



Tema 7:

Módulos secuenciales básicos

Fundamentos de computadores

José Manuel Mendías Cuadros

*Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid*

Contenidos



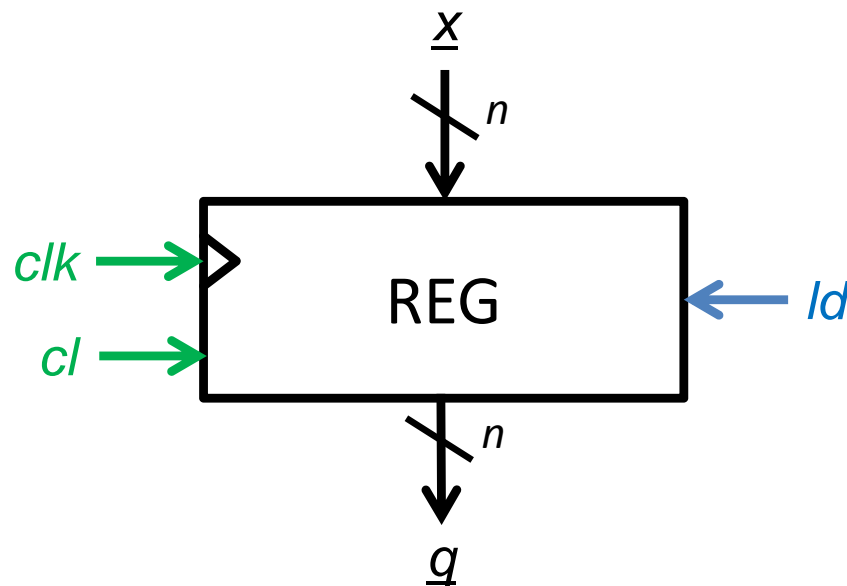
- ✓ Registro.
- ✓ Registro de desplazamiento.
- ✓ Contadores.
- ✓ Banco de registros.
- ✓ RAM (Random-Access Memory).
- ✓ Rutas de datos y controladores.

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

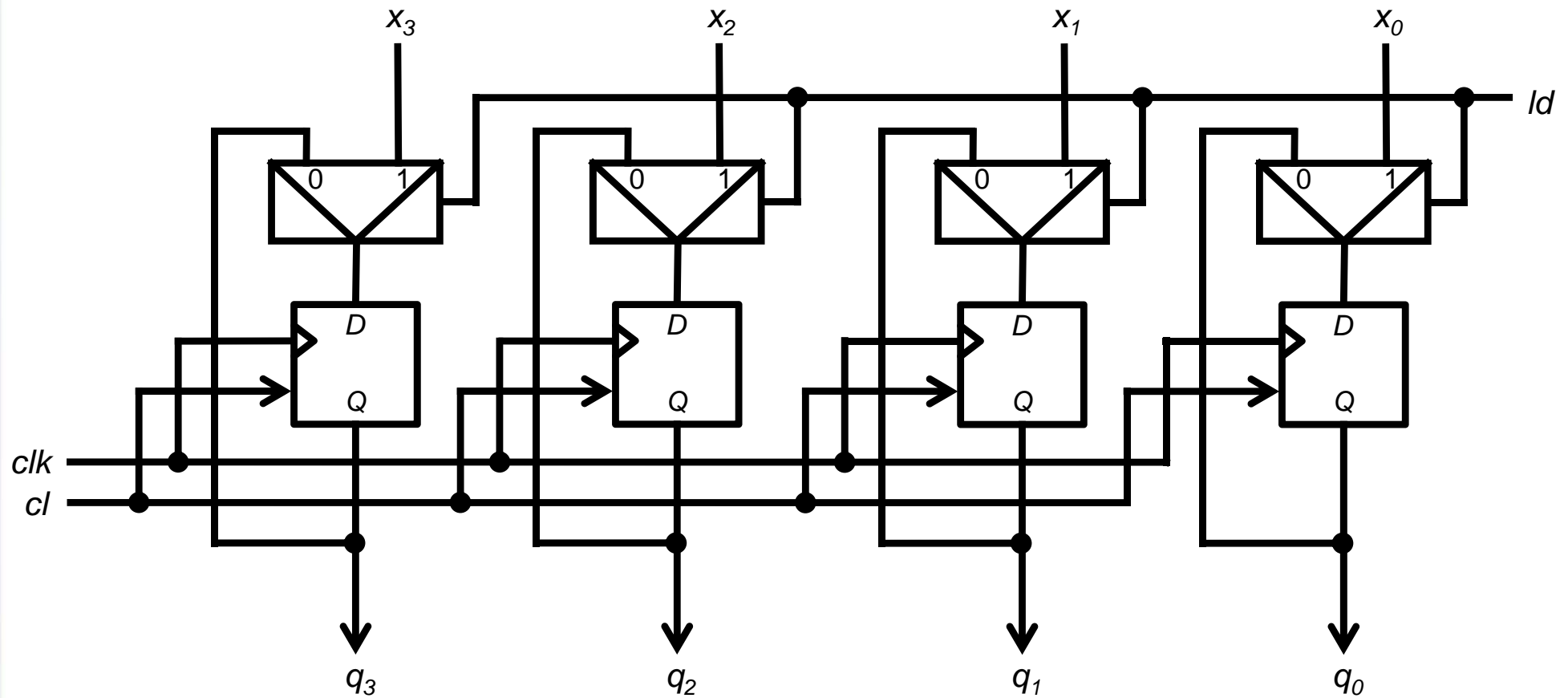
*si la entrada ld vale 1, almacena la entrada,
si no, conserva el valor almacenado*

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

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



Registro

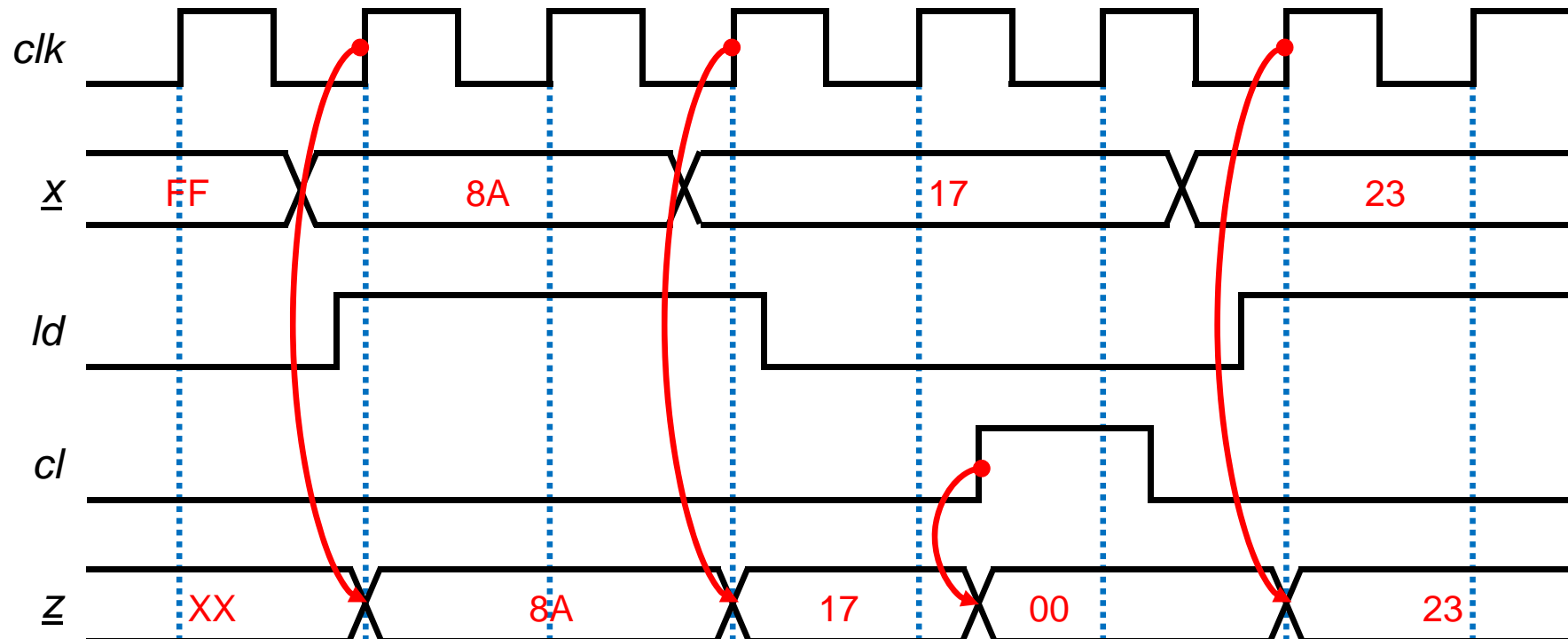
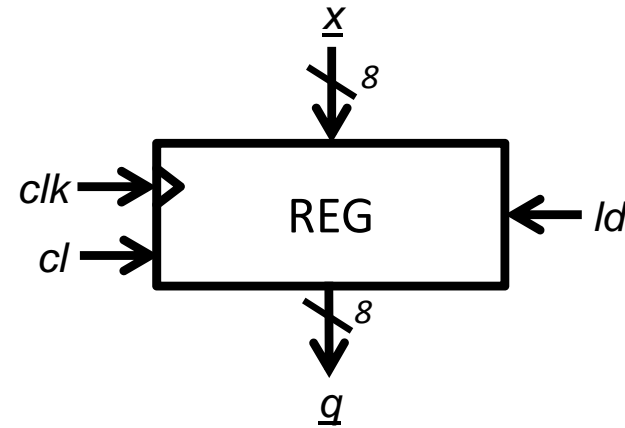


Implementación directa

Registro de 4 bits

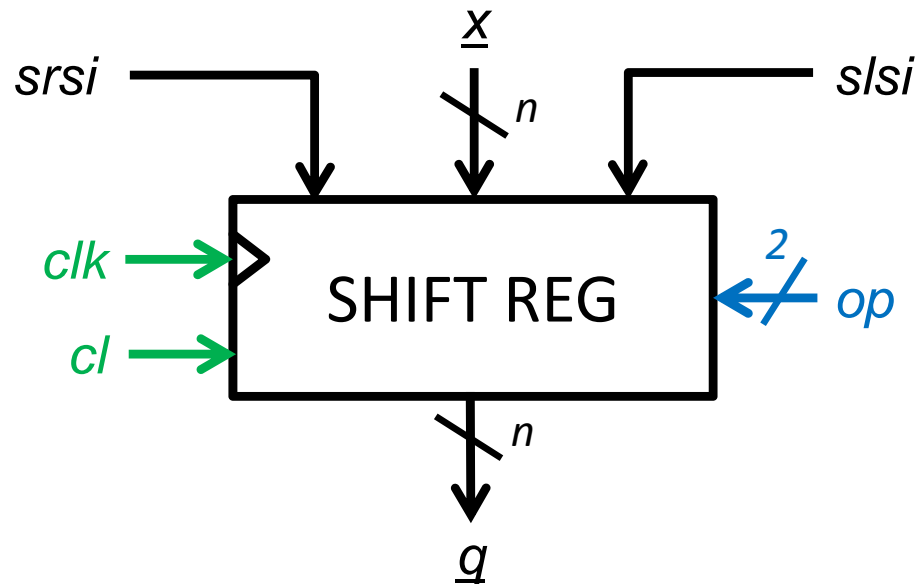


Registro





Registro de desplazamiento



Registro de desplazamiento de n bits

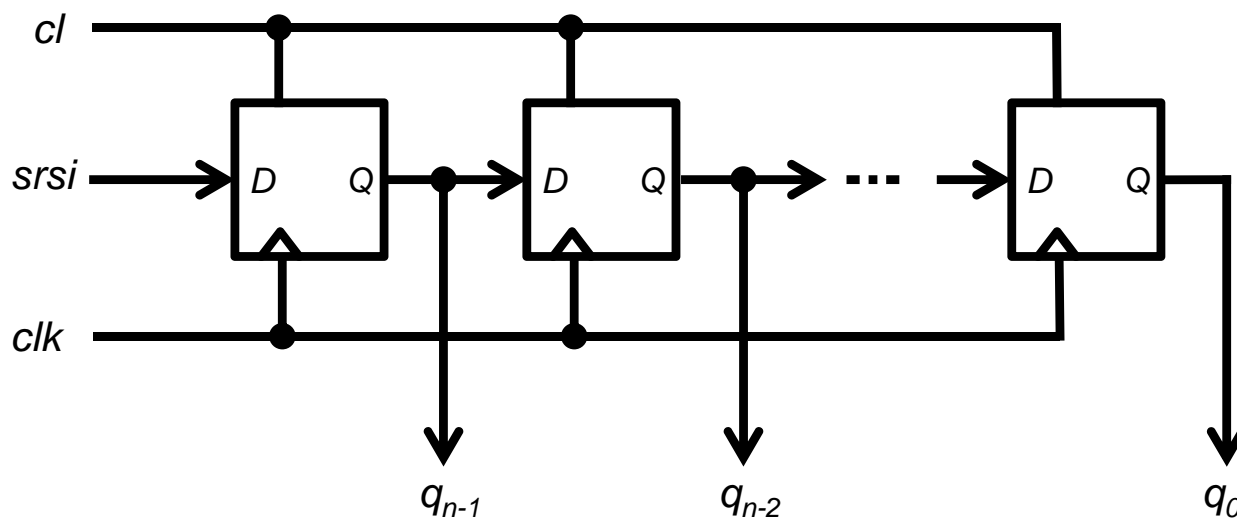
\underline{x}	1 entrada paralela de datos de n bits
\underline{q}	1 salida de datos de n bits
srsi	1 entrada serie de datos para desplazamiento a la derecha
slsi	1 entrada serie de datos para desplazamiento a la izquierda
\underline{op}	1 entrada de selección de operación
cl	1 entrada de inicialización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

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

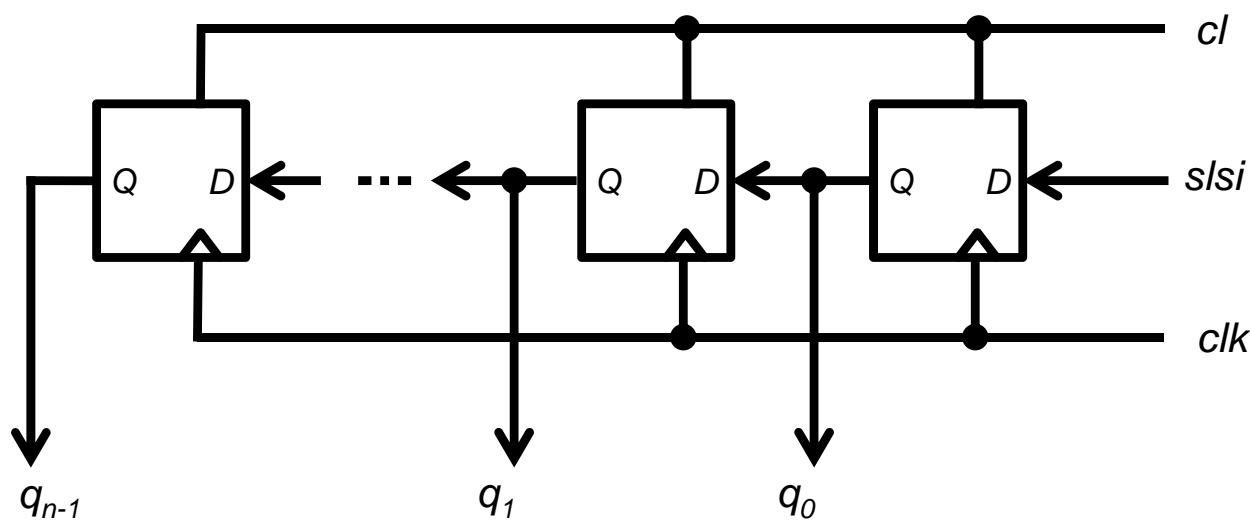
$$\text{si } cl(t) = 0 \Rightarrow \underline{q}(t+1) = \begin{cases} \underline{q}(t) & \text{si } op(t)=(00) \text{ sin cambio} \\ (srsi, q_{n-1}(t) \dots q_1(t)) & \text{si } op(t)=(01) \text{ deslaza a la derecha} \\ (q_{n-2}(t) \dots q_0(t), slsi) & \text{si } op(t)=(10) \text{ deslaza a la izquierda}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



Registro de desplazamiento



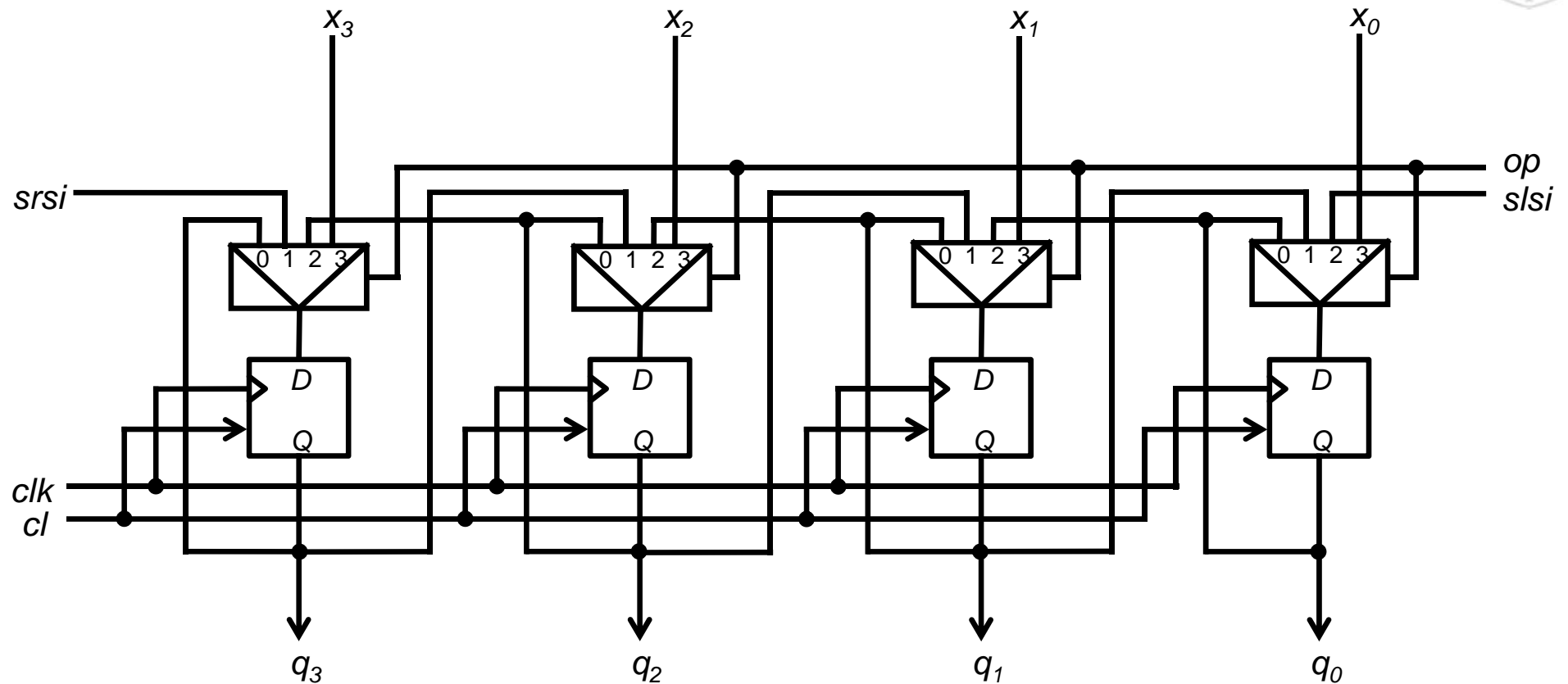
Desplazamiento
a la derecha



Desplazamiento
a la izquierda



Registro de desplazamiento



Implementación directa

Registro de desplazamiento de 4 bits

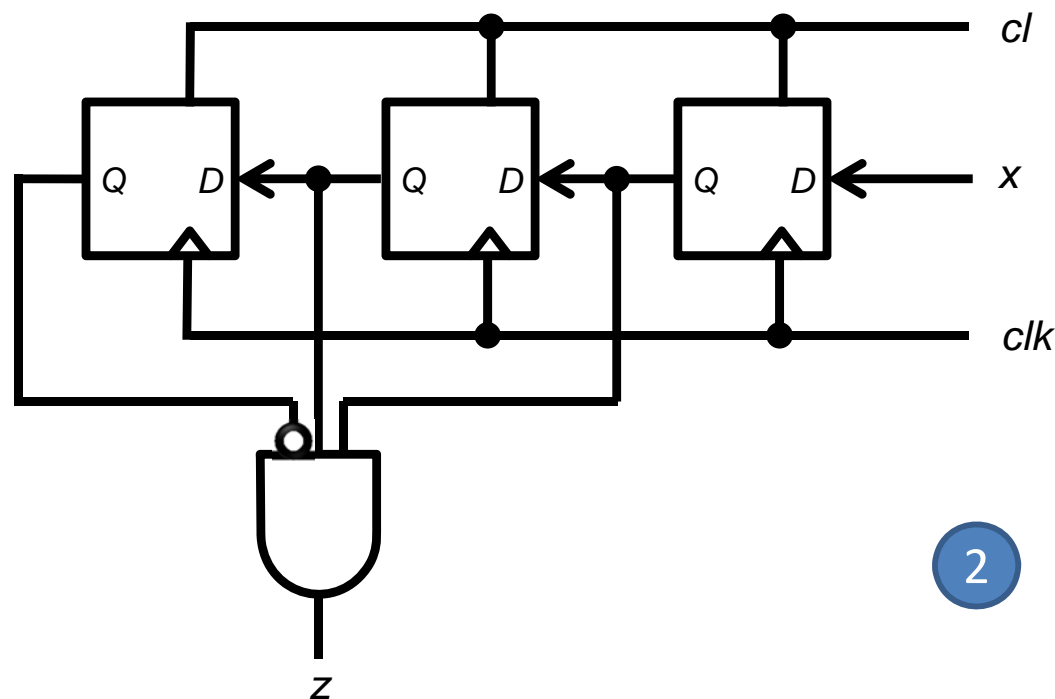


Registro de desplazamiento

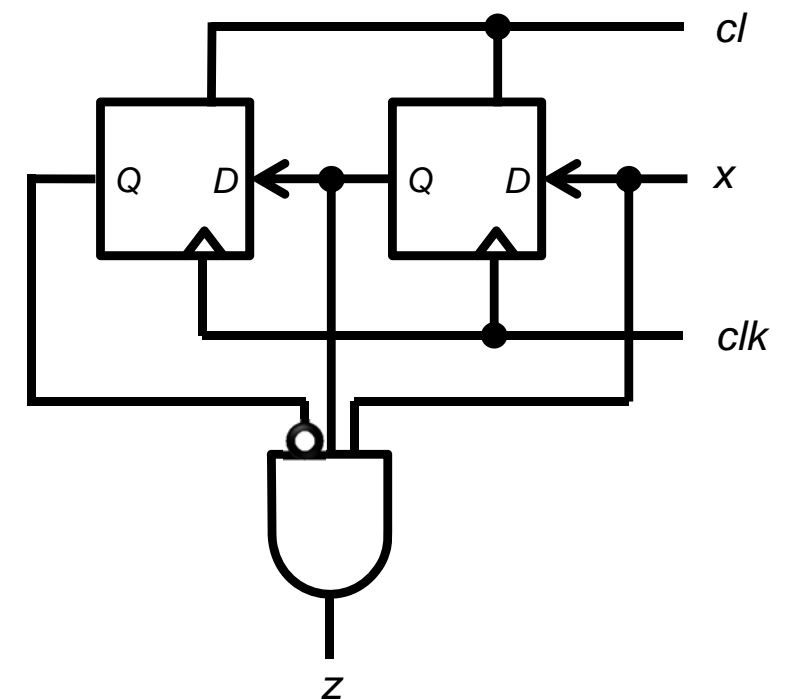
■ Aplicaciones al diseño:

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*





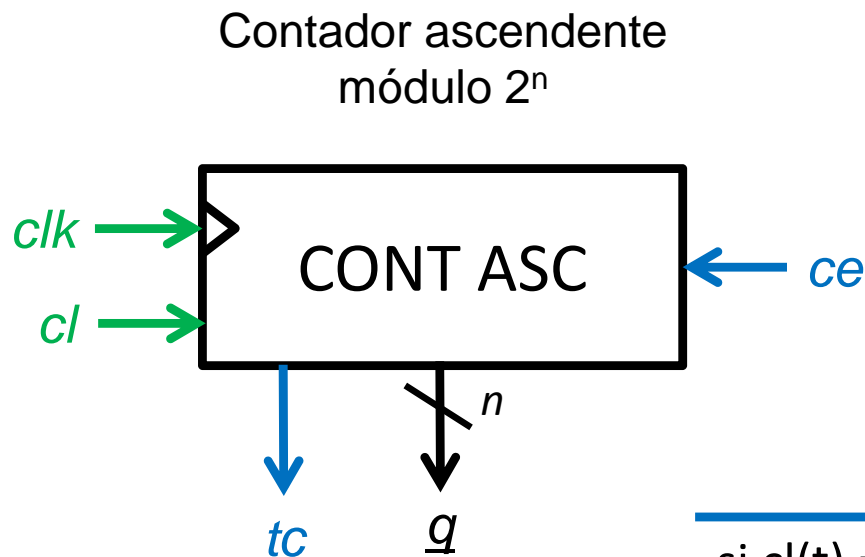
Contador ascendente

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

10



- 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

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

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

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

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



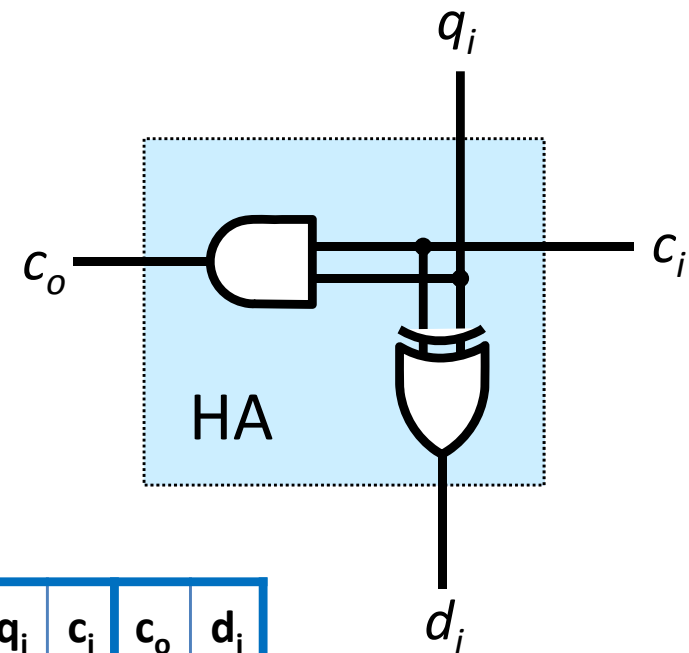
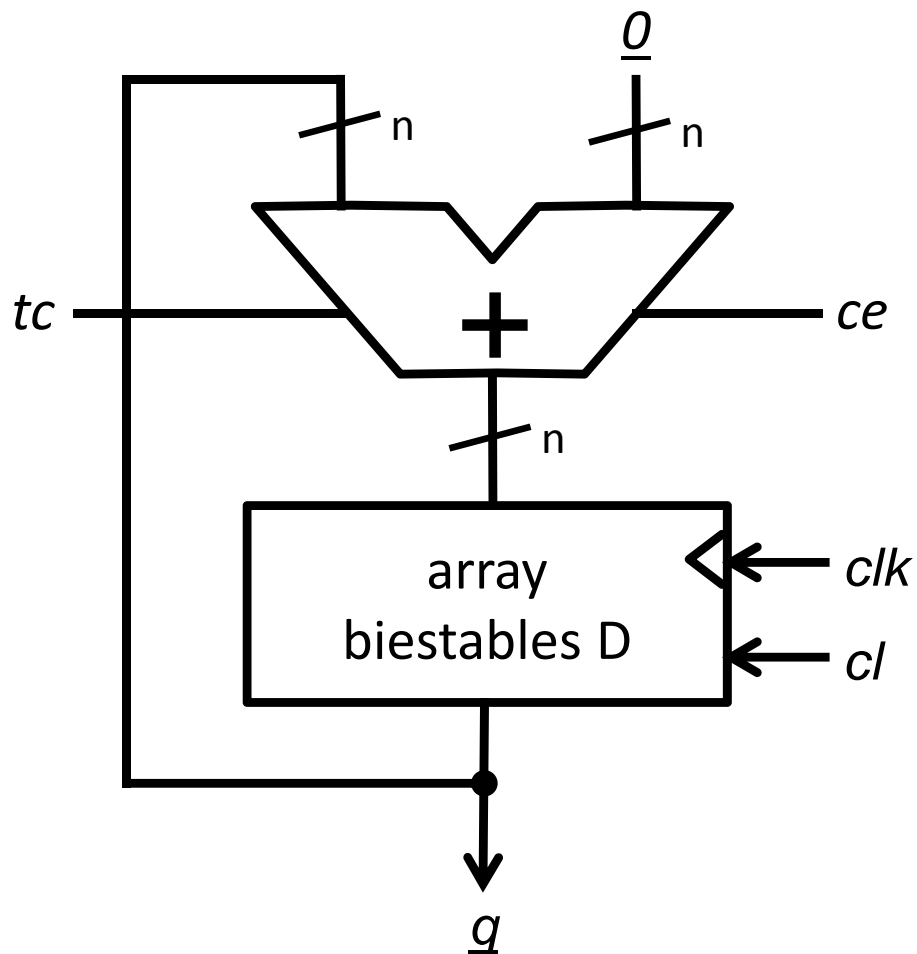
Contador ascendente

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

11



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 módulo 16



Contador descendente

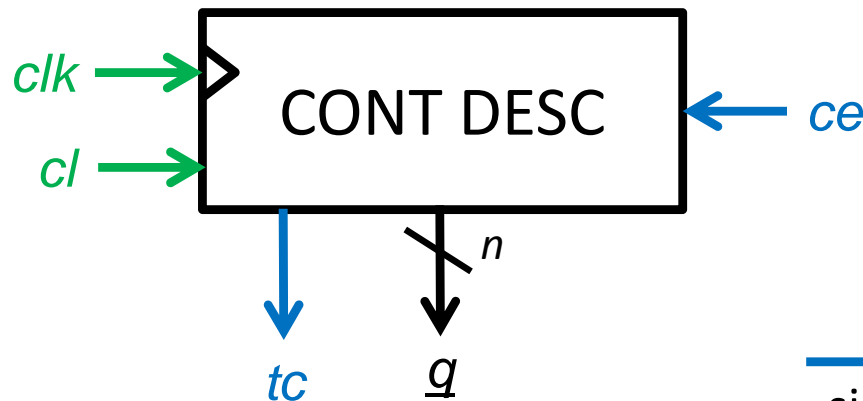
versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

13

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
-

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

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

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

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



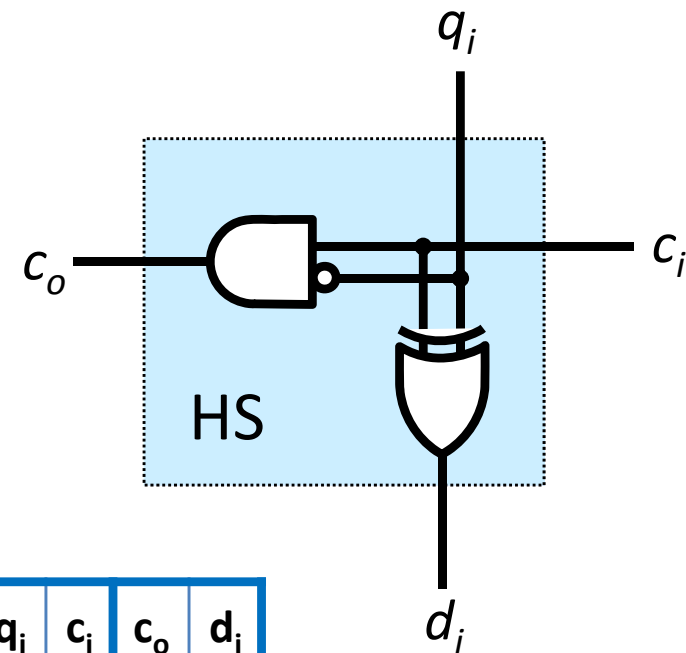
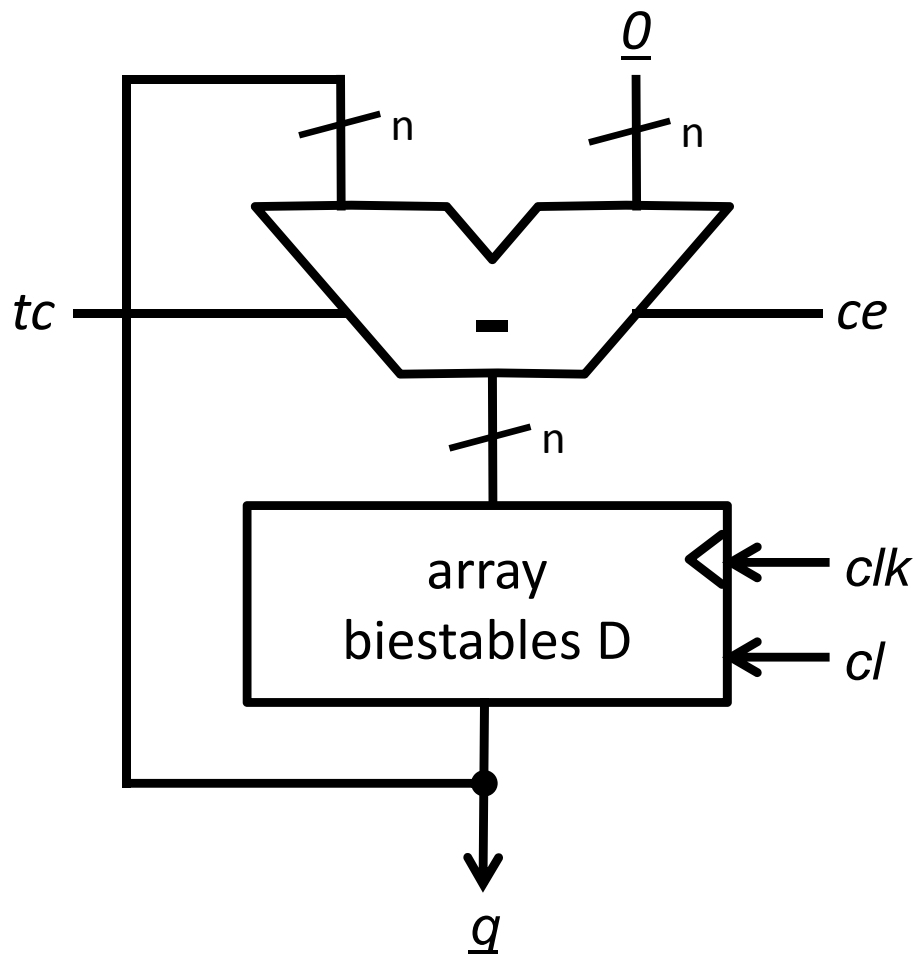
Contador descendente

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

14



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

$$d_i = q_i \oplus c_i$$

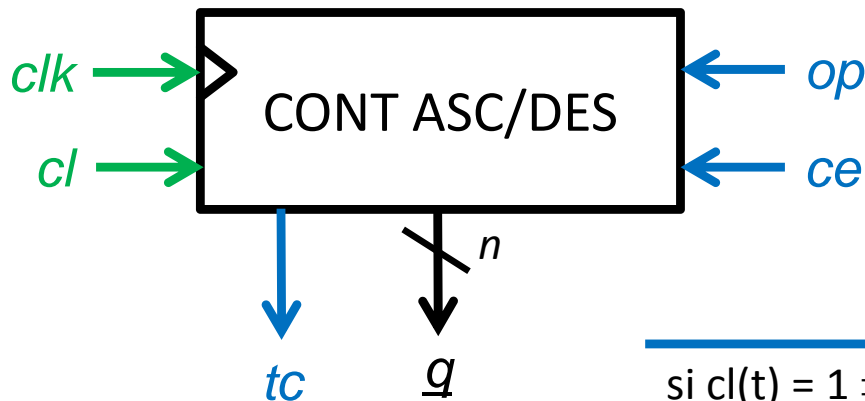
$$c_o = \overline{q_i} \cdot c_i$$

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

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

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

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

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

Contador ascendente/descendente

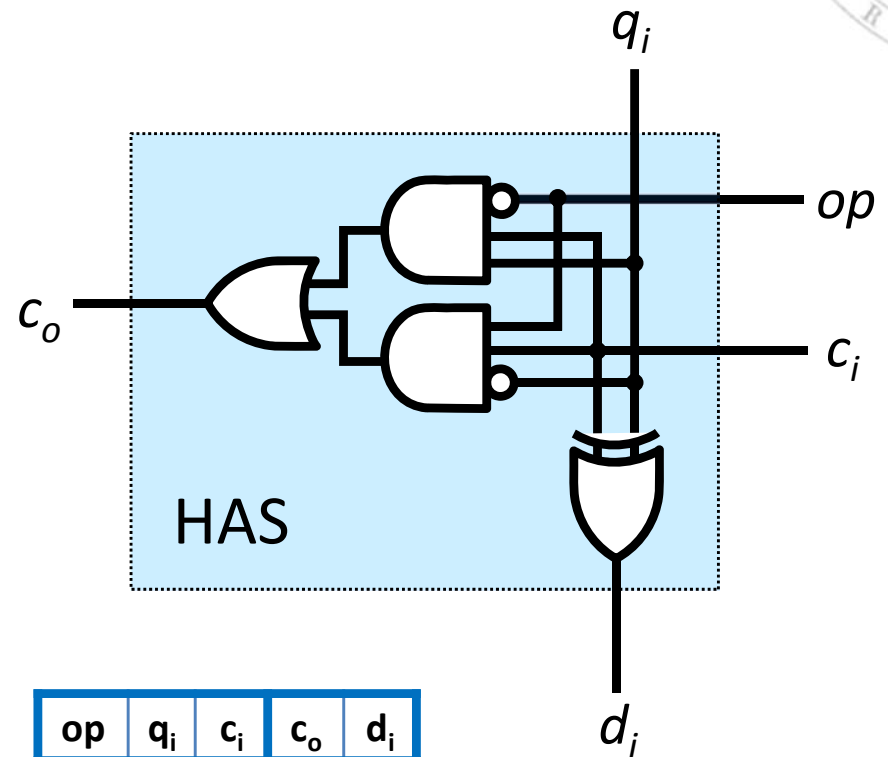
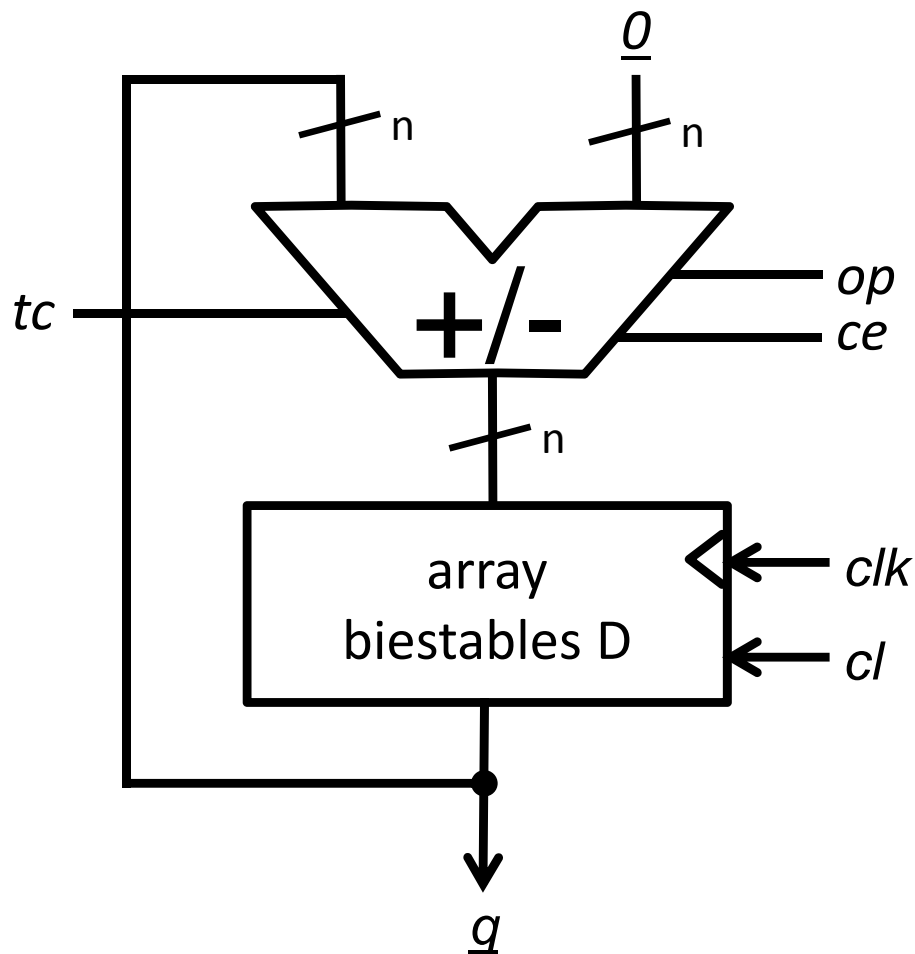


versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

17



op	q_i	c_i	c_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

$$d_i = q_i \oplus c_i$$

$$c_o = \overline{op} \cdot q_i \cdot c_i + op \cdot \overline{q_i} \cdot c_i$$



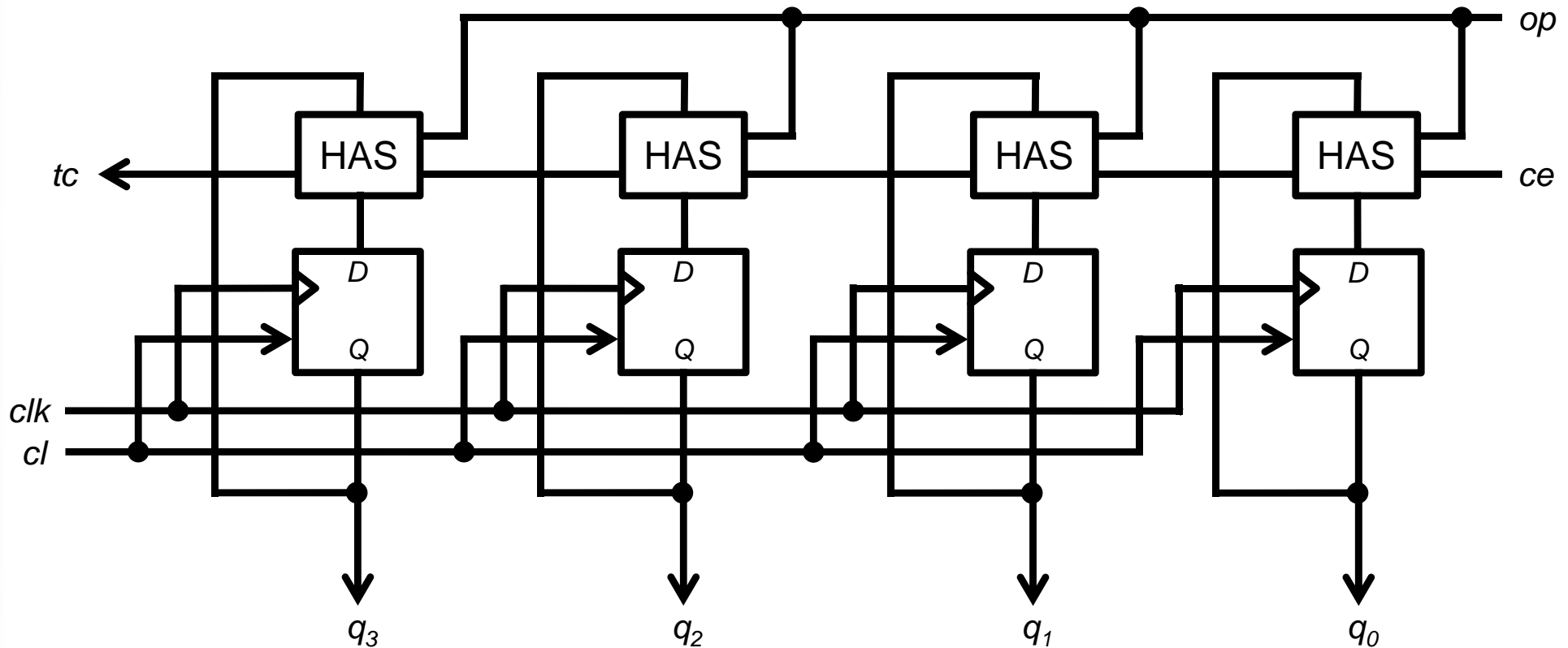
Contador ascendente/descendente

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

18

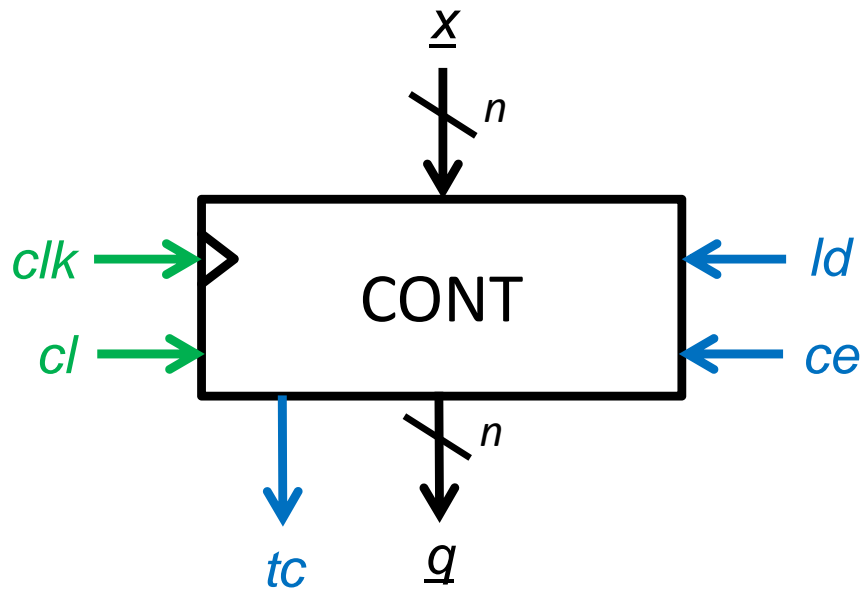


Implementación directa

Contador ascendente/descendente módulo 16



Contador con carga paralela



Contador ascendente con
carga paralela módulo 2^n

- \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

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

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

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

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



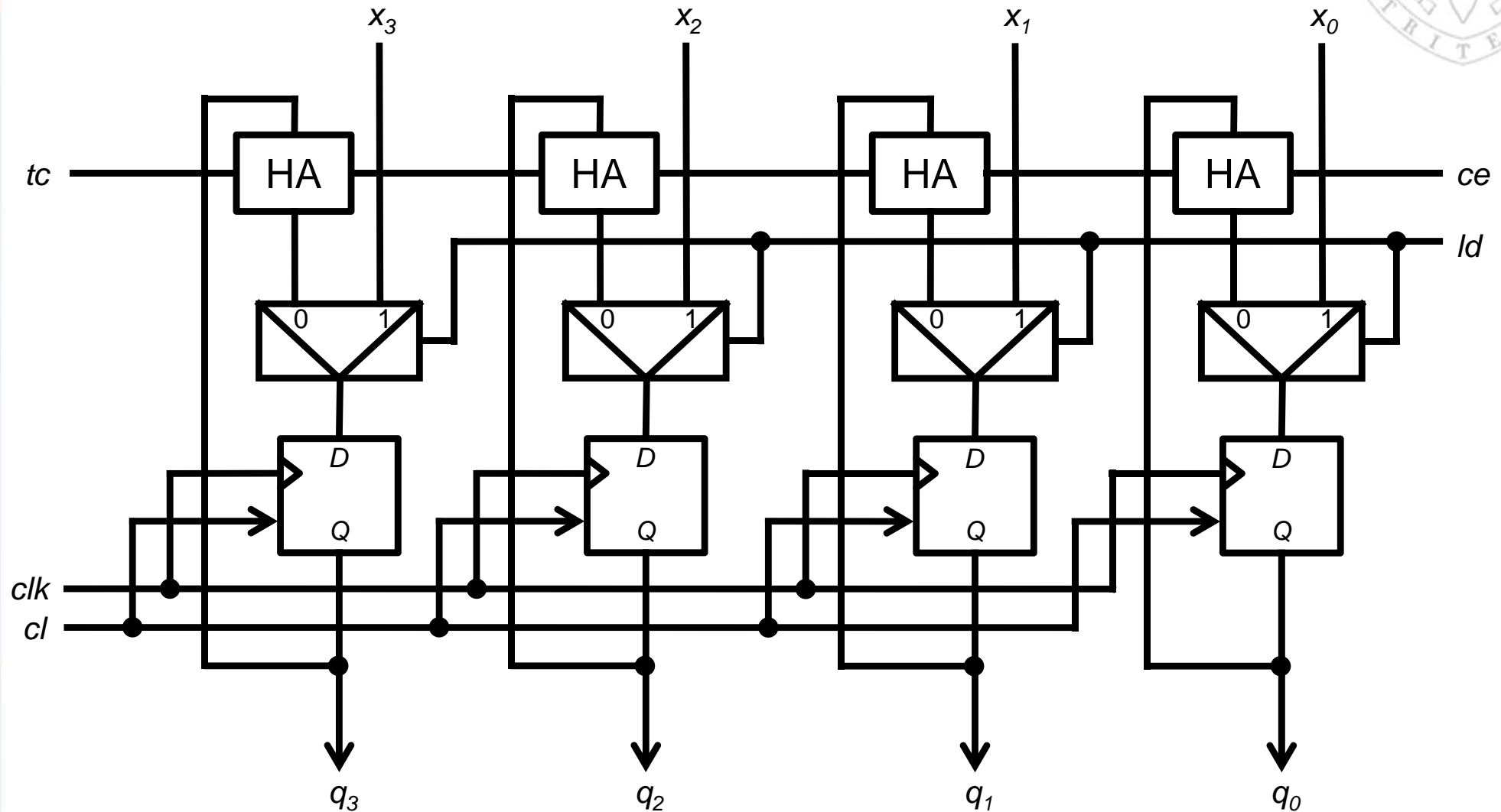
Contador con carga paralela

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

20



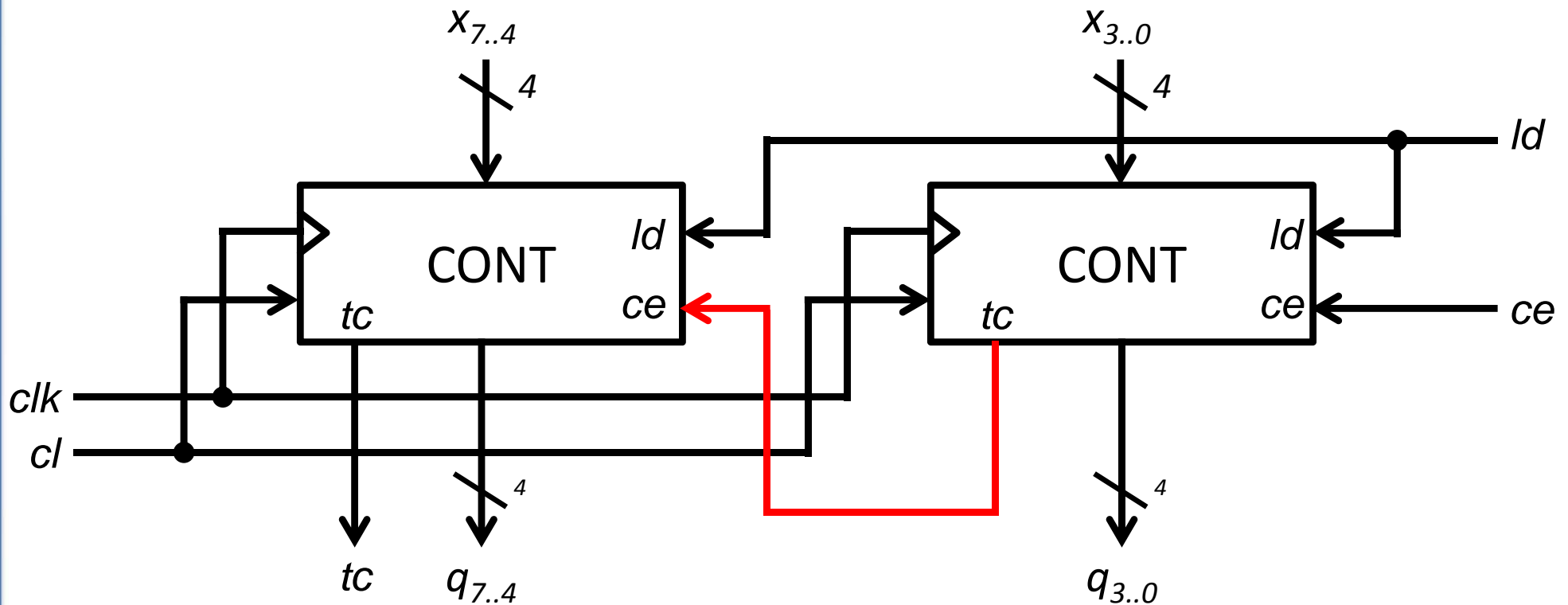
Implementación directa

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**.



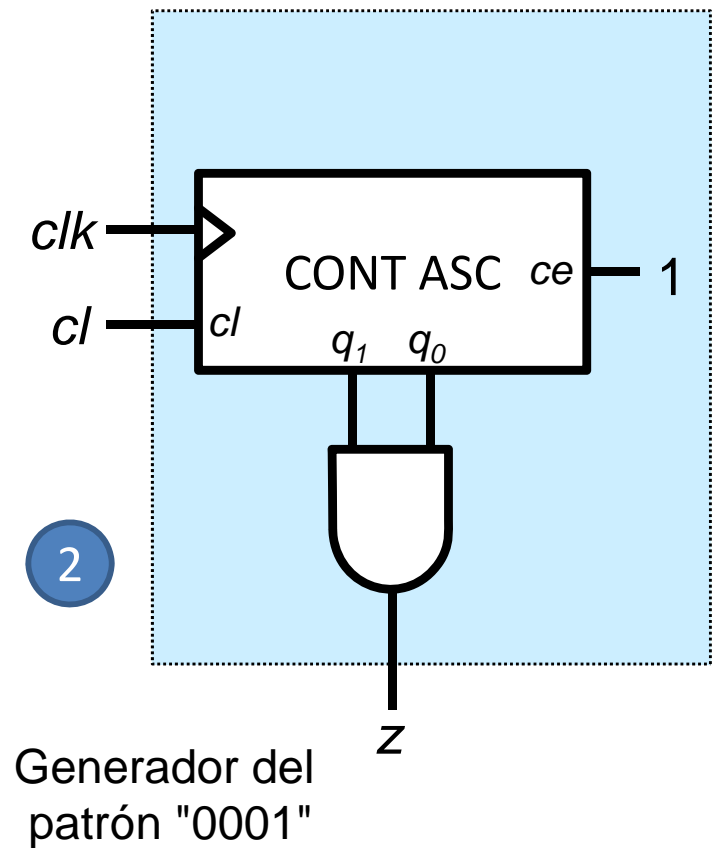
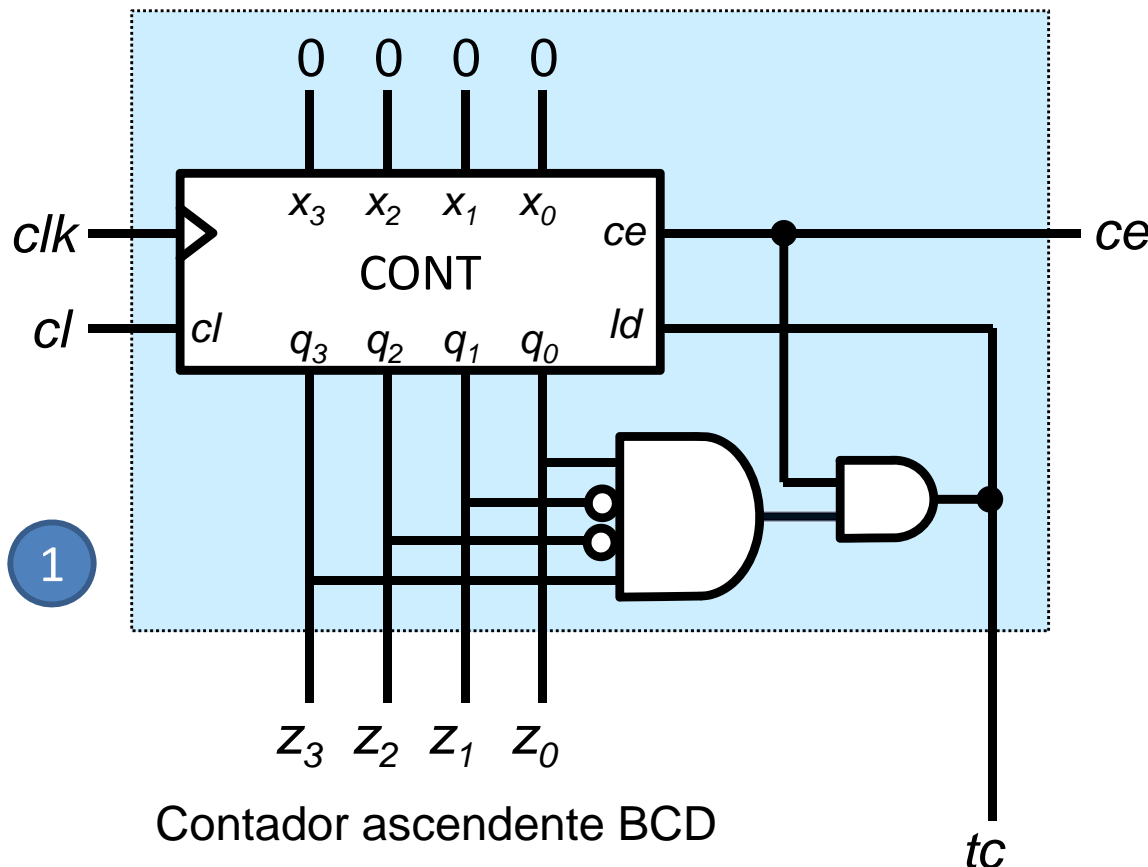
Implementación serie

Contador ascendente con carga paralela módulo 256



Contadores

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





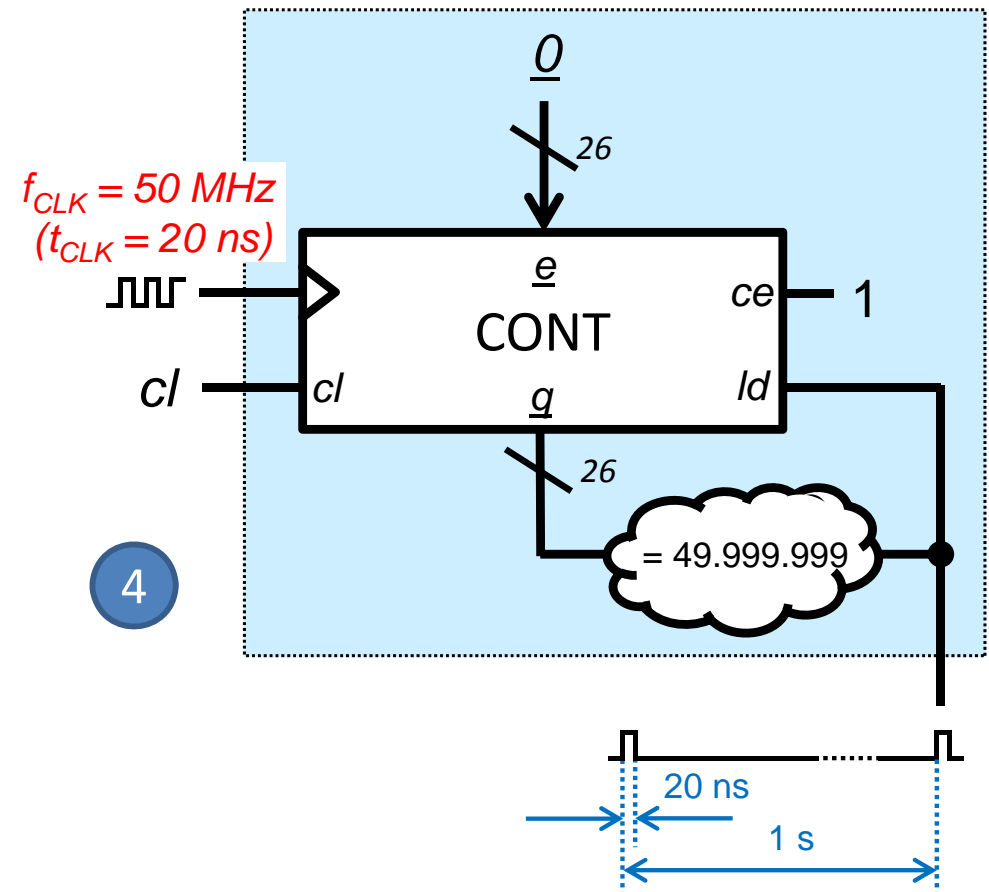
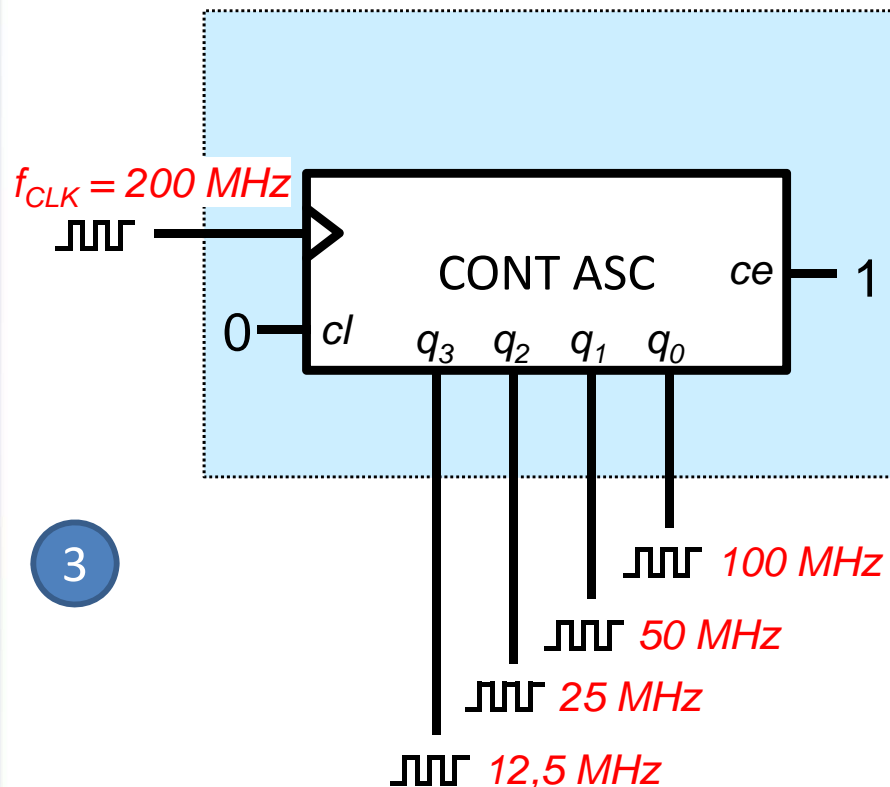
Contadores

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

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

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

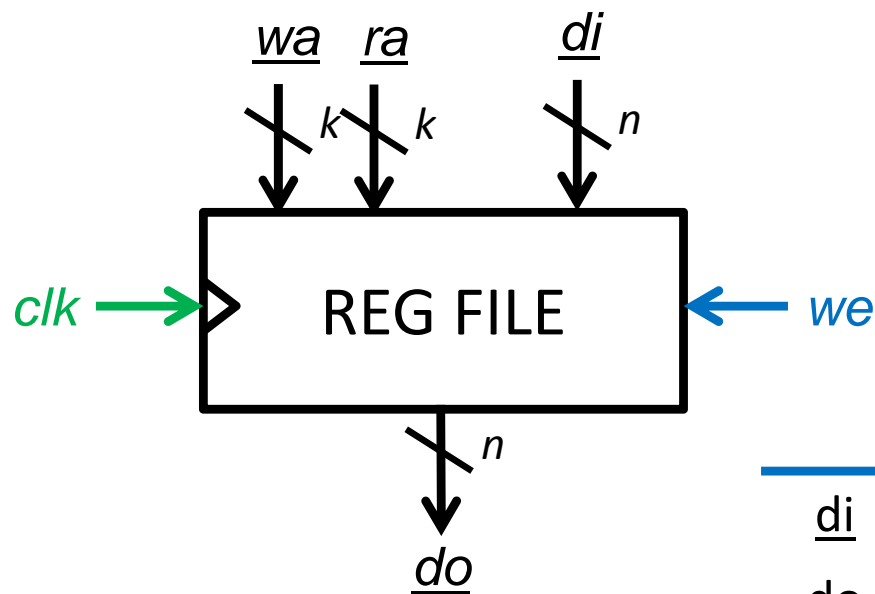
$$(1 \cdot 10^9 \text{ ns}) / (20 \text{ ns/ciclo}) = 50.000.000 \text{ ciclos}$$





Banco de registros

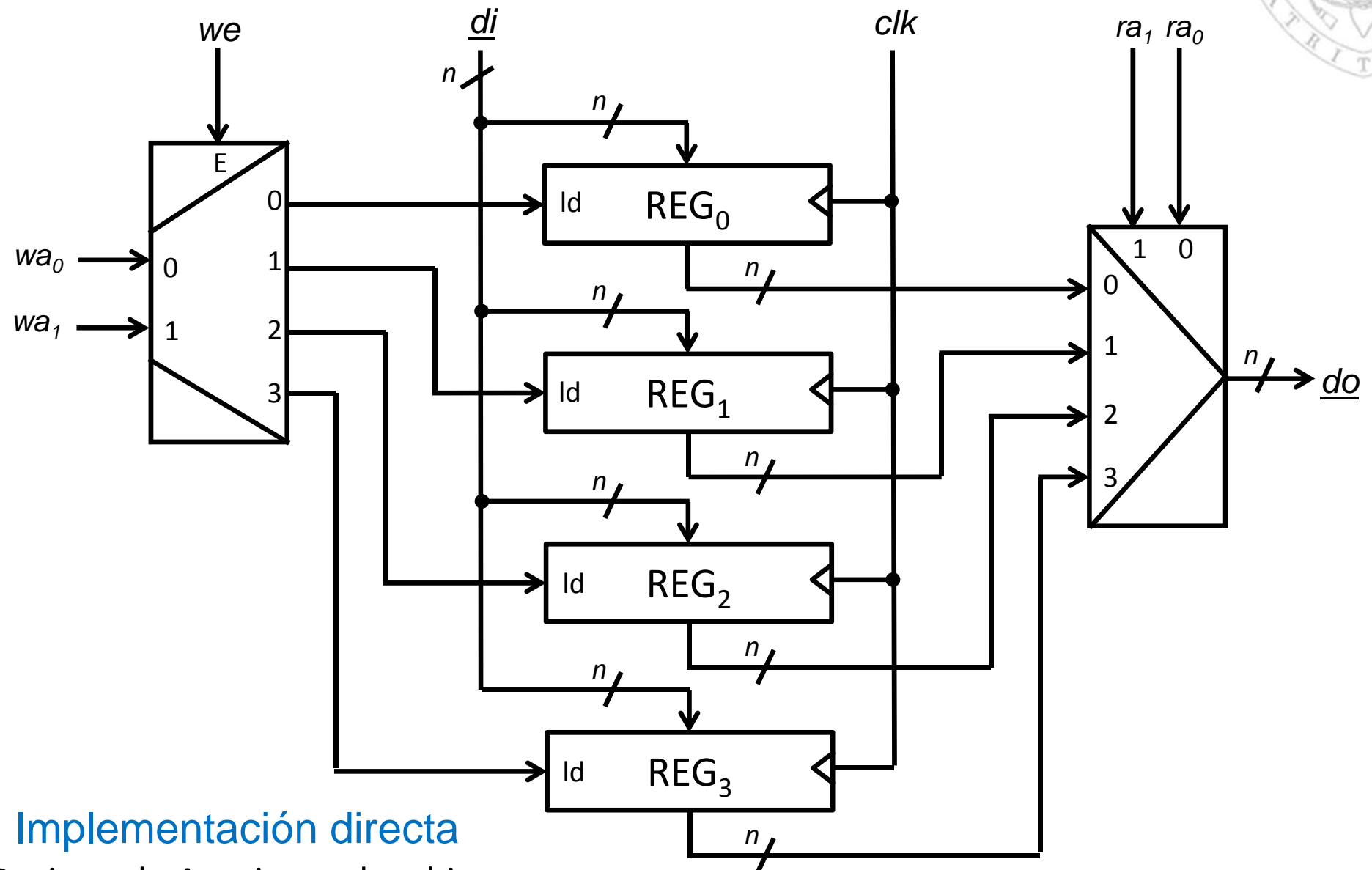
Banco de 2^k registros de n bits



<u>di</u>	1 entrada de datos de n bits
<u>do</u>	1 salida de datos de n bits
<u>wa</u>	1 entrada de dirección de escritura de k bits
<u>ra</u>	1 entrada 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

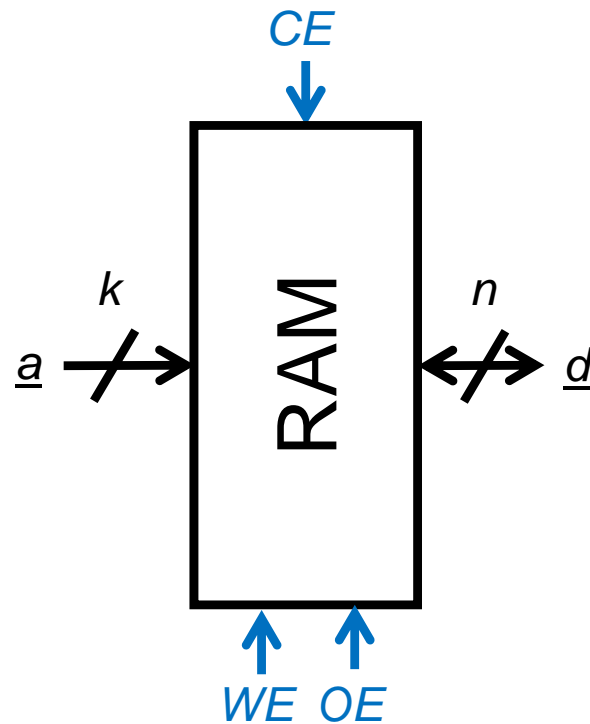


Implementación directa

Registro de 4 registros de n bits



RAM (Random-Access Memory)

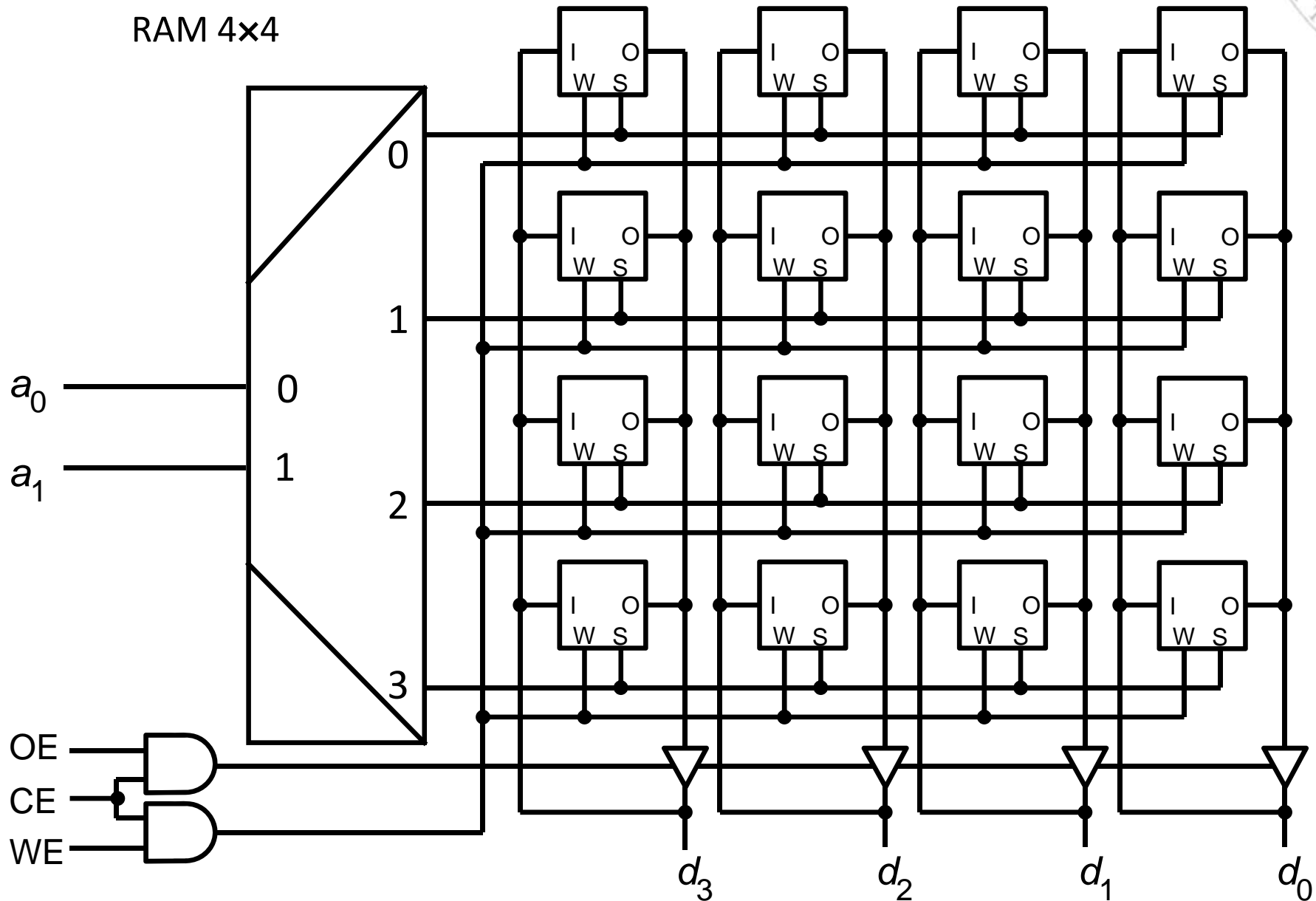


RAM $2^k \times n$
(2^k palabras de n bits)

\underline{d}	1 entrada/salida de datos de n bits
\underline{a}	1 entrada de dirección de k bits
\underline{CE}	1 entrada de capacitación
\underline{OE}	1 entrada de capacitación de lectura
\underline{WE}	1 entrada de capacitación de escritura

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

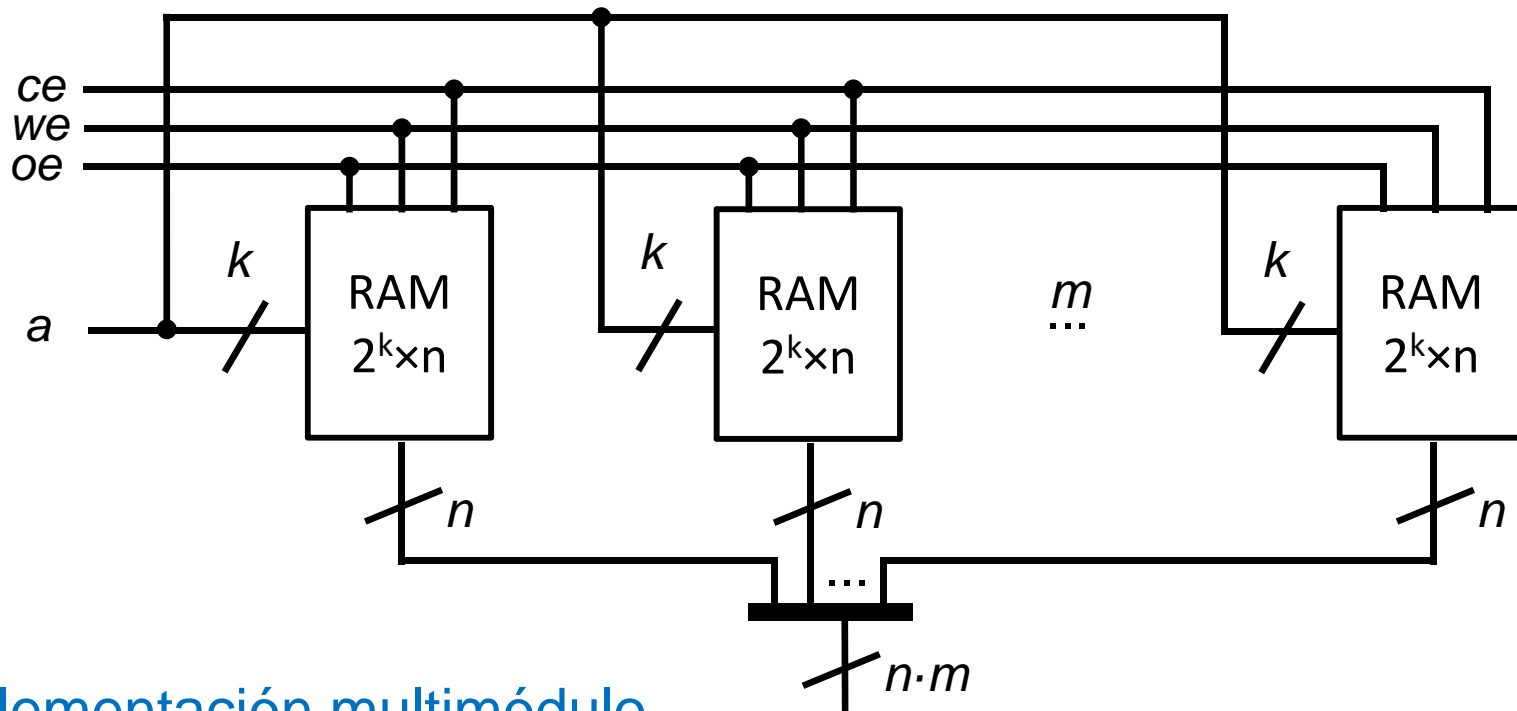
RAM (Random-Access Memory)





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 $2^k \times n$

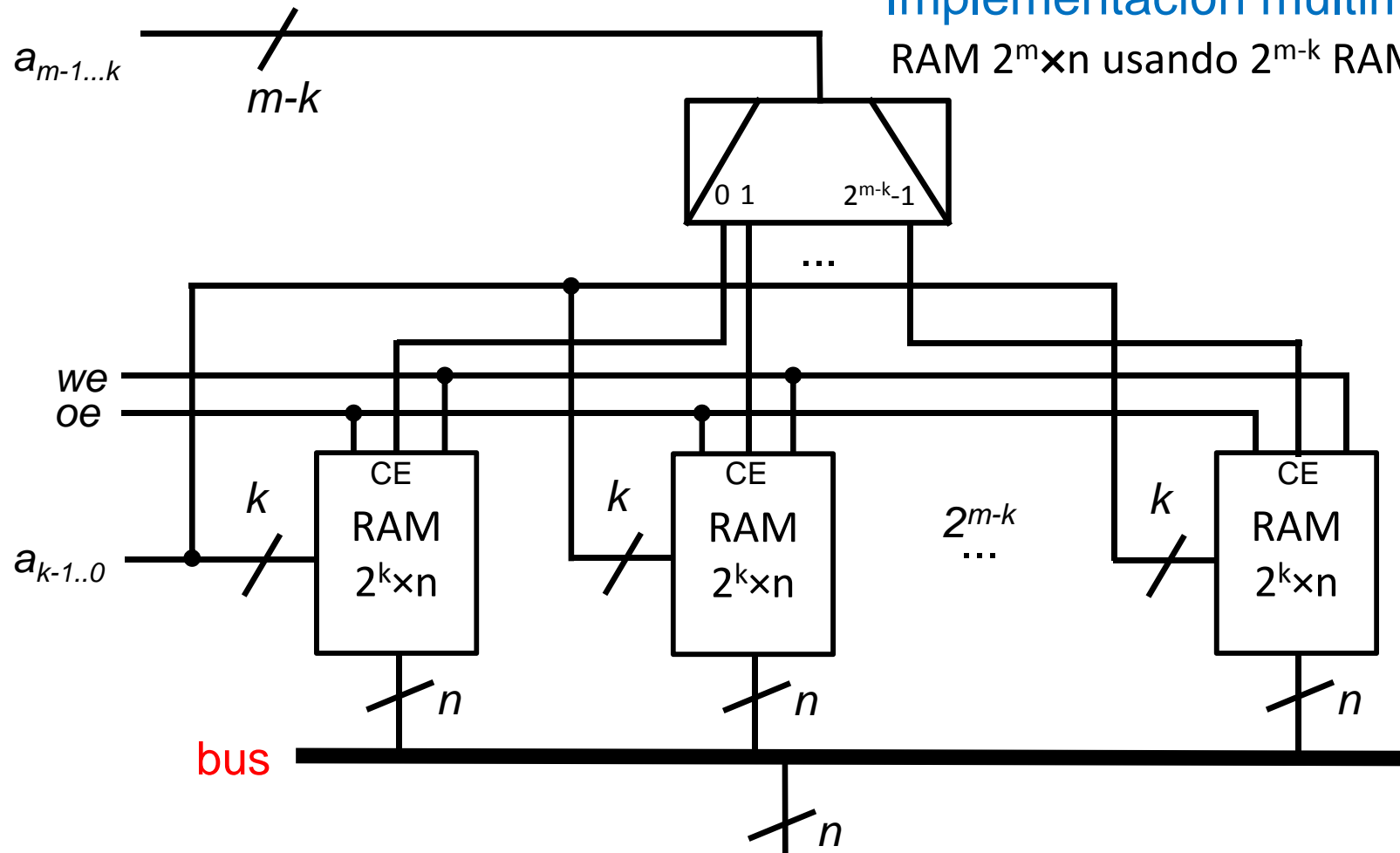


RAM (Random-Access Memory)

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

Implementación multimódulo

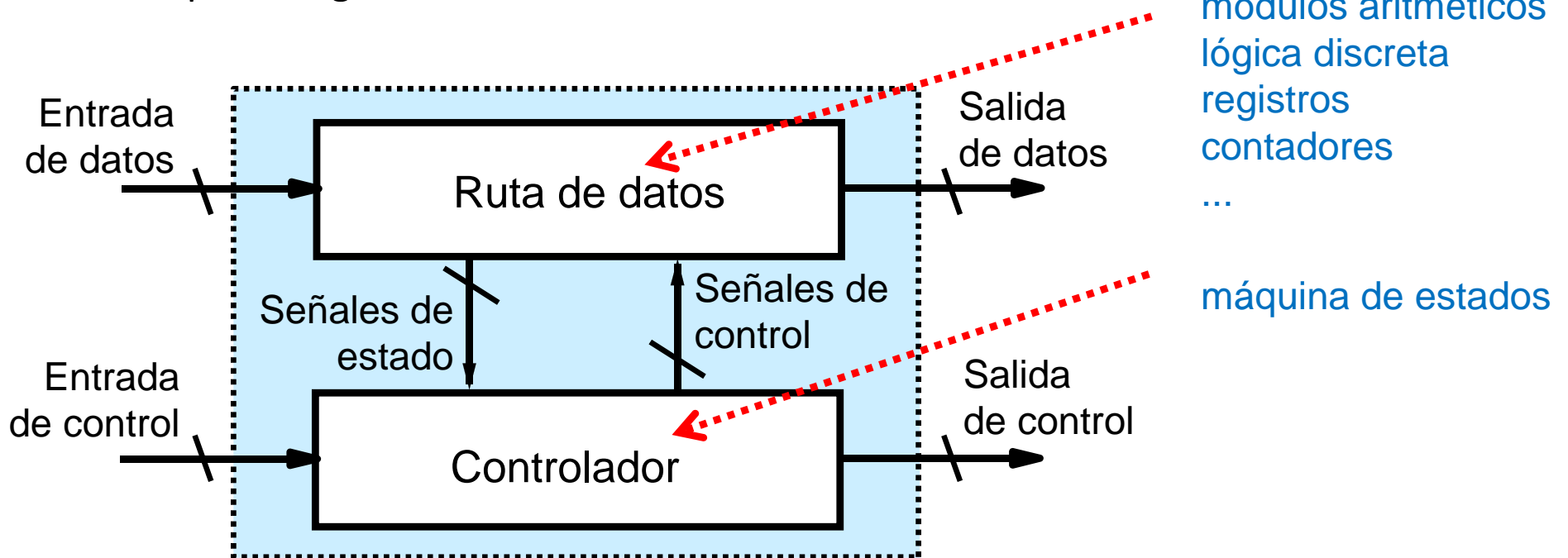
RAM $2^m \times n$ usando 2^{m-k} RAM $2^k \times n$





Rutas de datos y controladores

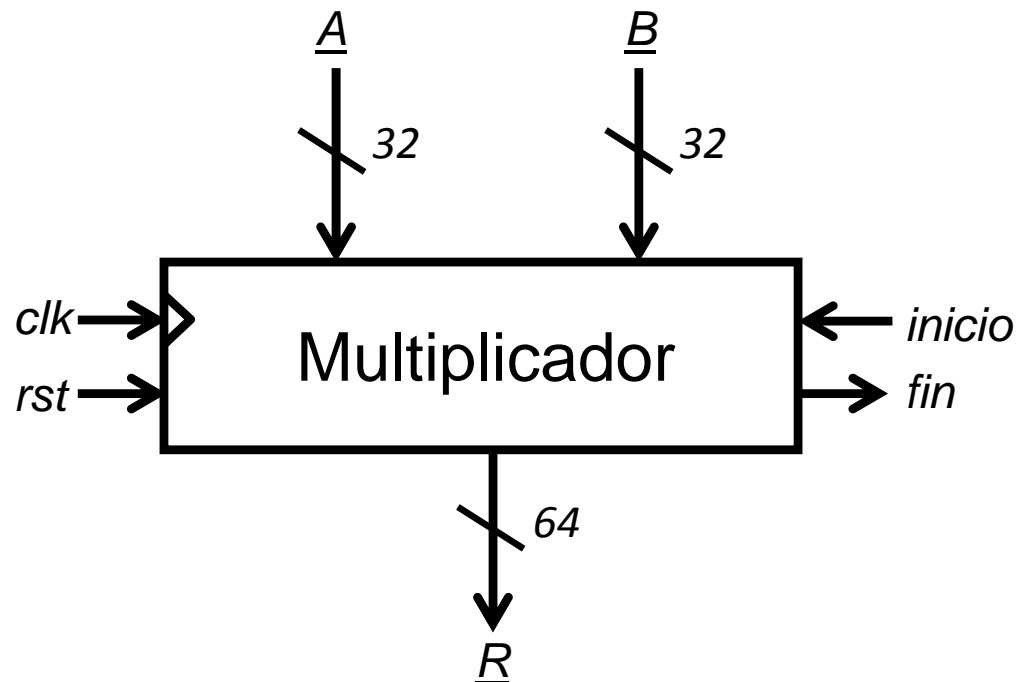
- Cuando un sistema secuencial realiza un **algoritmo complejo** **no es viable** especificarlo mediante un diagrama de estados.
- **Diseño algorítmico**: implementa un algoritmo en hardware.
 - **Ruta de datos**: realiza las operaciones y almacena resultados parciales.
 - **Controlador**: secuencia la realización de las operaciones según lo indicado por el algoritmo.





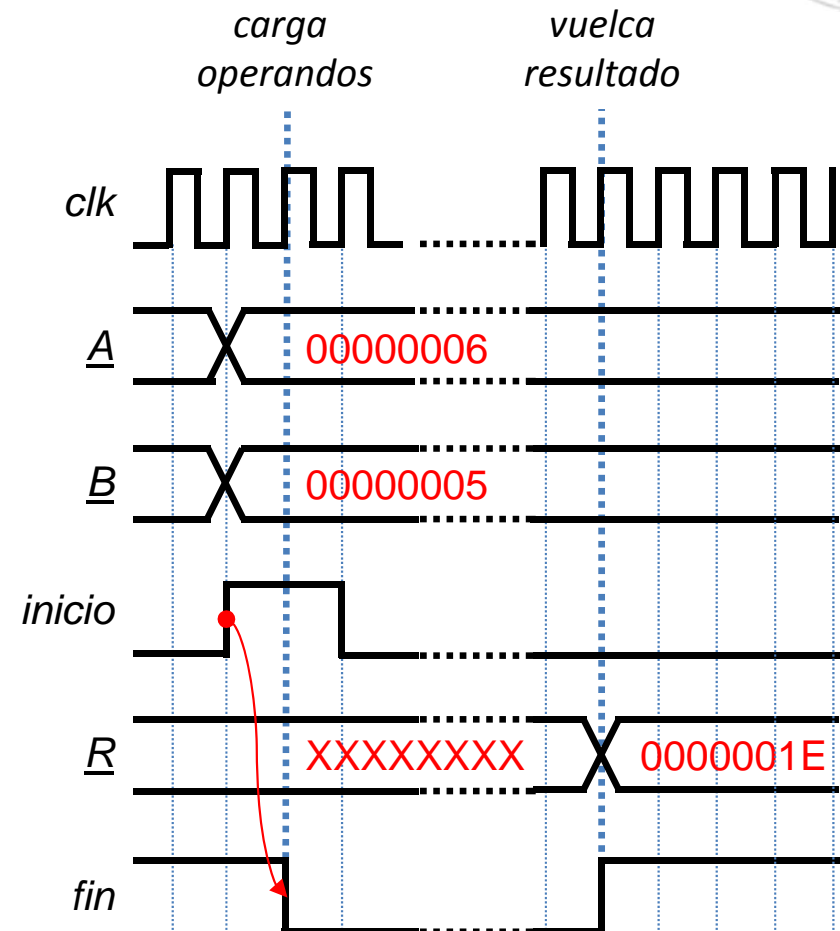
Rutas de datos y controladores

Especificación de alto nivel: interfaz



$\begin{array}{r} 110 \\ \times 101 \\ \hline 000110 \\ 000000 \\ + 011000 \\ \hline 011110 \end{array}$	$\begin{array}{r} a_2 a_1 a_0 \\ \times b_2 b_1 b_0 \\ \hline (000a_2 a_1 a_0) \times b_0 \\ (00a_2 a_1 a_0 0) \times b_1 \\ + (0a_2 a_1 a_0 00) \times b_2 \\ \hline r_5 r_4 r_3 r_2 r_1 r_0 \end{array}$
--	--

Algoritmo para **operandos de 3 bits**



1. Especificación de alto nivel



Rutas de datos y controladores

Especificación de alto nivel: algoritmo

Para operandos de 32 bits
el bucle sería de 0 to 31

```
begin
1.  A := Ain;
2.  B := Bin;
3.  R := 0;
   for C:=0 to 2 do begin
4.      if B0=1 then R := R+A;
5.      A := A << 1;
6.      B := B >> 1;
   end for;
7.  Rout := R;
end;
```

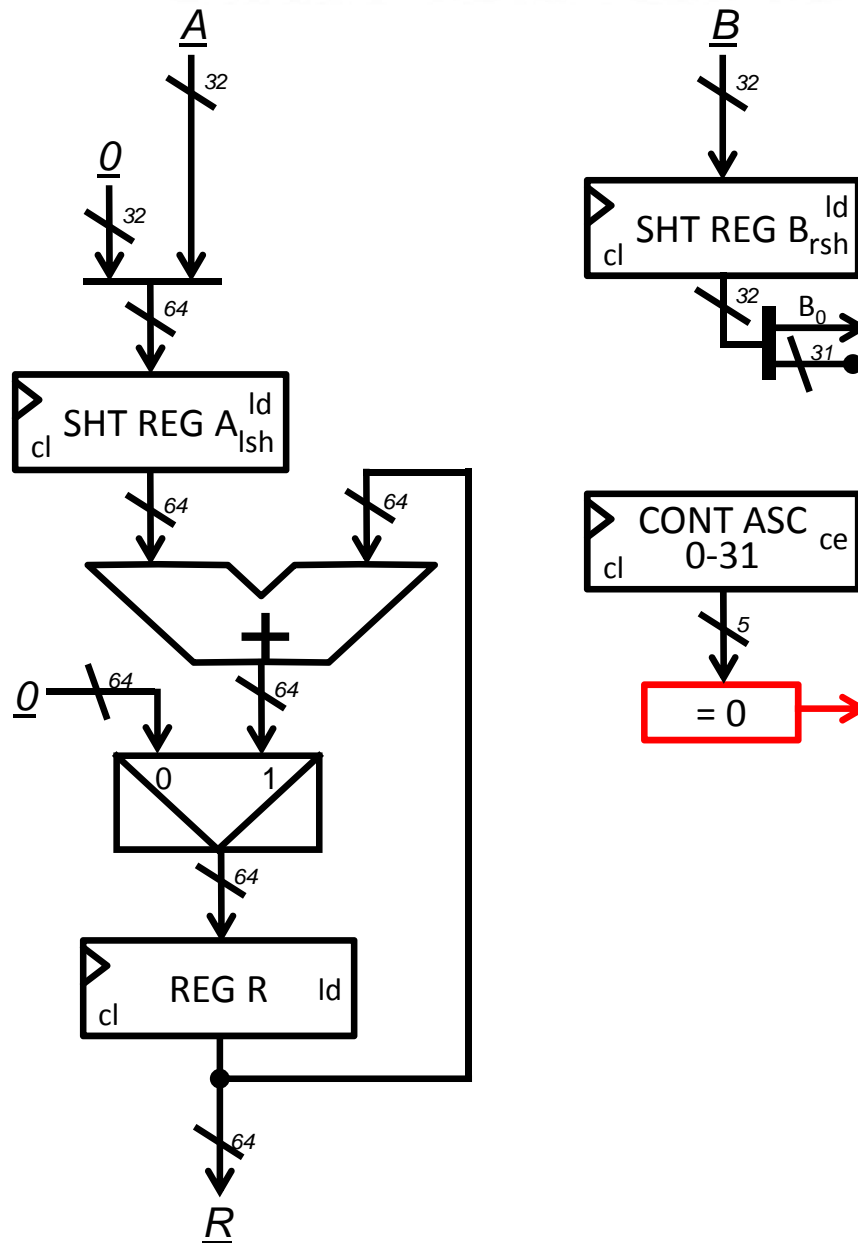
Algoritmo para operandos de 3 bits

	C	R ₅	R ₄	R ₃	R ₂	R ₁	R ₀	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀	B ₂	B ₁	B ₀
1.	-	-	-	-	-	-	-	0	0	0	1	1	0	-	-	-
2.	-	-	-	-	-	-	-	0	0	0	1	1	0	1	0	1
3.	-	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1
4.	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1
5.	0	0	0	0	1	1	0	0	0	1	1	0	0	1	0	1
6.	0	0	0	0	1	1	0	0	0	1	1	0	0	0	1	0
4.	1	0	0	0	1	1	0	0	0	1	1	0	0	0	1	0
5.	1	0	0	0	1	1	0	0	1	1	0	0	0	0	1	0
6.	1	0	0	0	1	1	0	0	1	1	0	0	0	0	0	1
4.	2	0	1	1	1	1	0	0	1	1	0	0	0	0	0	1
5.	2	0	1	1	1	1	0	1	1	0	0	0	0	0	0	1
6.	2	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0

2. Especificación del algoritmo

Rutas de datos y controladores

Diseño de la ruta de datos



```

begin
1.  A := Ain;
2.  B := Bin;
3.  R := 0;
   for C:=0 to 31 do begin
4.    if B0=1 then R := R+A;
5.    A := A << 1;
6.    B := B >> 1;
   end for;
7.  Rout := R;
end;

```

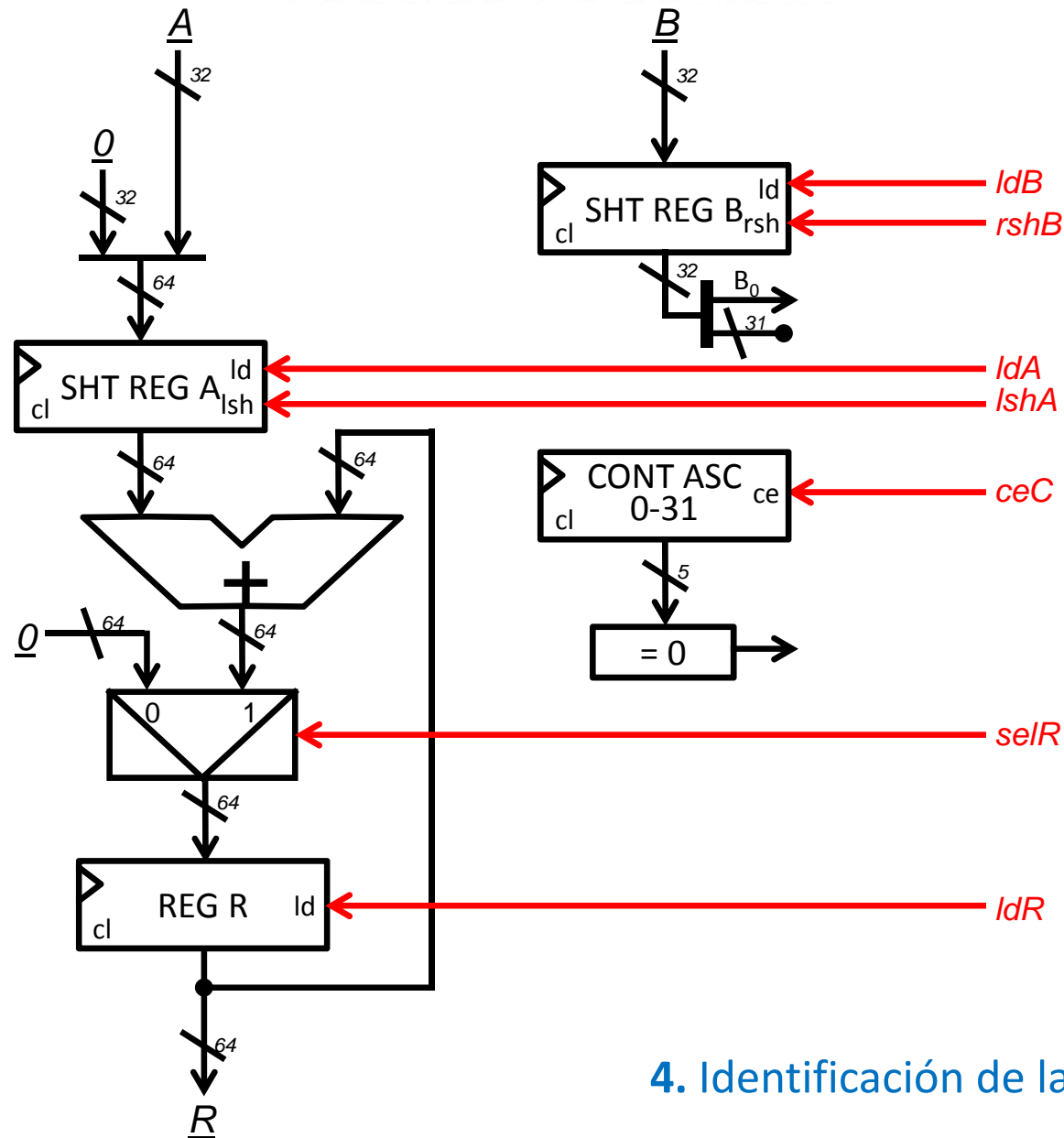
Algoritmo para
operandos de 32 bits



3. Diseño de la ruta de datos

Rutas de datos y controladores

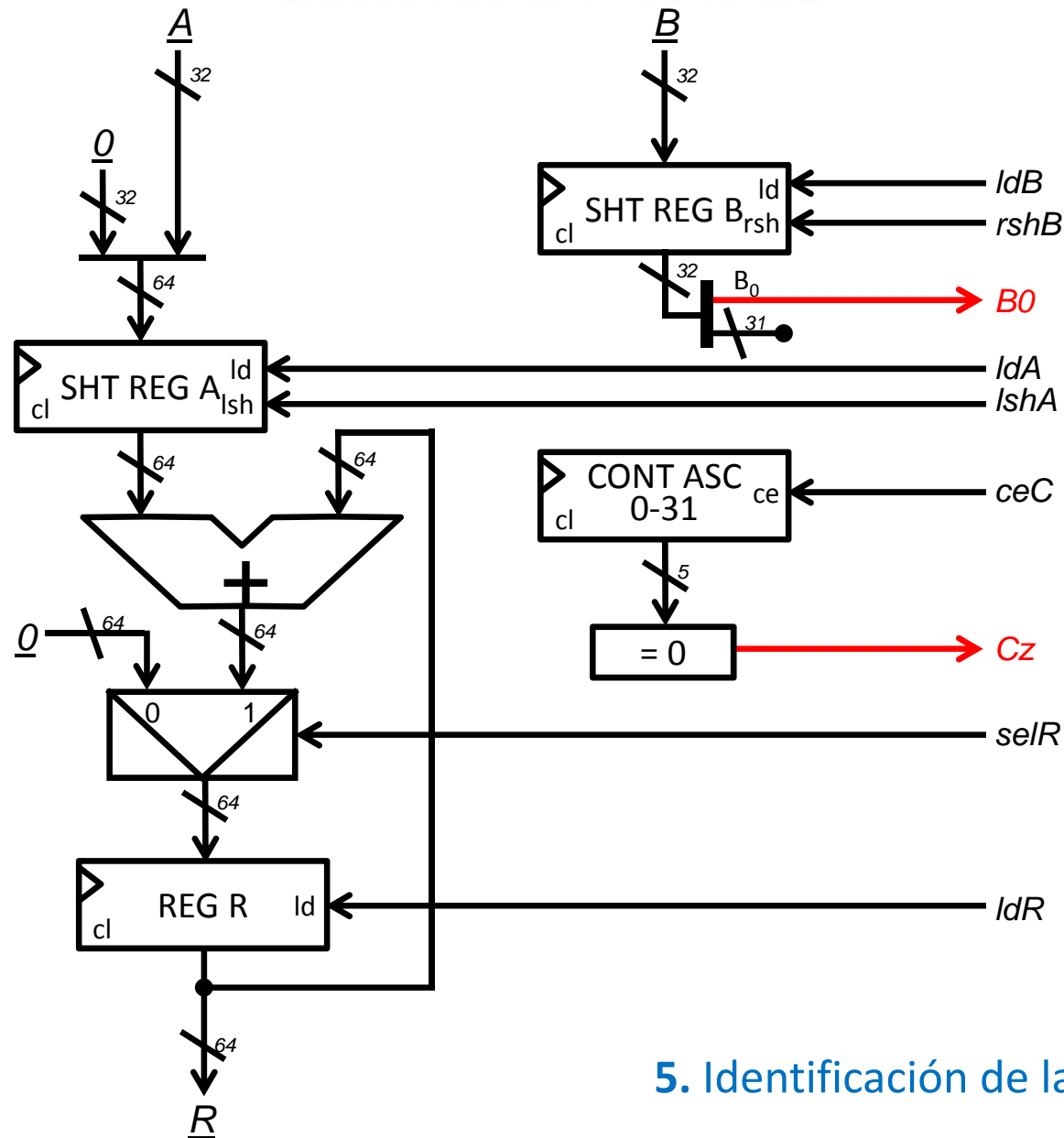
Señales de control



4. Identificación de las señales de control

Rutas de datos y controladores

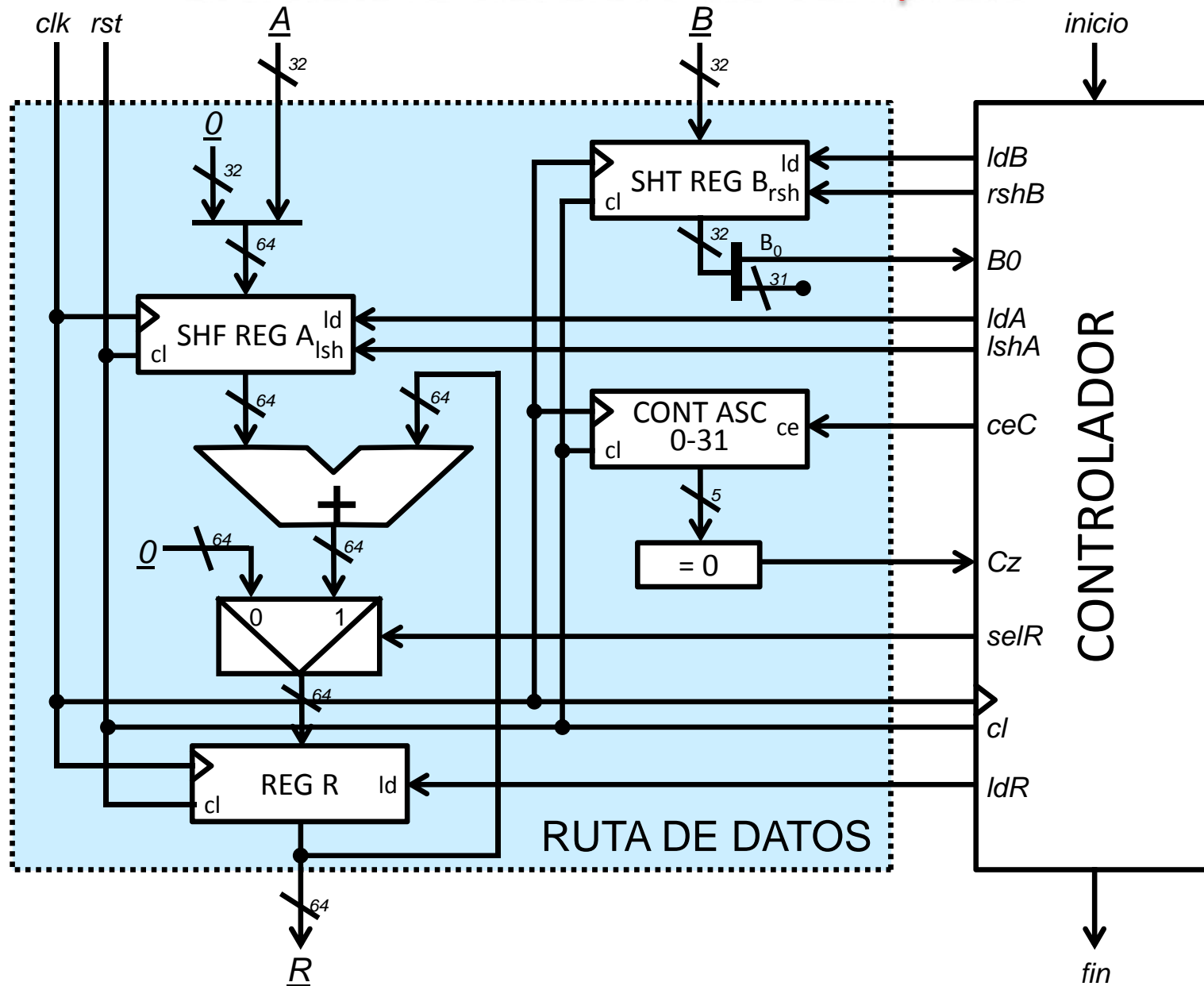
Señales de estado



5. Identificación de las señales de estado

Rutas de datos y controladores

Estructura del sistema completo



Rutas de datos y controladores

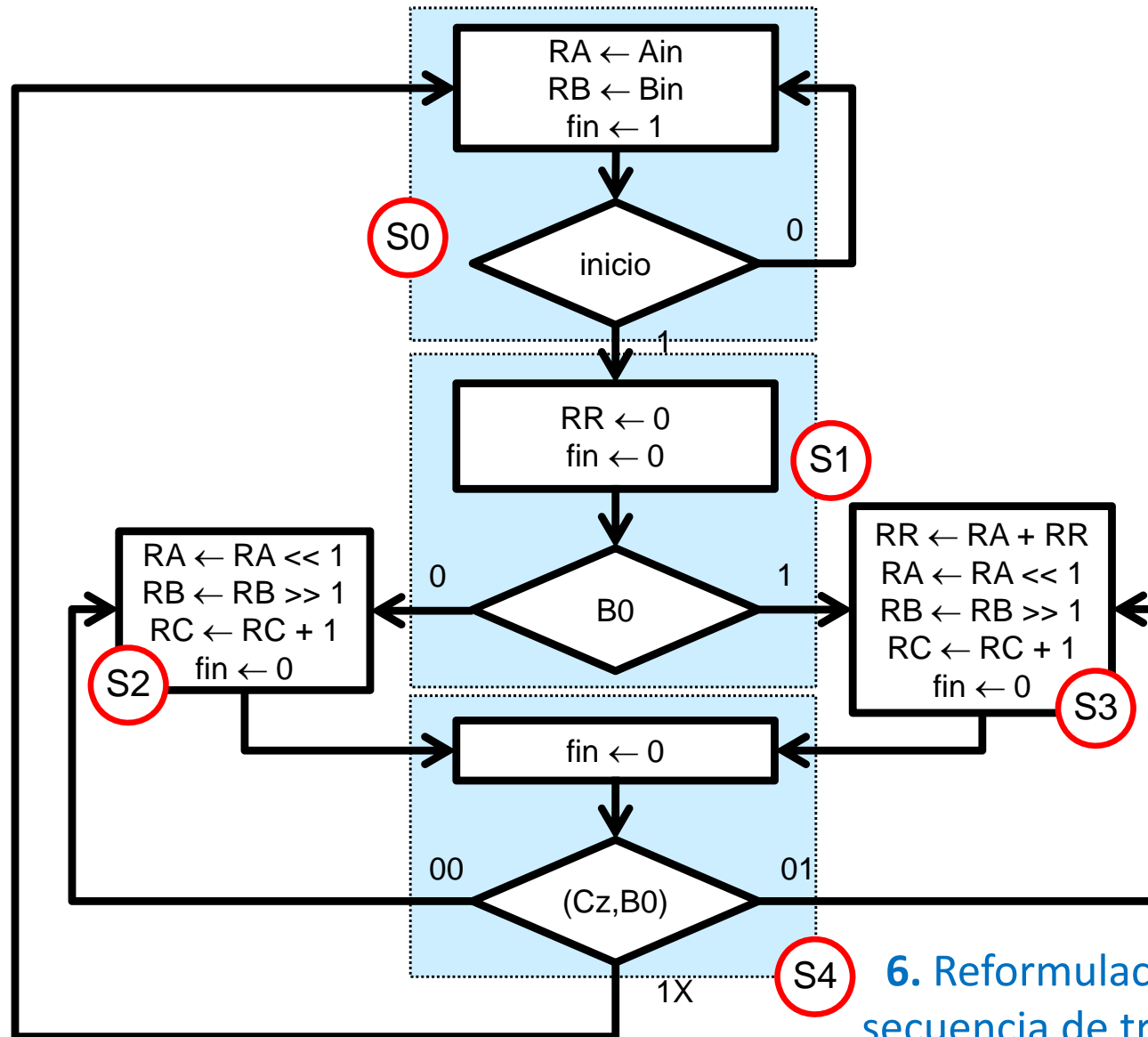
Formalismo ASM (Algorithmic State Maquine)



- Para facilitar la expresión del algoritmo como una FSM usaremos un **diagrama ASM** como formalismo intermedio.
- Un diagrama ASM es un grupo de bloques interconectados.
 - Cada **bloque** se corresponde con un **estado de la FSM** e indica las acciones que se realizan en paralelo en dicho estado.
- Todo **bloque** (recuadro azul) esta formado por:
 - 1 **caja de estado** (rectangular): que indica las transferencias entre registros que se realizan incondicionalmente en el estado.
 - 0..n **cajas de selección** (romboidales): indican las condiciones bajo las cuales se que se determina el estado siguiente del sistema o se realizan ciertas transferencias entre registros.
 - 0..n **cajas condicionales** (ovaladas): indica las trasferencias entre registros que se realizan condicionalmente en un estado.

Rutas de datos y controladores

Algoritmo en ASM



begin

1. $A := A_{in};$
2. $B := B_{in};$
3. $R := 0;$
- for $C := 0$ to 31 do *begin*
4. if $B_0 = 1$ then $R := R + A;$
5. $A := A \ll 1;$
6. $B := B \gg 1;$
- end for;
7. $R_{out} := R;$

end;

6. Reformulación del algoritmo como una secuencia de transferencias entre registros

Rutas de datos y controladores

Algoritmo en ASM

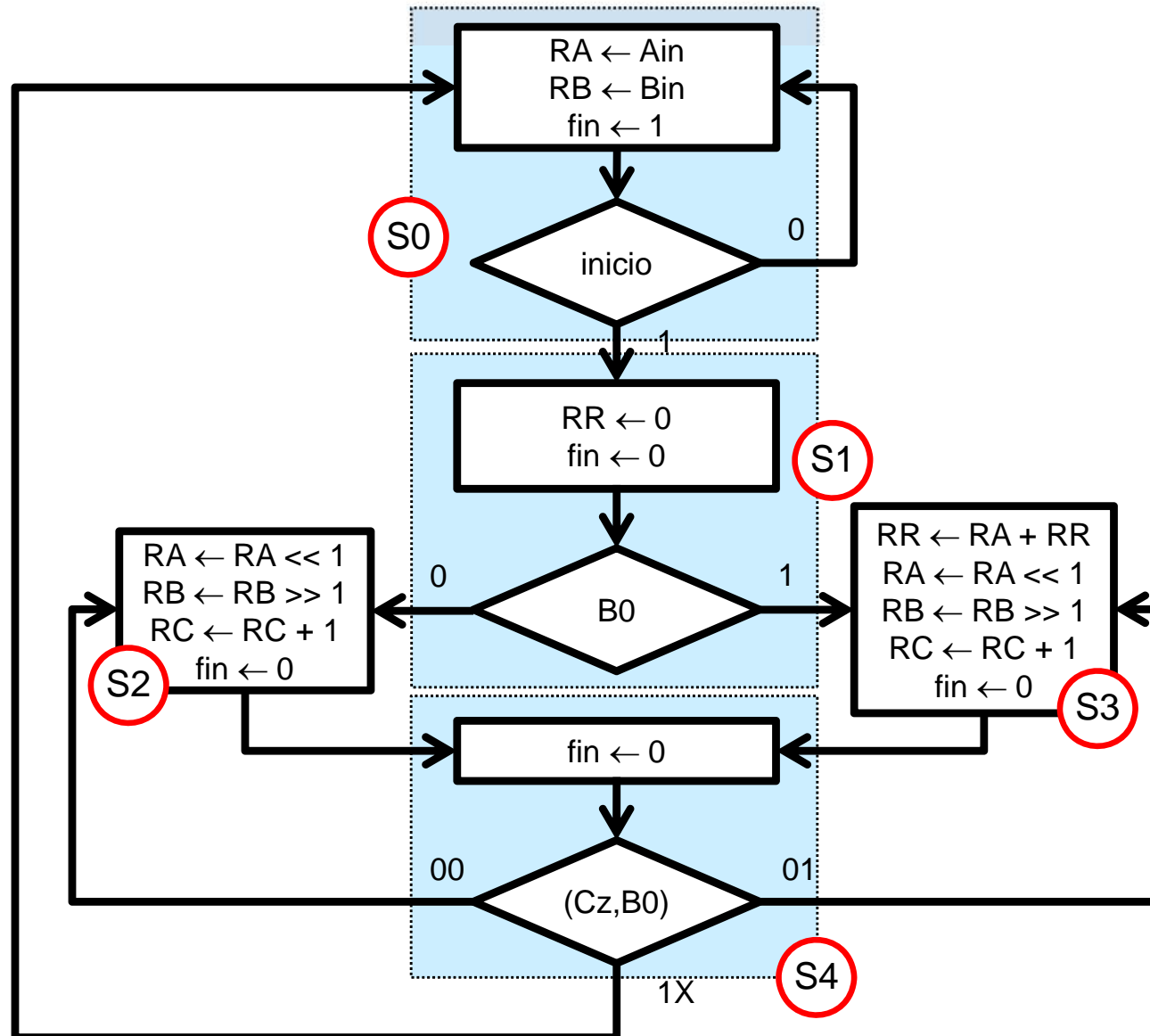


versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

40



S0 $RA \leftarrow A_{in}$; $RB \leftarrow B_{in}$;
 $fin \leftarrow 1$;
si ($inicio=0$) ir a S0;
si ($inicio=1$) ir a S1;

S1 $RR \leftarrow 0$; $fin \leftarrow 0$;
si ($B0=0$) ir a S2;
si ($B0=1$) ir a S3;

S2 $RA \leftarrow RA \ll 1$;
 $RB \leftarrow RB \gg 1$;
 $RC \leftarrow RC + 1$; $fin \leftarrow 0$;
ir a S4;

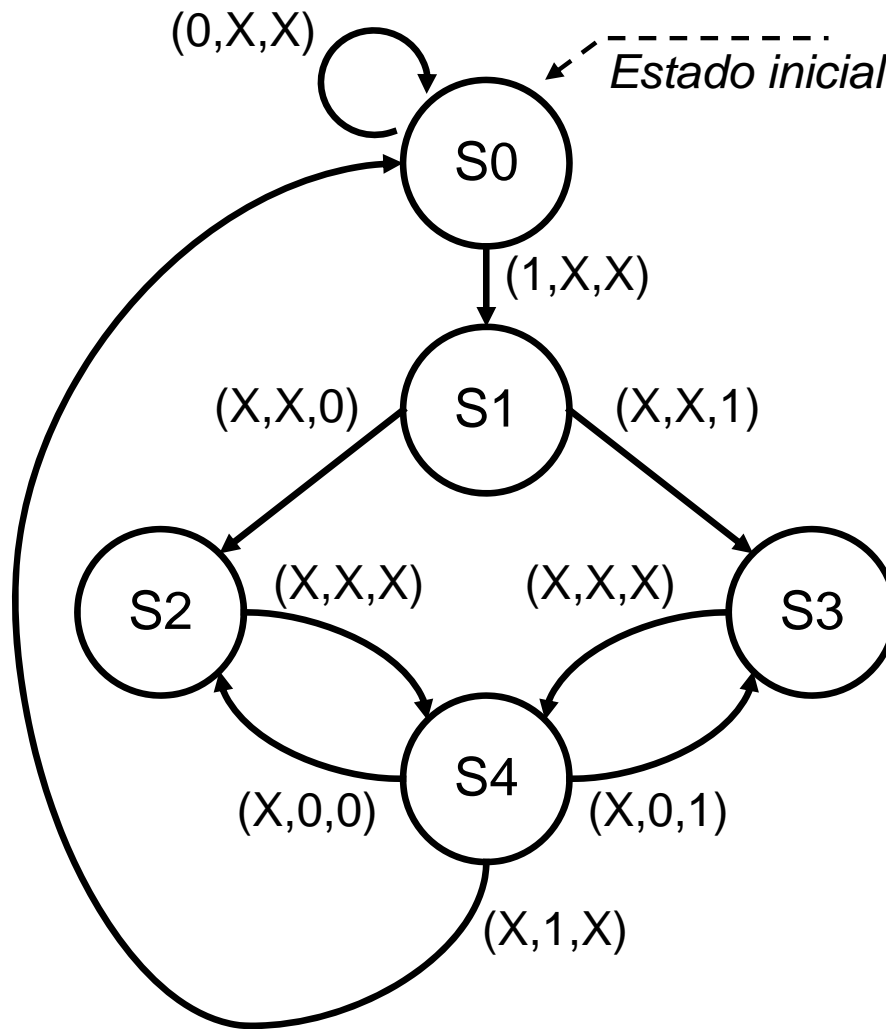
S3 $RR \leftarrow RA + RR$;
 $RA \leftarrow RA \ll 1$;
 $RB \leftarrow RB \gg 1$;
 $RC \leftarrow RC + 1$; $fin \leftarrow 0$;
ir a S4;

S4 $fin \leftarrow 0$;
si ($Cz=1$) ir a S0;
si ($Cz=0$ y $B0=0$) ir a S2;
si ($Cz=0$ y $B0=1$) ir a S3;

tiempo de cálculo: $2+32 \times 2 = 66$ ciclos

Rutas de datos y controladores

Algoritmo como una FSM



entradas del controlador: (inicio, Cz, B0)

Función de transición de estados

estado	inicio	Cz	B0	estado'
S0	0	X	X	S0
S0	1	X	X	S1
S1	X	X	0	S2
S1	X	X	1	S3
S2	X	X	X	S4
S3	X	X	X	S4
S4	X	0	0	S2
S4	X	0	1	S3
S4	X	1	X	S0

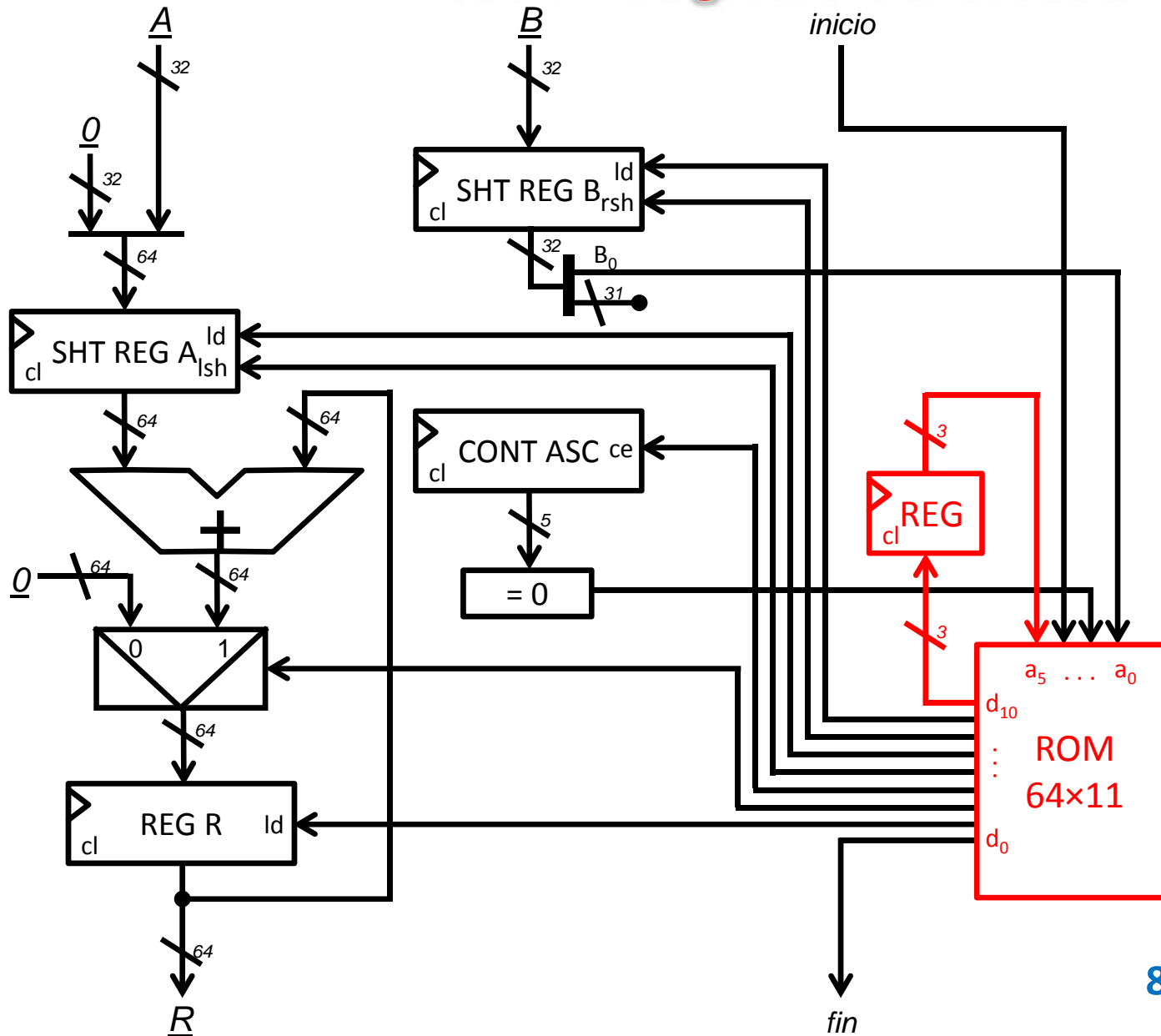
Función de salida

estado	ldB	rshB	ldA	lshA	ceC	selR	ldR	fin
S0	1	0	1	0	0	-	0	1
S1	0	0	0	0	0	0	1	0
S2	0	1	0	1	1	-	0	0
S3	0	1	0	1	1	1	1	0
S4	0	0	0	0	0	-	0	0

7. Especificación del controlador como FSM

Rutas de datos y controladores

ROM + registro de estado



- S0** $RA \leftarrow A_{in}; RB \leftarrow B_{in};$
 $fin \leftarrow 1;$
 si ($inicio=0$) ir a S0;
 si ($inicio=1$) ir a S1;
- S1** $RR \leftarrow 0; fin \leftarrow 0;$
 si ($B_0=0$) ir a S2;
 si ($B_0=1$) ir a S3;
- S2** $RA \leftarrow RA \ll 1;$
 $RB \leftarrow RB \gg 1;$
 $RC \leftarrow RC + 1; fin \leftarrow 0;$
 ir a S4;
- S3** $RR \leftarrow RA + RR;$
 $RA \leftarrow RA \ll 1;$
 $RB \leftarrow RB \gg 1;$
 $RC \leftarrow RC + 1; fin \leftarrow 0;$
 ir a S4;
- S4** $fin \leftarrow 0;$
 si ($Cz=1$) ir a S0;
 si ($Cz=0$ y $B_0=0$) ir a S2;
 si ($Cz=0$ y $B_0=1$) ir a S3;

8. Diseño del controlador



Rutas de datos y controladores

Generación del contenido de la ROM

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

43

q_2	q_1	q_0	inicio	Cz	B0
a_5	a_4	a_3	a_2	a_1	a_0
0	0	0	0	0	0
0	0	0	0	0	1
0	0	0	0	1	0
0	0	0	0	1	1
0	0	0	1	0	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	0	1	1	1
0	0	1	0	0	0
0	0	1	0	0	1
0	0	1	0	1	0
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	0	1
0	0	1	1	1	0
0	0	1	1	1	1
...					
1	1	1	1	1	1

dirección

	q_2'	q_1'	q_0'	ldB	$rshB$	ldA	$lshA$	ceC	$selR$	ldR	fin
DIR.	d_{10}	d_9	d_8	d_7	d_6	d_5	d_4	d_3	d_2	d_1	d_0
0	0	0	0	1	0	1	0	0	1	1	1
1	0	0	0	1	0	1	0	0	1	1	1
2	0	0	0	1	0	1	0	0	1	1	1
3	0	0	0	1	0	1	0	0	1	1	1
4	0	0	1	1	0	1	0	0	1	1	1
5	0	0	1	1	0	1	0	0	1	1	1
6	0	0	1	1	0	1	0	0	1	1	1
7	0	0	1	1	0	1	0	0	1	1	1
8	0	1	0	0	0	0	0	0	0	1	0
9	0	1	1	0	0	0	0	0	0	1	0
10	0	1	0	0	0	0	0	0	0	1	0
11	0	1	1	0	0	0	0	0	0	1	0
12	0	1	0	0	0	0	0	0	0	1	0
13	0	1	1	0	0	0	0	0	0	1	0
14	0	1	0	0	0	0	0	0	0	1	0
15	0	1	1	0	0	0	0	0	0	1	0
...	...										
63											

ROM 64x11

estado	inicio	Cz	B0	estado'
S0	0	X	X	S0
S0	1	X	X	S1
S1	X	X	0	S2
S1	X	X	1	S3
S2	X	X	X	S4
S3	X	X	X	S4
S4	X	0	0	S2
S4	X	0	1	S3
S4	X	1	X	S0

estado	ldB	rshB	ldA	lshA	ceC	selR	ldR	fin
S0	1	0	1	0	0	-	0	1
S1	0	0	0	0	0	0	1	0
S2	0	1	0	1	1	-	0	0
S3	0	1	0	1	1	1	1	0
S4	0	0	0	0	0	-	0	0

8. Diseño del controlador



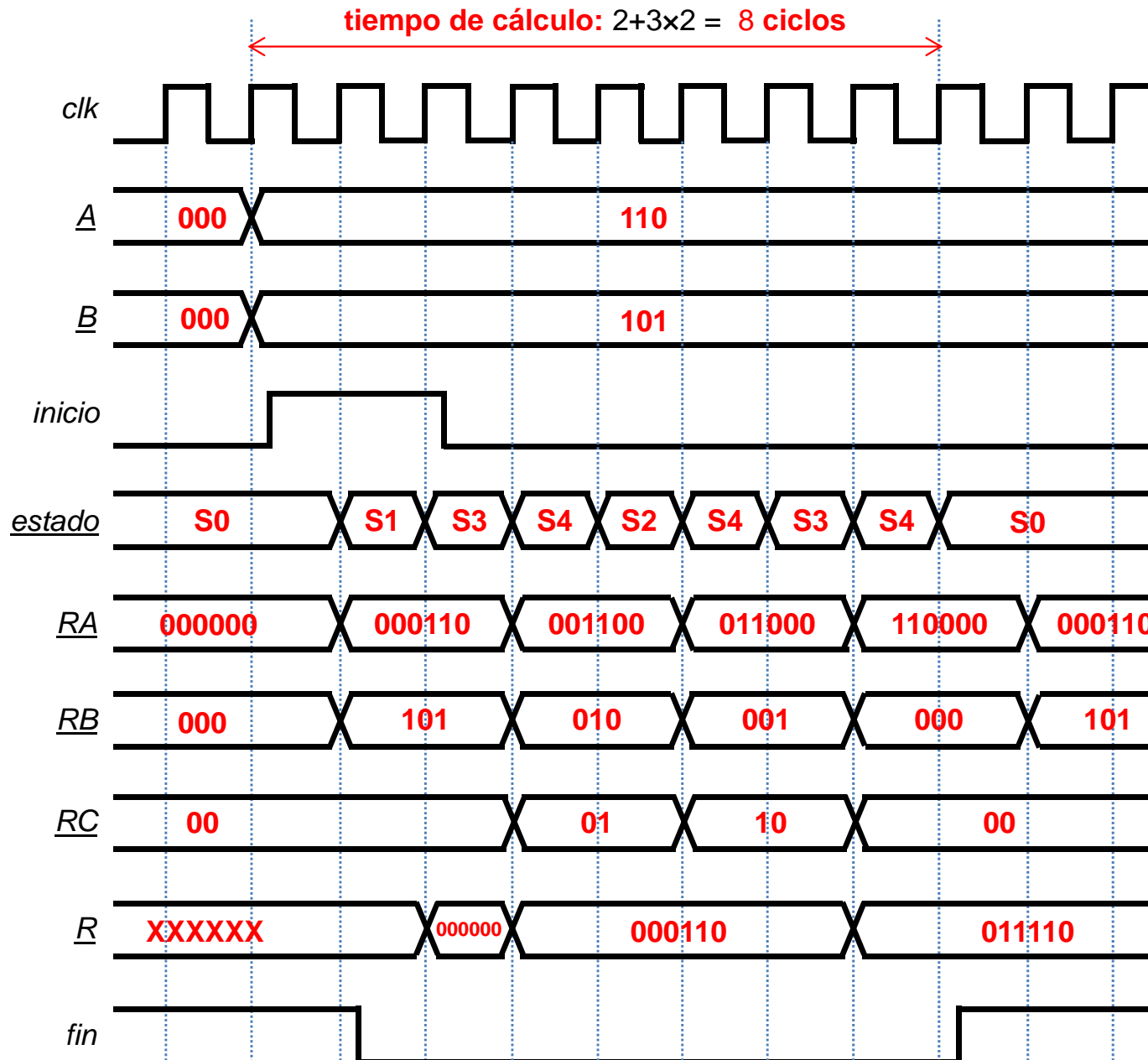
Rutas de datos y controladores

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

44



S0 $RA \leftarrow A_{in}; RB \leftarrow B_{in};$
 $fin \leftarrow 1;$
si ($inicio=0$) ir a S0;
si ($inicio=1$) ir a S1;

S1 $RR \leftarrow 0; fin \leftarrow 0;$
si ($B0=0$) ir a S2;
si ($B0=1$) ir a S3;

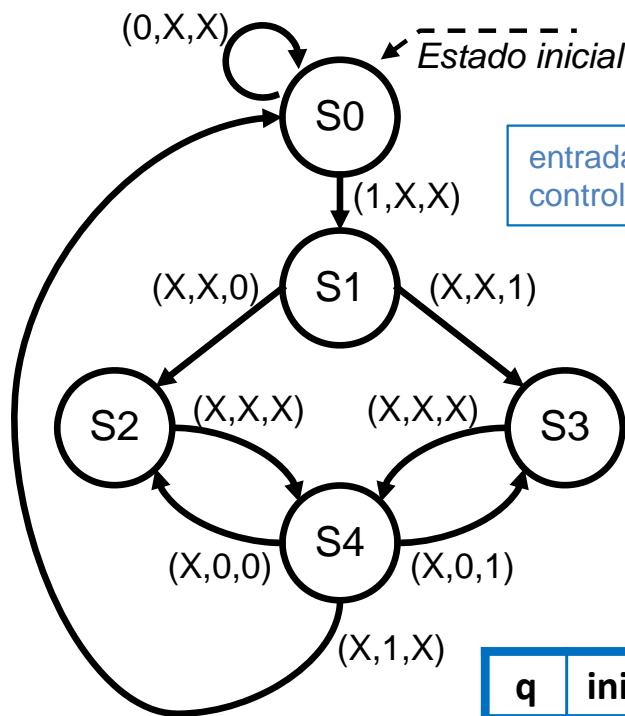
S2 $RA \leftarrow RA \ll 1;$
 $RB \leftarrow RB \gg 1;$
 $RC \leftarrow RC + 1; fin \leftarrow 0;$
ir a S4;

S3 $RR \leftarrow RA + RB;$
 $RA \leftarrow RA \ll 1;$
 $RB \leftarrow RB \gg 1;$
 $RC \leftarrow RC + 1; fin \leftarrow 0;$
ir a S4;

S4 $fin \leftarrow 0;$
si ($Cz=1$) ir a S0;
si ($Cz=0$ y $B0=0$) ir a S2;
si ($Cz=0$ y $B0=1$) ir a S3;

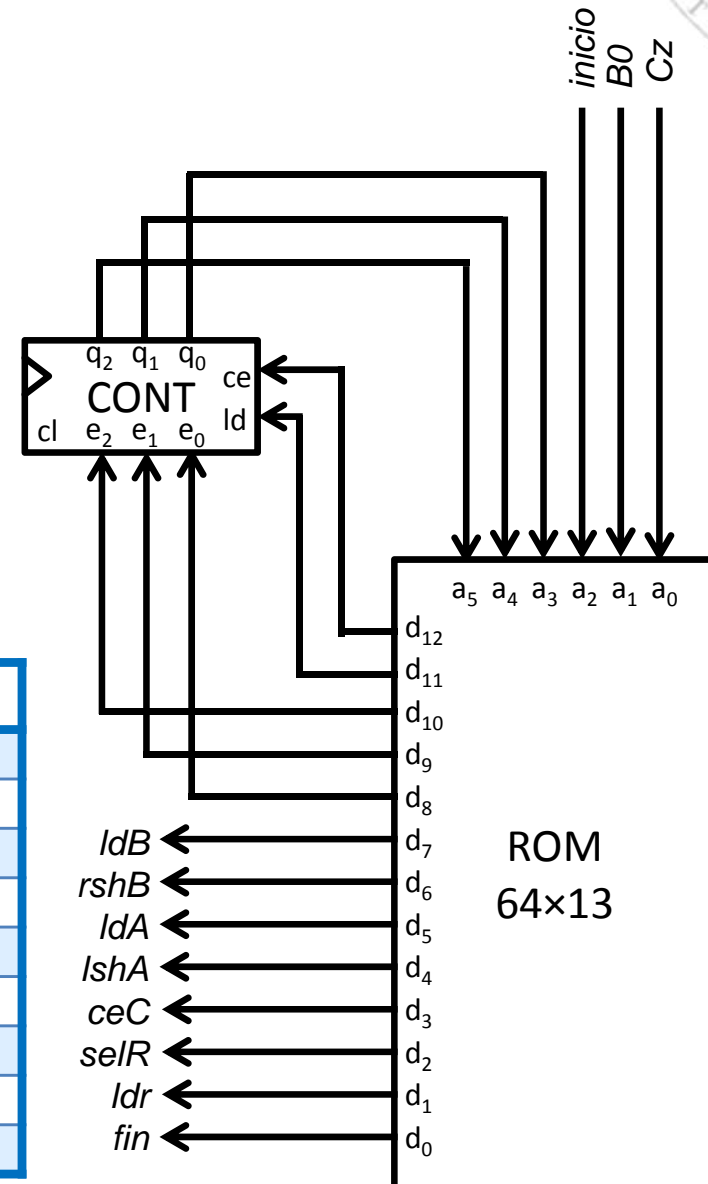
Rutas de datos y controladores

Otro posible controlador: ROM + contador



entradas del controlador: (inicio, Cz, B0)

q	inicio	Cz	B0	e	ld	ce
S0	0	X	X	-	0	0
S0	1	X	X	-	0	1
S1	X	X	0	-	0	1
S1	X	X	1	S3	1	0
S2	X	X	X	S4	1	0
S3	X	X	X	-	0	1
S4	X	0	0	S2	1	0
S4	X	0	1	S3	1	0
S4	X	1	X	S0	1	0





Rutas de datos y controladores

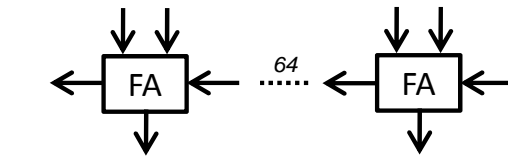
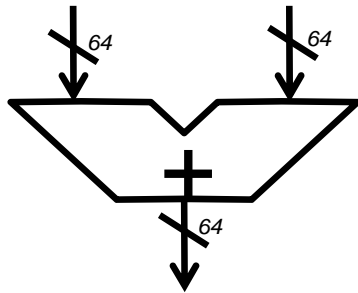
Cálculo del coste y tiempo de ciclo

versión 12/09/14

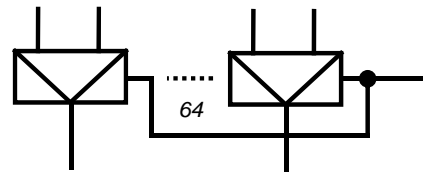
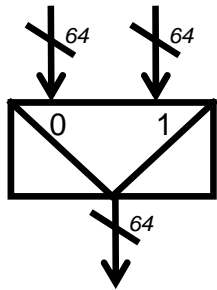
tema 7:
Módulos secuenciales básicos

FC

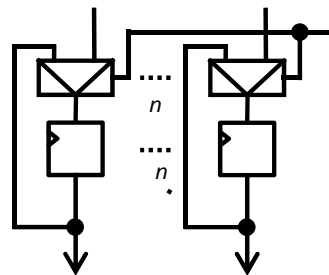
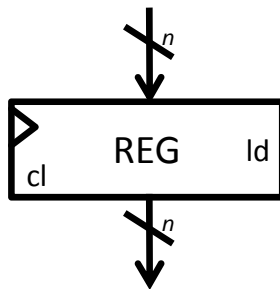
46



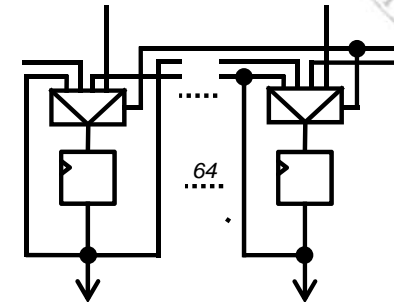
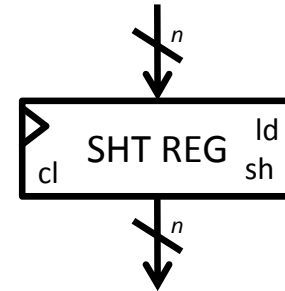
área: $64 \times 29.49 = 1887 \mu\text{m}^2$
retardo: $64 \times 226 = 14464 \text{ ps}$



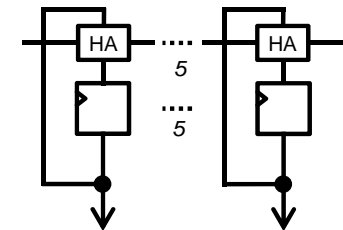
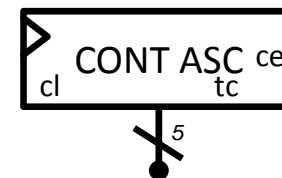
área: $64 \times 11.05 = 707.2 \mu\text{m}^2$
retardo: $1 \times 223 = 223 \text{ ps}$



área: $n \times 11.05 + n \times 32.26 = n \times 43.31 \mu\text{m}^2$
retardo CLK→Q: $1 \times 167 = 167 \text{ ps}$
retardo in→D: $1 \times 223 = 223 \text{ ps}$



área: $n \times 23.04 + n \times 32.26 = n \times 55.3 \mu\text{m}^2$
retardo CLK→Q: $1 \times 167 = 167 \text{ ps}$
retardo in→D: $1 \times 250 = 250 \text{ ps}$



área: $5 \times 15.77 + 5 \times 32.26 = 240.6 \mu\text{m}^2$
retardo CLK→Q: $1 \times 167 = 167 \text{ ps}$
retardo in→D: $5 \times 114 = 570 \text{ ps}$

= 0

área: $15,67 \mu\text{m}^2$
retardo: 126 ps

ROM
64×11

área: $971 \mu\text{m}^2$
retardo: 573 ps

Rutas de datos y controladores

Cálculo del tiempo de ciclo

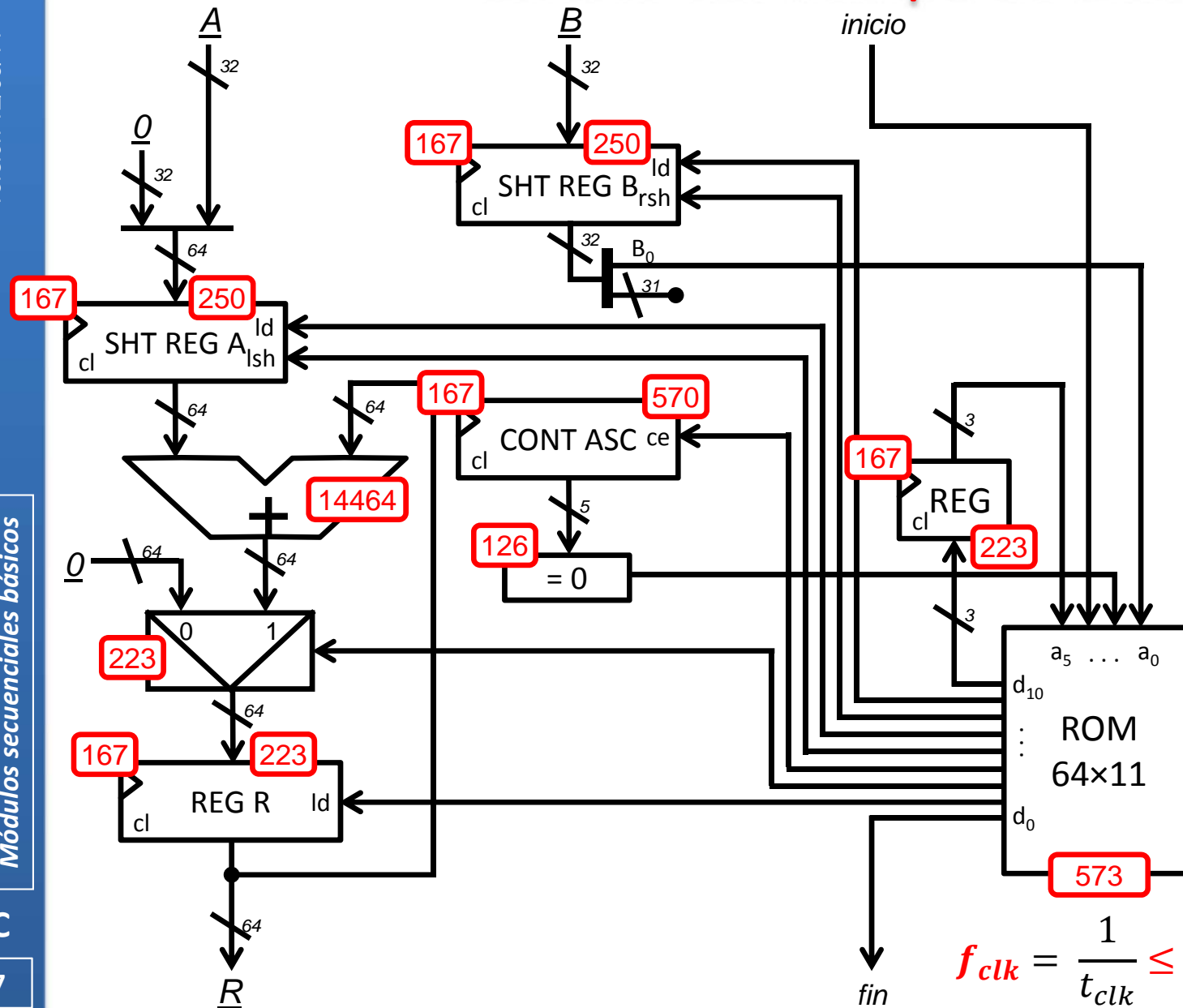


versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

47



tiempo de cálculo: 1 μ s
(66 ciclos \times 15077 ps/ciclo)

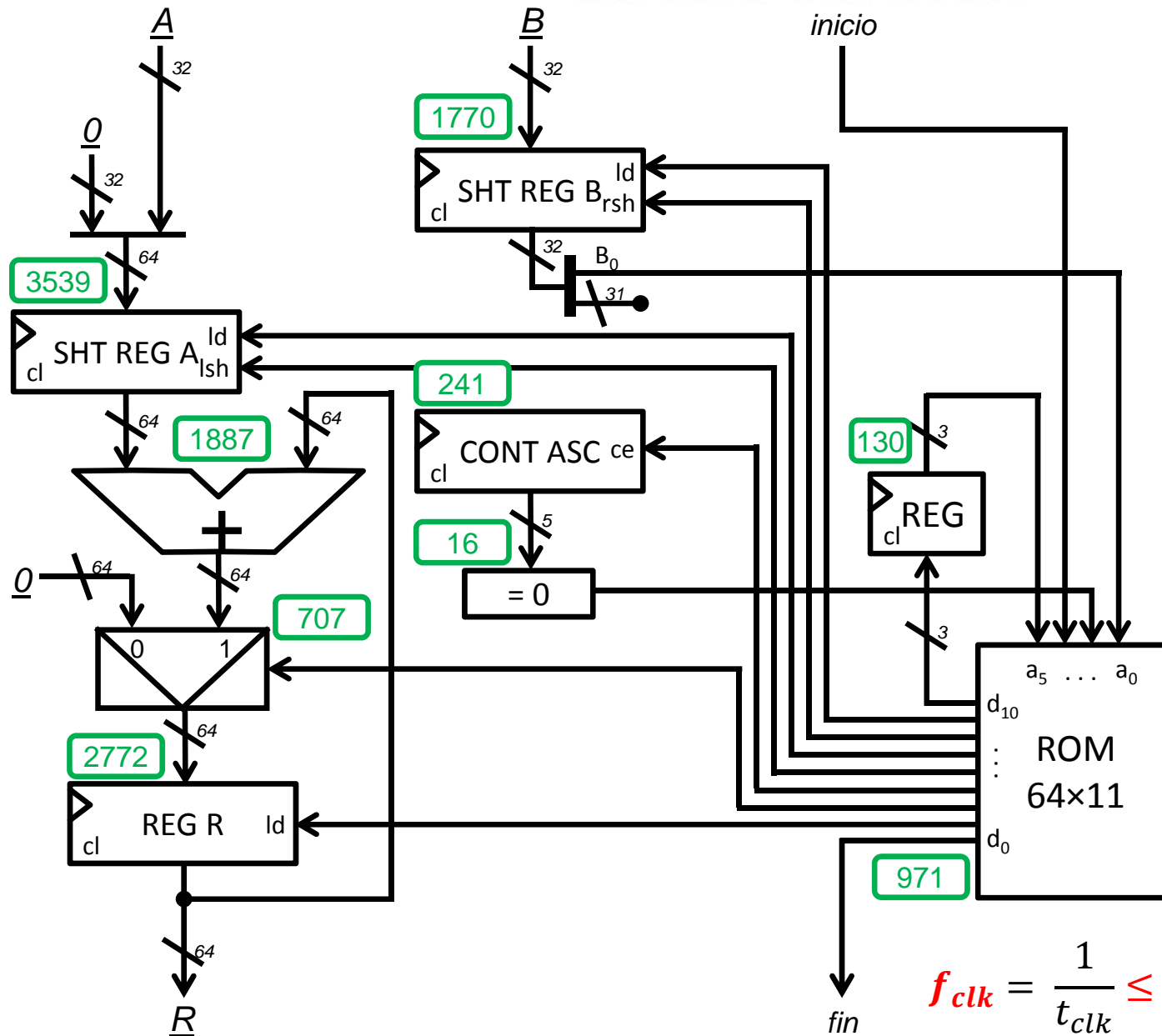
transferencia	retardo
RA \leftarrow Ain	990 ps
RB \leftarrow Bin	990 ps
fin \leftarrow X	740 ps
RR \leftarrow 0	1186 ps
RA \leftarrow RA \ll 1	990 ps
RB \leftarrow RB \gg 1	990 ps
RR \leftarrow RA + RR	15077ps
RC \leftarrow RC + 1	1350 ps
cálculo de estado	1089 ps
máximo	15077ps

$$f_{clk} = \frac{1}{t_{clk}} \leq \frac{1}{15077 \cdot 10^{-12} \text{s}} = \mathbf{66 \text{ MHz}}$$

Rutas de datos y controladores



Cálculo del coste



área: 12033 μm^2

tiempo de cálculo: 1 μs
(66 ciclos \times 15077 ps/ciclo)

transferencia	retardo
RA \leftarrow Ain	990 ps
RB \leftarrow Bin	990 ps
fin \leftarrow X	740 ps
RR \leftarrow 0	1186 ps
RA \leftarrow RA \ll 1	990 ps
RB \leftarrow RB \gg 1	990 ps
RR \leftarrow RA + RR	15077ps
RC \leftarrow RC + 1	1350 ps
cálculo de estado	963 ps
máximo	15077ps

$$f_{clk} = \frac{1}{t_{clk}} \leq \frac{1}{15077 \cdot 10^{-12}\text{s}} = 66 \text{ MHz}$$



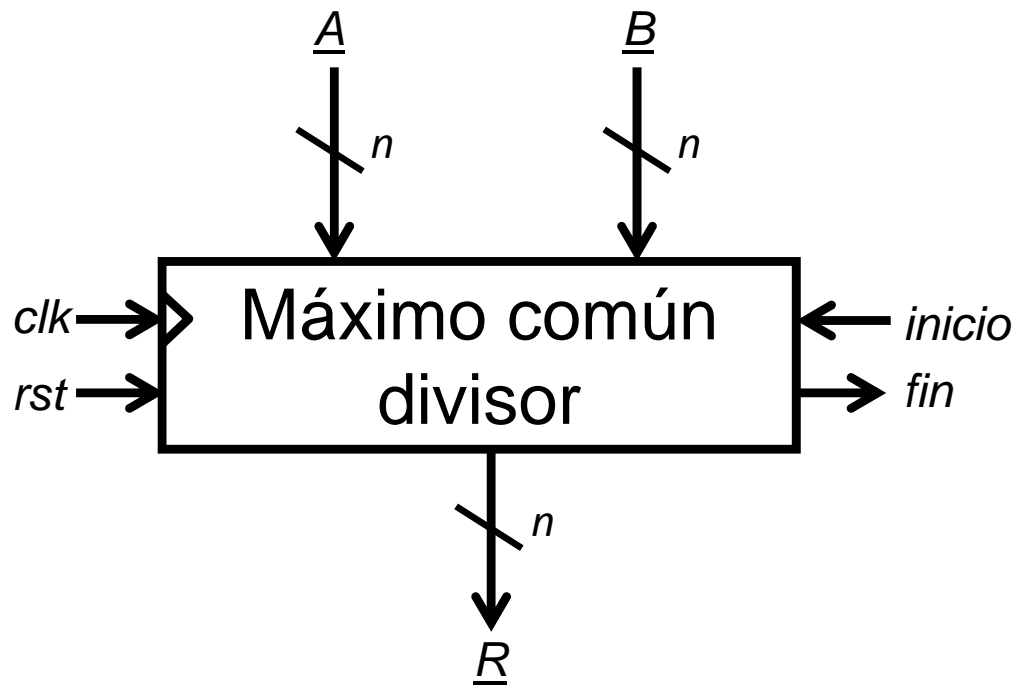
Rutas de datos y controladores

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

49



	A_3	A_2	A_1	A_0	B_3	B_2	B_1	B_0	R_3	R_2	R_1	R_0
1.	1	0	0	1	-	-	-	-	-	-	-	-
2.	1	0	0	1	0	1	1	0	-	-	-	-
3.	1	0	0	1	0	1	1	0	0	0	0	0
4.	0	0	1	1	0	1	1	0	0	0	0	0
5.	0	0	1	1	0	0	1	1	0	0	0	0
6.	0	0	1	1	0	0	1	1	0	0	1	1

```
begin
1.  A := Ain;
2.  B := Bin;
3.  R := 0;
   if (A<>0 and B<>0) then
     begin
       while (A<>B) do
         if A>B then
           A := A - B;
         else
           B := B - A;
       R := A;
     end if;
7.  Rout := R;
end;
```

algoritmo de Euclides

Rutas de datos y controladores

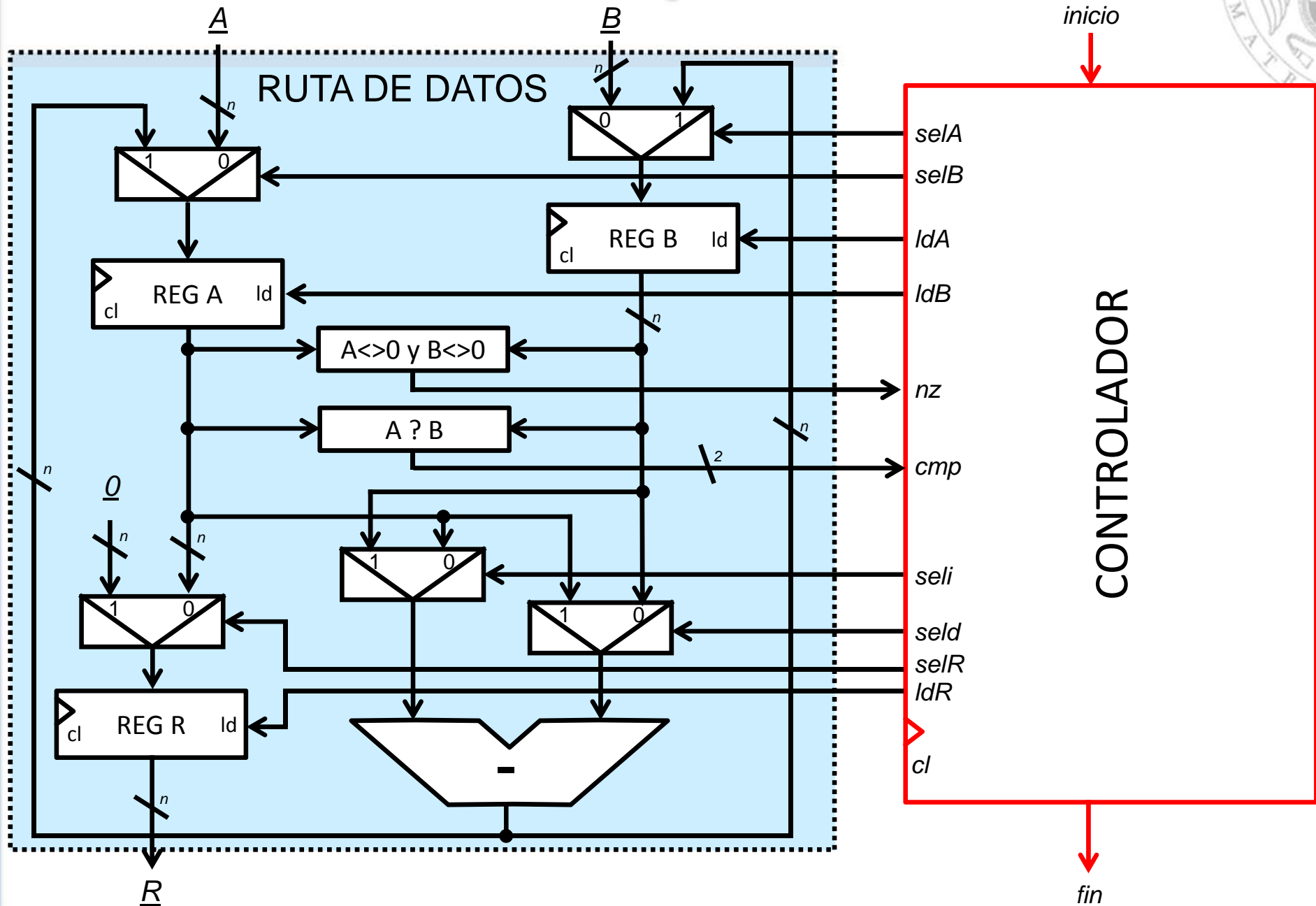


versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

50



Rutas de datos y controladores

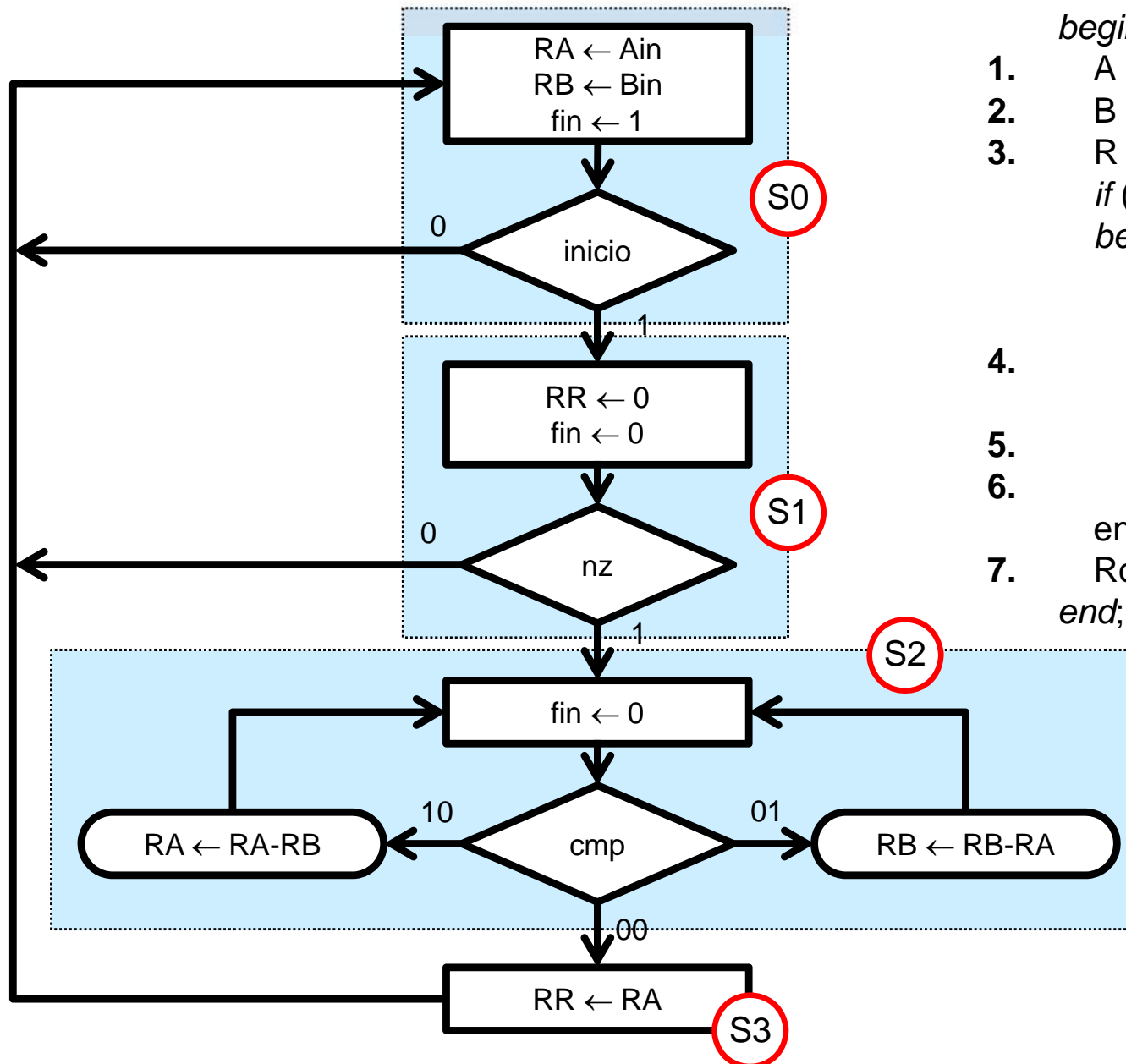


versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

51



begin

1. $A := A_{in};$
2. $B := B_{in};$
3. $R := 0;$
if $(A \neq 0 \text{ and } B \neq 0)$ *then*
begin
 while $(A \neq B)$ *do*
 if $A > B$ *then*
 $A := A - B;$
 else
 $B := B - A;$
 $R := A;$
 end if;
 $R_{out} := R;$
end;
- 4.
- 5.
- 6.
- 7.

Rutas de datos y controladores

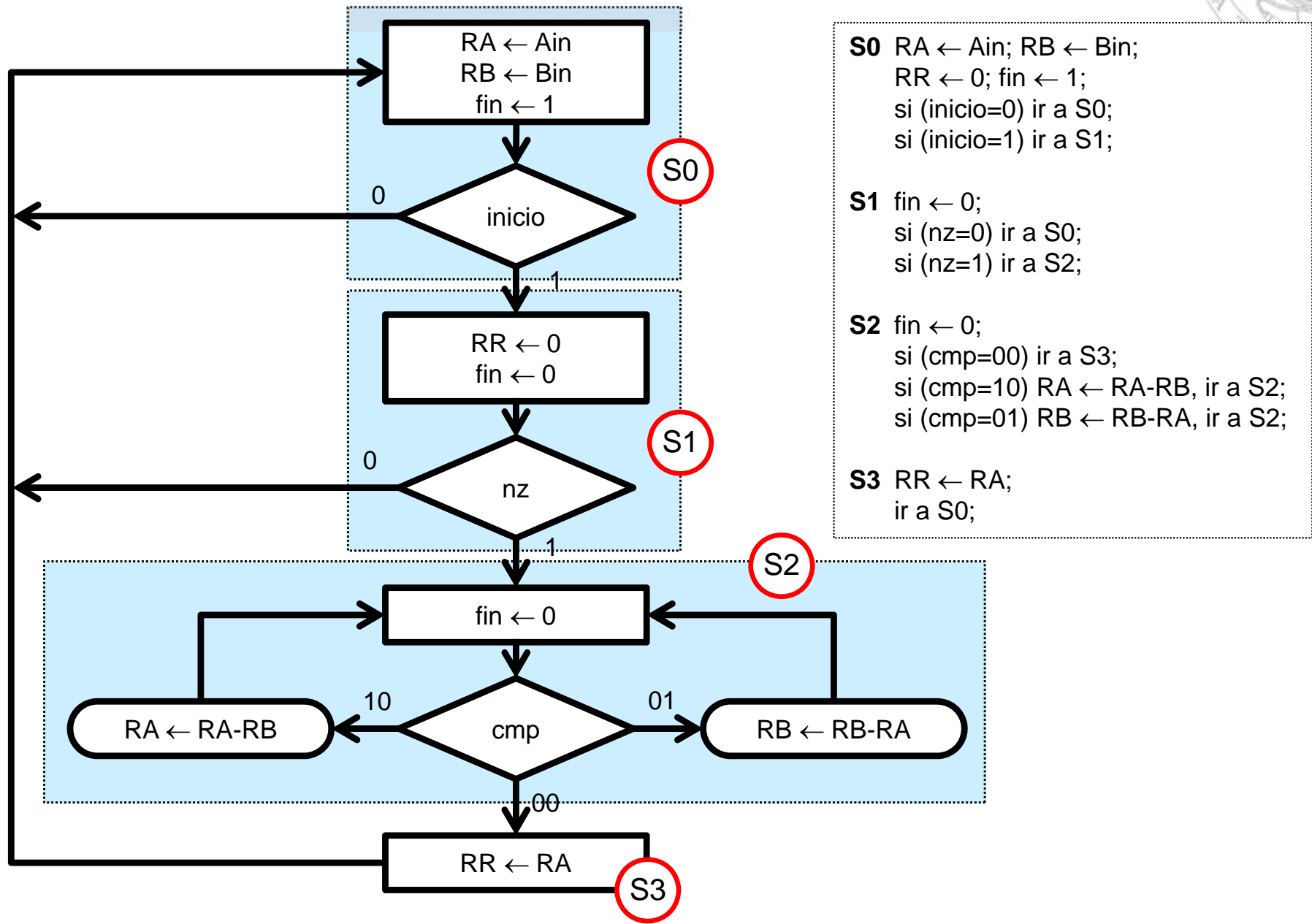


versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

52





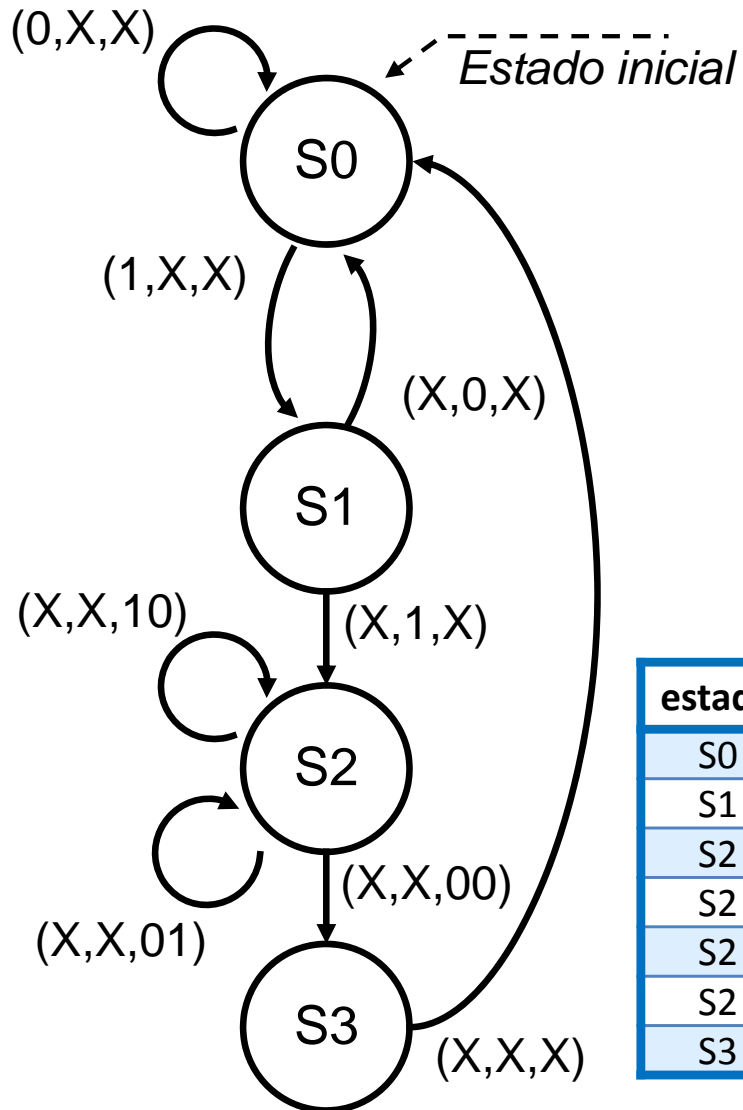
Rutas de datos y controladores

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

53



estado	inicio	nz	cmp ₁	cmp ₀	estado'
S0	0	X	X	X	S0
S0	1	X	X	X	S1
S1	X	0	X	X	S0
S1	X	1	X	X	S2
S2	X	X	0	0	S0
S2	X	X	0	1	S2
S2	X	X	1	0	S2
S3	X	X	X	X	S0

estado	cmp ₁	cmp ₀	selA	selB	ldA	ldB	seli	seld	selR	ldR	fin
S0	X	X	0	0	1	1	-	-	-	0	1
S1	X	X	-	-	0	0	-	-	1	1	0
S2	0	0	-	-	0	0	-	-	-	0	0
S2	0	1	-	1	0	1	0	0	-	0	0
S2	1	0	1	-	1	0	1	1	-	0	0
S2	1	1	-	-	-	-	-	-	-	-	-
S3	X	X	0	0	0	0	-	-	0	1	0

compatibles

compatibles

entradas del controlador: (inicio, nz, cmp)



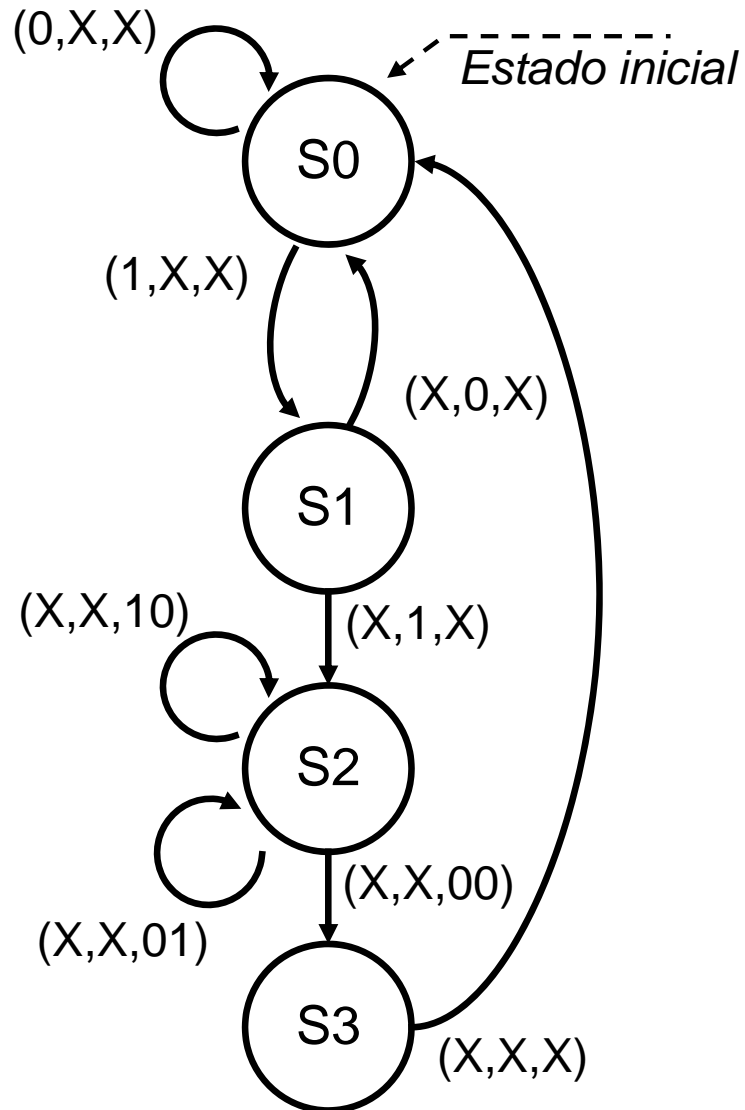
Rutas de datos y controladores

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

54



entradas del controlador: (inicio, nz, cmp)

estado	inicio	nz	cmp ₁	cmp ₀	estado'
S0	0	X	X	X	S0
S0	1	X	X	X	S1
S1	X	0	X	X	S0
S1	X	1	X	X	S2
S2	X	X	0	0	S0
S2	X	X	0	1	S2
S2	X	X	1	0	S2
S3	X	X	X	X	S0

estado	cmp ₁	cmp ₀	sela selB selR	IdA seli seld	IdB	IdR	fin
S0	X	X	0	1	1	0	1
S1	X	X	1	0	0	1	0
S2	0	0	-	0	0	0	0
S2	0	1	1	0	1	0	0
S2	1	0	1	1	0	0	0
S2	1	1	-	-	-	-	-
S3	X	X	0	0	0	1	0

Rutas de datos y controladores

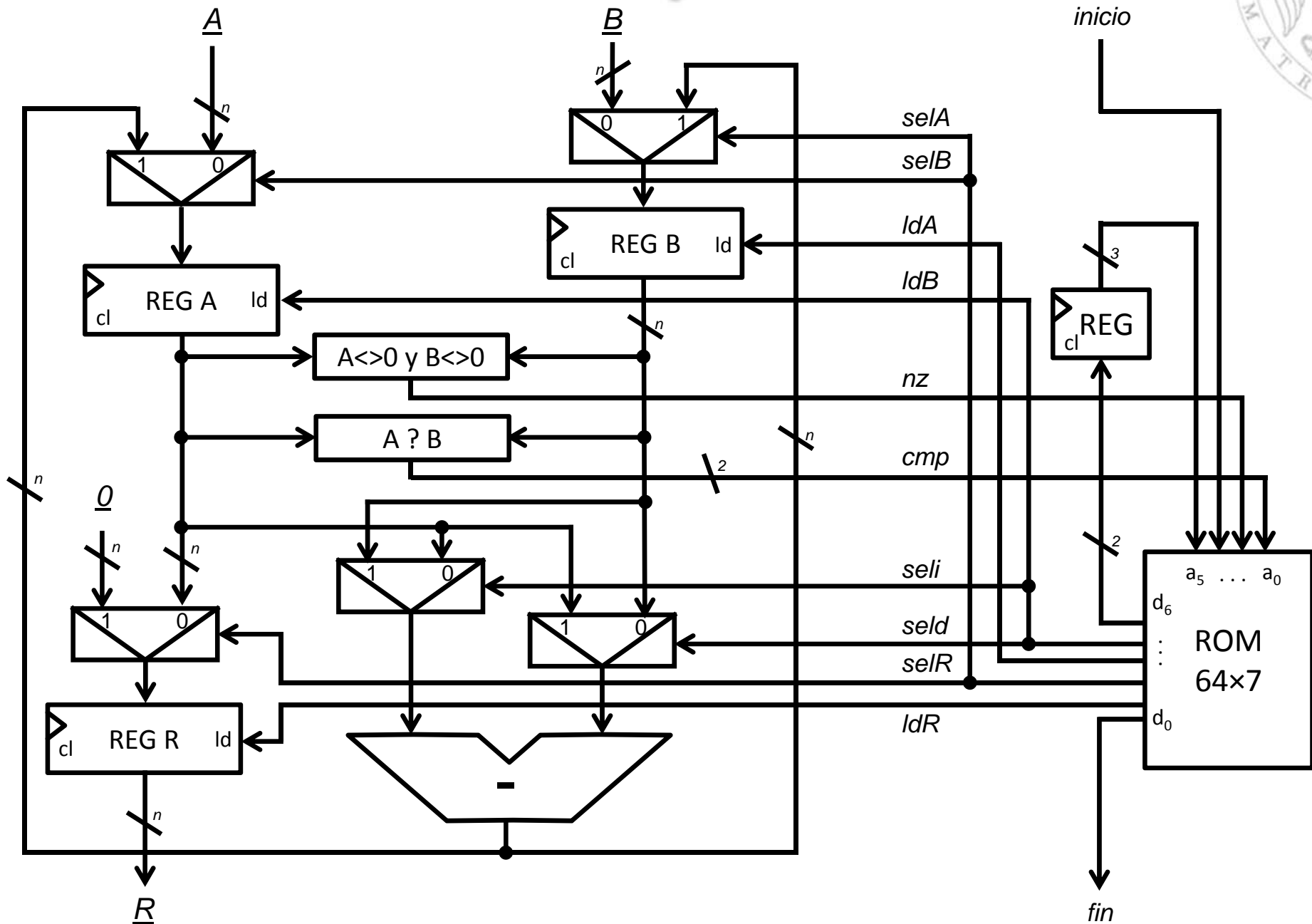


versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

55





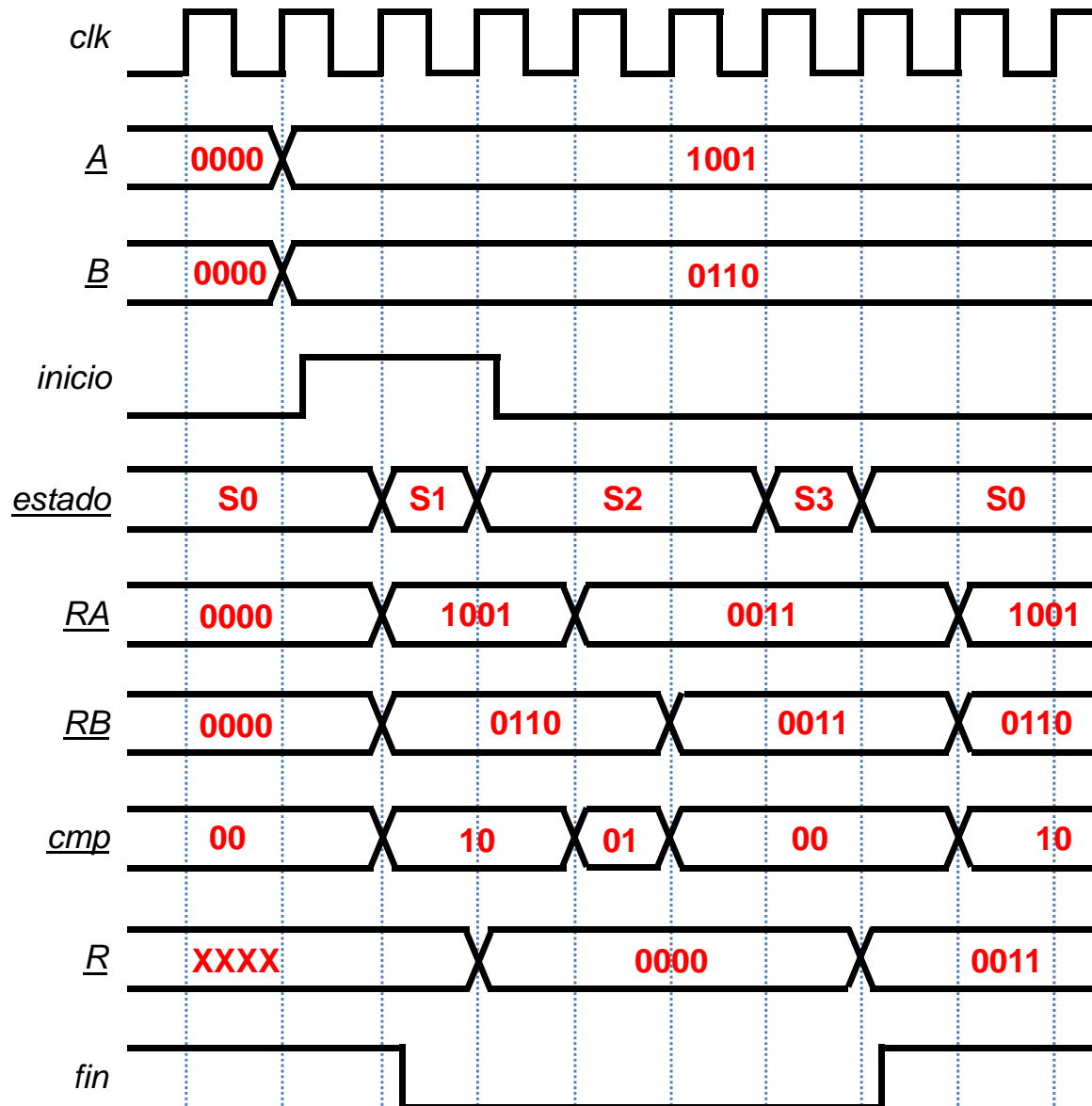
Rutas de datos y controladores

versión 12/09/14

tema 7:
Módulos secuenciales básicos

FC

56



S0 $RA \leftarrow A_{in}; RB \leftarrow B_{in};$
 $fin \leftarrow 1;$
si ($inicio=0$) ir a S0;
si ($inicio=1$) ir a S1;

S1 $RR \leftarrow 0; fin \leftarrow 0;$
si ($nz=0$) ir a S0;
si ($nz=1$) ir a S2;

S2 $fin \leftarrow 0;$
si ($cmp=00$) ir a S3;
si ($cmp=10$) $RA \leftarrow RA-RB$, ir a S2;
si ($cmp=01$) $RB \leftarrow RB-RA$, ir a S2;

S3 $RR \leftarrow RA;$
ir a S0;