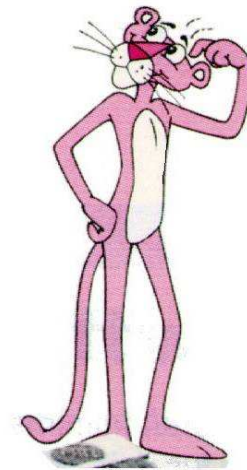
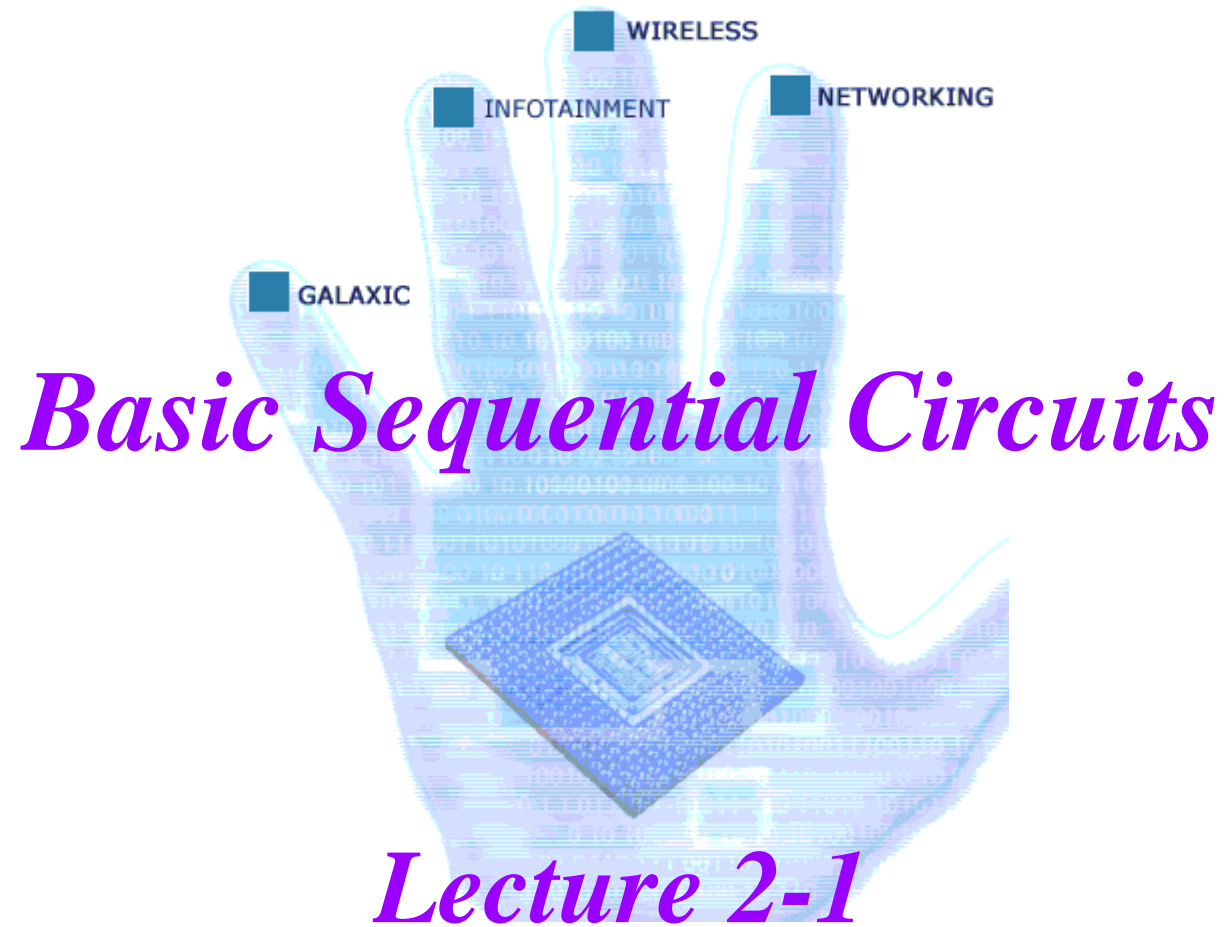
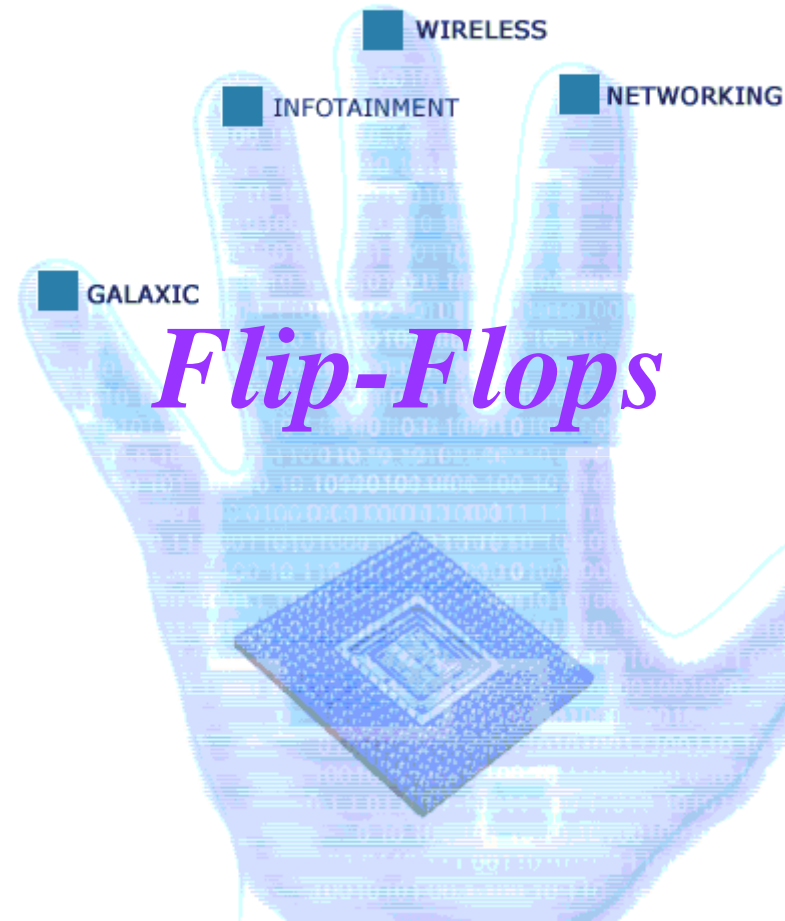


# *Digital System Design Course*







# *Circuitos secuenciales básicos*

❑ Los *circuitos secuenciales básicos* son de **propósito general**:

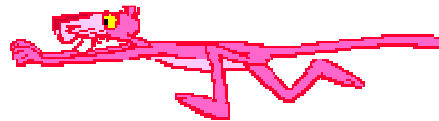
❖ Circuitos lógicos que *no se diseñan*: **ICs**

- Latches
- Flop-flops
- Registros
- Registros de desplazamiento
- Contadores



# *Tipos de flip-flops*

- ❑ Flip-flop S-R basado en *latches S-R*, modo nivel
- ❑ Flip-flop J-K basado en *latches S-R*, modo nivel
- ❑ Flip-flop D basado en *latches D*, modo flanco
- ❑ Flip-flop J-K basado en *flip-flop D*, modo flanco
- ❑ Flip-flop T basado en *flip-flop D*, modo flanco
- ❑ Flip-flop T basado en *flip-flop J-K*, modo flanco



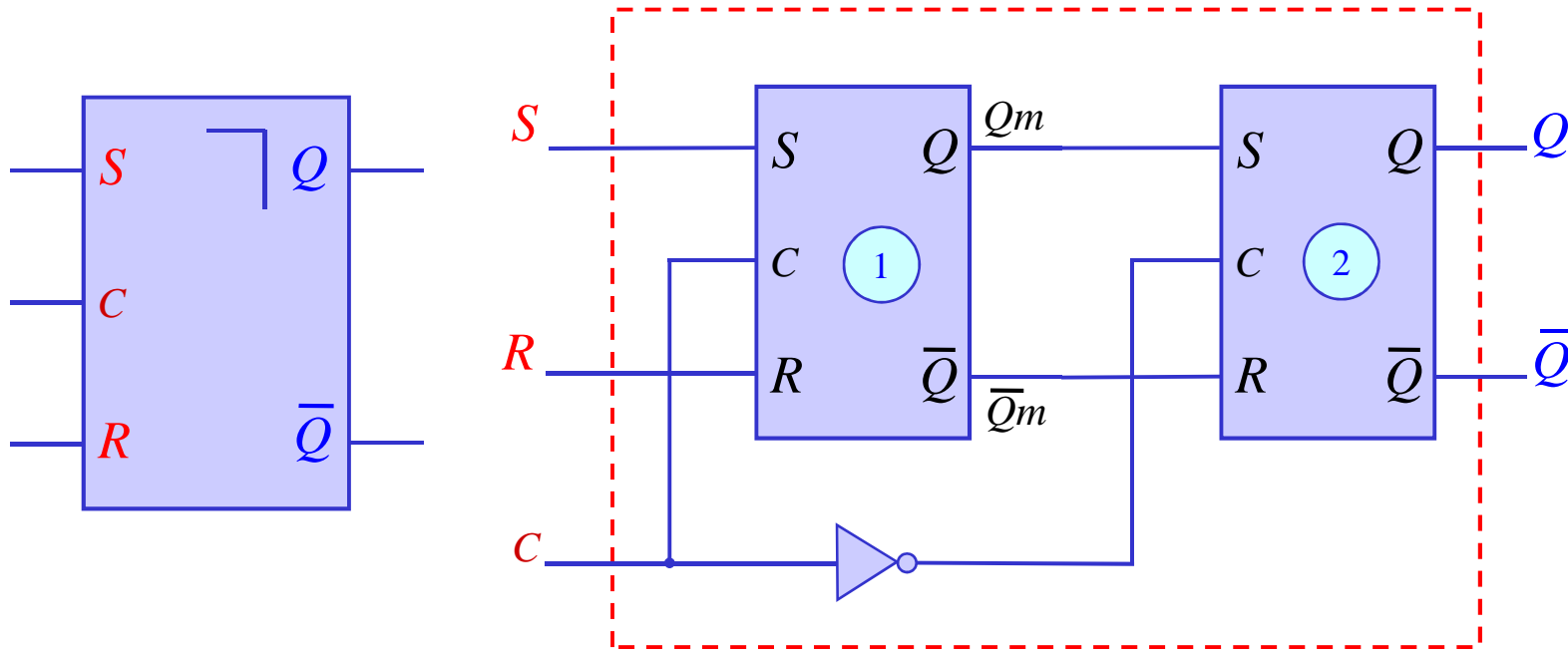
# *Flip-Flop S-R*

## ❑ Consideraciones:

- ❖ Dos latches S-R en serie conforman un *flip-flop S-R master/slave*
- ❖ El nivel lógico transferido a la salida del *flip-flop* sobre el *flanco de caída* de la *señal C* depende de la última **señal de Set** o **Clear** del latch master, mientras la señal C estaba en alto
- ❖ El *flip-flop S-R master/slave* no es realmente *conmutado en el flanco*. Este *es un latch* que sigue a sus entradas durante todo el intervalo que la señal C permanece en alto, pero que cambia sus salidas únicamente cuando la señal C va abajo, *y refleja el último nivel atrapado*
- ❖ Los *flip-flops* con este tipo de comportamiento, son algunas veces llamados *flip-flops* que *conmutan* por *pulso o nivel*: **pulse-triggered flip-flops**

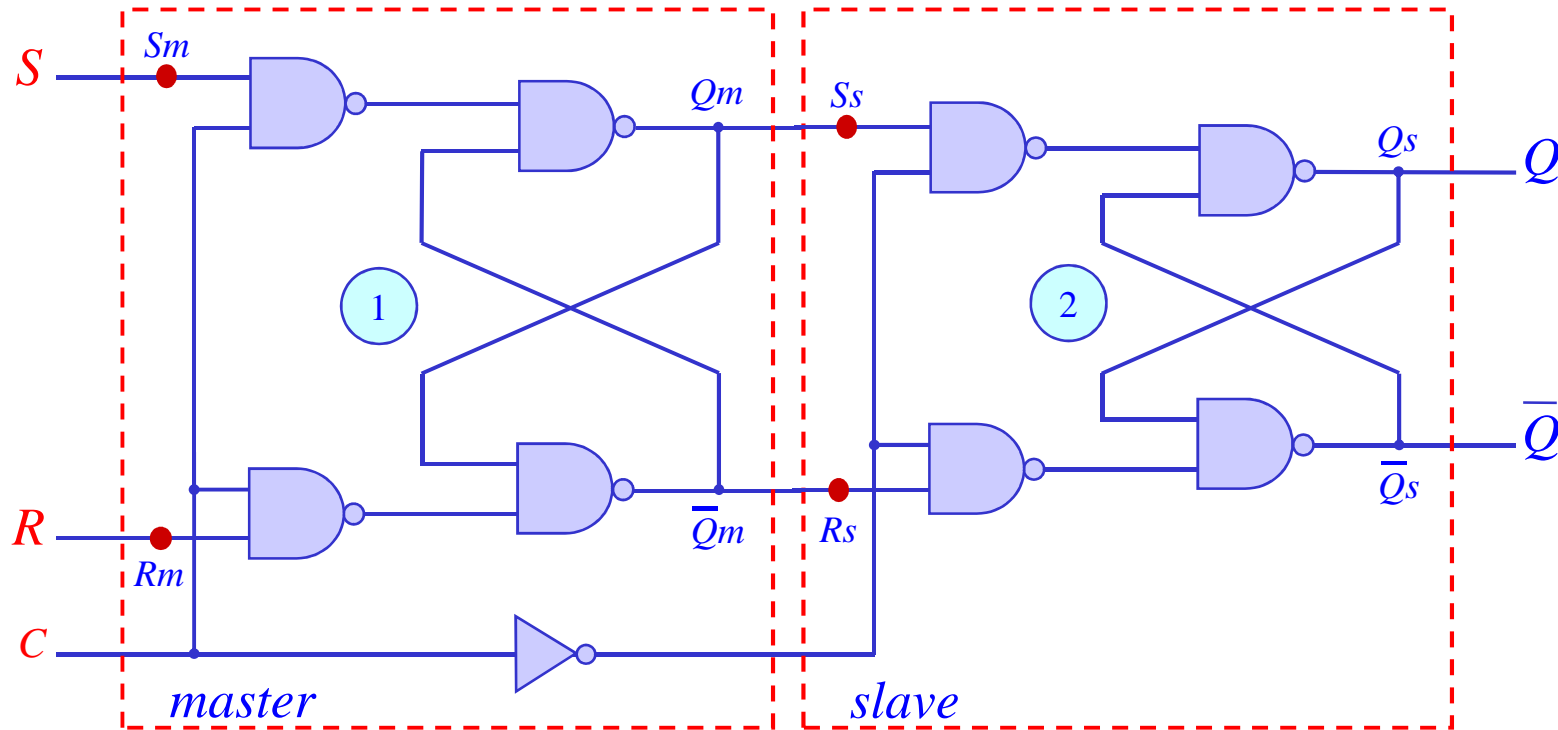
# *Flip-Flop S-R*

❖ Símbolo y diagrama lógico: Flip-flop S-R master/slave



# Flip-Flop S-R





❖ Diagrama lógico: FF-SR a nivel de compuertas





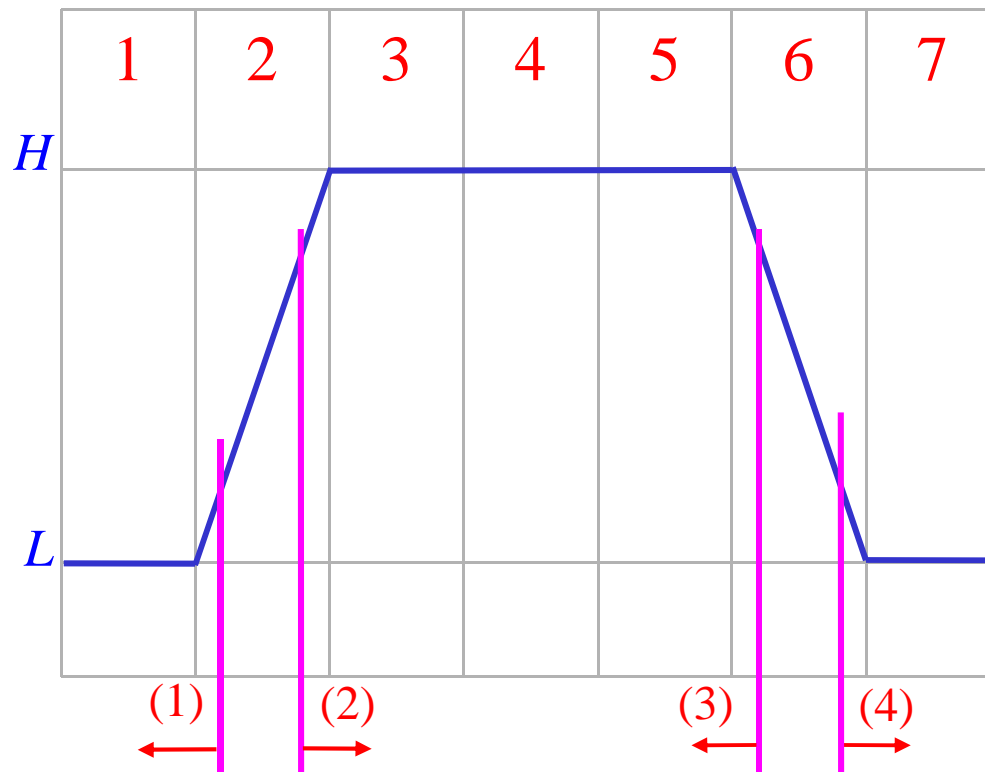
# *Flip-Flop S-R*

❖ Tabla de verdad: FF-SR

S	R	CLK	$Q$	$\overline{Q}$
X	X	0	$Q$	$\overline{Q}$
0	0		$Q$	$\overline{Q}$
0	1		0	1
1	0		1	0
1	1		Indefi.	Indefi.

# *Flip-Flop S-R*

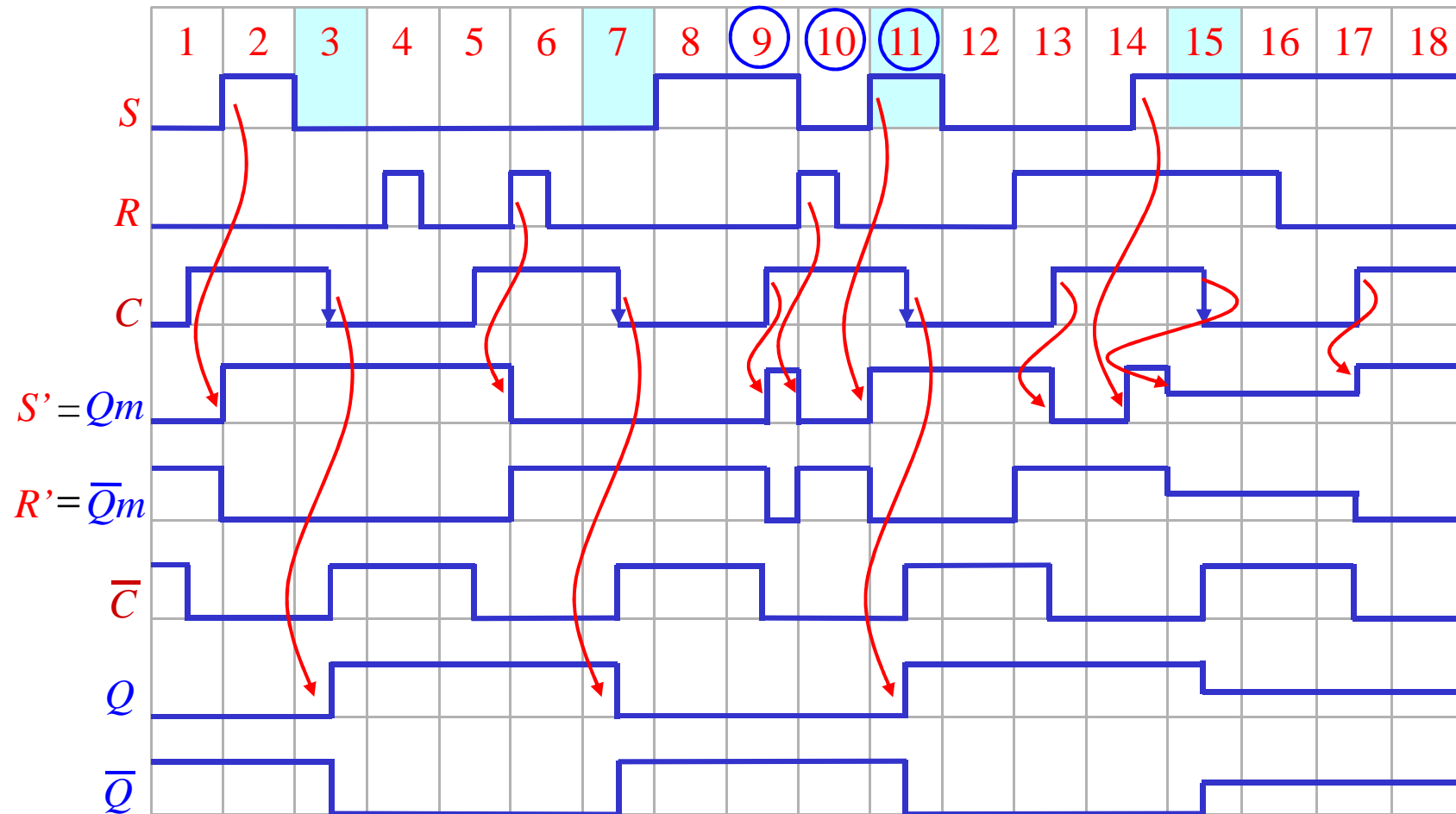
❖ Diagrama de timing: FF-SR funcionamiento



- (1) Aislar: maestro/esclavo
- (2) Entrar datos: maestro
- (3) Inhabilitar: entradas
- (4) Transferir datos: maestro/esclavo

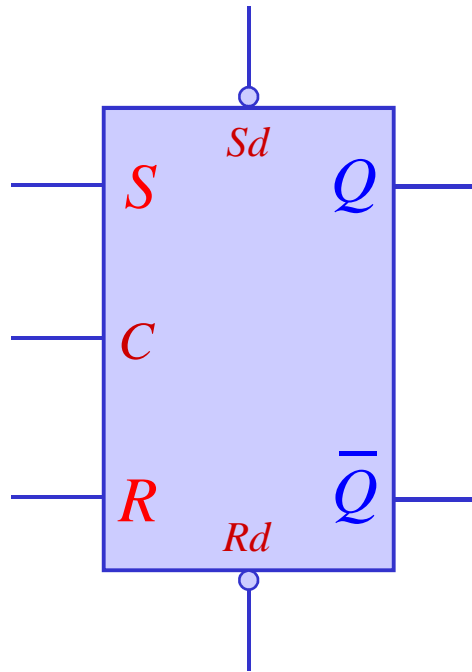
# *Flip-Flop S-R*

❖ Diagrama de timing: FF-SR







# *Flip-Flop S-R*

❑ Flip-flop S-R master/slave con set y reset: *Símbolo*



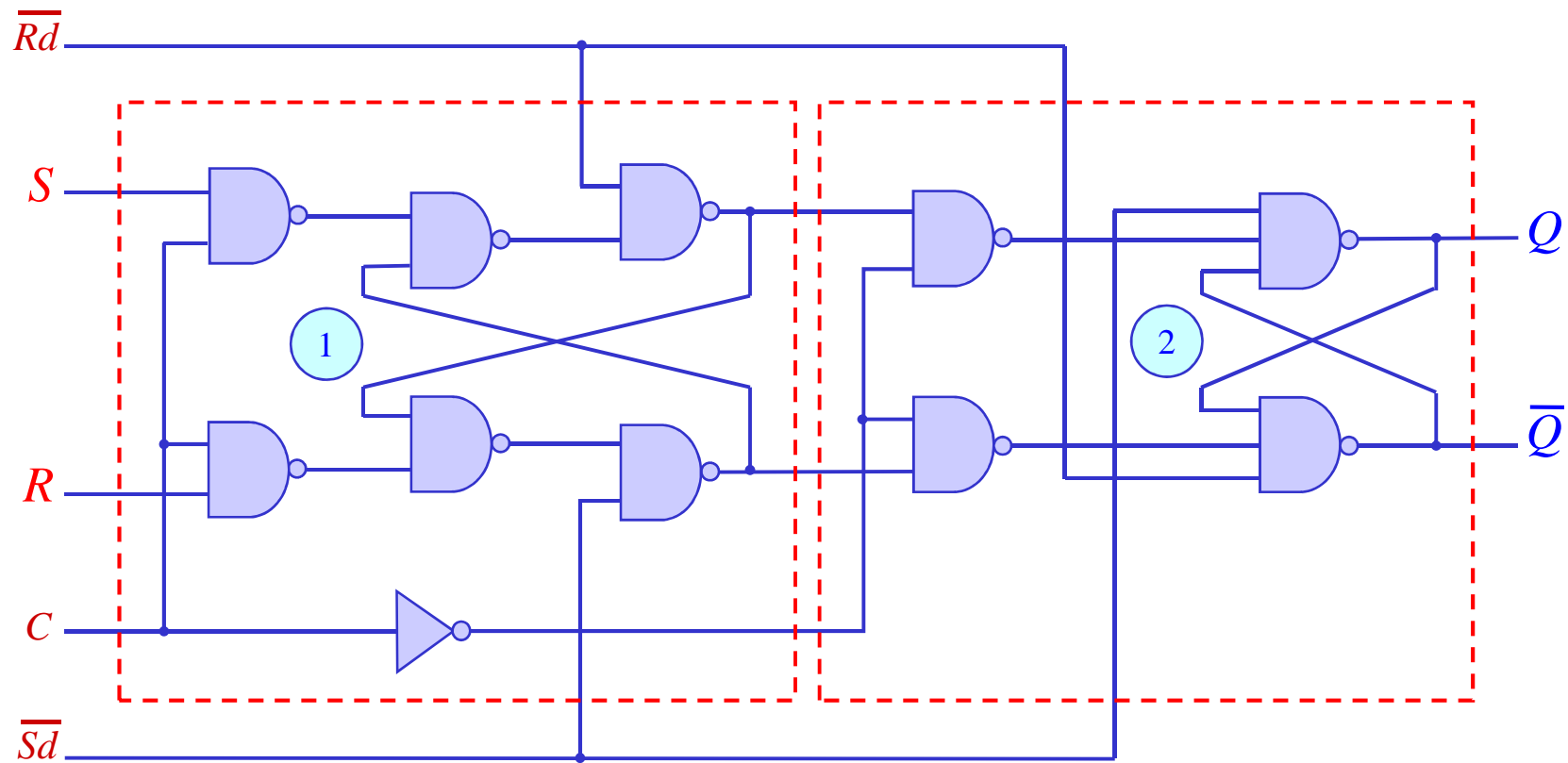
# Flip-Flop S-R

❑ Flip-flop S-R master/slave con set y reset: *Tabla de verdad*

$\overline{Sd}$	$\overline{Rd}$	S	R	CLK	Q	$\overline{Q}$
0	1	X	X	X	1	0
1	0	X	X	X	0	1
1	0	X	X	0	Q	$\overline{Q}$
1	1	0	0		Q	$\overline{Q}$
1	1	0	1		0	1
1	1	1	0		1	0
1	1	1	1		Indefi	Indefi.

# *Flip-Flop S-R*

❖ Diagrama lógico



# *Flip-Flop S-R*

## ❑ Consideraciones.

- ❖ Las entradas J y K no necesitan estar activadas al final del pulso de disparo para que la salida del flip – flop cambie. (1)
- ❖ El problema que surge cuando S y R son activadas simultáneamente es solucionado con un flip – flop J –K master slave. Las entradas J y K son análogas a S y R.

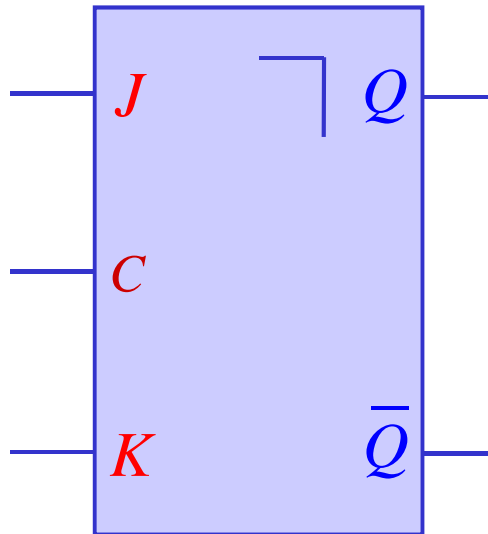
## *Flip-Flop J-K*

- ❖ El Flip – Flop J –K puede atrapar: gating, en las entradas S y R del latch master, los niveles lógicos 1S y 0S, permitiendo que la salida del flip – flop cambie aunque K y J no estén activadas en el final del flanco del pulso de disparo; este comportamiento es conocido como **Catching de 1S y 0S**.
- ❖ Debido al comportamiento de **Catching de 1S y 0S**, las entradas J y K de un flip – flop J – K master/slave deben ser mantenidas válidas durante todo el intervalo en que C es 1.



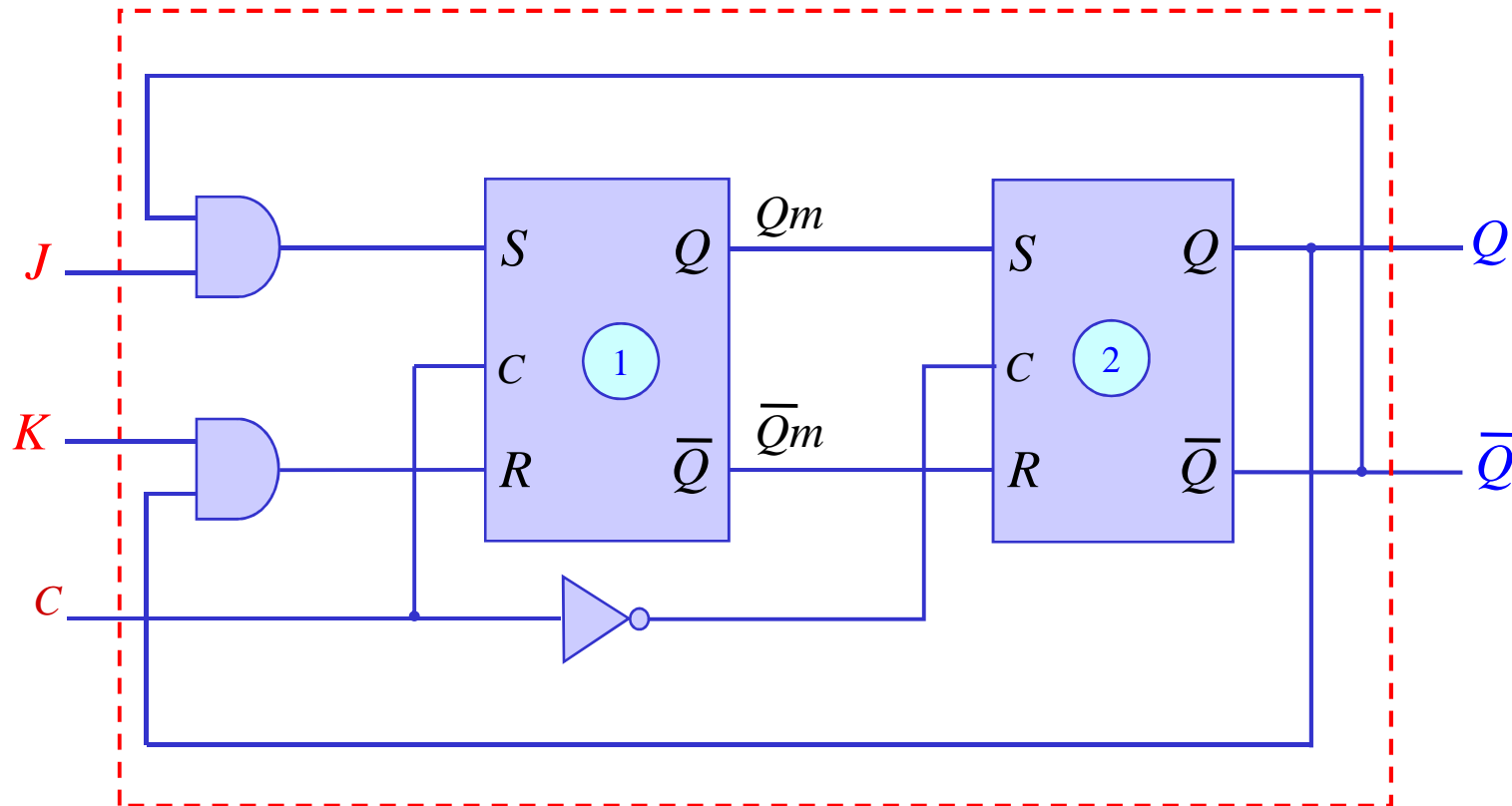
# *Flip-Flop J-K*

❑ Flip-flop J-K master/slave: *Símbolo*







# Flip-Flop J-K

- ❑ Flip-flop J-K master/slave: *Diagrama lógico*



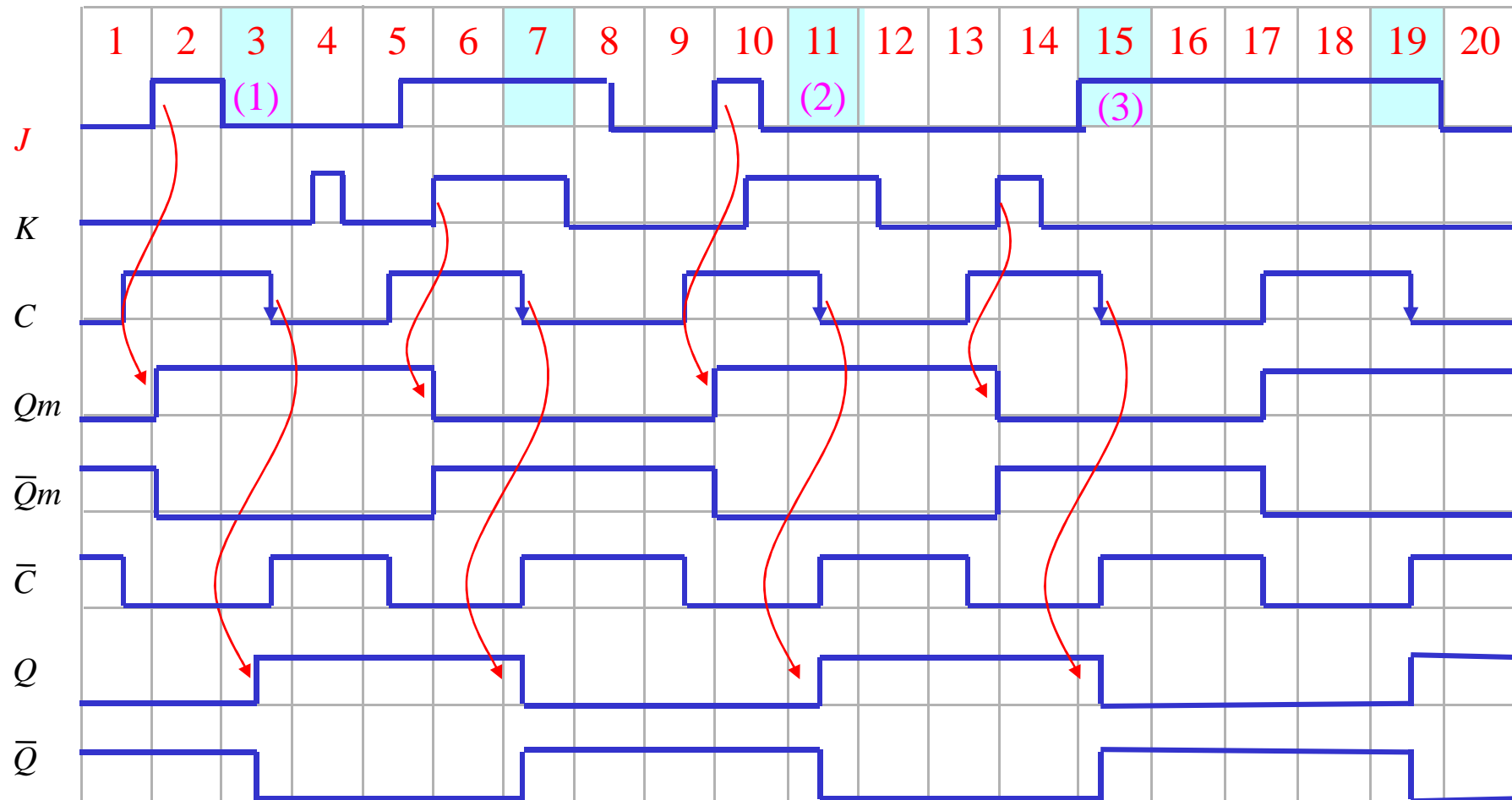
# *Flip-Flop J-K*

❖ Tabla de verdad

J	K	CLK	$Q$	$\overline{Q}$
X	X	0	$Q$	$\overline{Q}$
0	0		$Q$	$\overline{Q}$
0	1		0	1
1	0		1	0
1	1		$\overline{Q}$	$Q$

# Flip-Flop J-K

❖ Diagrama de timing: FF J-K master/slave



# *Flip-Flop D*

❑ Flip – Flop D. Disparado por flanco

❖ El flip –flop D disparado en el flanco positivo, se implementa con un par de latches D.

❖ El flip –flop D muestrea o atrapa la información presente en su entrada D únicamente en el flanco de subida de la señal de control C / CK .

❖ El latch master está habilitado y sigue la señal de la entrada cuando CLK está bajo.

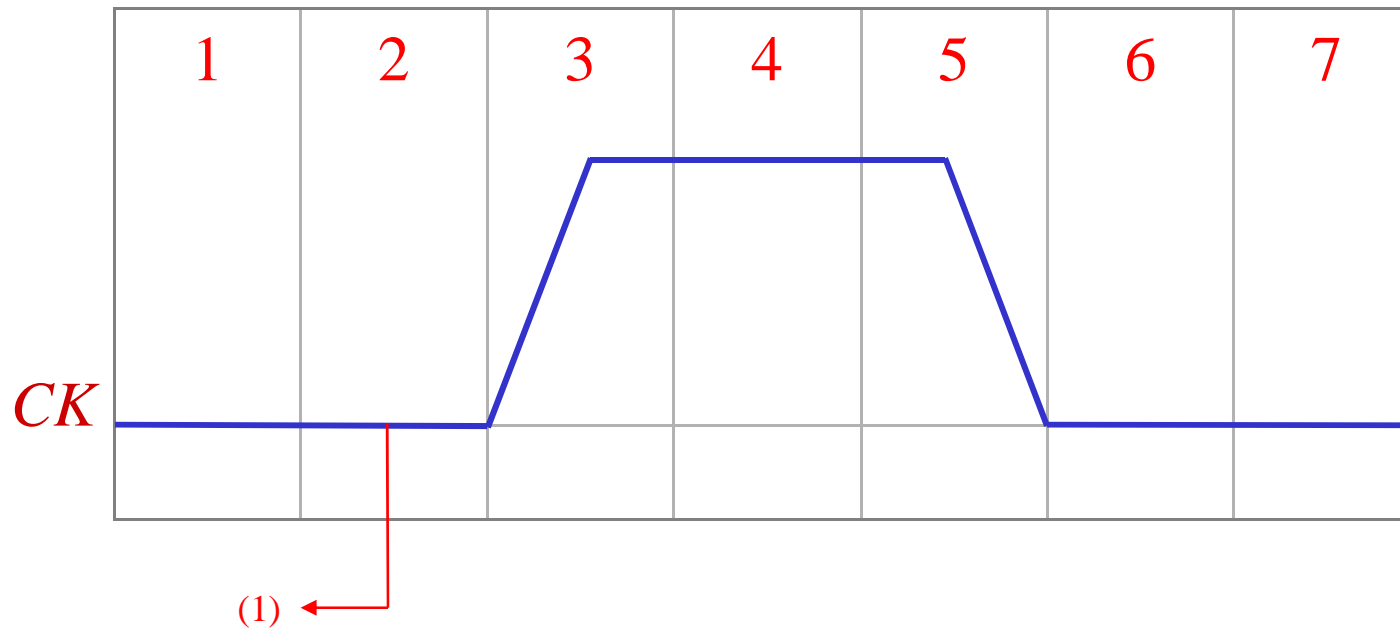
## *Flip-Flop D*

- ❖ Cuando CLK está alto el master está inhabilitado y su salida es transferida al latch slave. El latch slave está habilitado mientras la señal CK está alta, pero solamente cambia en el comienzo de este intervalo, debido a que el master esta inhabilitado y no cambia durante el resto del intervalo.
- ❖ El comportamiento funcional del flip – flop D se muestra en el diagrama de Timing.

# *Flip-Flop D*

□(1) El master está habilitado.

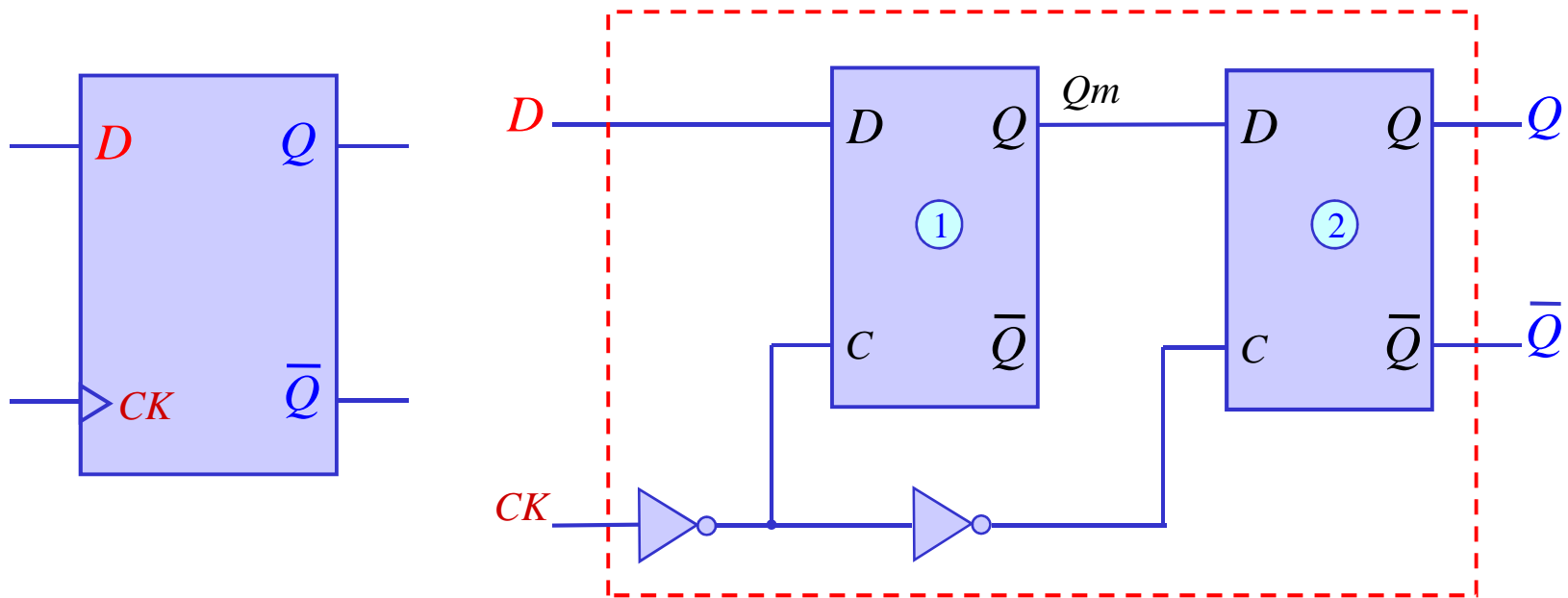
El slave está inhabilitado.



# Flip-Flop D

## ❑ Símbolo y diagrama lógico: Flip-flop D

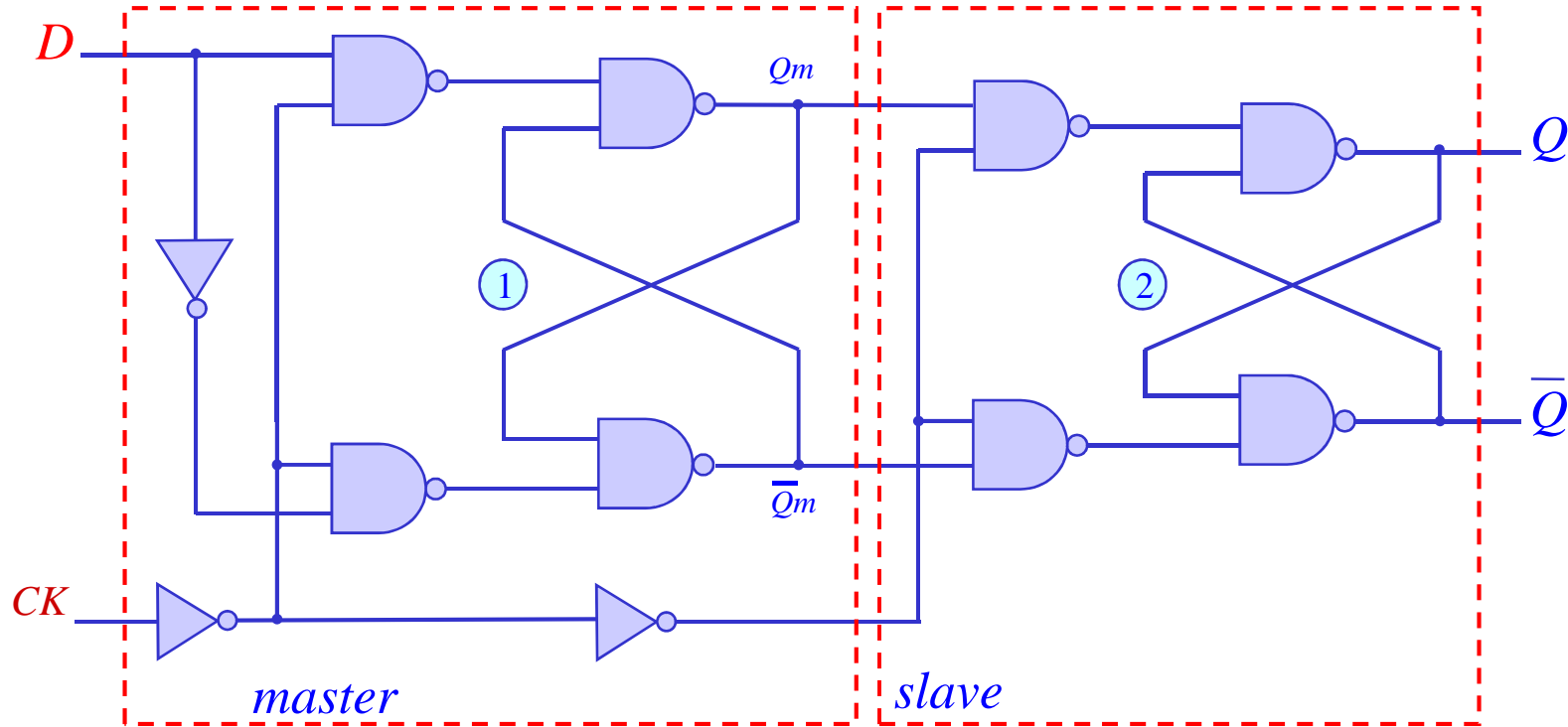
❖ FFD conmutado por flanco (edge triggered)





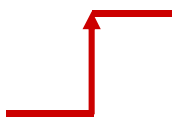

## Flip-Flop D

❑ Diagrama lógico: nivel de compuertas



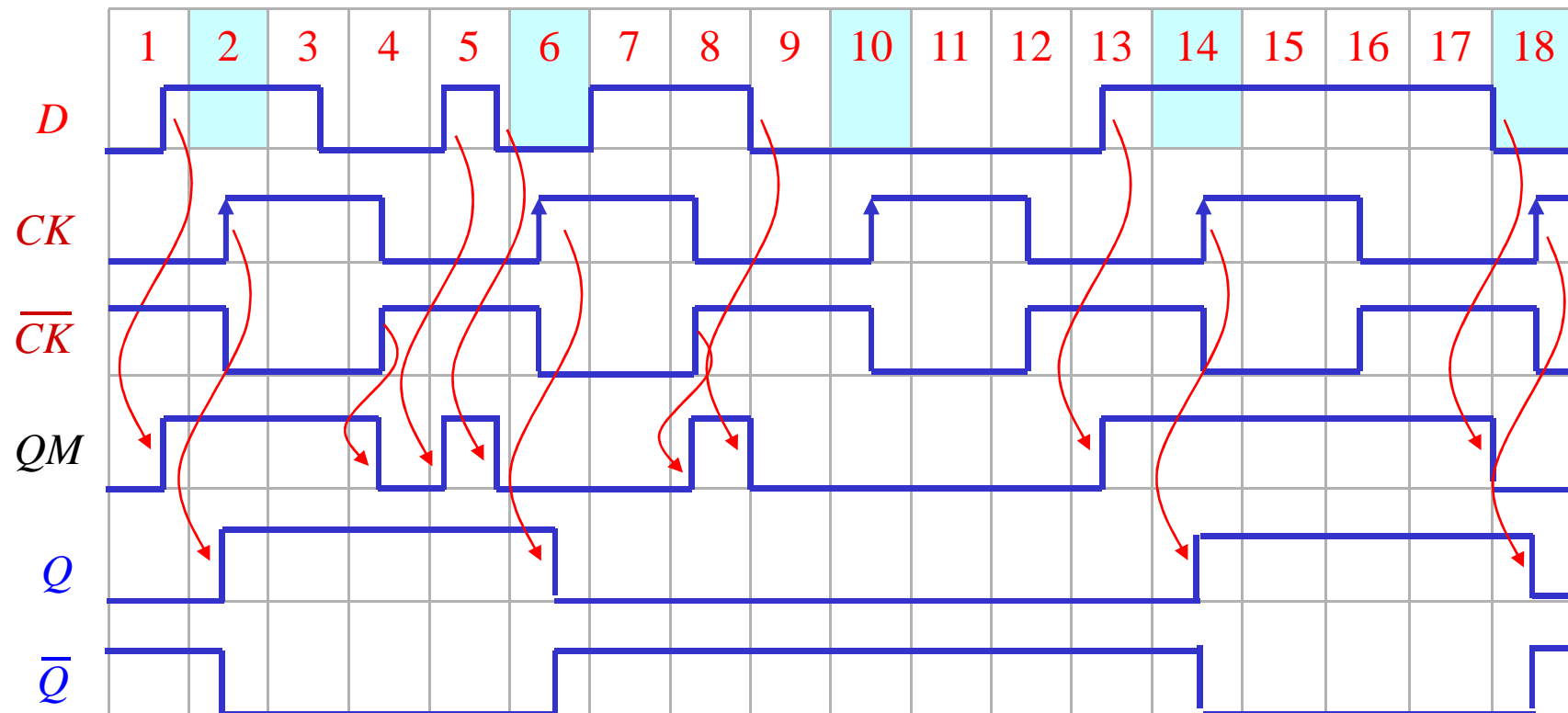
# *Flip-Flop D*

❑ Tabla de verdad

D	CK	$Q$	$\overline{Q}$
0		0	1
1		1	0
X	0	$Q$	$\overline{Q}$
X	1	$Q$	$\overline{Q}$

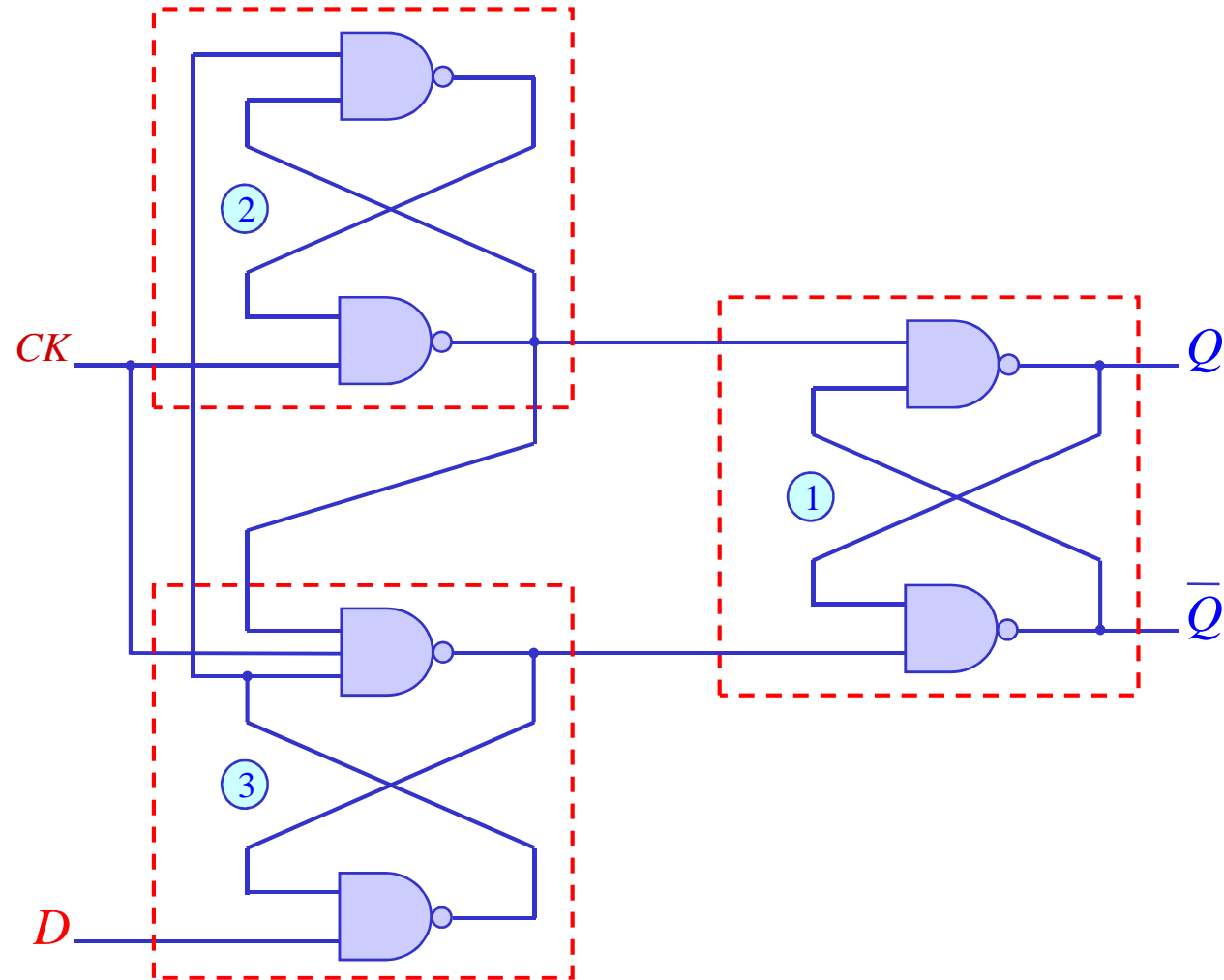
# *Flip-Flop D*

➤ Diagrama de Timing: FF -D



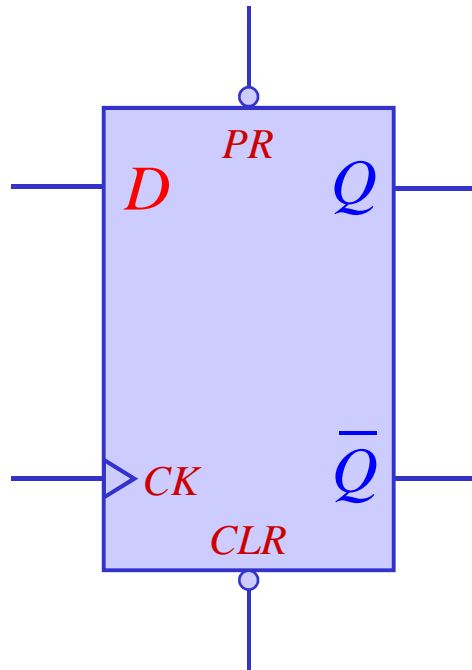
# *Flip-Flop D*

- ❖ Diagrama lógico: nivel de compuertas



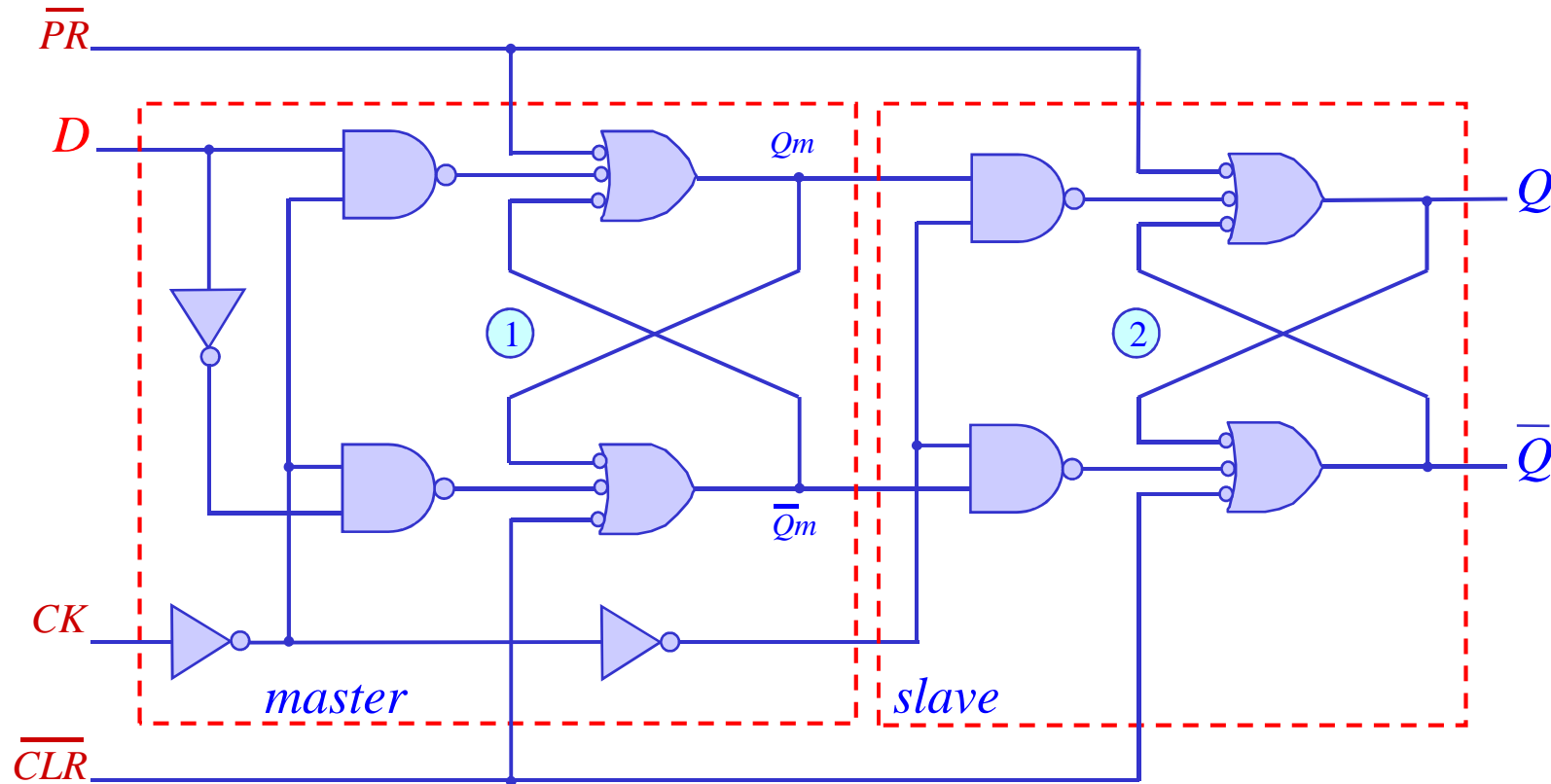
# *Flip-Flop D*

❑ Flip-flop D con preset y clear: *Símbolo*



# Flip-Flop D

- ❖ Diagrama lógico: nivel de puertas



# *Flip-Flop D*

## ❑ Parámetros de Timing

❖ Todos los atrasos de propagación son medidos a partir del flanco de subida del reloj: CK, puesto que ésto es lo que únicamente causa un cambio en la salida.

❖ El flip – flop D activado por flanco tiene definida una ventana de tiempos: setup y hold time, tiempo establecimiento y mantenimiento. Esta ventana ocurre alrededor del flanco de disparo de CK.

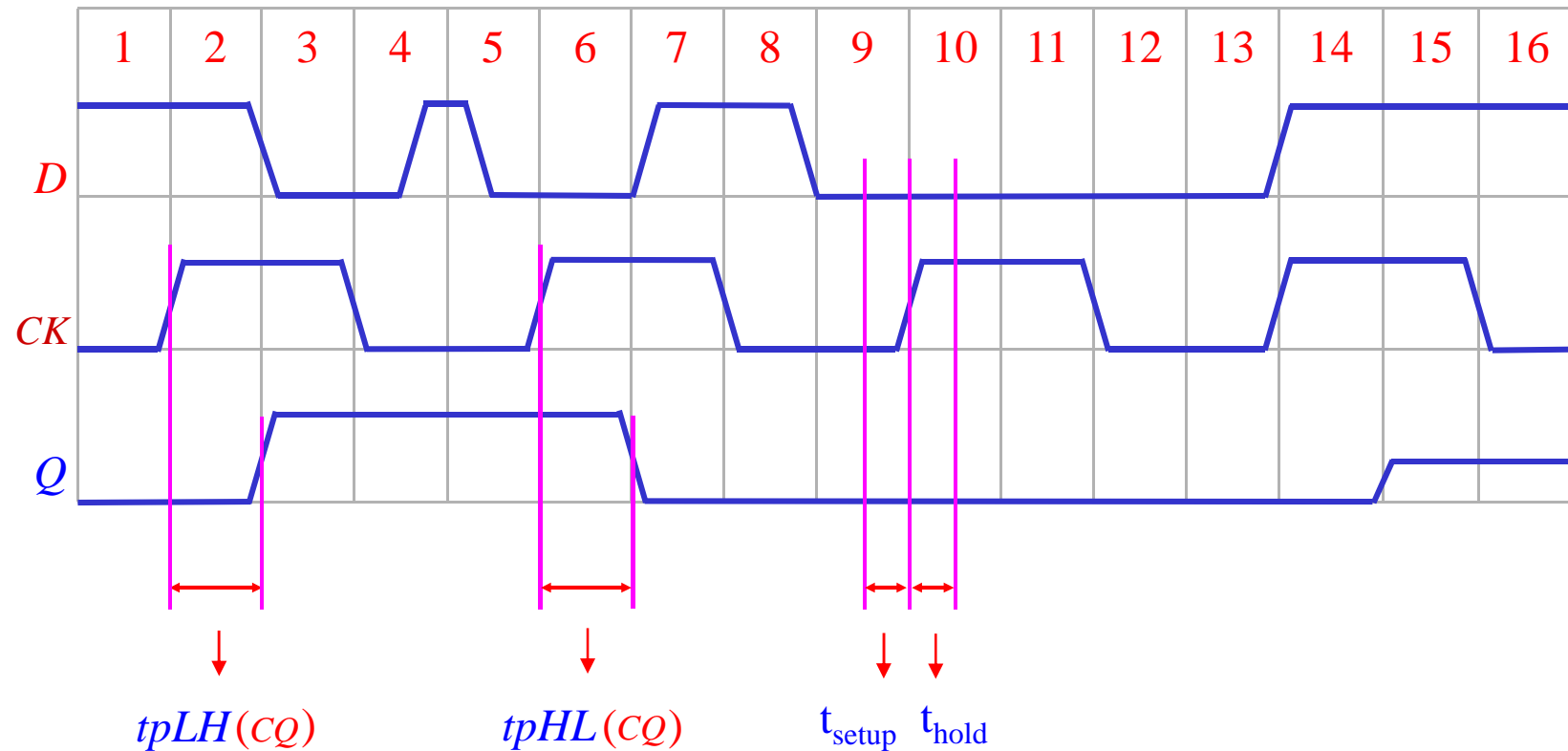
## *Flip-Flop D*

- ❖ Si los requerimientos para los tiempos de setup y hold time no se cumplen, la salida del flip-flop usualmente iría a un estado estable aunque, impredecible, 0 ó 1. En algunos casos, sin embargo la salida oscilaría o iría a un estado metaestable, es decir, a un estado intermedio entre 0 y 1.



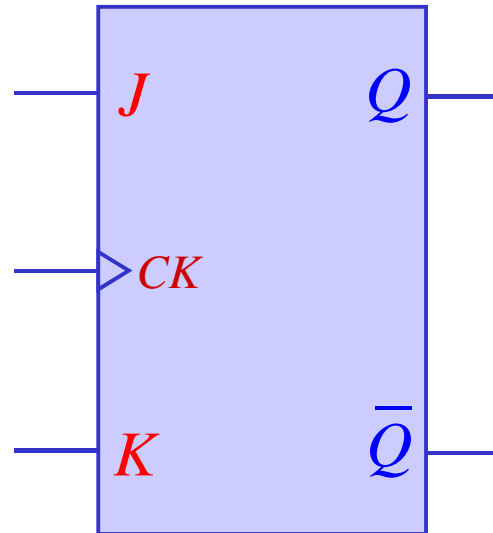
# *Flip-Flop D*

❑ Parámetros de timing: FF-D






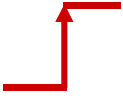
# *Flip-Flop JK*

❑ *Símbolo:* FF-JK



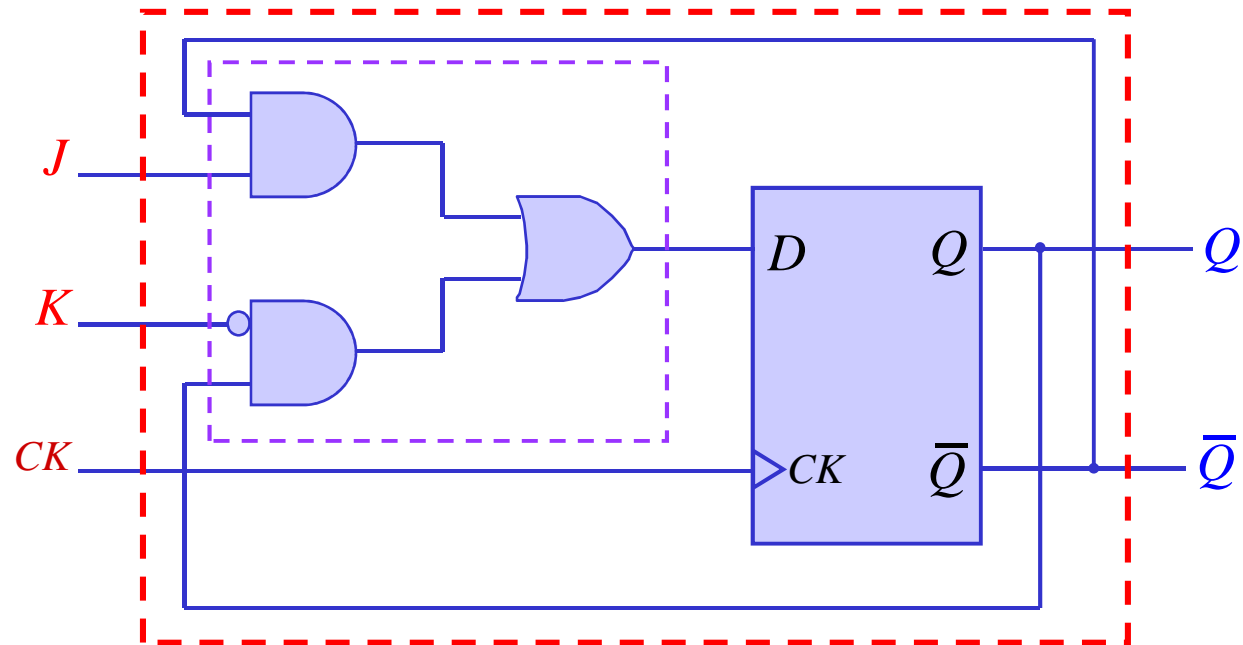
# *Flip-Flop JK*

❖ Tabla de verdad: FF-JK

J	K	CK	$Q$	$\bar{Q}$
X	X	0	$Q$	$\bar{Q}$
X	X	1	$Q$	$\bar{Q}$
0	0		$Q$	$\bar{Q}$
0	1		0	1
1	0		1	0
1	1		$\bar{Q}$	$Q$

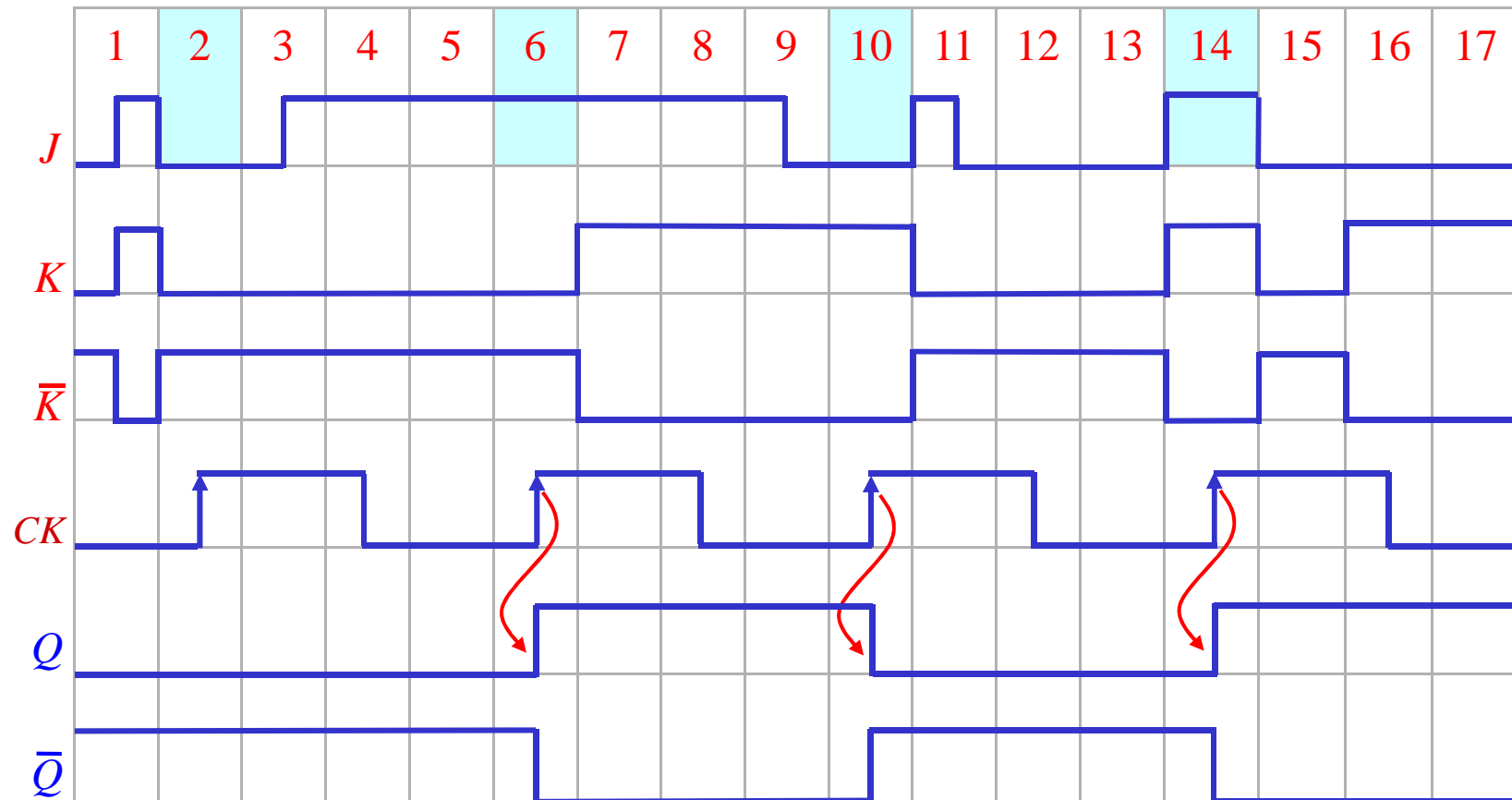
# *Flip-Flop JK*

- ❑ Diagrama lógico: FF-JK implementado con un FF-D



# Flip-Flop JK

❖ Diagrama de timing: FF-JK conmutado en el flanco



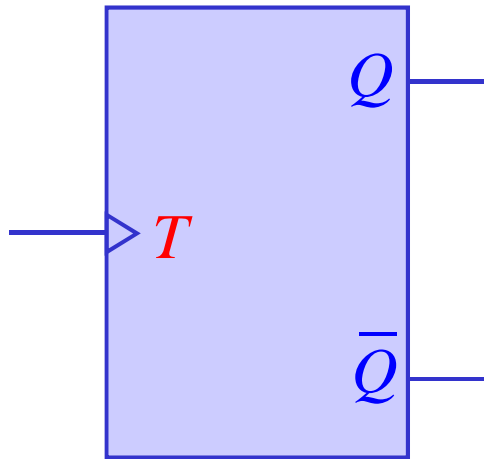
# *Flip-Flop T*

## ❑ Consideraciones


- ❖ El *flip-flop T (Toggle)* cambia de estado en cada flanco positivo o flanco negativo de la señal del reloj
- ❖ La *frecuencia* de la señal de *salida Q del flip-flop T* es exactamente la *mitad de la frecuencia* de la señal de *entrada T*
- ❖ El *flip-flop T* es frecuentemente usado en *contadores* y *divisores de frecuencia*

# *Flip-Flop T*

❖ Símbolo: FF-T

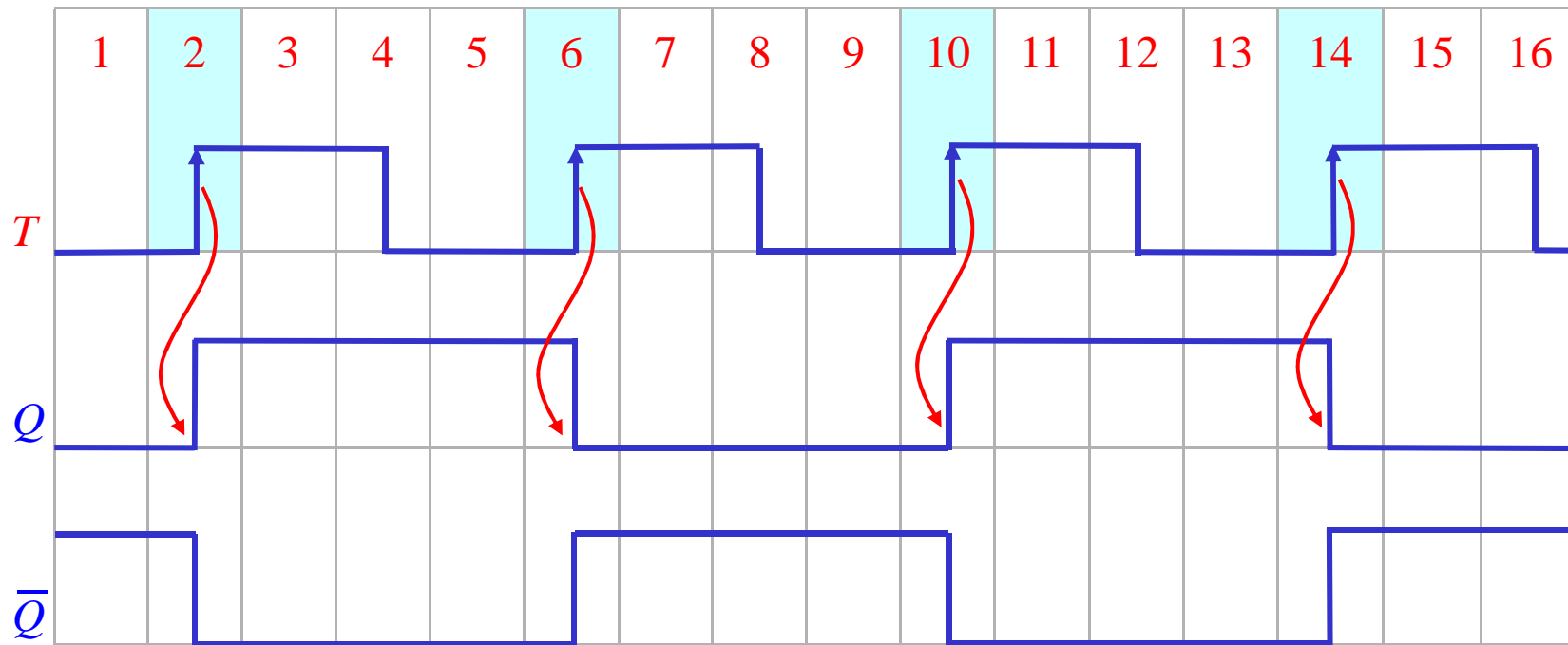


❖ Tabla de verdad: FF-T

$T$	$Q$	$\bar{Q}$
0	$Q$	$\bar{Q}$
1	$Q$	$\bar{Q}$
	$\bar{Q}$	$Q$

# *Flip-Flop T*

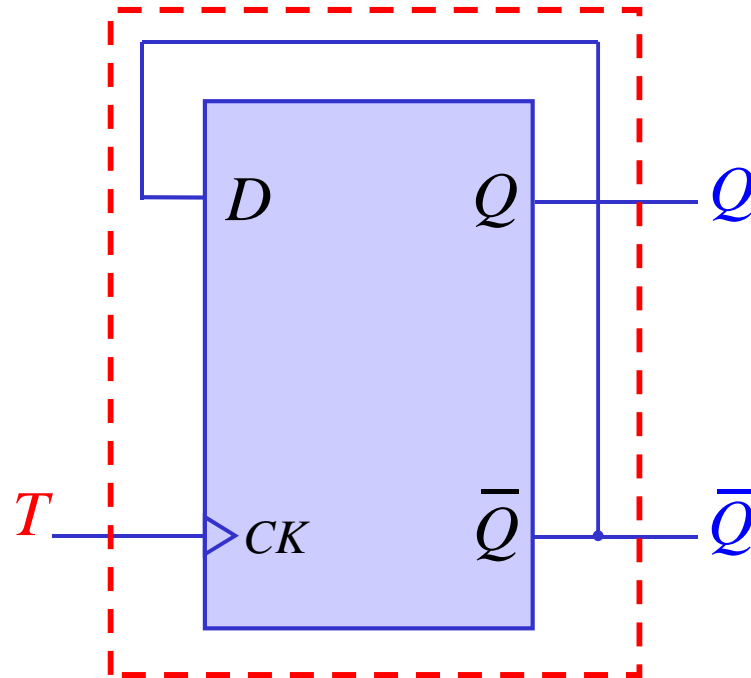
❖ Diagrama de timing: FF-T





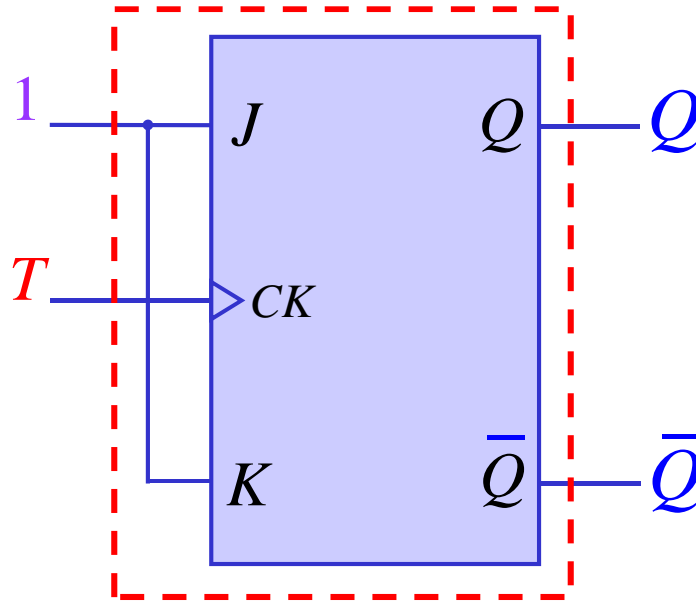
# *Flip-Flop T*

❖ *Flip-flop T* implementado con un *flip-flop D*



# *Flip-Flop T*

❖ *Flip-flop T* implementado con un *flip-flop JK*



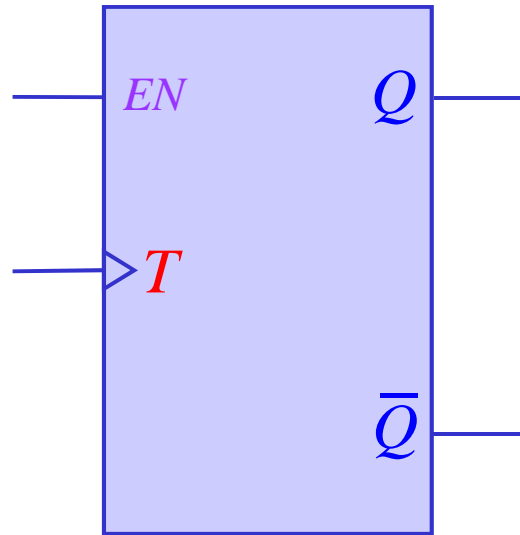
# *Flip-Flop T con Enable*

## ❑ Consideraciones

- ❖ En muchas aplicaciones el flip-flop T no necesita ser conmutado en cada ciclo de reloj, es decir, en cada flanco
- ❖ El *flip-flop T con enable* cambia o conmuta de estado en el flanco del reloj, únicamente si la *señal enable* está habilitada

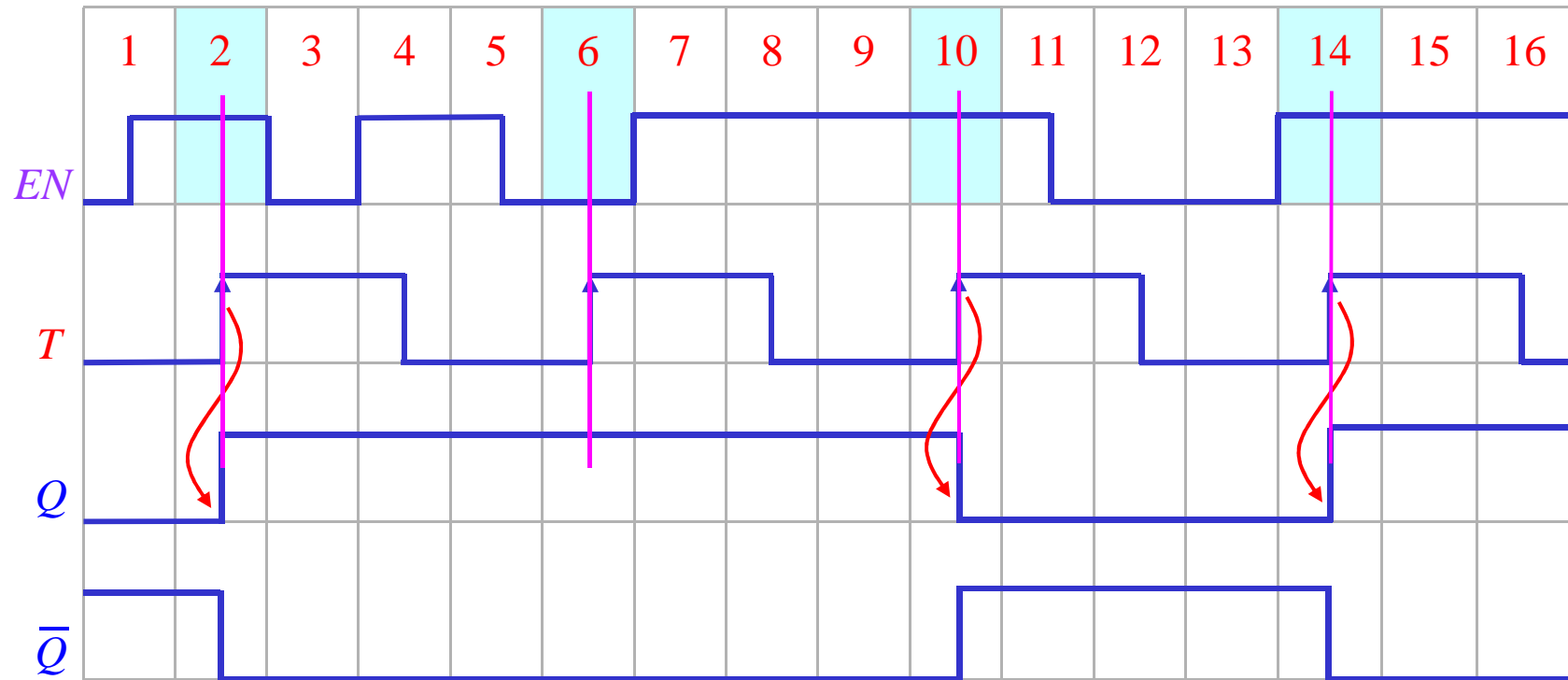
# *Flip-Flop T con Enable*

❖ Símbolo: FF-T con enable



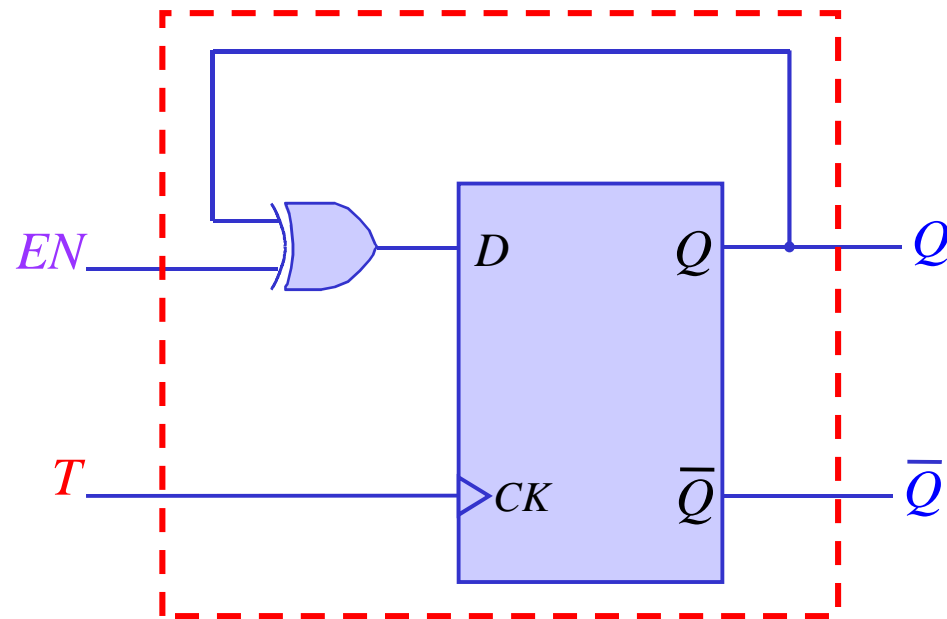
# *Flip-Flop T con Enable*

❖ Diagrama de timing: FF-T con enable



# *Flip-Flop T con Enable*

❖ *Flip-flop T con enable* implementado con un *flip-flop D*



# *Flip-Flop T con Enable*

❖ *Flip-flop T con enable* implementado con un *flip-flop JK*

