

En muchos circuitos digitales, el orden en el cual las cosas pasan es crítico:

- a veces, una acción tiene que preceder a otra
- a veces, dos acciones deben ocurrir simultáneamente

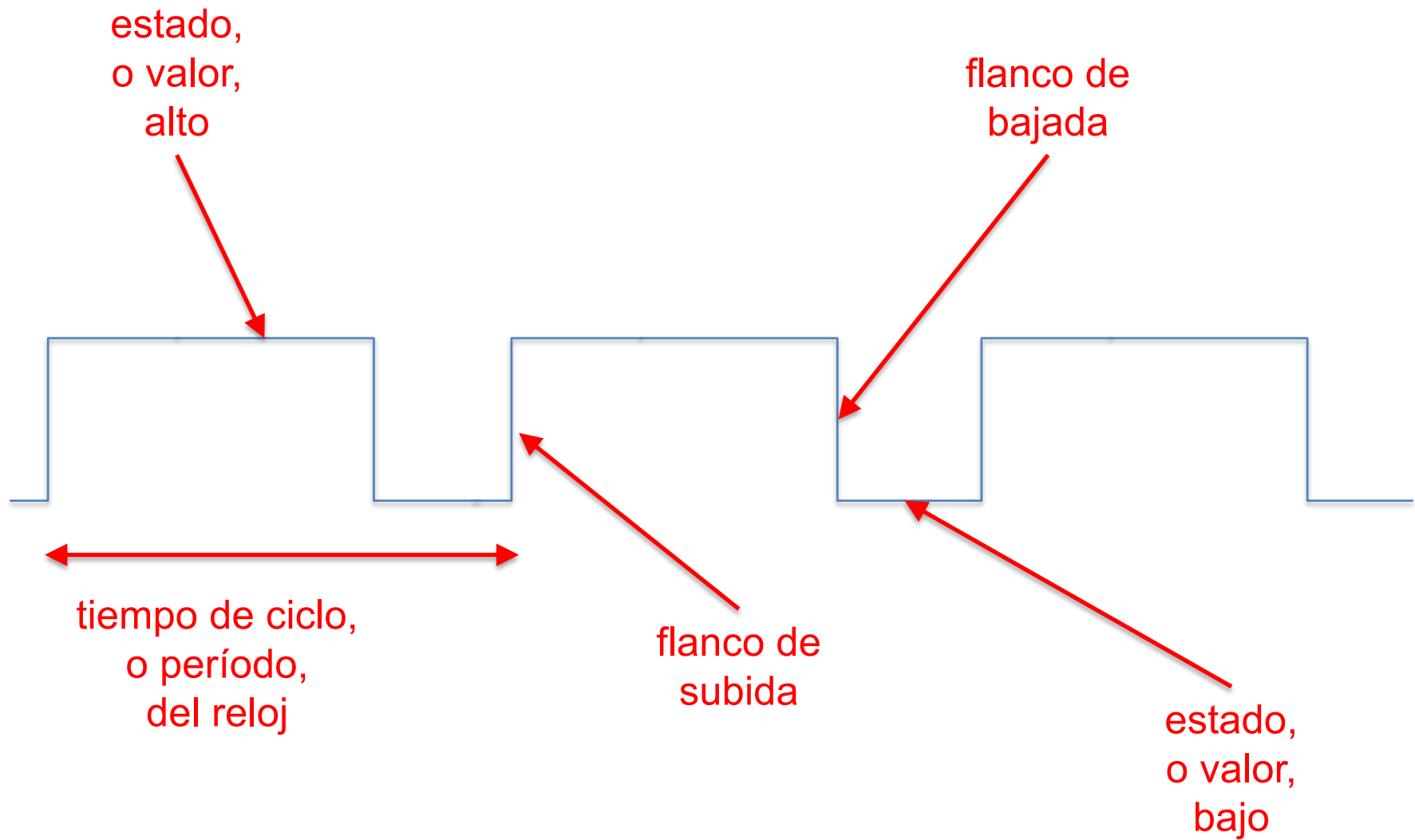
Los circuitos digitales usan *relojes* para proporcionar sincronización

Un **reloj** —en un circuito digital— es simplemente un circuito que emite una serie de pulsos, con dos propiedades:

- el ancho de pulso es preciso
- el intervalo entre pulsos consecutivos es preciso

El **tiempo de ciclo** del reloj es el intervalo de tiempo entre los flancos correspondientes —los *flancos de subida* o los *flancos de bajada*— de dos pulsos consecutivos

La **frecuencia** del reloj —p.ej., desde 100 MHz y hasta 4 GHz— es el inverso del tiempo de ciclo (o período)



Una componente esencial de todo computador es su *memoria*:

- sin memoria, no habría computadores tal como los conocemos hoy día
- la memoria se usa para almacenar tanto instrucciones a ser ejecutadas  
... como datos que se usan en la ejecución de las instrucciones

Para tener una memoria (de un bit), necesitamos un circuito que de alguna manera recuerde valores de input previos —es decir, que almacene su *estado*:

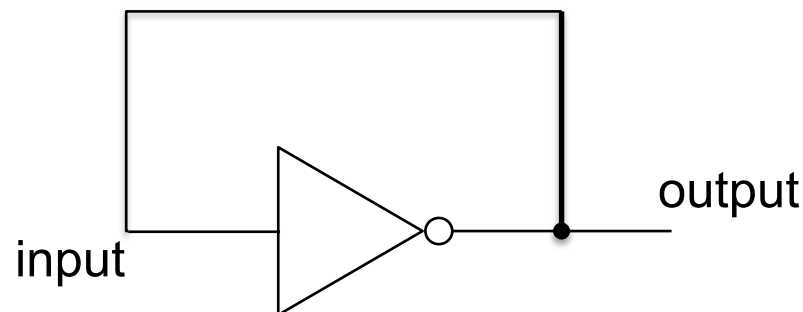
- el output del circuito depende tanto de los datos de input como del valor almacenado dentro del circuito

El funcionamiento de un *latch* SR se basa en que cada compuerta tiene un pequeño *retardo de propagación*:

- hay un retardo entre el instante en que el input cambia y el instante en que el output cambia consecuentemente
- lo construimos a partir de dos compuertas NAND (o dos compuertas NOR)

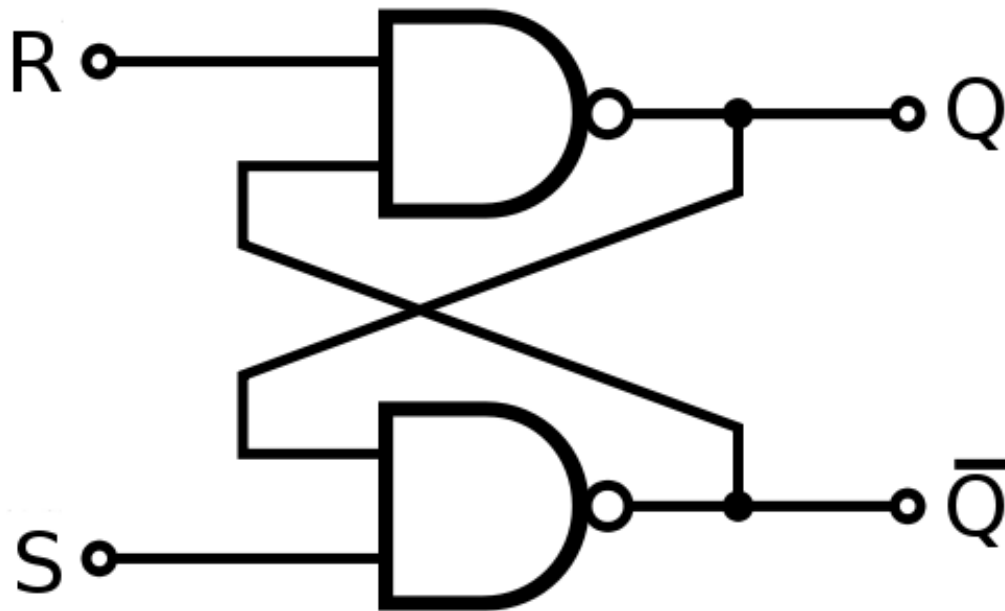
## Efecto del retardo de propagación en un circuito muy simple:

- si el output del circuito es 0, entonces el input del circuito es 0  
... esto parece una contradicción, ya que la (única) compuerta del circuito es un inversor (NOT)  
... excepto que sólo una vez que ha transcurrido el *retardo de propagación* (menos de un nanosegundo), el output cambia a 1  
... y sólo una vez que ha transcurrido otro retardo de propagación, el output vuelve a 0 nuevamente
- en principio, este ciclo continúa para siempre, por lo que el circuito oscila: el output cambia una y otra vez entre 0 y 1



## Latch SR:

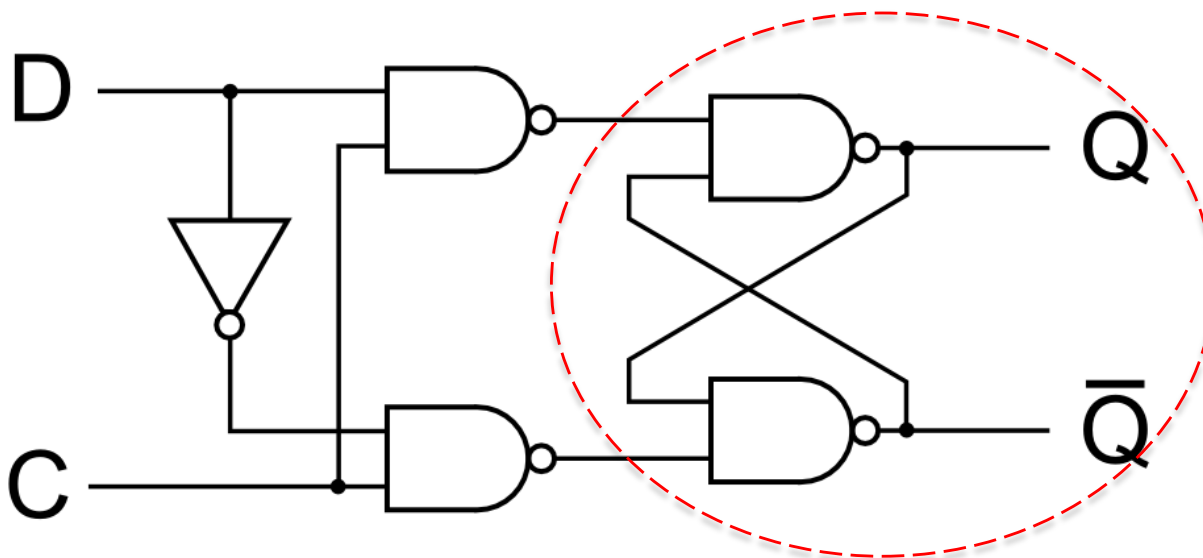
- construido a partir de dos compuertas NAND (o dos compuertas NOR)
- dos inputs,  $S$  (*setting*) y  $R$  (*resetting*)
- dos outputs,  $Q$  y  $\bar{Q}$ , complementarios
- los outputs no están determinados únicamente por los inputs vigentes —no es un circuito combinacional (o, más propiamente, *combinatorio*)
- cuando  $R = S = 0$ , el circuito se vuelve no determinista



R	S	$Q^{t+1}$
0	0	-
0	1	1
1	0	0
1	1	$Q^t$

**Latch D** (controlado por reloj) —a menudo conviene permitir que el latch cambie de estado sólo en ciertos instantes específicos:

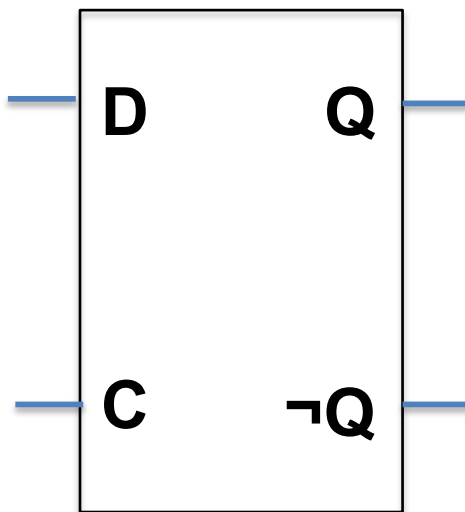
- evita el caso no determinista del *latch* SR previniendo su ocurrencia
- el input  $D$  y su complemento,  $\neg D$ , entran a sendas compuertas NANDs justo antes del *latch*
- ... → los inputs  $R$  y  $S$  del *latch* no pueden ser ambos 0 simultáneamente
- cuando el reloj,  $C$ , está en 1, el *latch* “se carga con” el valor de  $D$
- el valor almacenado en el *latch* está siempre disponible en el output  $Q$



C	D	$Q^{t+1}$
0	0/1	$Q^t$
1	0	0
1	1	1



**Latch D** (controlado por reloj) —más esquemáticamente:



*D*: dato (input)  
*C*: control (reloj)  
**Q**: estado (output)

C	D	$Q^{t+1}$
0	0/1	$Q^t$
1	0	0
1	1	1

En el diseño y construcción de computadores digitales, necesitamos circuitos que puedan leer el valor de una señal *en un instante particular en el tiempo* y almacenarlo

Esto elimina los problemas que podrían ocurrir si varias señales fueran leídas en momentos en el tiempo levemente diferentes

... que es lo que podría ocurrir en el caso de los *latches* como el anterior, en que la señal  $D$  puede ser leída en cualquier instante mientras el reloj está en su valor alto, o 1

Esto se puede conseguir con un circuito conocido como **flip-flop**

La transición de estado —es decir, el almacenamiento en el *latch* del valor que hay en el input *D*

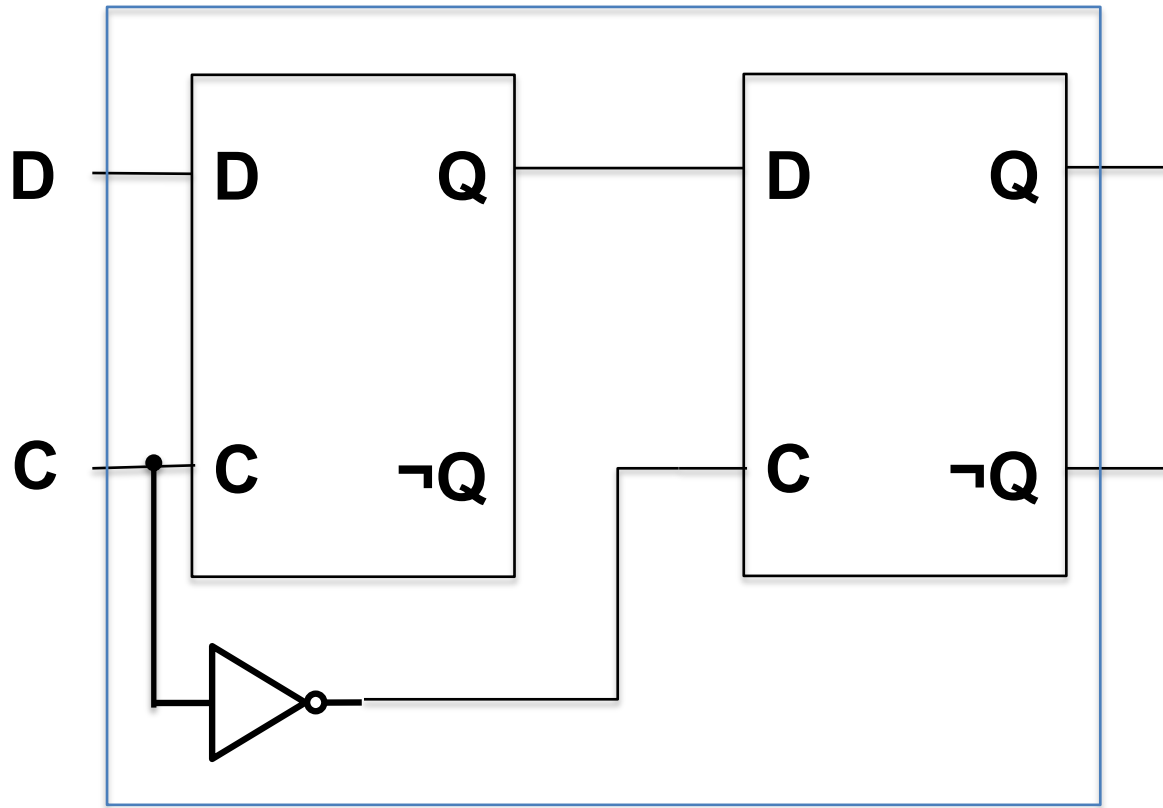
... ocurre **durante la transición** del reloj de 0 a 1 (*flanco de subida*) o de 1 a 0 (*flanco de bajada*)

( ... y no cuando el reloj está en 1 )

La próxima diapositiva muestra un flip-flop D construido a partir de dos latches D y un inversor (o compuerta NOT)

## Flip-flop D:

- cuando la señal del reloj  $C$  cambia de 0 a 1 (o, en otros circuitos equivalentes, de 1 a 0), el output  $Q$  almacena el valor del input  $D$
- usamos un arreglo de flip-flops D para construir un **registro** que puede almacenar un dato de varios bits, tal como un byte o una palabra



$C$	$D$	$Q^{t+1}$
0/1/ $\downarrow$	0/1	$Q^t$
$\uparrow$	0	0
$\uparrow$	1	1

Usamos un arreglo de flip-flops D para construir un **registro** ...

... que puede almacenar un dato de varios bits, tal como un byte o una palabra:

- p.ej., un registro de 4 bits:

