



1. CIRCUITOS LOGICOS SECUENCIALES

Se denominan circuitos secuenciales a aquellos circuitos lógicos cuya salida no esta condicionada solamente por la combinación de las variables de entrada, sino también por el orden de las mismas. Podemos decir que un circuito secuencial posee salidas que estarán en **0** o **1** lógico dependiendo no solo del valor actual de las variables de entrada, sino también de la historia del sistema. Dicho de otra forma, se dice que un circuito secuencial posee una cierta memoria.

En electrónica, un circuito que cumple con la definición anterior es el denominado **biestable**. Este se caracteriza por presentar dos salidas complementarias denominadas **Q** y \overline{Q} . Estas salidas presentan dos estados estables (0 y 1), significando ello que pueden permanecer en forma indefinida en alguno de estos estados, aún cuando haya desaparecido la causa que originó su cambio. El cambio en la salida de estos circuitos se produce a través de entradas de control.

1.1. Clasificación de los circuitos biestables

Existen una gran variedad de biestables con distintas particularidades en relación a la lógica de control o disparo, formas de las señales de control (nivel o flancos ascendentes o descendentes), sincrónicos o no sincrónicos, con entradas sincrónicas o asincrónicas, etc. Veamos su clasificación y sus características principales.

1.2. Clasificación por la lógica de control

Esta clasificación surge de la forma lógica de cambio de las salidas del biestable (**Q** y \overline{Q}) por el cambio en los valores lógicos de las entradas de control. Se clasifican en:

- R – S (Reset – Set)
- J – K
- D
- T

1.3. Clasificación según la forma de las señales de control

- Los que producen el cambio lógico de sus salidas (disparo) mediante **niveles** de tensión en las señales de control. Estos biestables se denominan *básculas* o *latch*.
- Los que producen el cambio de sus salidas (disparo) mediante el **cambio** de los niveles de tensión de las entradas de control. El disparo puede producirse con el flanco de subida o con el flanco de bajada de las señales de control. Estos biestables se denominan *Flip Flop*.

1.4. Clasificación según el sincronismo o no de una señal patrón (reloj)

Se entiende por reloj a un generador de pulsos eléctricos (oscilador de onda rectangular o cuadrada) que se utiliza para sincronizar el funcionamiento de los circuitos secuenciales complejos. Según esto, los biestables se clasifican en:

- Asincrónicos: funcionan sin el control de una señal de frecuencia patrón.
- Sincrónicos sencillos: son biestables de nivel que funcionan con una frecuencia patrón.
- Sincrónicos, Edge-Triggered: son biestables cuyas salidas se modifican con el cambio de nivel de las señales de control, sincronizadas con una frecuencia patrón.
- Sincrónicos Master-Slave: de funcionamiento similar a los Edge-Triggered, pero ya obsoletos.

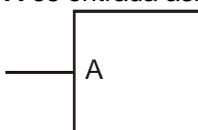
1.5. Entradas sincrónicas y asincrónicas

Los biestables disponen de entradas de control que pueden estar en sincronismo o no con la señal reloj de sincronismo. Por ejemplo las señales de entradas de control R-S, J-K, D o T son sincrónicas. Existen otras señales de control que modifican las salidas (**Q** y \overline{Q}) y que no están sincronizadas con la señal de reloj. Estas entradas son *puesta a 1* (preset) y *borrado o puesta a 0* (clear).

1.6. Simbología utilizada para las entradas

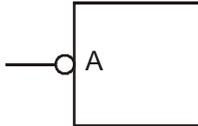
Consideremos una entrada genérica **A**, pueden darse los siguientes casos:

- **A** es entrada asincrónica activada por nivel alto de tensión (1 lógico)

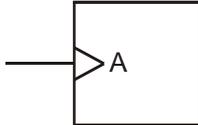




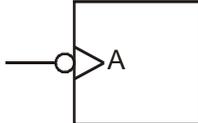
- **A** es entrada asincrónica activada por nivel bajo de tensión (0 lógico)



- **A** es entrada síncrona activa por flanco de subida (transición de 0 a 1)



- **A** es entrada síncrona activa por flanco de bajada (transición de 1 a 0)



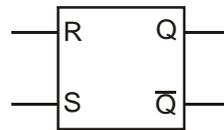
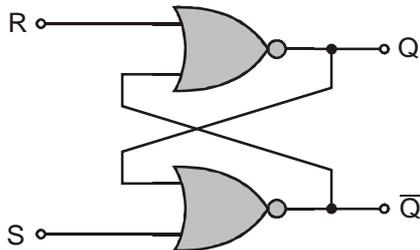
2. BASCULAS ASINCRONICAS (LATCH)

Son circuitos biestables, asincrónicos (o sea que no tienen entrada de pulsos de reloj) y son activados por niveles de tensión en sus entradas.

2.1. Básulas R-S (Set – Reset)

Estas básulas se pueden realizar con compuertas NOR o con compuertas NAND. Veamos su circuito, tabla de verdad, y simbología:

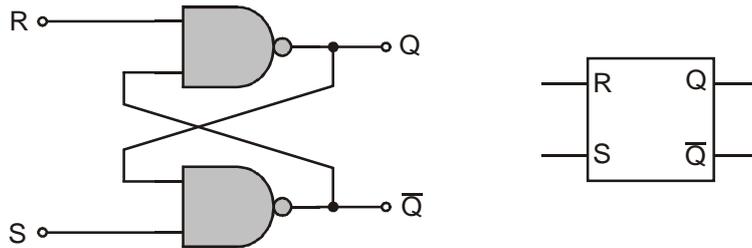
- **R-S NOR:**



S	R	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
0	0	$Q_{(t)}$	$\bar{Q}_{(t)}$
0	1	0	1
1	0	1	0
1	1	0	0



• **R-S NAND:**



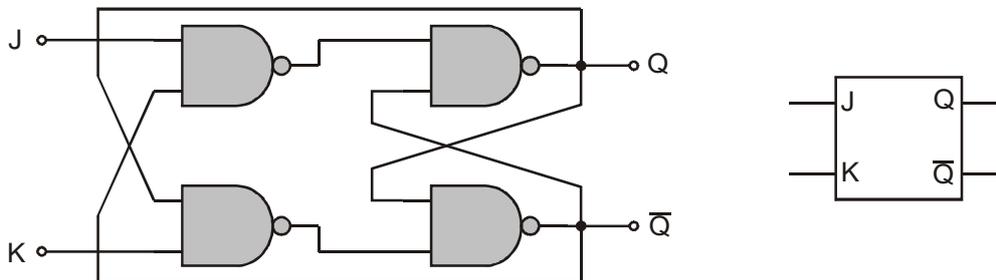
S	R	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
0	0	1	1
0	1	0	1
1	0	1	0
1	1	$Q_{(t)}$	$\bar{Q}_{(t)}$

En las tablas de verdad, $Q_{(t)}$ simboliza el estado de la salida Q antes de producirse el nuevo cambio de las señales de control R-S. Asimismo, $Q_{(t+1)}$ simboliza el estado de la salida Q después de producirse el cambio de las señales de control.

Observar que tanto en los latch NOR y NAND existen combinaciones de las señales de entrada que deben evitarse ($R = S = 1$ para el latch NOR, y $R = S = 0$ para el latch NAND), ya que con estas combinaciones las salidas Q y \bar{Q} toman valores iguales, lo que resulta en una indeterminación (las resaltadas en rojo en cada tabla)

2.2. Báscula J-K

Esta báscula elimina la condición de indeterminación de las anteriores. Se realiza partiendo de una báscula R-S NAND a la que se le agregan en sus entradas dos compuertas NAND. El circuito, simbología y tablas son las siguientes:



J	K	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
0	0	$Q_{(t)}$	$\bar{Q}_{(t)}$
0	1	1	0
1	0	0	1
1	1	$\bar{Q}_{(t)}$	$Q_{(t)}$

3. BASCULAS SINCRONICAS (FLIP FLOP)

Estos también son circuitos biestables, pero en este las salidas se activan según el cambio de las entradas de control y un pulso de reloj. Según la báscula, la activación puede ser por el flanco ascendente o por el flanco descendente del pulso de reloj.

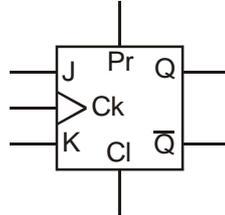
3.1. Flip Flop J-K

Este flip flop posee una entrada “reloj” para sincronizar las entradas de control J y K. Las salidas del flip flop responden a la lógica de estas entradas cuando ingresa el flanco de la señal de reloj; algunos flip flop se diseñan para responder al flanco ascendente del reloj, y otros al flanco descendente.



Este flip flop también posee dos entradas asincrónicas **preset** (puesta a 1) y **clear** (puesta a 0), con las que se puede modificar la salida independientemente del valor que tomen las entradas J-K y del sincronismo del reloj. Algunos flip flop se diseñan para que estas entradas asincrónicas sean activas en 0, y en otros estas entradas son activas en 1. Para que el flip flop trabaje con sus entradas J-K y reloj, las entradas clear y preset deben dejarse desactivadas.

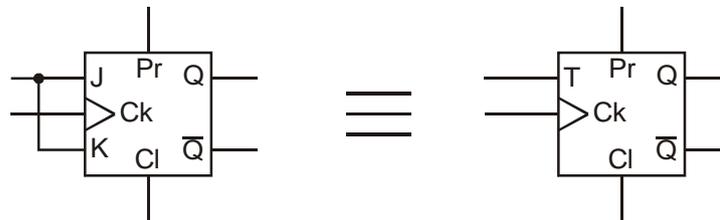
Veamos el símbolo y la tabla de verdad de un flip flop J-K con entradas síncronas por flanco ascendente y entradas asincrónicas activas en alto:



Pr	Cl	J	K	Ck	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
0	1	X	X	X	0	1
1	0	X	X	X	1	0
1	1	X	X	X	1	1
0	0	0	0		$Q_{(t)}$	$\bar{Q}_{(t)}$
0	0	0	1		0	1
0	0	1	0		1	0
0	0	1	1		$\bar{Q}_{(t)}$	$Q_{(t)}$

3.2. Flip Flop T

Este flip flop se obtiene puentando las entradas de un flip flop J-K. Se denomina flip flop T por Toggle (conmutar), ya que de la tabla de verdad resultante se puede observar que cuando la entrada T es 0, la salida Q permanece sin cambio; pero cuando T es 1 la salida Q es conmutada en cada ciclo de reloj.

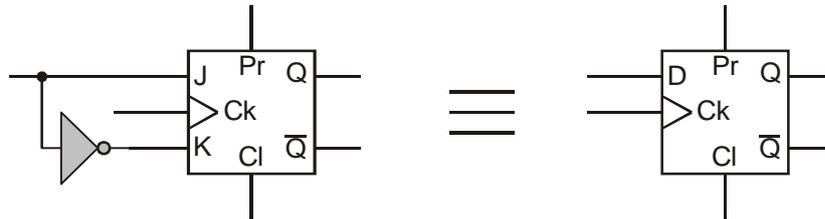


Pr	Cl	T	Ck	$Q_{(t+1)}$	$\bar{Q}_{(t+1)}$
0	1	X	X	0	1
1	0	X	X	1	0
1	1	X	X	1	1
0	0	0		$Q_{(t)}$	$\bar{Q}_{(t)}$
0	0	1		$\bar{Q}_{(t)}$	$Q_{(t)}$



3.3. Flip Flop D

Este tipo de flip flop también se obtiene a partir de un J-K, pero en este caso se coloca una compuerta NOT entre las entradas J y K. Se denomina flip flop D por Data (sigue el dato en la entrada). Es decir, en cada transición de la señal de reloj, el dato en la entrada D es copiado a la salida Q.



Pr	Cl	T	Ck	$Q_{(t+1)}$	$\overline{Q}_{(t+1)}$
0	1	X	X	0	1
1	0	X	X	1	0
1	1	X	X	1	1
0	0	0		0	1
0	0	1		1	0

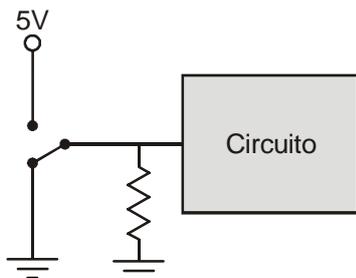
4. APLICACION DE LOS BIESTABLES

Los biestables tienen numerosas aplicaciones, por ejemplo:

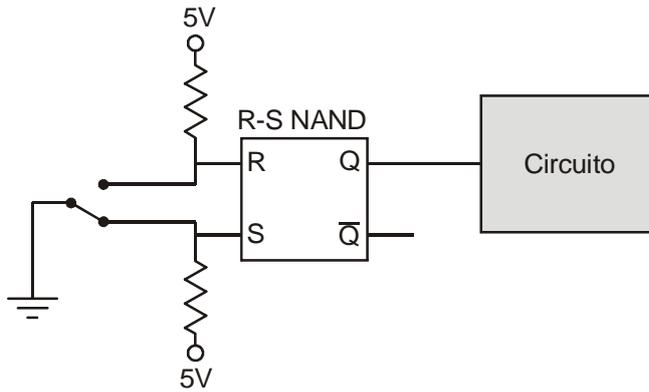
- Memorias activas.
- Contadores binarios.
- Registros de desplazamientos.
- Sistemas secuenciales para activar micro órdenes en unidades de control de microprocesadores.
- Eliminadores de rebotes de conmutadores.
- Etc.

4.1. Conmutador sin rebotes

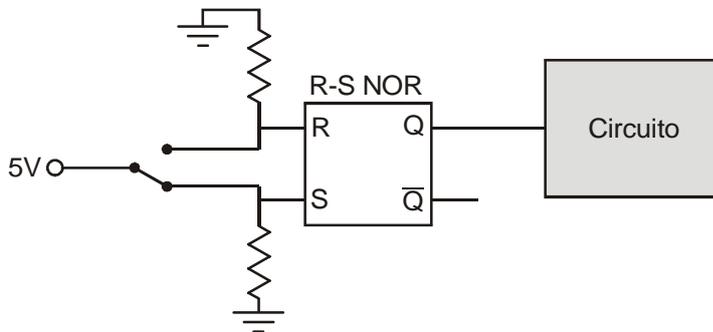
Hay ocasiones en que la entrada de un circuito es un conmutador que manualmente se coloca en la posición **0** o **1**, por ejemplo como el siguiente circuito:



Puede ocurrir que al pasar el conmutador desde la posición **0** a la posición **1**, se produzcan rebotes mecánicos en el conmutador. Estos rebotes pueden hacer que el circuito en cuestión interprete que el conmutador se accionó varias veces, cuando en realidad se accionó solo una vez. Por lo tanto el efecto de estos rebotes no es deseado y se lo debe anular de alguna manera. Una forma de eliminar los rebotes es mediante un latch R-S NAND:

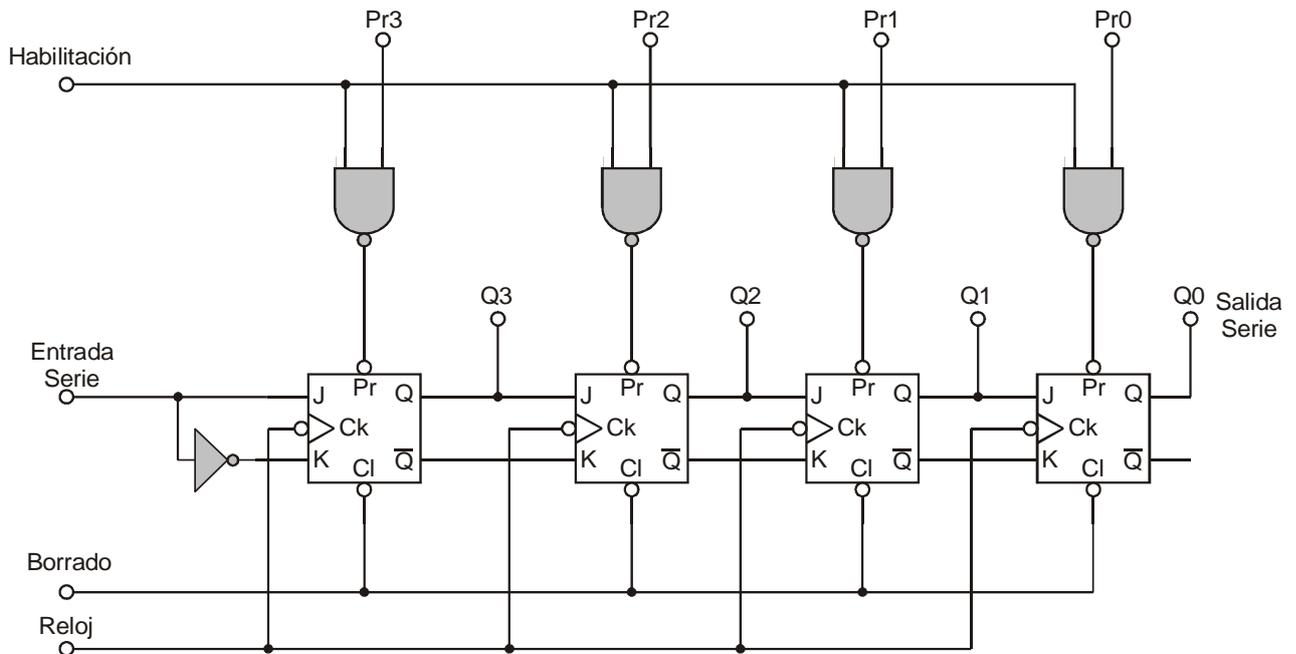


Cuando el conmutador esta en la posición inferior, $R = 1$ y $S = 0$ con lo cual $Q = 0$. Con el conmutador en la posición superior $R = 0$ y $S = 1$, con lo que resulta $Q = 1$. En el caso en que se produzcan rebotes en el conmutador R y S quedan en 1, que en un R-S NAND es la combinación que no modifica la salida Q . Si en lugar de utilizar un R-S NAND se utiliza un NOR, como en este la condición de no cambio en salida es $R = S = 0$, las resistencias deben conectarse a masa y el conmutador a V_{cc} :



4.2. Registro de desplazamiento

Un registro de desplazamiento es una combinación de N flip flops que permiten hacer transformaciones serie-paralelo y paralelo-serie de N bits. Un circuito típico es el siguiente:



Este circuito permite memorizar y operar un dato de en este caso 4 bits. El registro de desplazamiento puede operar este dato de las siguientes formas:

- Entrada de información serie a una determinada frecuencia, y salida serie a una frecuencia distinta.
- Entrada de información serie, memorización, y posterior salida en paralelo.
- Entrada de información en paralelo, memorización, y posterior salida en serie.



Todas las anteriores funciones son esenciales por ejemplo en las USART, que son dispositivos que permiten una comunicación serie (RS232, RS485, etc.) entre dos dispositivos como ser computadoras, modems, PLC, etc. Veamos las distintas funciones del registro de desplazamiento:

Convertor serie a paralelo

Antes de comenzar a registrar (guardar) la información de entrada (información seriada), es conveniente que borremos el contenido del registro. Para esto colocamos un 0 en la entrada **borrado** y en la de **habilitación**, con lo que las salidas Q3 Q2 Q1 Q0 pasan a 0. Hecho esto, dejamos **habilitación** en 0 y **borrado** en 1.

A continuación se aplica en la entrada serie del registro de desplazamiento el tren de pulsos seriales que conforman los datos de entrada. Sincrónicamente con cada dato de entrada debe aplicarse un pulso en la señal de **reloj** del registro.

Primero se ingresa el bit menos significativo de los datos. Al aplicarse el primer pulso de reloj, este bit se almacenará en la salida Q del primer flip flop (configurado como flip flop D). A continuación se aplica a la entrada serie del registro de desplazamiento el segundo bit de datos, y se aplica otro pulso de reloj. Con este segundo pulso se logra que el primer bit almacenado en Q3 ahora se desplace a la salida Q2 del segundo flip flop, y también que el segundo bit de datos se almacene en Q3. Y así sucesivamente se ingresan los datos en la entrada serie y se aplican pulsos en la entrada de reloj, hasta que se tengan almacenados los 4 bits de datos en Q3 Q2 Q1 Q0. En esta etapa, disponemos en paralelo los 4 bits que ingresaron en forma seriada.

Convertor serie a serie

Partiendo de la situación anterior en la que ya tenemos memorizados los 4 bits de datos que entraron serialmente, si ahora seguimos aplicando pulsos en la entrada **reloj** del registro, estos mismos datos irán saliendo serialmente en la salida Q0 del registro, pero a la frecuencia de la señal de reloj aplicada. Esto es útil cuando deben comunicarse dos dispositivos que “hablan” serialmente, pero que lo hacen a distintas velocidades de transmisión. En este caso el registro de desplazamiento se utiliza como un convertor de velocidades.

Convertor paralelo a serie

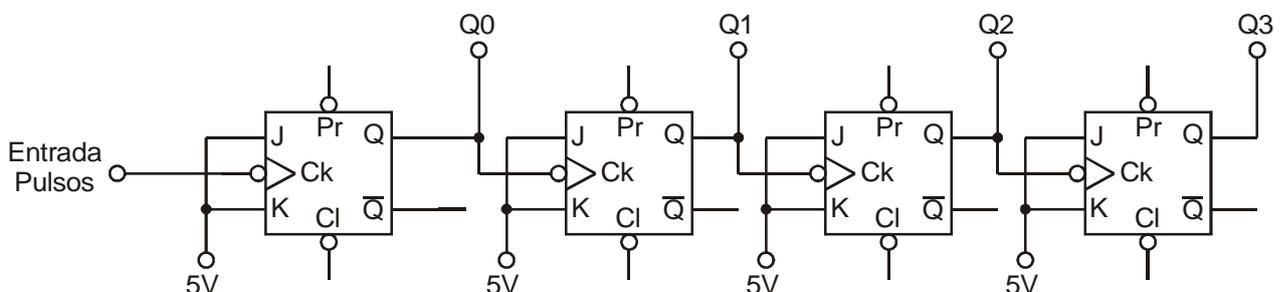
En el caso en que dispongamos de los datos en forma paralela y los queramos sacar en forma serie, el procedimiento es el siguiente: primero cargamos los bits en formato paralelo en los 4 flip fops que conforman el registro de desplazamiento. Para ello primero borramos las salidas Q3 Q2 Q1 Q0 como se explico anteriormente. Luego aplicamos los bits de datos en las entradas Pr3 Pr2 Pr1 Pr0 del registro, y aplicamos un 1 en la entrada de **habilitación**, con lo que se memorizará en el registro el dato de 4 bits. Posteriormente dejamos a **habilitación** en 0, y comenzamos a aplicar pulsos en la entrada **reloj**. Con cada pulso de reloj que se aplique, se irá obteniendo en forma seriada en Q0 el dato que originalmente estaba en paralelo.

4.3. Circuitos contadores

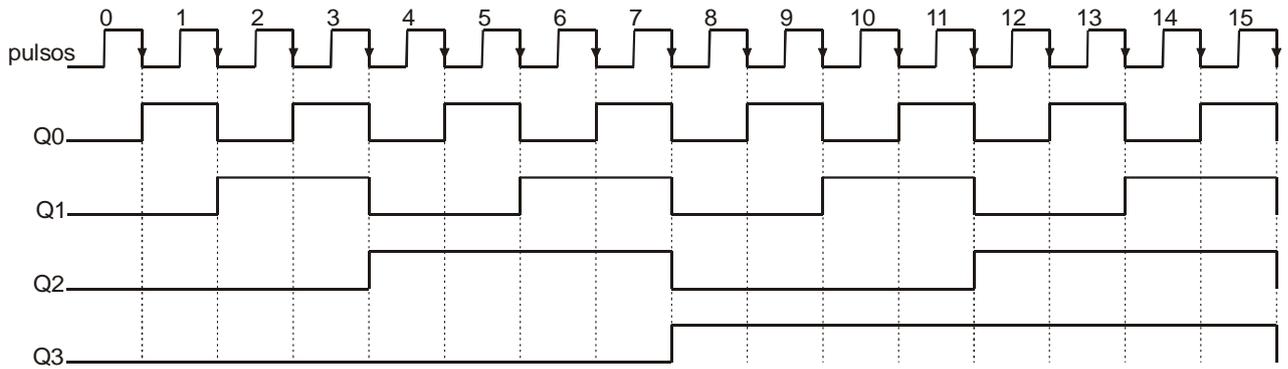
Son circuitos realizados con flip fops conectados en cascada, y cuya finalidad es contar impulsos o eventos, medir tiempos, medir frecuencias, etc. Se los clasifica en dos formas: asincrónicos y sincrónicos.

4.4. Contador binario asincrónico

En este caso tenemos un contador con 4 flip fops, es decir con una cuenta de 4 bits. Con 4 bits se pueden hacer $2^4 = 16$ conteos, cuyo código estará disponible en las salidas Q3 Q2 Q1 Q0.

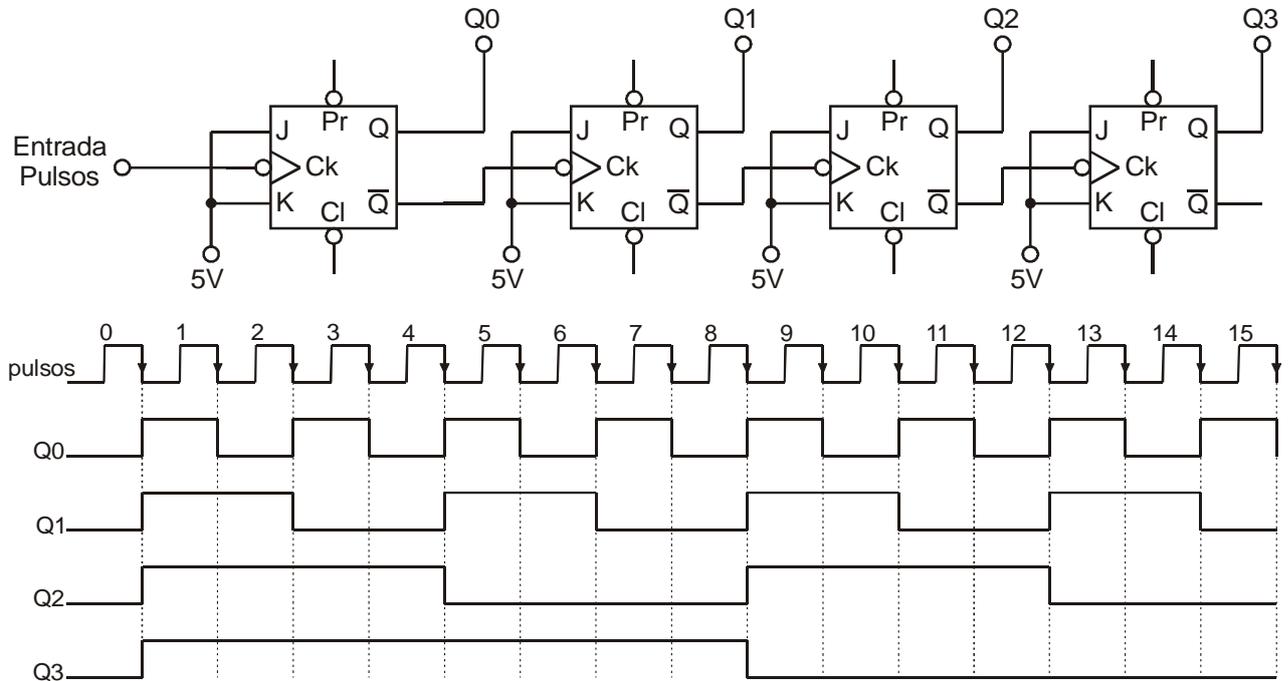


Los pulsos a ser contados se introducen en la entrada de reloj del flip flop de mas a la izquierda. En todos los flip fops las entradas J-K se conectan a 1, lo que hace que los flip fops se comporten como tipo T (la salida Q cambia por cada pulso que se le aplique a la entrada de reloj Ck, en este caso en el flanco de bajada).



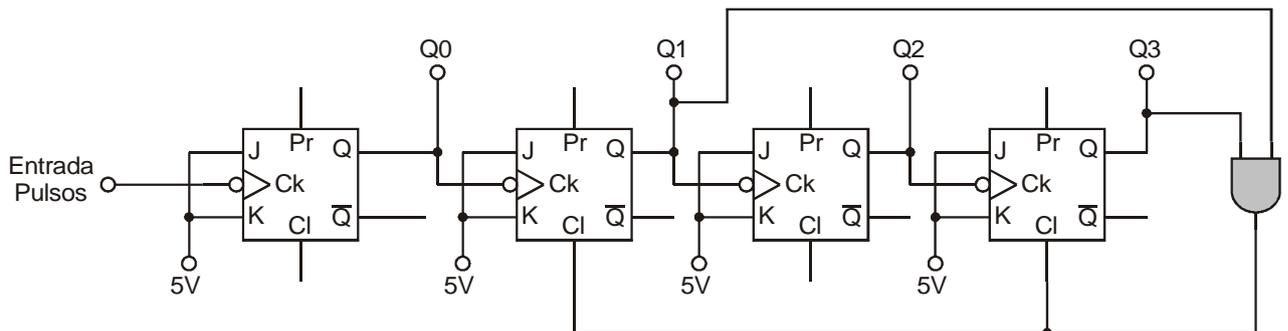
4.5. Contador descendente asincrónico

Este contador cuenta en forma descendente. Cuando llega el primer pulso, debido a que la conexión entre flip flops se hace a través de las salidas negadas \bar{Q} hacia las entradas de reloj, todas las salidas pasan a valor 1. En pulsos sucesivos de reloj, las transiciones de los flip flops hacen que comiencen a contar en forma descendente.



4.6. Contador/divisor por N

Los contadores anteriores cuentan en base 2, es decir que si el contador posee 4 flip flops cuentan hasta $2^4 = 16$. Si deseamos contar en una base distinta, digamos por ejemplo en base 10, necesitamos un contador que cuando llegue al décimo pulso de conteo pase a 0. Esto se puede lograr haciendo una pequeña modificación en el contador binario ascendente:





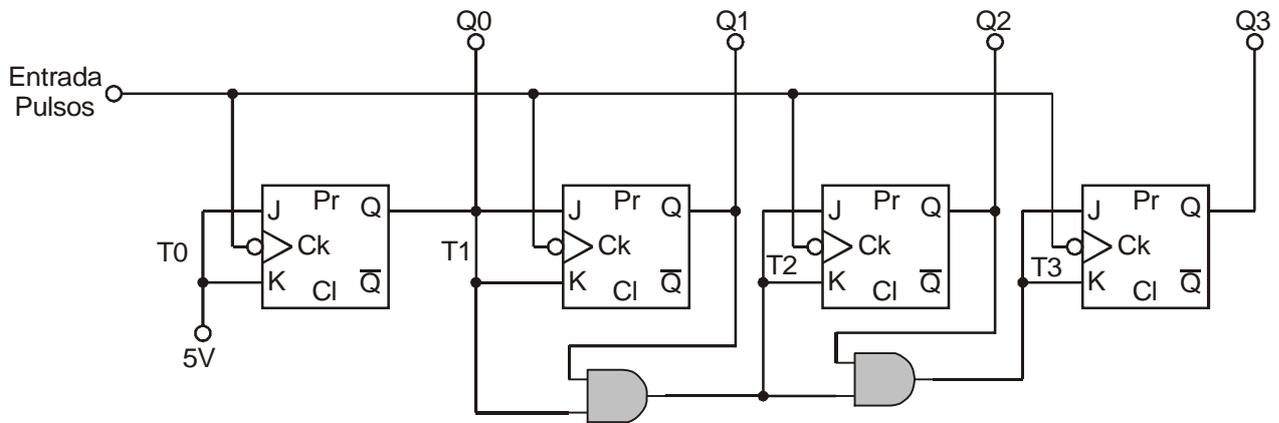
Si queremos contar en base 10, es decir que el contador vuelva a 0 en el décimo pulso, agregamos una compuerta AND cuyas entradas sean Q3 y Q1. Cuando se llegue al décimo pulso, estas dos salidas estarán en 1 por lo que la salida de AND también será un 1 y entonces se resetearán los flip flops, pasándose a la cuenta cero.

4.7. Contadores sincrónicos

La velocidad de conteo de los contadores asincrónicos está limitada por el tiempo de propagación, que es el tiempo requerido para que el contador complete su respuesta a un pulso de entrada. El mayor tiempo de propagación se va a dar cuando todos los flip flops cambien sus salidas (por ejemplo de 1111 a 0000). En algunas aplicaciones, este tiempo puede llegar a ser mayor que el período de los pulsos de entrada a ser contados, lo cual puede hacer que el conteo sea erróneo. Esto limita la frecuencia máxima de conteo de los contadores asincrónicos.

Para aumentar esta frecuencia de conteo, se utilizan los contadores sincrónicos o en paralelo, en los que todos los flip flops se disparan simultáneamente (en paralelo) mediante los pulsos de entrada a ser contados. Como los pulsos llegan a todos los flip flops, se utiliza un circuito lógico para determinar cual de los flip flops debe cambiar según vaya avanzando la cuenta.

Veamos el circuito de un contador sincrónico de 4 bits:

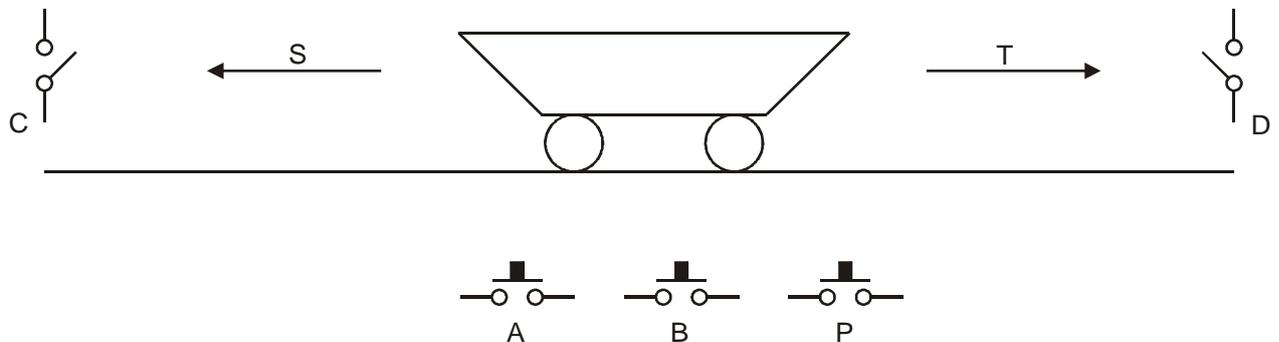


Como vemos el contador está compuesto por cuatro flip flops con lógica de control T ($J = K$). Si $T = 0$ el flip flop no cambia su salida cuando llega el pulso de reloj (pulso a ser contado). Cuando $T = 1$, la salida Q del flip flop se complementa cada vez que se presente un pulso de reloj. Notemos que:

- Q0 (bit menos significativo de la cuenta) cambia con cada pulso de reloj (ya que $T_0 = 1$).
- Q1 cambia con cada pulso de reloj, pero sólo cuando Q0 es 1.
- Q2 cambia con cada pulso de reloj, sólo si $Q_0 = Q_1 = 1$.
- Q3 cambia con cada pulso de reloj, sólo si $Q_0 = Q_1 = Q_2 = 1$.

5. EJERCICIO EJEMPLO

Realizar un automatismo para controlar el movimiento de una vagoneta mediante dos accionamientos S y T, tres pulsadores A, B y P, y dos finales de carrera C y D, según el siguiente esquema:





Cuando se pulsa **A**, se debe activar el accionamiento **S** para mover la vagoneta hacia la izquierda, siempre que no esté activada la salida **T**, el pulsador de paro **P** o el final de carrera **C**. Por otra parte, la salida **S** debe desactivarse si se acciona el final de carrera **C** o el pulsador de paro **P**.

Cuando se pulsa **B**, se debe activar el accionamiento **T** para mover la vagoneta hacia la derecha, siempre que no esté activada la salida **S**, el pulsador de paro **P** o el final de carrera **D**. Por otra parte, la salida **T** debe desactivarse si se acciona el final de carrera **D** o el pulsador de paro **P**.