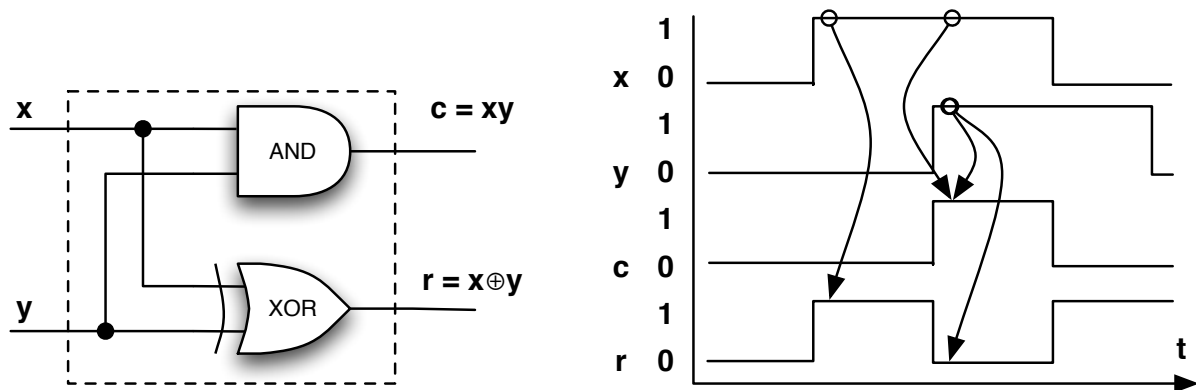


Sistemas Secuenciales

Un **circuito secuencial** es un circuito en donde las salidas no sólo dependen de los valores de las entradas en el presente, sino que también dependen de un estado interno del circuito. Este estado interno varía en función de las historia de los valores de las entradas.

Definición: **Diagrama de tiempo** es un gráfico de los valores que toman un conjunto de líneas (señales, conexiones) en función del tiempo.



Ejemplo: Circuito C3 con su diagrama de tiempo

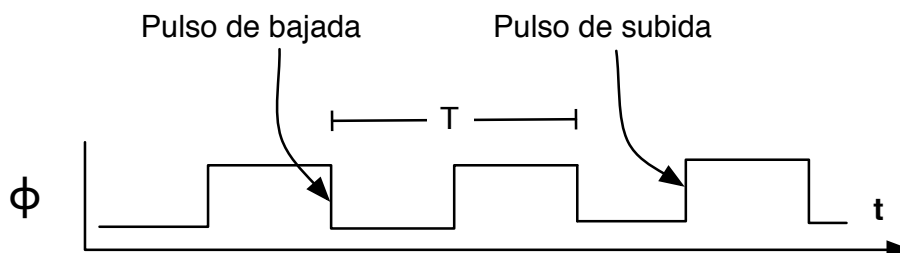
En C3, los valores de **c** y **r** se calculan en función de los valores de **x** e **y** en el mismo instante. (En la práctica, el cambio en **c** o **r** siempre se produce con un pequeño retardo, después que cambian **x** o **y**, hoy es +- 10 picosegundos = 10^{-11} segundo)

La flecha indica causalidad: el cambio de **x** origina el cambio en **r**. Al contrario, no hay causalidad entre el cambio de **x** e **y**. En un diagrama de tiempo, no es necesario colocar todas las flechas de causalidad, sólo se colocan aquellas que sean útiles para el lector del diagrama.

Resumiendo: en el ejemplo de circuito combinacional C3

$$\begin{aligned} c(t) &= f(x(t), y(t)) \\ r(t) &= g(x(t), y(t)) \end{aligned}$$

Definición: **Reloj (ϕ)** es una compuerta que genera una señal que cambia periódicamente de valor.



Valores típicos de período del reloj $T = 125$ microsegundos a 0.3 nanosegundos.
frecuencia = $f = 1/T = 8\text{Mhz}$ a 3.3 Ghz

Circuitos Secuenciales

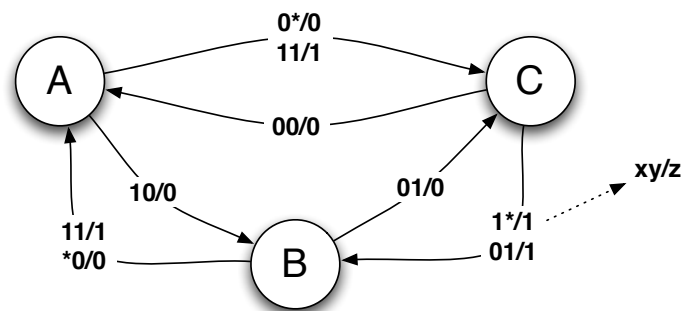
En este tipo de circuitos, las salidas no sólo dependen de las entradas, sino que también dependen del estado interno del circuito. Concretamente:

1. Un circuito secuencial **posee un número finito de estados**. En un instante dado el circuito se encuentra en uno y solo uno de sus estados.
2. Una de las entradas del circuito secuencial se denomina el reloj. El intervalo entre dos pulsos de bajada del reloj es un ciclo del circuito. **El estado del circuito se mantiene constante durante un ciclo del reloj.**
3. En un instante dado, el circuito **calcula sus salidas en función de sus entradas**, exceptuando el reloj, y de su estado interno.
4. **Al final de cada ciclo** (justo antes del pulso de bajada del reloj) el circuito calcula su estado válido para el próximo ciclo en función de su estado actual y sus entradas.

Un circuito secuencial actúa como circuito combinacional dentro de un ciclo. Si cambian sus entradas, también pueden cambiar sus salidas.

Especificación de circuitos secuenciales: Diagrama de estados

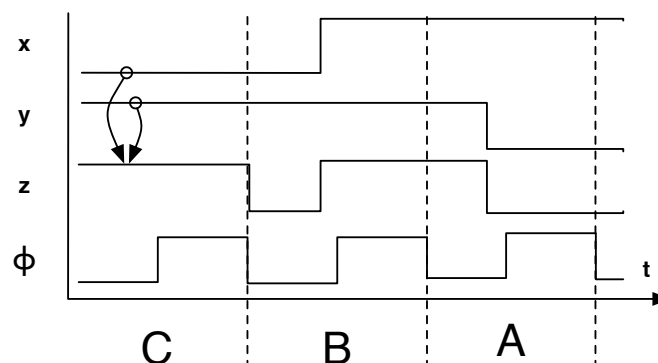
Un diagrama de estados especifica **formalmente** el comportamiento de un circuito secuencial.



Ejemplo: Diagrama de Estados E1 para un circuito con entradas **x** e **y** y salida **z**.

Los círculos rotulados representan los estados del circuito. Las flechas representan las transiciones de estado y los rótulos especifican las salidas en función de las entradas para el estado del cual sale la flecha.

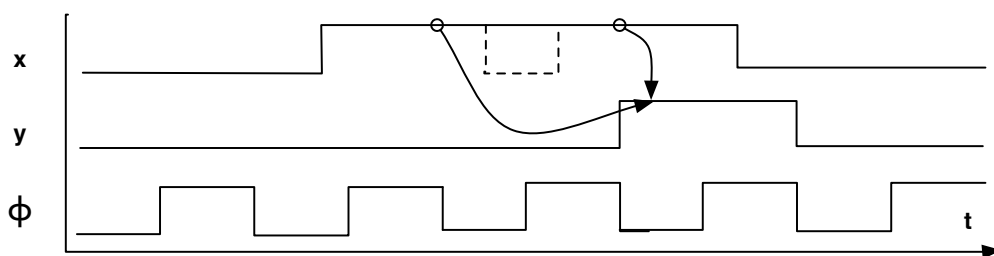
El comportamiento **informal** de un circuito secuencial se aprecia mejor en un diagrama de tiempo.



Un diagrama de estados especifica completamente un circuito secuencial. Por lo tanto por cada estado salen flechas con rótulos para todas las combinaciones de entradas posibles. El * en una entrada se usa para especificar 2 rótulos simultáneamente: uno con esa entrada en 0 y el otro en 1.

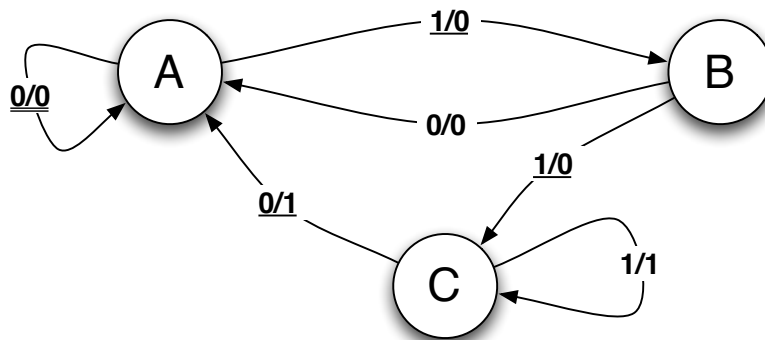
Al contrario de un diagrama de estado, un diagrama de tiempo es una especificación incompleta del circuito. Pueden haber muchos diagramas de estado que satisfacen el mismo diagrama de tiempo. Pero dado un diagrama de estado, un estado inicial y valores para sus entradas durante un intervalo de tiempo, el diagrama de tiempo para sus salidas es uno y sólo uno.

Sin embargo, un problema usual es encontrar al menos un diagrama de estado que satisfaga un diagrama de tiempo. Por ejemplo, el siguiente diagrama de tiempo describe informalmente un circuito detector de secuencias 1 1 en su entrada.



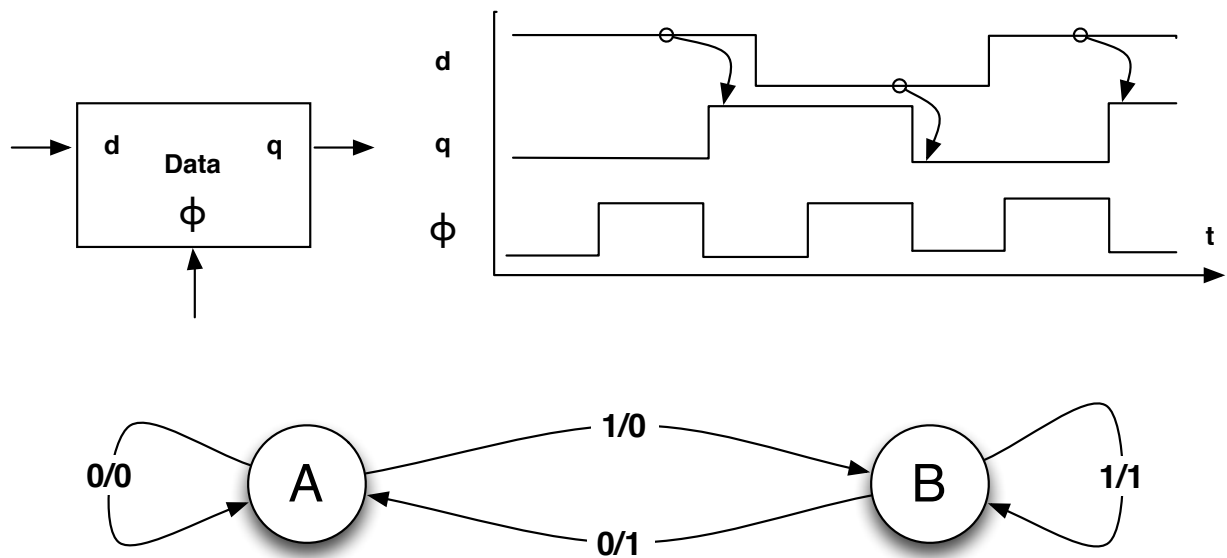
Observe que el circuito secuencial solo puede registrar lo que sucede al final de cada ciclo. El circuito puede señalar que hubo dos 1 consecutivos aún cuando x tuvo una corta transición a 0 en la mitad del ciclo.

Un diagrama de estado para este diagrama de tiempo es:



Los rótulos subrayados son necesarios para que este diagrama de estado satisfaga el diagrama de tiempo anterior.

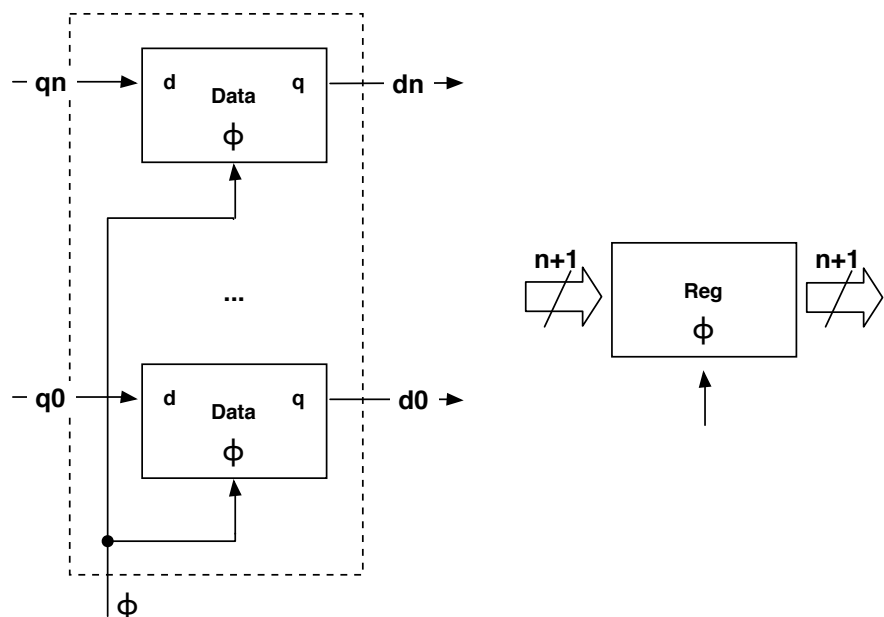
El circuito secuencial mínimo: el Flip-Flop Data



En un ciclo, el valor de **q** es el valor que tenía **d** justo antes del comienzo del ciclo (i.e. el pulso de bajada del reloj).

El flip-flop data es el primero de los biestables que veremos. Es importante porque permite almacenar un valor binario durante todo un ciclo. Si se desea almacenar varios bits basta colocar varios flip-flop en paralelo.

Los microprocesadores usan flip-flops data para almacenar los registros.

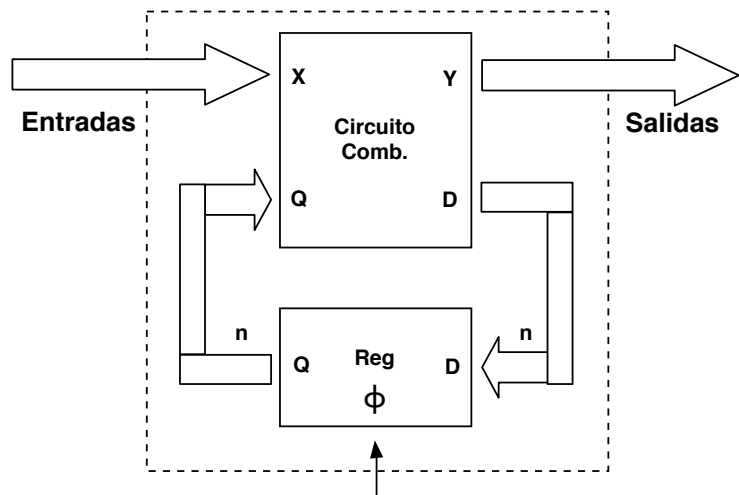


Implementación de circuitos secuenciales

El problema esencial es representar y almacenar el estado interno del circuito. Supongamos que sabemos implementar el flip-flop data. Entonces podemos codificar los estados de cualquier circuito secuencial en la forma de números binarios y luego utilizar flip-flops data para almacenar la codificación elegida.

Forma general de la implementación de un circuito secuencial:

Este circuito secuencial puede poseer a lo más 2^n estados posibles. En cada ciclo el circuito combinatorio recibe las entradas del circuito secuencial: (**X**) y la codificación del estado actual (**Q**), los que usa para calcular las salidas (**Y**) y la codificación del próximo estado (**D**). El registro almacena **D** en el pulso de bajada del reloj y se lo envía al circuito combinatorio (**Q**), lo que marca el comienzo de un nuevo ciclo.



Metodología de diseño de un circuito secuencial, a partir de un diagrama de estado

Presentaremos esta metodología aplicando al diagrama de estados E1

Primero: Codificación de estados: para representar 3 estados, se necesitan $\lceil \log_2 3 \rceil$ flip-flops data:

Estado	q1	q0
A	0	0
B	0	1
C	1	0

Segundo: Tablas de verdad: a partir del diagrama del estado se construye una tabla de verdad para especificar las salidas **Z**, **d₁** y **d₀** del circuito combinacional.

	q ₁	q ₀	x	y	d ₁	d ₀	z
A	0	0	0	0	1	0	0
A	0	0	0	1	1	0	0
A	0	0	1	0	0	1	0
A	0	0	1	1	1	0	1
B	0	1	0	0	0	0	0
B	0	1	0	1	1	0	0
B	0	1	1	0	0	0	0
B	0	1	1	1	0	0	1

	q ₁	q ₀	x	y	d ₁	d ₀	z
C	1	0	0	0	0	0	0
C	1	0	0	1	0	1	1
C	1	0	1	0	0	1	1
C	1	0	1	1	0	1	1
?	1	1	0	0	X	X	X
?	1	1	0	1	X	X	X
?	1	1	1	0	X	X	X
?	1	1	1	1	X	X	X

Tercero: Mapas de Karnaugh y fórmulas algebraicas

$q_1 q_0 \backslash xy$	0 0	0 1	1 1	1 0
0 0	1	1	1	0
0 1	0	1	0	0
1 1	X	X	X	X
1 0	0	0	0	0

$$d_1 = \sim q_1 \sim q_0 \sim x + \sim q_1 \sim q_0 y + \sim q_1 \sim xy$$

$q_1 q_0 \backslash xy$	0 0	0 1	1 1	1 0
0 0	0	0	0	1
0 1	0	0	0	0
1 1	X	X	X	X
1 0	0	1	1	1

$$d_0 = q_1 y + \sim q_0 x \sim y$$

$q_1 q_0 \backslash xy$	0 0	0 1	1 1	1 0
0 0	0	0	1	0
0 1	0	0	1	0
1 1	X	X	X	X
1 0	0	1	1	1

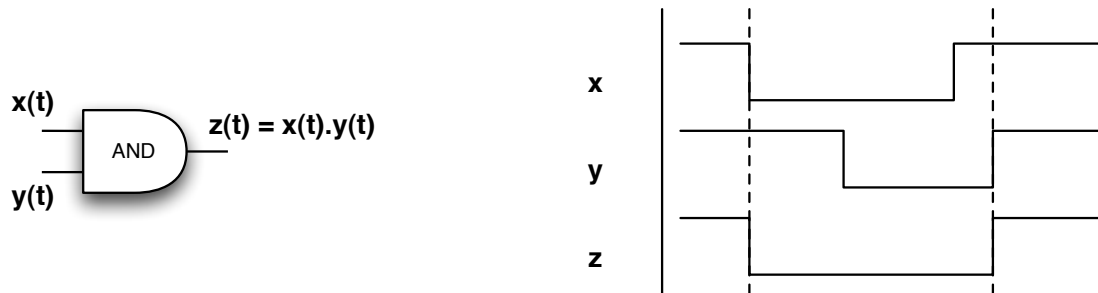
$$z = xy + q_1 y + q_1 x$$

Cuarto: Diagrama circuital completo (ejercicio): basta hacer el circuito para el circuito combinacional y insertarlo en el circuito secuencial.

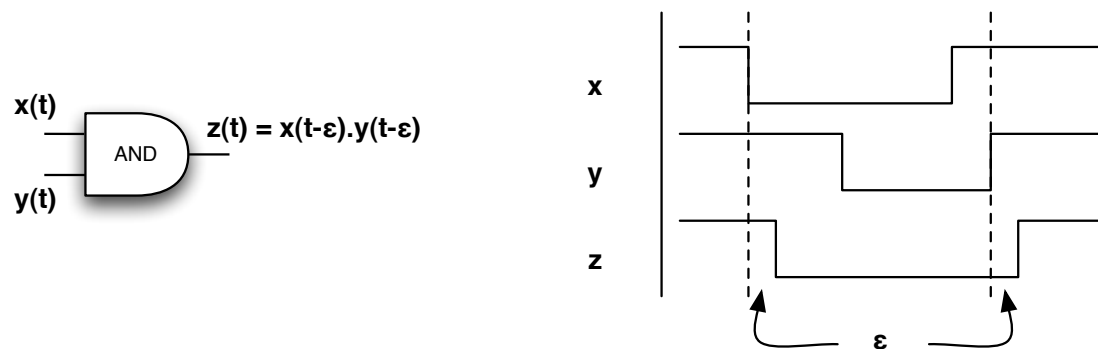
Implementación de Biestables

Previo: Tiempos de Retardo

Hasta el momento se ha supuesto que las salidas de una compuerta elemental (ej: AND) se calculan en forma instantánea. En un diagrama de tiempo, esto se graficaría:



En la realidad una compuerta calcula sus salidas con un cierto tiempo de retardo ϵ :

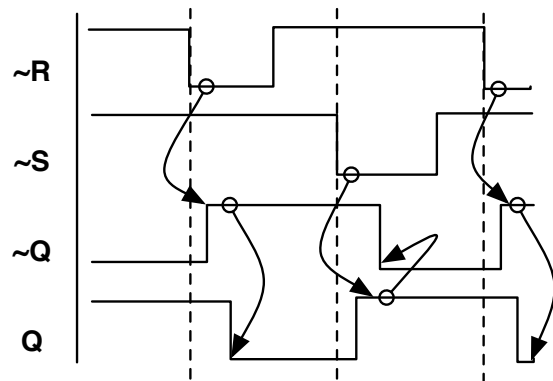


No todas las compuertas tienen el mismo retardo, pero por simplicidad supondremos el mismo tiempo de retardo para todas las compuertas básicas. Para n compuertas básicas anidadas (puestas en cascada) el tiempo de retardo es **$T = n \cdot \epsilon$**

Para un sumador de números de n bits que usa el circuito sumador de 3 bits en cascada (i.e. la generalización del sumador de 4 bits visto en la fin de circuitos combinacionales). Si en un instante dado el bit menos significativo de un argumento cambia, después de cuanto tiempo se ha calculado la nueva suma?

Solución: Si cada sumador de 3 bits tiene un retardo 3ϵ , **$T = 3\epsilon n$**

El Biestable Reset/Set



Descripción: Cuando $\sim R = \sim S = 1$, Q y $\sim Q$ se mantienen constantes y con valores opuestos:

Si $\sim R \rightarrow 0 \Rightarrow Q \rightarrow 0, \sim Q \rightarrow 1$

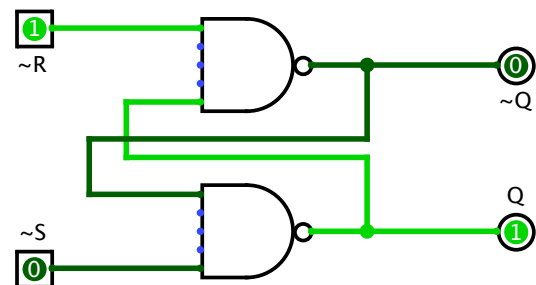
Si $\sim S \rightarrow 0 \Rightarrow Q \rightarrow 1, \sim Q \rightarrow 0$

¡ $\sim S = \sim R = 0$ esta prohibido ! (Por que ?)

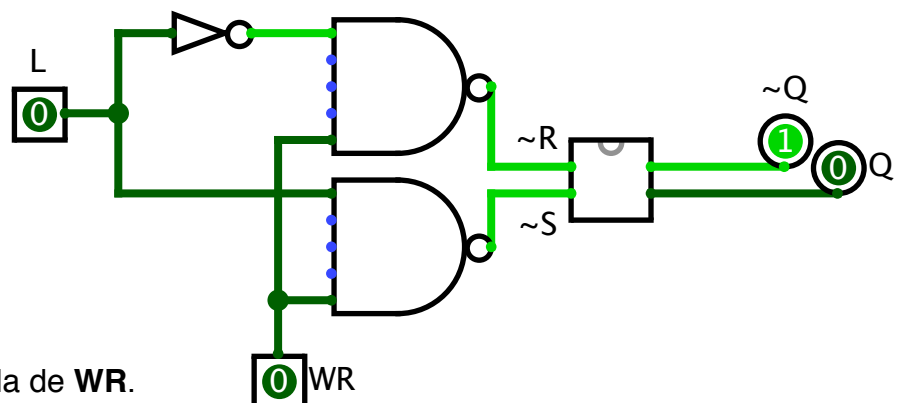
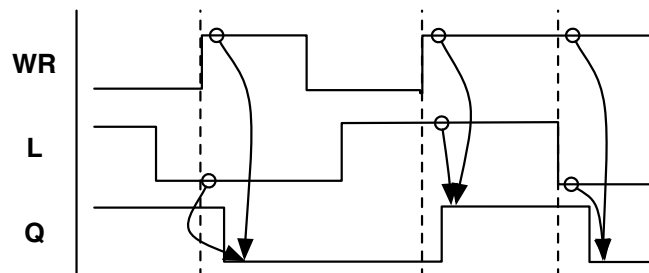
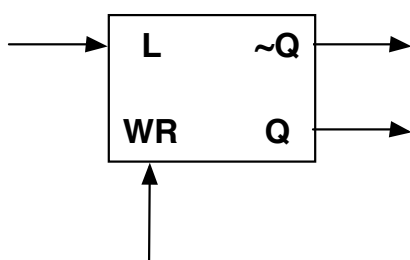
Observacion: **R** viene de Reset, y **S** de set.

La negación sobre **R** y **S** ($\sim R$ y $\sim S$) significa que la acción Reset o Set se produce cuando la señal está en **0** (y no en 1).

Existe un biestable que usa **R** y **S**, pero históricamente ha sido más caro de fabricar, (usa NOR en vez de NAND) entonces no se usa.



El Latch



Descripción:

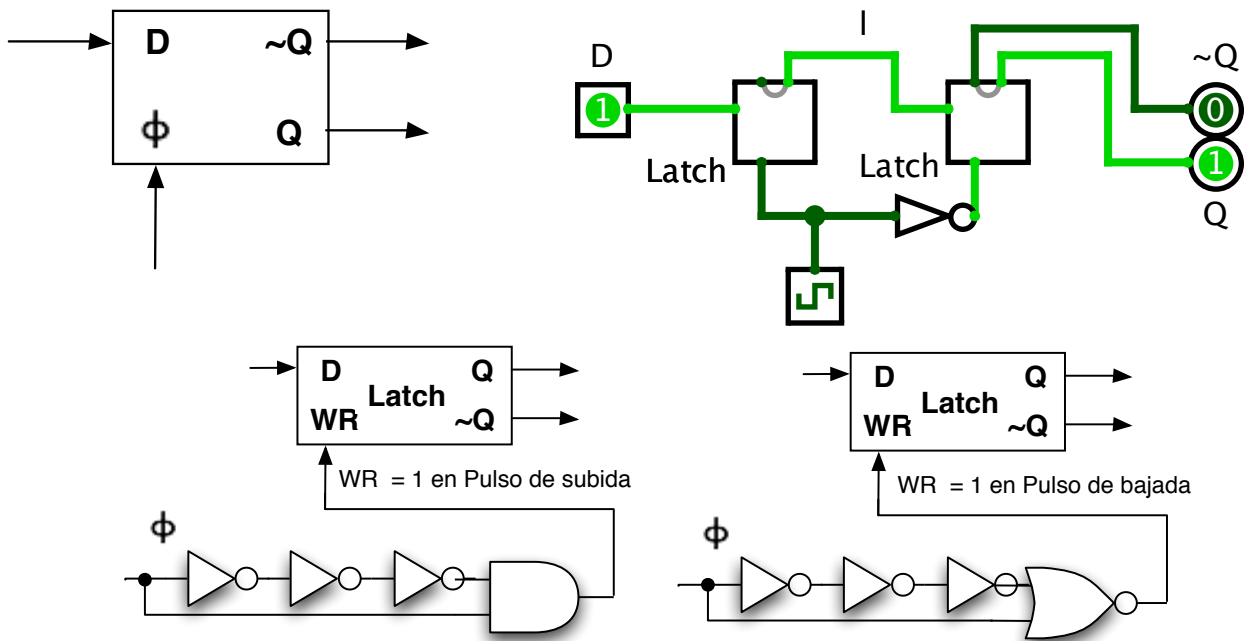
Mientras **WR** = 1, **Q** = L.

Cuando **WR** = 0,

Q vale lo que tenía **L**

justo antes del pulso de bajada de **WR**.

El Flip-Flop Data

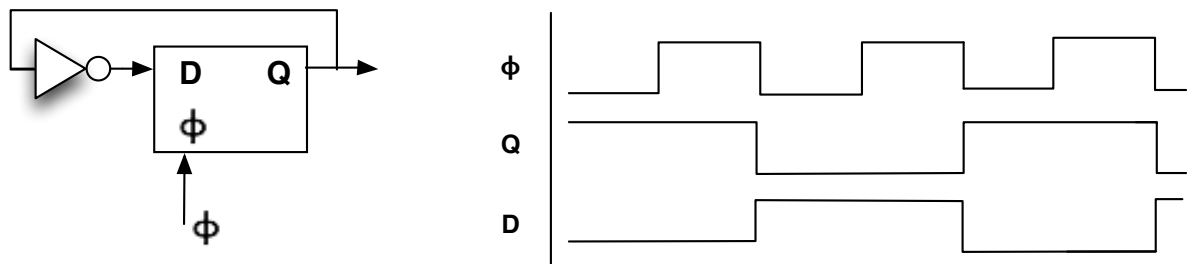


La implementación visto es de un registro master/slave: equivalente a un flip-flop data pero mas robusto, porque el flip-flop data depende de los tiempos de retardo:.

Ejercicio: Verifique en un diagrama de tiempo que el Flip Flop Data sólo cambia en el pulso de bajada de ϕ . Para ello fije valores para D y ϕ y en los valores iniciales de I y Q . Luego calcula L , Q y $\sim \phi$ en función de D y ϕ .

Observación: El latch no sirve para hacer circuitos secuenciales. Ejemplo: en el siguiente circuito secuencial si se reemplaza el Flip Flop Data por un latch:

Cuando ϕ está en 1, Q adquiere valores oscilatorios que difícilmente pueden servir para



algo.

