

## Unidad 4.2 – CIRCUITOS LÓGICOS SECUENCIALES

# APLICACIONES CON FLIP-FLOPS

- **Aplicaciones con flip-flops**
  - **Concepto y clasificación**
- **Registros de desplazamiento**
  - **Serie-serie**
  - **Serie-paralelo**
  - **Paralelo-serie**
  - **Paralelo-paralelo**
  - **Universal**
- **Contadores binarios**
  - **Asincrónicos**
  - **Sincrónicos**
- **Diseño de contadores**
  - **Memorias**
  - **Ejemplos**

- > **Floyd T. (2006). FUNDAMENTOS DE SISTEMAS DIGITALES. Capítulo 7: Latches, flip-flops y temporizadores; capítulo 8: Contadores; capítulo 9: Registros de desplazamiento.**
- > **Tocci R. (2007). SISTEMAS DIGITALES PRINCIPIOS Y APLICACIONES. Capítulo 5: Flip-flops y dispositivos relacionados; capítulo 7: Contadores y registros.**
- > **Mano M. (2003). DISEÑO DIGITAL. Capítulo 5: Lógica secuencial sincrónica; capítulo 6: Registros y contadores.**
- > **Brown S. (2006). FUNDAMENTOS DE LÓGICA DIGITAL CON DISEÑO VHDL. Capítulo 7: Flip-flops, registros, contadores y un procesador simple.**

## Concepto

Las unidades biestables (flip-flops) se utilizan como unidades complementarias para configurar circuitos secuenciales utilizando como base estructuras combinacionales.



## Aplicaciones

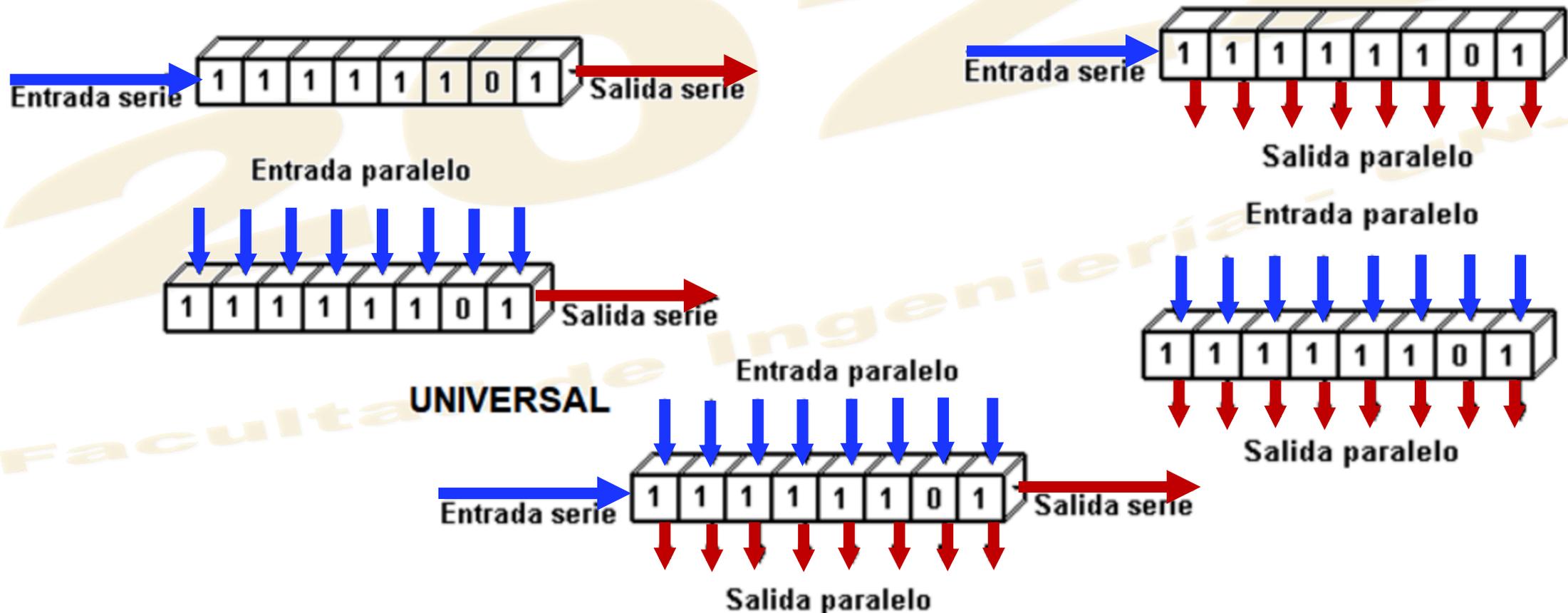


## Concepto

(Floyd, pg. 553)

Son unidades construidas en base a flip-flops para formar estructuras de memoria con diferentes configuraciones.

Unidades de flip-flops interconectados que pueden almacenar un conjunto de bits en en secuencias pre-establecidas.



## Estructura serie - serie

(Brown, pg. 399; Floyd, pg. 553)

Unidad de 4 bits con FF tipo D

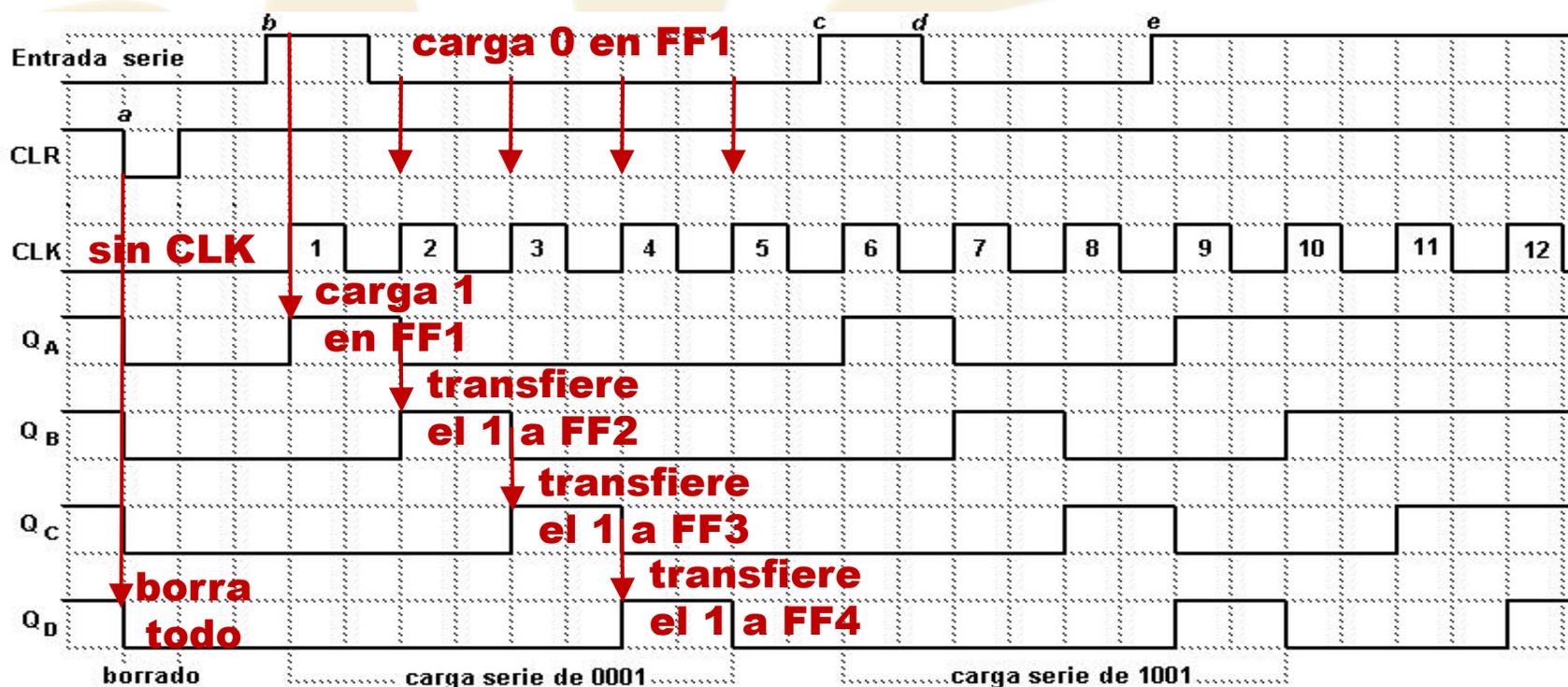
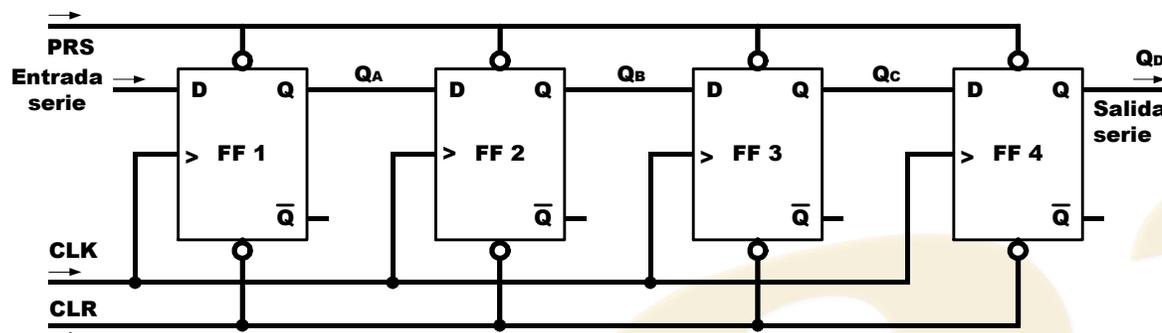
CLK por flanco (+)

No requiere borrado previo

Datos ingresan por FF1 y se leen por FF4 (Q<sub>D</sub>)

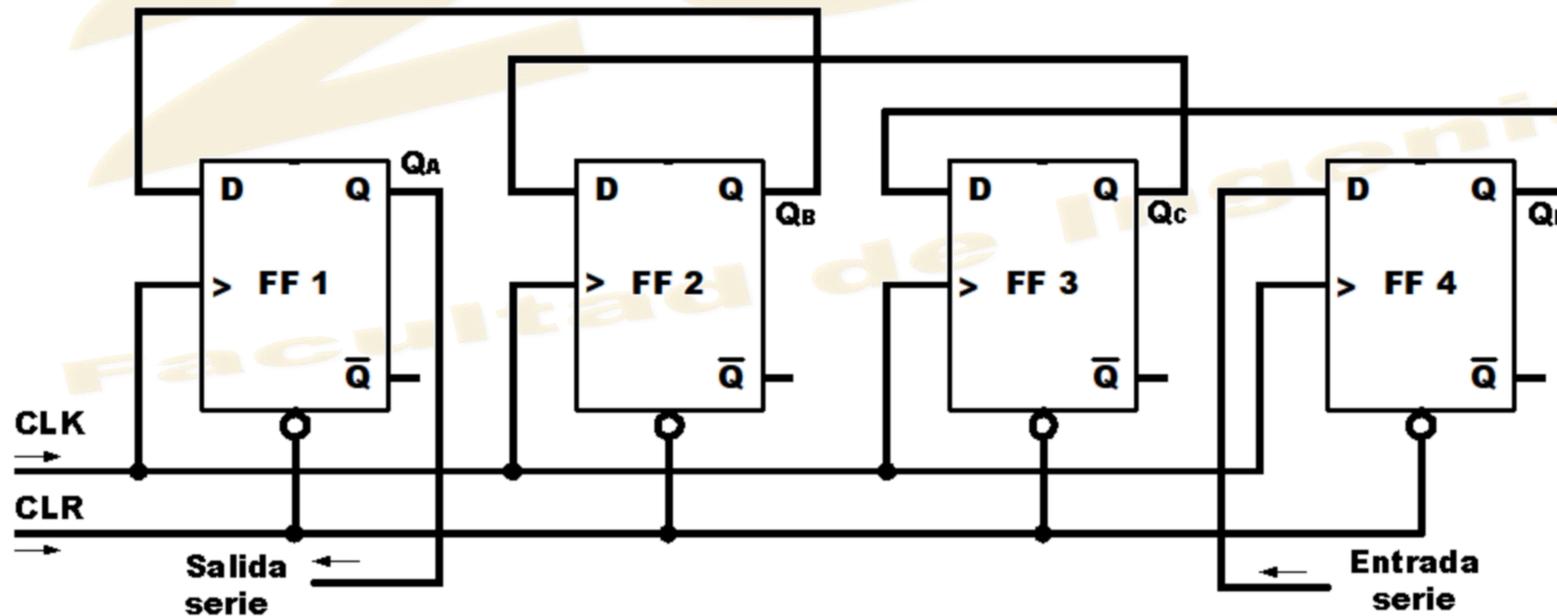
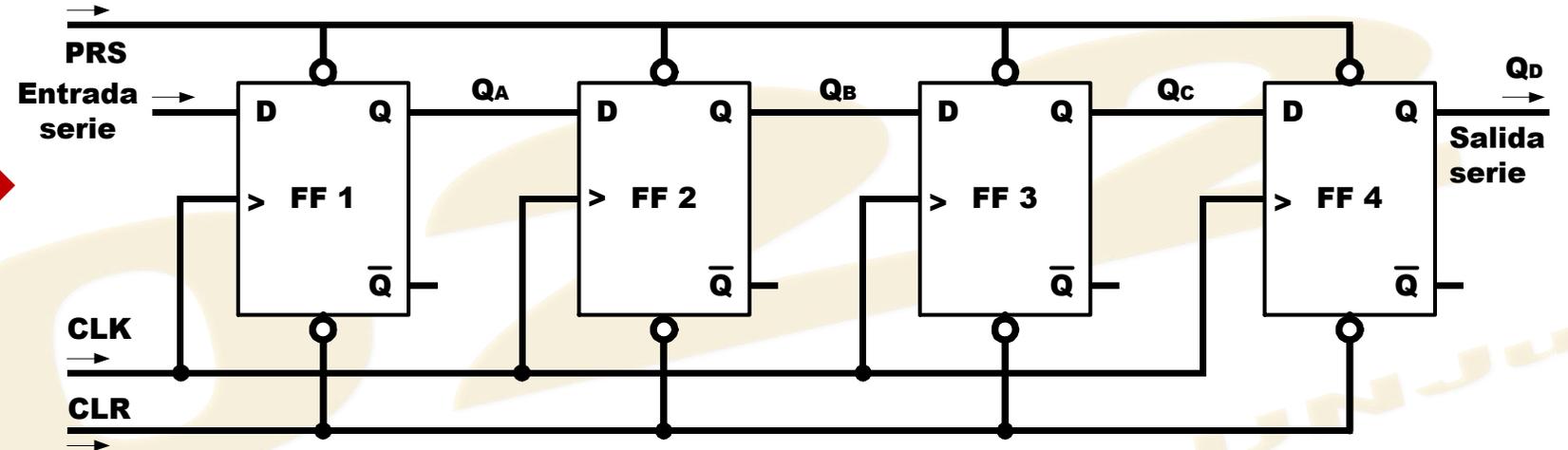
Desplazamiento a derecha

Es el más simple, pero también el más lento



## Estructura serie - serie

**Desplazamiento  
a derecha**



**Desplazamiento  
a izquierda**

## Estructura serie - serie

Simulación en Matlab

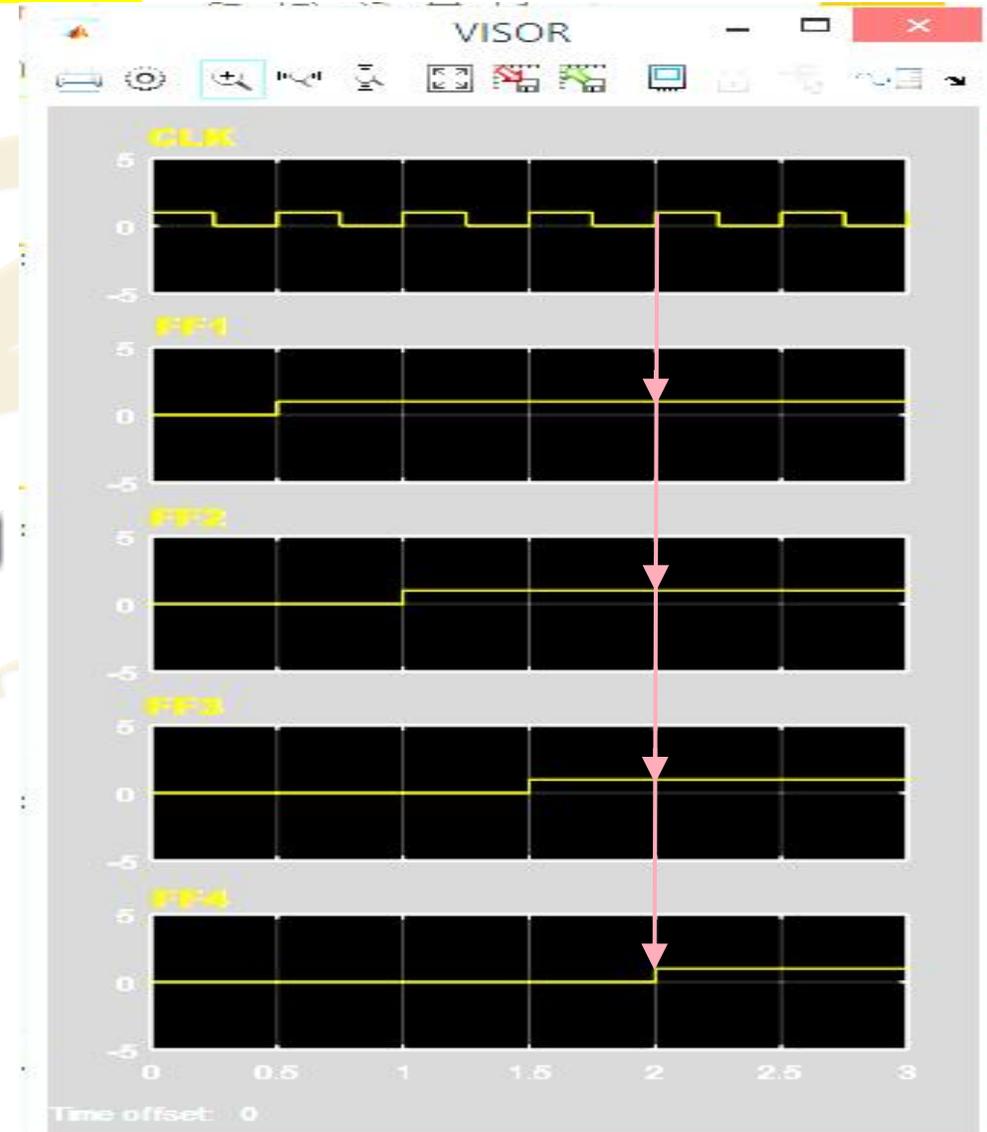
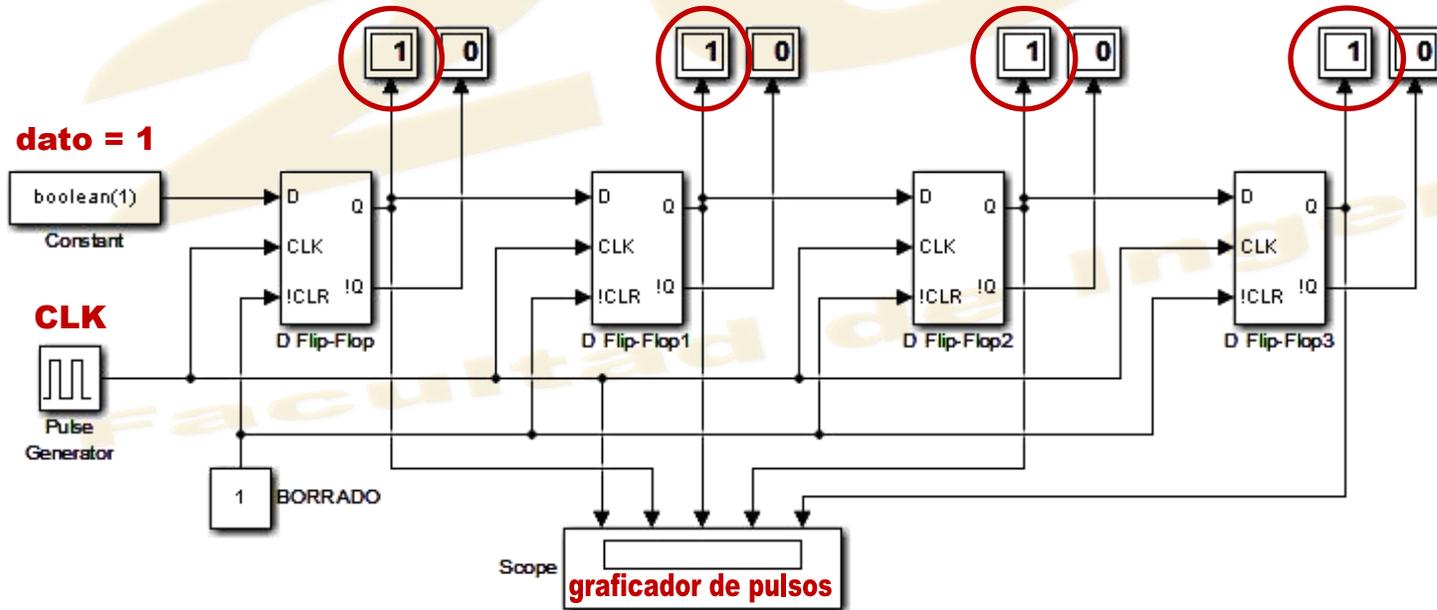
CLK por flanco (+).

Requiere borrado previo.

Los datos ingresan por FF1 y se leen por FF4 (Q<sub>D</sub>).

Es el más simple, pero también el más lento.

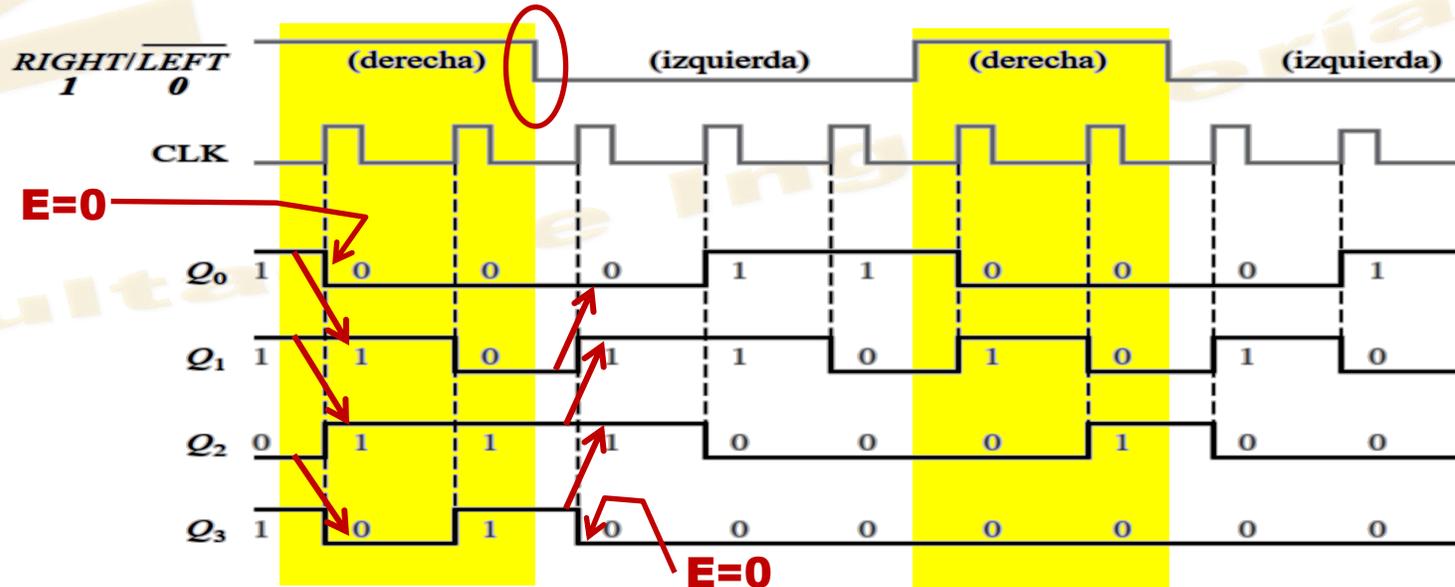
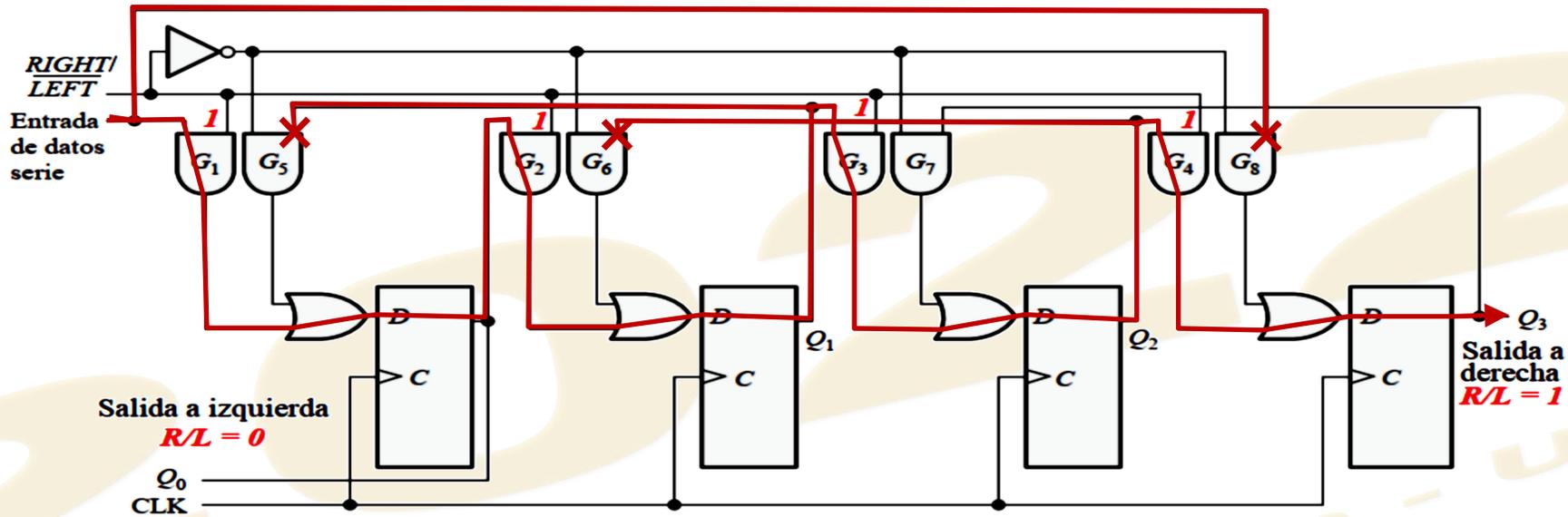
REGISTRO DE DESPLAZAMIENTO SERIE - SERIE DE 4 BITS



## Estructura serie - serie *bidireccional*

(Floyd, pg. 567)

**(a dezq) R/L=0**  
**(entrada) E=0**



## Estructura serie - paralelo

Modelo de 4 bits con  
FF tipo D.

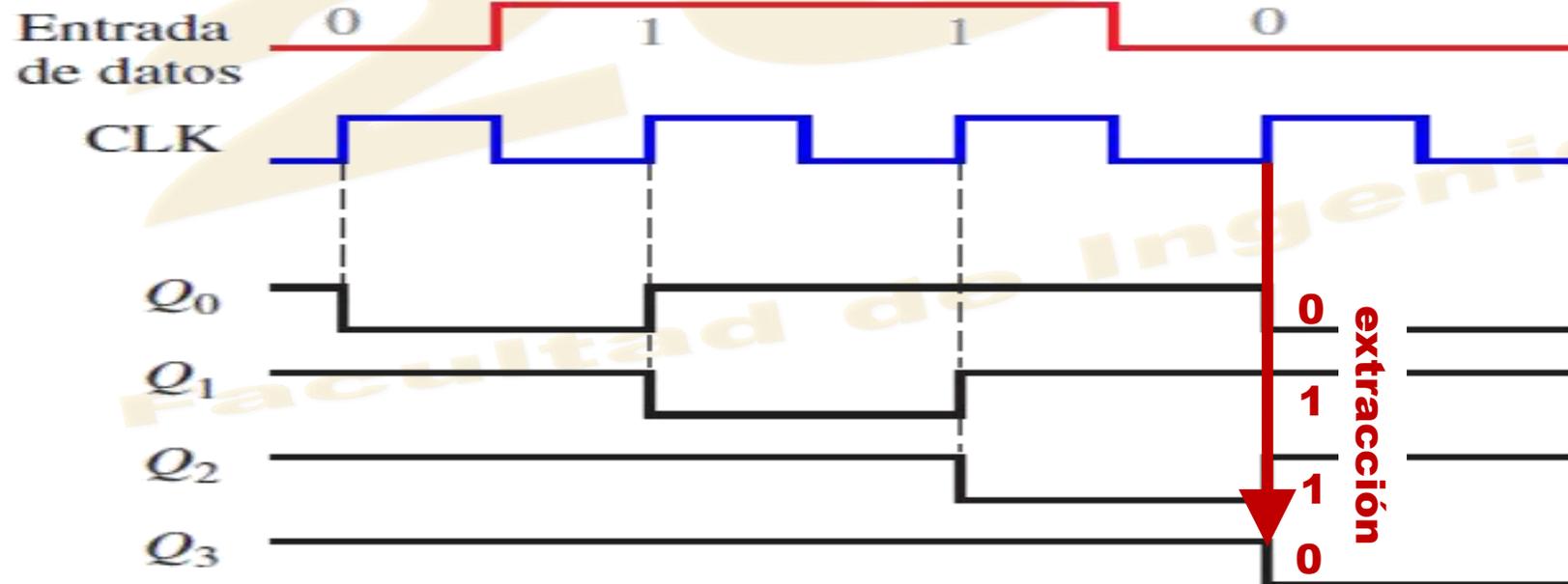
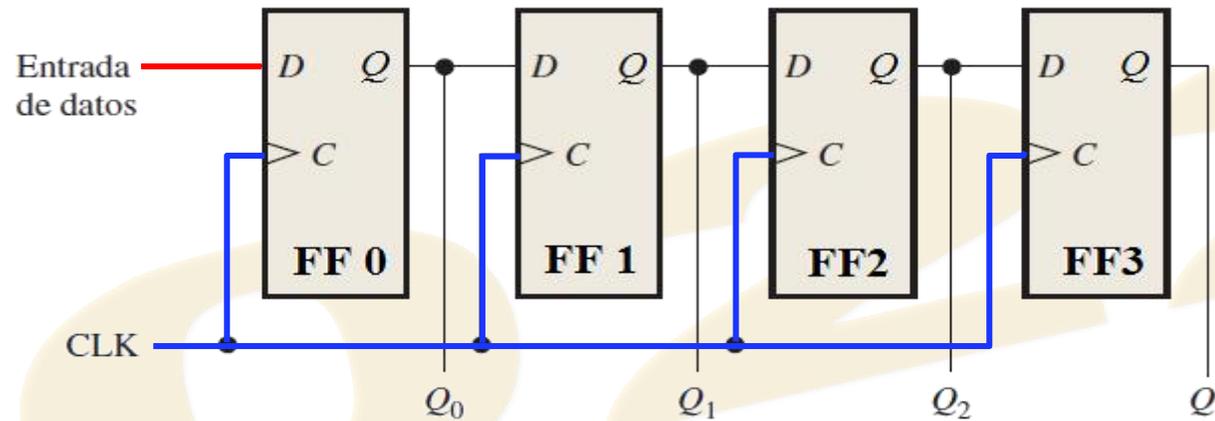
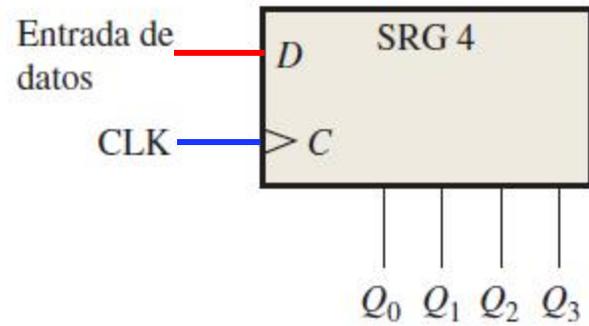
CLK por flanco (+).

Los datos ingresan por  
FF0 (D) y se leen por  
( $Q_0 Q_1 Q_2 Q_3$ ).

Debe esperarse el  
ingreso de todos los  
datos para no leer con  
error.

Es más rápido que el  
registro S-S

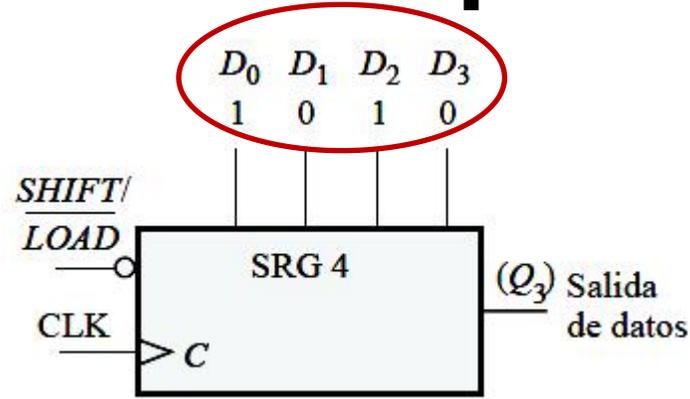
### Bloque funcional



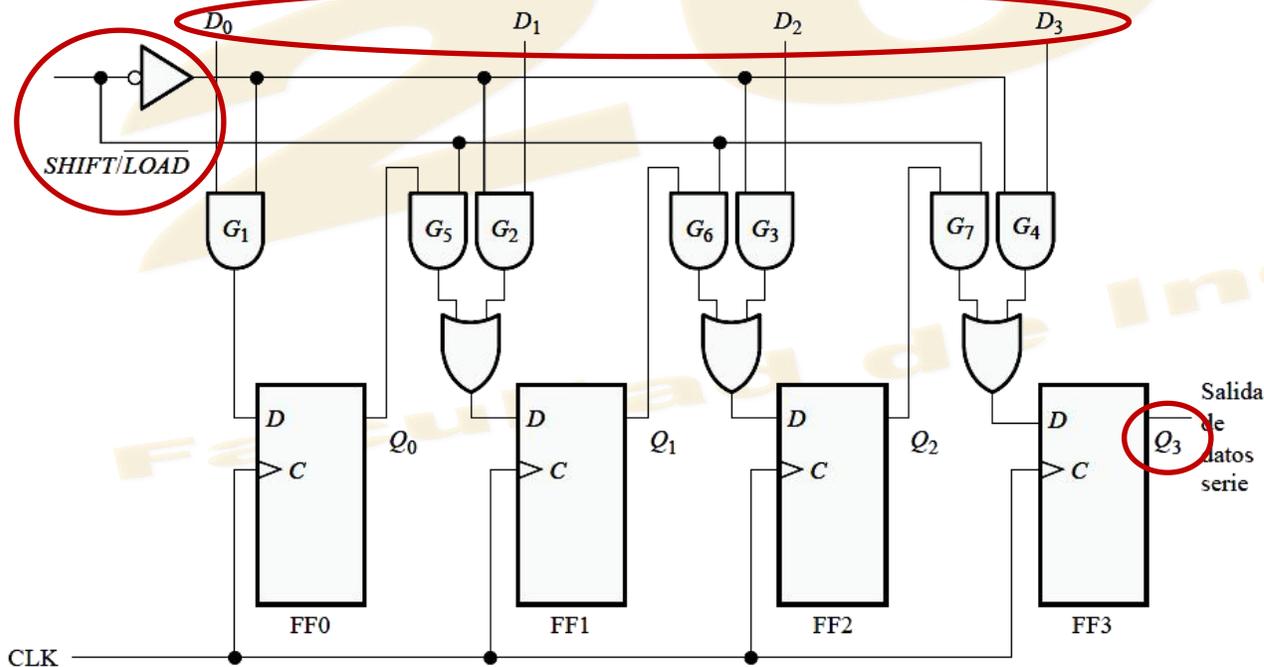
### Diagrama temporal

(Floyd, pg. 558)

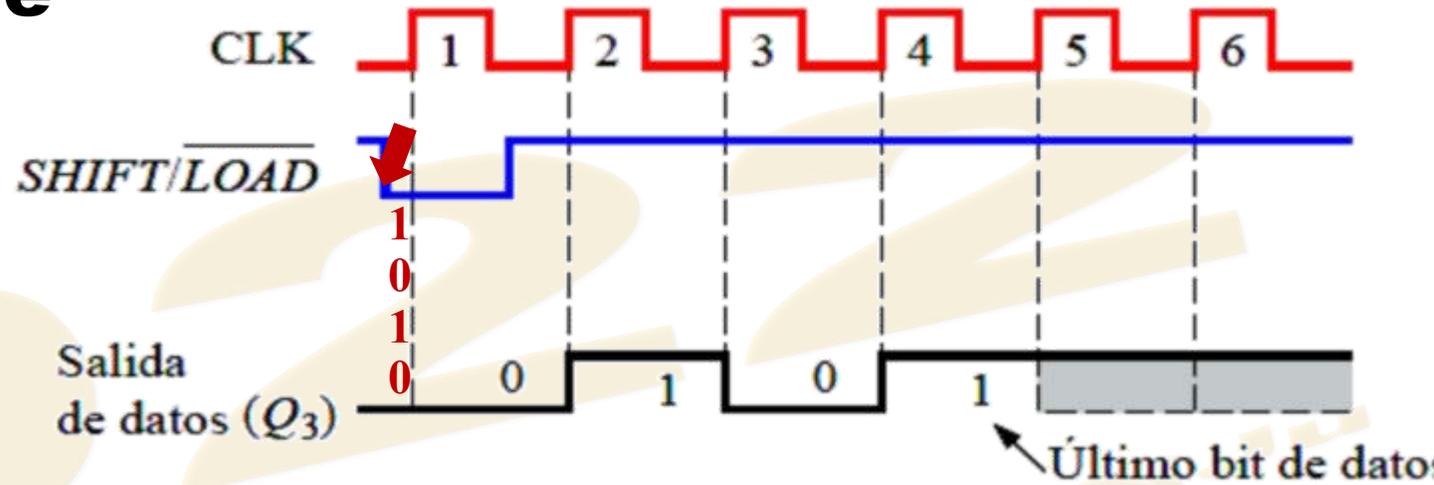
## Estructura paralelo - serie



**Bloque funcional**



**Diagrama temporal**



**Modelo de 4 bits con FF tipo D**

**CLK por flanco (+)**

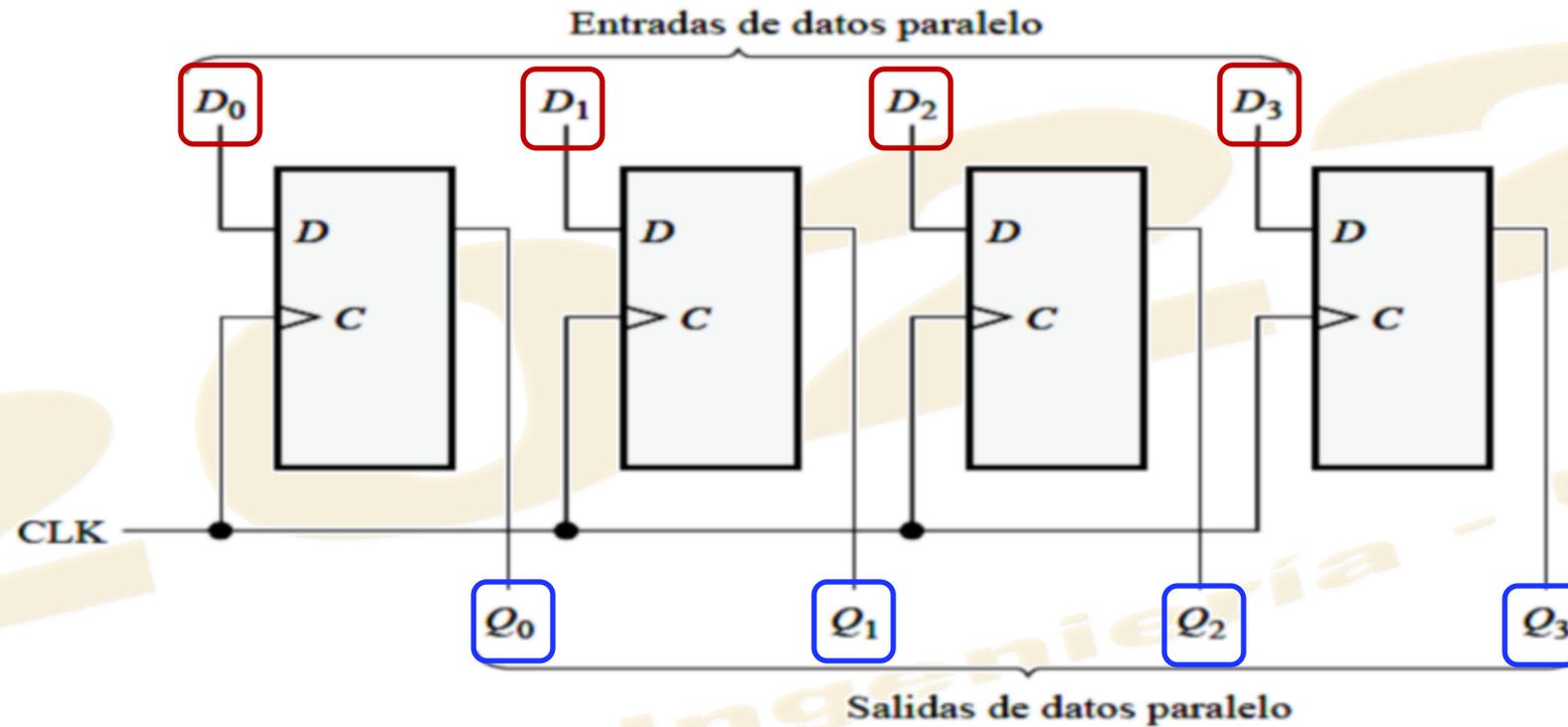
**Los datos ingresan por (D<sub>0</sub> D<sub>1</sub> D<sub>2</sub> D<sub>3</sub>) y se leen por FF3 (Q<sub>3</sub>)**

**Control para carga de datos (LOAD! con 0) y desplazamiento (SHIFT con 1)**

(Floyd, pg. 560)

## Estructura paralelo - paralelo

(Floyd, pg. 564)



**Modelo de 4 bits con FF tipo D. Esquema simplificado, precursor de la memoria.**

**CLK por flanco (+).**

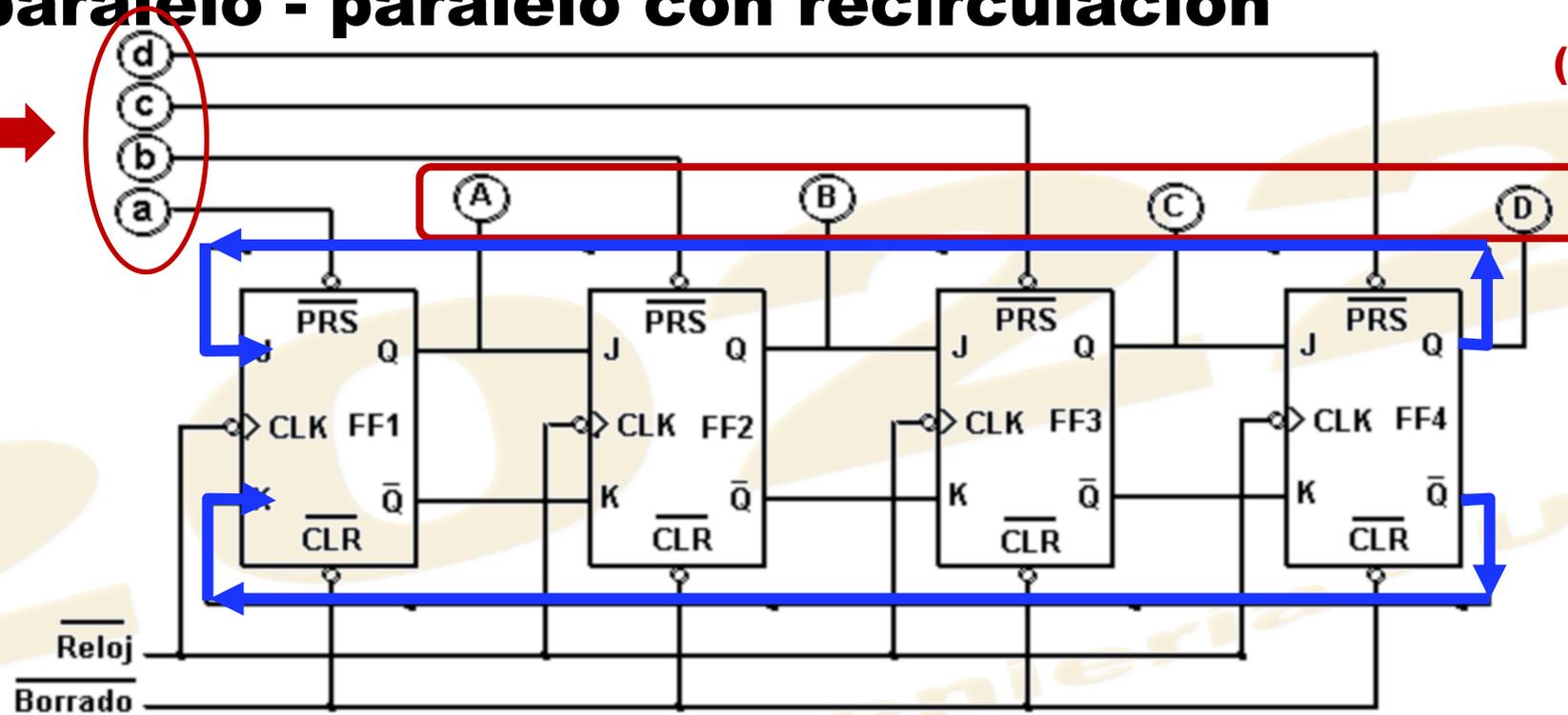
**Los datos ingresan por las líneas ( $D_0 D_1 D_2 D_3$ ) y se leen en las salidas de cada flip-flop ( $Q_0 Q_1 Q_2 Q_3$ ).**

**Acción rápida. Los datos están disponibles a la salida luego del 1º pulso de reloj.**

## Estructura paralelo - paralelo con recirculación

(Floyd, pg. 572)  
(Brown, pg. 400)

Ingreso de  
datos →



Modelo de 4 bits, con FF tipo J-K

CLK por flanco (-). Requiere borrado previo.

Los datos ingresan por las líneas PRS (a, b, c, d) **invertidos**. Sólo se cargan los datos [1].

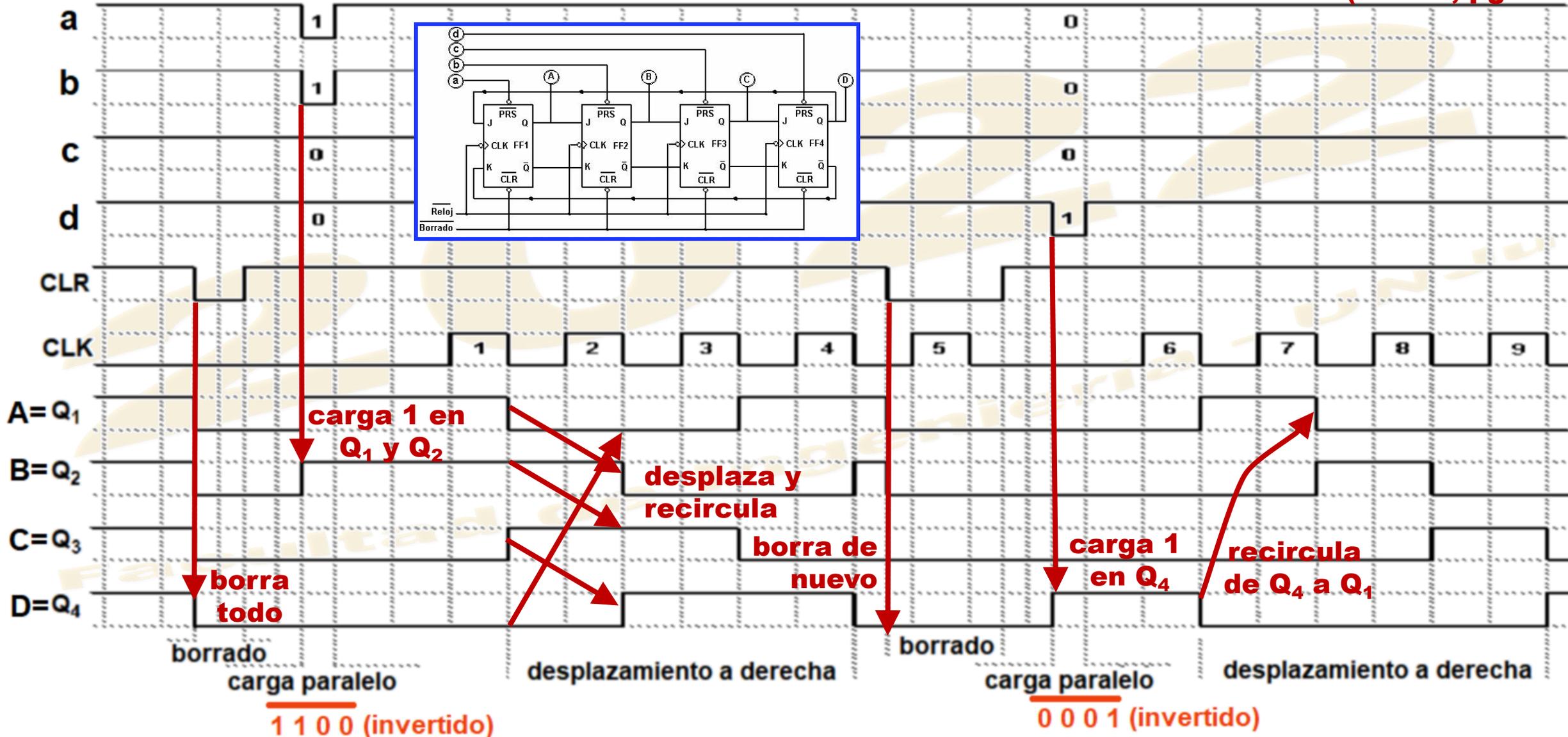
Los datos se leen en las **salidas** de cada flip-flop (A, B, C, D).

Los flip-flops J-K trabajan como **tipo D**.

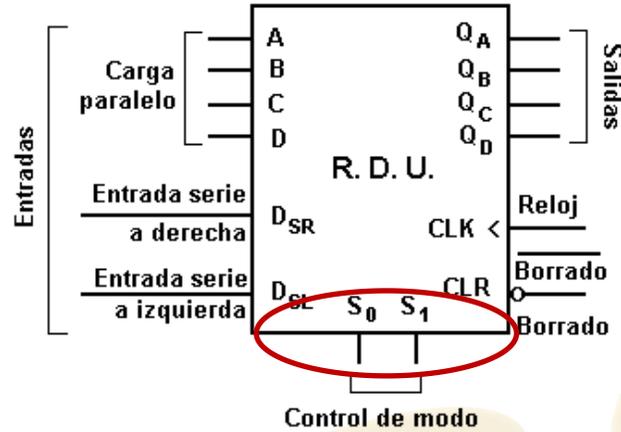
Las realimentaciones configuran un registro con **recirculación**.

## Estructura paralelo - paralelo con recirculación

(Floyd, pg. 572)  
(Brown, pg. 400)

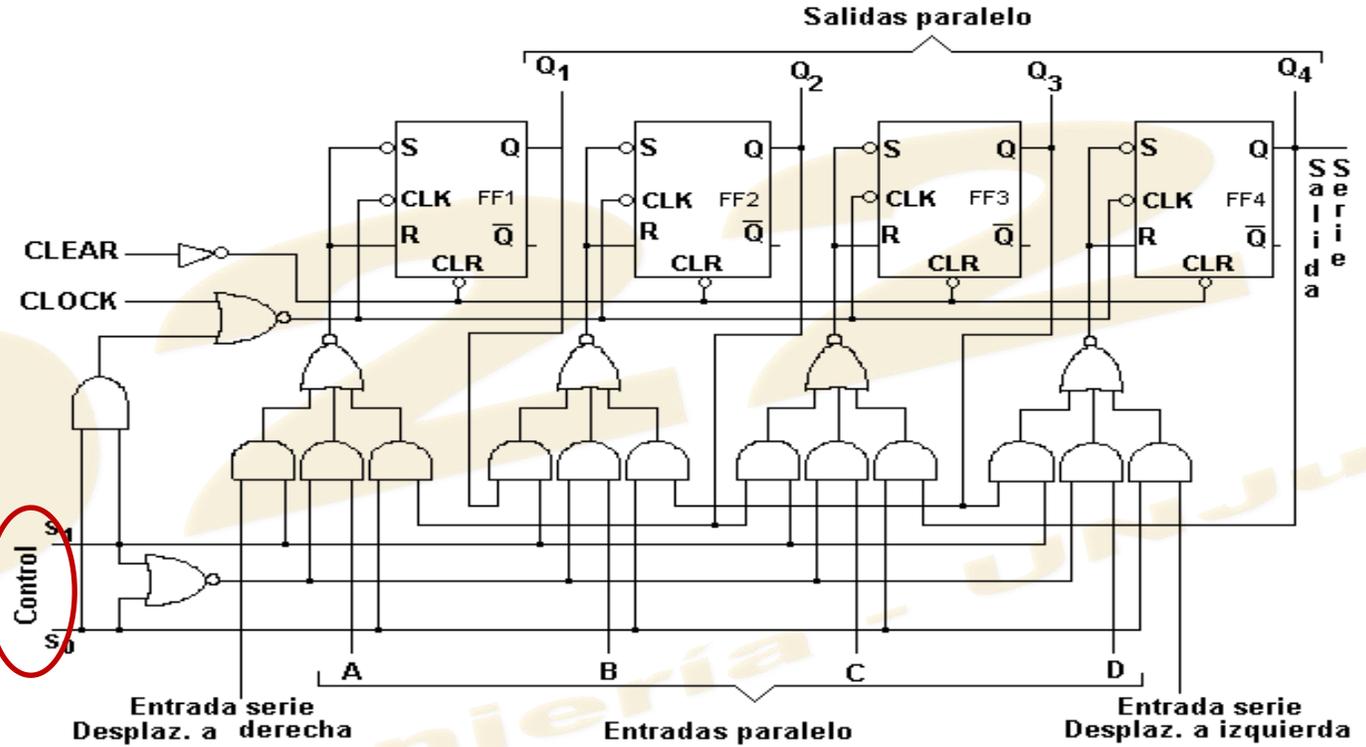


## Registro Universal



MODO DE OPERACIÓN	ENTRADAS							SALIDAS			
	CLK	CLR	S <sub>1</sub>	S <sub>0</sub>	D <sub>SR</sub>	D <sub>SL</sub>	D <sub>n</sub>	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>
Reset (borrado).	X	L	X	X	X	X	X	L	L	L	L
Mantenimiento (no opera).	X	H	0	0	X	X	X	q <sub>A</sub>	q <sub>B</sub>	q <sub>C</sub>	q <sub>D</sub>
Desplazamiento a izquierda.	↑	L	1	0	X	l	X	q <sub>B</sub>	q <sub>C</sub>	q <sub>D</sub>	L
	↑	H	1	0	X	h	X	q <sub>B</sub>	q <sub>C</sub>	q <sub>D</sub>	L
Desplazamiento a derecha.	↑	H	1	0	l	X	X	L	q <sub>B</sub>	q <sub>C</sub>	q <sub>D</sub>
	↑	H	1	0	h	X	X	L	q <sub>B</sub>	q <sub>C</sub>	q <sub>D</sub>
Carga paralelo.	↑	H	1	1	X	X	d <sub>n</sub>	d <sub>0</sub>	d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>

H nivel de tensión alto.  
h nivel de tensión alto un instante antes de L-H del reloj.  
L nivel de tensión bajo.  
l nivel de tensión bajo, un instante antes de L-H del reloj.  
d<sub>n</sub> (q<sub>n</sub>) entradas (salidas) un instante antes de L-H.  
X estado irrelevante.  
↑ transición de reloj L-H.  
\* la transición solo debe ocurrir mientras CLK está en alto.



**Unidad de 4 bits con FF S-R trabajando como FF D.**

**CLK por flanco (+).**

**El código ingresado por el *Control de Modo*, determina el tipo de acceso y configuración del registro.**

**(Floyd, pg. 568)**

## Concepto / Definición

Son unidades de **una** entrada y **N** salidas que muestran a la salida, en el código y secuencia preestablecidos, la cantidad de pulsos registrados en la entrada.

## Clasificación

Contadores binarios	{	Evolución interna	Asincrónicos
			Sincrónicos
	{	Módulo	Fijo
			Programable
{	Código de cuenta	Decimales	
		Otros códigos	
{	Progresión	Ascendentes	
		Descendentes	
		Bidireccionales	



En los **modelos asincrónicos**, la señal de cuenta (CLK) ingresa solamente al primer FF y luego se propaga a través de ellos.

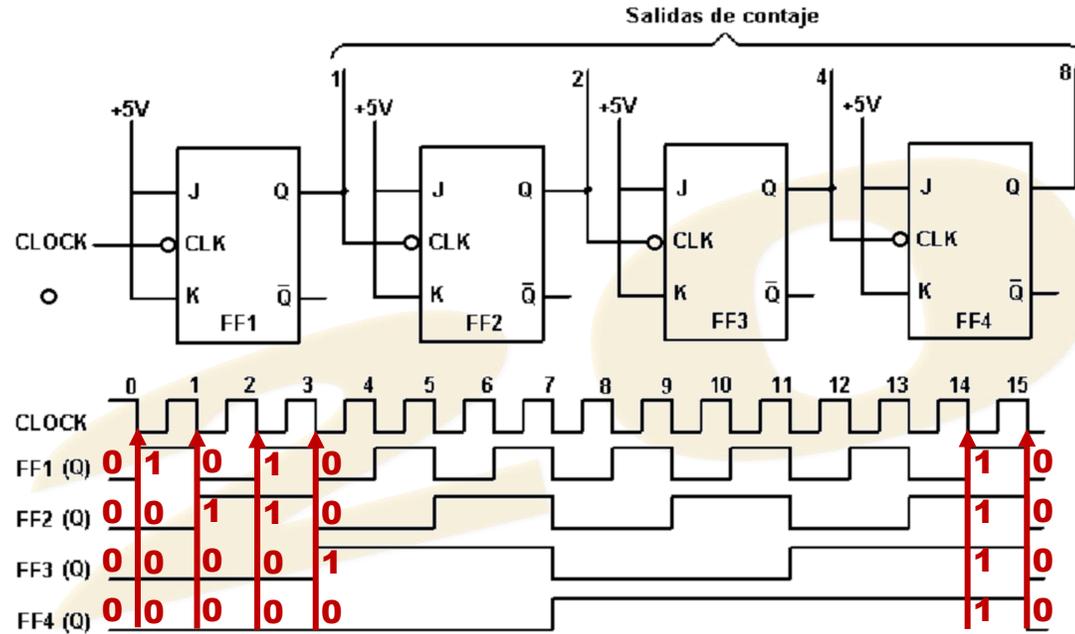
En los **modelos sincrónicos**, la señal de cuenta (CLK) llega simultáneamente a todos los FF.

Los restantes parámetros requeridos por el contador (**módulo, código de cuenta, progresión**), serán seleccionados e implementados durante el proceso de diseño.

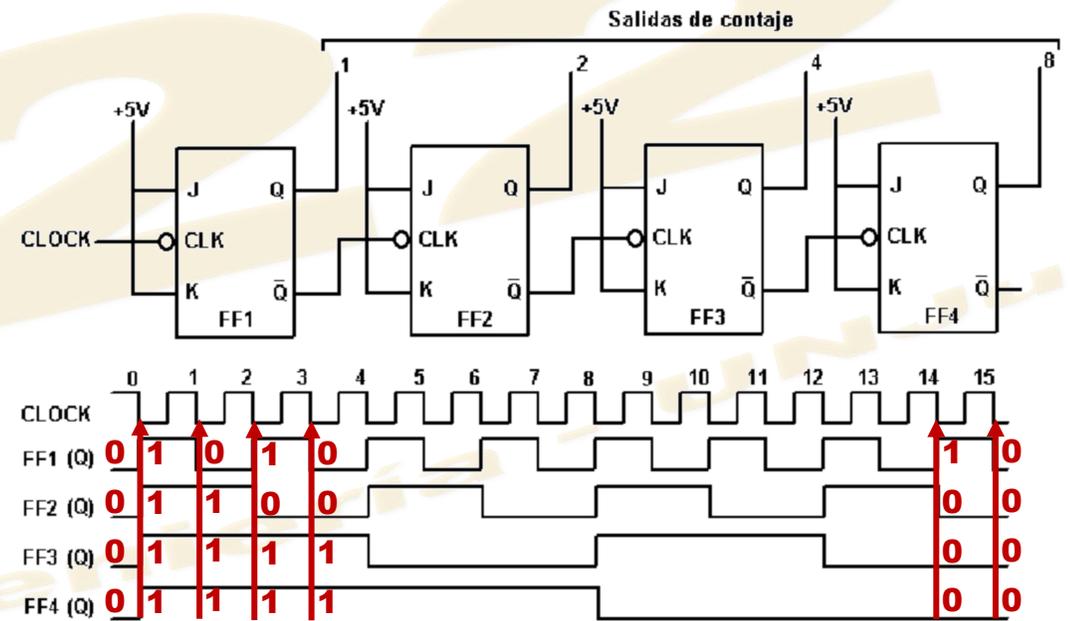
## Modelos básicos - binario natural

(Brown, pg. 402); Floyd. pg. 479

### Progresión creciente



### Progresión decreciente



**Cada flip-flop activa al siguiente ingresando por la línea de CLK.**

**Para que un FF se active, el anterior debe cambiar de estado, en el flanco correcto.**

**Los FF están configurados como tipo T.**

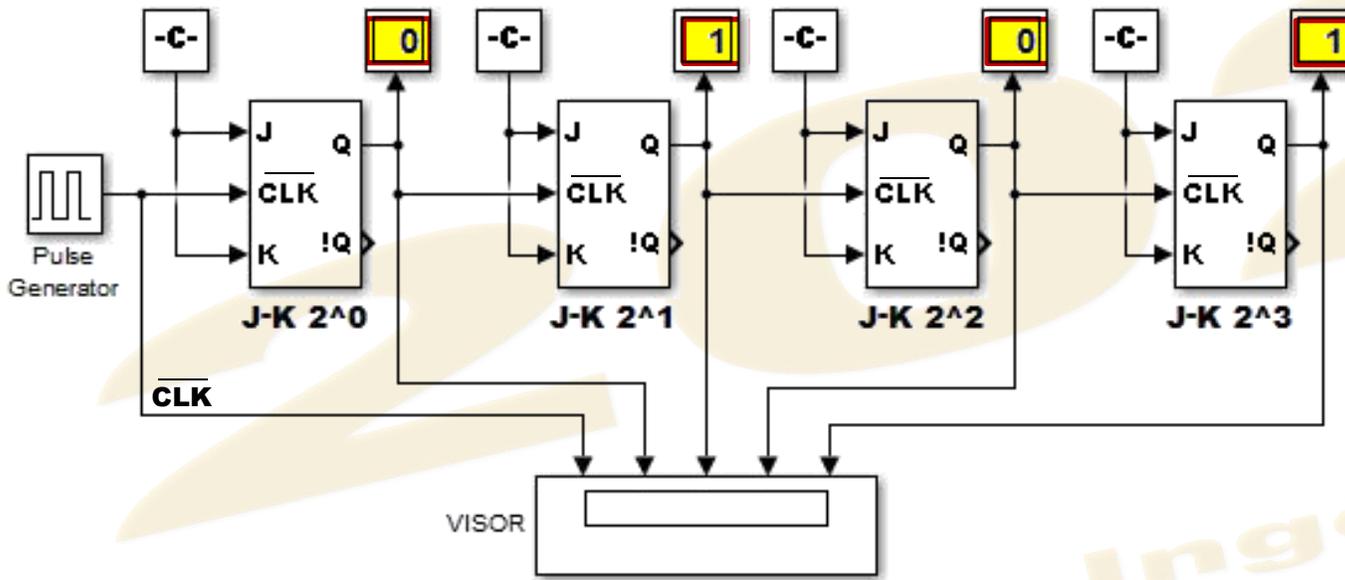
**Para progresión creciente, la activación se hace con la salida Q directa anterior.**

**Para progresión decreciente, la activación se hace con la salida Q negada anterior.**

## Ejemplo: Contador binario asincrónico 4 bits ascendente

Simulación en Matlab

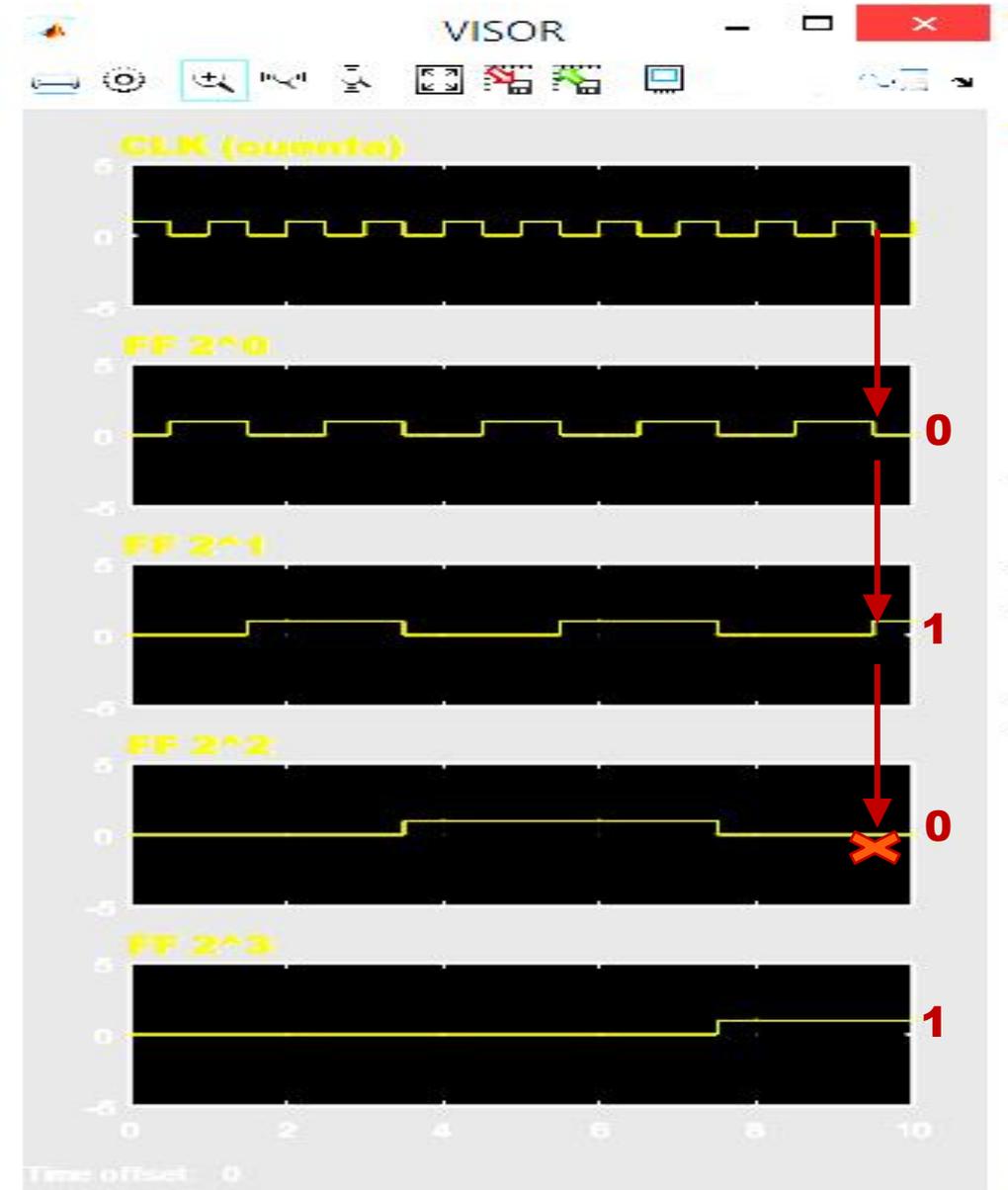
CONTADOR ASINCRÓNICO ASCENDENTE 4 BITS



Cada flip-flop activa al siguiente ingresando por la línea de CLK.

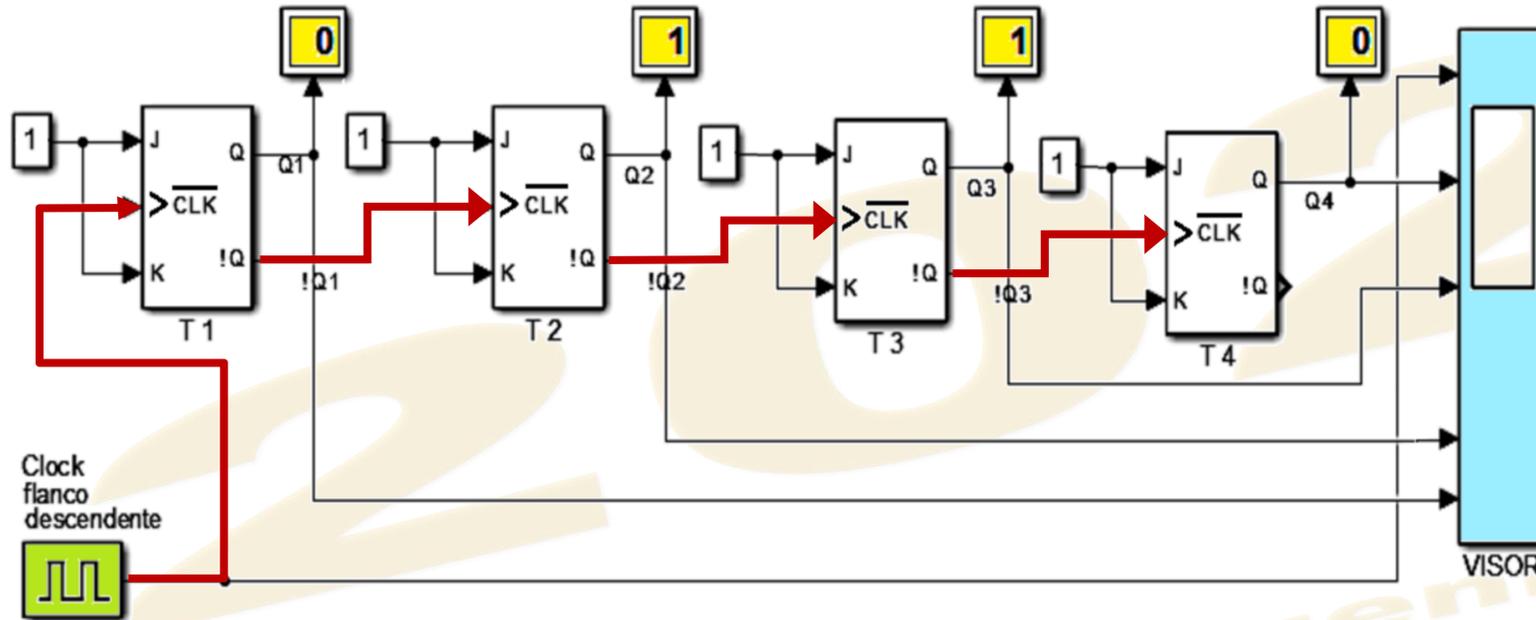
Los f-f están configurados como tipo T. Con el flanco descendente [-] cambian de estado.

Para conteo ascendente, la activación se hace con la salida Q directa anterior.



## Contador binario asincrónico 4 bits descendente

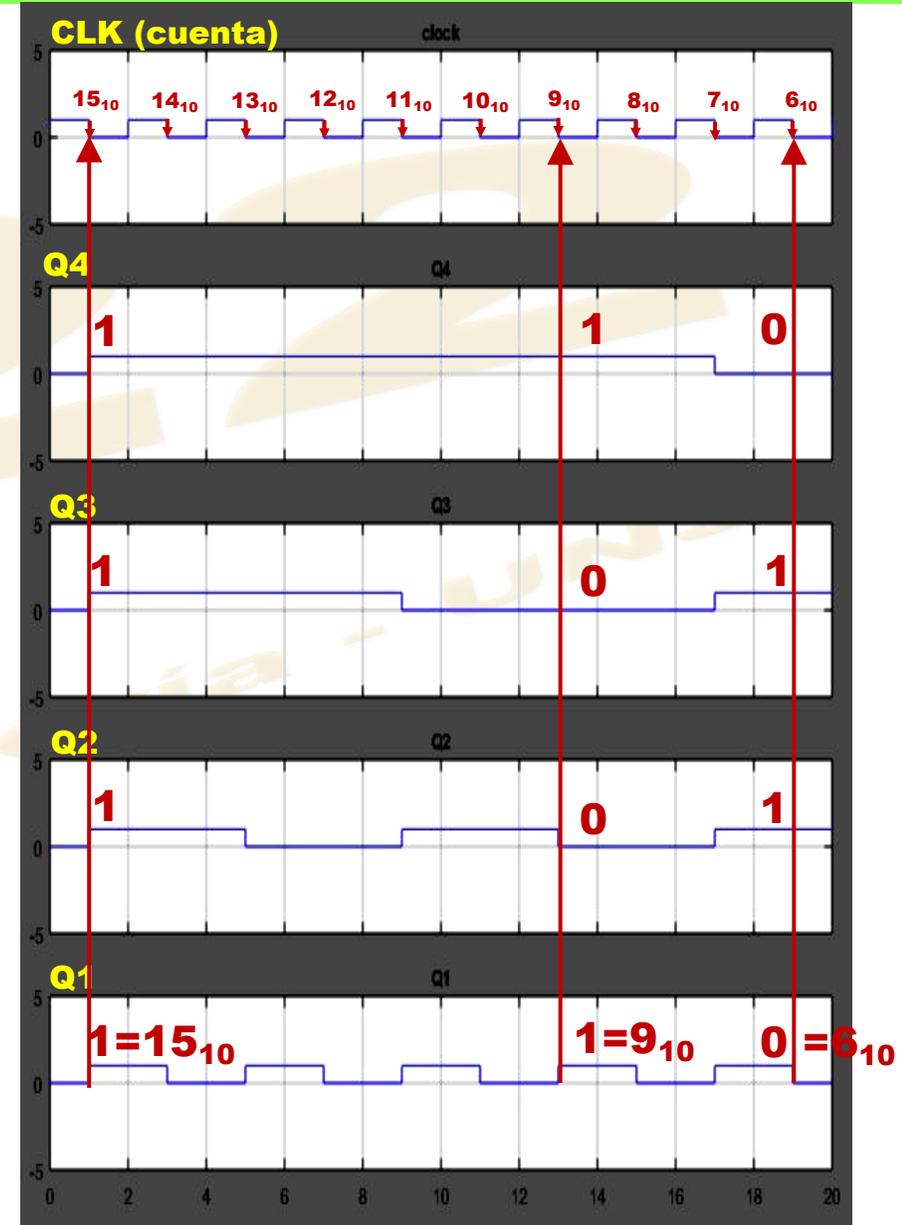
Simulación en Matlab



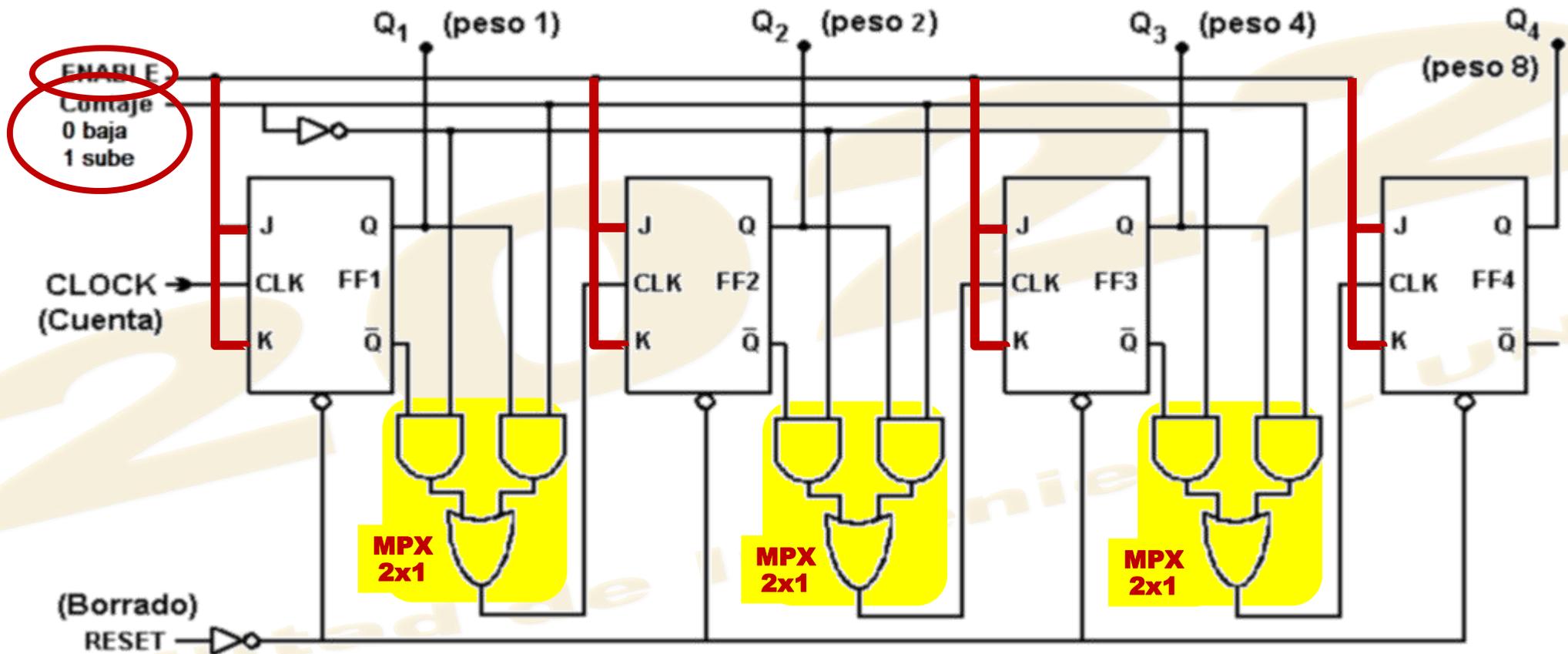
Los f-f están configurados como tipo T, activados por flanco descendente.

En el proceso asincrónico, la salida de cada flip-flop activa al siguiente ingresando por la línea de CLK. Sólo el primero recibe el CLK externo.

Para conteo descendente, la activación se hace con la salida Q **negada** anterior.



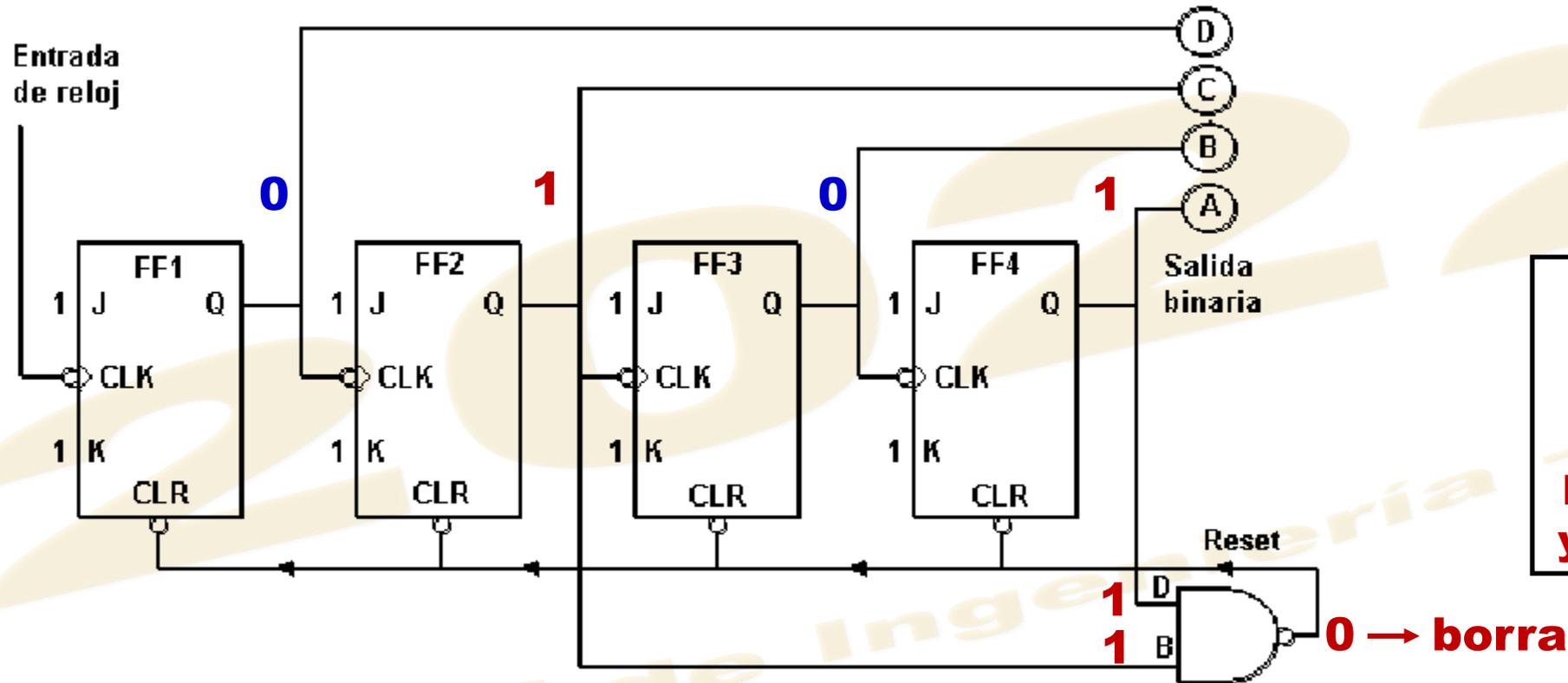
## Modelo bidireccional



Los f-f J-K están activados como f-f tipo T. **Eneable = 1** habilita la cuenta.  
La transferencia de [Q] ó [Q!] se realiza con MPX 2x1, todos simultáneamente.  
El ingreso en la línea de contaje [0 ó 1] determina qué canal de los MPX se activa.

## Modelo BCD natural

(Brown, pg. 399; Floyd, pg. 480)



**Cuenta final**  
**A B C D**  
**1 0 1 0**  
**(10 decimal)**  
**borra el contador**  
**y reinicia en 0000**

**Modelo de 4 bits, activado por flanco descendente.**

**Cuenta en binario natural de 4 bits.**

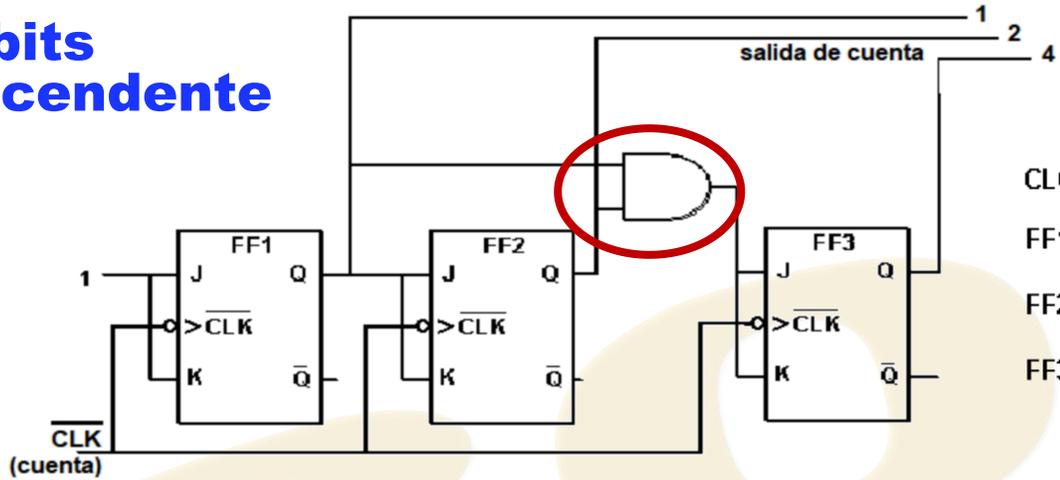
**Al llegar a la cuenta 10 decimal, la compuerta NAND activa las líneas de CLEAR.**

**No es muy eficiente.**

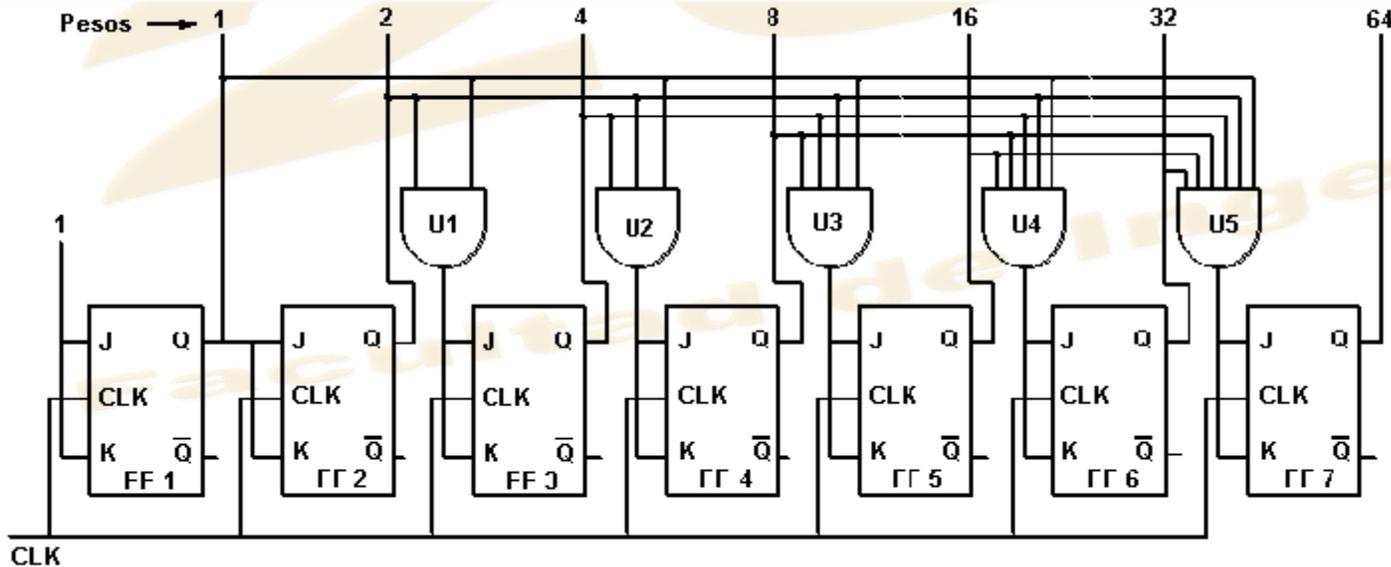
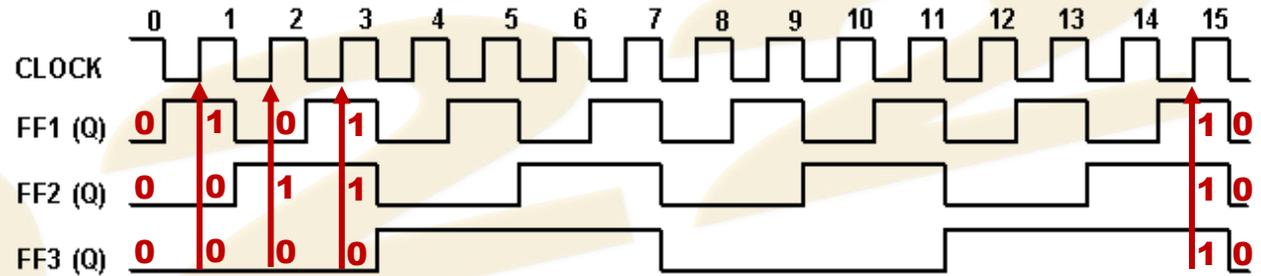
## Modelo básico - binario natural

(Brown, pg. 404)  
(Floyd, pg. 488)

**3 bits ascendente**



**diagrama de tiempos**



**7 bits - ascendente Diseño directo**

**Utiliza FF J-K configurados en formato T.**

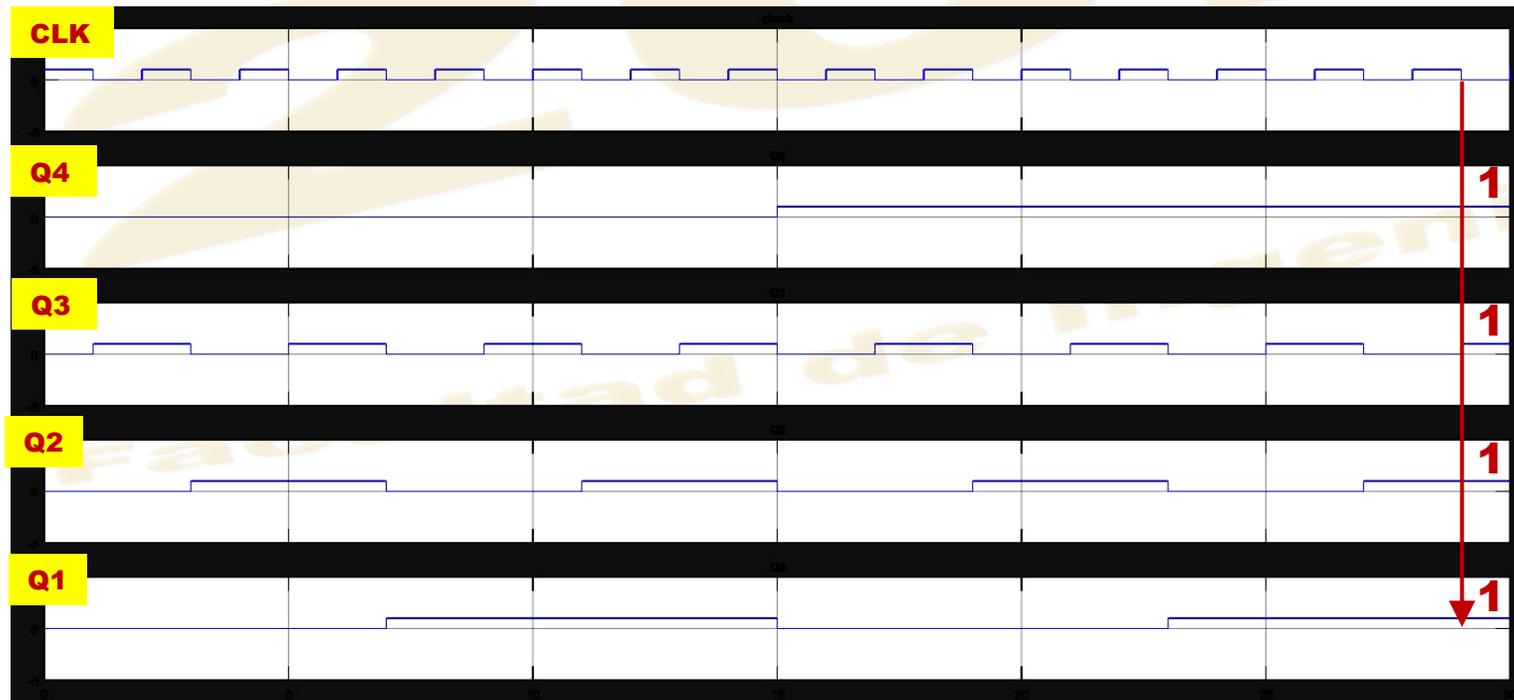
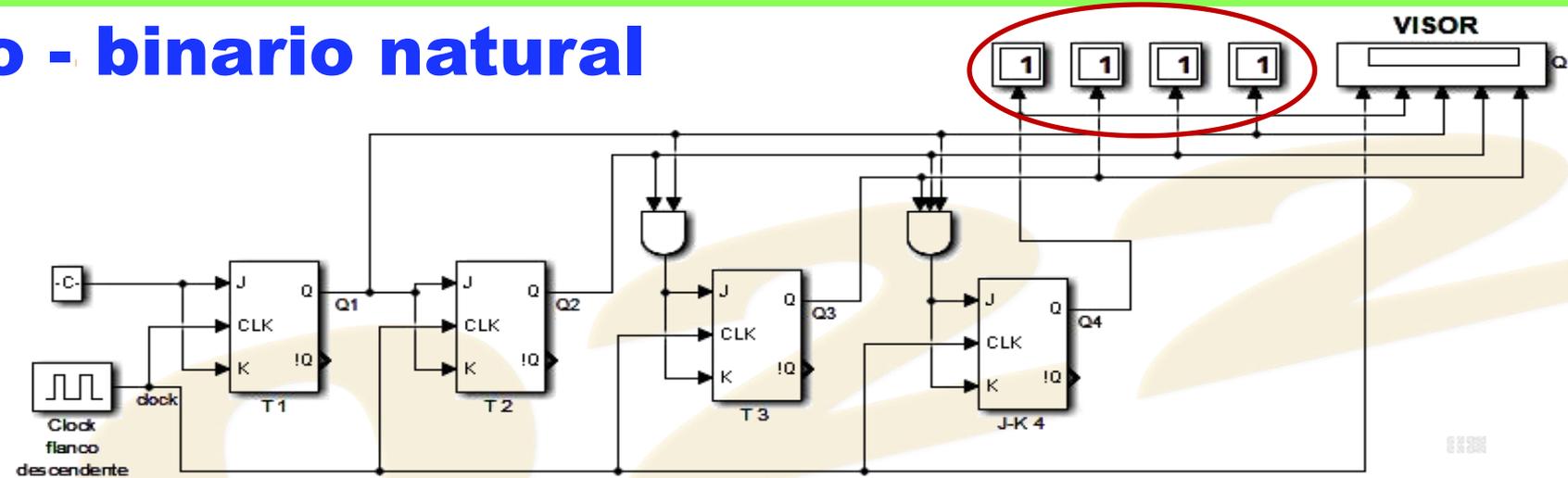
**Cada FF se alimenta con una AND que colecta todas las salidas anteriores.**

**Un FF no cambia de estado hasta que todos los anteriores estén en 1 (requerimiento para generar binario natural).**

**Se puede diseñar en forma directa. La limitación estaría en las compuertas AND más significativas.**

## Modelo básico - binario natural

**Modelo de simulación**



**Diagrama temporal**

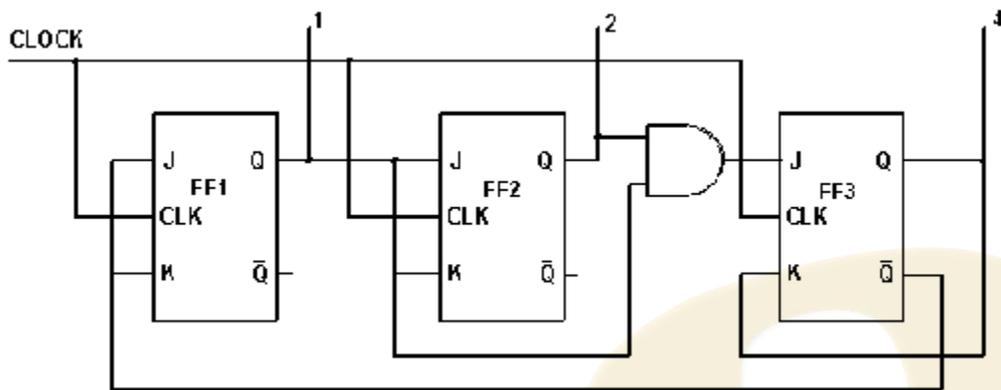
**Modelo de 4 bits activado por flanco [-].**

**Configurado con FF J-K conectados como FF T.**

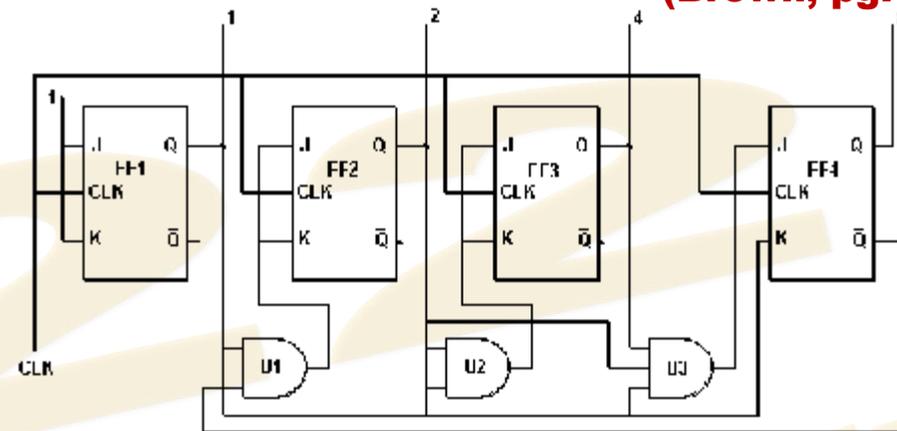
**Última cuenta [1 1 1 1] antes de reiniciar en [0 0 0 0].**

## Formatos de módulo distinto de $2^N$

(Brown, pg. 412)

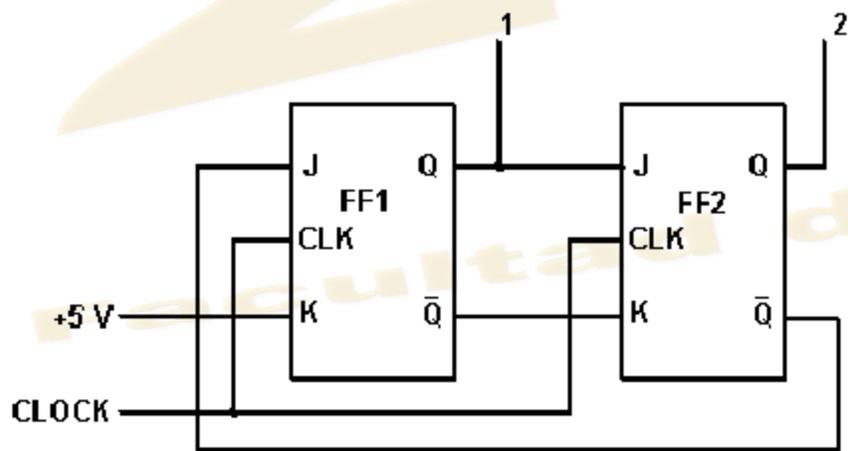


**Módulo 5 - ascendente**

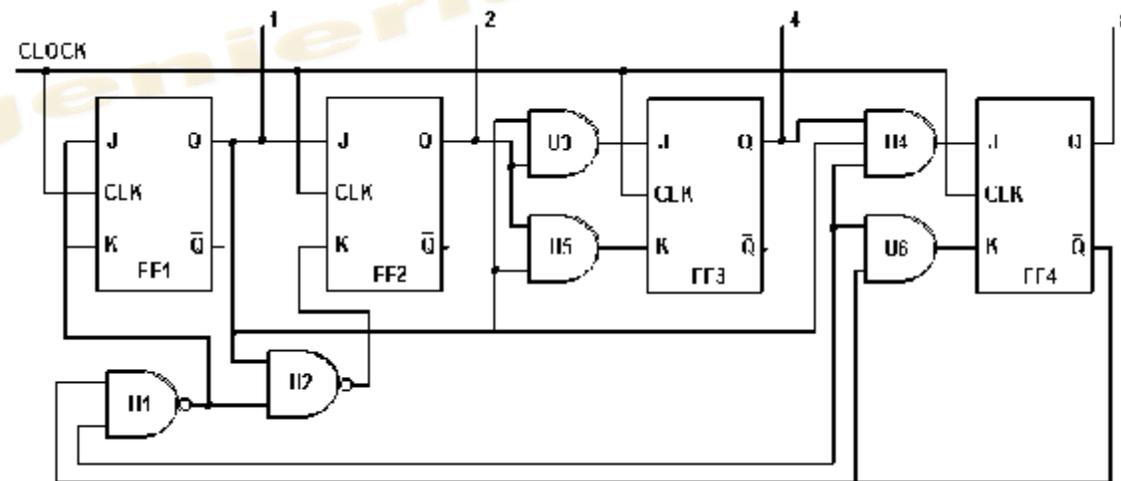


**BCD - ascendente**

**En general requieren diseño por T.V.**



**Módulo 3 - ascendente**



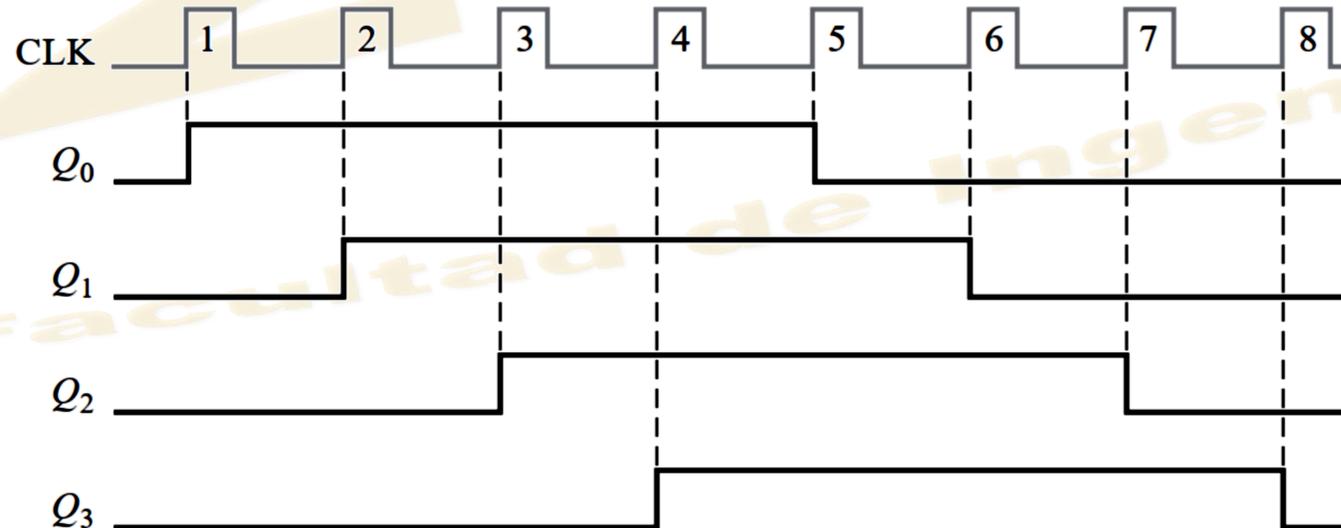
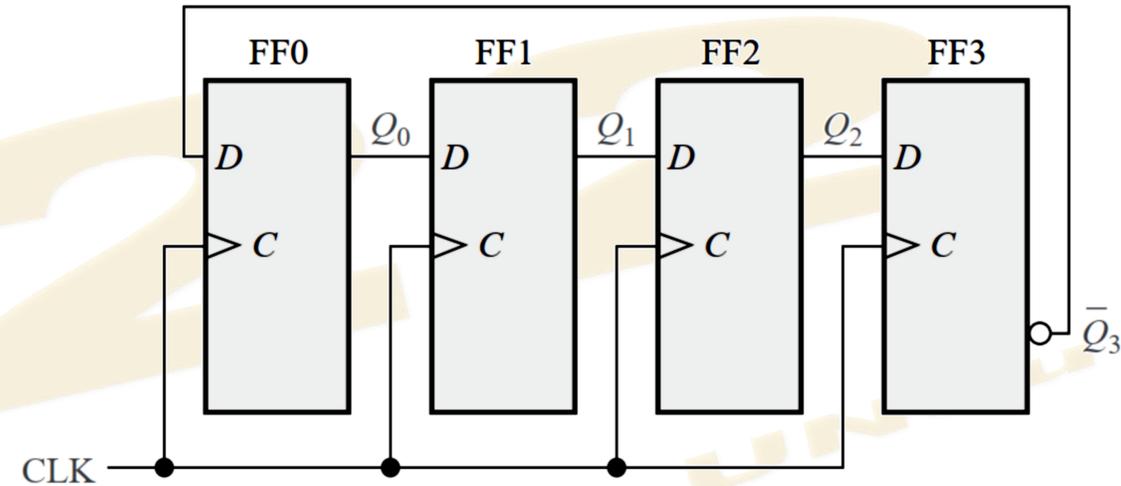
**Módulo 11 - ascendente**

## Formatos de módulo distinto de $2^N$

(Floyd, pg. 571)

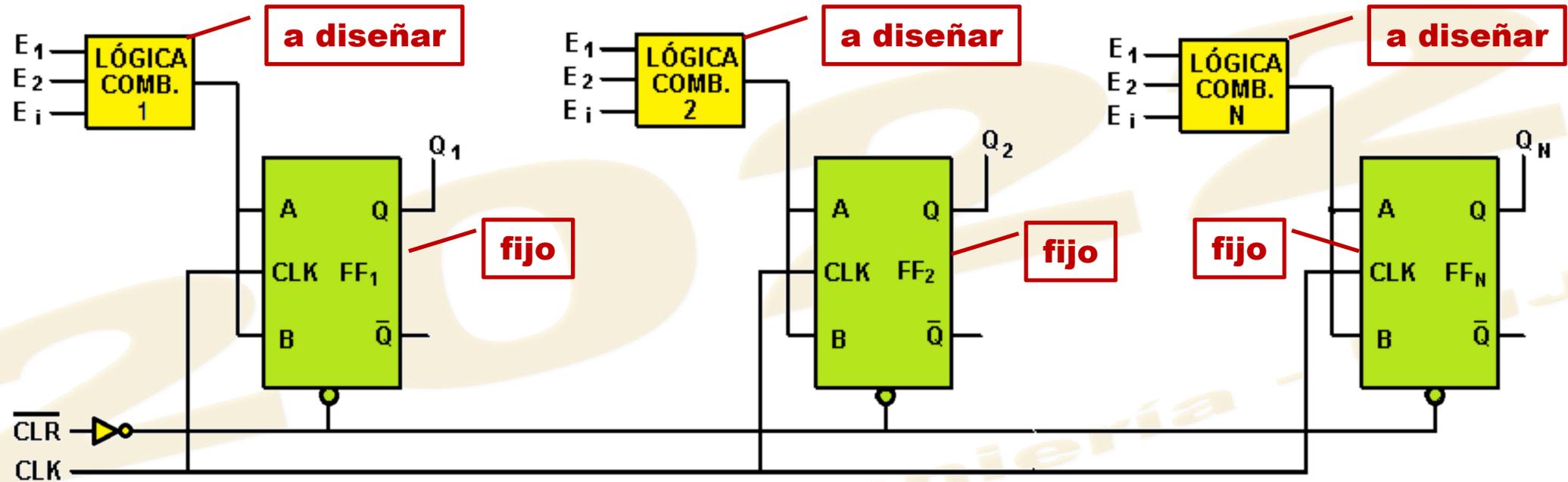
### Contador Johnson de 4 bits con FF tipo D

**Este modelo no  
requiere diseño  
por T.V.**



**Secuencia  
de  
temporización**

## Esquema de diseño genérico - base flip-flops



La cantidad de FF se establece de acuerdo a la longitud del código.

El tipo de FF, en general, es a criterio del diseñador.

El diseño específico es para los bloques combinatoriales que alimentan a las entradas de los FF.

Los bloques combinatoriales deben generar las entradas adecuadas para que los FF generen en sus salidas el código requerido.

Todo el proceso se desarrolla sobre la tabla de estados.

## Proceso de diseño

Se define tipo y cantidad de FF a utilizar, según el código elegido (ej. Johnson M1PD).

Fundamentalmente consiste en diseñar los bloques combinatoriales que alimentarán a cada FF. Habrán tantos circuitos como FF y entradas se definan.

Se genera la tabla de secuencia (o tabla de estados o tabla de transiciones).

Las columnas de estados anteriores y estados actuales contienen al código a generar, en tipo y progresión. Están desfasadas un paso.

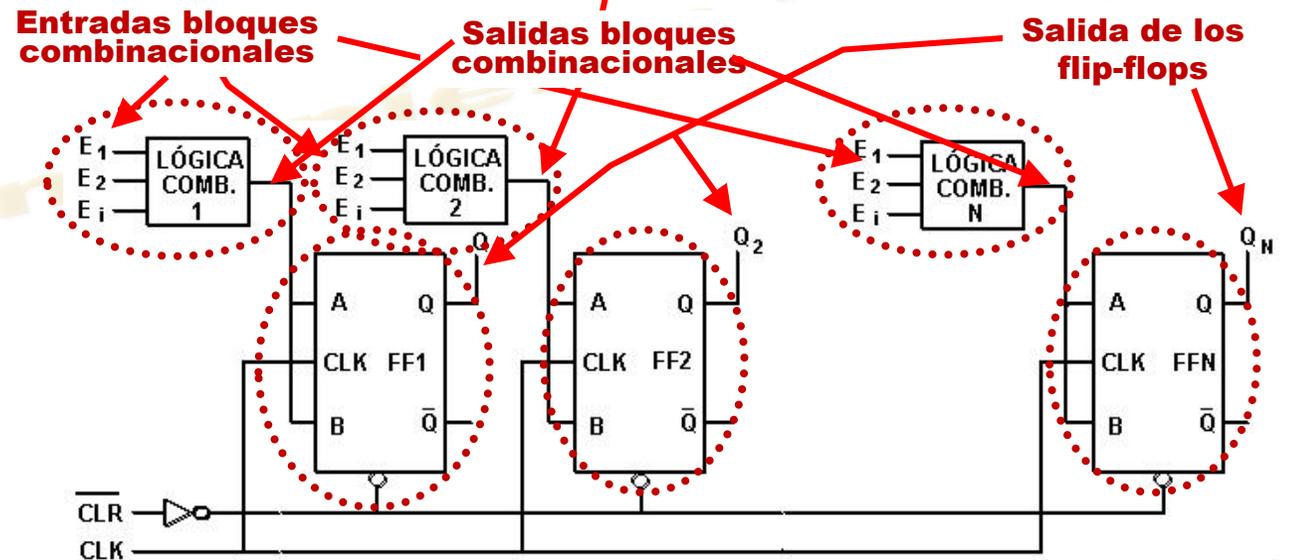
La primera palabra de estados anteriores y la última de estados actuales coinciden (se reinicia la secuencia).

Los estados anteriores son las entradas de los bloques combinatoriales.

Las entradas de los FF son las salidas de los bloques combinatoriales para el diseño.

Los estados actuales son las salidas de los FF (las que generarán la cuenta).

ESTADOS ANTERIORES			ENTRADAS DE LOS FLIP-FLOPS						ESTADOS ACTUALES		
Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	J <sub>3</sub>	K <sub>3</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>
0	0	0	1	0	0	1	0	1	1	0	0
1	0	0	1	0	1	0	0	1	1	1	0
1	1	0	1	0	1	0	1	0	1	1	1
1	1	1	0	1	1	0	1	0	0	1	1
0	1	1	0	1	0	1	1	0	0	0	1
0	0	1	0	1	0	1	0	1	0	0	0

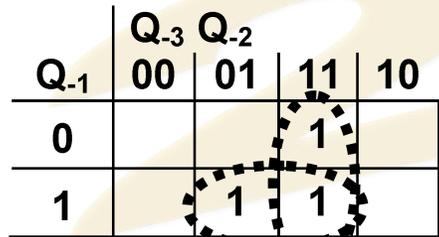
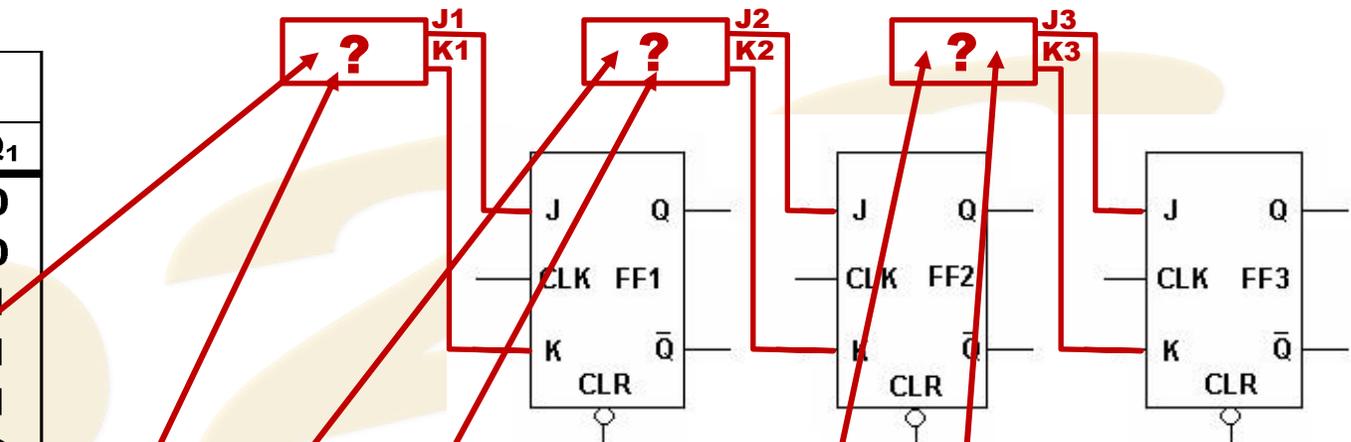


(Floyd, pg. 499)

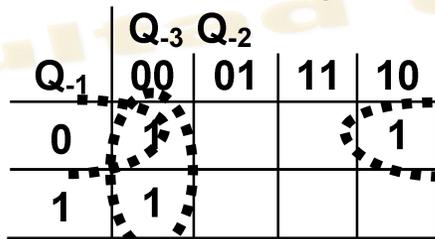
# CONTADORES BINARIOS SINCRÓNICOS

**Ej. Contador síncrono Johnson de 3 bits (marca 1 - progresión a izquierda), descendente con flip-flop J-K .**

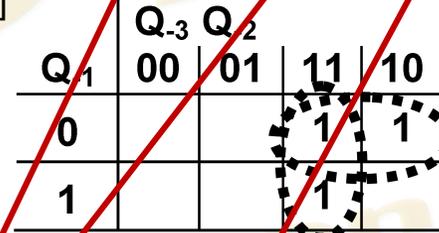
ESTADOS ANTERIORES			ENTRADAS DE LOS FLIP-FLOPS						ESTADOS ACTUALES		
Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	J <sub>3</sub>	K <sub>3</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>
0	0	0	1	0	0	1	0	1	1	0	0
1	0	0	1	0	1	0	0	1	1	1	0
1	1	0	1	0	1	0	1	0	1	1	1
1	1	1	0	1	1	0	1	0	0	1	1
0	1	1	0	1	0	1	1	0	0	0	1
0	0	1	0	1	0	1	0	1	0	0	0



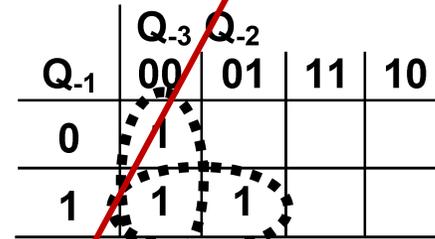
$$J_1 = Q_3 \cdot Q_2 + Q_2 \cdot Q_1$$



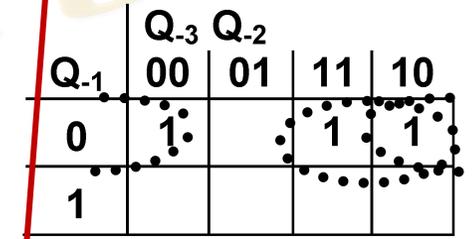
$$K_1 = \bar{Q}_3 \cdot \bar{Q}_2 + \bar{Q}_2 \cdot \bar{Q}_1$$



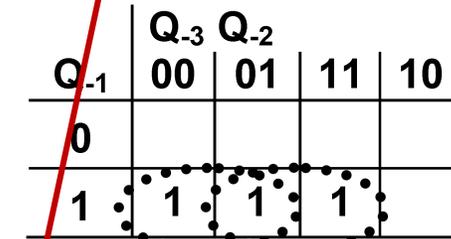
$$J_2 = Q_3 \cdot Q_2 + Q_3 \cdot \bar{Q}_1$$



$$K_2 = \bar{Q}_3 \cdot \bar{Q}_2 + \bar{Q}_3 \cdot Q_1$$



$$J_3 = Q_3 \cdot \bar{Q}_1 + \bar{Q}_2 \cdot \bar{Q}_1$$



$$K_3 = \bar{Q}_3 \cdot Q_1 + Q_2 \cdot Q_1$$

# CONTADORES BINARIOS SINCRÓNICOS

**Ej. Contador síncrono Johnson de 3 bits (marca 1 - progresión a izquierda), descendente con flip-flop J-K (continuación).**

$$J_3 = Q_3 \cdot \bar{Q}_1 + \bar{Q}_2 \cdot \bar{Q}_1$$

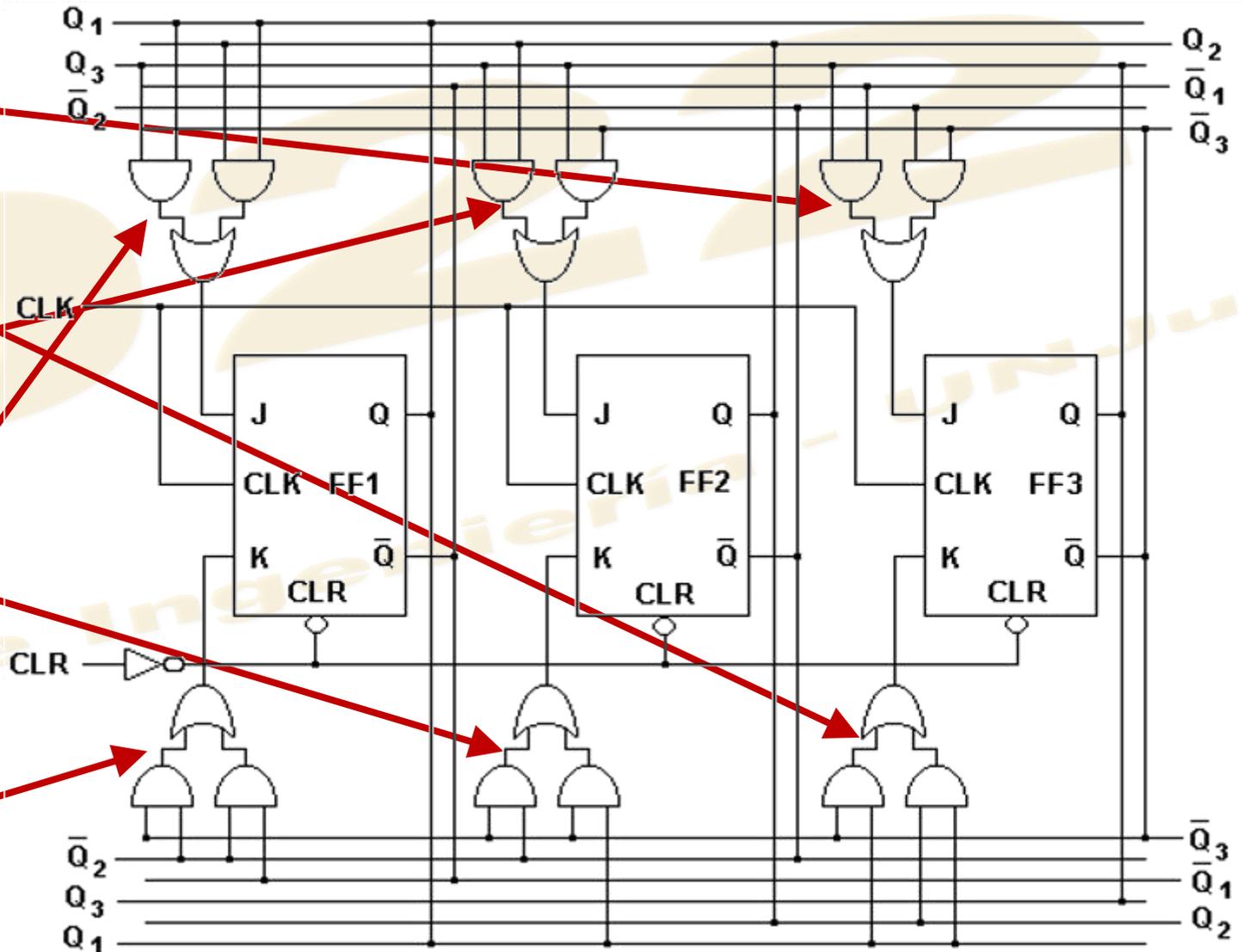
$$K_3 = \bar{Q}_3 \cdot Q_1 + Q_2 \cdot Q_1$$

$$J_2 = Q_3 \cdot Q_2 + Q_3 \cdot \bar{Q}_1$$

$$K_2 = \bar{Q}_3 \cdot \bar{Q}_2 + \bar{Q}_3 \cdot Q_1$$

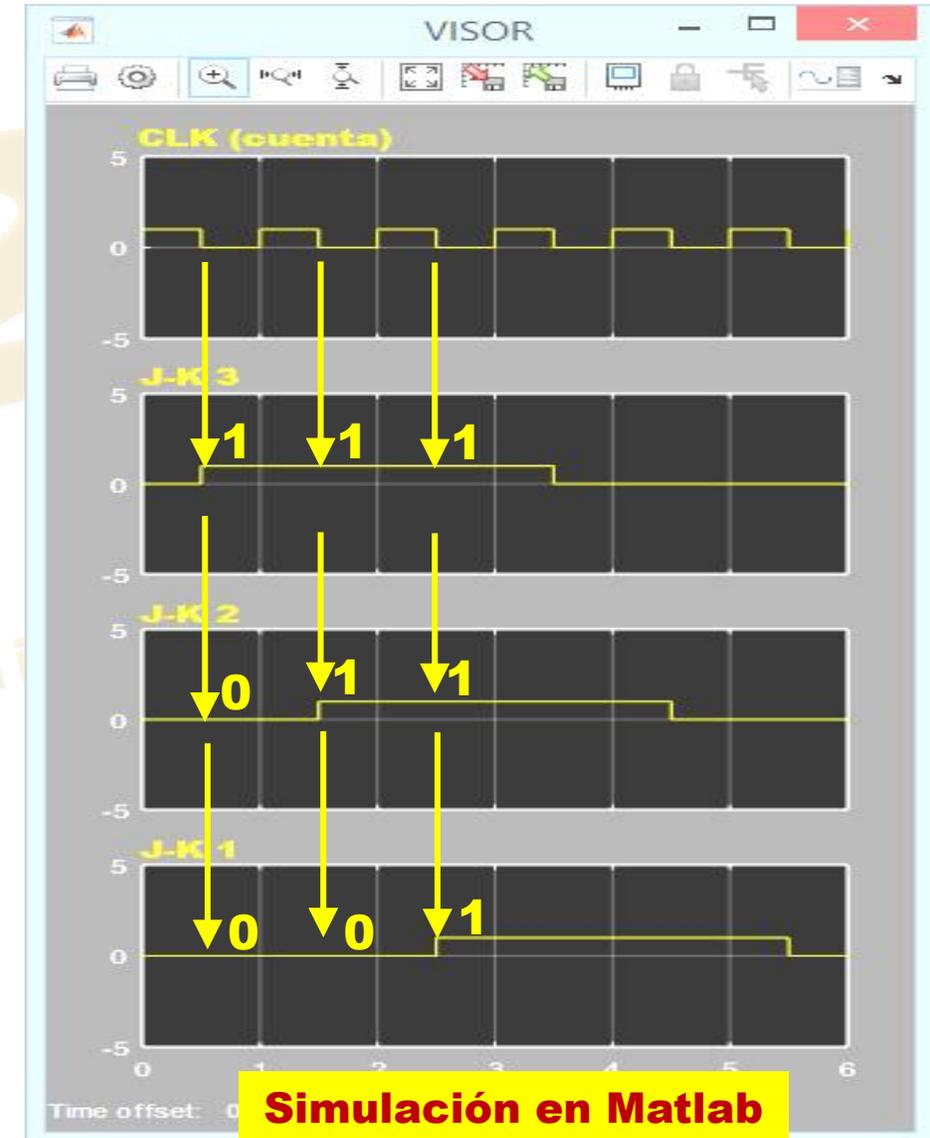
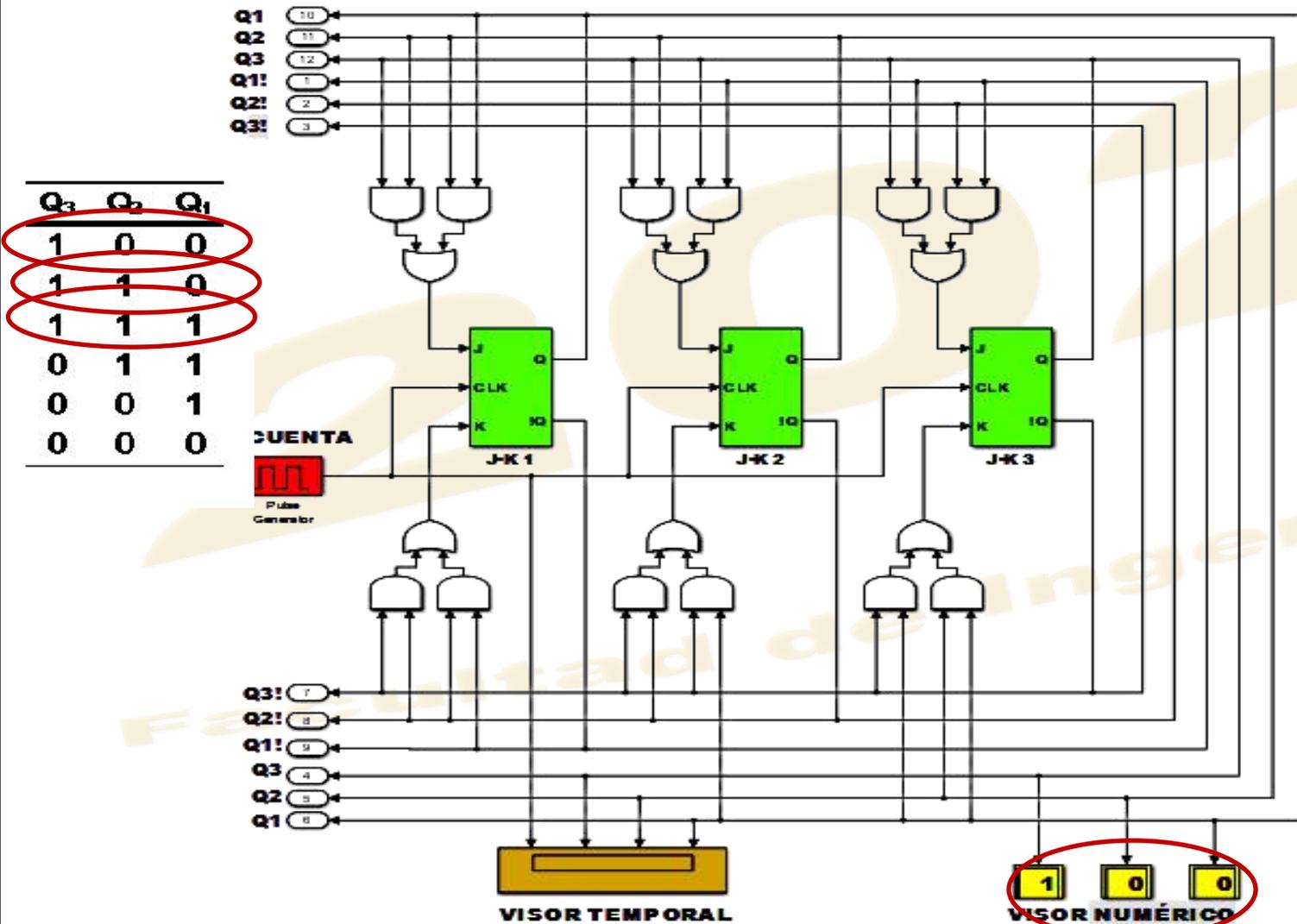
$$J_1 = Q_3 \cdot Q_2 + Q_2 \cdot Q_1$$

$$K_1 = \bar{Q}_3 \cdot \bar{Q}_2 + \bar{Q}_2 \cdot \bar{Q}_1$$



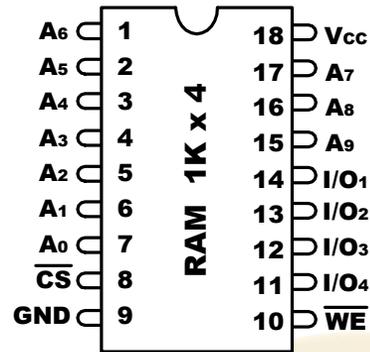
# CONTADORES BINARIOS SINCRÓNICOS

Ej. Contador síncrono Johnson de 3 bits (marca 1 - progresión a izquierda), descendente con flip-flop J-K (continuación).

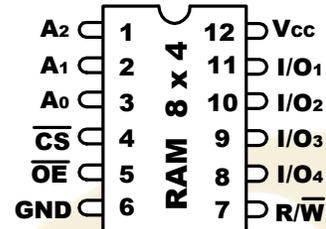


## Ej. Memoria RAM de 8x4.

Esquema real



Esquema reducido (ficticio)



- Memoria RAM con flip-flop tipo D.
- 8 registros de 4 bits
- Bus de direcciones  $A_0 A_1 A_2$  (3 bits).
- Bus de datos bidireccional (4 bits)  $DIO_0 DIO_1 DIO_2 DIO_3$
- Lectura/escritura R/W!
- Selección de chip CS!
- Habilitación de salida OE!

