



Tema 6:

# Implementación de sistemas secuenciales síncronos

Fundamentos de computadores

**José Manuel Mendías Cuadros**

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



# Contenidos

- ✓ Biestable SR.
- ✓ Biestable D.
- ✓ Síntesis con biestables D.
- ✓ Inicialización de sistemas secuenciales.
- ✓ Aspectos tecnológicos.

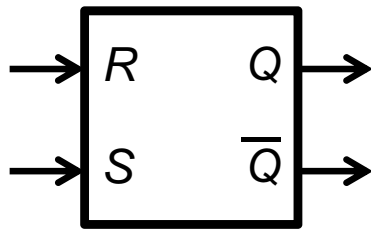
Transparencias basadas en los libros:

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

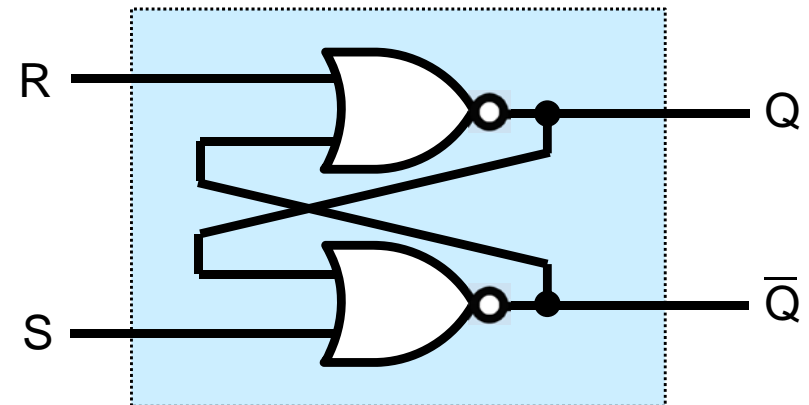
# Biestable



- Dispositivo capaz de almacenar **físicamente** un **bit** de información (tener 2 estados estables).
  - mediante un circuito combinacional **realimentado**



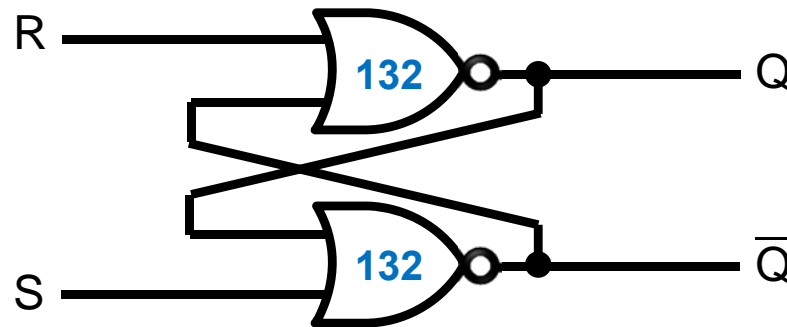
R(t)	S(t)	Q(t+Δt)
0	0	Q(t)
0	1	1
1	0	0
1	1	prohibido



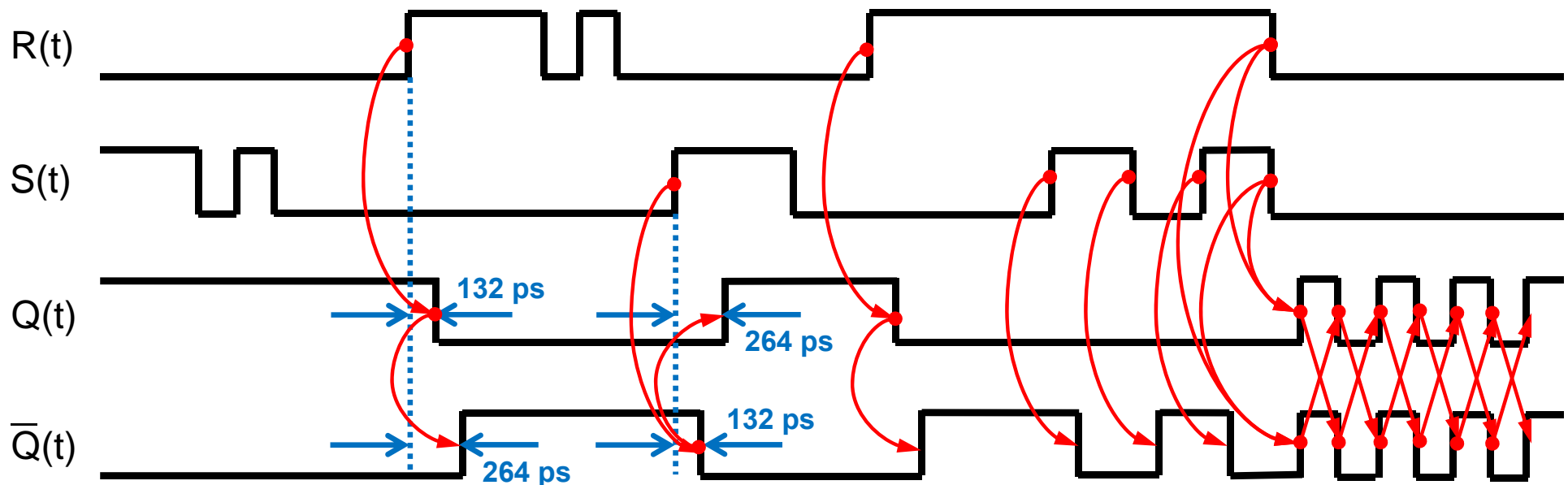
Biestable SR asíncrono



# Biastable SR asíncrono



R(t)	S(t)	Q(t+Δt)
0	0	Q(t)
0	1	1
1	0	0
1	1	prohibido





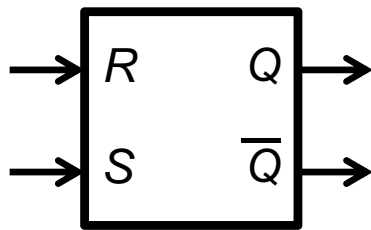
# Biastable SR asíncrono

versión 12/09/14

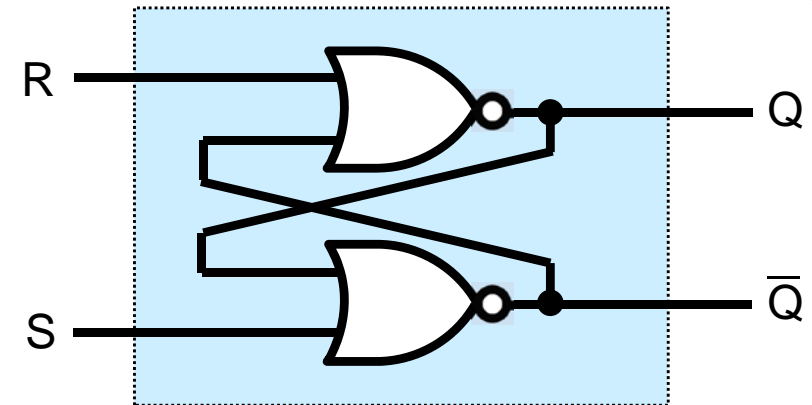
tema 6:  
Implementación de sistemas secuenciales síncronos

FC

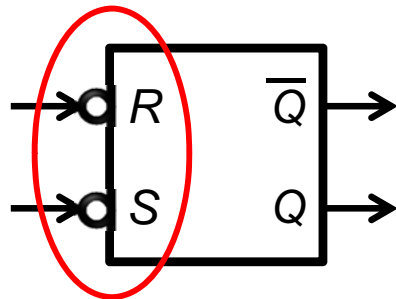
5



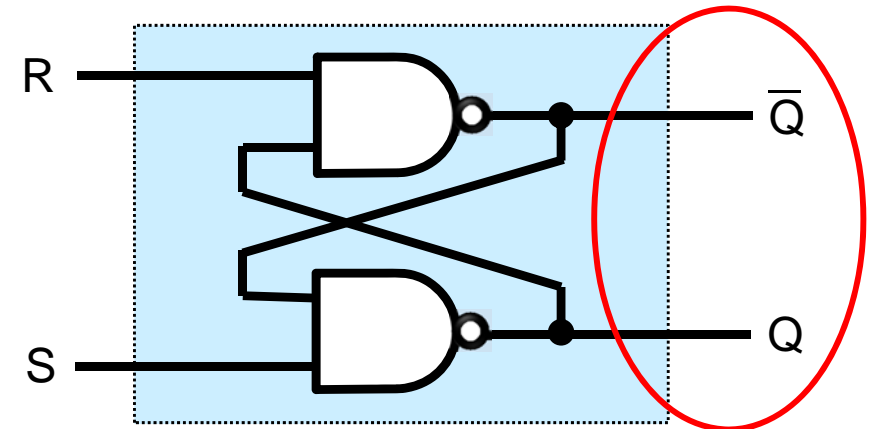
$R(t)$	$S(t)$	$Q(t+\Delta t)$
0	0	$Q(t)$
0	1	1
1	0	0
1	1	prohibido



Biastable SR asíncrono  
(implementación con NOR)



$R(t)$	$S(t)$	$Q(t+\Delta t)$
0	0	prohibido
0	1	0
1	0	1
1	1	$Q(t)$



Biastable SR asíncrono  
(implementación con NAND)



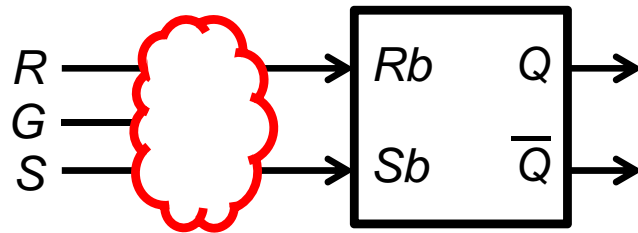
# Biastable SR síncrono (por nivel)

versión 12/09/14

tema 6:  
Implementación de sistemas secuenciales síncronos

FC

6



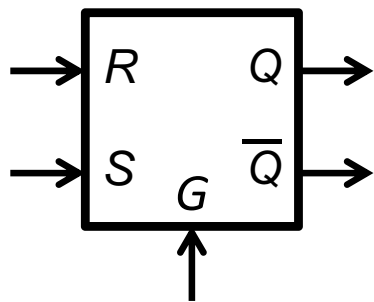
$$Rb/Sb = \begin{cases} 0 & \text{si } G=0 \\ R/S & \text{si } G=1 \end{cases}$$

G	R	Rb
0	0	0
0	1	0
1	0	0
1	1	1

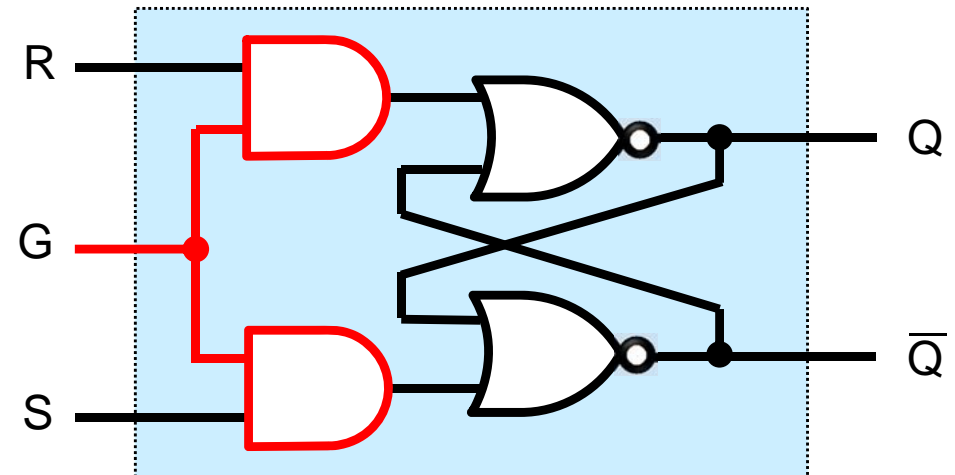
$$Rb = G \cdot R$$

G	S	Sb
0	0	0
0	1	0
1	0	0
1	1	1

$$Sb = G \cdot S$$



G(t)	R(t)	S(t)	Q(t+Δt)
0	X	X	Q(t)
1	0	0	Q(t)
1	0	1	1
1	1	0	0
1	1	1	prohibido



Biastable SR síncrono disparado por nivel  
(Latch SR)

# Biestable SR síncrono (por nivel)

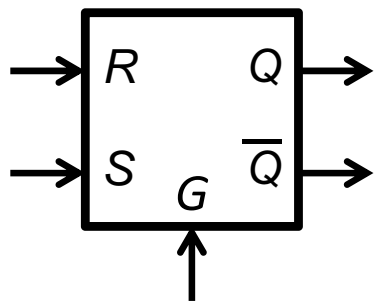


versión 12/09/14

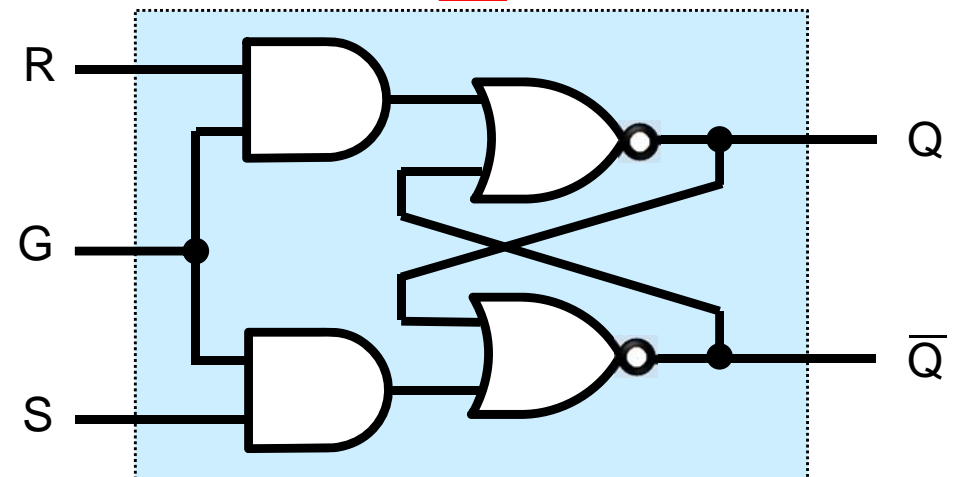
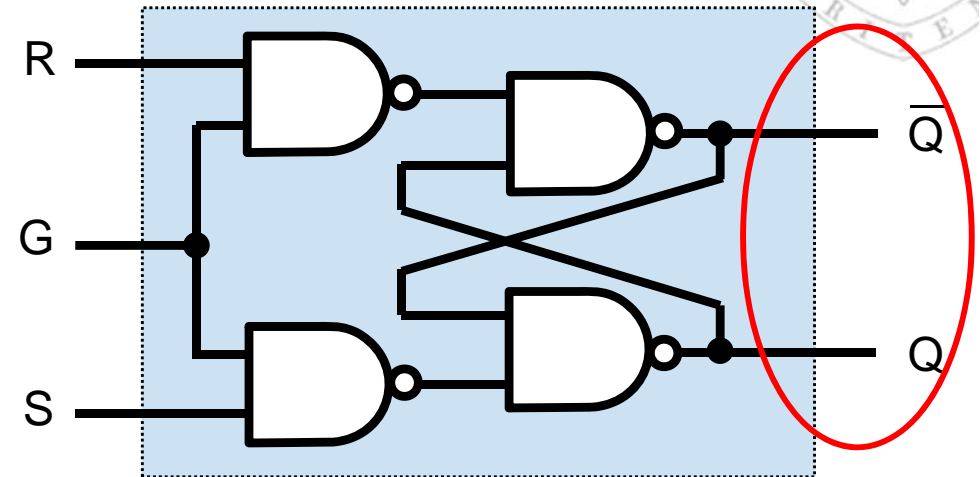
tema 6:  
Implementación de sistemas secuenciales síncronos

FC

7



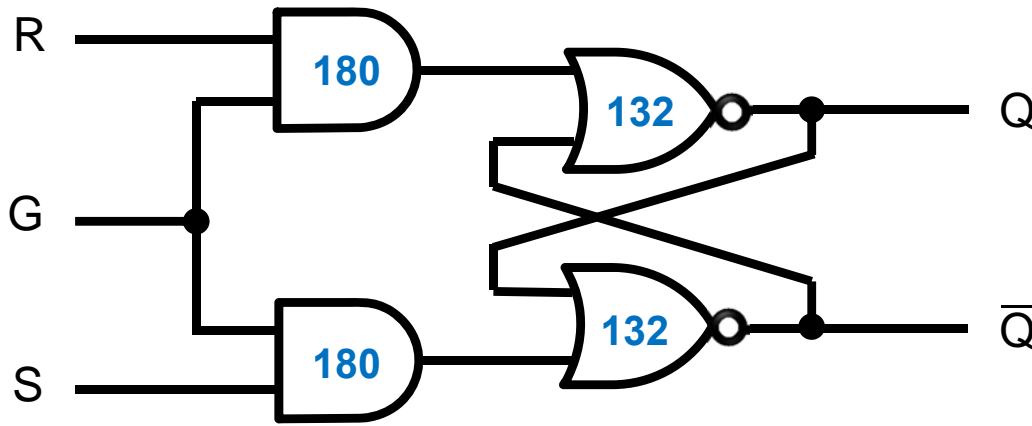
$G(t)$	$R(t)$	$S(t)$	$Q(t+\Delta t)$
0	X	X	$Q(t)$
1	0	0	$Q(t)$
1	0	1	1
1	1	0	0
1	1	1	prohibido



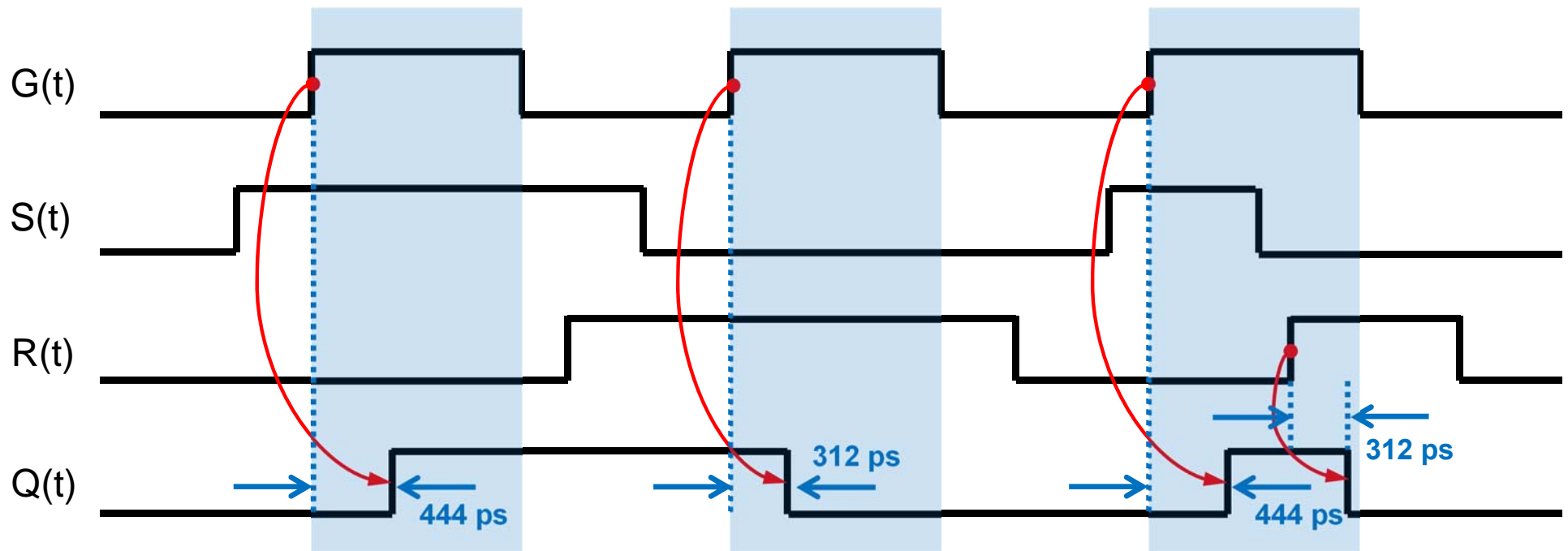
Biestable SR síncrono disparado por nivel  
(Latch SR)



# Biastable SR síncrono (por nivel)



R(t)	S(t)	G(t)	Q(t+ $\Delta t$ )
X	X	0	Q(t)
0	0	1	Q(t)
0	1	1	1
1	0	1	0
1	1	1	prohibido







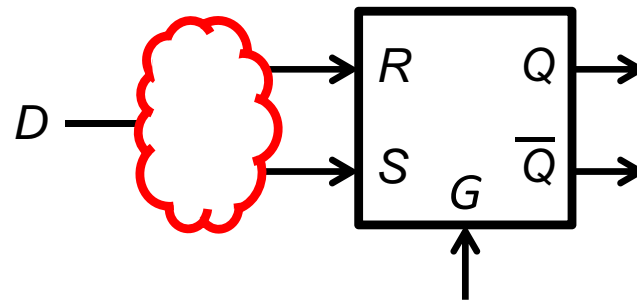
# Biastable D síncrono (por nivel)

versión 12/09/14

tema 6:  
Implementación de sistemas secuenciales síncronos

FC

9

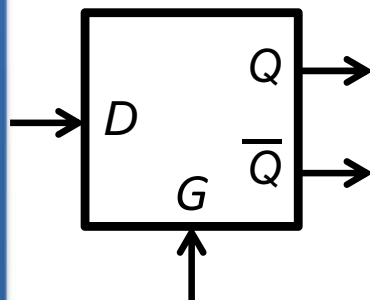


D	R
0	1
1	0

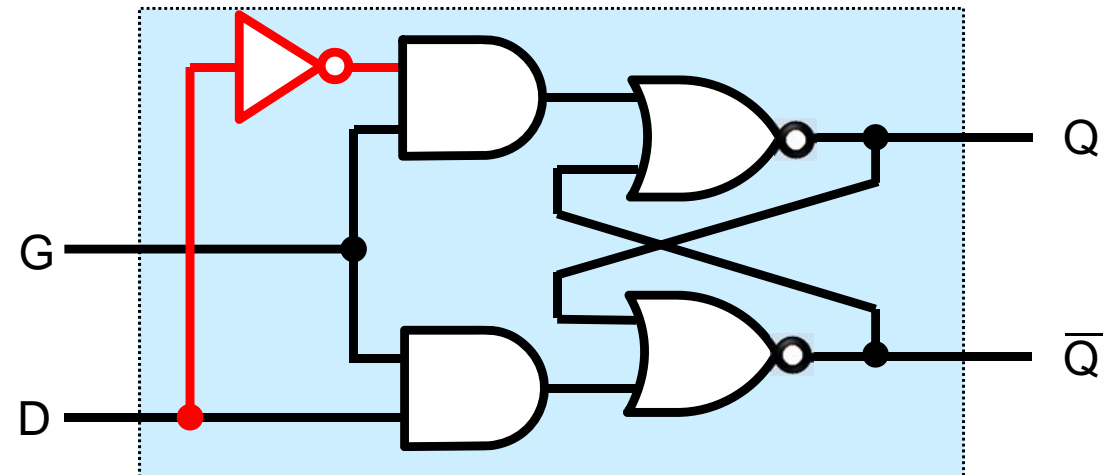
$$R = \bar{D}$$

D	S
0	0
1	1

$$S = D$$



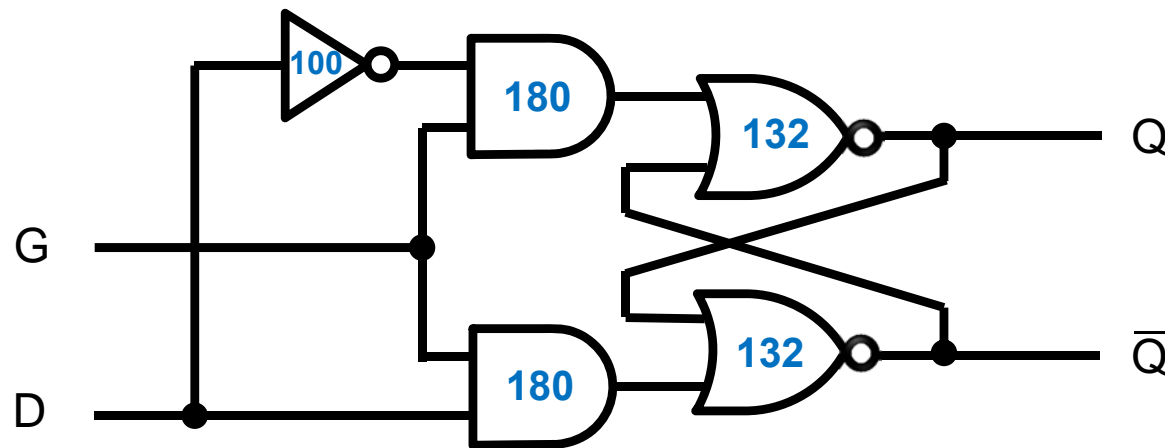
G(t)	D(t)	Q(t+Δt)
0	X	Q(t)
1	0	0
1	1	1



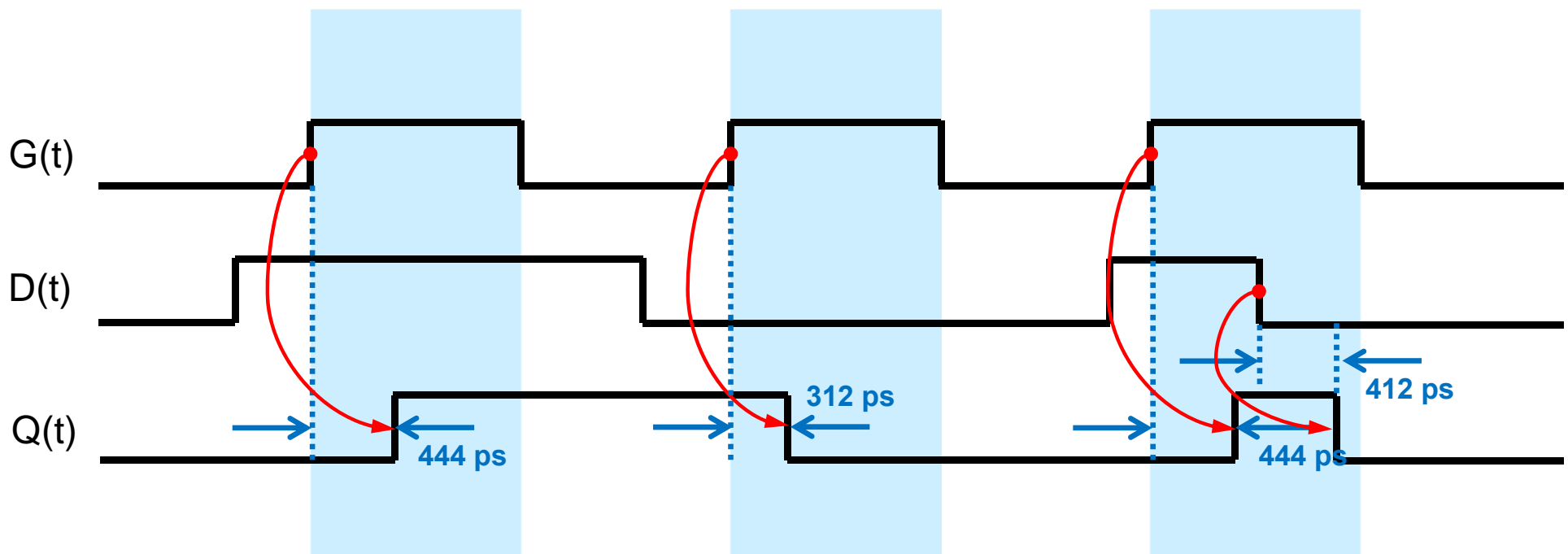
Biastable D síncrono disparado por nivel  
(Latch D)



# Biastable D síncrono (por nivel)



G(t)	D(t)	Q(t+Δt)
0	X	Q(t)
1	0	0
1	1	1





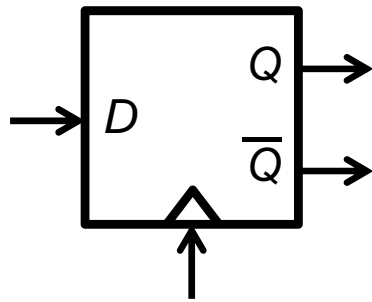
# Biastable D síncrono (por flanco)

versión 12/09/14

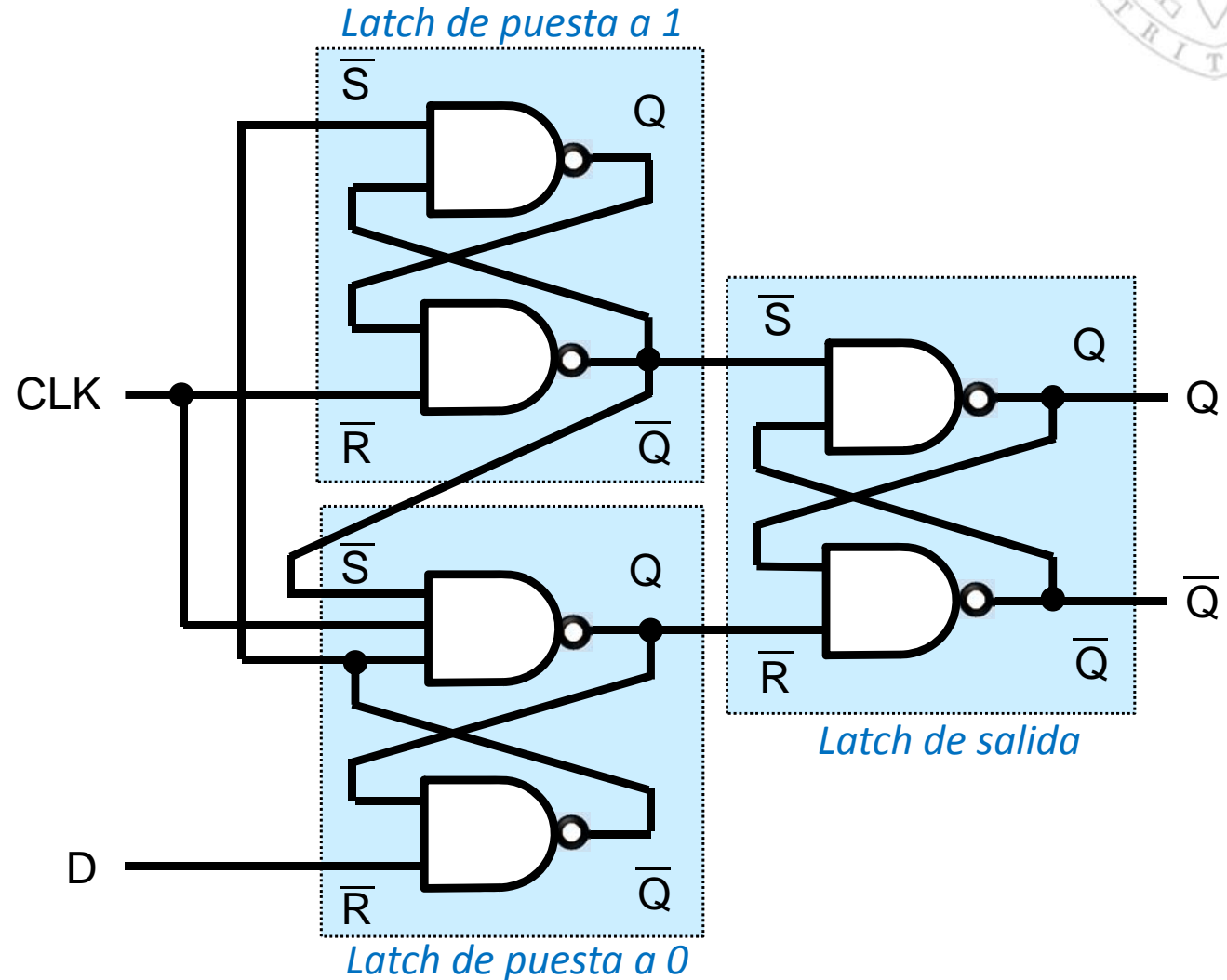
tema 6:  
Implementación de sistemas secuenciales síncronos

FC

11



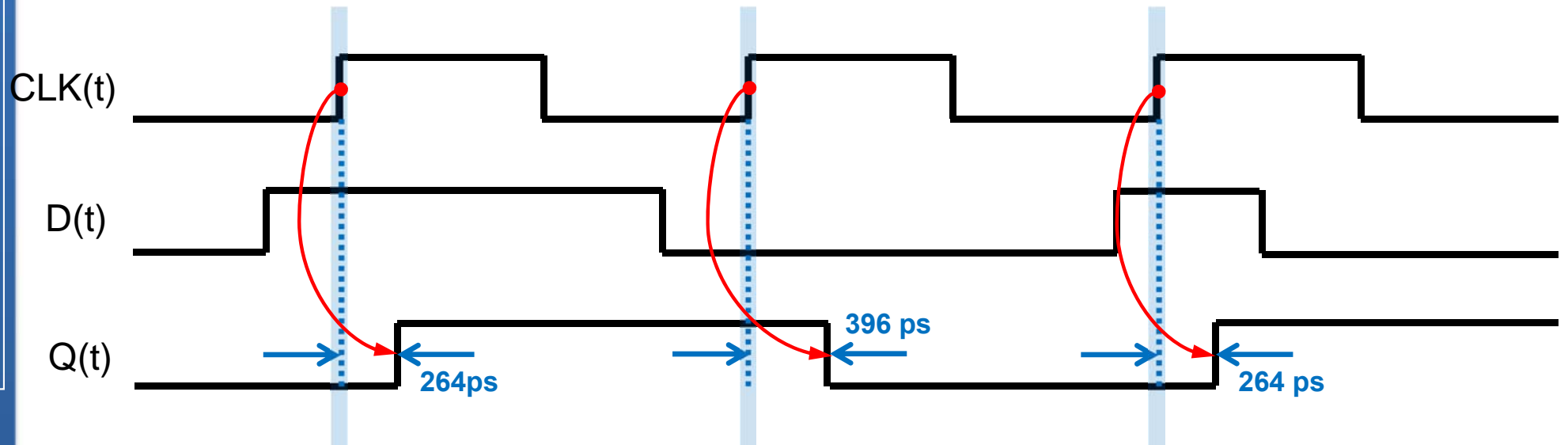
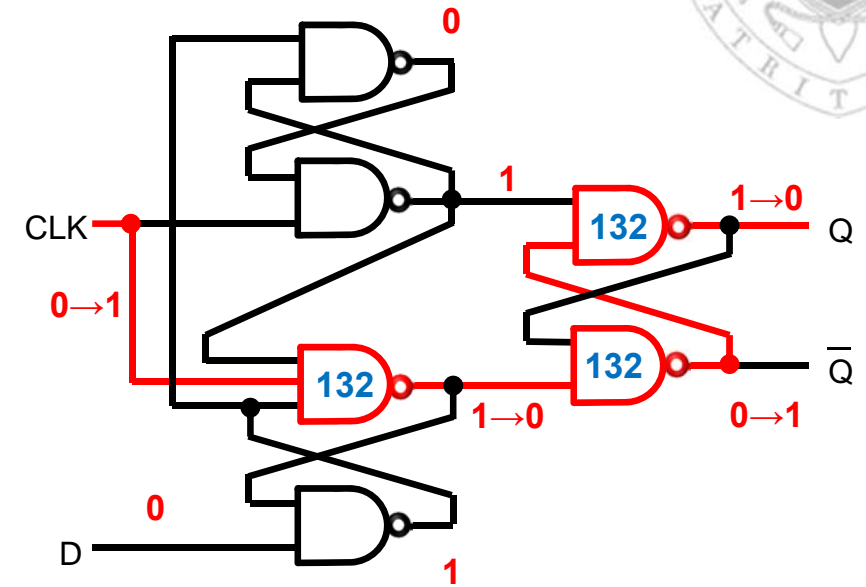
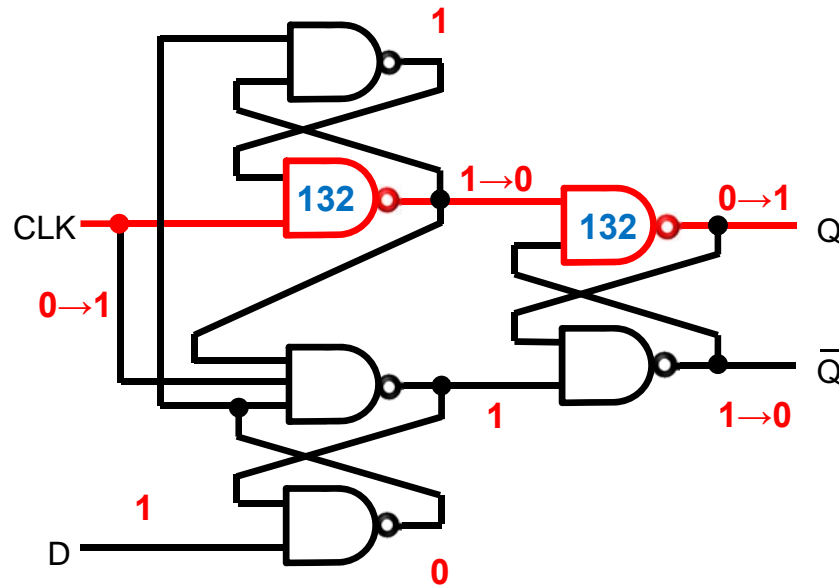
D(t)	CLK	Q(t+1)
0	↑	0
1	↑	1
resto		Q(t)



Biastable D síncrono disparado por flanco de subida  
(Flip-flop D, implementación con NAND)



# Biestable D síncrono (por flanco)





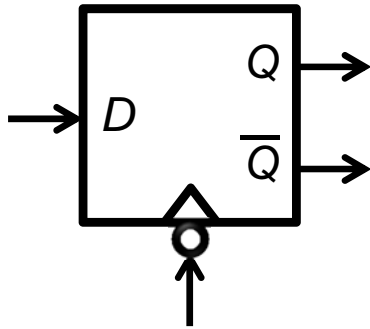
# Biestable D síncrono (por flanco)

versión 12/09/14

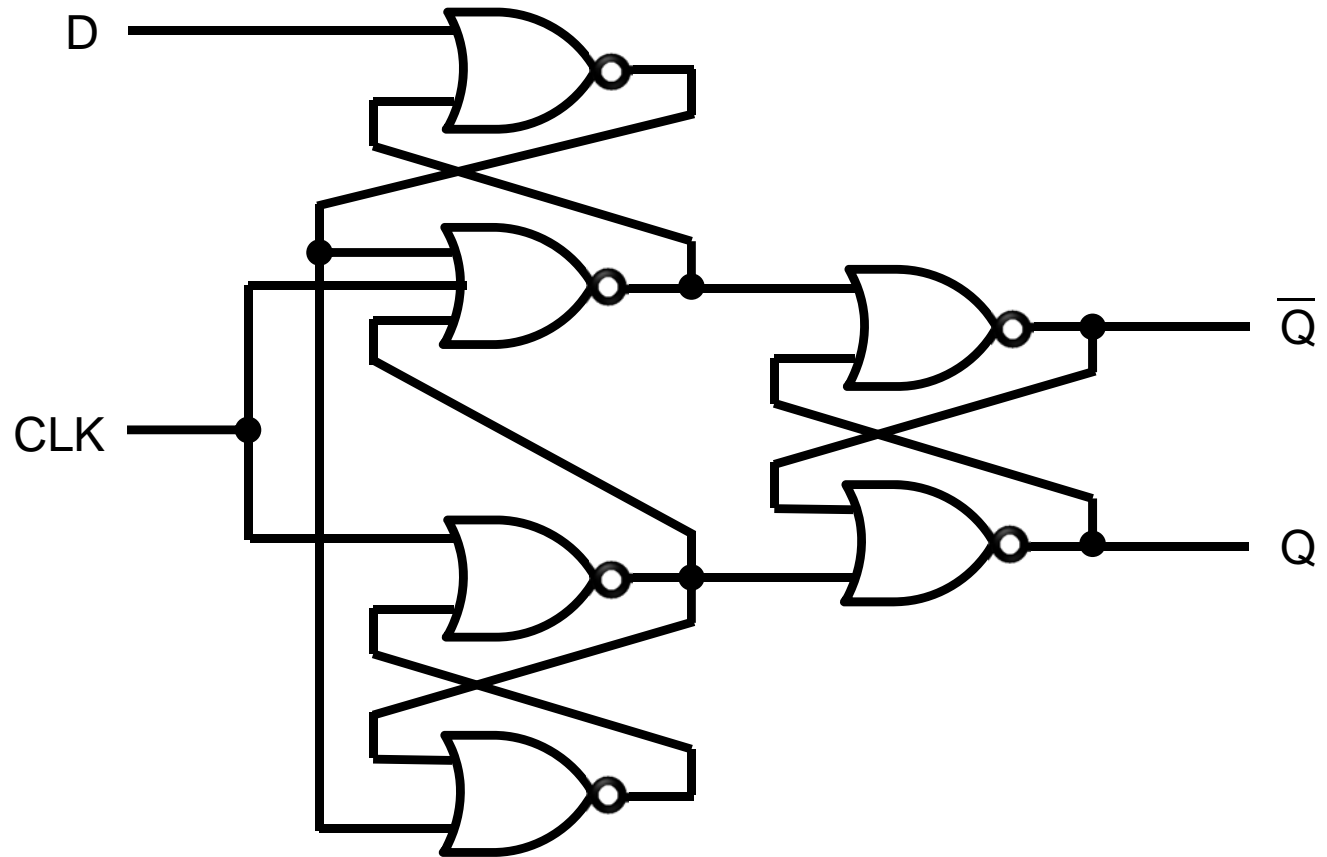
tema 6:  
Implementación de sistemas secuenciales síncronos

FC

13



D(t)	CLK	Q(t+1)
0	↓	0
1	↓	1
resto		Q



Biestable D síncrono disparado por flanco de bajada  
(Flip-flop D, implementación con NOR)



# Síntesis con biestables D

- Dada una especificación de una conducta secuencial implementarla como una red de módulos combinacionales y biestables D, en donde:
  - Todos los biestables se conectan a una señal de reloj periódica.
  - Todos los biestables se disparan por flancos de la misma polaridad.
  - Toda realimentación incluye al menos un biestable.
- Implementación canónica: realización directa de un diagrama de estados:
  - El registro de estado se implementa como un array de biestables D disparados por flanco (todos de la misma polaridad) con reloj común.
  - 2 bloques de lógica combinacional implementan las funciones de salida y de transición de estados.

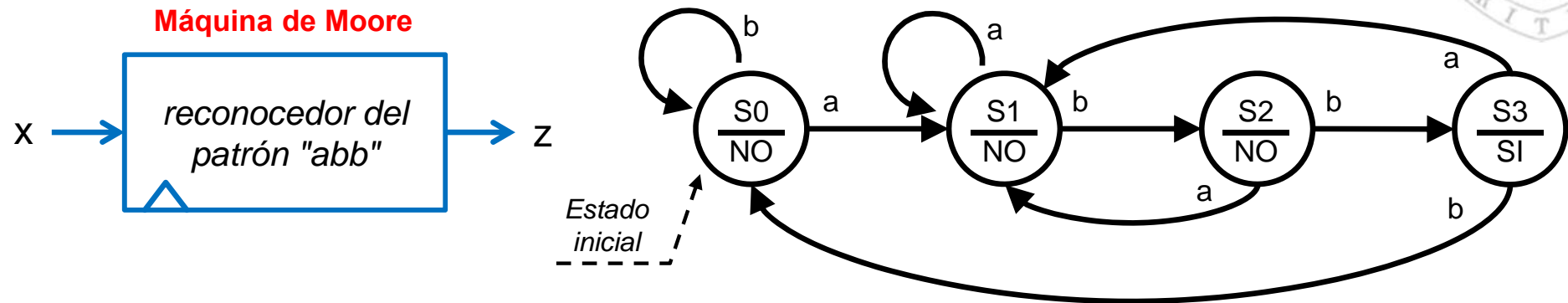


# Síntesis con biestables D

- Este método de síntesis sigue un modelo de **temporización síncrona por flanco** de reloj global, en donde:
  - Los cálculos que realiza el sistema se realizan **ciclo a ciclo**.
  - Las **fronteras del ciclo** están marcadas por las transiciones de igual polaridad en el reloj común.
  - Al **comienzo del ciclo**, el sistema hace un cambio de estado mediante la **actualización simultánea de todos los biestables**.
  - El **nuevo estado provoca transiciones** en las entradas de los módulos combinacionales que a su vez provocarán transiciones en sus salidas.
  - El cálculo a realizar en el ciclo finaliza cuando todos los **sistemas combinacionales han alcanzado su régimen permanente**.
  - Los valores permanentes a **la salida de los módulos combinacionales son utilizados para actualizar los biestables** al comienzo del ciclo siguiente.



# Síntesis con biestables D



- Codificación domino:  $\{ a \rightarrow 0, b \rightarrow 1 \}$
- Codificación codominio:  $\{ NO \rightarrow 0, SI \rightarrow 1 \}$
- Codificación estados:  $\{ S0 \rightarrow (00), S1 \rightarrow (01), S2 \rightarrow (10), S3 \rightarrow (11) \}$

Función de transición  
de estados

$x$	$q_1$	$q_0$	$q_1'$	$q_0'$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Función de salida

$q_1$	$q_0$	$z$
0	0	0
0	1	0
1	0	0
1	1	1

$$q_1' = x(q_1 \oplus q_0)$$

$$q_0' = \bar{x} + q_1 \bar{q}_0$$

$$z = q_1 q_0$$





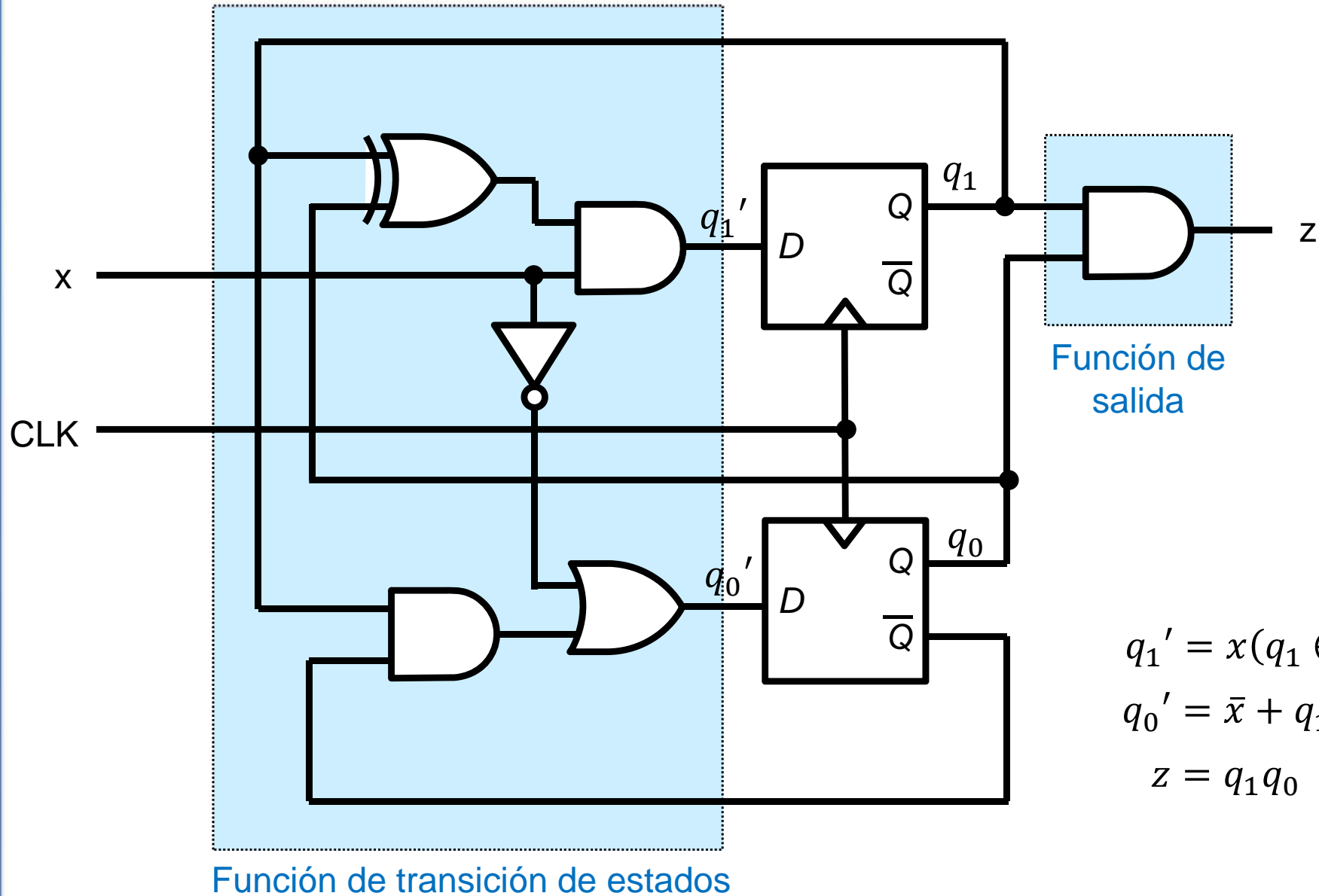
# Síntesis con biestables D

versión 12/09/14

tema 6:  
Implementación de sistemas secuenciales síncronos

FC

17





# Síntesis con biestables D

- Codificaciones distintas dan lugar a implementaciones diferentes de la misma máquina de estados.
  - Por ello es interesante elegir aquella codificación que reduzca al máximo el coste/retardo de los circuitos de transición y salida.
- Codificación domino:  $\{ a \rightarrow 0, b \rightarrow 1 \}$
- Codificación codominio:  $\{ NO \rightarrow 0, SI \rightarrow 1 \}$
- Codificación estados:  $\{ S0 \rightarrow (01), S1 \rightarrow (00), S2 \rightarrow (10), S3 \rightarrow (11) \}$

Función de transición  
de estados

x	q <sub>1</sub>	q <sub>0</sub>	q <sub>1</sub> '	q <sub>0</sub> '
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

Función de salida

q <sub>1</sub>	q <sub>0</sub>	z
0	0	0
0	1	0
1	0	0
1	1	1

requiere 2 puertas menos

$$q_1' = x\overline{q_0}$$

$$q_0' = x(q_0 + q_1)$$

$$z = q_1q_0$$



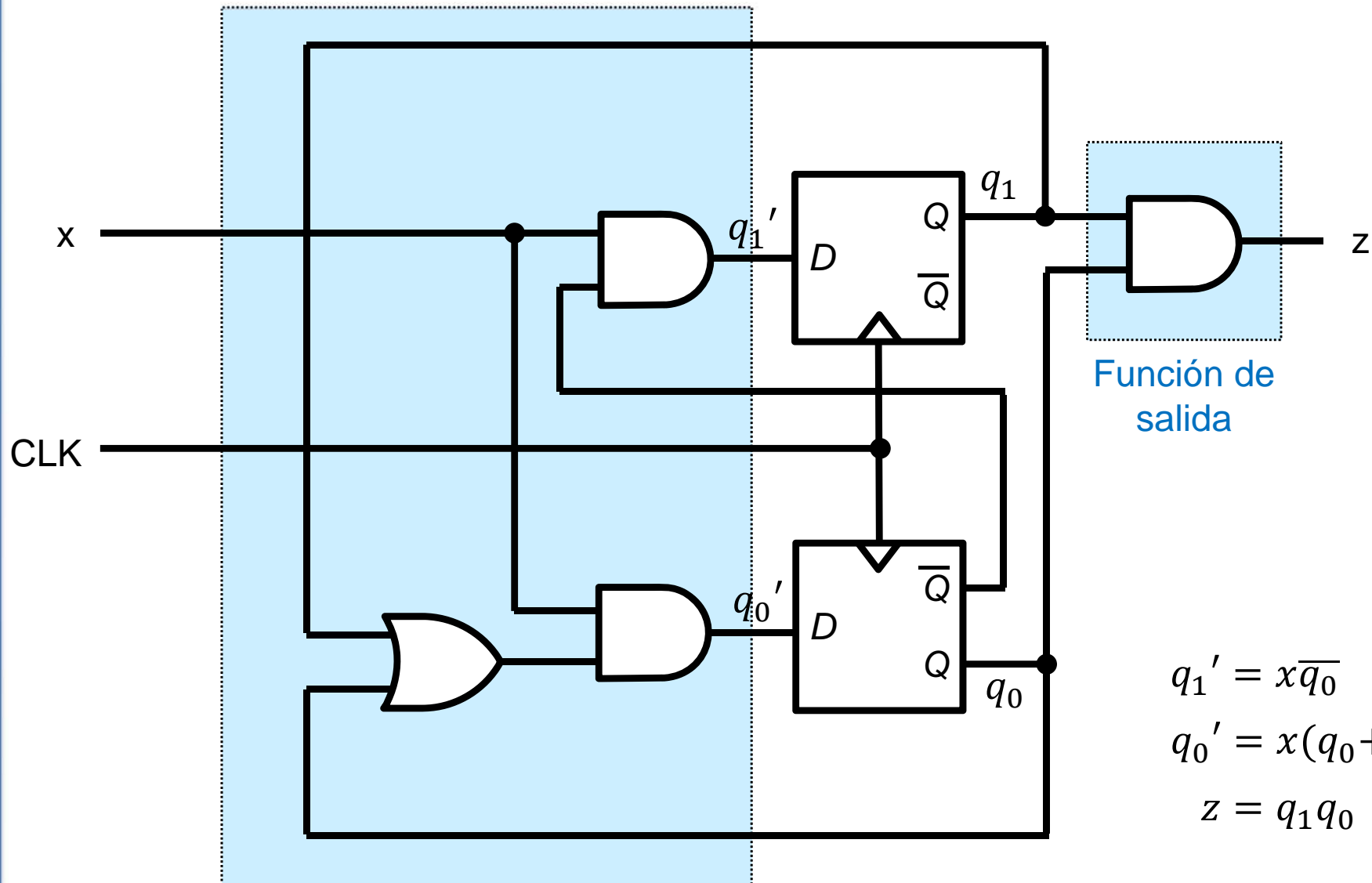
# Síntesis con biestables D

versión 12/09/14

tema 6:  
Implementación de sistemas secuenciales síncronos

FC

19



$$q_1' = x\overline{q_0}$$

$$q_0' = x(q_0 + q_1)$$

$$z = q_1q_0$$



# Inicialización

- ¿Cual es el estado de un biestable al encenderlo?
  - Cualquiera de los 2 posibles.
- ¿Cual es el estado de un sistema secuencial al encenderlo?
  - Cualquiera de los posibles (**incluyendo prohibidos**).
- Todos los sistemas secuenciales tienen una **entrada de inicialización** para llevarlos **asíncronamente** a un estado inicial.
  - Esta **entrada global de reset** deberá conectarse **según la codificación del estado inicial** a la **entrada de inicialización** que corresponda de cada uno de los biestables del circuito.



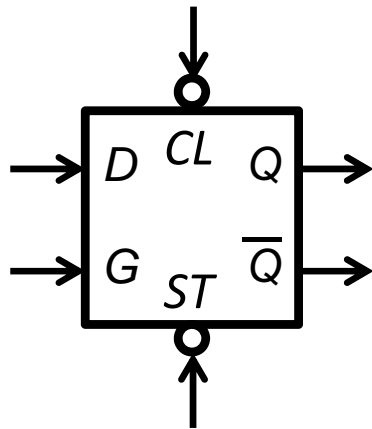
# Inicialización

versión 12/09/14

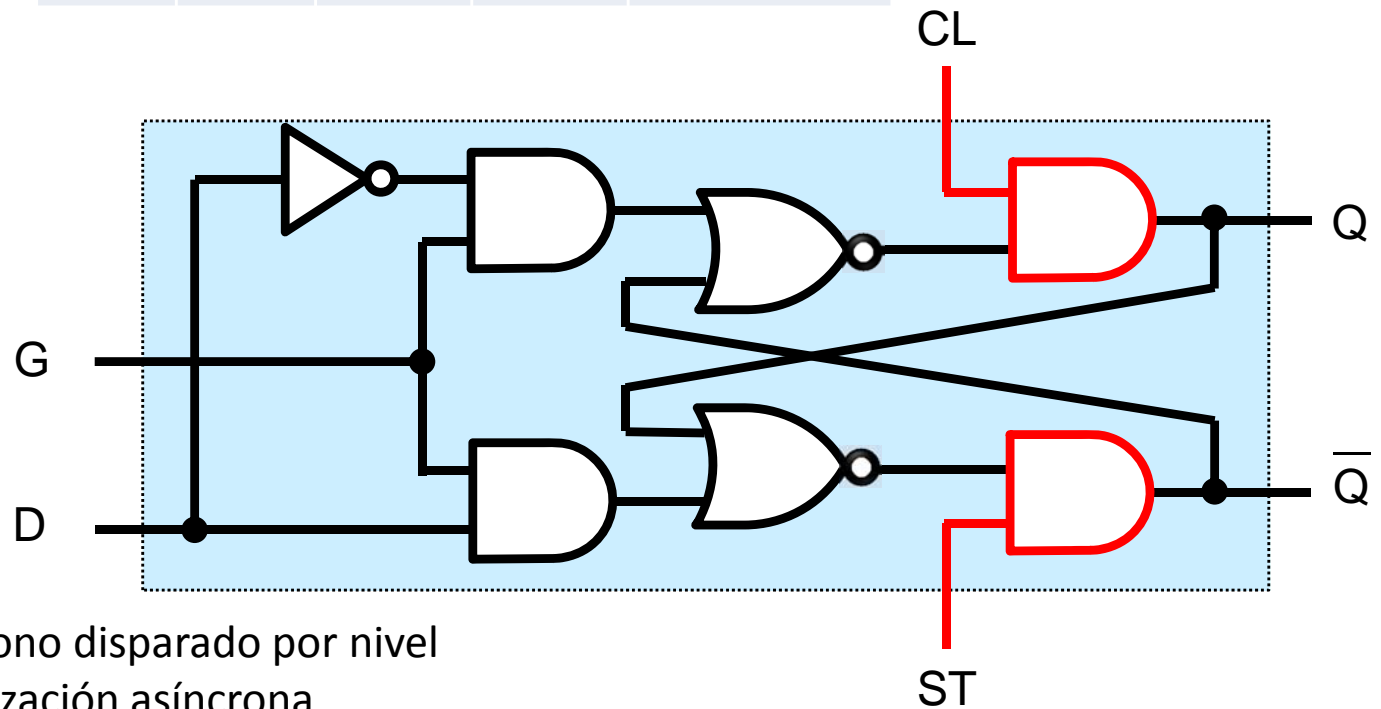
tema 6:  
Implementación de sistemas secuenciales síncronos

FC

21



$G(t)$	$D(t)$	$CL(t)$	$ST(t)$	$Q(t+\Delta t)$
X	X	0	0	prohibido
X	X	0	1	0
X	X	1	0	1
0	X	1	1	$Q(t)$
1	0	1	1	0
1	1	1	1	1



Biastable D síncrono disparado por nivel  
con inicialización asíncrona



# Inicialización

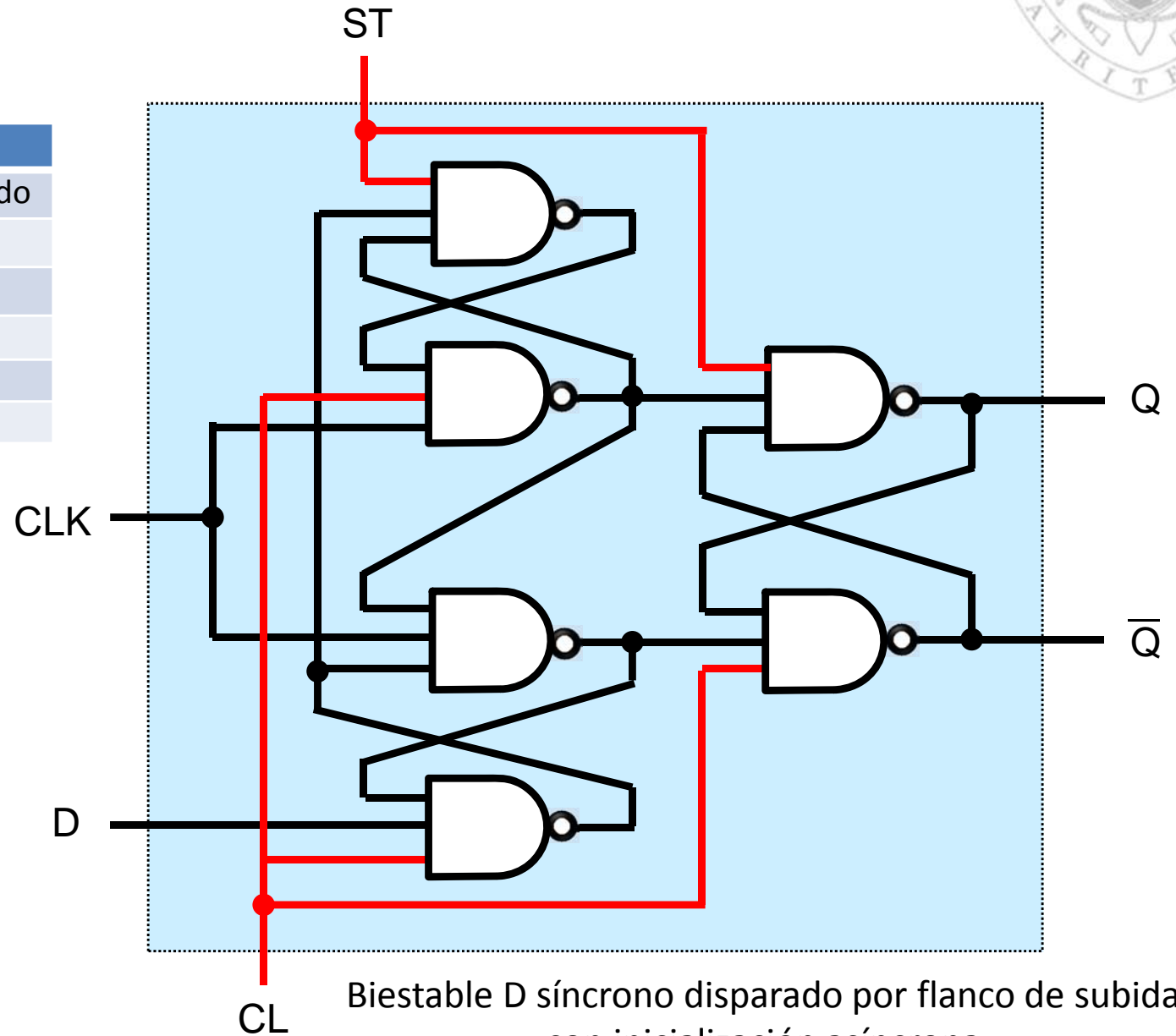
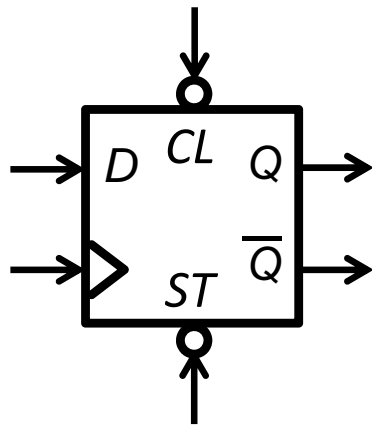
versión 12/09/14

tema 6:  
Implementación de sistemas secuenciales síncronos

FC

22

D	CLK	CL	ST	Q'
X	X	0	0	prohibido
X	X	0	1	0
X	X	1	0	1
0	↑	1	1	0
1	↑	1	1	1
resto		1	1	Q

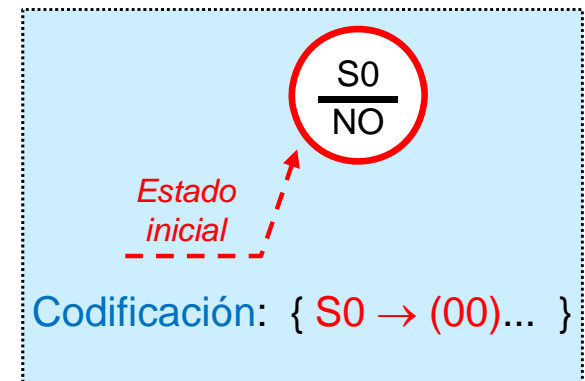
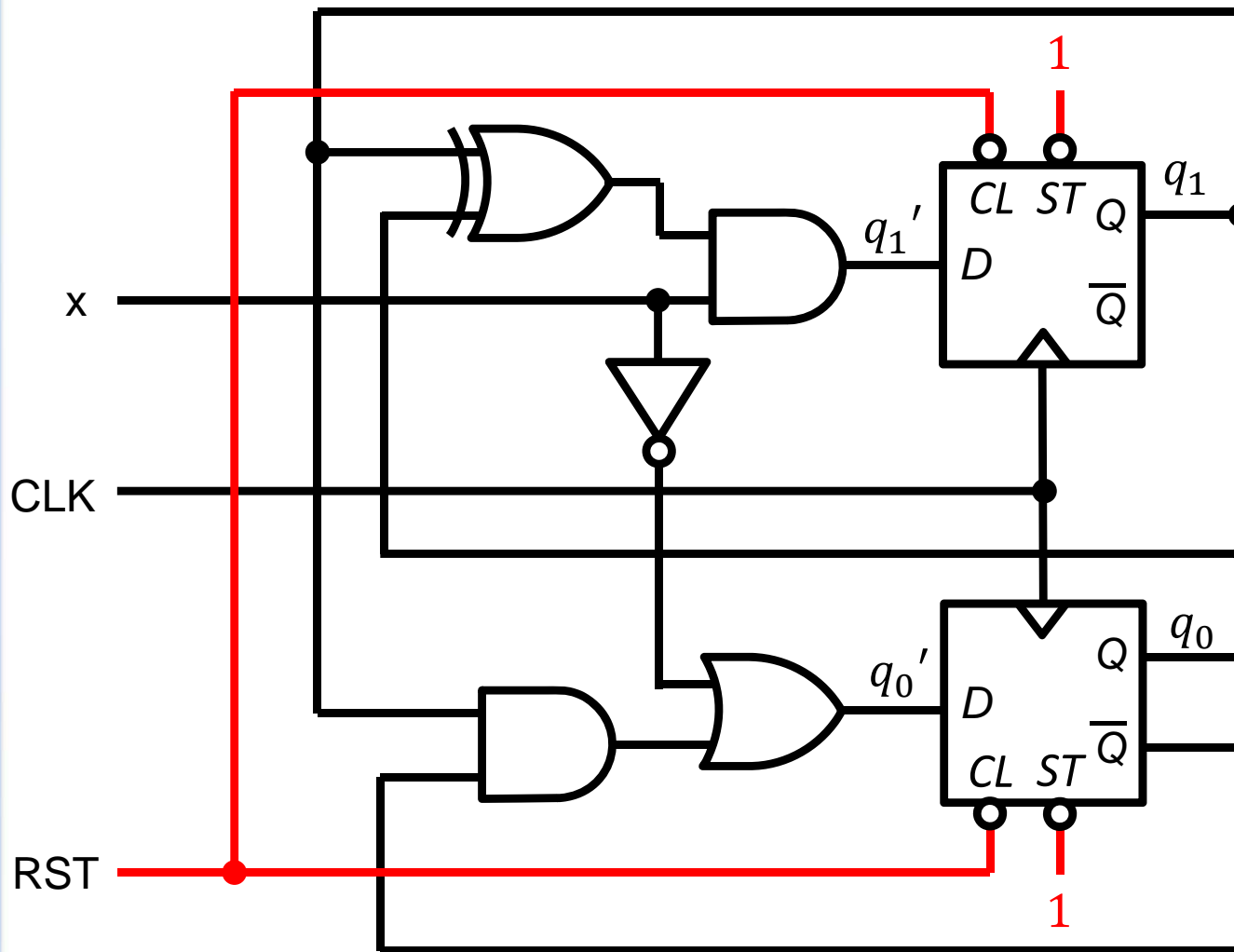


Biestable D síncrono disparado por flanco de subida  
con inicialización asíncrona



# Inicialización

- El **reset se distribuye** según la codificación del estado inicial:



$$q_1' = x(q_1 \oplus q_0)$$

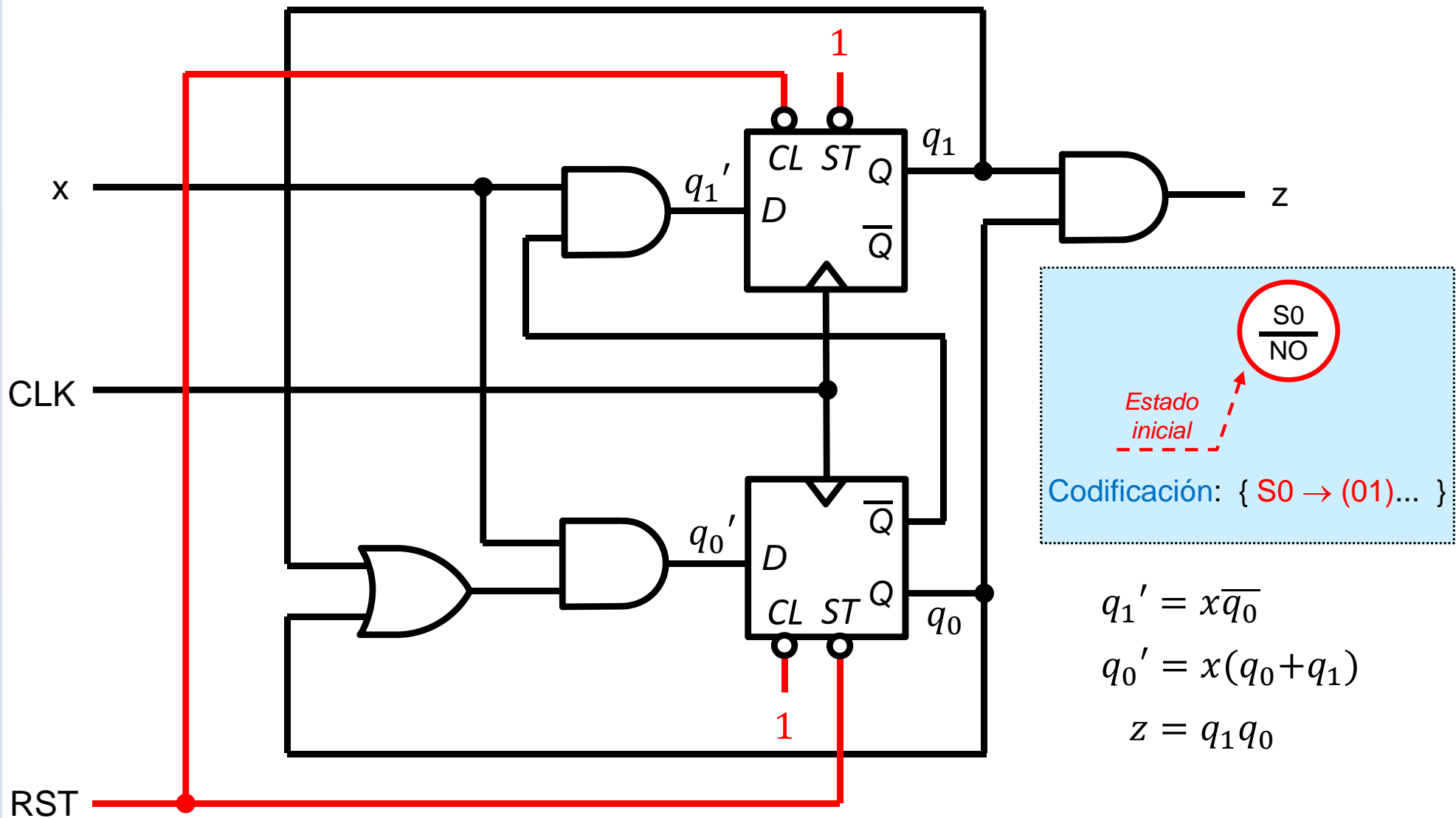
$$q_0' = \bar{x} + q_1 \bar{q}_0$$

$$z = q_1 q_0$$



# Inicialización

- El **reset se distribuye** según la codificación del estado inicial.



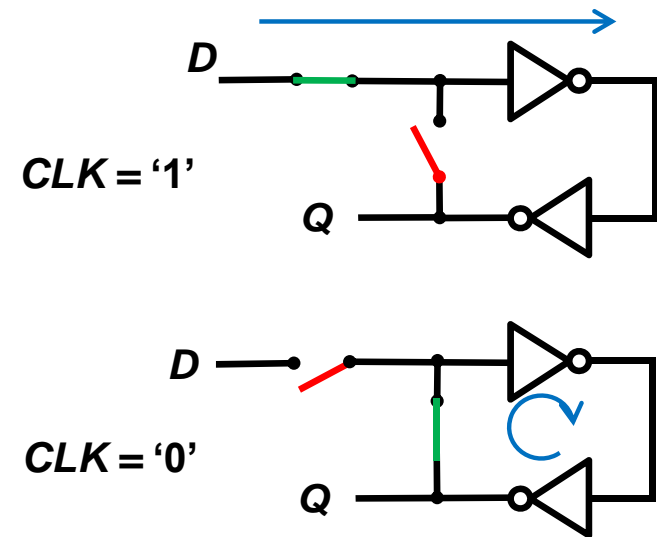
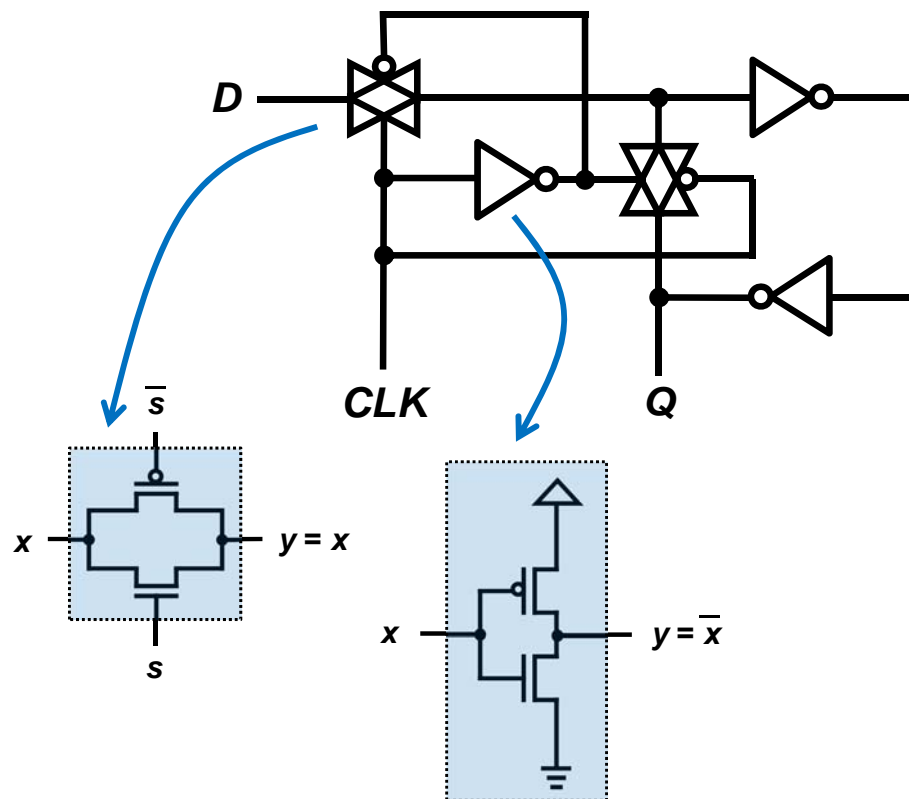


# Aspectos tecnológicos

## Implementación de biestables

- Un **biestable CMOS** suele implementarse realimentando parejas de inversores a través de puestas de paso.

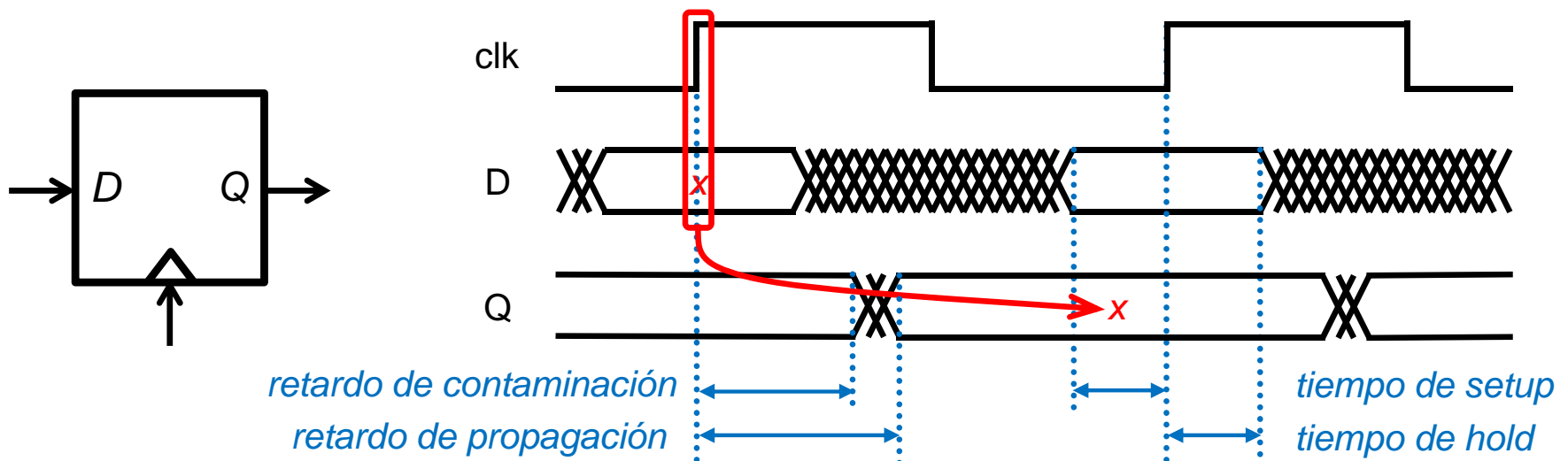
Latch D CMOS



# Aspectos tecnológicos

## Retardo

- En un **biestable disparado por flanco**, las salidas cambian en respuesta al flanco del reloj y **NO** al cambio de la entrada.
  - los retardos se miden desde dicho flanco, el máximo se denomina **retardo de propagación** y el mínimo, **retardo de contaminación**.
- Además, para que tenga un **comportamiento predecible**, la entrada debe estar estable en las proximidades del flanco:
  - Como mínimo debe estar estable durante el **tiempo de setup** (antes del flanco) y durante **tiempo de hold** (después del flanco).

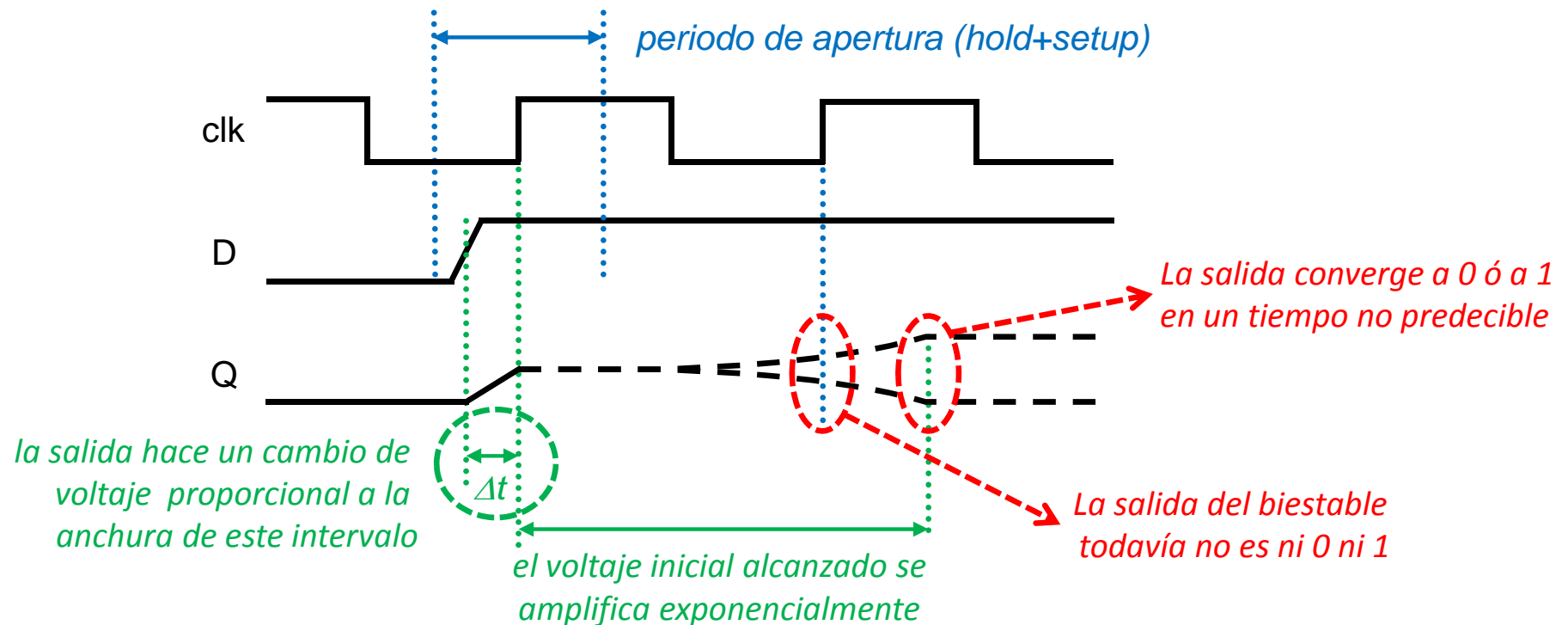




# Aspectos tecnológicos

## Metaestabilidad

- Cuando se **viola el tiempo de hold o el de setup**, el biestable entra en un estado **metaestable** caracterizado por:
  - El retardo de propagación del biestable no está acotado.
  - El valor de salida del biestable es impredecible y, por tanto, se propagan en cadena valores inconsistentes por todo el circuito.

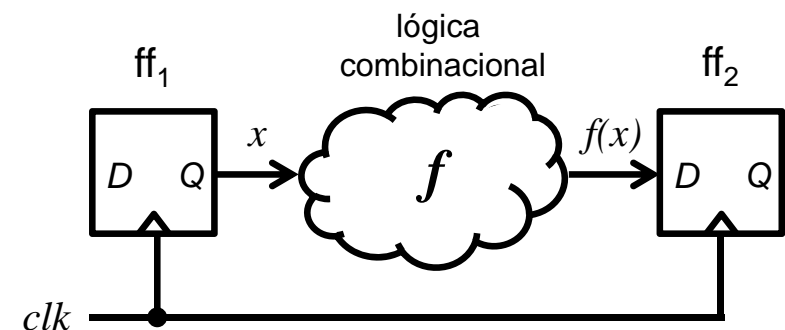
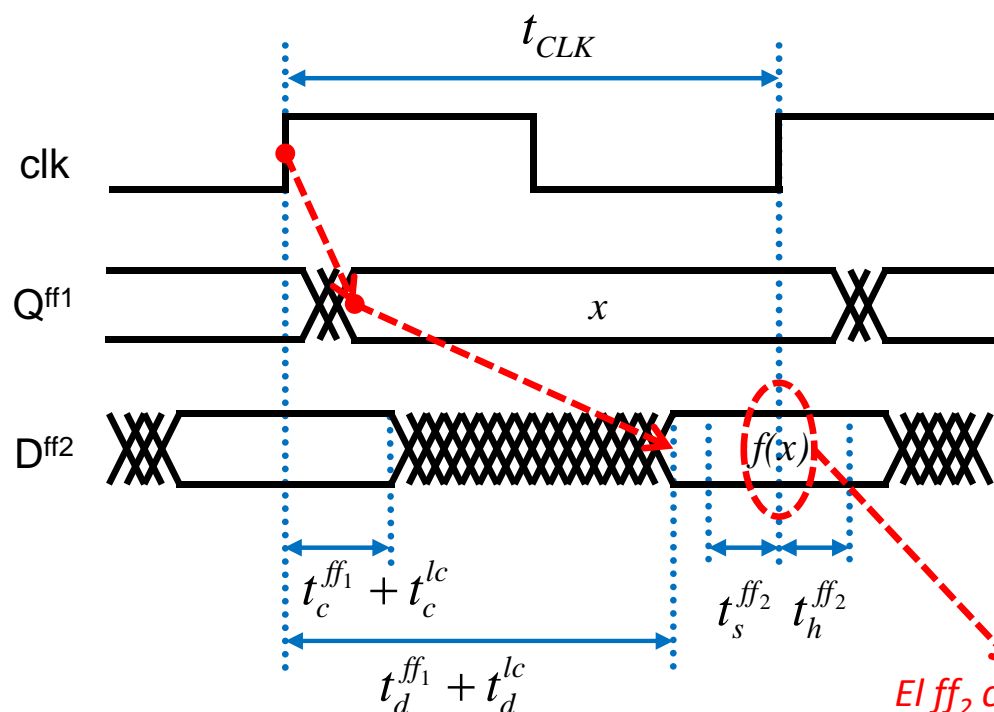




# Aspectos tecnológicos

## Reglas de diseño

- Por tanto, un sistema secuencial con temporización por flanco tendrá un **comportamiento correcto** si:
  - El **tiempo de ciclo del reloj** es lo suficientemente largo para que todos los sistemas combinatoriales alcancen su **régimen permanente**.
  - Las **entradas de todos los biestables permanecen estables** durante su **periodo de apertura** (hold+setup).



**ligadura de retardo máximo:**

$$t_{CLK} \geq (t_d^{ff1} + t_c^{lc} + t_s^{ff2})$$

**ligadura de retardo mínimo:**

$$(t_c^{ff1} + t_c^{lc}) \geq t_h^{ff2}$$

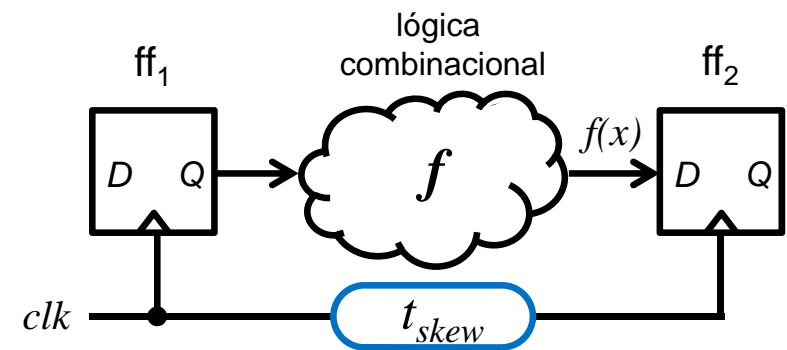
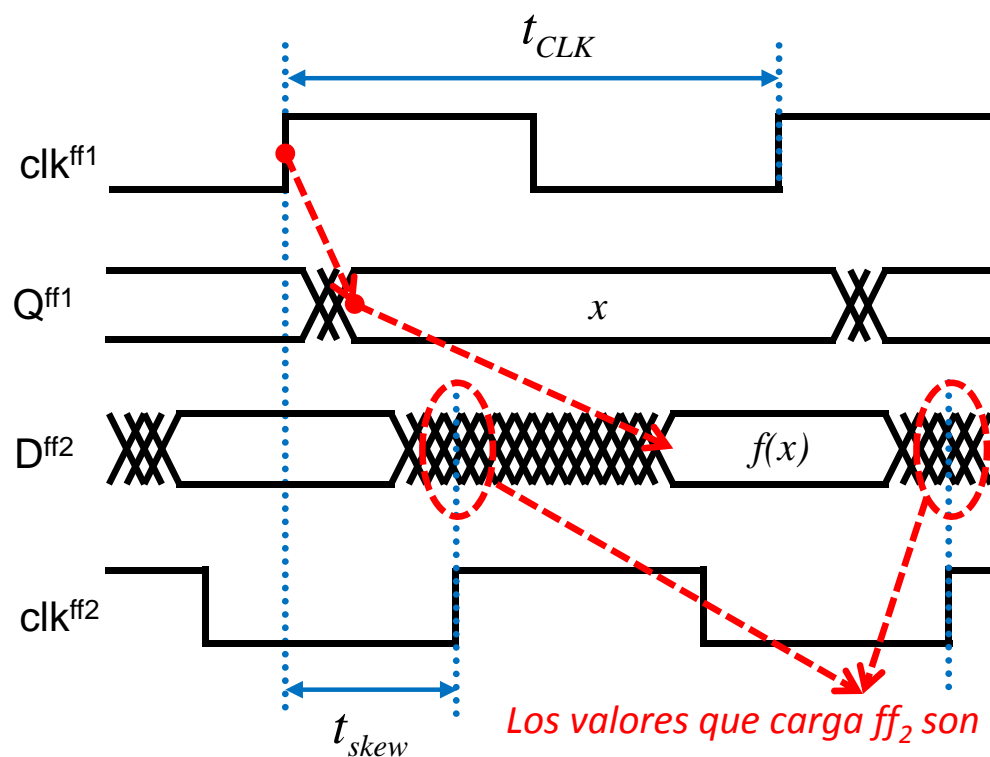
El ff<sub>2</sub> carga f(x)



# Aspectos tecnológicos

## Reglas de diseño

- No obstante, un sistema correctamente temporizado **puede fallar** si la señal de reloj no se distribuye adecuadamente:
  - Si la señal de reloj llega con retraso (**skew**) a las entradas de reloj de algunos flip-flops, el sistema se desincroniza.
  - Ídem si la frecuencia del reloj no es perfectamente regular (**jitter**).



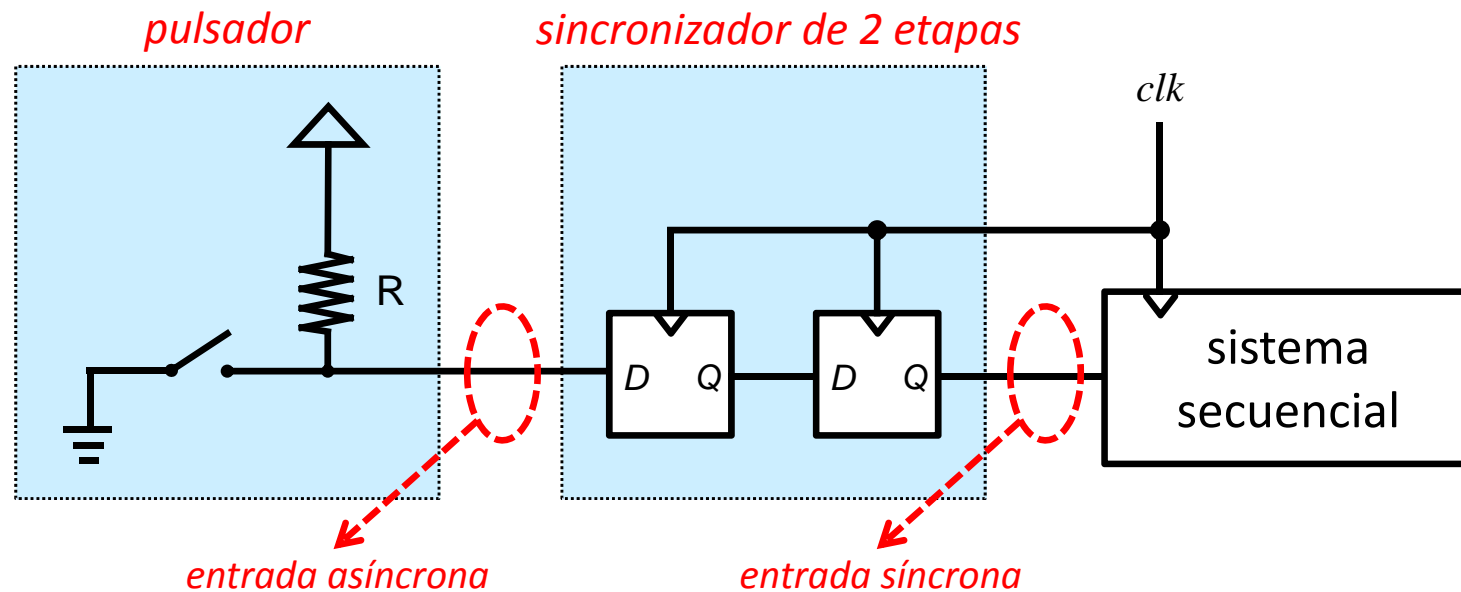
- Distinta longitud de cable
- Ruido (interferencias)
- Diferente carga local
- Variaciones locales de temperatura
- etc...



# Aspectos tecnológicos

## Reglas de diseño

- Asimismo, un sistema también puede fallar si sus **entradas cambian asíncronamente** porque están conectadas a:
  - Las salidas de otro sistema con distinto reloj.
  - Un dispositivo puramente asíncrono (i.e pulsador)
- En ambos casos es necesario utilizar **sincronizadores**.
  - El más básico consiste en conectar varios biestables D en cascada.

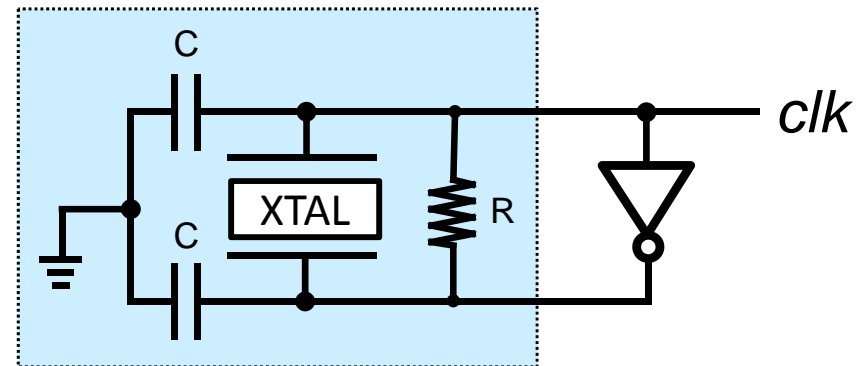
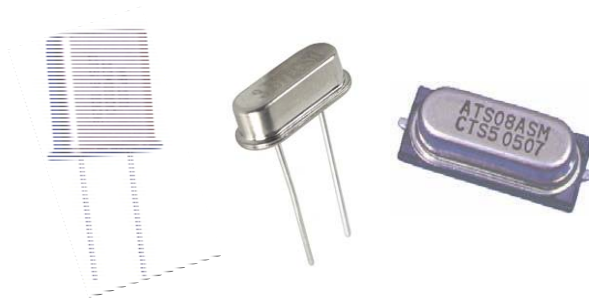


# Aspectos tecnológicos

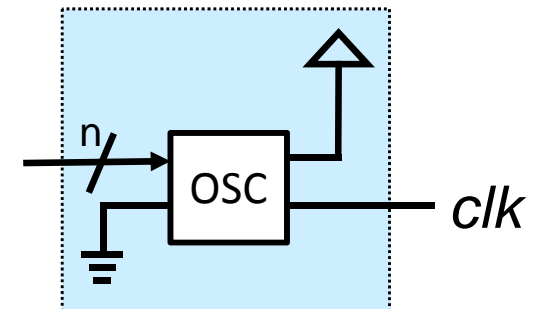
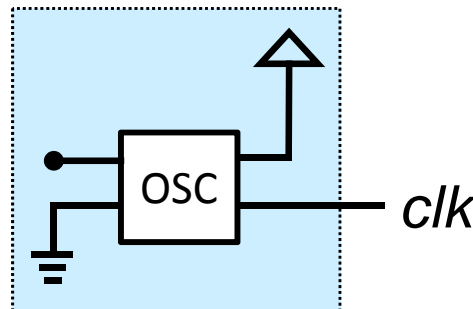
## Generación de reloj

- La señal de **reloj primaria** se genera externamente usando:

- Cristal de cuarzo



- Oscilador integrado con frecuencia fija o programable



- Dentro del chip esta señal se acondiciona y se multiplica/divide para obtener la frecuencia deseada.





# Aspectos tecnológicos

## Generación de reloj

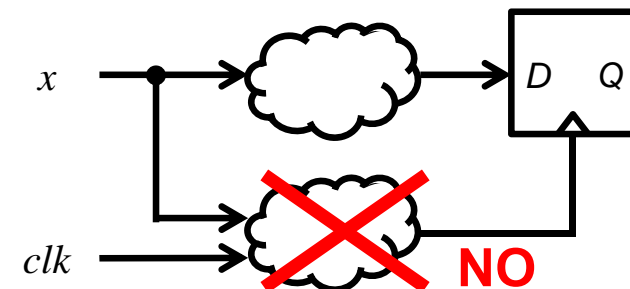
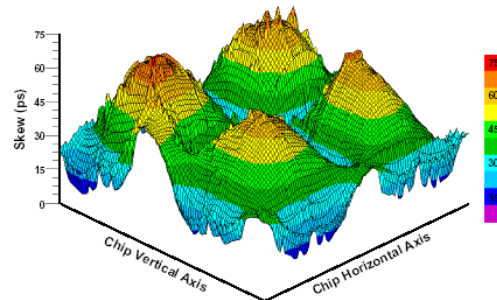
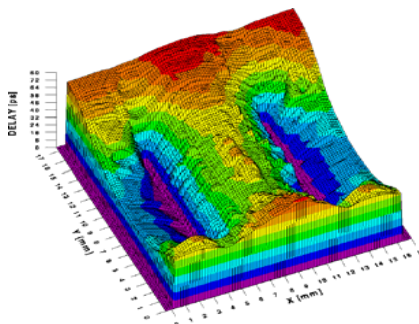
- La frecuencia (periodo) de reloj aumenta (disminuye) en cada generación tecnológica
- Evolución de las frecuencia de reloj de procesadores Intel
  - 4004 (1971): 108 KHz (periodo 9.25  $\mu$ s)
  - 8086 (1978): 4.77 MHz (periodo 0.21  $\mu$ s)
  - 80386 (1985): 16 MHz (periodo 62.5 ns)
  - Pentium (1993): 66 MHz (periodo 15.2 ns)
  - Pentium 4 (2000): 1.5 GHz (periodo 667 ps)
  - Intel Core 2 Quad (2007): 2.4 GHz (periodo 417 ps)
  - Intel Core I7-4770 (2013): 3.4 GHz (periodo 294 ps)
- Si la velocidad de los coches hubiera crecido tan rápido:
  - Coche (1971): 100 Km/h (Madrid-Barcelona 6h aprox.)
  - Coche (2013): 3 150 000 Km/h (Madrid-Barcelona en menos de 1s)



# Aspectos tecnológicos

## Generación de reloj

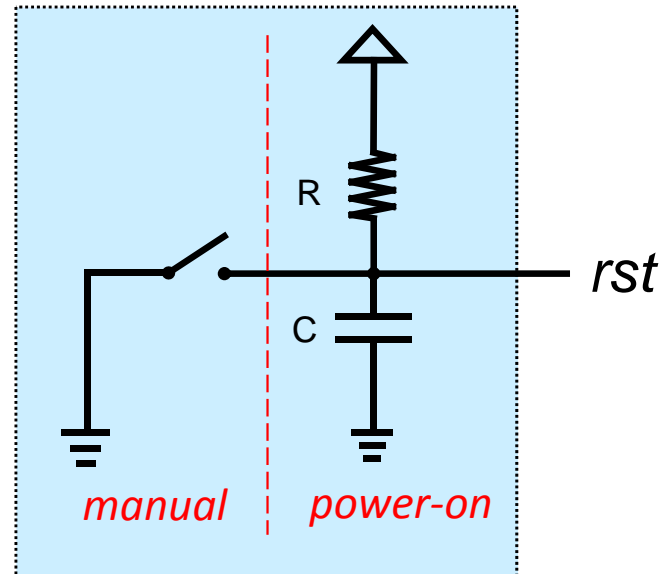
- El diseño (físico) de una la red de **distribución de reloj** en un chip es un proceso complejo:
  - La señal de reloj tiene mucha conectividad (tiene mucha carga) y las interconexiones son largas (muy resistivas y vulnerables al ruido).
  - Requiere trazar un **árbol /red equilibrada** de interconexiones, buffers y otros elementos correctores de desfase (PLL, DLL).
- En general es **mala práctica** que el reloj atraviese lógica:
  - Introduce un skew variable debido a la incertidumbre de la red.
  - Puede producir glitches que provoquen cambios espurios de estado.
    - Esto también es aplicable al reset, si se desea evitar inicializaciones espurias.



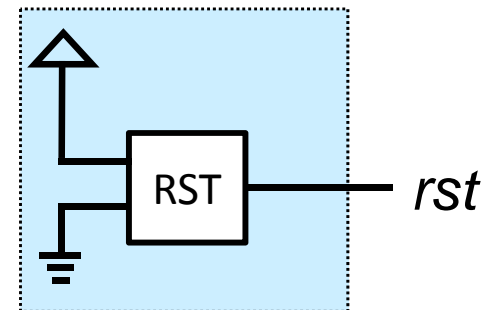
# Aspectos tecnológicos

## Generación de reset

- La señal de **reset** se genera externamente usando:
  - Un **circuito RC**



- Generador de reset integrado

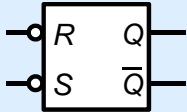
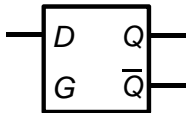
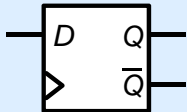
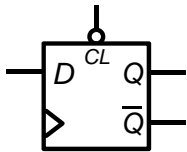
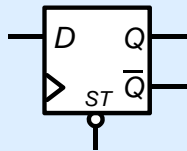
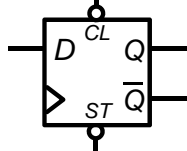




# Aspectos tecnológicos

## Biblioteca de celdas: CMOS 90 nm

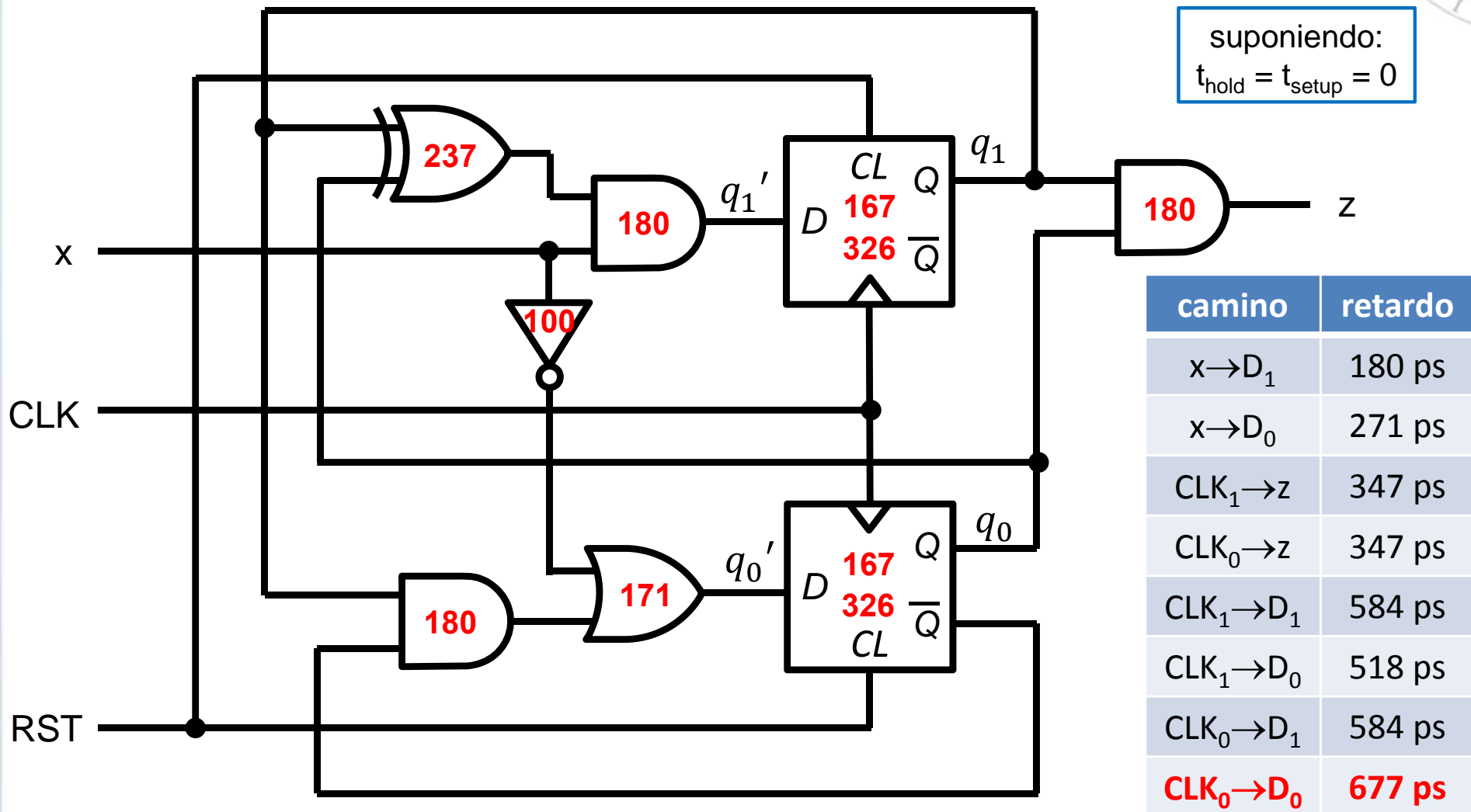
fuelle: Synopsys (SAED EDK 90 nm)

Biestable	Área ( $\mu\text{m}^2$ )	Retardo (ps)	Consumo estático (nW)	Consumo dinámico (nW/MHz)
	10.1376	221 (Q) 386 (QN)	621	2359
	22.1184	219 (Q) 234 (QN)	144	463
	24.8832	217 (Q) 193 (QN)	140	284
	32.2560	167 (Q) 326 (QN)	164	281
	31.3344	412 (Q) 372 (QN)	152	161
	35.0208	212 (Q) 365 (QN)	167	215



# Aspectos tecnológicos

## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)

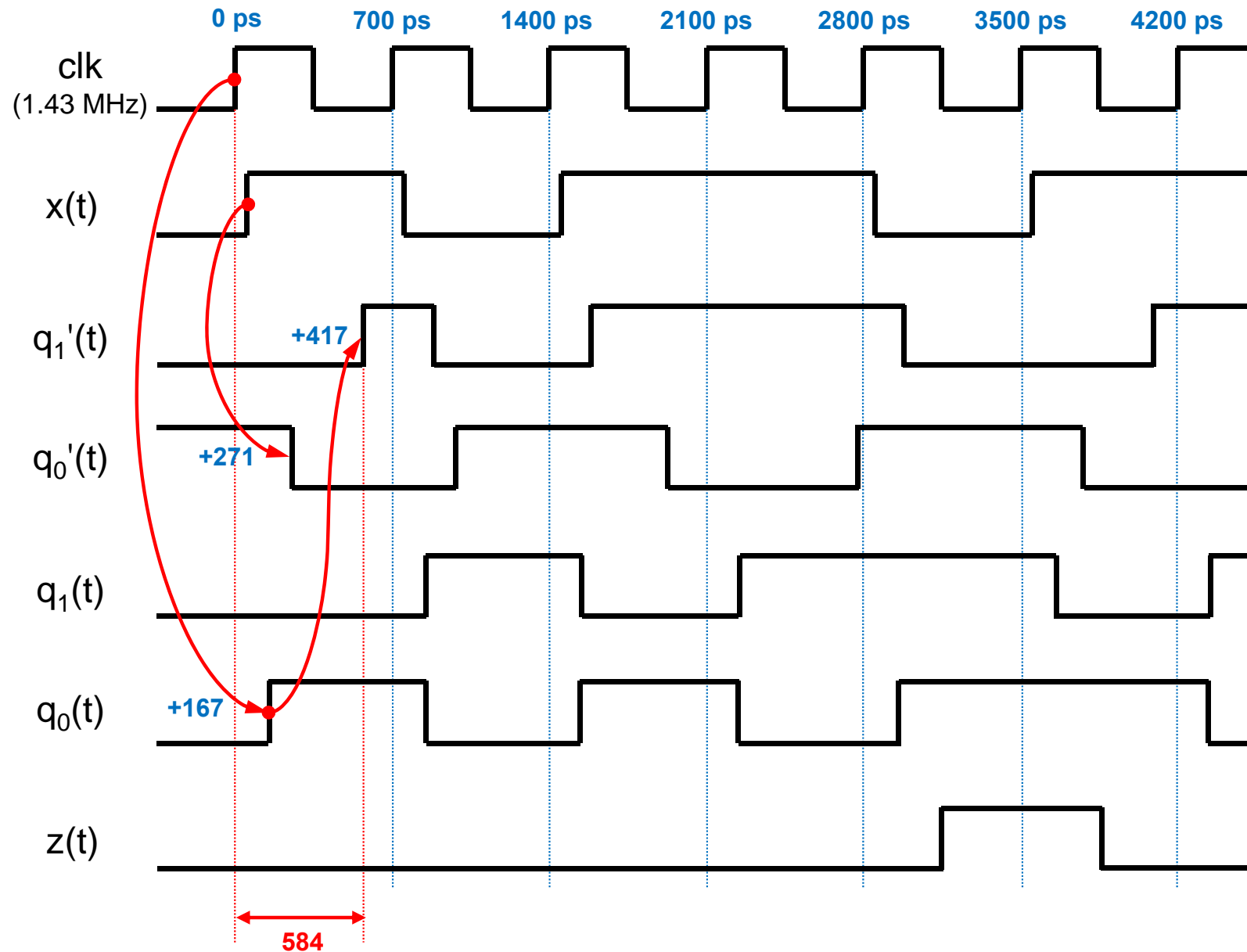


$$t_{\text{clk}} \geq \max(\text{retardo}) = 677 \text{ ps} \Rightarrow f_{\text{clk}} = \frac{1}{t_{\text{clk}}} \leq \frac{1}{677 \cdot 10^{-12} \text{ s}} = \mathbf{1.47 \text{ GHz}}$$



# Aspectos tecnológicos

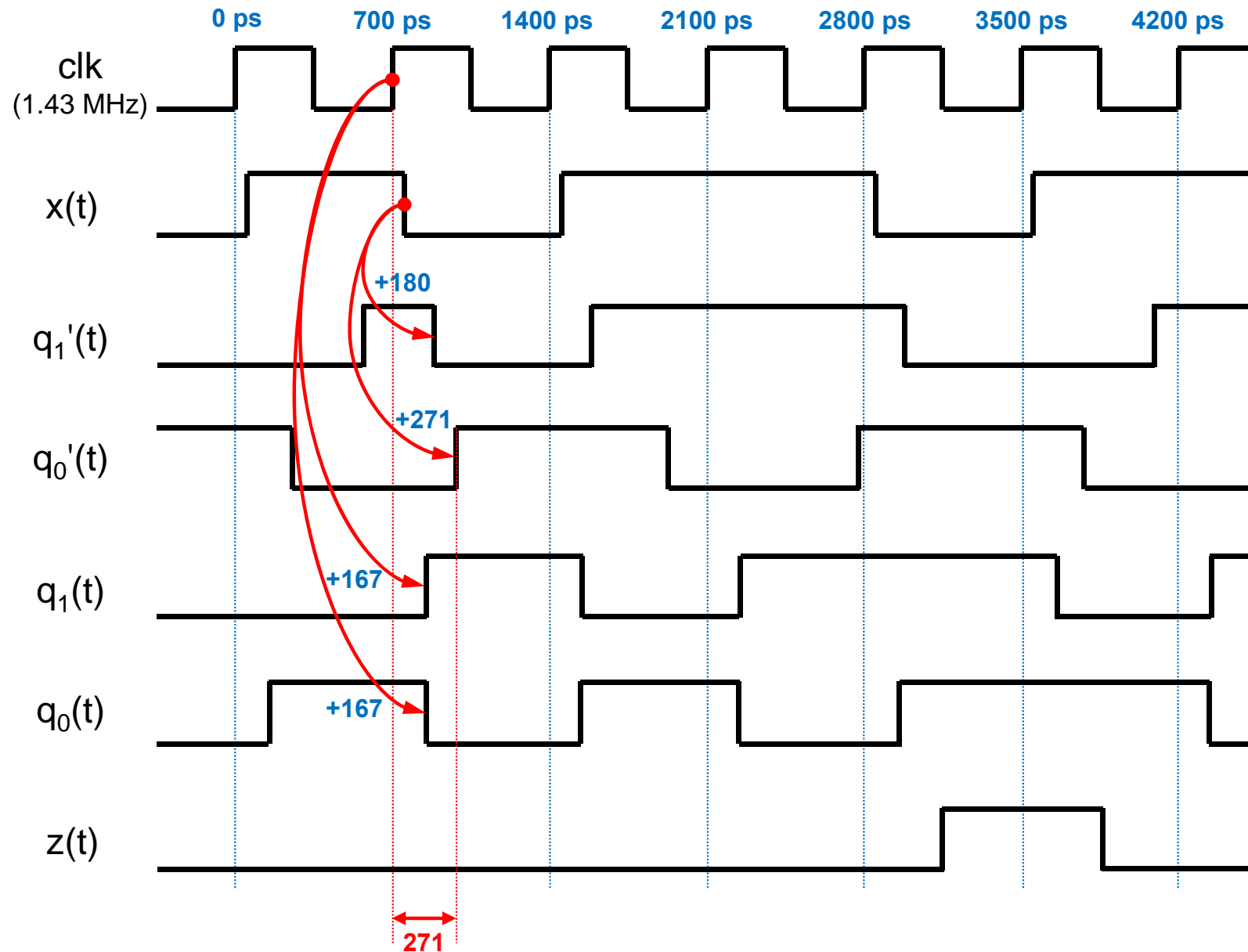
## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)





# Aspectos tecnológicos

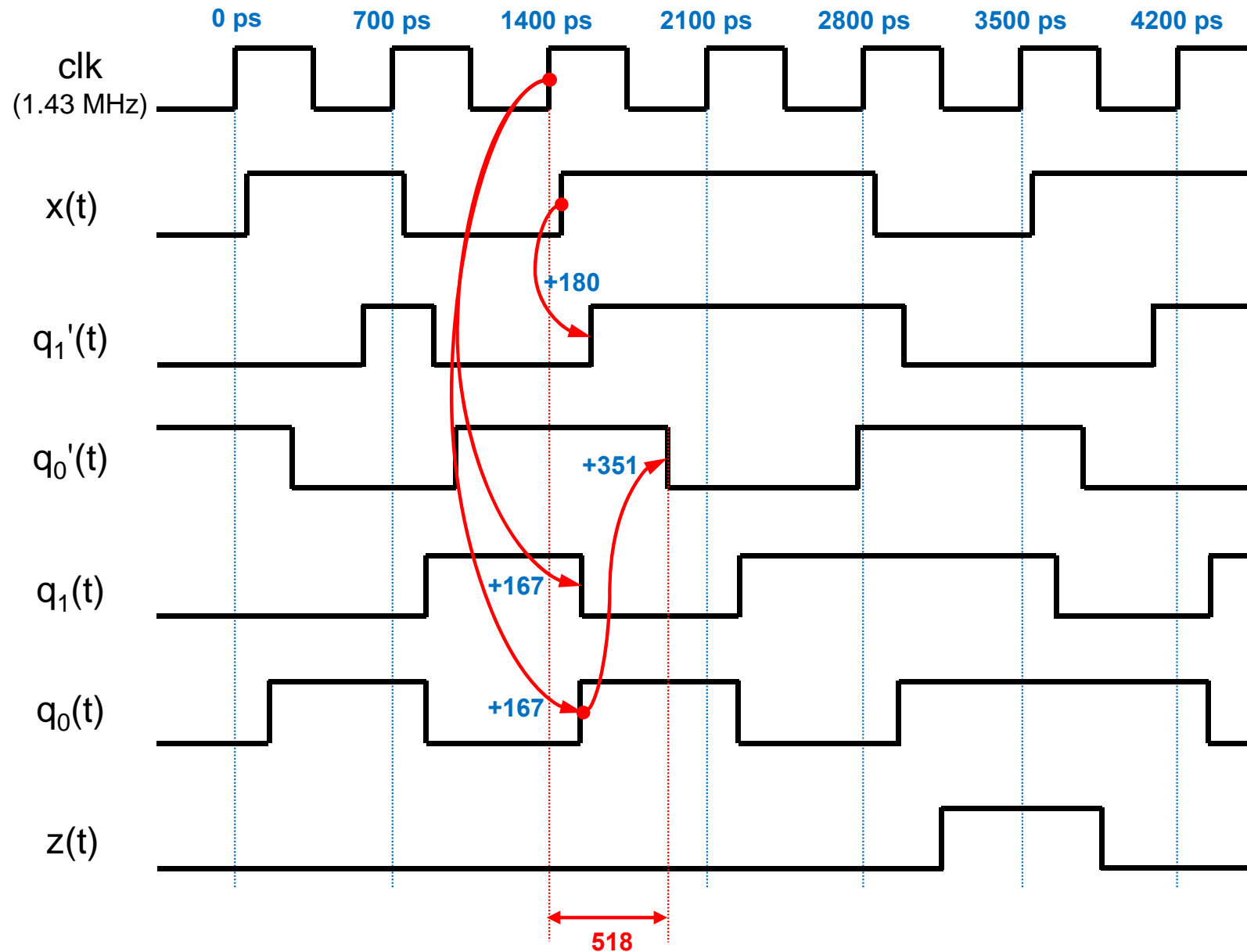
## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)





# Aspectos tecnológicos

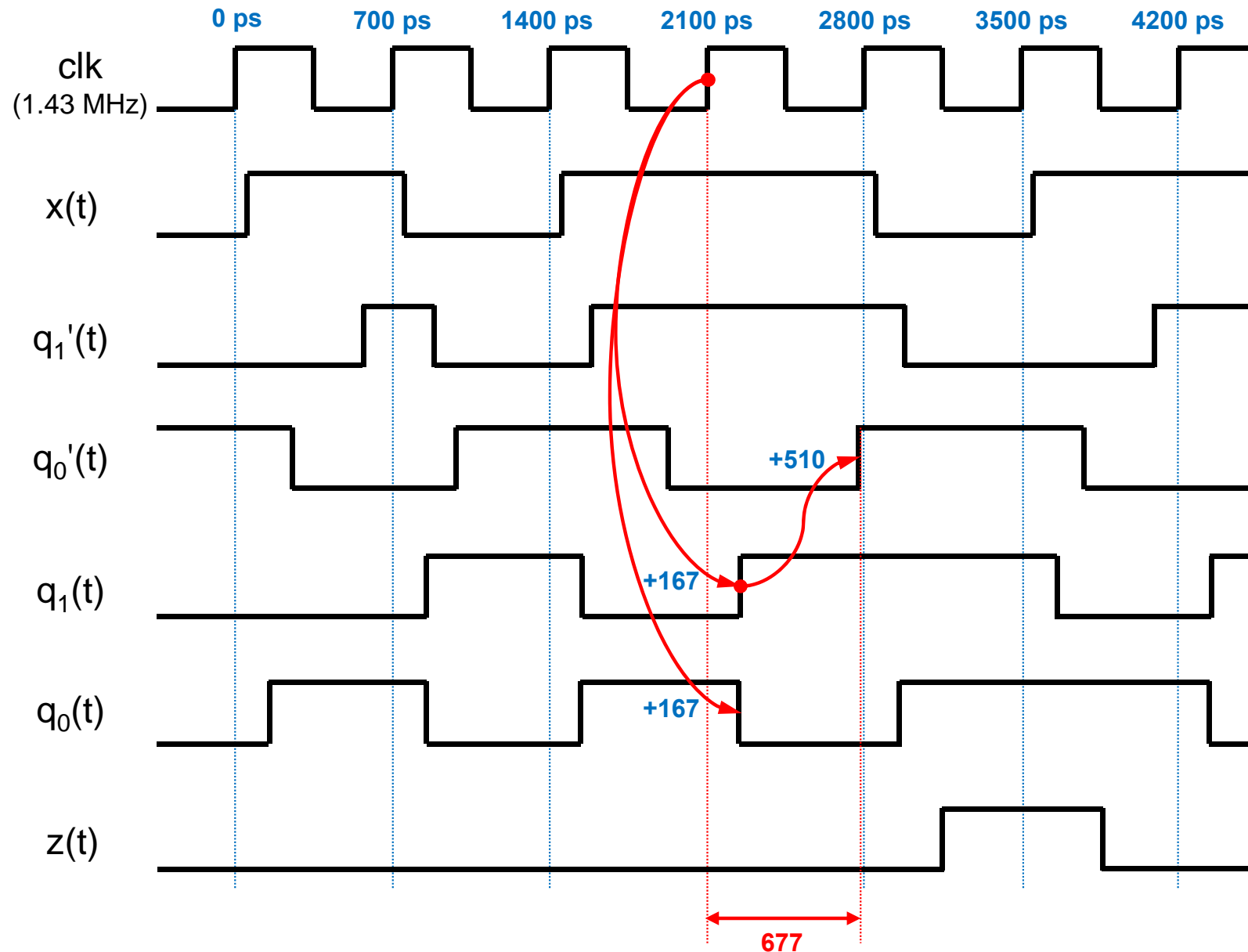
## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)





# Aspectos tecnológicos

## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)

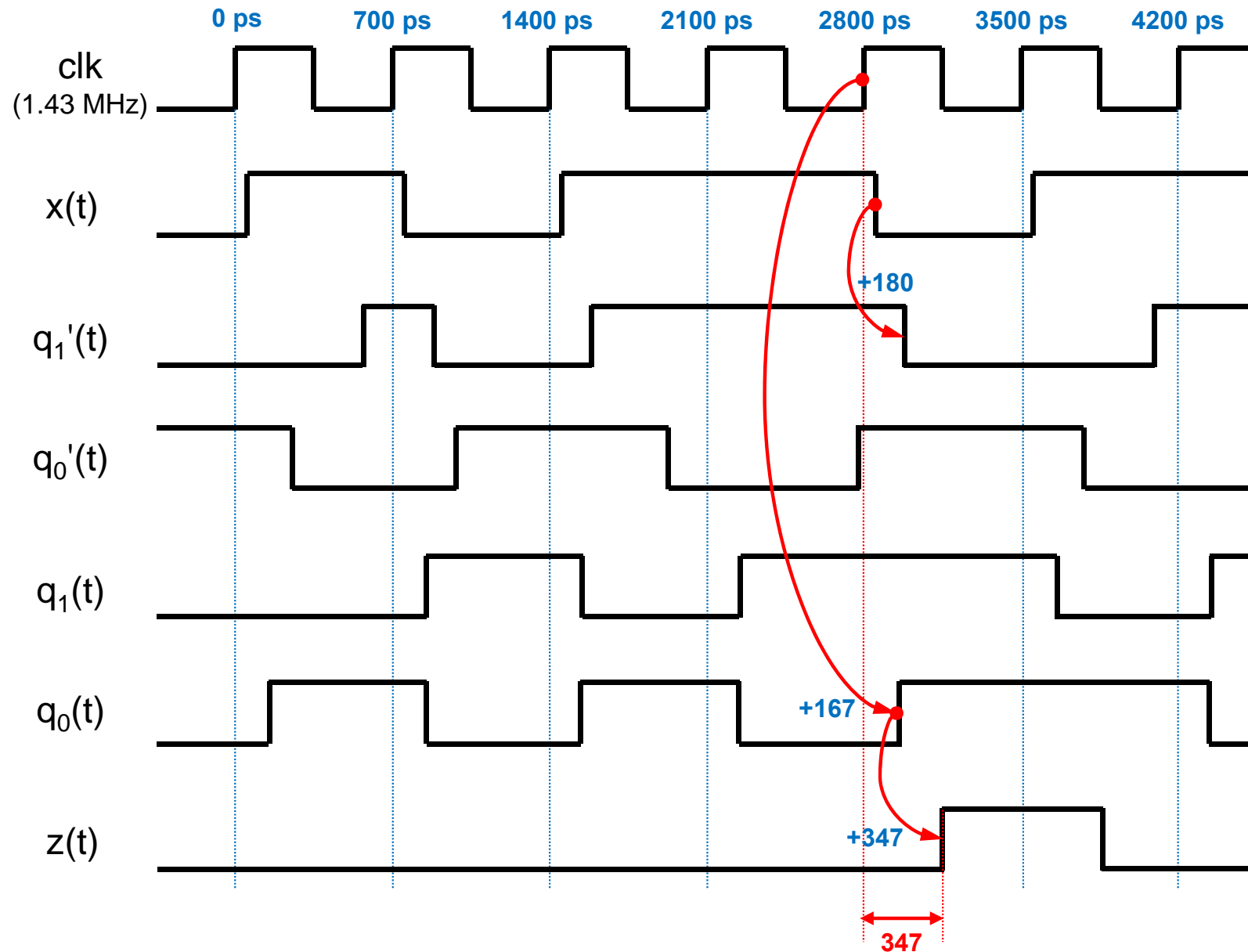






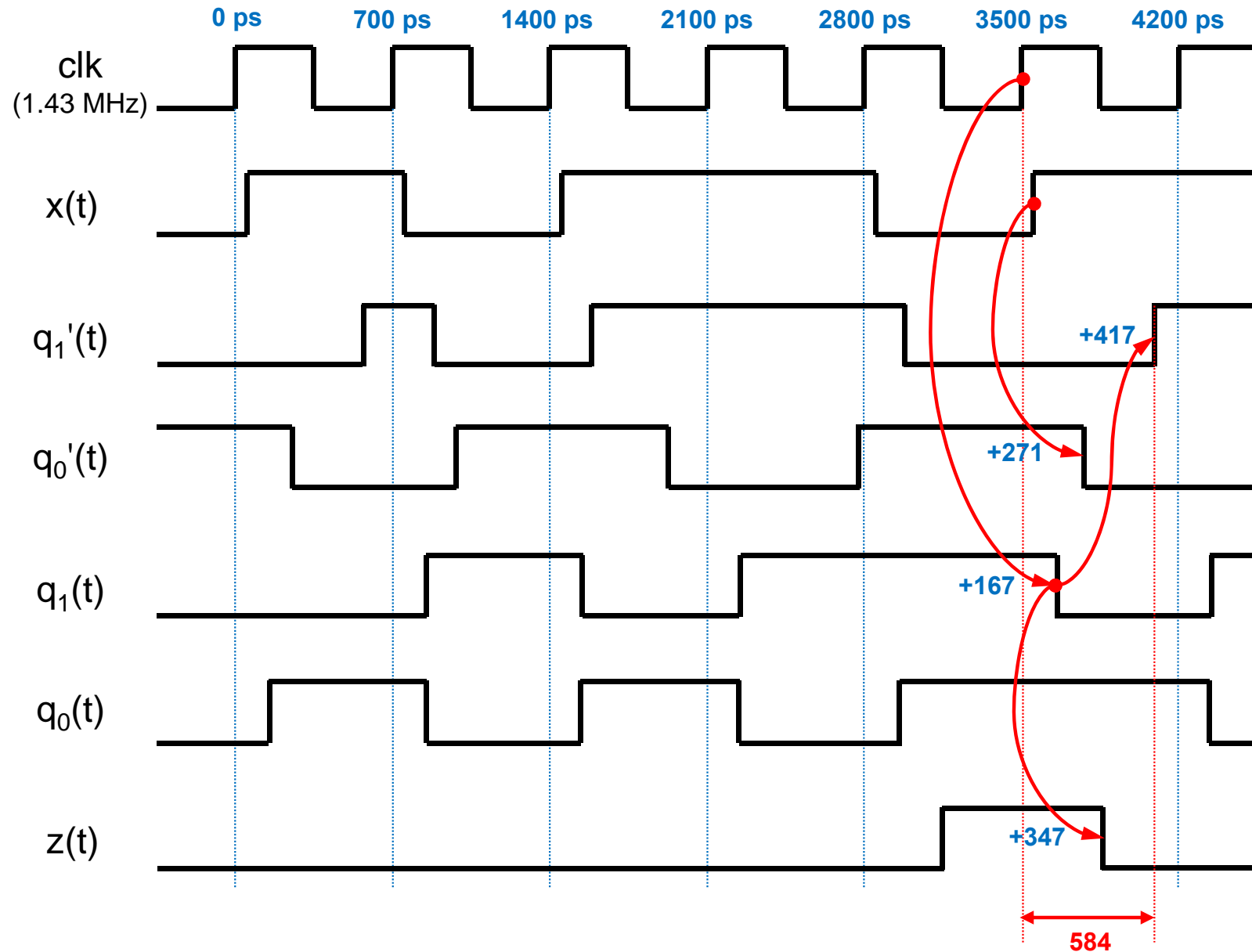
# Aspectos tecnológicos

## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)



# Aspectos tecnológicos

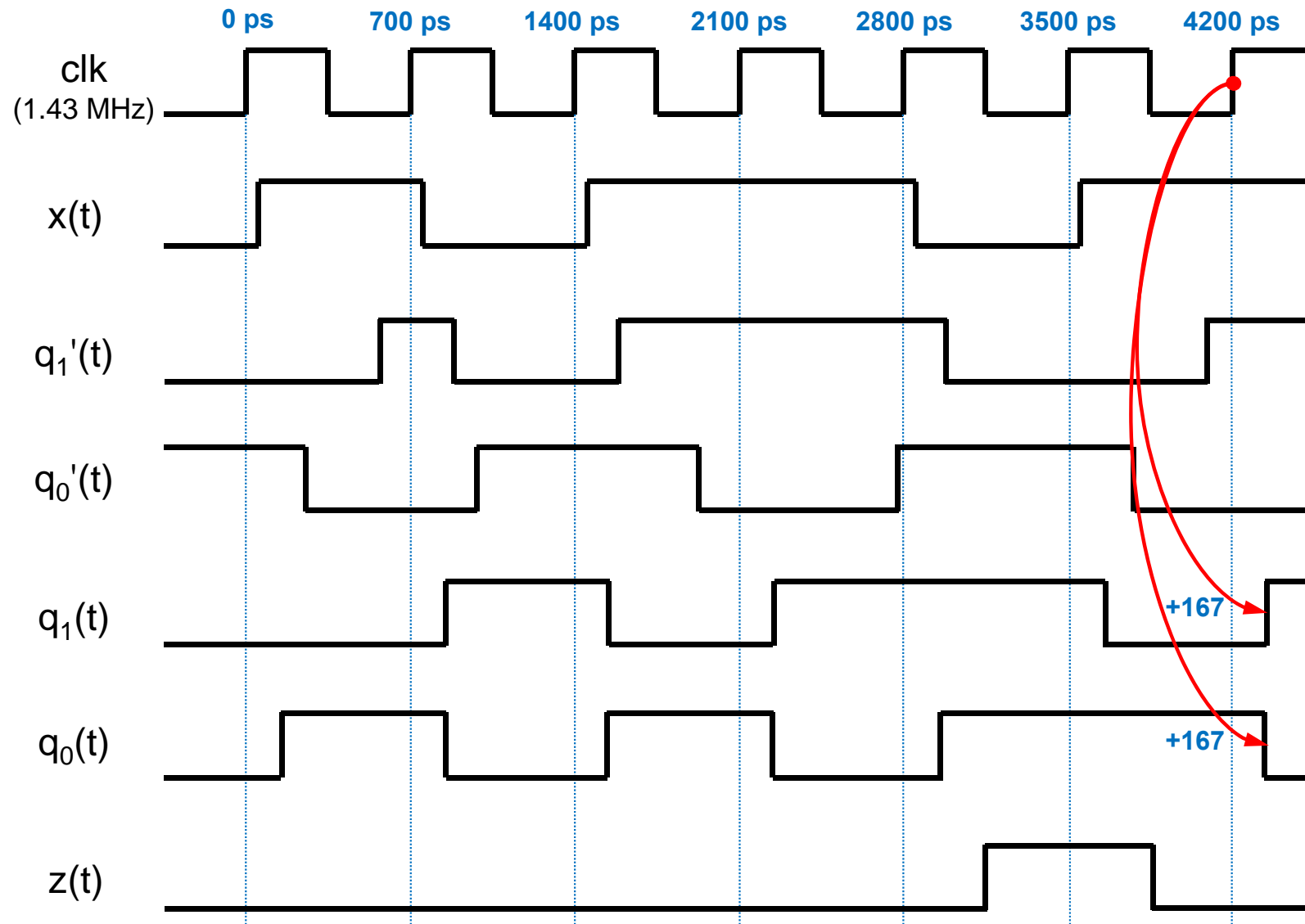
## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)





# Aspectos tecnológicos

## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)





# Aspectos tecnológicos

## Cálculo de la frecuencia máx. de reloj (CMOS 90 nm)

