



Tema 7:

Módulos secuenciales básicos

Fundamentos de computadores I

José Manuel Mendías Cuadros

Dpto. Arquitectura de Computadores y Automática

Universidad Complutense de Madrid





Contenidos

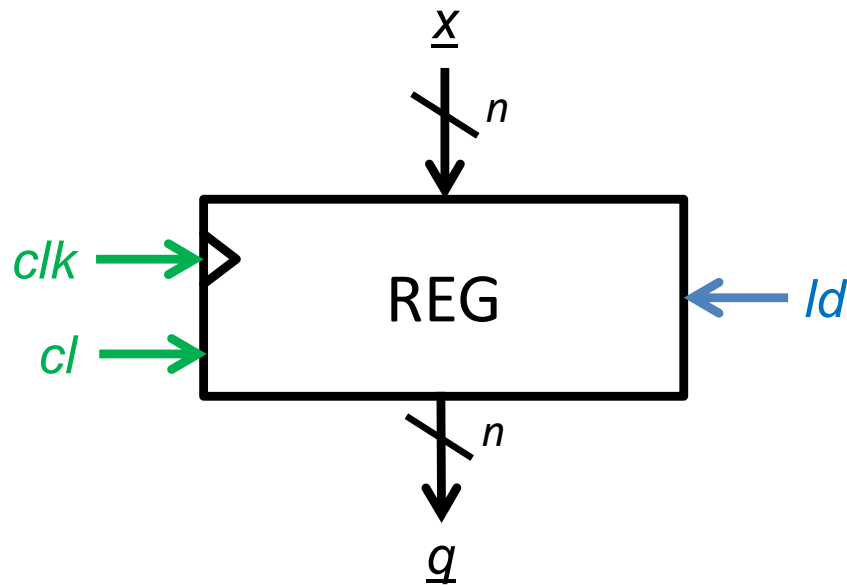
- ✓ Registro.
- ✓ Registro de desplazamiento.
- ✓ Contadores.
- ✓ Banco de registros.
- ✓ RAM (Random-Access Memory).

Transparencias basadas en los libros:

- R. Hermida, F. Sánchez y E. del Corral. *Fundamentos de computadores.*
- D. Gajsky. *Principios de diseño digital.*



Registro



Registro de n bits

- \underline{x} 1 entrada de datos de n bits
- \underline{q} 1 salida de datos de n bits
- ld 1 entrada de carga paralela
- cl 1 entrada de inicialización asíncrona
- clk 1 entrada de reloj

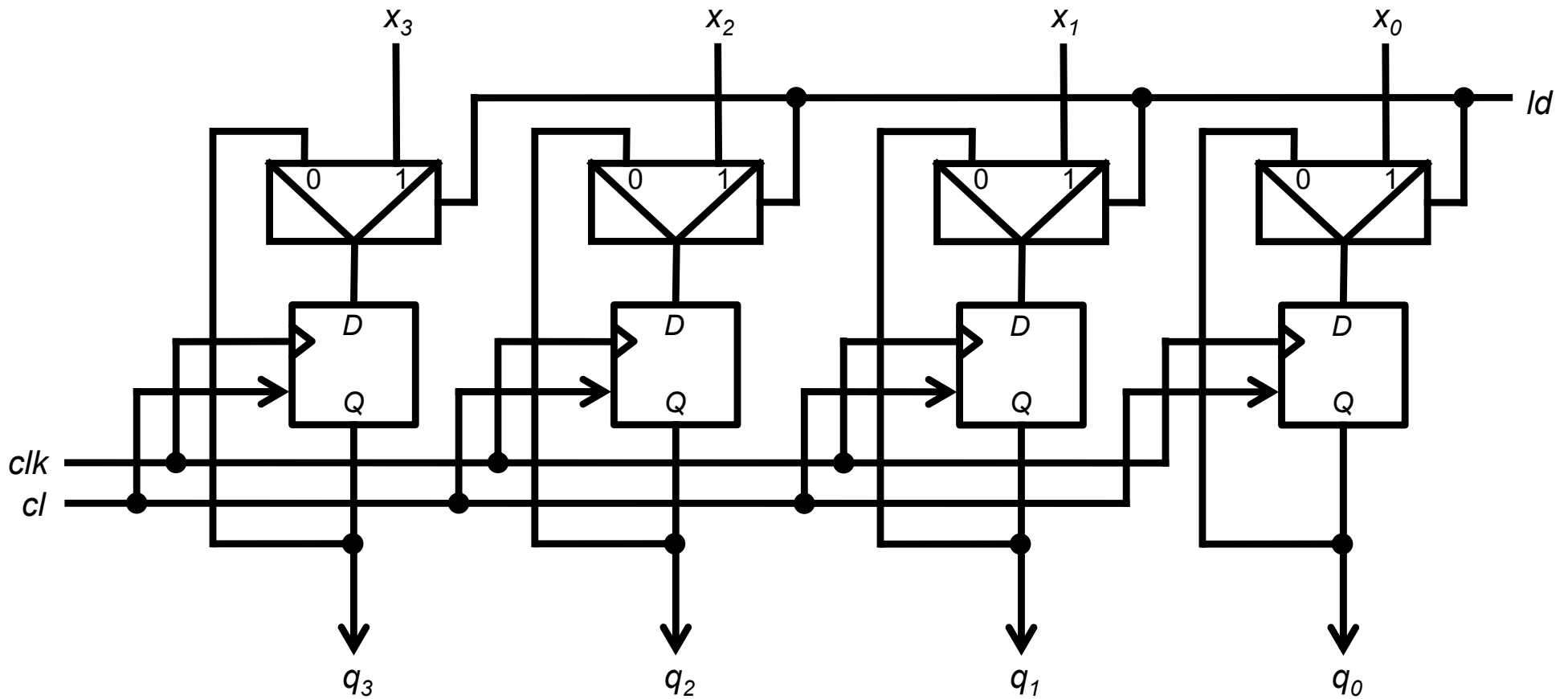
si la entrada ld vale 1, almacena la entrada,
si no, conserva el valor almacenado

$$\text{si } cl(t) = 1 \Rightarrow \underline{q}(t+\Delta t) = \underline{0}$$

$$\text{si } cl(t) = 0 \Rightarrow \underline{q}(t+1) = \begin{cases} \underline{q}(t) & \text{si } ld(t)=0 \\ \underline{x}(t) & \text{si } ld(t)=1 \end{cases}$$



Registro

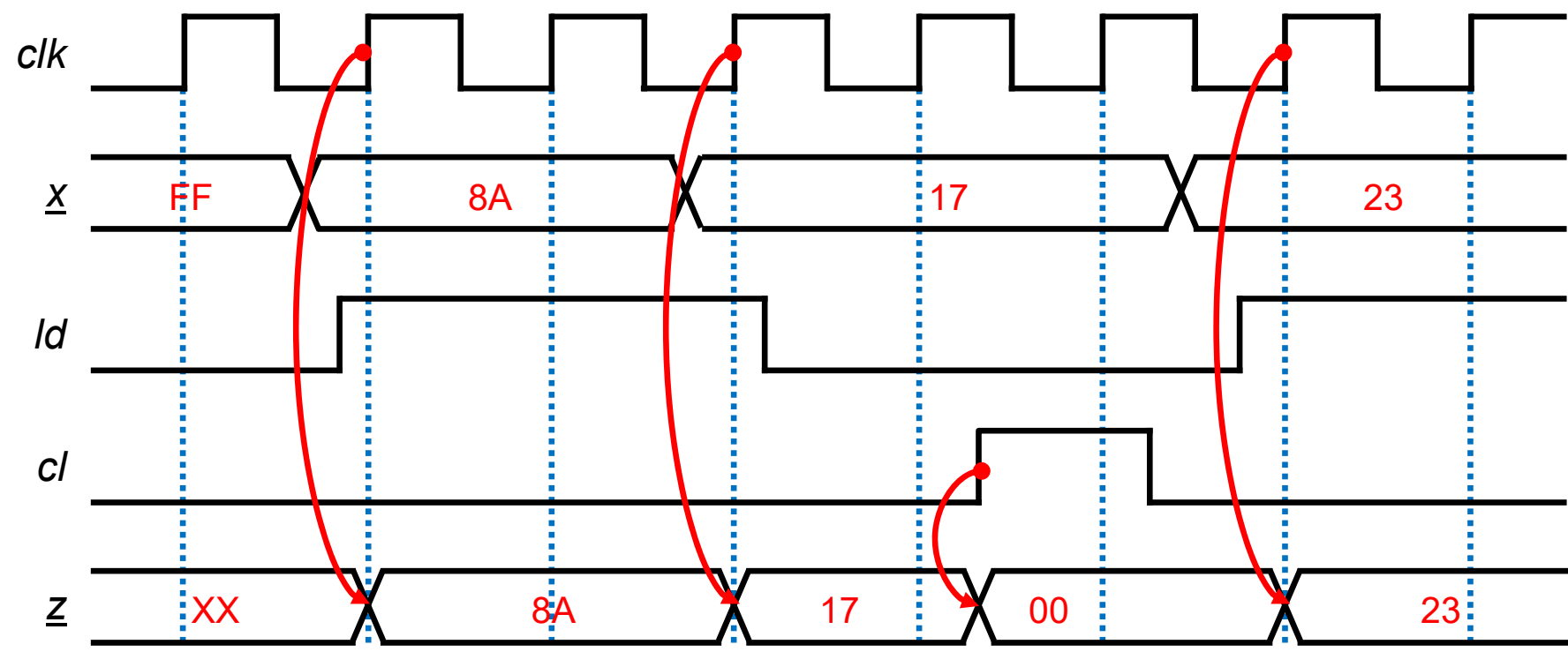
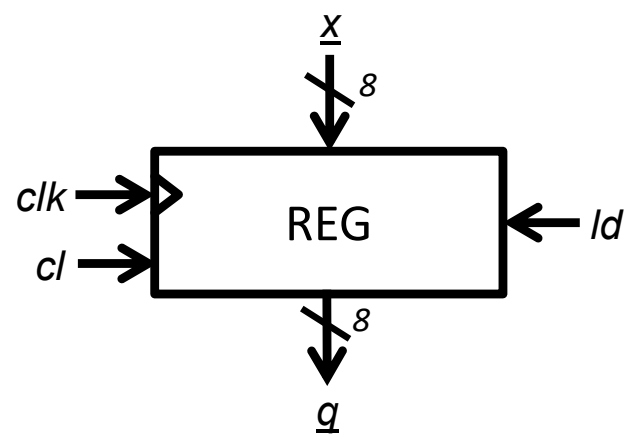


Implementación directa

Registro de 4 bits

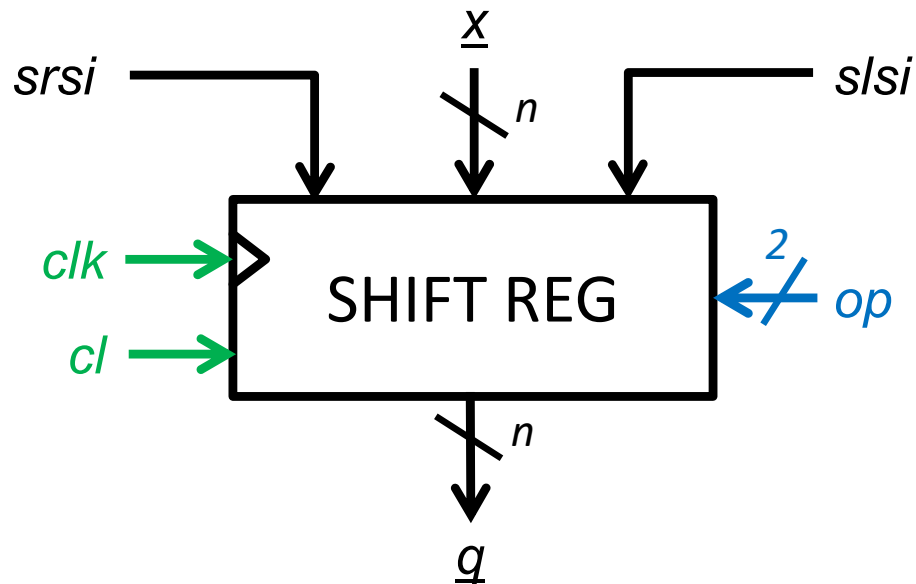


Registro





Registro de desplazamiento



Registro de desplazamiento de n bits

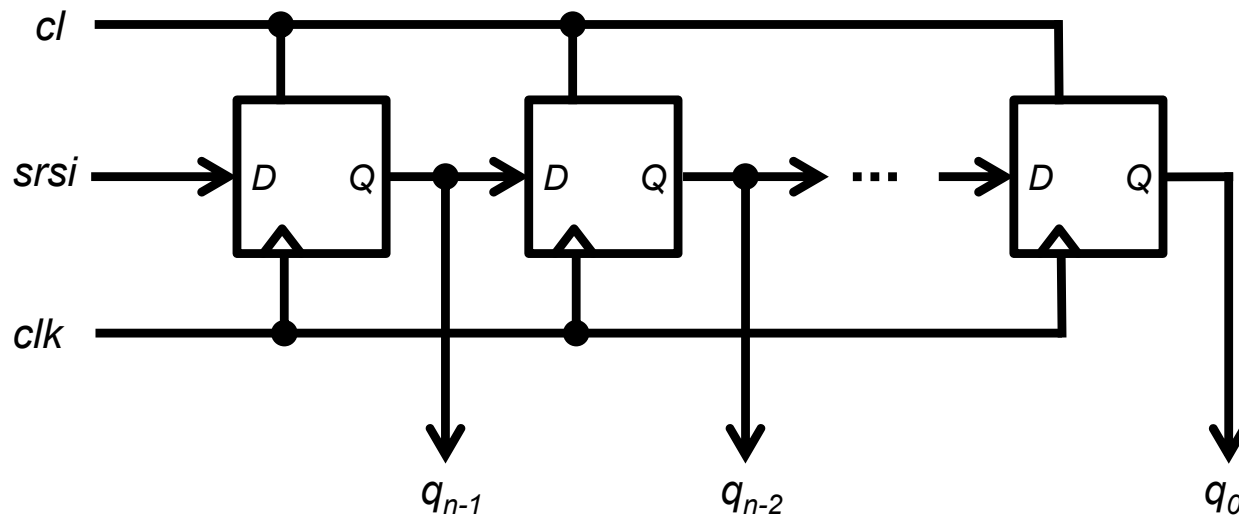
- \underline{x} 1 entrada paralela de datos de n bits
- \underline{q} 1 salida de datos de n bits
- srsi 1 entrada serie de datos para desplazamiento a la derecha
- slsi 1 entrada serie de datos para desplazamiento a la izquierda
- \underline{op} 1 entrada de selección de operación
- cl 1 entrada de inicialización asíncrona
- clk 1 entrada de reloj

$$\text{si } cl(t) = 1 \Rightarrow \underline{q}(t+\Delta t) = \underline{0}$$

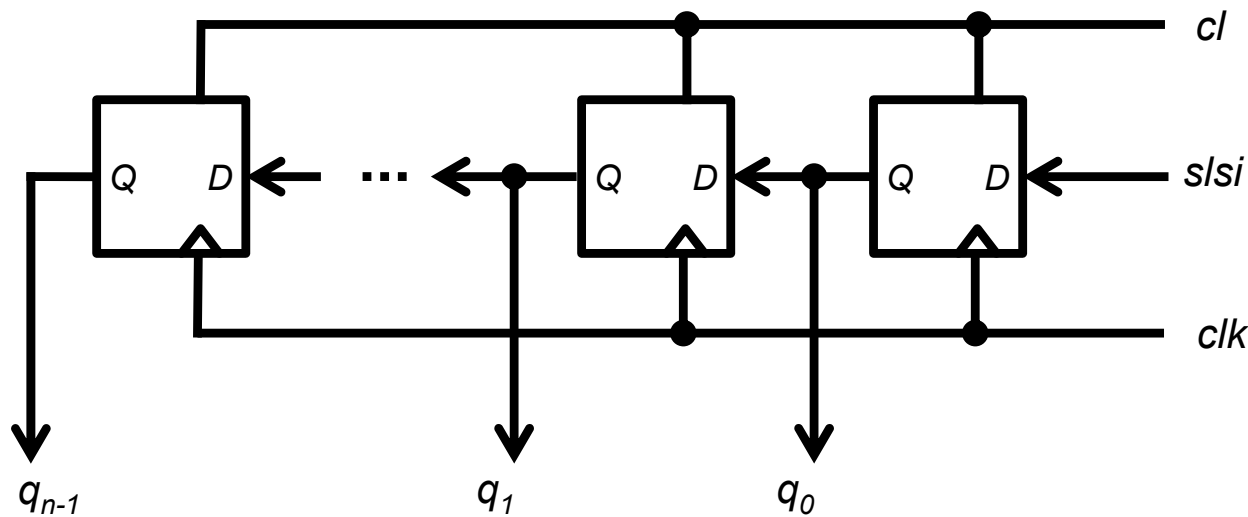
$$\text{si } cl(t) = 0 \Rightarrow \underline{q}(t+1) = \begin{cases} \underline{q}(t) & \text{si } op(t)=(00) \quad \textit{sin cambio} \\ (srsi, q_{n-1}(t) \dots q_1(t)) & \text{si } op(t)=(01) \quad \textit{desplaza a la derecha} \\ (q_{n-2}(t) \dots q_0(t), slsi) & \text{si } op(t)=(10) \quad \textit{desplaza a la izquierda} \\ \underline{x}(t) & \text{si } op(t)=(11) \quad \textit{carga paralela} \end{cases}$$



Registro de desplazamiento



Desplazamiento
a la derecha



Desplazamiento
a la izquierda

Registro de desplazamiento

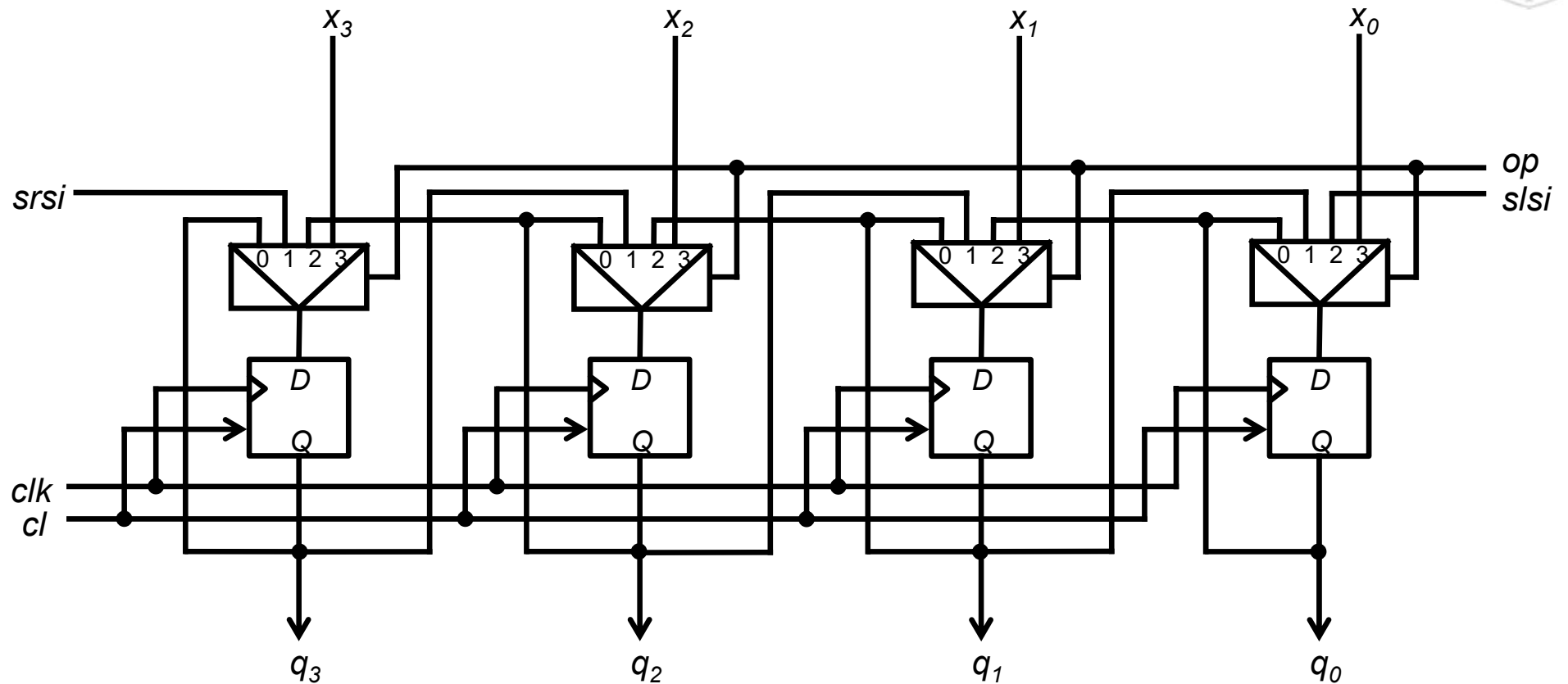


versión 14/07/23

tema 7:
Módulos secuenciales básicos

FC-1

8



Implementación directa

Registro de desplazamiento de 4 bits

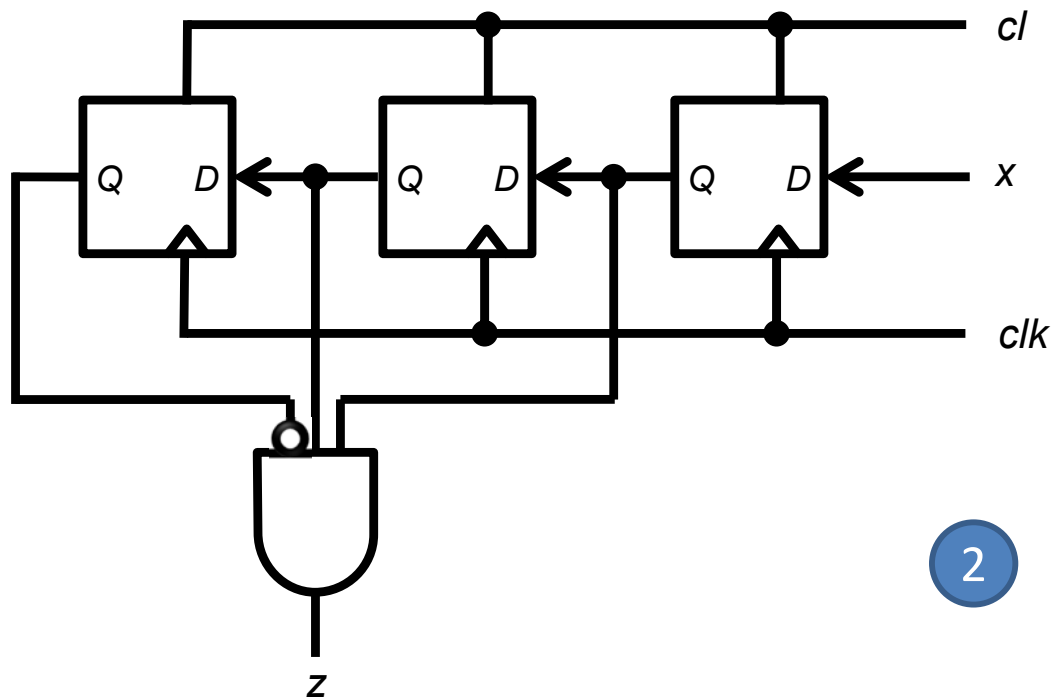


Registro de desplazamiento

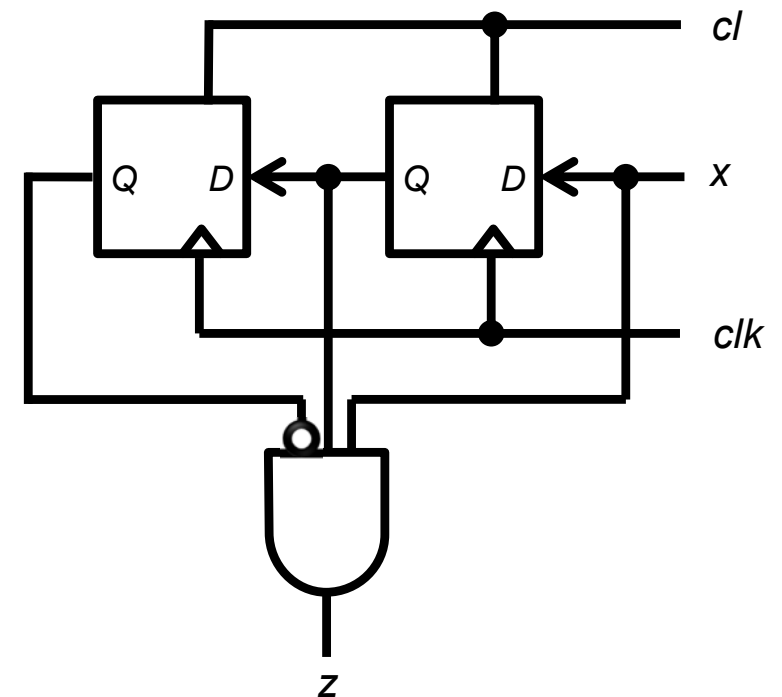
■ Aplicaciones al diseño:

1. Convertir datos serie en datos paralelo y viceversa.
2. Implementar reconocedores de patrones.

*Reconocedor del patrón = "011"
como máquina de Moore*



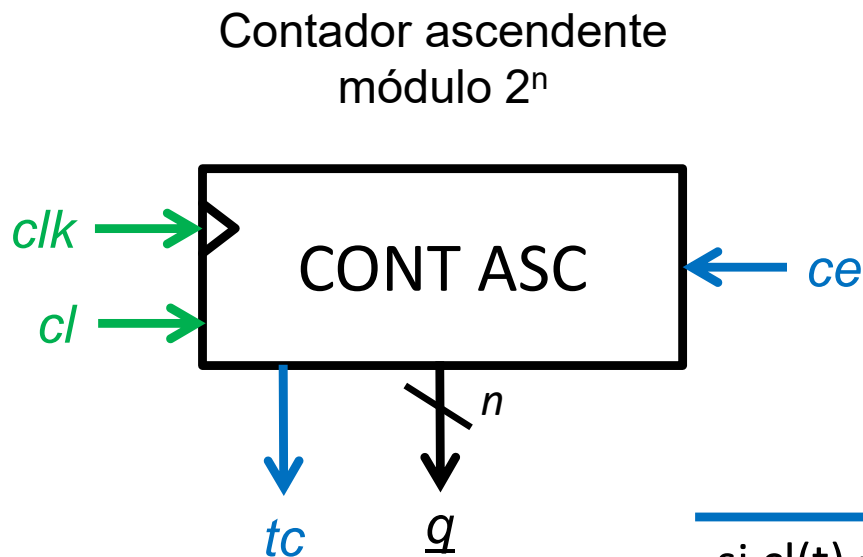
*Reconocedor del patrón = "011"
como máquina de Mealy*



2



Contador ascendente



-
- \underline{q} 1 salida de datos de n bits
 - ce 1 entrada de capacitación de cuenta
 - tc 1 salida de fin de cuenta
 - cl 1 entrada de inicialización asíncrona
 - clk 1 entrada de reloj
-

$$\text{si } cl(t) = 1 \Rightarrow tc(t+\Delta t) = 0$$

$$\text{si } cl(t) = 0 \Rightarrow tc(t+\Delta t) = \begin{cases} 1 & \text{si } \underline{q}(t)=2^n-1 \text{ y } ce(t)=1 \\ 0 & \text{en caso contrario} \end{cases}$$

$$\text{si } cl(t) = 1 \Rightarrow \underline{q}(t+\Delta t) = \underline{0}$$

$$\text{si } cl(t) = 0 \Rightarrow \underline{q}(t+1) = \begin{cases} [\underline{q}(t)+1] \bmod 2^n & \text{si } ce(t)=1 \\ \underline{q}(t) & \text{en caso contrario} \end{cases}$$



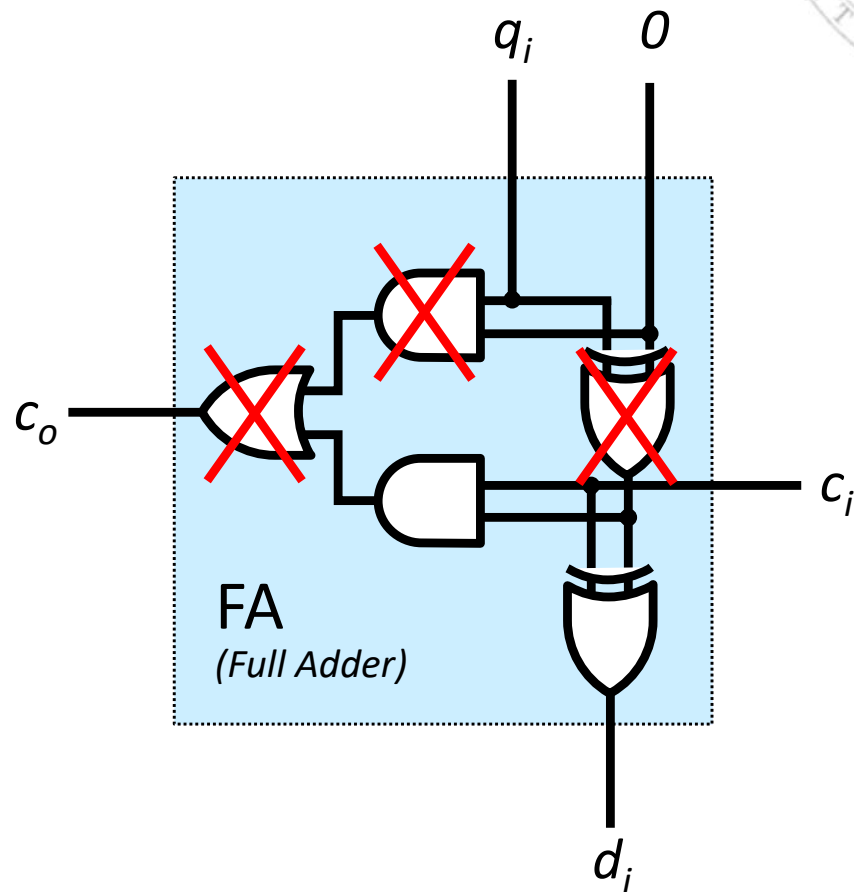
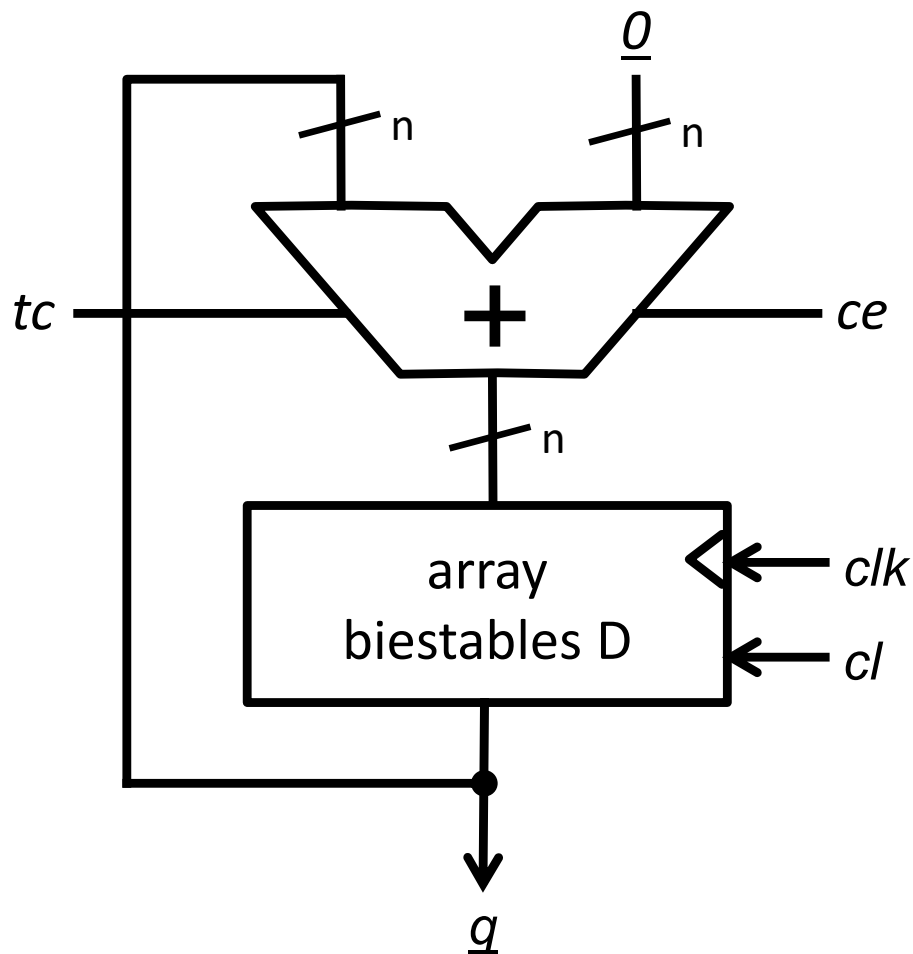
Contador ascendente

versión 14/07/23

tema 7:
Módulos secuenciales básicos

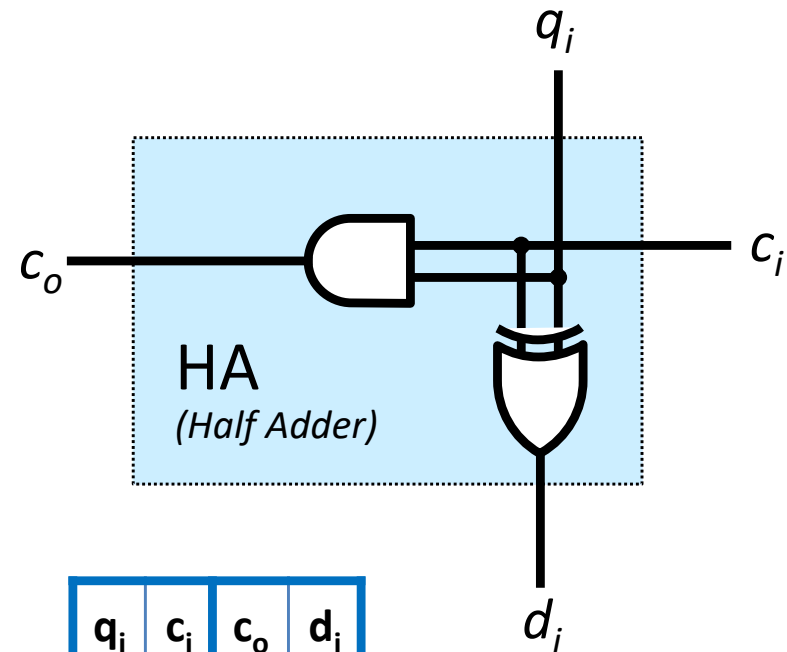
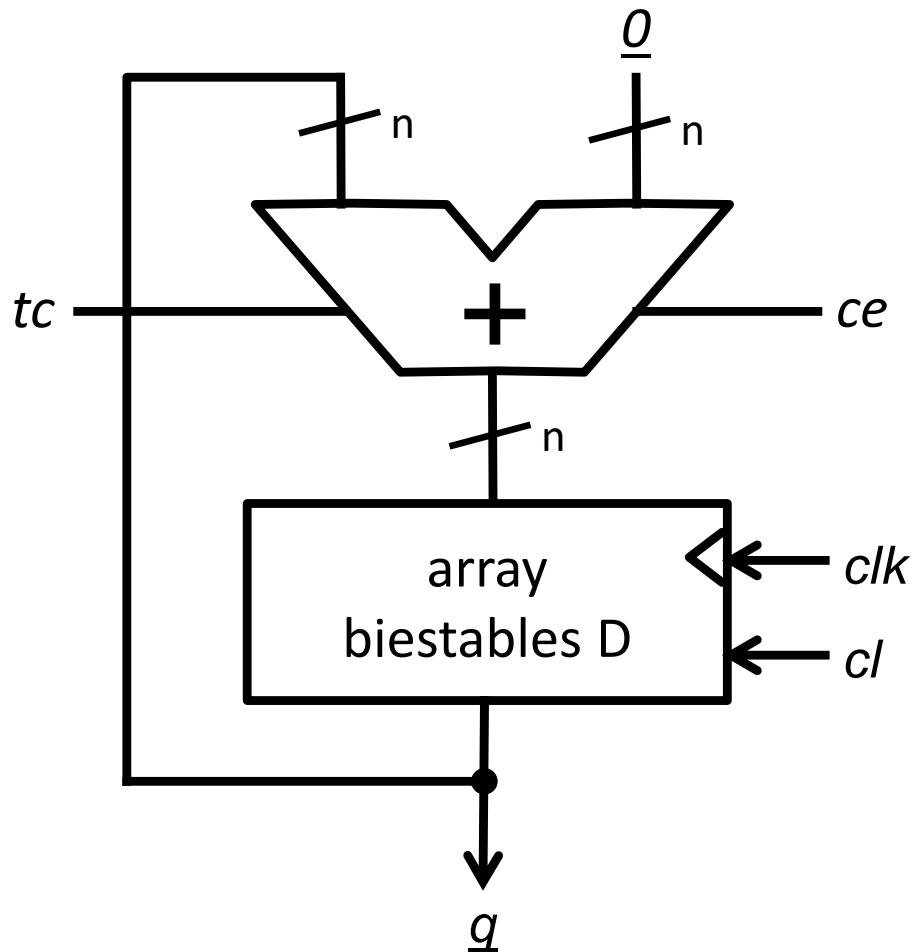
FC-1

11





Contador ascendente



q_i	c_i	c_o	d_i
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$d_i = q_i \oplus c_i$$

$$c_o = q_i \cdot c_i$$



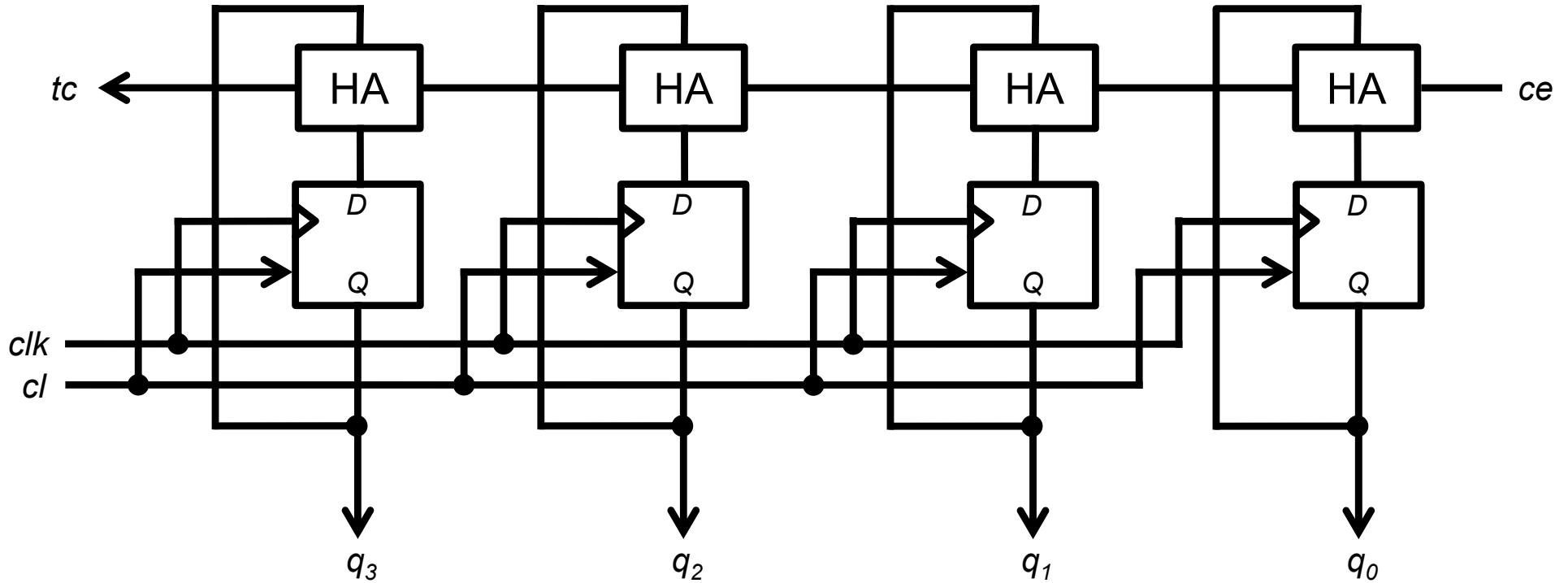
Contador ascendente

versión 14/07/23

tema 7:
Módulos secuenciales básicos

FC-1

13



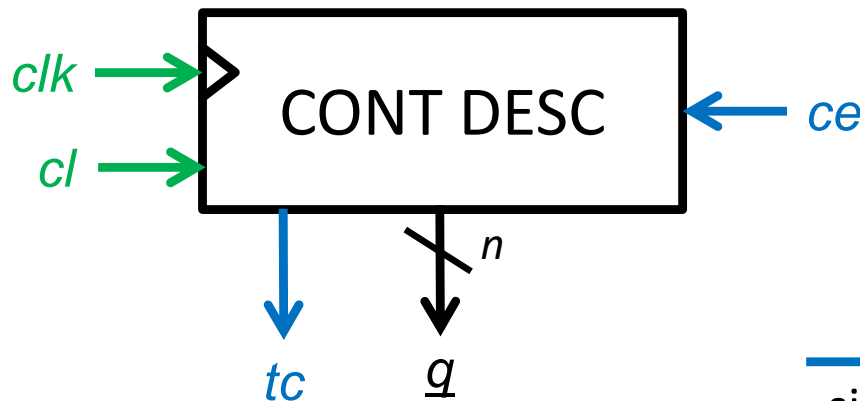
Implementación directa

Contador ascendente módulo 16



Contador descendente

Contador descendente
módulo 2^n



- \underline{q} 1 salida de datos de n bits
- ce 1 entrada de capacitación de cuenta
- tc 1 salida de fin de cuenta
- cl 1 entrada de inicialización asíncrona
- clk 1 entrada de reloj

$$\text{si } cl(t) = 1 \Rightarrow tc(t+\Delta t) = 0$$

$$\text{si } cl(t) = 0 \Rightarrow tc(t+\Delta t) = \begin{cases} 1 & \text{si } \underline{q}(t)=0 \text{ y } ce(t)=1 \\ 0 & \text{en caso contrario} \end{cases}$$

$$\text{si } cl(t) = 1 \Rightarrow \underline{q}(t+\Delta t) = \underline{0}$$

$$\text{si } cl(t) = 0 \Rightarrow \underline{q}(t+1) = \begin{cases} [\underline{q}(t)-1] \bmod 2^n & \text{si } ce(t)=1 \\ \underline{q}(t) & \text{en caso contrario} \end{cases}$$



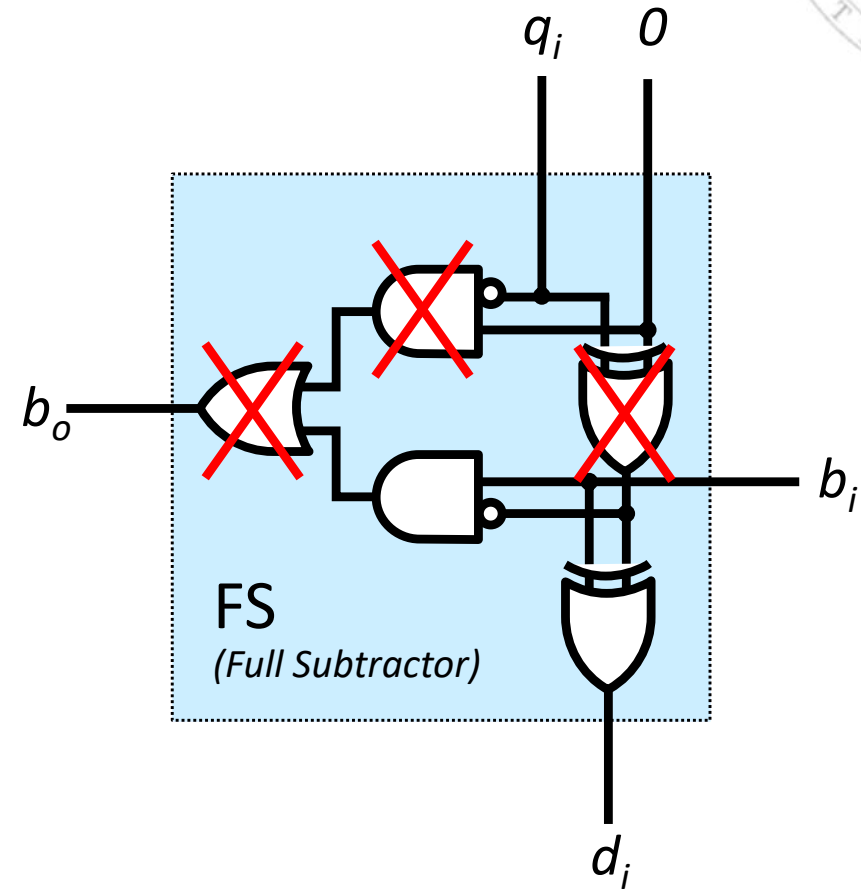
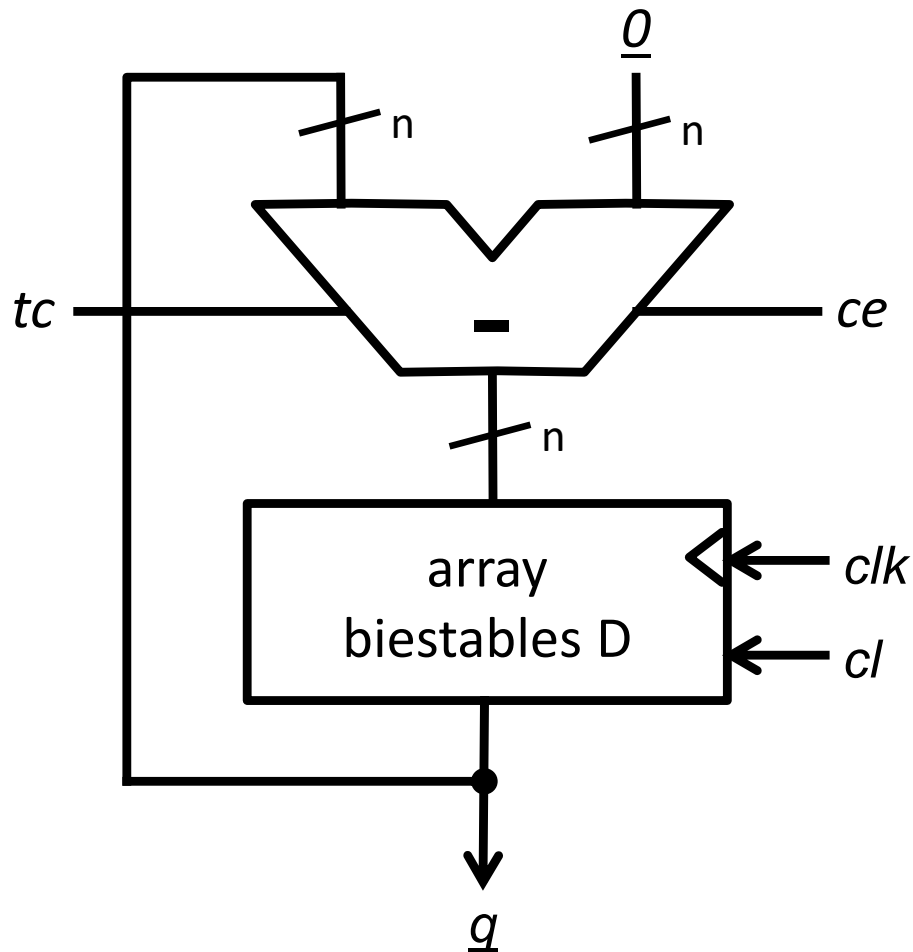
Contador descendente

versión 14/07/23

tema 7:
Módulos secuenciales básicos

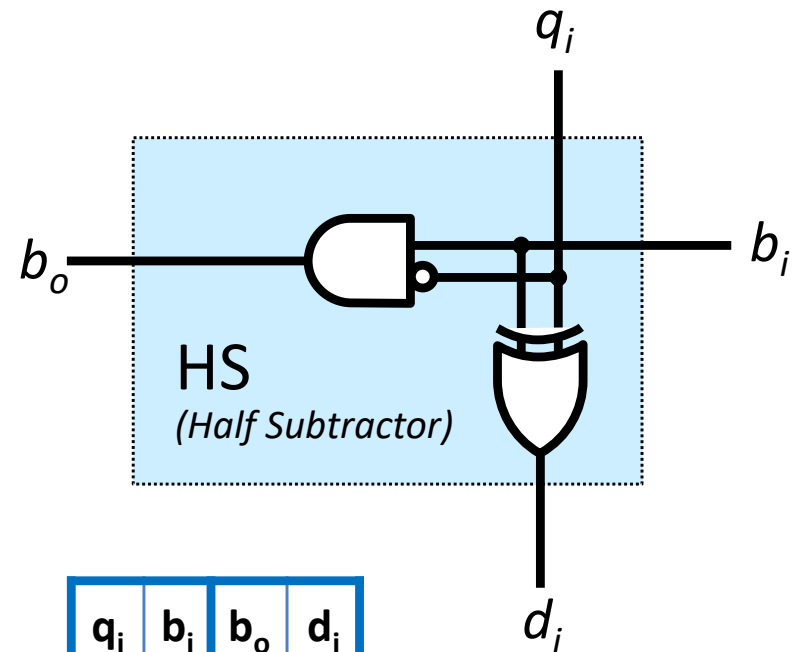
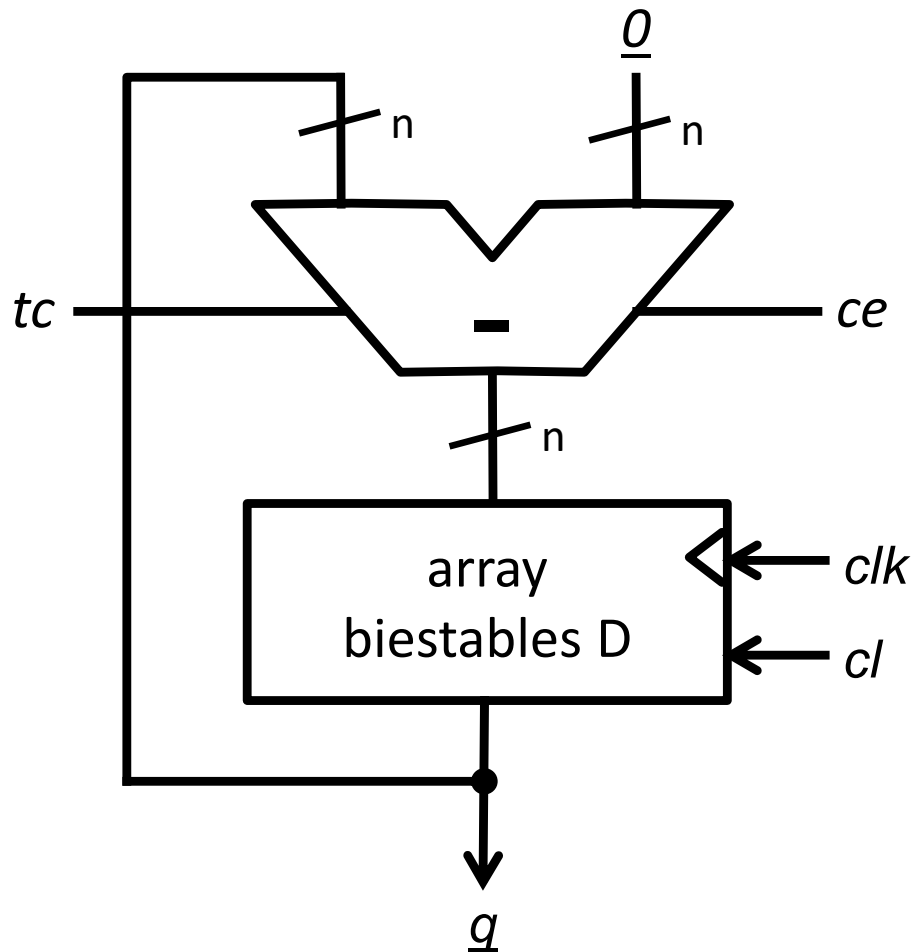
FC-1

15





Contador descendente



q_i	b_i	b_o	d_i
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$d_i = q_i \oplus b_i$$

$$b_o = \bar{q}_i \cdot b_i$$



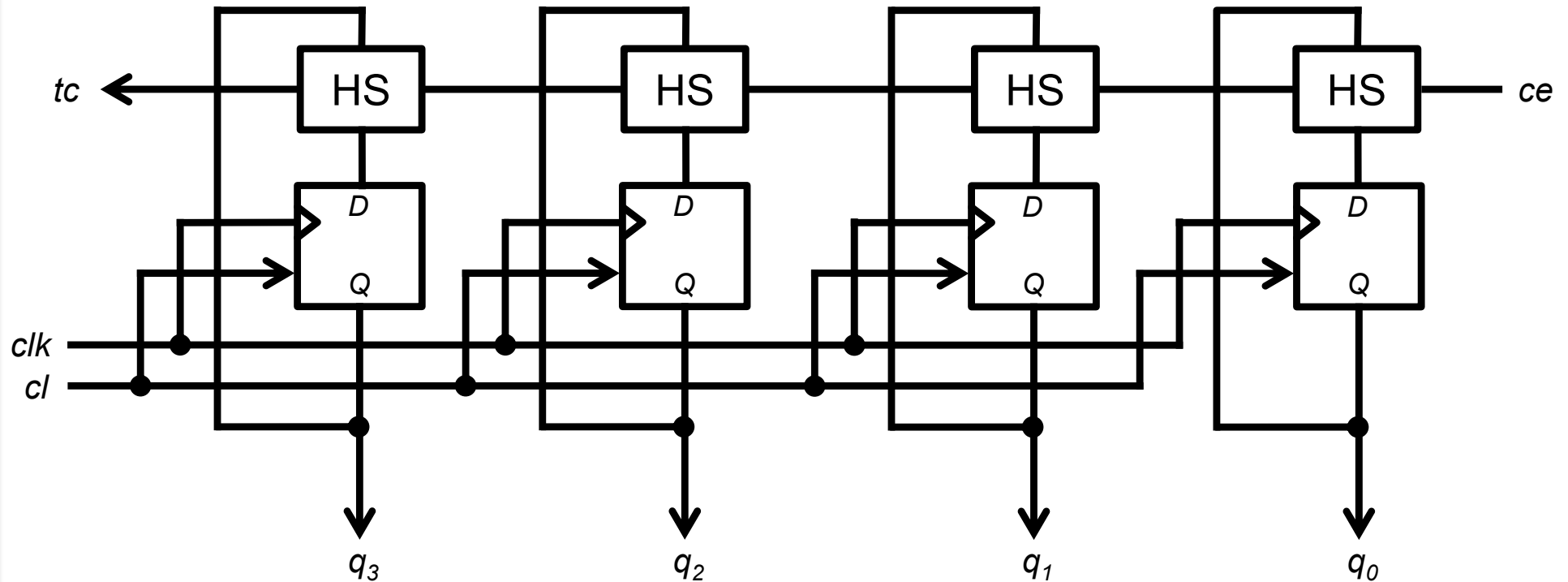
Contador descendente

versión 14/07/23

tema 7:
Módulos secuenciales básicos

FC-1

17



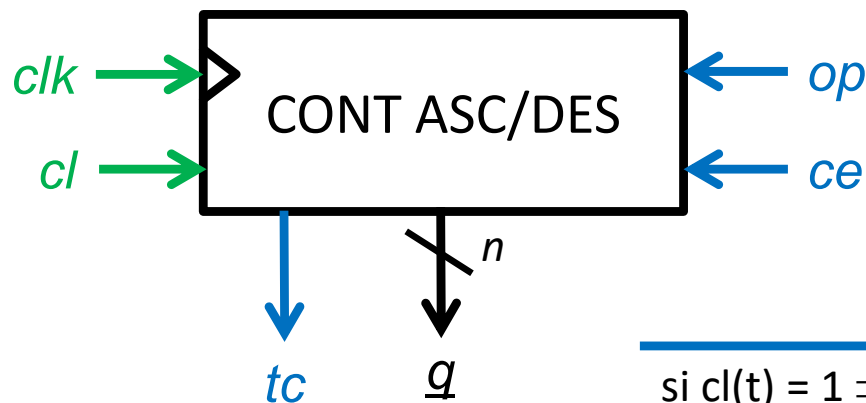
Implementación directa

Contador descendente módulo 16



Contador ascendente/descendente

Contador ascendente/descendente
módulo 2^n



- \underline{q} 1 salida de datos de n bits
- op 1 entrada de selección de operación
- ce 1 entrada de capacitación de cuenta
- tc 1 salida de fin de cuenta
- cl 1 entrada de inicialización asíncrona
- clk 1 entrada de reloj

$$\text{si } cl(t) = 1 \Rightarrow tc(t+\Delta t) = 0$$

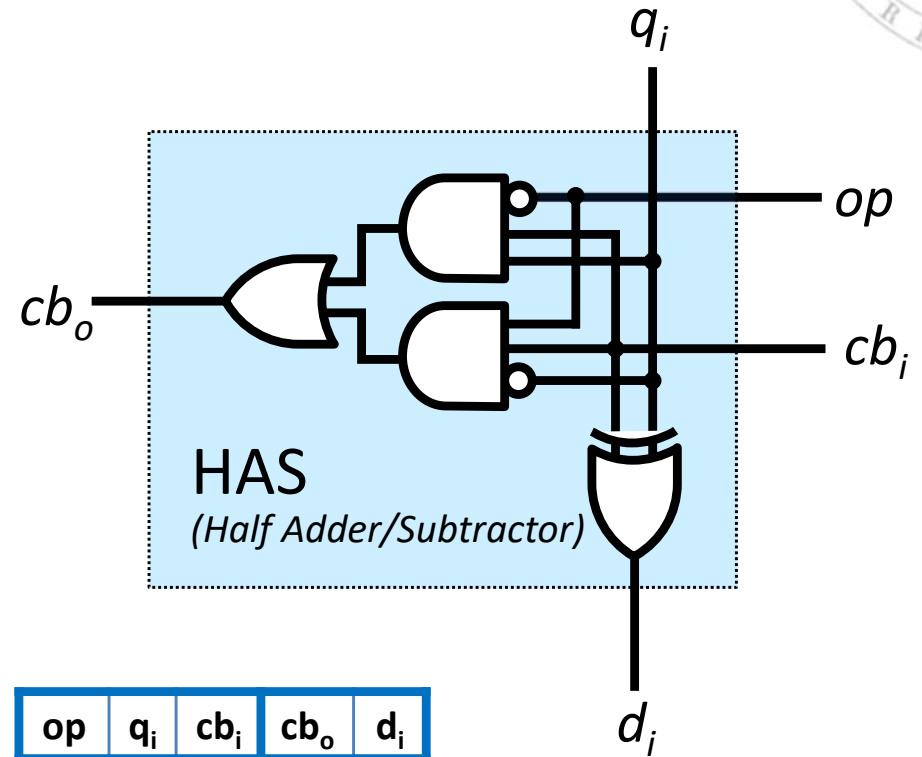
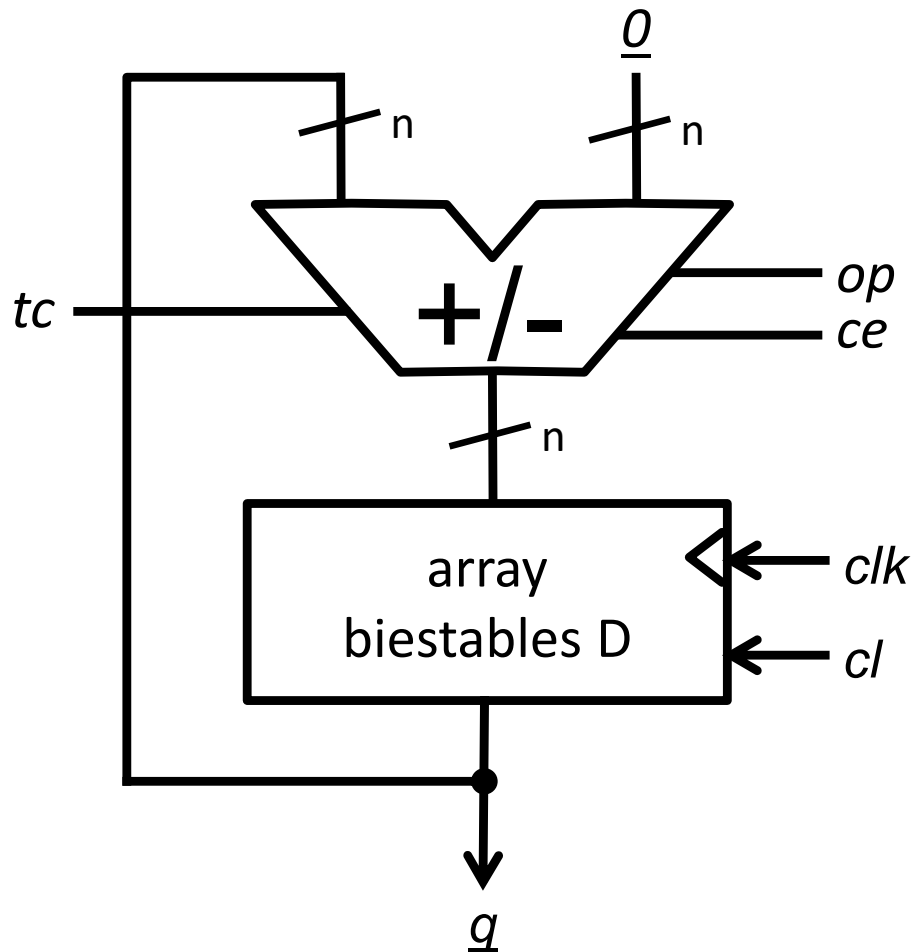
$$\text{si } cl(t) = 0 \Rightarrow tc(t+\Delta t) = \begin{cases} 1 & \text{si } (\underline{q}(t)=2^n-1 \text{ y } op(t)=0 \text{ y } ce(t)=1) \\ & \text{ó } (\underline{q}(t)=0 \text{ y } op(t)=1 \text{ y } ce(t)=1) \\ 0 & \text{en caso contrario} \end{cases}$$

$$\text{si } cl(t) = 1 \Rightarrow \underline{q}(t+\Delta t) = \underline{0}$$

$$\text{si } cl(t) = 0 \Rightarrow \underline{q}(t+1) = \begin{cases} [\underline{q}(t)+1] \bmod 2^n & \text{si } op(t)=0 \text{ y } ce(t)=1 \\ [\underline{q}(t)-1] \bmod 2^n & \text{si } op(t)=1 \text{ y } ce(t)=1 \\ \underline{q}(t) & \text{en otro caso} \end{cases}$$



Contador ascendente/descendente



op	q _i	cb _i	cb _o	d _i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0

$$d_i = q_i \oplus cb_i$$

$$cb_o = \overline{op} \cdot q_i \cdot cb_i + op \cdot \overline{q_i} \cdot cb_i$$



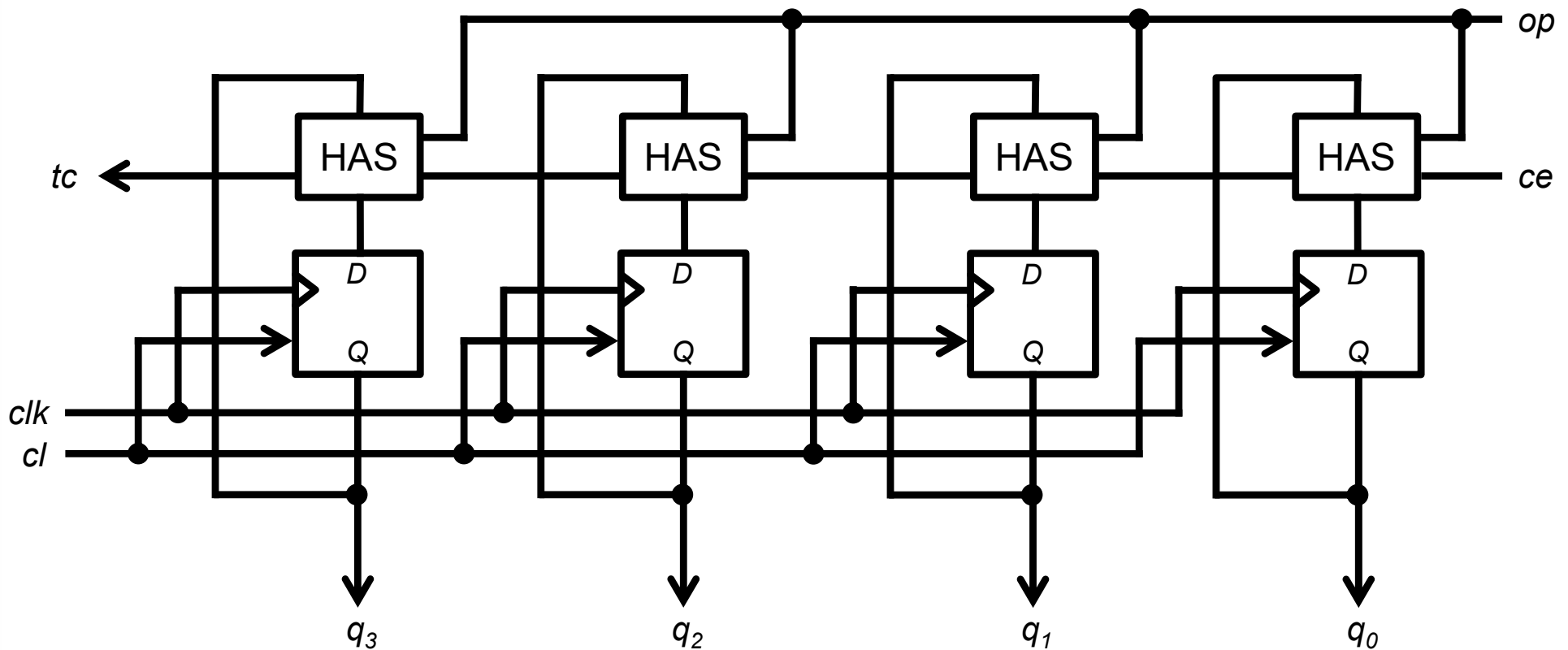
Contador ascendente/descendente

versión 14/07/23

tema 7:
Módulos secuenciales básicos

FC-1

20

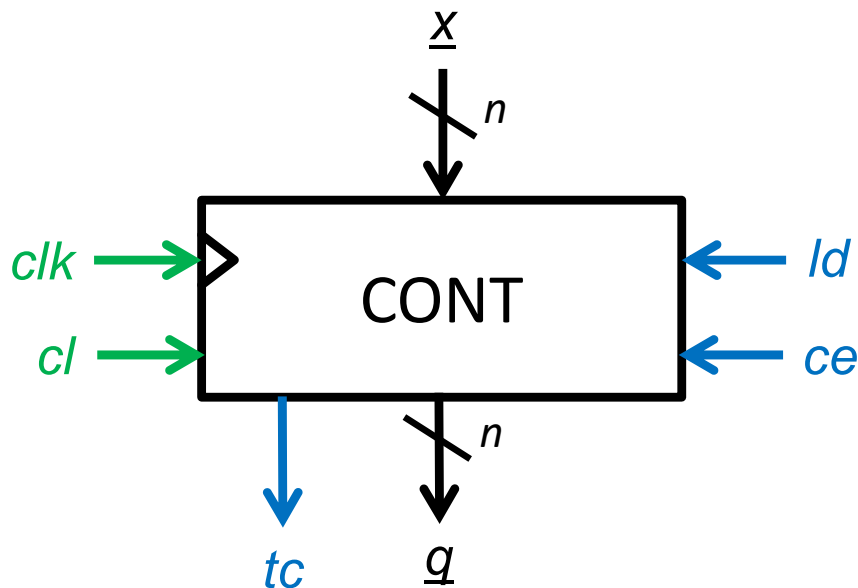


Implementación directa

Contador ascendente/descendente módulo 16



Contador con carga paralela



- \underline{x} 1 entrada de datos de n bits
- \underline{q} 1 salida de datos de n bits
- ld 1 entrada de carga paralela
- ce 1 entrada de capacitación de cuenta
- tc 1 salida de fin de cuenta
- cl 1 entrada de inicialización asíncrona
- clk 1 entrada de reloj

Contador ascendente con carga paralela módulo 2^n

$$\text{si } cl(t) = 1 \Rightarrow tc(t+\Delta t) = 0$$

$$\text{si } cl(t) = 0 \Rightarrow tc(t+\Delta t) = \begin{cases} 1 & \text{si } \underline{q}(t)=2^n-1 \text{ y } ld(t)=0 \text{ y } ce(t)=1 \\ 0 & \text{en caso contrario} \end{cases}$$

$$\text{si } cl(t) = 1 \Rightarrow \underline{q}(t+\Delta t) = \underline{0}$$

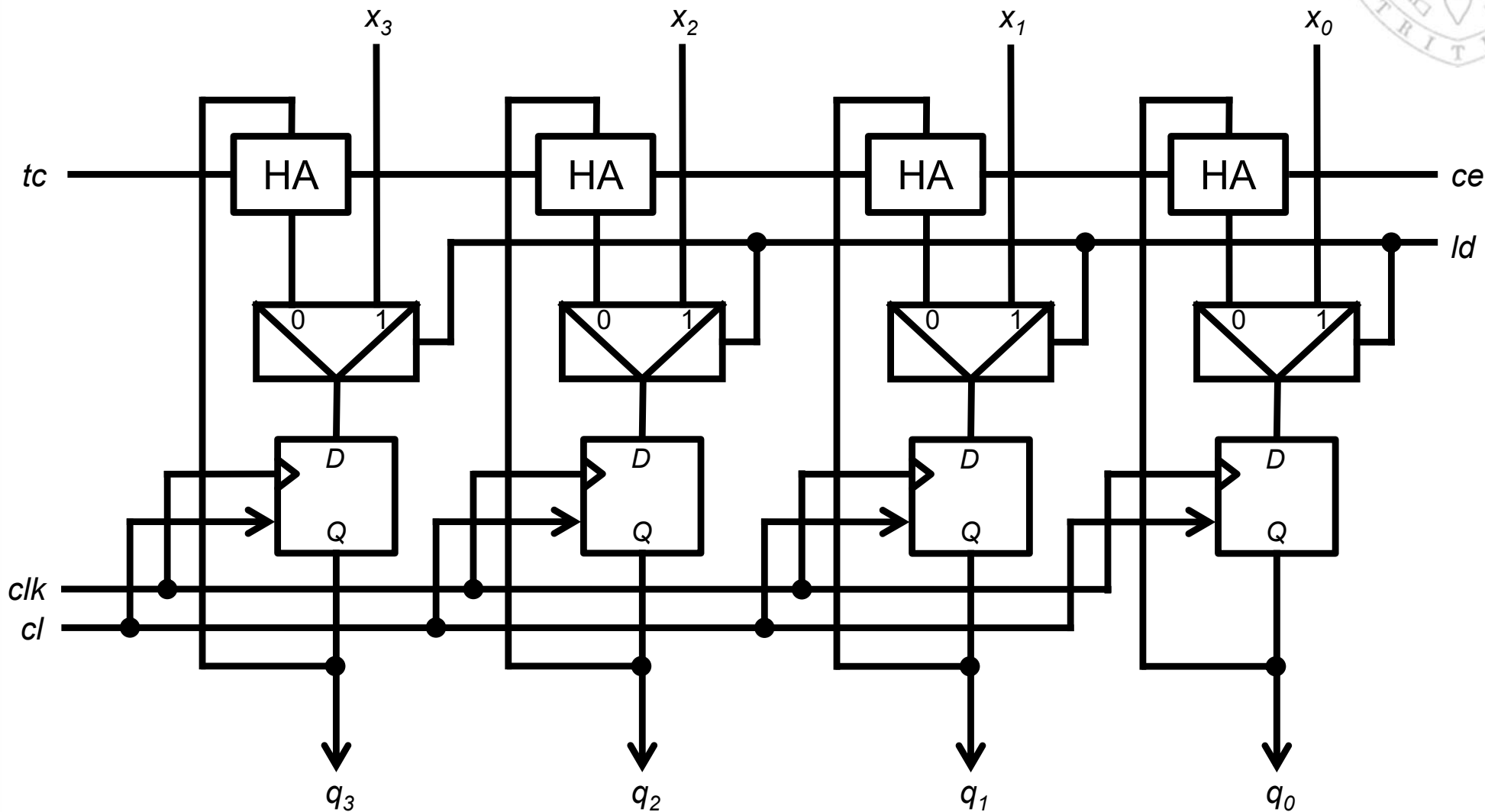
$$\text{si } cl(t) = 0 \Rightarrow \underline{q}(t+1) = \begin{cases} \underline{x}(t) & \text{si } ld(t)=1 \\ [\underline{q}(t)+1] \bmod 2^n & \text{si } ld(t)=0 \text{ y } ce(t)=1 \\ \underline{q}(t) & \text{en caso contrario} \end{cases}$$



Contador con carga paralela

versión 14/07/23

tema 7:
Módulos secuenciales básicos



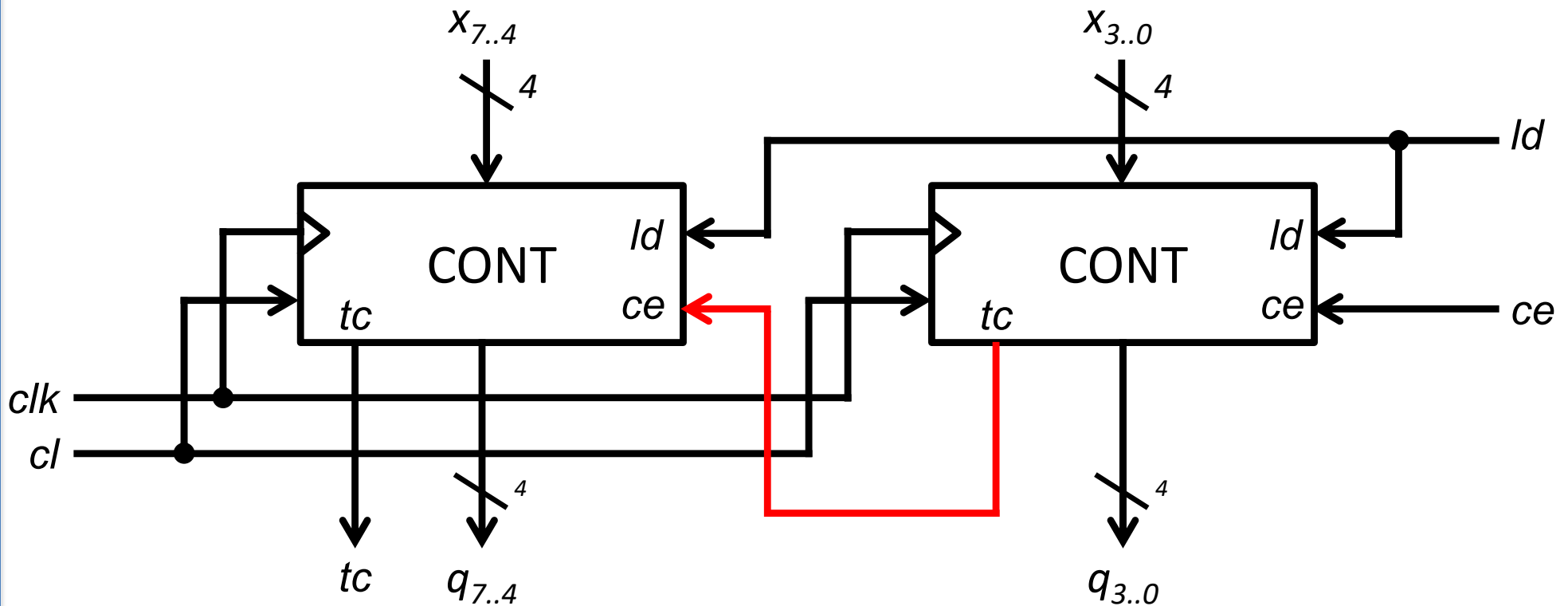
Implementación directa

Contador ascendente con carga paralela módulo 16



Contador con carga paralela

- Varios contadores se pueden componer en serie para comportarse como un contador de **mayor anchura**.



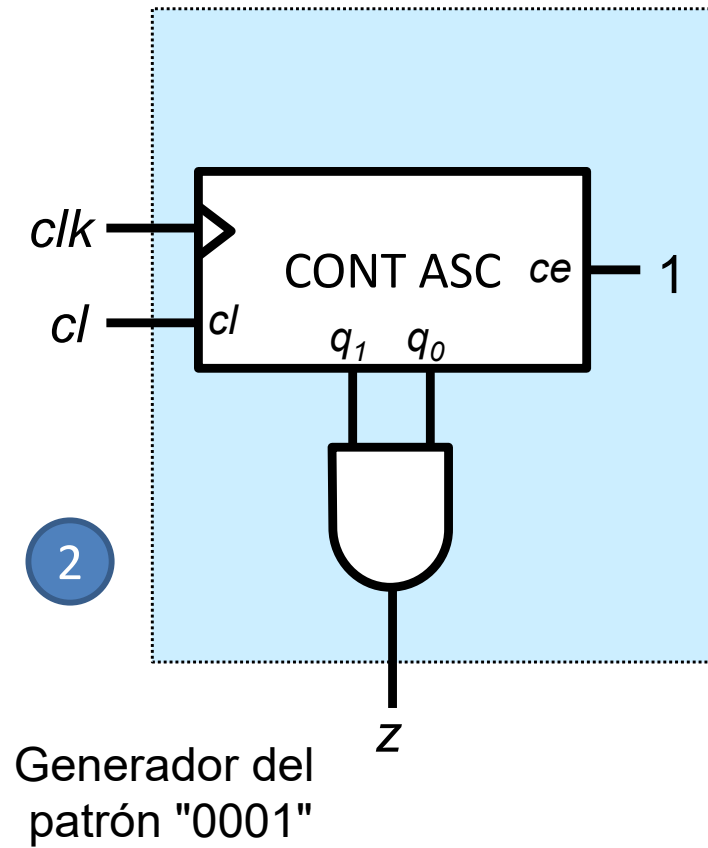
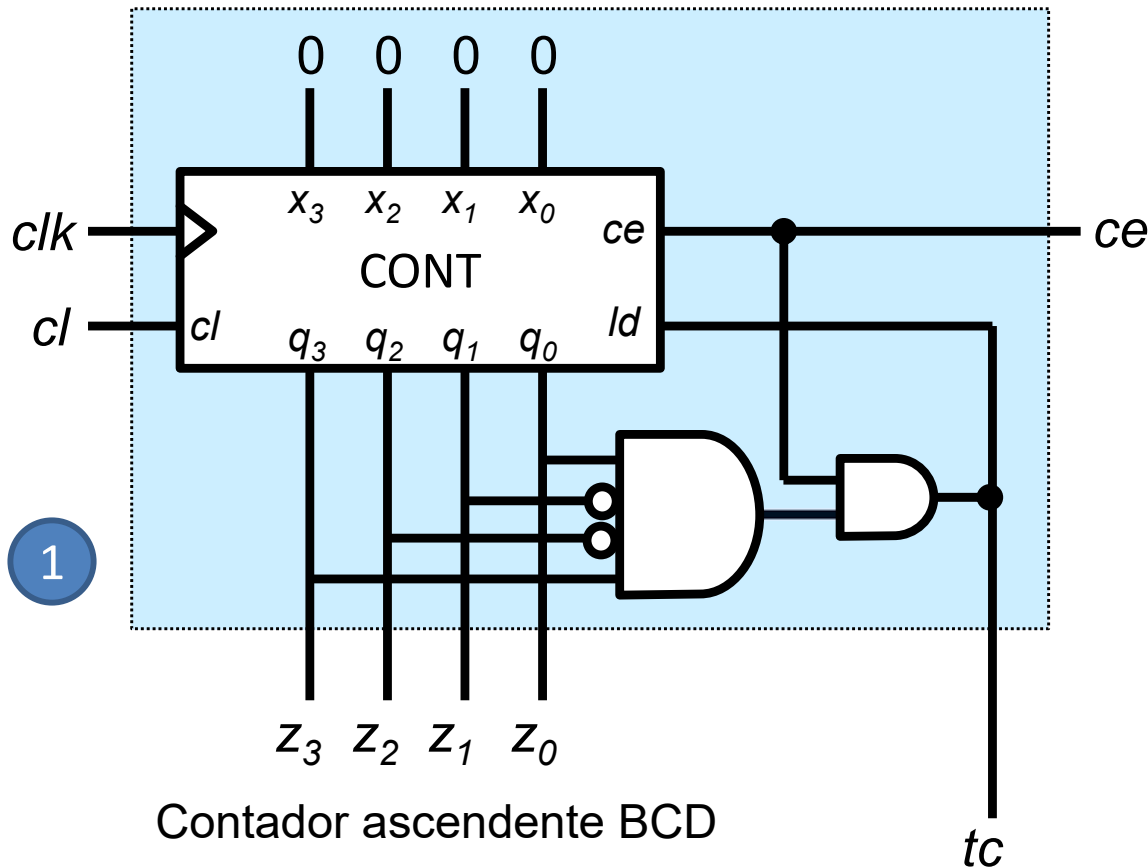
Implementación en cascada

Contador ascendente con carga paralela módulo 256



Contadores

- Aplicaciones al diseño:
 1. Generar secuencias (secuenciador).
 2. Generar patrones.





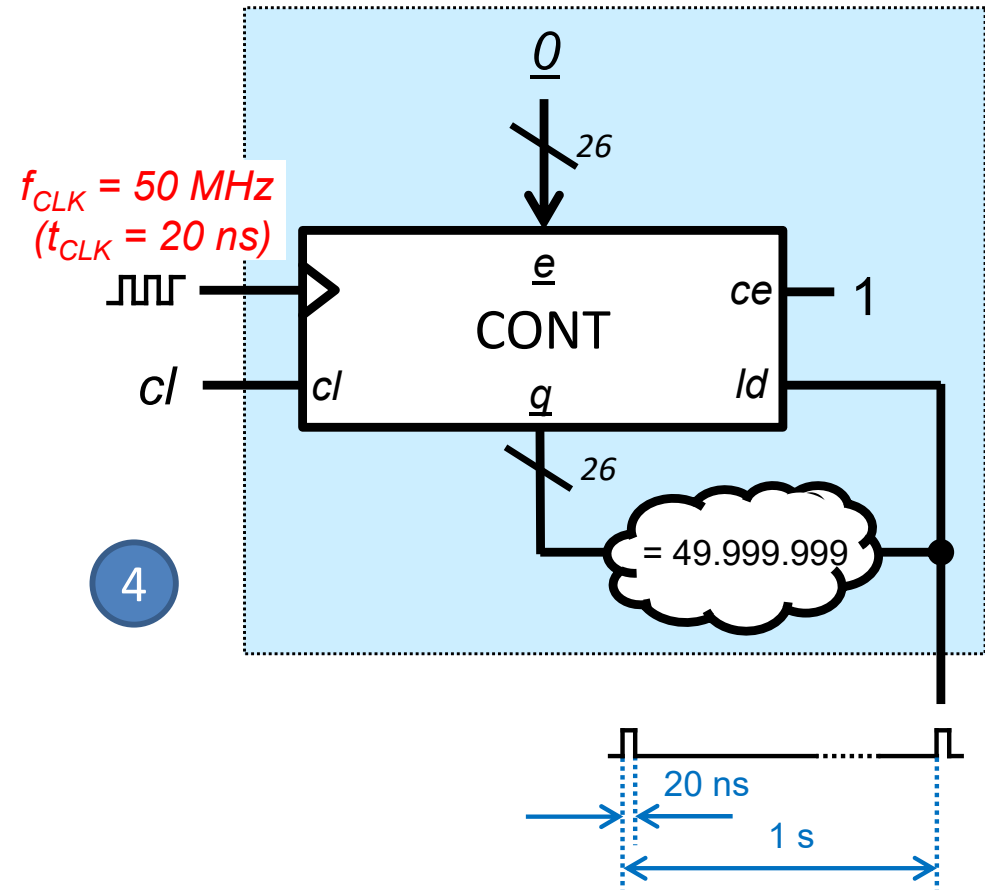
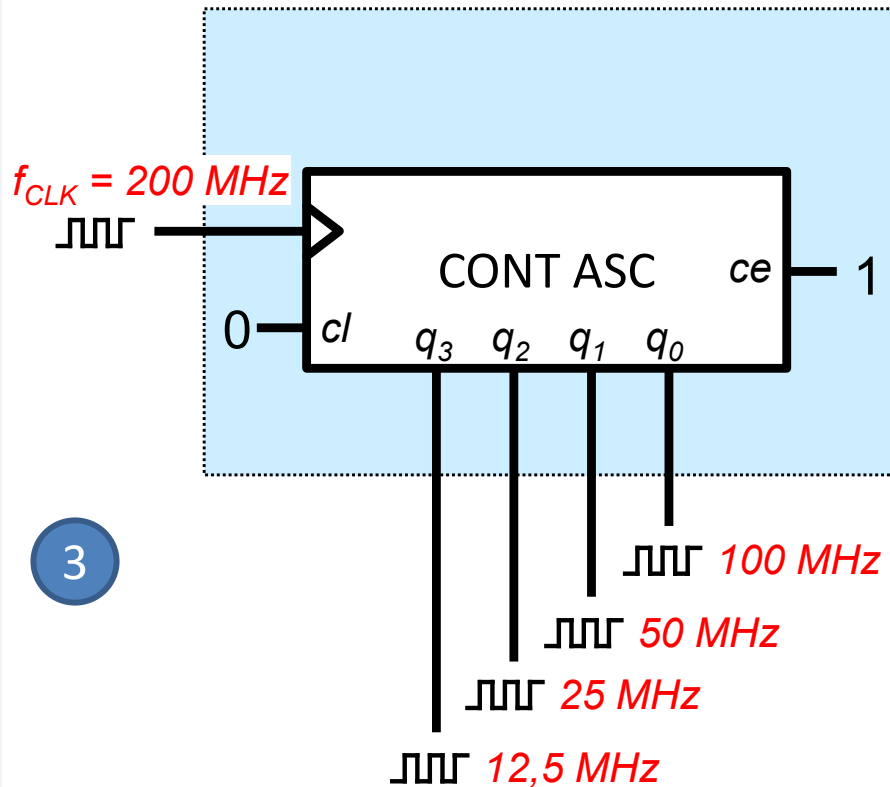
Contadores

- Aplicaciones al diseño:
 3. Dividir frecuencias.
 4. Medir tiempo (temporizador).

$$(\text{num. ciclos}) = \text{tiempo} / t_{\text{clk}} \equiv \text{tiempo} \times f_{\text{clk}}$$

$$1 \text{ s} \equiv 1 \cdot 10^9 \text{ ns}$$

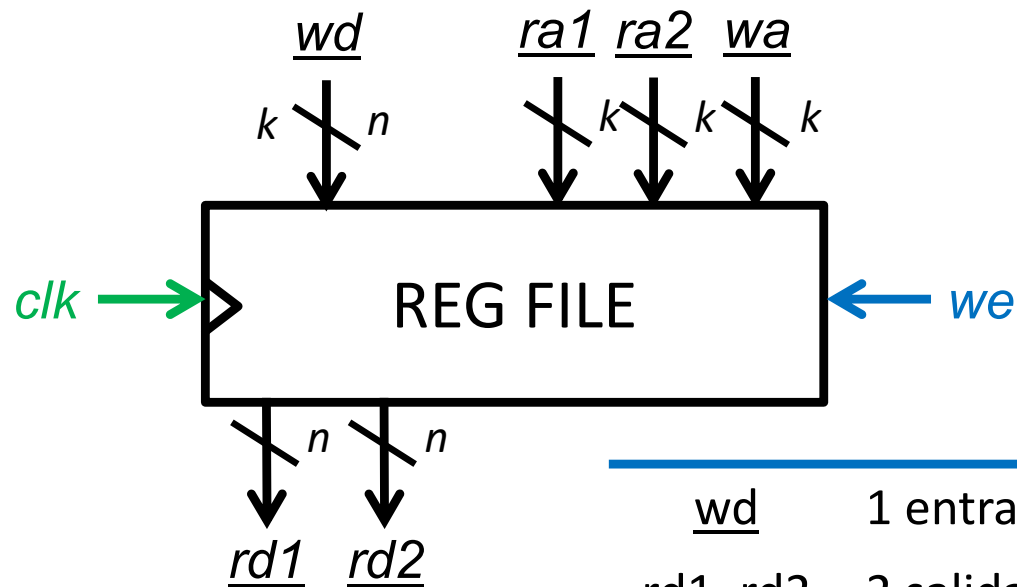
$$(1 \cdot 10^9 \text{ ns}) / (20 \text{ ns/ciclo}) = 50.000.000 \text{ ciclos}$$





Banco de registros

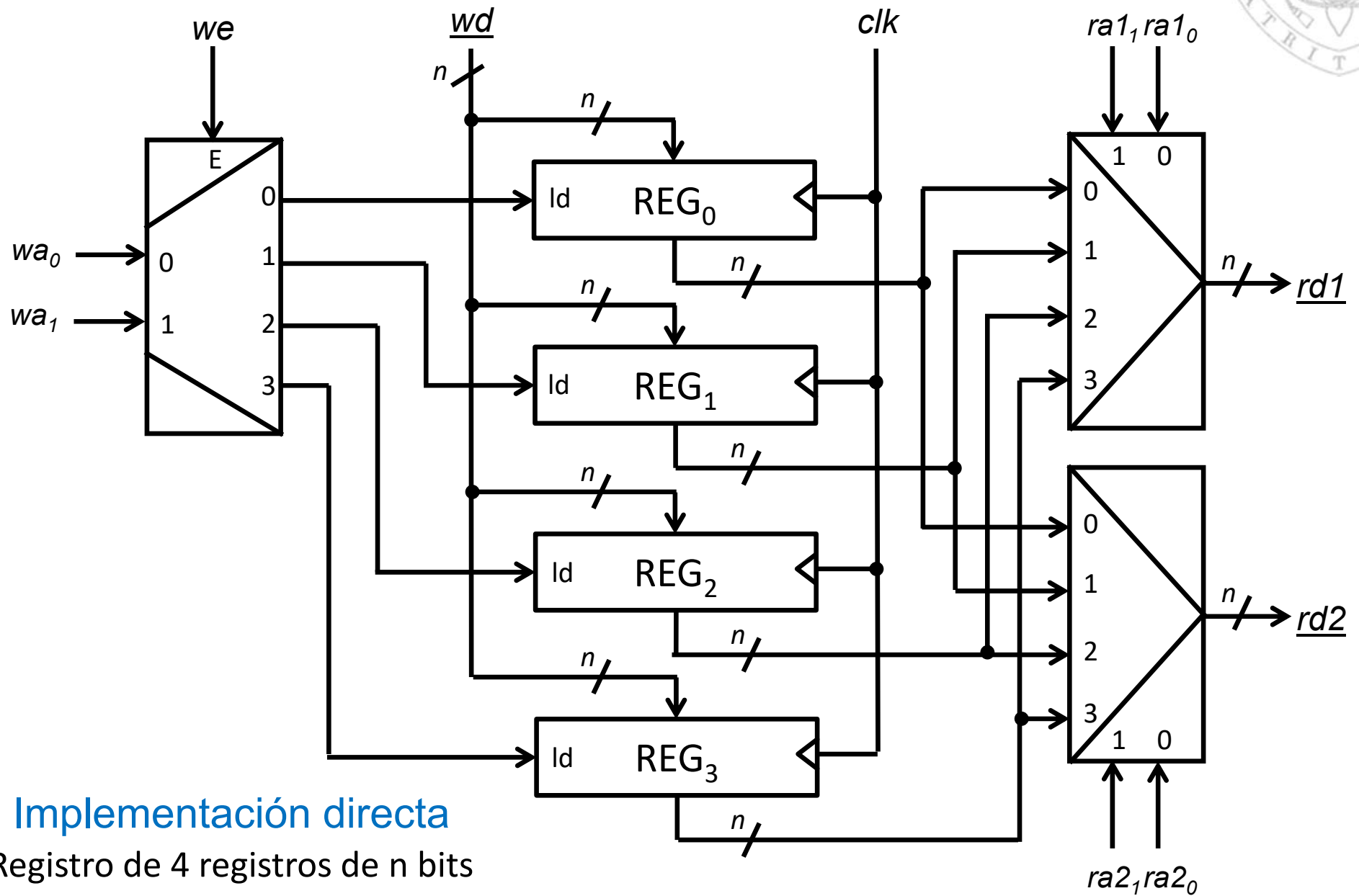
Banco de 2^k registros de n bits
(con 1 entrada de datos y 2 salidas de datos)



<u>wd</u>	1 entrada de datos de n bits
<u>rd1</u> , <u>rd2</u>	2 salidas de datos de n bits
<u>wa</u>	1 entrada de dirección de escritura de k bits
<u>ra1</u> , <u>ra2</u>	2 entradas de dirección de lectura de k bits
<u>we</u>	1 entrada de capacitación de escritura
<u>clk</u>	1 entrada de reloj



Banco de registros

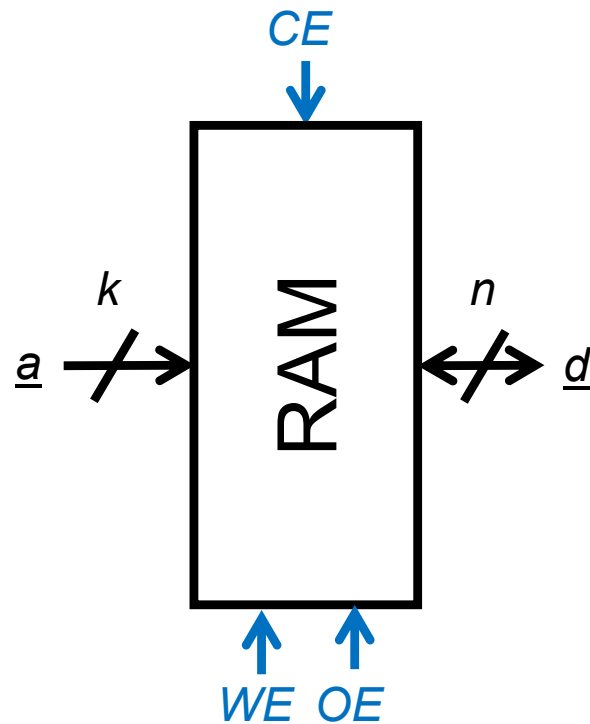


Implementación directa

Registro de 4 registros de n bits



RAM (Random-Access Memory)



RAM $2^k \times n$
(2^k palabras de n bits)

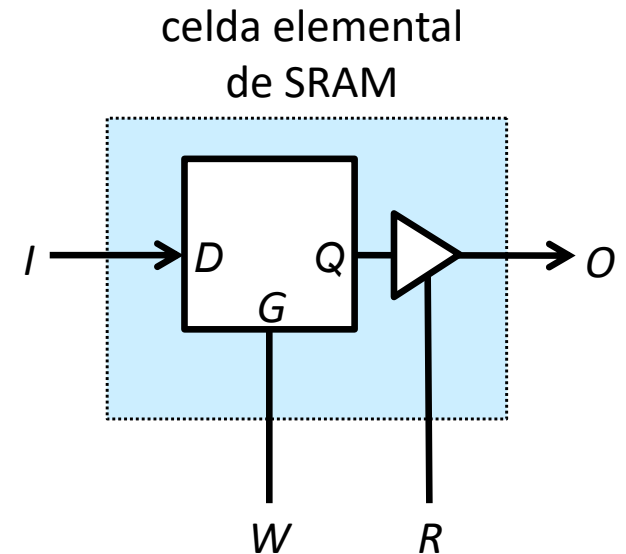
-
- d 1 entrada/salida de datos de n bits
 - a 1 entrada de dirección de k bits
 - CE 1 entrada de capacitación
 - OE 1 entrada de capacitación de lectura
 - WE 1 entrada de capacitación de escritura
-

*memoria volátil de capaz de almacenar
 2^k palabras de n bits cada una*

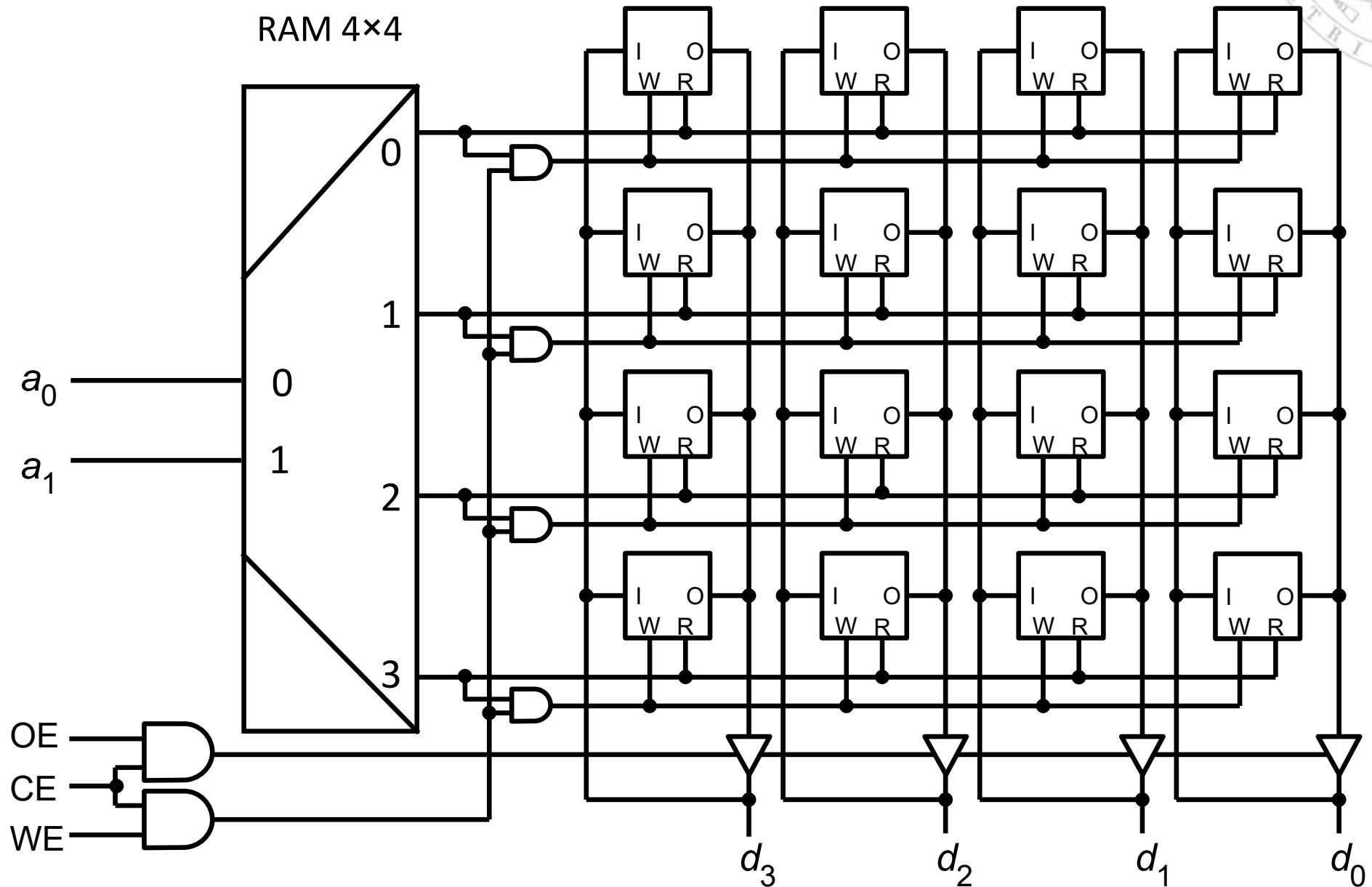


RAM (Random-Access Memory)

- **SRAM (Static RAM)**
 - Cada bit se almacena en un latch.
 - No requiere refresco.
- **DRAM (Dynamic RAM)**
 - Cada bit se almacena en un condensador.
 - Requiere refresco.
- **SDRAM (Synchronous Dynamic RAM)**
 - Cada bit se almacena en un condensador.
 - Requiere refresco.
 - El interfaz de lectura/escritura es síncrono.



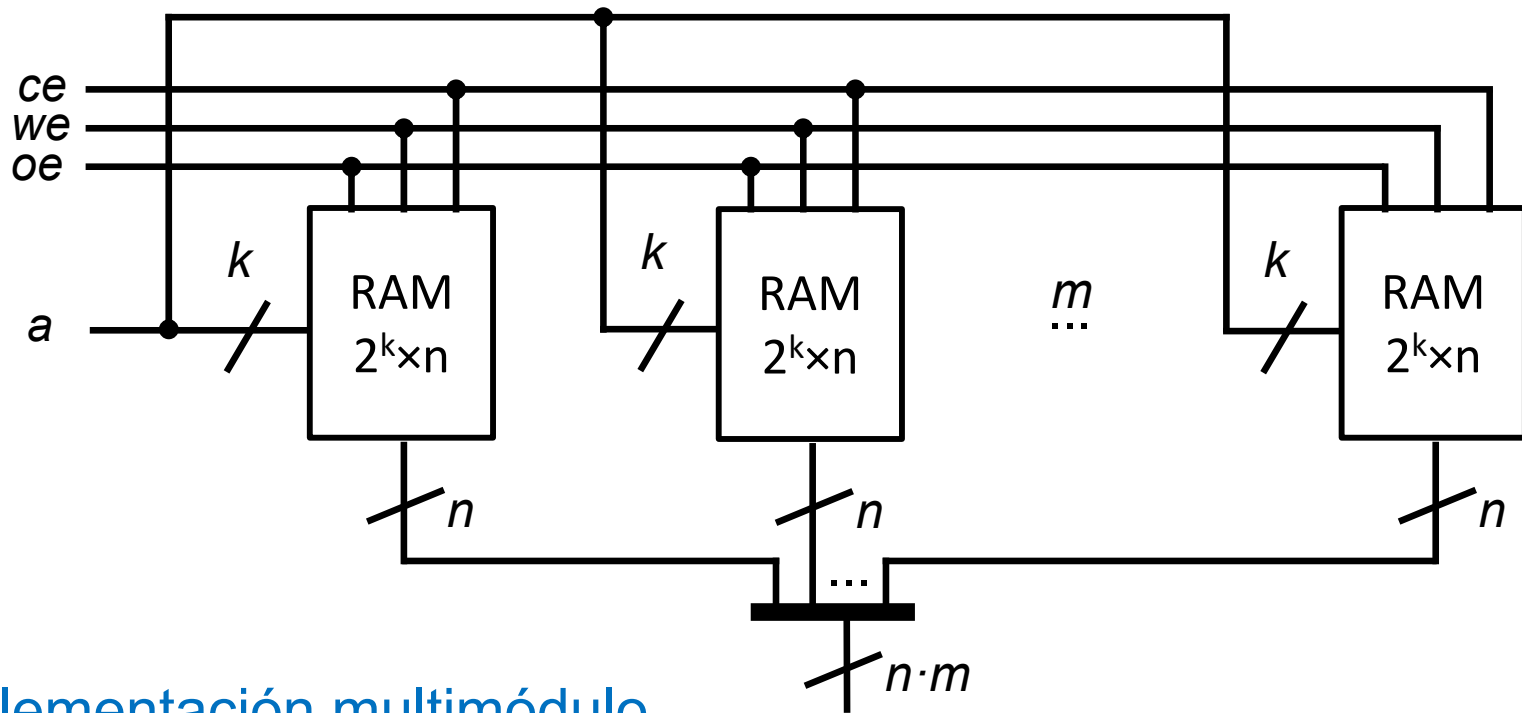
RAM (Random-Access Memory)





RAM (Random-Access Memory)

- Varias RAM se pueden componer para comportarse como una RAM de **mayor anchura de palabra**.



Implementación multimódulo

RAM $2^k \times (n \cdot m)$ usando m RAM $2^k \times n$

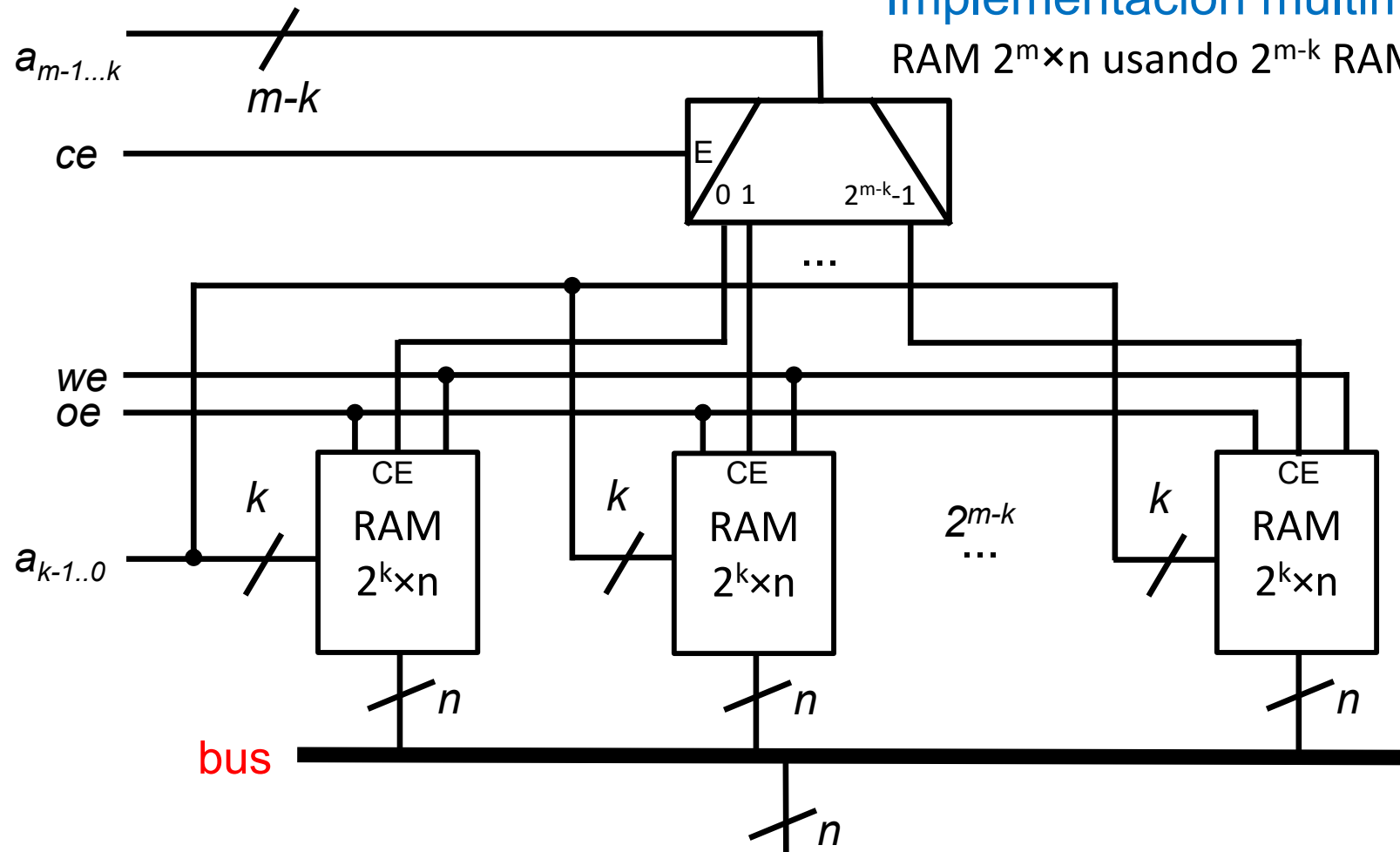


RAM (Random-Access Memory)

- Varias RAM se pueden componer para comportarse como una RAM de **mayor profundidad**.

Implementación multimódulo

RAM $2^m \times n$ usando 2^{m-k} RAM $2^k \times n$



Acerca de *Creative Commons*



■ Licencia CC (**Creative Commons**)

- Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:



Reconocimiento (*Attribution*):

En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.



No comercial (*Non commercial*):

La explotación de la obra queda limitada a usos no comerciales.



Compartir igual (*Share alike*):

La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: <https://creativecommons.org/licenses/by-nc-sa/4.0/>