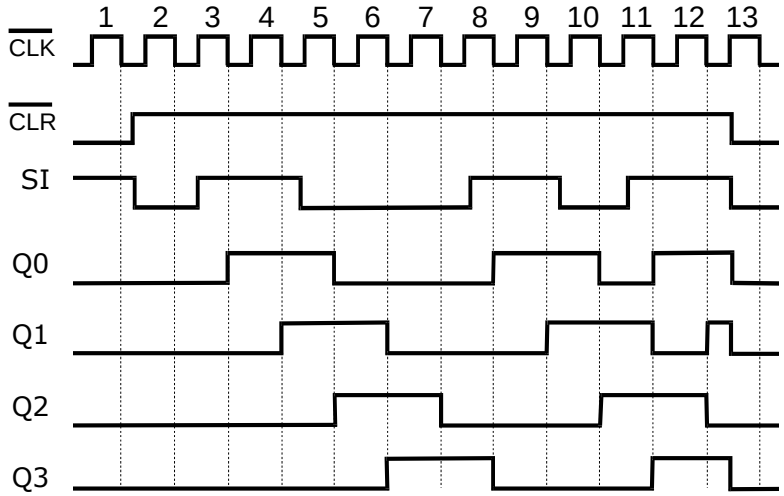


Entrada serie – salida paralelo:



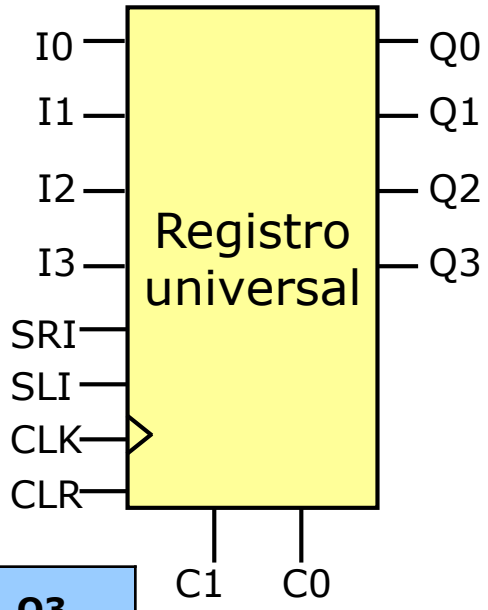
Entradas			Salidas			
$\overline{CLR}$	$\overline{CLK}$	SI	$Q0_{t+1}$	$Q1_{t+1}$	$Q2_{t+1}$	$Q3_{t+1}$
0	X	X	0	0	0	0
1	0 o 1	X	$Q0_t$	$Q1_t$	$Q2_t$	$Q3_t$
1	↓	si	si	$Q0_t$	$Q1_t$	$Q2_t$





Registro de desplazamiento universal:

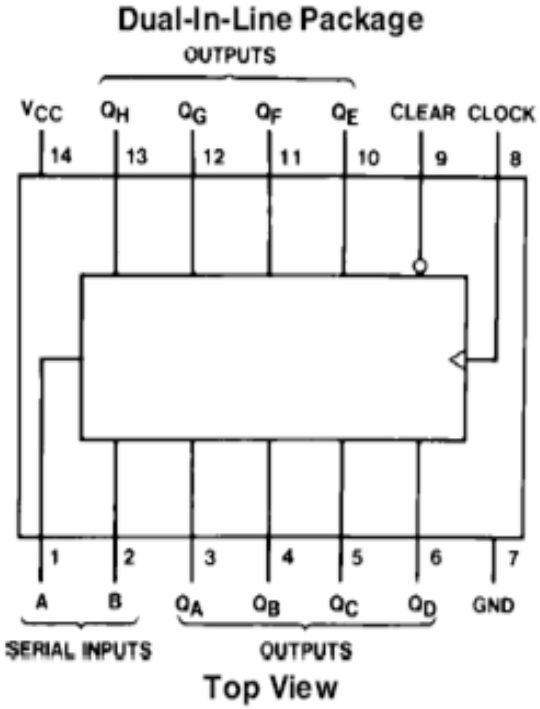
- Dispone de terminales de control (C1 y C0, síncronos) para elegir:
  - Mantener datos
  - Desplazar a derechas
  - Desplazar izquierdas
  - Cargar datos en paralelo



CLR	C1	C0	CLK	SRI	SLI	I0	I1	I2	I3	$Q0_{t+1}$	$Q1_{t+1}$	$Q2_{t+1}$	$Q3_{t+1}$
1	X	X	X	X	X	X	X	X	X	0	0	0	0
0	0	0	↑	X	X	X	X	X	X	$Q0_t$	$Q1_t$	$Q2_t$	$Q3_t$
0	0	1	↑	sri	X	X	X	X	X	sri	$Q0_t$	$Q1_t$	$Q2_t$
0	1	0	↑	X	sli	X	X	X	X	$Q1_t$	$Q2_t$	$Q3_t$	sli
0	1	1	↑	X	X	i0	i1	i2	i3	i0	i1	i2	i3

74HC164:

Connection and Logic Diagrams



Truth Table

Inputs				Outputs			
Clear	Clock	A	B	Q_A	Q_B	...	Q_H
L	X	X	X	L	L		L
H	L	X	X	Q_{AO}	Q_{BO}		Q_{HO}
H	↑	H	H	H	Q_{An}		Q_{Gn}
H	↑	L	X	L	Q_{An}		Q_{Gn}
H	↑	X	L	L	Q_{An}		Q_{Gn}

H = High Level (steady state), L = Low Level (steady state)

X = Irrelevant (any input, including transitions)

↑ = Transition from low to high level.

Q_{AO}, Q_{BO}, Q_{HO} = the level of Q_A, Q_B, or Q_H, respectively, before the indicated steady state input conditions were established.

Q_{An}, Q_{Gn} = The level of Q_A or Q_G before the most recent ↑ transition of the clock; indicated a one-bit shift.

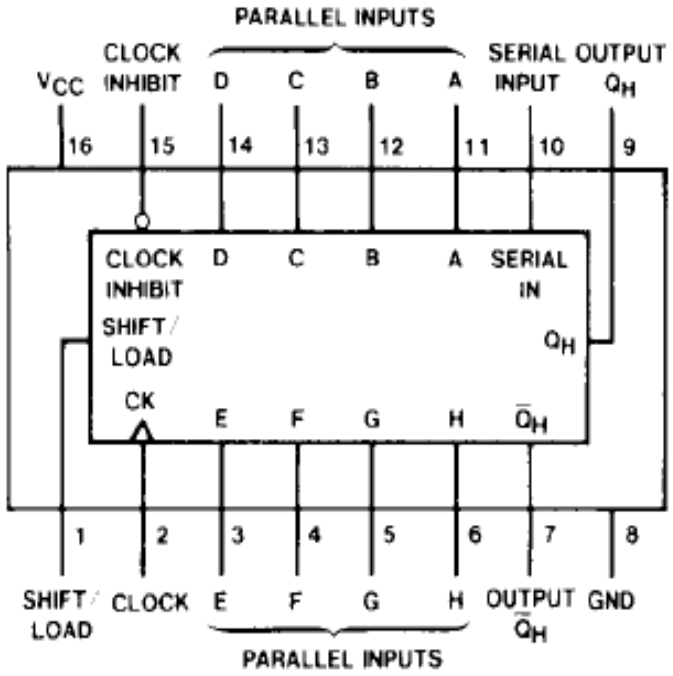
Nota.- Si se llamase SI = A AND B, en modo desplazamiento, al llegar el flanco activo, $Q_A(t+1)=SI$

Registros
Ejemplos comerciales

- 74HC165:

Connection Diagram

Dual-In-Line Package



Top View

Function Table

Inputs					Internal Outputs		Output Q_H
Shift/ Load	Clock Inhibit	Clock	Serial	Parallel A . . . H	Q_A	Q_B	
L	X	X	X	a . . . h	a	b	h
H	L	L	X	X	Q_{A0}	Q_{B0}	Q_{H0}
H	L	↑	H	X	H	Q_{AN}	Q_{GN}
H	L	↑	L	X	L	Q_{AN}	Q_{GN}
H	H	X	X	X	Q_{A0}	Q_{B0}	Q_{H0}

H = High Level (steady state), L = Low Level (steady state)

X = Irrelevant (any input, including transitions)

↑ = Transition from low to high level

Q_{A0}, Q_{B0}, Q_{H0} = The level of Q_A, Q_B, or Q_H, respectively, before the indicated steady-state input conditions were established.

Q_{AN}, Q_{GN} = The level of Q_A or Q_G before the most recent ↑ transition of the clock; indicates a one-bit shift.

Nota.- Un nombre más acorde para el pin Clock Inhibit, de acuerdo a la función que realiza, sería Clock Enable (algunos fabricantes lo llaman así)

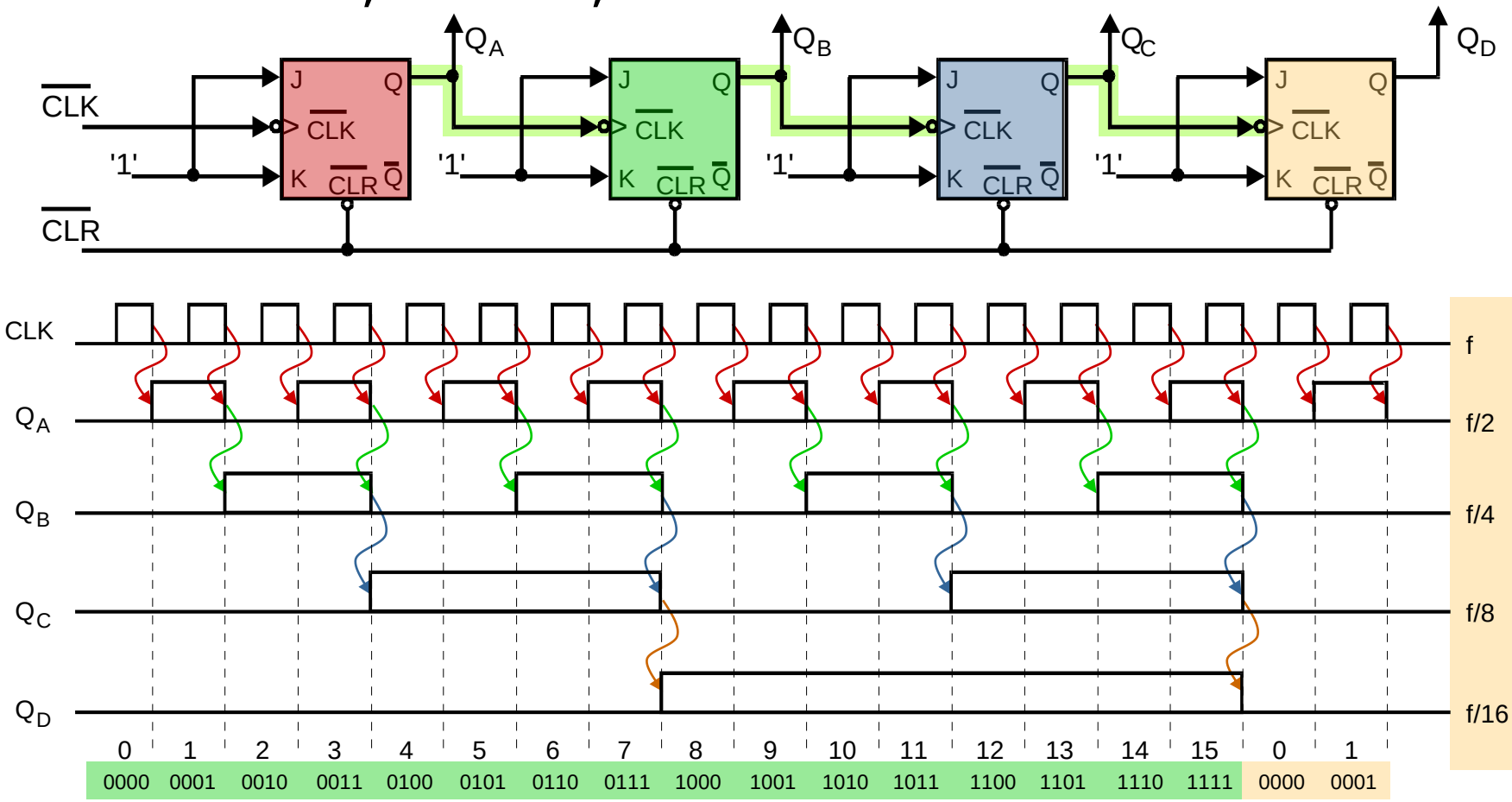
Contadores
Introducción

- Definición:
  - Circuito lógico capaz de generar (contar, *en código binario*) el número de flancos activos de reloj recibidos
  - Siempre generan *secuencia cíclica*
- Clasificación según relación con reloj:
  - Síncronos:
    - El *mismo* reloj llega a todos los biestables (cambian todos a la vez)
  - Asíncronos:
    - Puede haber relojes distintos atacando a los biestables (cambian en distintos instantes)

Contadores
Introducción

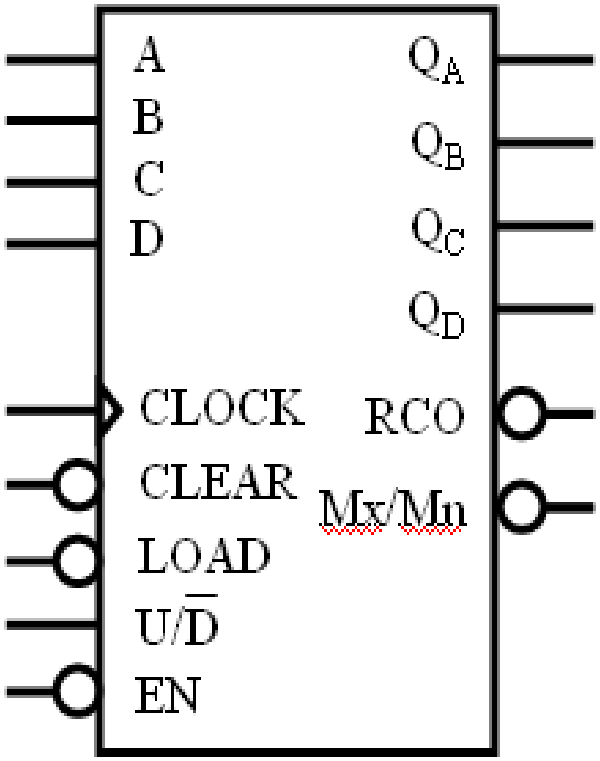
- Clasificación según sentido de cuenta:
  - Ascendente
  - Descendente
- Módulo de un contador:
  - Longitud del ciclo que se repite (Nº de estados)
- Información sobre su funcionamiento:
  - En cronogramas y tablas de las *datasheets*
- Suelen estar constituidos por biestables J-K (modo T) síncronos por flanco:
  - nº biestables = nº bits del código de la secuencia
    - cada biestable aporta un bit del código (menor peso…)

- Ascendente, binario, 4 bits:



Contadores
Contadores universales

- Modo:
  - Cuentan Binario/BCD
- Terminales:
  - Típicos de control (Cl,Load…)
    - A veces tienen:
      - clock inhibit, 2 enables …



SALIDA

- **RCO:** Indica si se ha llegado al fin de cuenta (usual. durante $0{,}5T_{CLK}$)
- **Mx/Mn** (MáxMin): Como RCO, (lo indica normal. durante un T_{CLK})

- Algunos, contadores bidireccionales, tienen entradas de reloj independ. para contar Up y Down (en lugar de señal U/$\overline{D}$):

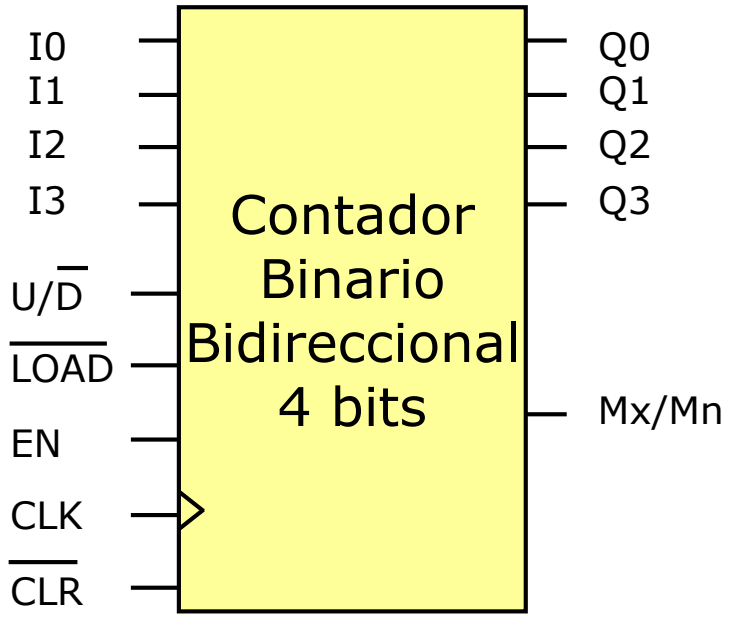
  CLK_U y CLK_D (para cuenta ascendente se introducen los flancos a contar por pin CLK_u y para descendente por CLK_D)

Ejemplo contador bidireccional

Mx/Mn se activa a nivel '1' (durante un T_{CLK}):

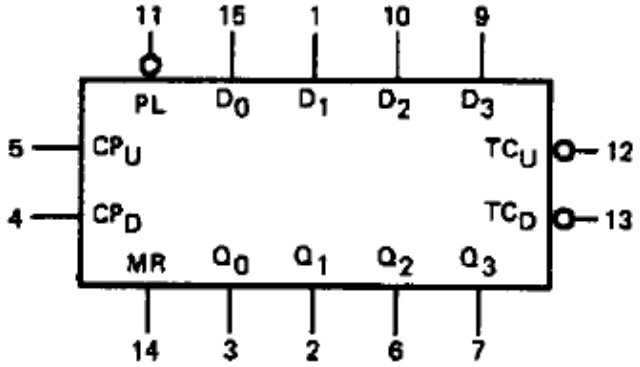
cuando en cuenta ascendente Q[3:0]=1111

cuando en cuenta descendente Q[3:0]=0000



$\overline{CLR}$	EN	CLK	$\overline{LOAD}$	U/$\overline{D}$	I0	I1	I2	I3	$Q0_{t+1}$	$Q1_{t+1}$	$Q2_{t+1}$	$Q3_{t+1}$
0	X	X	X	X	X	X	X	X	0	0	0	0
1	0	X	X	X	X	X	X	X	$Q0_t$	$Q1_t$	$Q2_t$	$Q3_t$
1	1	↑	0	X	i0	i1	i2	i3	i0	i1	i2	i3
1	1	↑	1	1	X	X	X	X	cuenta ascendente			
1	1	↑	1	0	X	X	X	X	cuenta descendente			

74x193



Notes

1. H = HIGH voltage level
   L = LOW voltage level
   X = don't care
   ↑ = LOW-to-HIGH clock transition

Para indicar fin de cuenta (como RCO)

2. $\overline{TC}_U = CP_U$ at terminal count up (HHHH)
3. $\overline{TC}_D = CP_D$ at terminal count down (LLLL)

Function table

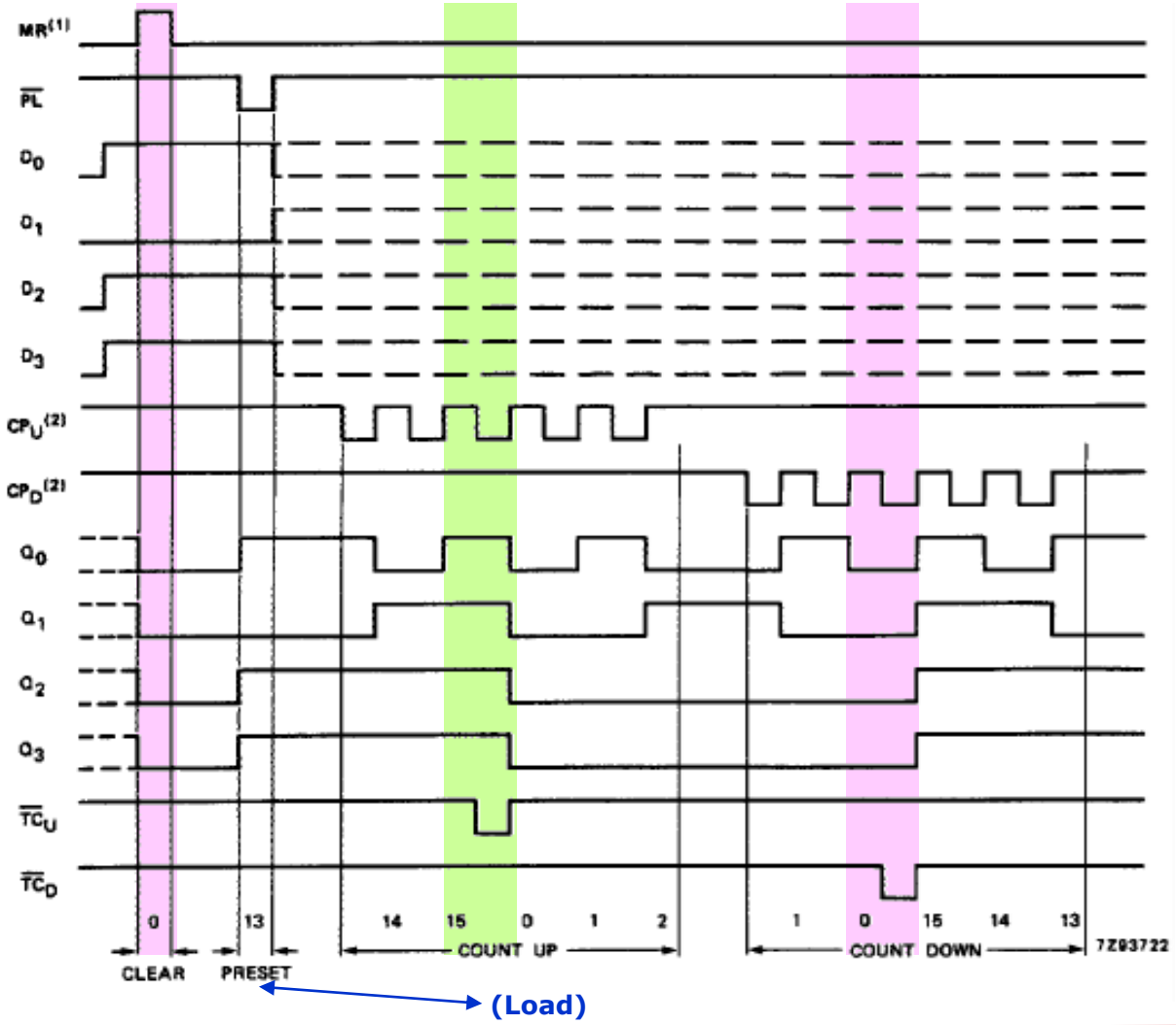
OPERATING MODE	INPUTS								OUTPUTS					
	MR	$\overline{PL}$	CP_U	CP_D	D_0	D_1	D_2	D_3	Q_0	Q_1	Q_2	Q_3	$\overline{TC}_U$	$\overline{TC}_D$
reset (clear)	H H	X X	X X	L H	X X	X X	X X	X X	L L	L L	L L	L L	H H	L H
parallel load	L L L L	L L L L	X X L H	L H X X	L L H H	L L H H	L L H H	L L H H	L L H H	L L H H	L L H H	L L H H	H H L H	L H H H
count up	L	H	↑	H	X	X	X	X	count up				$H^{(2)}$	H
count down	L	H	H	↑	X	X	X	X	count down				H	$H^{(3)}$

Contadores

Contadores comerciales

- 74x193

Typical clear, load and count sequence



(1) Clear overrides load, data and count inputs.

(2) When counting up the count down clock input (CP_D) must be HIGH, when counting down the count up clock input (CP_U) must be HIGH.

Sequence

Clear (reset outputs to zero);

load (preset) to binary thirteen;

count up to fourteen, fifteen, terminal count up, zero, one and two;

count down to one, zero, terminal count down, fifteen,

Autómata de Mealy

Función de salida (**g**) z^t = **g** $[x^t, q^t]$ Función de transición (**f**) q^{t+1} = **f** $[x^t, q^t]$

Autómata de Moore

Función de salida (**g**) z^t = **g** $[q^t]$ Función de transición (**f**) q^{t+1} = **f** $[x^t, q^t]$

Autómatas de estados finitos

Fundamentos: definiciones

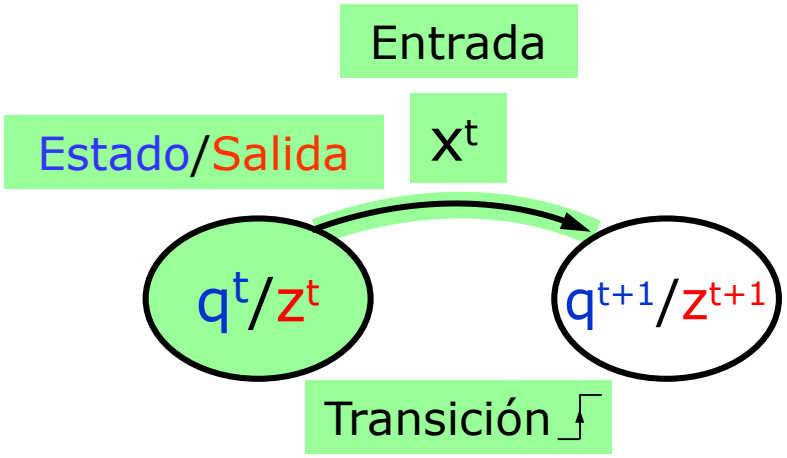
- El **ESTADO** es toda la información que se necesita saber (almacenar) para dadas las entradas en unos instantes determinados, deducir las correspondientes salidas (y próximos estados)
  - El estado define la situación del sistema en todo momento.
  - El sistema está en cualquier instante en un estado perfectamente definido de entre un número posible, finito, de estados (FSM: Finite State Machine).
- El sistema evoluciona realizando una **TRANSICIÓN** entre un estado (*estado actual*) y otro estado (*próximo estado o estado siguiente*). Sistema síncrono: transiciones en flancos de *clk*
- Las transiciones se realizan, desde cada estado, dependiendo de los valores de las **ENTRADAS** que recibe el sistema
- El sistema entrega **SALIDAS**, que pueden generarse:
  - A partir del estado actual (Moore),
  - A partir del estado actual y de las entradas (Mealy)
- Las máquinas de estados finitos se describen por medio de:
  - Grafos y tablas de estados

Autómatas de estados finitos

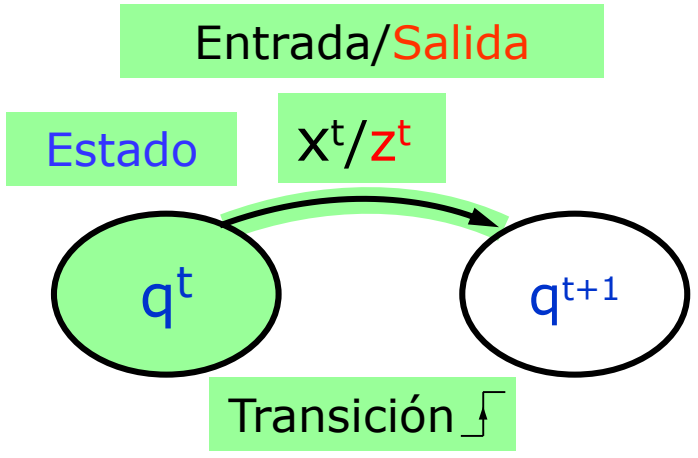
Fundamentos: representación de grafos

Grafos

Autómata de **Moore**



Autómata de **Mealy**



En **Moore**: $\mathbf{z^t}$ (salida) sólo depende de $\mathbf{q^t}$ (estado actual)

- $\mathbf{z^t}$ cambiará solo en los flancos activos (⎍) de CLK (como $\mathbf{q^t}$)

En **Mealy**: $\mathbf{z^t}$ depende de $\mathbf{x^t}$ (entrada) y de $\mathbf{q^t}$ (estado actual)

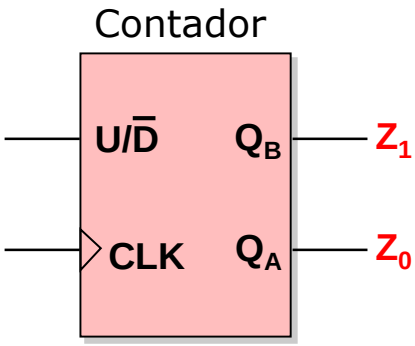
- $\mathbf{z^t}$ puede cambiar en cualquier instante que lo haga $\mathbf{x^t}$ o en un ⎍ (como $\mathbf{q^t}$)

En **Moore** y **Mealy**: $\mathbf{q^{t+1}}$ (estado siguiente) depende de $\mathbf{x^t}$ y de $\mathbf{q^t}$

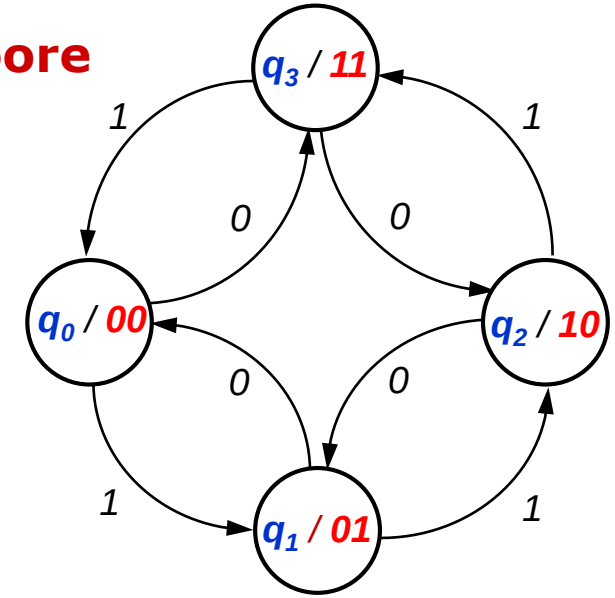
Autómatas de estados finitos

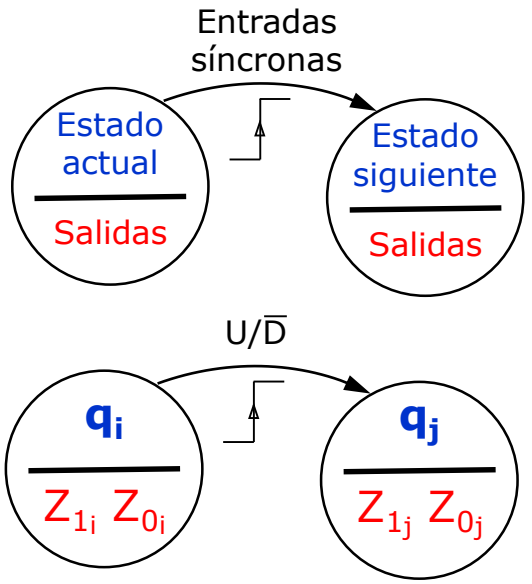
Fundamentos: ejemplos

Ejemplo 1: Contador de 2 bits y cuenta ascendente/descendente en función de señal U/$\overline{D}$



Moore





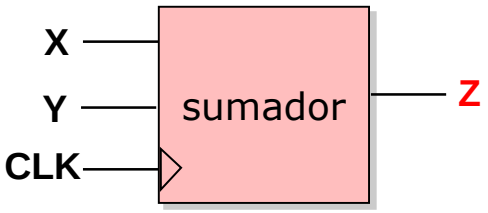
- **Estados**: (q_0, q_1, q_2, q_3) valores de cuenta
- **Salidas**: Z_1 Z_0

¡¡**IMPORTANTE**!!- Contadores, registros y circuitos cuyas salidas cambien típicamente en los flancos activos de clk se harán con máquinas de Moore, que tendrán **como mínimo** tantos estados como salidas posibles.

Autómatas de estados finitos

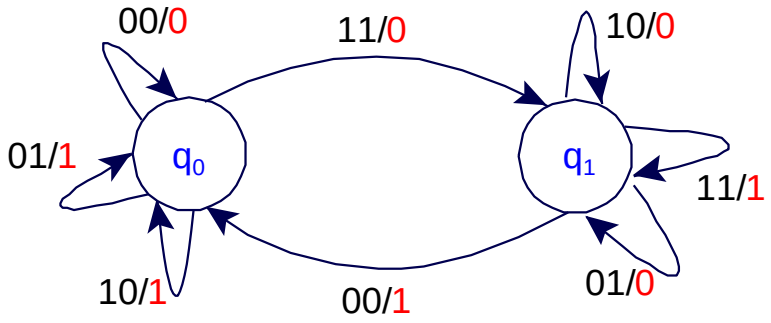
Fundamentos: ejemplos

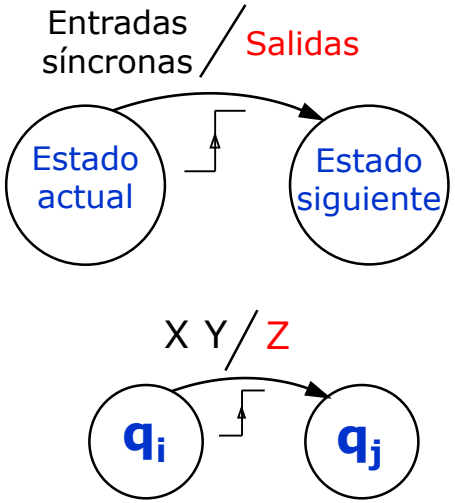
Ejemplo 2: Sumador (con grafo Mealy) de dos datos que llegan en serie, por parejas, sincronizados con los flancos activos de CLK, comenzando por los bits de menor peso. El resultado se va presentando a la salida en serie a medida que se va generando, teniendo en cuenta los posibles acarreos producidos



X: … 0 0 1 0 1 1 1 1 0 1 0 …
Y: … 0 1 1 0 0 1 1 0 0 1 1 …
Z: … 0 1 0 1 1 0 1 0 1 0 0 …

Mealy





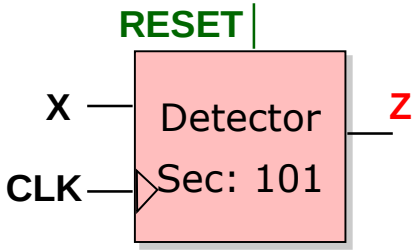
- **Estados**:

q_0: estado con acarreo=0
q_1: estado con acarreo=1

Autómatas de estados finitos

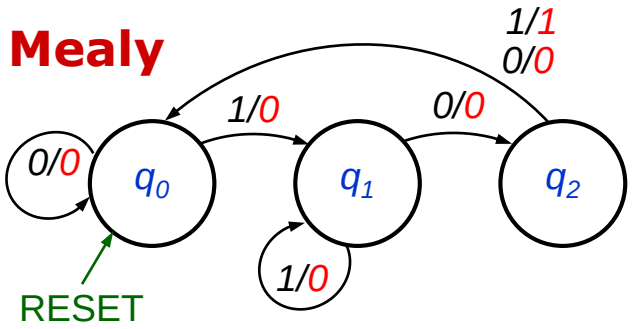
Fundamentos: ejemplos

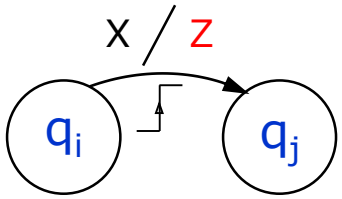
Ejemplo 3: Detector cíclico de la secuencia 101, con reset asíncrono.
Un mismo bit no puede pertenecer a más de una secuencia válida



X: … 0 1 0 0 **1 0 1** 0 1 0 1…
Z: … 0 0 0 0 0 0 **1** 0 0 0 **1**…

Mealy

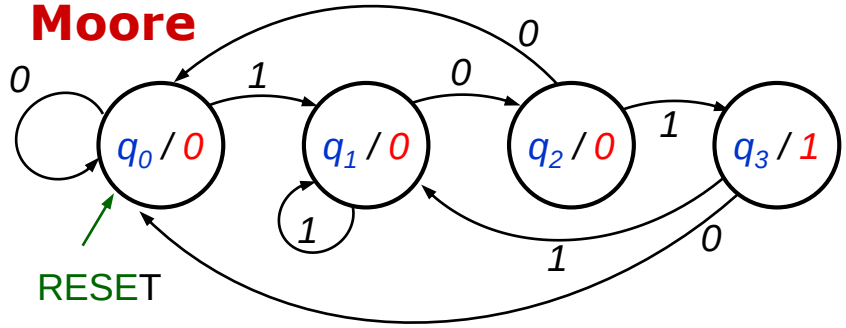


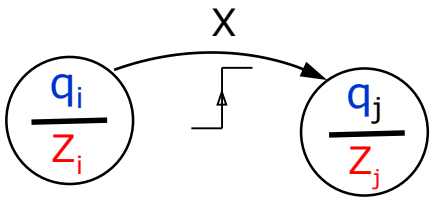


- **Estados**:

q_0: ningún bit válido
(ó 3er bit válido recibido)
q_1: 1er bit válido recibido
q_2: 2º bit válido recibido

Moore





- **Estados**:

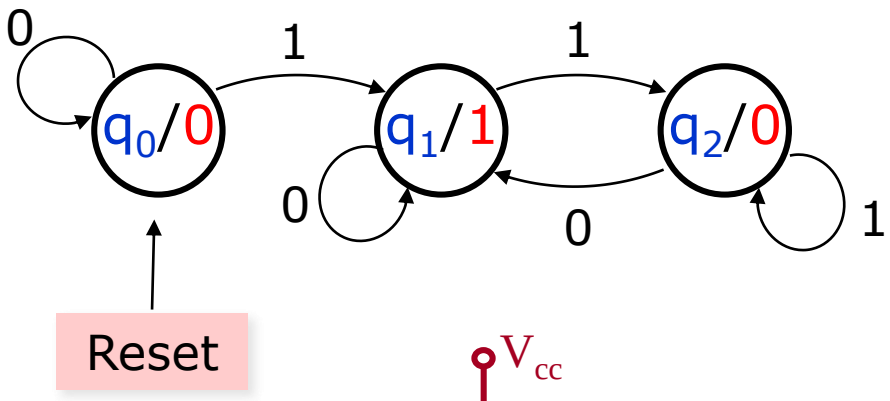
q_0: ningún bit válido recib.
q_1: 1er bit válido recibido
q_2: 2º bit válido recibido
q_3: 3er bit válido recibido

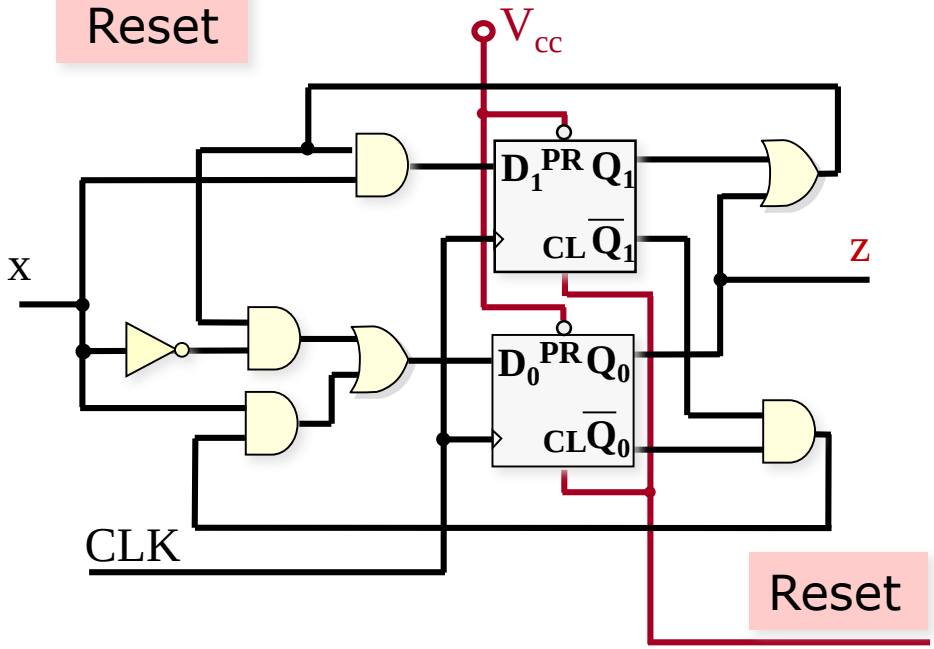
Comparación entre los autómatas de Moore y de Mealy

- Todos los circuitos secuenciales síncronos pueden implementarse tanto como autómatas de Moore como de Mealy
- Menor complejidad del circuito cuando se resuelve como autómata de Mealy
- En los autómatas de Mealy las modificaciones en las entradas provocan cambios en la salida en el momento en el que se producen
- En los autómatas de Moore las salidas solamente cambian cuando se produce un flanco de reloj y cambia el estado
- Solamente utilizaremos autómatas de Mealy:
  - cuando los cambios en las entradas del circuito estén sincronizados con la señal de reloj, o
  - cuando los cambios en otros momentos no afecten negativamente al funcionamiento del sistema global

Comparación entre autómatas de Moore y Mealy: complejidad relativa

Autómata de Moore





Autómata de Mealy