

## 66.70 Estructura del Computador

### **Registros y contadores**

# Registros

**Se caracterizan por:**

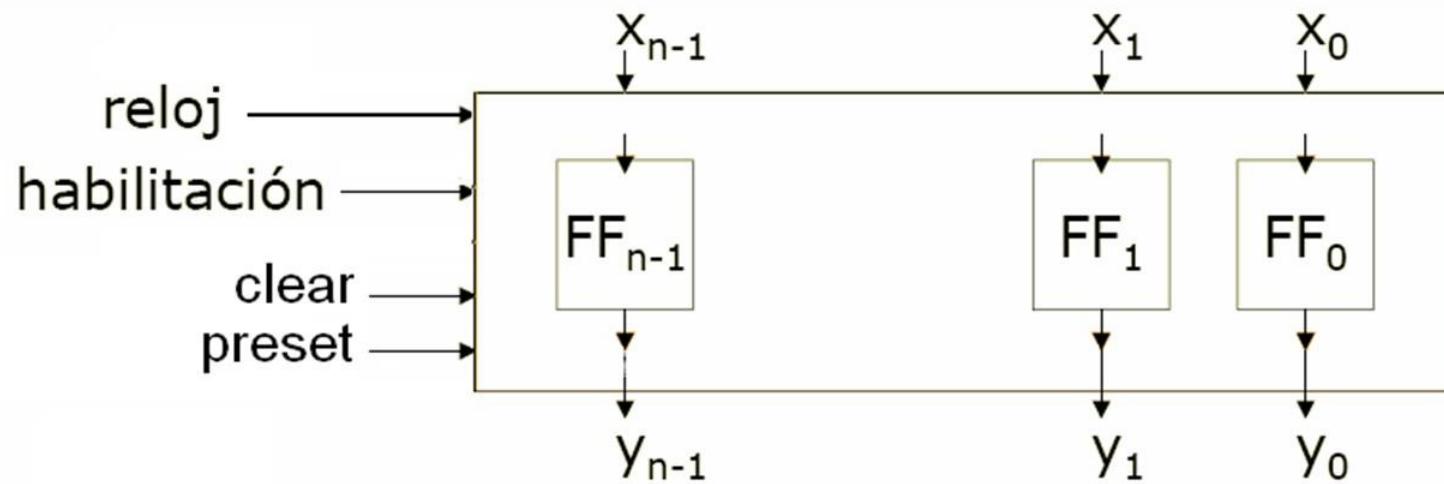
- Tienen capacidad de guardar información
- Su estructura:  $n$  FlipFlops operando sincrónicamente + lógica de compuertas

**Aplicaciones:**

- Almacenamiento (memoria)
- Desplazamiento (fines lógicos o aritméticos) (conversión de datos)

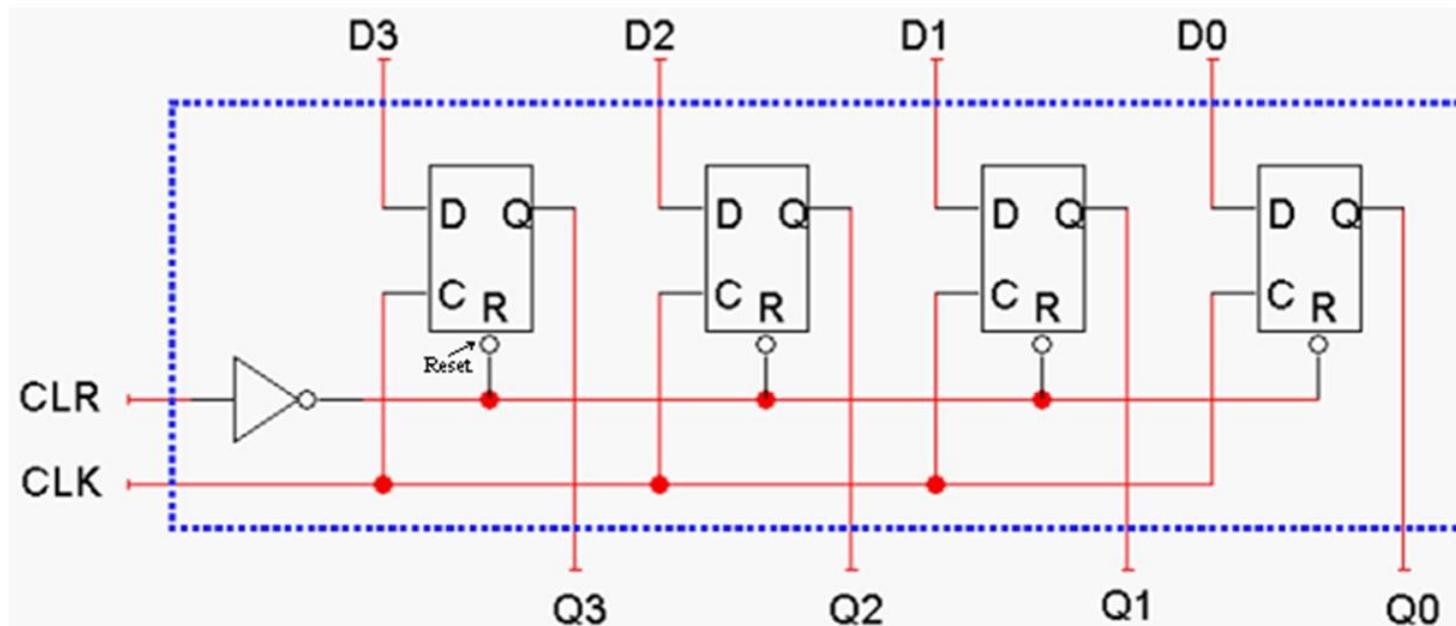
Entrada	Salida	Aplicación
Serie	Serie	Almacenamiento
Serie	Paralelo	Conversión
Paralelo	Serie	Conversión
Paralelo	Paralelo	Almacenamiento

# Registros



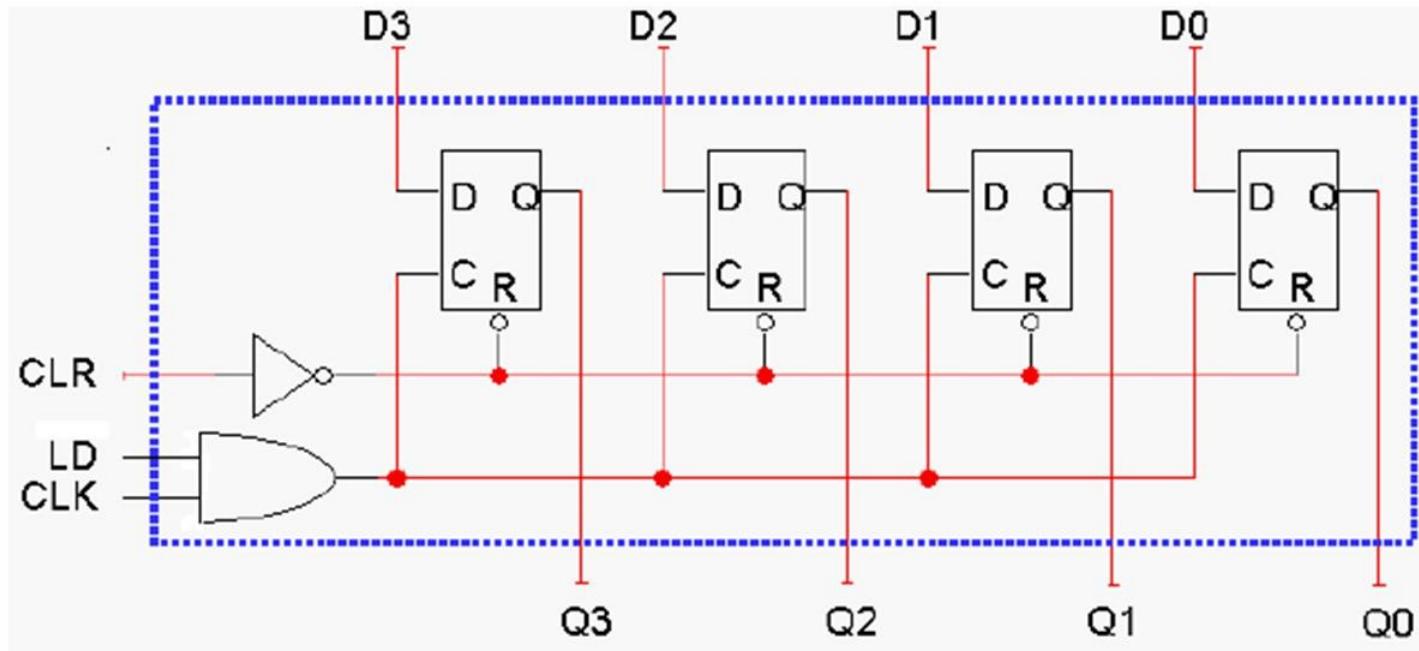
*Registro paralelo-paralelo de  $n$  bits*

# Registros de almacenamiento



- Con cada ciclo de reloj la entrada de 4 bits es copiada en la salida

# Registros de almacenamiento



Modifica la entrada de clock de cada FF

- LD=0 previene la aparición de flancos de reloj a la entrada de los FF
- LD=1 el reloj es aplicado a todos los FlipFlops

# Registros de almacenamiento

## Campos de aplicación

En un microprocesador

- ✓ Se usan para presentar los operadores a la Unid. Aritmético Lógica (ALU)
- ✓ Se usan para guardar los resultados de operaciones aritméticas-lógicas
- ... y más

Capacidad de almacenamiento

CPU	Cantidad de Reg.	Tamaño	Cache			RAM
			L1	L2	L3	
Pentium 4	8	32 bits	8 kB	512 kB		> 0.5 GB
Athlon 64	16	64 bits	64 kB	1024 kB		> 0.5 GB
Core 2 duo	16	64 bits	64 kB	6000 kB		> 2 GB
i7	16	64 bits	64 kB	256 kB	8000 kb	> 4GB
PowerPC 970 (mac G5)	32	64 bits	64 kB	512 kB		> 1 GB

# Registros de almacenamiento

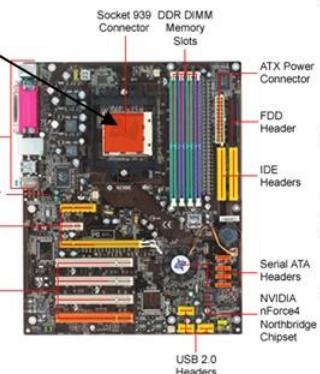
## Campos de aplicación

CPU	Cantidad de Reg.	Tamaño	Cache			RAM
			L1	L2	L3	
Pentium 4	8	32 bits	8 kB	512 kB		> 0.5 GB
Athlon 64	16	64 bits	64 kB	1024 kB		> 0.5 GB
Core 2 duo	16	64 bits	64 kB	6000 kB		> 2 GB
i7	16	64 bits	64 kB	256 kB	8000 kb	> 4GB
PowerPC 970 (mac G5)	32	64 bits	64 kB	512 kB		> 1 GB

Los registros ocupan la parte más cara del motherboard

El cache es memoria muy rápida,  
... pero no tan rápida como los registros

- › Desventaja de los registros: son caros!
- › La mayoría de los programas requieren más memoria  
de la que los registros pueden proveer => necesitamos RAM



# Registros de desplazamiento

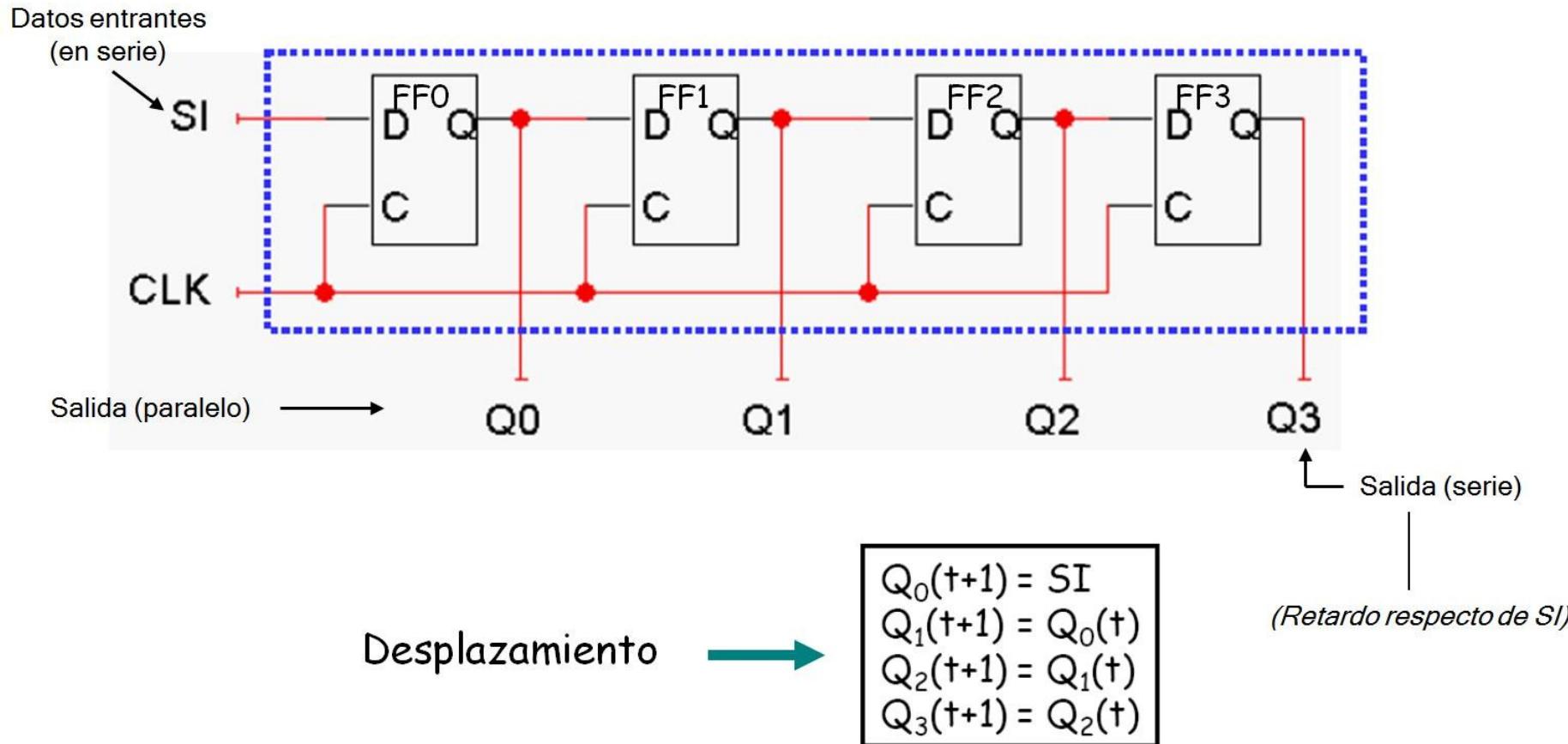
Carga de datos en serie

## Aplicaciones:

- ✓ Multiplicación y división por desplazamiento de bits
- ✓ Conversión serie/paralelo

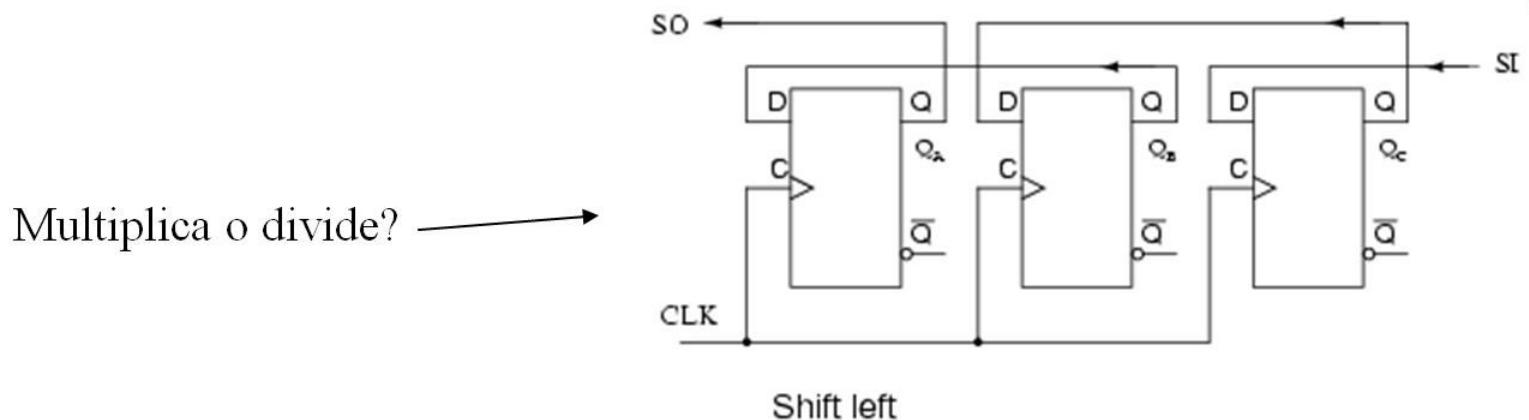
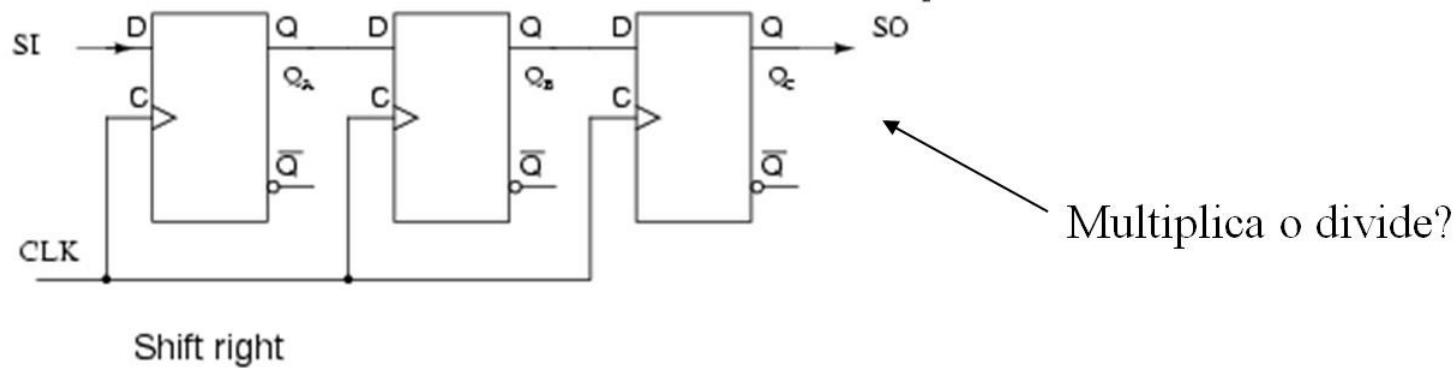
# Registros de desplazamiento

Carga de datos en serie



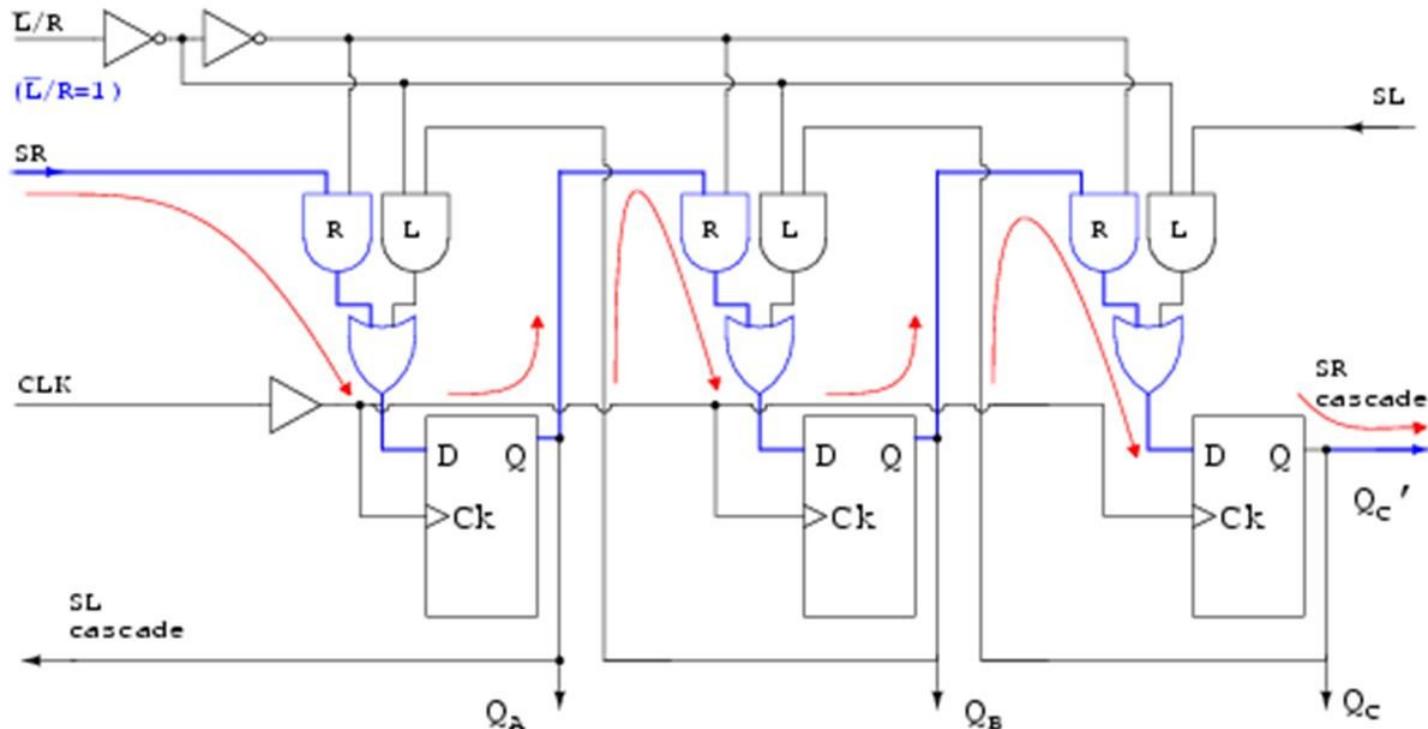
# Registros de desplazamiento

Dirección del desplazamiento fija (cableada)



# Registros de desplazamiento

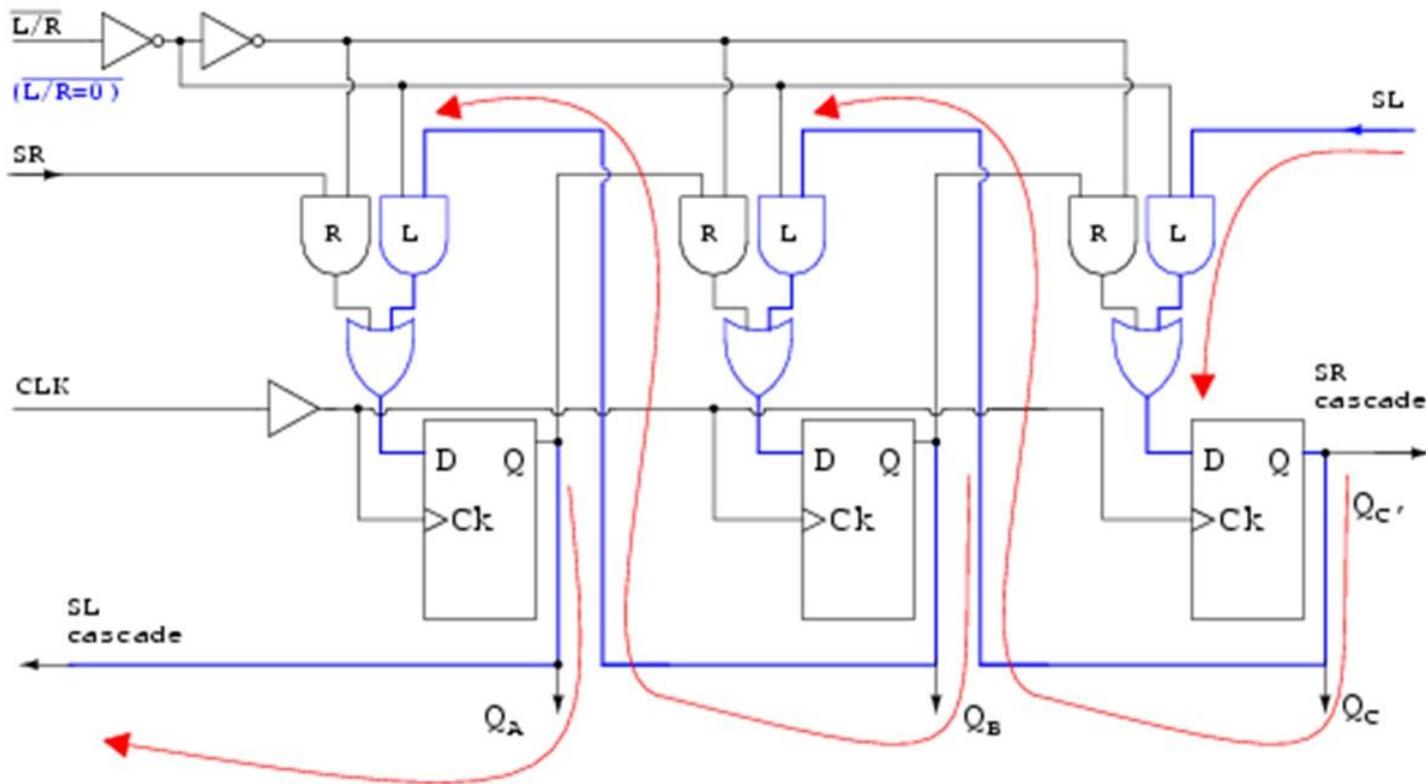
Control de la dirección del desplazamiento



Desplazamiento a derecha

# Registros de desplazamiento

Control de la dirección del desplazamiento



Desplazamiento a izquierda

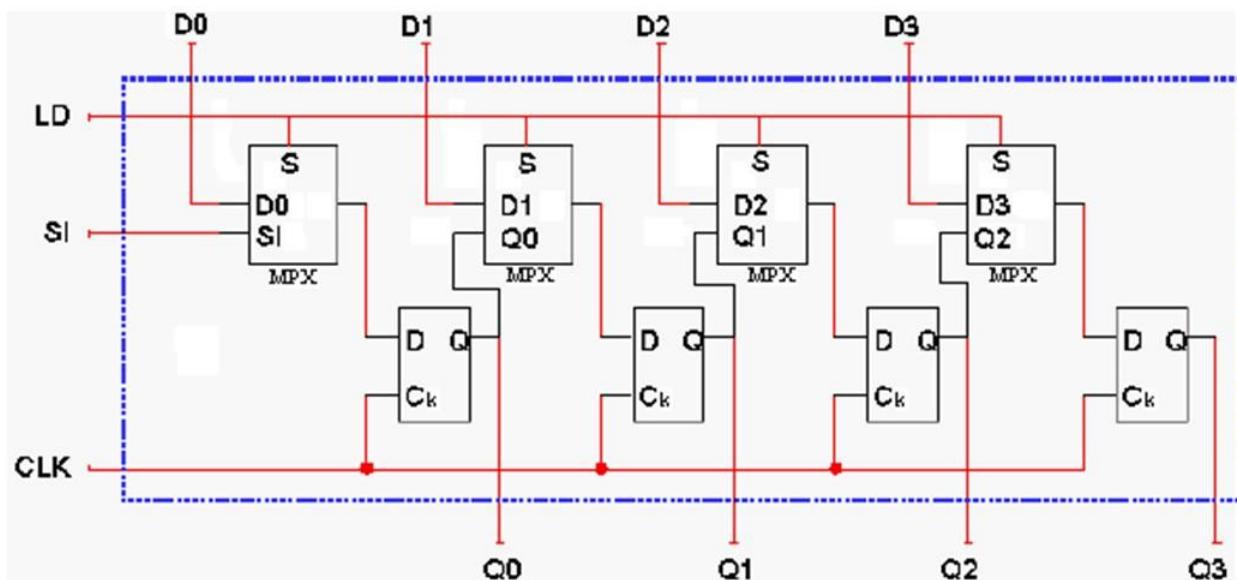
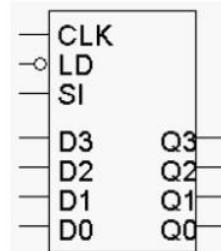
# Registros de desplazamiento

## Carga de datos en paralelo

SI: bit de entrada serie

LD= 0 carga en paralelo

LD=1 desplaza



# Aplicación de registros de desplazamiento a la conversión serie/paralelo

- **Las computadoras trabajan con palabras (conjunto de bits)**
  - Números: Enteros, punto flotante (32 bits, 64 bits)
  - Caracteres alfanuméricos ASCII (8 bits)

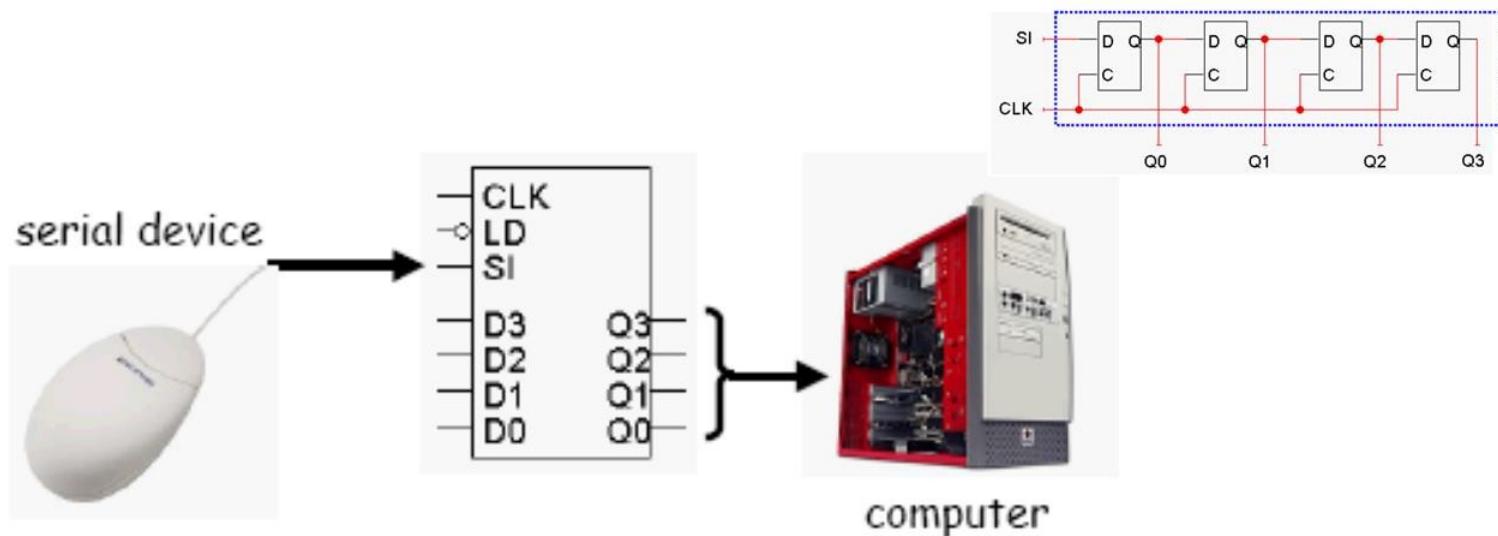
*Sin embargo...*

- **La transmisión muchas veces debe hacerse de a un bit por vez**
  - Mouse y teclado
  - Impresoras
  - Puertos serie, USB, IEEE 1394 (FireWire/i.link)
  - Discos rígidos: Serial ATA

¿Porque razón hay una tendencia hacia las interfaces serie?

# Recepción de datos desde un dispositivo serie

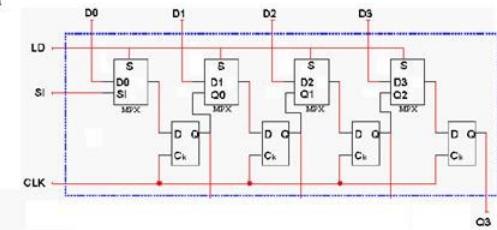
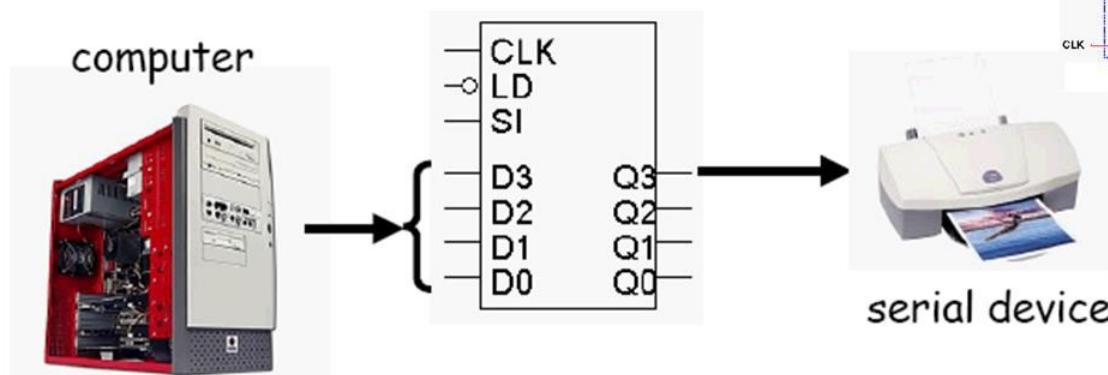
- Dispositivo serie conectado a la entrada SI
  - Salidas Q3-Q0 conectadas a la computadora
- ✓ El dispositivo serie transmite 1 bit por ciclo de reloj
- ✓ Luego de 4 ciclos de reloj una palabra de 4 bits completa esta disponible.
- ✓ En un solo paso la computadora lee Q3-Q0 a la salida del Reg. de Despl.



# Envío de datos a un dispositivo serie

- La CPU conectada a las entradas D del registro
- La salida SO (Q3) conectada al dispositivo serie

- ✓ En un solo ciclo de reloj la computadora almacena los 4 bits en el reg.
- ✓ Con c/u de los ciclos de reloj siguientes el dispositivo lee la salida serie
- ✓ Después de 4 ciclos la palabra de 4 bits fue transmitida



---

# Contadores

## Aplicaciones

- ✓ Contar cantidad de veces que algún evento ha ocurrido
- ✓ Medir tiempo
- ✓ Contar la cantidad de bits que fueron enviados o recibidos desde o hacia un dispositivo

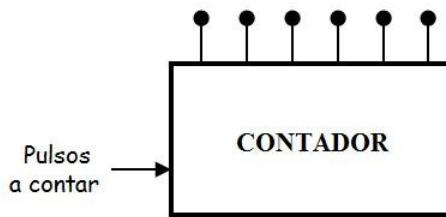
# Contadores

- Con cada ciclo de reloj la salida (n bits) del contador se incrementa en 1
- Vuelve al primer estado una vez agotada la capacidad de cuenta
- Al igual que en los registros las salidas Q son el estado del contador

cEntradas?

Tabla de estados

	Estado actual	Próximo estado
	0	1
	1	2
	2	3
	3	4
	4	5
	5	6
	6	7
	7	0



Para contar 10 pulsos:

Cuántos estados necesito?

Cuántos bits necesito?

Cuántos FlipFlops necesito?

# Descripción formal de un contador

- Tabla de estados
- Diagrama de estados
- Módulo y código de cuenta
- Diagrama de tiempos

**"Analizar"** un contador → *a partir del circuito determinar todo lo anterior*

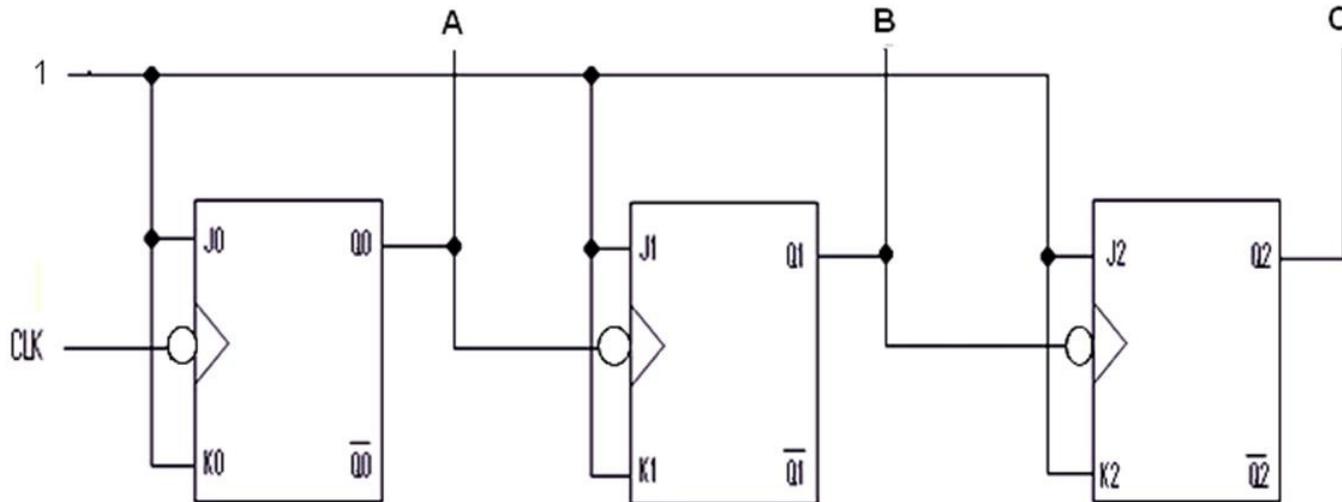
# Algunas definiciones

- ➊ módulo
- ➋ código
- ➌ estados prohibidos
- ➍ secuencia cerrada
- ➎ secuencia prohibida

# Formas de clasificación

- ⇒ Sincrónicos y asincrónicos
- ⇒ Módulo  $2^n$  y de módulo distinto de  $2^n$  ( $n$  = cantidad de FF)
- ⇒ Sincrónicos de transporte serie, transporte paralelo o mixto
- ⇒ De cuenta ascendente y de cuenta descendente
- ⇒ Contadores en anillo y de cadena abierta

# Contador asincrónico de 3 bits

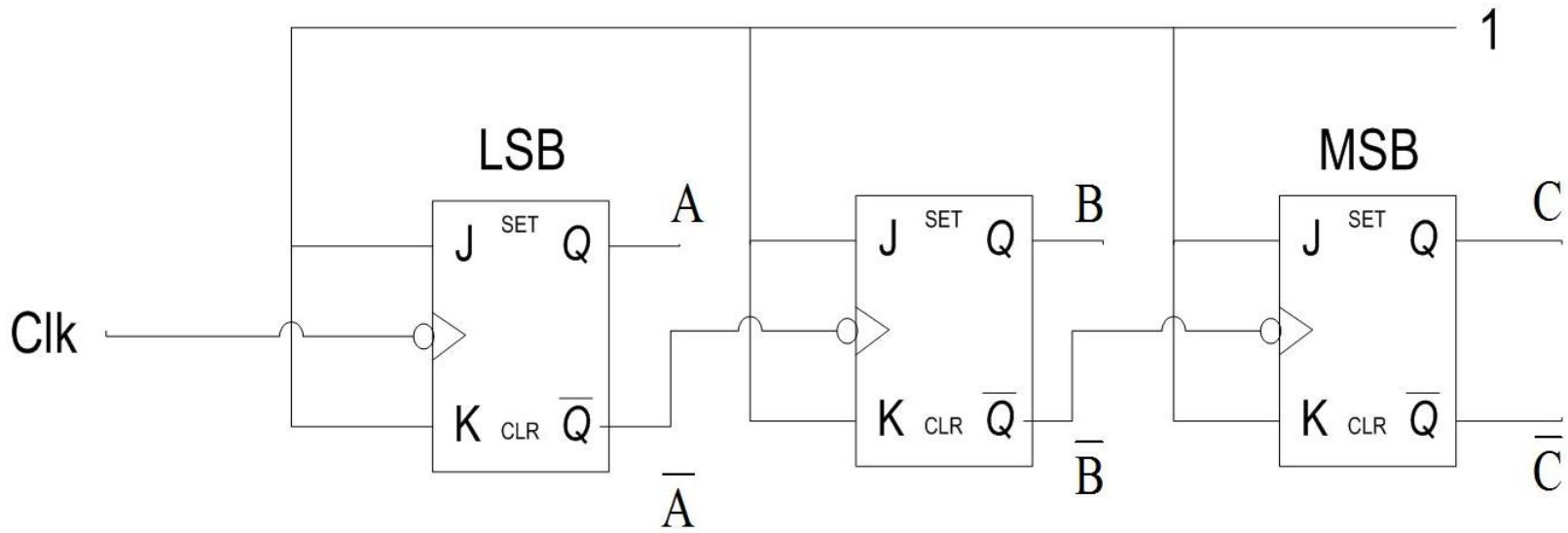


... Analizar este contador

... Estudiar como convertirlo en un contador de módulo 6

... Como sería un contador de décadas (módulo 10) asincrónico?

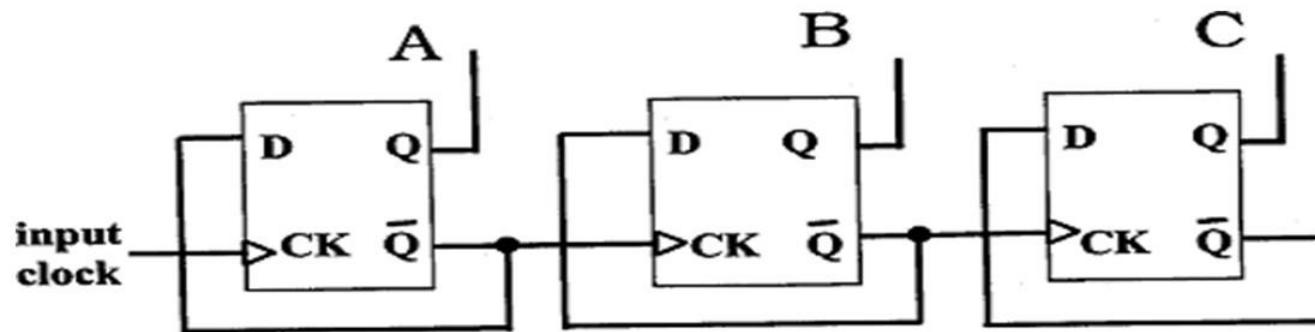
# Otro contador asincrónico de 3 bits



... Determinar tabla de estados, diagr. de estados, módulo y código de cuenta

# Otro contador asincrónico de 3 bits

Ahora con FF tipo D



... Determinar tabla de estados, diagr. de estados, módulo y código de cuenta

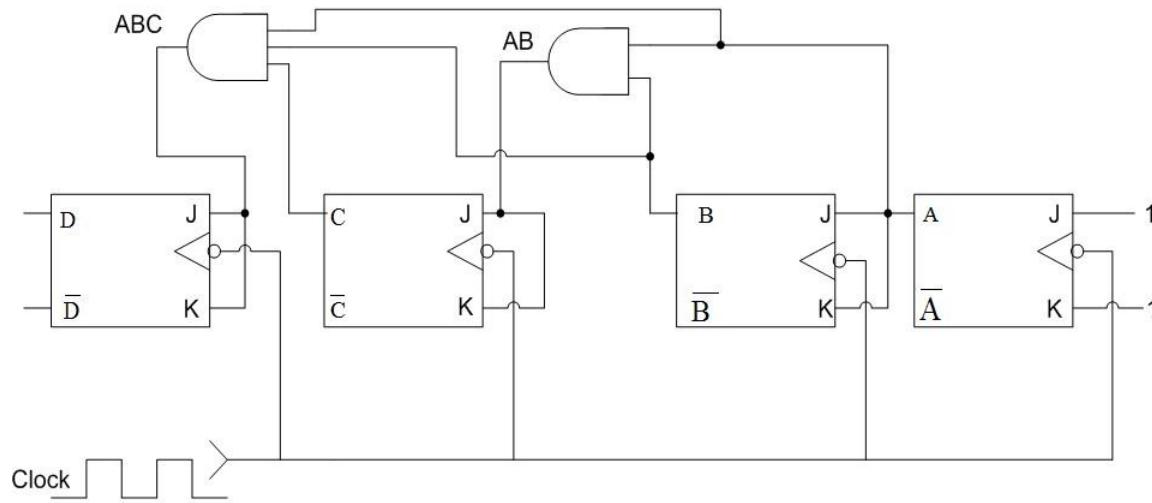
... Qué tipos de FF pueden utilizarse para construir contadores?

# Contadores sincrónicos

- ❖ *Contadores sincrónicos Vs. Contadores Asincrónicos*
- ❖ *Estructura general*

# Contadores sincrónicos

## Ejemplo



... Analizar

... Estimar frecuencia máxima con retardos dados: FFs 22 nseg, ANDs: 8 nseg

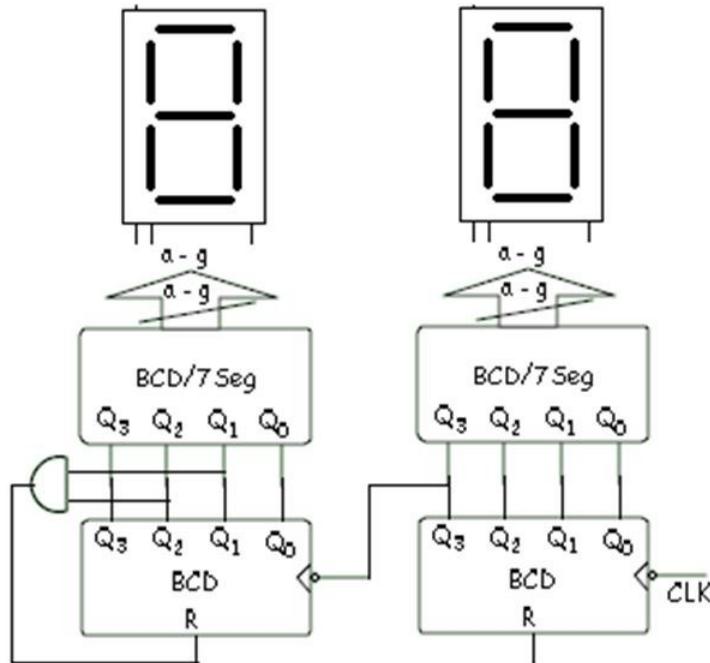
... Comparar con la velocidad esperable de un contador asincrónico similar

# Contadores sincrónicos

*Comparar “contadores sincrónicos” con “contadores Asincrónicos”*

- ❖ *velocidad?*
- ❖ *complejidad?*
- ❖ *versatilidad?*

# Ejemplo de aplicación: Segundero Digital



- *Dar las características requeridas para los contadores*
- *Cuál es la función de la compuerta AND?*
- *Cómo conectaría un módulo de minutos?*

# Diseño de contadores sincrónicos

- 1) Elijo el módulo y el código de cuenta (=> diagr. de estados)
- 2) Módulo => cantidad de FlipFlops necesarios
- 3) Elegir un tipo dado de FF
- 4) Obtener la lógica de compuertas en base al código de cuenta

El contador puede hacerse con cualquier tipo de FF (D, T, JK) pero la lógica de compuertas será diferente en cada caso

Para este tipo de diseño conviene representar el funcionamiento de los FF en base a su “tabla de transiciones” en vez de su tabla de estados.

# Diseño de contadores sincrónicos

TABLA DE TRANSICIONES  
DEL FLIP-FLOP T

$Q_n$	$Q^{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

TABLA DE TRANSICIONES  
DEL FLIP-FLOP JK

$Q_n$	$Q^{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Variables independientes

Variable dependientes

Variables independientes

Variables dependientes

# Introducción al diseño de circuitos secuenciales sincrónicos por el método de las transiciones

Ejemplo:

## Contador módulo 4 con entrada de habilitación

### Especificaciones

- Entrada de habilitación: X

$X=1$  responde a los pulsos de entrada  
 $X=0$  no responde, permanece en el  
mismo estado

- Módulo: 4

- Código de cuenta: binario

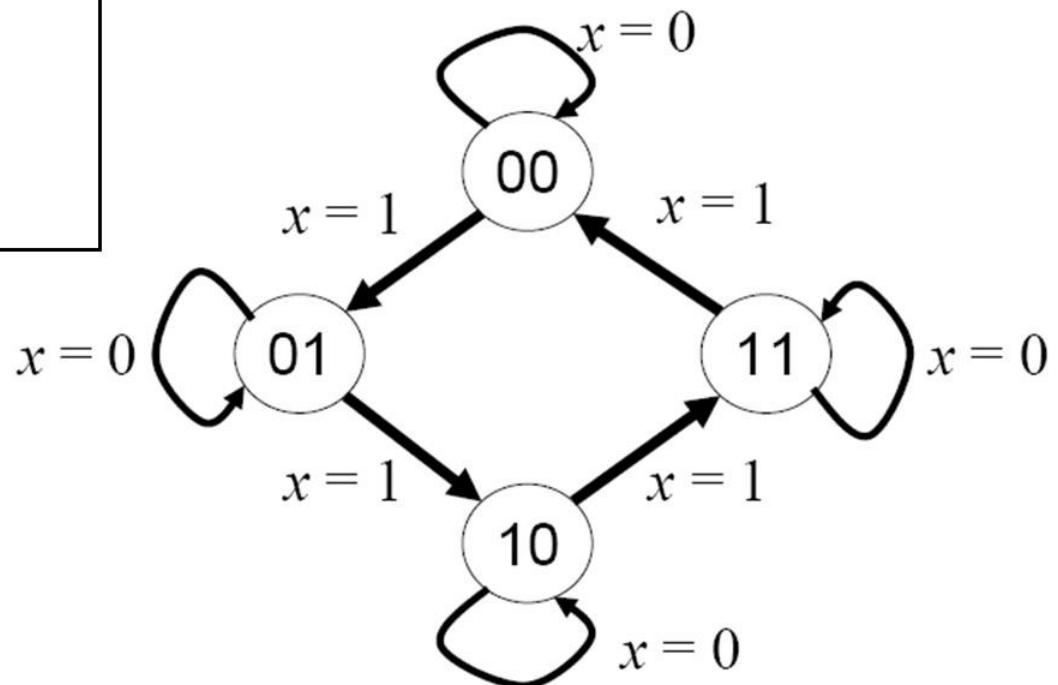


Diagrama de estados

Introducción al diseño de circuitos secuenciales sincrónicos  
por el método de las transiciones

*Ejemplo:*

## Contador módulo 4 con entrada de habilitación

TABLA DE TRANSICIONES DEL CONTADOR

Habil.	Estado actual		Estado siguiente		Entradas FF	
X	$Q_0$	$Q_1$	$Q_0$	$Q_1$	$T_0$	$T_1$
1	0	0	0	1		
1	0	1	1	0		
1	1	0	1	1		
1	1	1	0	0		
0	0	0	0	0		
0	0	1	0	1		
0	1	0	1	0		
0	1	1	1	1		

Introducción al diseño de circuitos secuenciales sincrónicos  
por el método de las transiciones

*Ejemplo:*

# Contador módulo 4 con entrada de habilitación

TABLA DE TRANSICIONES DEL CONTADOR

Habil.	Estado actual		Estado siguiente		Entradas FF	
X	$Q_0$	$Q_1$	$Q_0$	$Q_1$	$T_0$	$T_1$
1	0	0	0	1		
1	0	1	1	0		
1	1	0	1	1		
1	1	1	0	0		
0	0	0	0	0		
0	0	1	0	1		
0	1	0	1	0		
0	1	1	1	1		

n	$n+1$	T
0	0	0
0	1	1
1	0	1
1	1	0

T. de transiciones  
del FF-T

Introducción al diseño de circuitos secuenciales sincrónicos  
por el método de las transiciones

*Ejemplo:*

# Contador módulo 4 con entrada de habilitación

TABLA DE TRANSICIONES DEL CONTADOR

Habil.	Estado actual		Estado siguiente		Entradas FF	
X	$Q_0$	$Q_1$	$Q_0$	$Q_1$	$T_0$	$T_1$
1	0	0	0	1	0	1
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	0	0	1	1
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	0	0

n	$n+1$	T
0	0	0
0	1	1
1	0	1
1	1	0

T. de transiciones  
del FF-T

# Introducción al diseño de circuitos secuenciales sincrónicos por el método de las transiciones

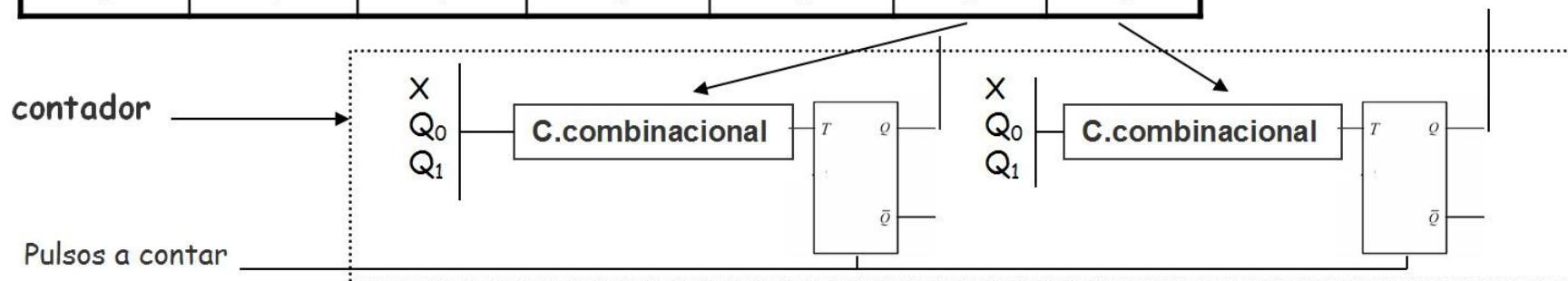
Ejemplo:

## Contador módulo 4 con entrada de habilitación

Variables independientes

Variables dependientes

Habil.	Estado actual		Estado siguiente		Entradas FF	
X	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>1</sub>	T <sub>0</sub>	T <sub>1</sub>
1	0	0	0	1	0	1
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	0	0	1	1
0	0	0	0	0	0	0
0	0	1	0	1	0	0
0	1	0	1	0	0	0
0	1	1	1	1	0	0



# Método de las transiciones

1. Formalizar el funcionamiento esperado por medio de un diagrama de estado.  
Identificar las variables entrada y de salida
2. El número de flip-flops necesarios para el circuito es el número de bits que tienen los estados.
3. Se realiza la tabla de estados y se agregan columnas con los 0's y 1's necesarios en las entradas de cada FF para que esa transición se produzca (tomada de la tabla de transiciones del FF).
4. Se diseña el circuito combinacional para cada entrada de cada flip-flop usando mapas de Karnaugh.
5. Se implementa el circuito en base a las ecuaciones de entrada a los FF

Introducción al diseño de circuitos secuenciales sincrónicos  
por el método de las transiciones

# Método de las transiciones

Aplicar este método a los siguientes diseños:

1. Contador sincrónico de décadas que cuente en binario
2. Idem con una entrada que permita elegir si la cuenta es ascendente o descendente.
3. Contador de 3 bits cuya cuenta es representada por un 1 que se desplaza “en anillo”

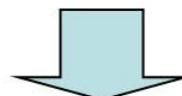
Introducción al diseño de circuitos secuenciales sincrónicos  
por el método de las transiciones

# Método de las transiciones

Aplicar este método a los siguientes diseños:

1. Contador sincrónico de décadas que cuente en binario
2. Idem con una entrada que permita elegir si la cuenta es ascendente o descendente.
3. Contador de 3 bits cuya cuenta es representada por un 1 que se desplaza “en anillo”

**Este método de diseño no está limitado a contadores.**



**Un circuito secuencial cualquiera, que podría incluir otras varias entradas y salidas, puede diseñarse en forma similar.**

