
Tema 4. Subsistemas secuenciales

Electrónica Digital
Grado en Ingeniería Electrónica Industrial
Escuela Politécnica Superior
Universidad de Sevilla
Curso 2019/2020



Contenidos

- Introducción
- Contadores
- Registros
- Subsistemas secuenciales programables



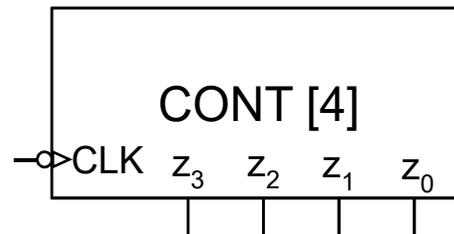
Introducción

- Sistemas secuenciales que aparecen con frecuencia, lo que justifica que los fabricantes los encapsulen.
- Formado por n biestables (n bits) que operan de forma conjunta para la realización de una o más tareas determinadas.
- Su operación se interpreta en base al dato de n bits que almacenan y no en base a cada bit por separado.
- Su funcionalidad es lo bastante general para encontrar aplicación en una diversidad de problemas de diseño de circuitos secuenciales.
- Contadores: proporcionan una secuencia de números consecutivos (cuenta).
- Registros: almacenan un dato para su uso posterior.



Contador. Introducción

- Concepto: Circuito secuencial que cuenta pulsos de reloj de forma cíclica. En su versión más básica no tiene entradas y n salidas que indican el estado de cuenta.
- El número de estados distintos p se denomina *MÓDULO*.
- Símbolo:



Contador. Clasificación

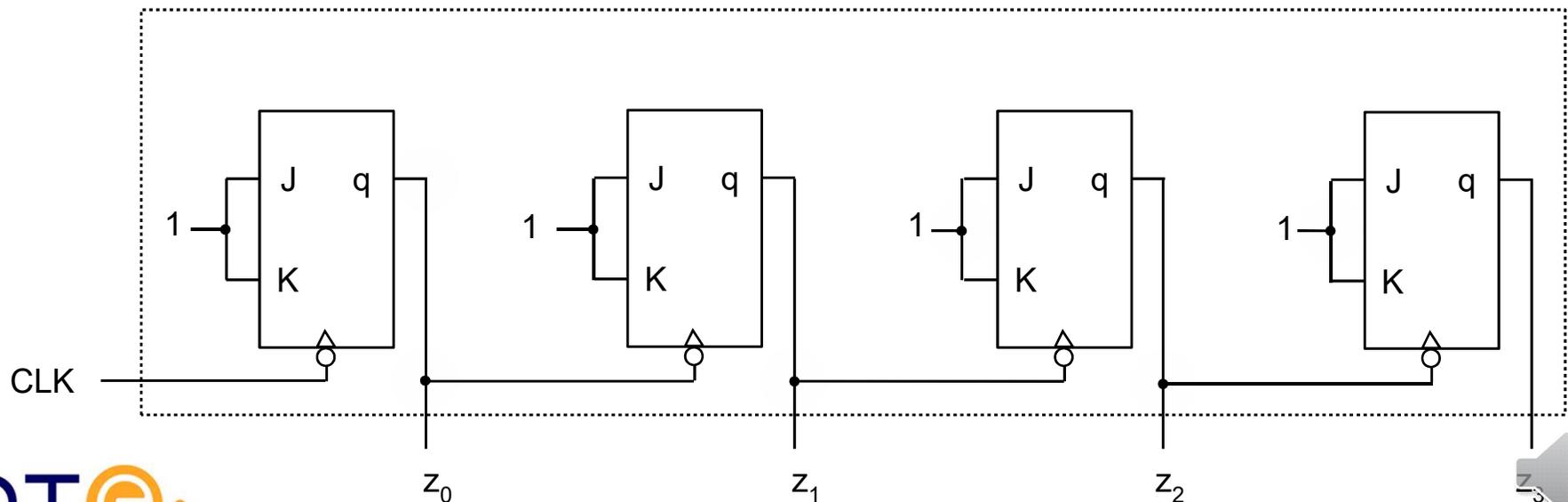
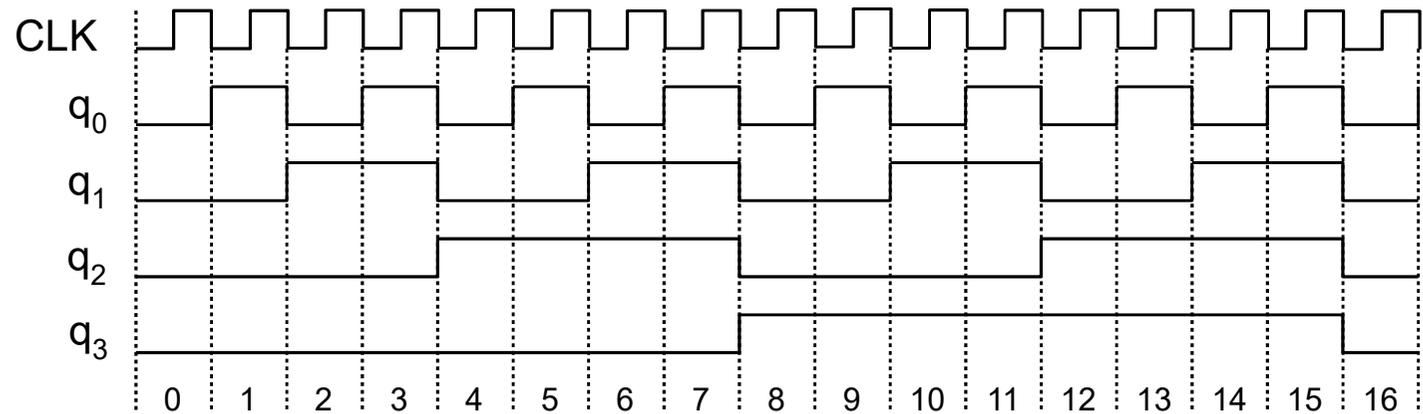
Clasificación según diversos criterios (en cursiva la opción por defecto):

- ❑ Flanco de disparo del reloj: *Ascendente* / Descendente.
- ❑ Señal de reloj de biestables: *Síncronos* / Asíncronos.
- ❑ Código de cuenta: *Natural* / BCD / Gray / Johnson / En anillo.
- ❑ Sentido de cuenta: *Ascendente* / Descendente / Reversible.
- ❑ Relación módulo/bits salida: *Completo* ($p=2^n$) / Incompleto ($p<2^n$)



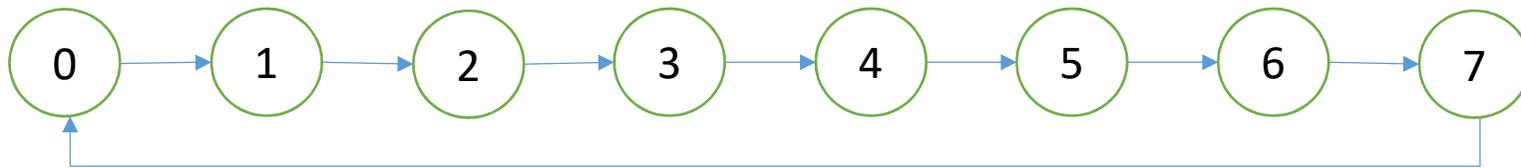
Contador asíncrono (*ripple*)

Los biestables NO comparten una única señal de reloj. Sencillos y gran densidad de integración, pero cada etapa se retrasa respecto a la anterior.

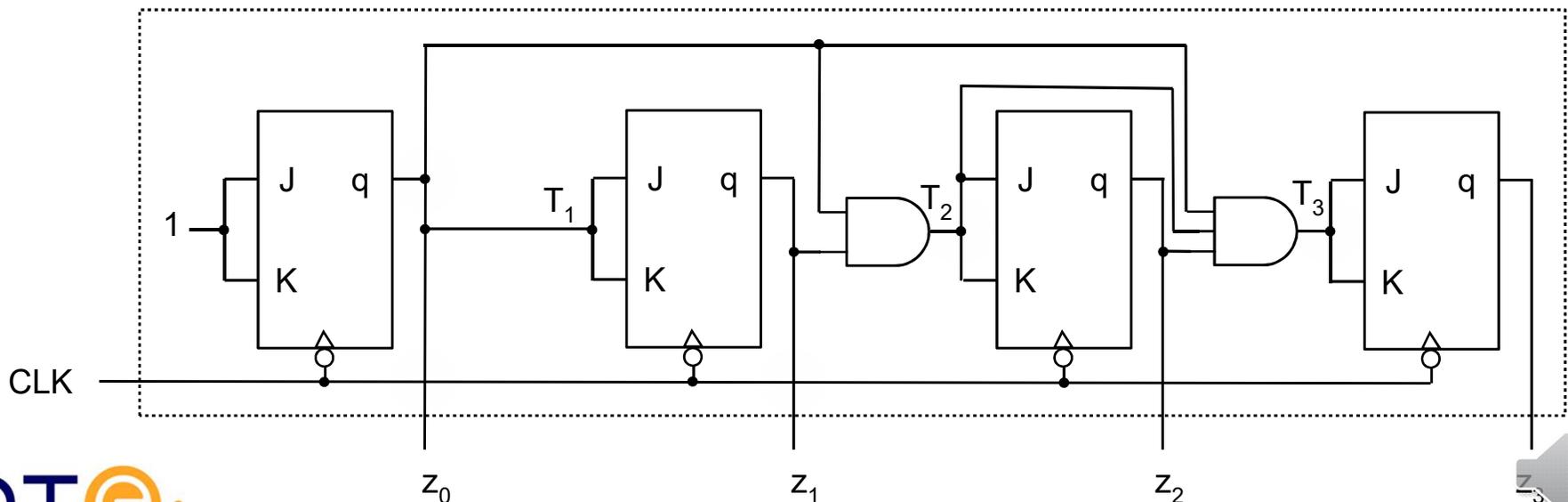


Contador síncrono (1)

Los biestables comparten una única señal de reloj. Más complejos, pero más rápidos. Las salidas son síncronas y se actualizan simultáneamente. Se diseña como una máquina de Moore.



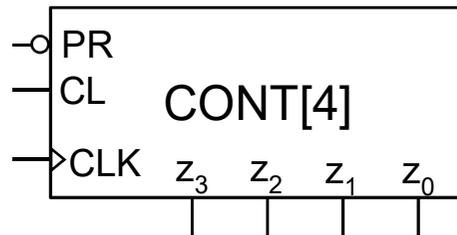
$$T_0 = 1 \quad T_1 = q_0 \quad T_2 = q_1 q_0 \quad T_3 = q_2 q_1 q_0 \quad T_i = \prod_{k=0}^{i-1} q_k$$



Contador síncr. Entradas/salidas especiales (1)

Entradas de puesta a cero y puesta a uno:

- Tipo: asíncrona / síncrona
- Actividad: en alta / en baja



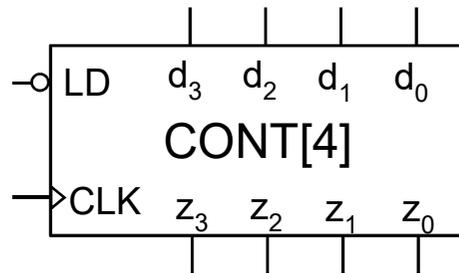
PR# CL	q ←	z =	Tipo
0 1	Prohibido		asínc.
0 0	1..1	[CONT]	asínc.
1 1	0		asínc.
1 0	$q+1 _{\text{mod } 16}$		sínc.



Contador síncr. Entradas/salidas especiales (2)

Entrada de carga en paralelo:

- Tipo: asíncrona / síncrona
- Actividad: en alta / en baja



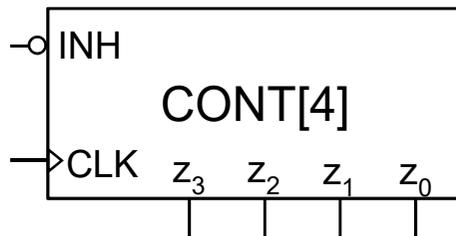
LD#	$q \leftarrow$	$z =$	Tipo
0	$d_{3:0}$	[CONT]	asínc.
1	$q+1 _{\text{mod } 16}$		sínc.



Contador síncr. Entradas/salidas especiales (3)

Entrada de inhibición de cuenta:

- Actividad: en alta / en baja



INH#	$q \leftarrow$	$z =$
0	q	[CONT]
1	$q+1 _{\text{mod } 16}$	



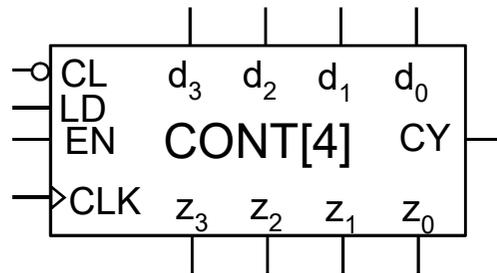
Contador síncr. Entradas/salidas especiales (4)

Salida de carry (contador ascendente)/borrow (descendente):

- Se activa en el último estado de cuenta.

$$CY = q_3q_2q_1q_0$$

$$BW = \overline{q_3 + q_2 + q_1 + q_0}$$

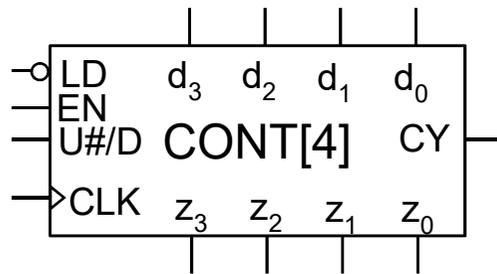


CL#	LD	EN	q ←	z =	CY =
0	-	-	0	[CONT]	1 sii [CONT=1..1]
1	1	-	d _{3:0}		
1	0	1	$q+1 _{\text{mod } 16}$		
1	0	0	q		



Contador síncr. Entradas/salidas especiales (5)

Contador reversible. Permite cambiar la dirección de cuenta. La salida de final de cuenta es carry (modo ascendente) o borrow (modo descendente):

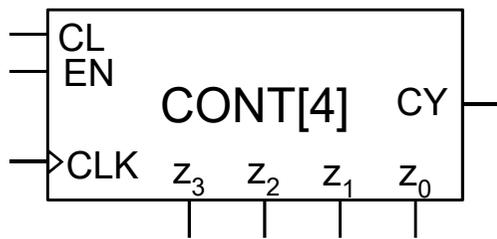


LD#	EN	U#/D	$q \leftarrow$	$z =$	CY =
0	-	-	$d_{3:0}$	[CONT]	1 sii [CONT=1..1] y U#/D=0 o [CONT=0] y U#/D=1
1	0	-	Q		
1	1	0	$q+1 _{\text{mod } 16}$		
1	1	1	$q-1 _{\text{mod } 16}$		

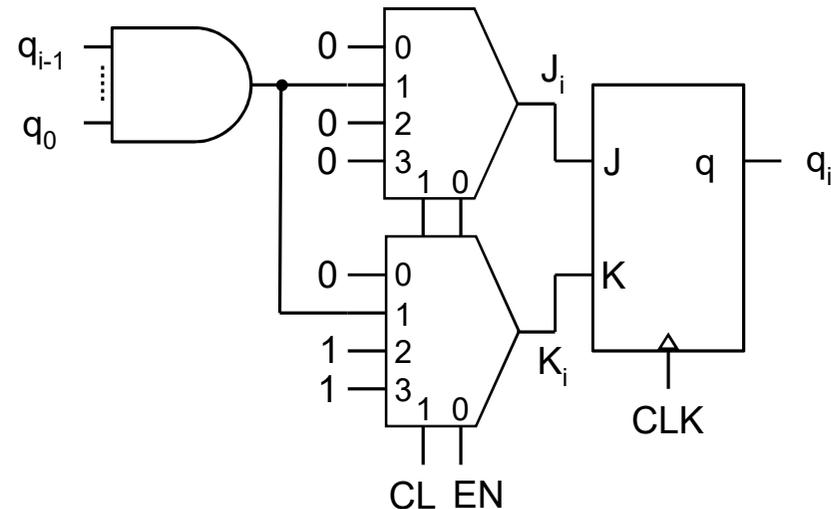
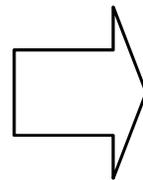
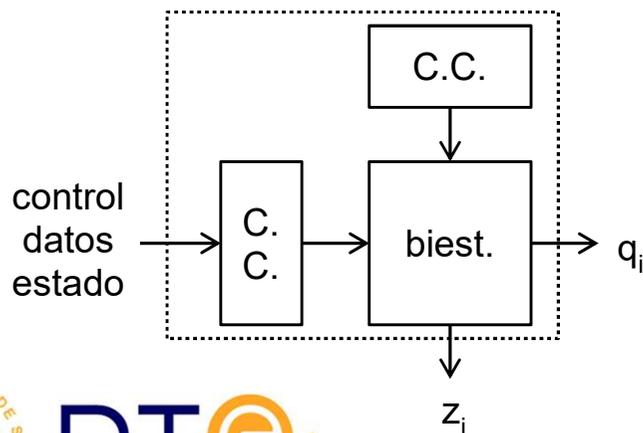


Contador síncr. Diseño modular

El diseño de un contador con varias entradas de control no es viable con las técnicas habituales de diseño. Por ello se usa un sistema modular: se calculan las entradas de excitación de los biestables para cada operación por separado y después se usa un MUX para seleccionar la adecuada según las señales de control activas.

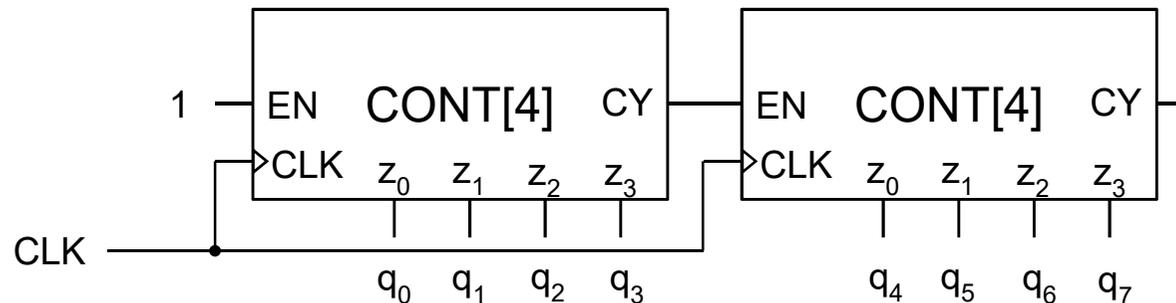


CL, EN	Operación	Et. típica	Et. 0
1 -	$q \leftarrow 0$	$J_i=0, K_i=1$	$J_0=0, K_0=1$
0 1	$q \leftarrow q+1 \bmod 16$	$J_i=K_i=q_{i-1} \dots q_0$	$J_0=K_0=1$
0 0	$q \leftarrow q$	$J_i=K_i=0$	$J_0=K_0=0$



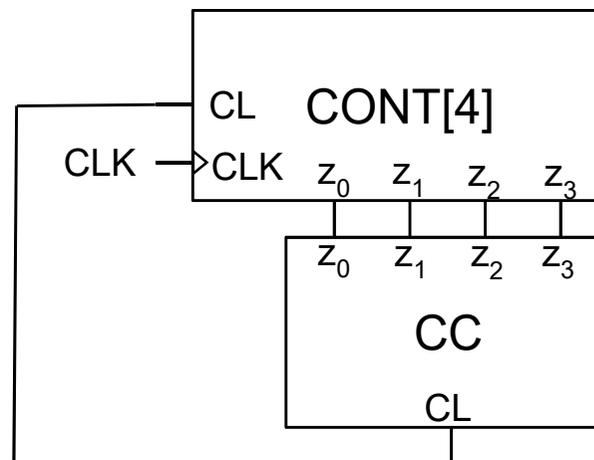
Unión de contadores

- ❑ Es posible combinar contadores para obtener un nuevo contador con mayor número de estados de cuenta que los originales.
- ❑ Combinando dos contadores de módulos p y q se puede conseguir un nuevo contador módulo $p \cdot q$ (ej: módulos 2 y 4, resultante, 8).
- ❑ La combinación de los contadores se puede hacer de forma más simple (con menos componentes adicionales) si los contadores poseen entradas de control y estado adecuadas (habilitación y fin de cuenta).
- ❑ Versiones asíncronas (distintos relojes) y síncronas.



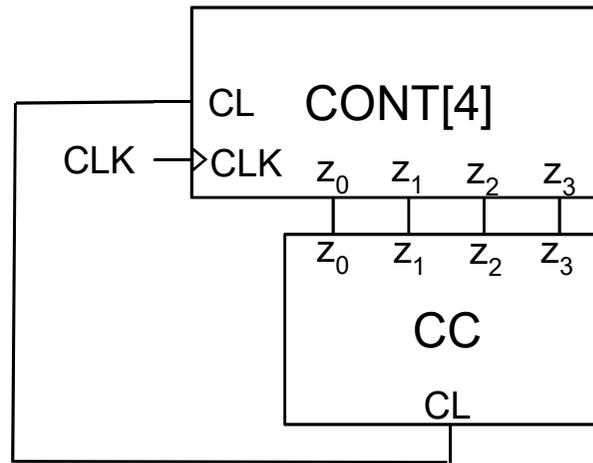
Contador incompleto

- Se pueden diseñar desde cero, pero es más fácil obtenerlos a partir de uno completo. En ese caso, se requiere de una entrada de puesta a cero o carga en paralelo.
- El circuito combinacional detecta el estado de cuenta final y activa la entrada de control para llevar al contador al estado inicial.
- Si la entrada de control es asíncrona, se debe detectar el estado siguiente al final. Esto supone que dicho estado aparece momentáneamente a la salida.



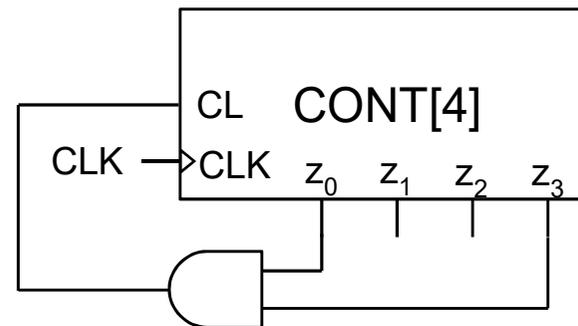
Contador BCD

- Detectar el estado 9 (entrada es síncrona) o 10 (entrada asíncrona).



	$z_3 z_2$			
$z_1 z_0$	00	01	11	10
00	0	0	-	0
01	0	0	-	1
11	0	0	-	-
10	0	0	-	-

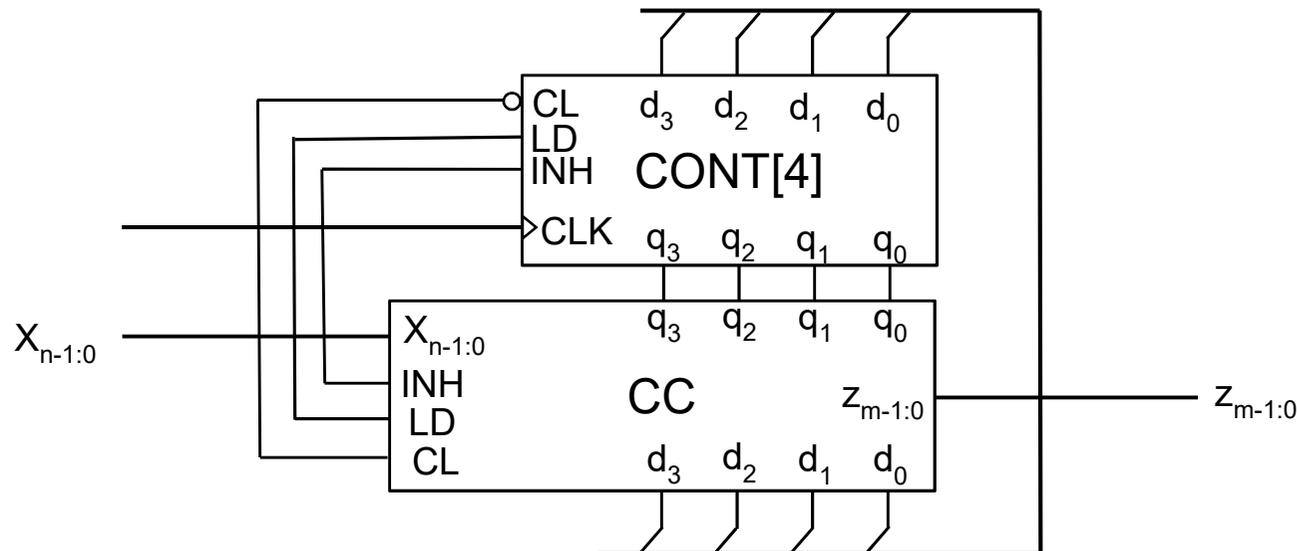
$$CL = z_3 z_0$$



Diseño de CSS con contadores (1)

Estrategia de diseño:

- Adaptar el funcionamiento de un contador a una FSM con el uso de un circuito combinacional CC.
- Entradas de CC: entradas de la FSM ($x_{n-1:0}$) y salidas del contador.
- Salidas de CC: salidas de la FSM ($z_{m-1:0}$) y entradas del contador.



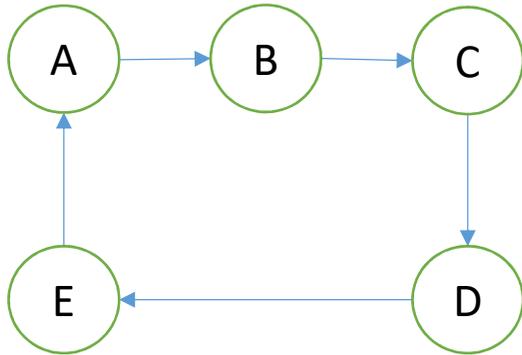
Diseño de CSS con contadores (2)

Metodología de diseño:

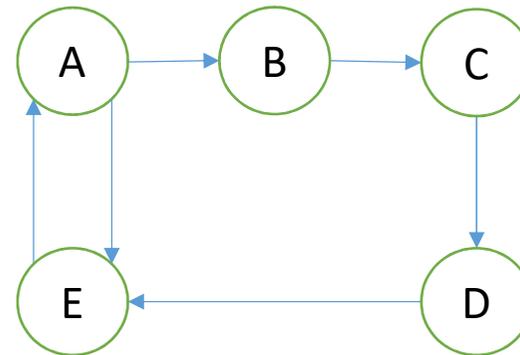
- ❑ Obtención del diagrama de estados del CSS (caso particular: contador incompleto visto).
- ❑ Determinar el contador (módulo y operaciones). La carga en paralelo permite realizar cualquier diseño, pero supone un gran coste por el número de señales (control y datos) y complejidad.
- ❑ Asignar estados de cuenta a cada estados de la FSM. Escoger como estado 0 aquel que reciba más transiciones. Procurar que el número de saltos entre estados que no tengan asignados códigos binarios consecutivos sea mínimo.
- ❑ Asignar operaciones del contador a cada arista. Minimizar (eliminar si es posible) el uso de la carga en paralelo dado su coste.



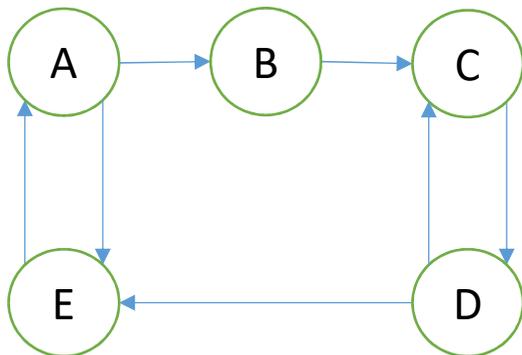
Diseño de CSS con contadores (3)



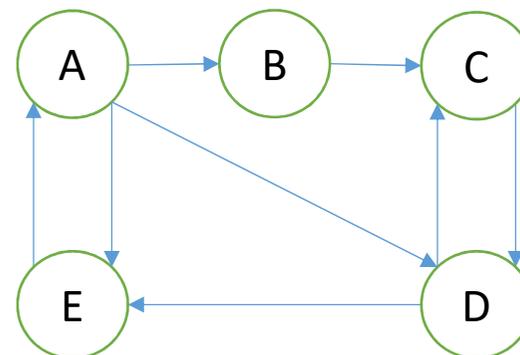
Puede hacerse con contador con puesta a cero



Contador con puesta a cero. Hacer $E=0$, $A=1, \dots$



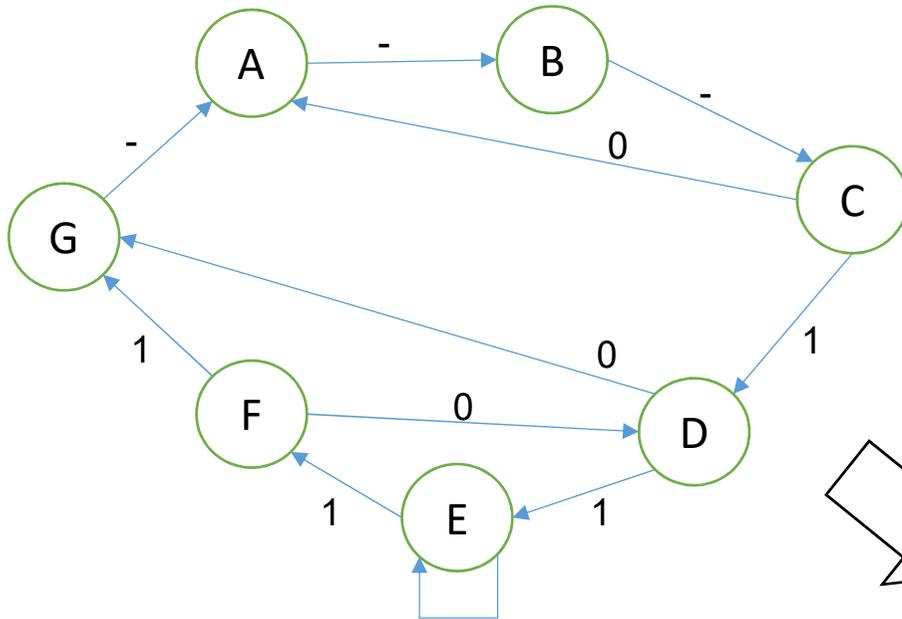
Contador reversible con puesta a cero



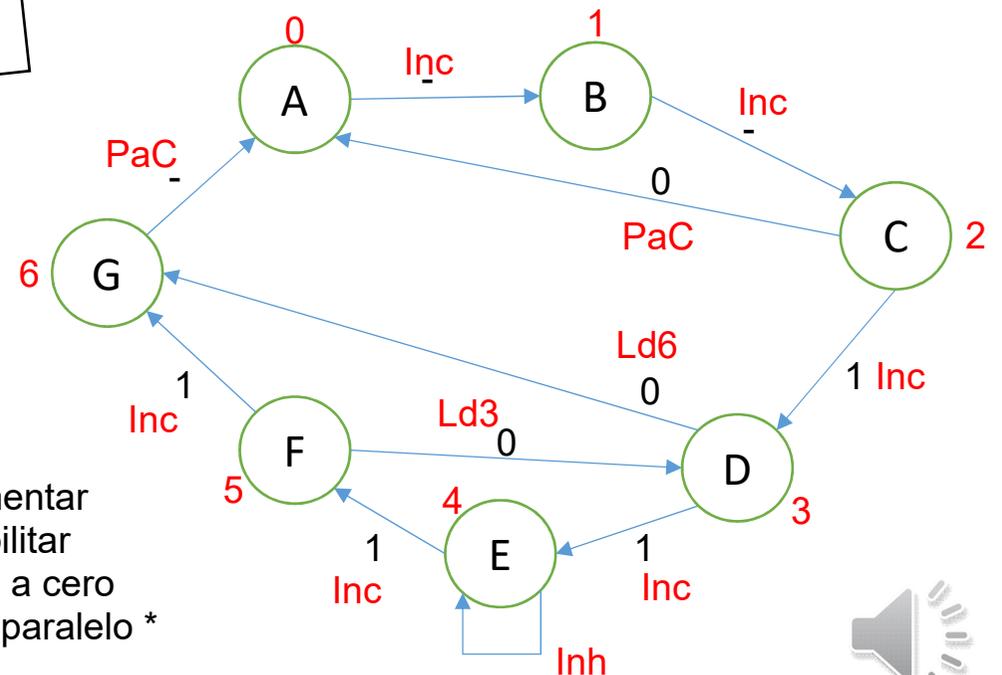
Contador con puesta a cero y carga en paralelo. Valorar si reversible



Diseño de CSS con contadores. Ejemplo (1)



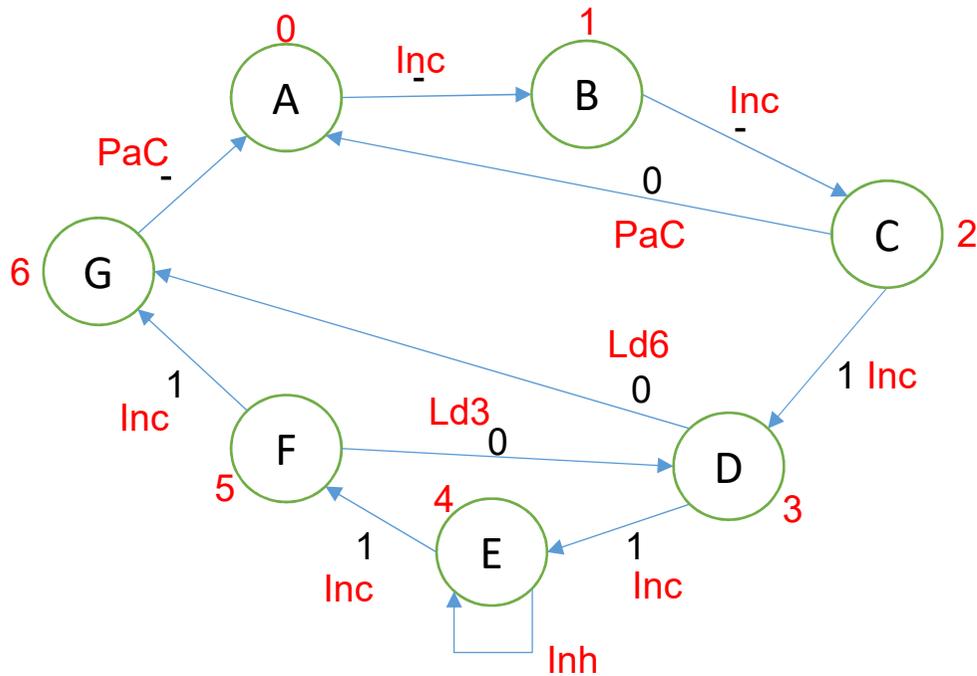
Contador módulo 8 con puesta a cero, inhabilitación y carga en paralelo



Inc: Incrementar
 Inh: Inhabilitar
 PaC: Puesta a cero
 Ld*: Cargar en paralelo *



Diseño de CSS con contadores. Ejemplo (2)



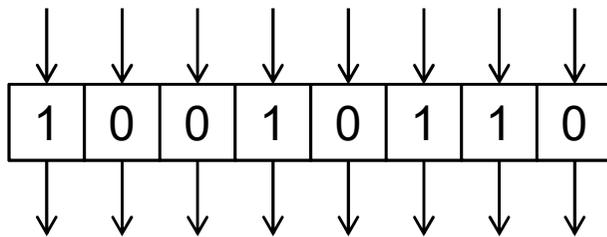
S	$q_2q_1q_0x$	CL LD EN $d_2d_1d_0$
A	0000	0 0 1 - - -
	0001	0 0 1 - - -
B	0010	0 0 1 - - -
	0011	0 0 1 - - -
C	0100	1 - - - - -
	0101	0 0 1 - - -
D	0110	0 1 - 110
	0111	0 0 1 - - -
E	1000	0 0 0 - - -
	1001	0 0 1 - - -
F	1010	0 1 - 011
	1011	0 0 0 - - -
G	1100	1 - - - - -
	1101	1 - - - - -
-	1110	- - - - - -
	1111	- - - - - -



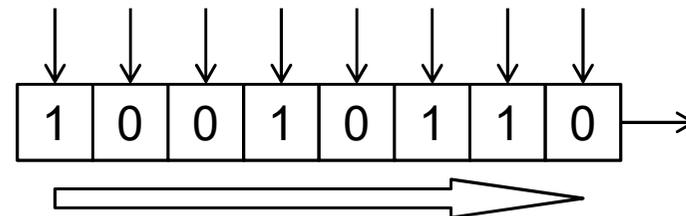
Registro. Introducción

- Concepto: Circuito secuencial que almacena una palabra de n bits.
- Se clasifican según la forma en que entra y sale la información, paralelo o serie, dando lugar a 4 tipos de registro:

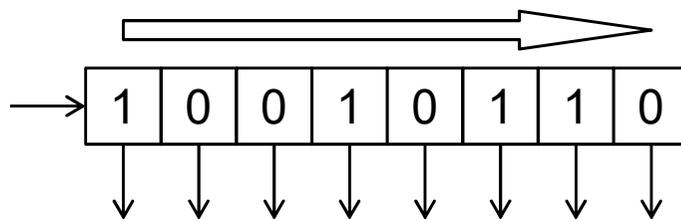
paralelo/paralelo



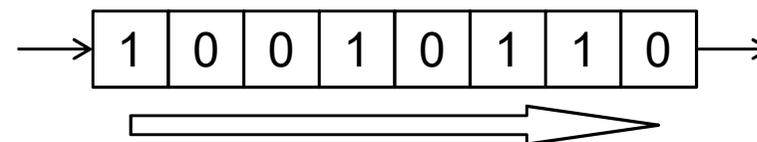
paralelo/serie



serie/paralelo



serie/serie

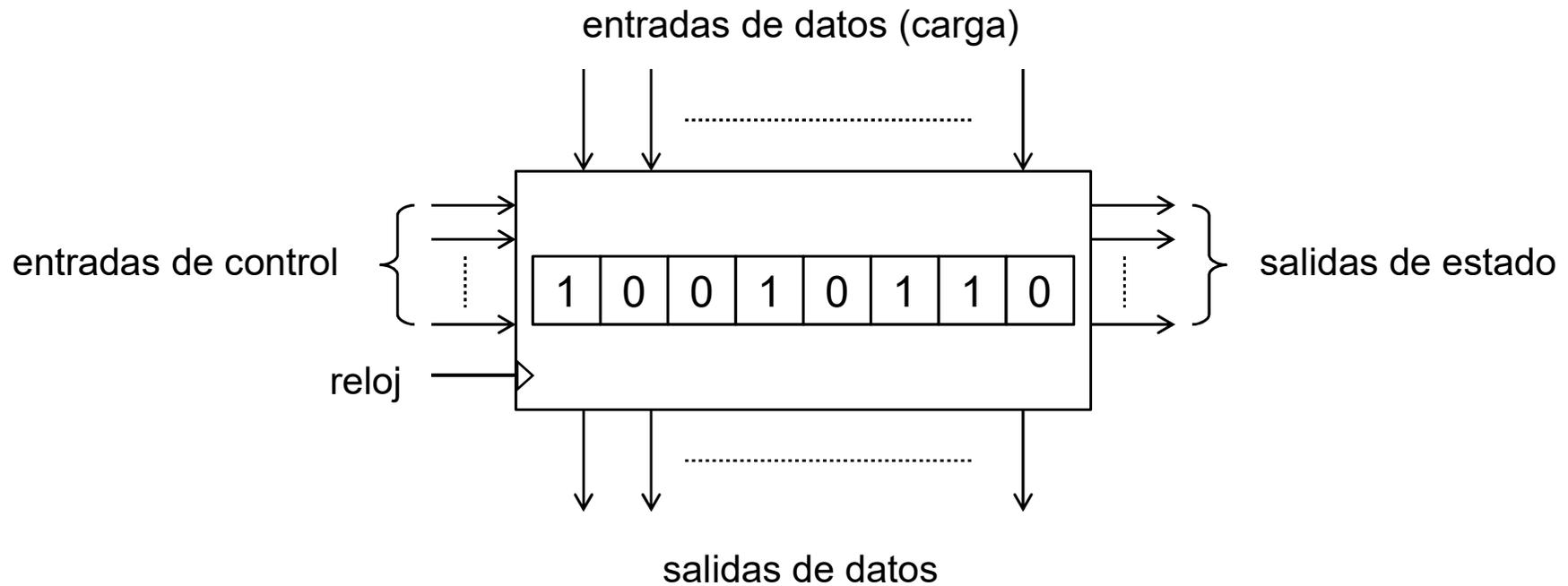


Registro. Clasificación

- ❑ Entrada en paralelo: Todos los bits pueden cargarse a la vez (en el mismo ciclo de reloj) dado que se dispone de una línea de entrada para cada bit.
- ❑ Entrada serie: sólo hay un bit de entrada, por lo que se carga un bit en cada ciclo de reloj, desplazando el contenido interno del registro.
- ❑ Salida en paralelo: Hay una salida para cada bit interno. Todos los bits pueden ser leídos a la vez.
- ❑ Salida serie: Hay una única salida por la que sale el *msb/l**s**b*. Para acceder al resto de los bits es necesario desplazarlos internamente. Sólo se accede a un bit por ciclo de reloj.



Registro. Símbolo general

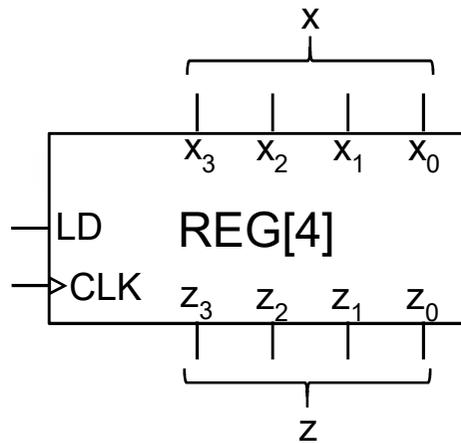


Registro. Entradas/salidas

- Señales de control:
 - Determinan la operación a realizar. Ej: puesta a cero (CL), inhibición (INH), carga en paralelo (LD).
 - Pueden ser activas en alta (CL) o en baja (CL#).
 - Síncronas (alteran el dato en el flanco activo de la señal de reloj) o asíncronas (alteran el dato de forma inmediata tras la activación de la señal de control).
- Entradas de datos: proporcionan el dato a cargar.
- Salidas de datos: permiten obtener (observar) el dato almacenado.
- Salidas de estado: indican información sobre el contenido del subsistema: si el contenido es cero, fin de estado de cuenta,...

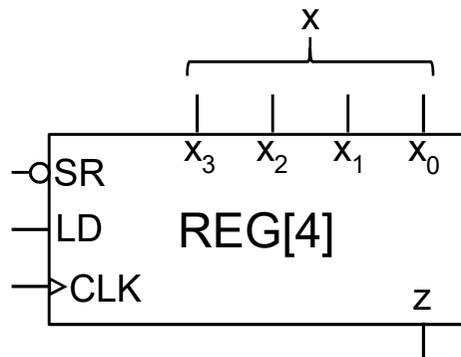


Registro paralelo/paralelo y paralelo/serie



Registro paralelo/paralelo

LD	$q \leftarrow$	$z =$
0	q	[REG]
1	x	



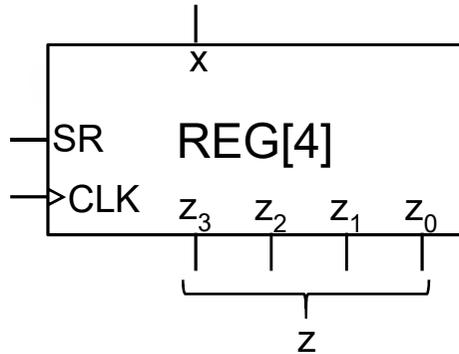
Registro paralelo/serie

LD SR#	$q \leftarrow$	$z =$
1 -	x	[REG ₀]
0 1	q	
0 0	SHR(REG, x ₃)	



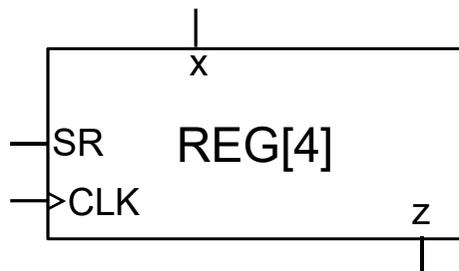
Registro serie/paralelo y serie/serie

Registro serie/paralelo



SR	$q \leftarrow$	z=
0	q	[REG]
1	SHR(REG, x)	

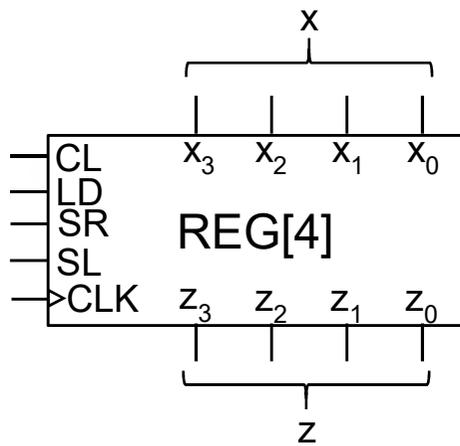
Registro serie/serie



SR	$q \leftarrow$	z=
0	q	[REG ₀]
1	SHR(REG, x)	



Registro universal



CL	LD	SR	SL	$q \leftarrow$	$z =$
1	-	-	-	0	[REG]
0	1	-	-	x	
0	0	1	-	SHR(REG, x_3)	
0	0	0	1	SHL(REG, x_0)	
0	0	0	0	q	



