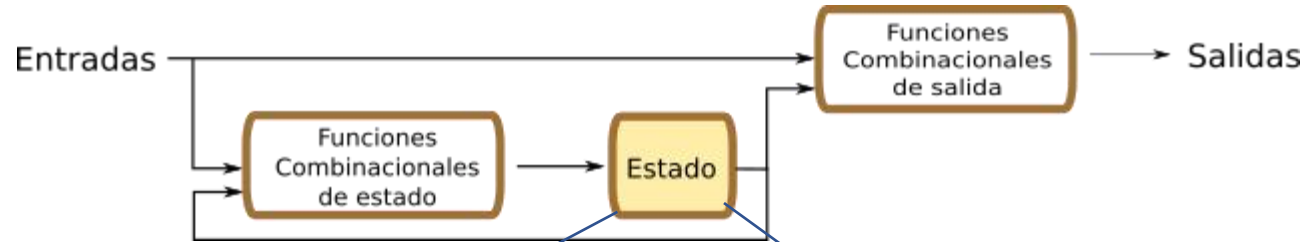


Tema 6: Lógica Secuencial

Parte 2.

Prof: Juan J. Pombo

Circuito secuencial frente a combinacional. Recapitulando.



El estado de un circuito se puede definir como una “captura” de todos los elementos de memoria asociados al circuito en un momento dado.

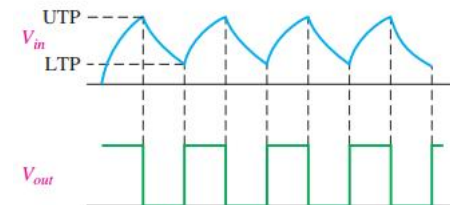
Podemos “memorizar” datos mediante el elemento *biestable*.

Flip-flops y *latches* son 2 tipos particulares de biestables

Generación del reloj

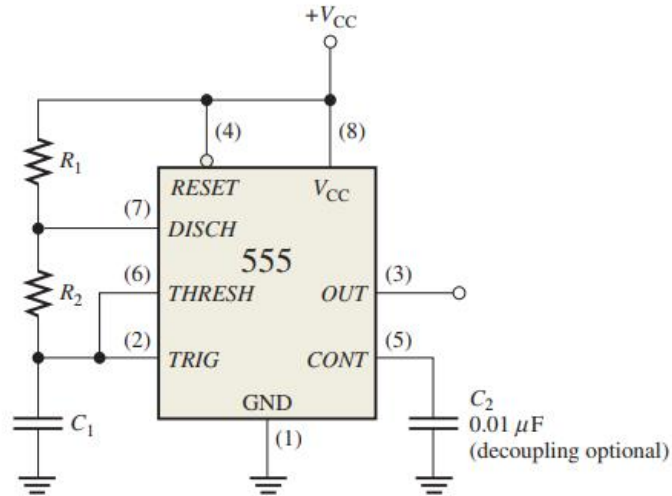
El reloj es una parte fundamental del circuito secuencial.

- Marca los puntos donde los *latches* toman los valores de entrada para generar el nuevo estado.
- Es un factor tecnológico que nos define el rendimiento de los circuitos, pero que estará supeditado a las características del diseño: está limitado en frecuencia por el tiempo máximo de retardo de la cascada de puertas más “larga”/”lenta” del circuito.
- Para generar el reloj se pueden utilizar los circuitos multivibradores, del cual uno de los más conocidos y clásicos es el “**multivibrador astable**” (astable : sin estado estable) basado en el integrado 555.
- Hoy en día, sin embargo, el uso del 555, aún siendo frecuente en el mundo *hobby*, ha ido dejando lugar a circuitos específicos
- Los circuitos multivibradores astables tienen un principio de funcionamiento basado en generar estados internos que van cambiando cíclicamente, donde la temporización casi siempre vendrá marcada por tiempos de carga de elementos condensadores y/o resonancia en cristales de cuarzo, lo que los puede dotar de gran precisión.
- A menudo tendrán varias frecuencias de trabajo seleccionables digitalmente.
- En los circuitos prácticos además aparecerán combinados con divisores de frecuencia para conseguir operaciones multiciclo.

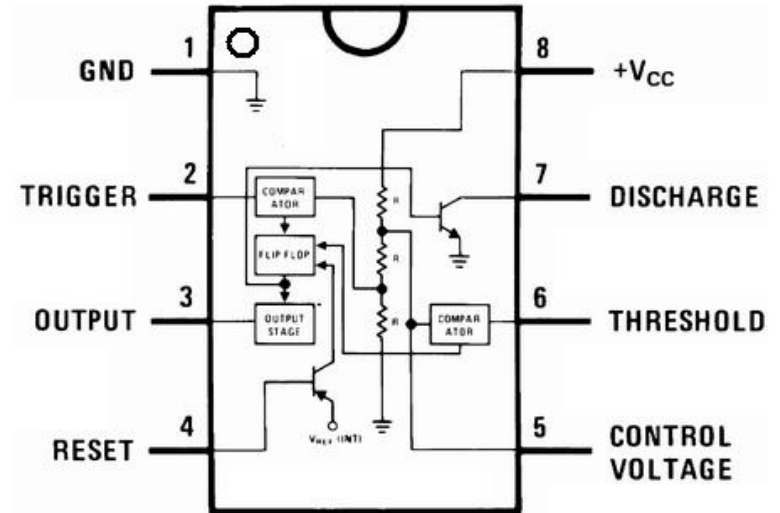


Generación del reloj

- Ejemplo de generación de un reloj con un 555 configurado en modo astable.



$$f = \frac{1.44}{(R_1 + 2R_2)C_1}$$



Tri-state buffers

Otro dispositivo asociado habitualmente a circuitos con flip-flops.

Se trata de un dispositivo electrónico que permite poner una salida en uno de estos 3 estados:

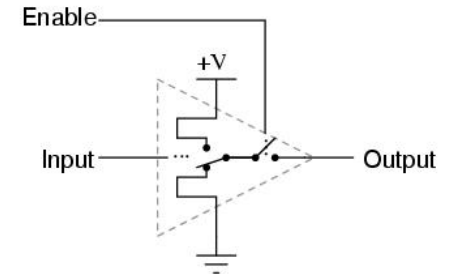
- ALTA (H), BAJA (L) y alta impedancia (HIGH-Z).

El estado Hi-Z es equivalente a un circuito abierto. En este estado lo que venga después queda “desconectado” del circuito previo.

- Utilidad:
 - Imaginemos circuitos con subpartes que ponen datos que comparten una misma línea de cableado (un camino hacia una memoria, por ejemplo). Las distintas lógicas podrían estar pidiendo estados H y estados L. Esto generaría cortocircuitos! Y posibles daños a los componentes. Necesitamos poder conectar unas partes u otras del circuito según una lógica de selección, relojes, etc.
 - El problema descrito en el apartado anterior se denomina técnicamente **contención de bus**.

Una configuración típica será en combinación con decodificadores, que sirven como elemento de selección actuando sobre los “enables” para conectar un camino de datos u otro al bus.

- Ejemplo: 74LS241 (8 outs)

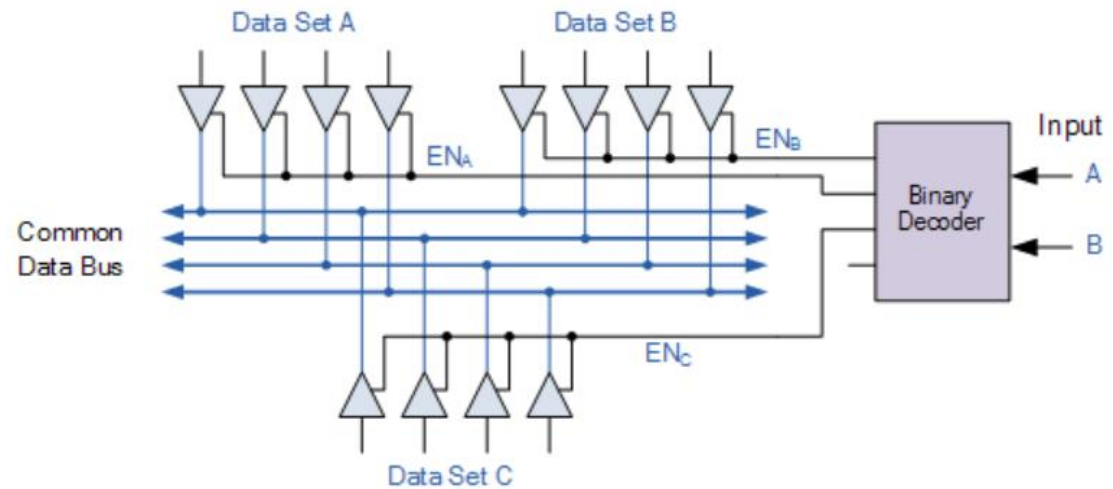


Triestado activo en alta

Enable	IN	OUT
0	0	Hi-Z
0	1	Hi-Z
1	0	0
1	1	1

Tri-state buffers (II)

Ejemplo de uso de un decodificador para seleccionar la habilitación entre 3 conjuntos de datos.



Aplicaciones de los Flip-Flops. Contadores

- Formalmente es un circuito secuencial que va pasando por una serie de estados
- El más común es el contador binario de n -bit. Usa n Flip-flops y 2^n estados.
 - Normalmente al alcanzar el estado $2^n - 1$, pasará el estado 0 de nuevo.
- A su vez las utilidades de los contadores son varias:
 - Simplemente “contar”
 - Generar retardos
 - Generar secuencias de control para la lógica del circuito
 - Divisor de otras señales periódicas para generar nuevos relojes de otra frecuencia
 - Etc.. .
- Tipos de contadores:
 - Ripple Counters
 - Contadores síncronos. ← Nos centramos en este tipo, el más habitual.

Contador síncrono con Flip-Flops Tipo D

- 1) Repasamos la tabla característica de un Flip-Flop tipo D (aquí Q representa el estado actual y Q' el estado siguiente):

En realidad para un flip-flop tipo D, Q' no depende de Q, solo de D (la señal de entrada)

Q	D	Q'
0	0	0
0	1	1
1	0	0
1	1	1

D	Q'
0	0
1	1

- 2) Tabla de transición de estados

Se trata de una forma distinta de ver la misma información pero que resulta más cómoda para el diseño de los circuitos secuenciales.

Estado Actual	Estado Siguiente	Entradas FF (flip-flop)
0	0	0
0	1	1
1	0	0
1	1	1

* Notar que aquí Q' represente el estado SIGUIENTE (no confundir con el estado negado)

Ejemplo: Contador síncrono 0 a 7

$Q_2Q_1Q_0$	$Q'_2Q'_1Q'_0$	$D_2D_1D_0$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 0
0 1 0	0 1 1	0 1 1
0 1 1	1 0 0	1 0 0
1 0 0	1 0 1	1 0 1
1 0 1	1 1 0	1 1 0
1 1 0	1 1 1	1 1 1
1 1 1	0 0 0	0 0 0

Vamos a desarrollarlo paso a paso para comprender cómo se aborda el diseño de un circuito secuencial simple:

Lo prepararemos con 3 FF (*flip flops*) tipo D

A la izquierda se muestra la tabla de transición de estados, preparada de la manera que se indicaba anteriormente:

- Para cada estado actual -> ponemos al lado columnas para el estado siguiente
- Después escribimos las columnas con las entradas que requerirían los FF

En un FF tipo D, el estado siguiente, Q' , es especialmente simple: es exactamente lo que se pone a la entrada: $Q'=D$

A continuación escribiremos las funciones combinaciones para cada uno de los D_i , como función de los estados actuales (siguiente diapositiva)

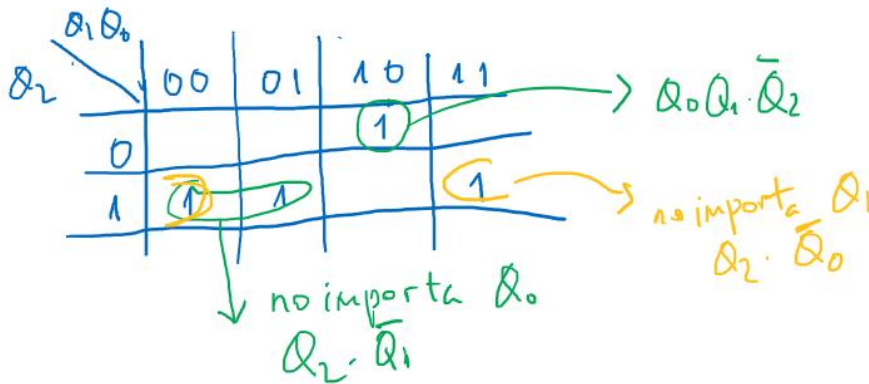
Ejemplo: Contador síncrono 0 a 7 (2)

$$D_0 = \bar{Q}_0$$

$D_1 = Q_1 \text{ EXOR } Q_0$. Como se ve claramente en la tabla.

D_2 es un poco más elaborado de obtener. Requerirá usar un mapa de Karnaugh:

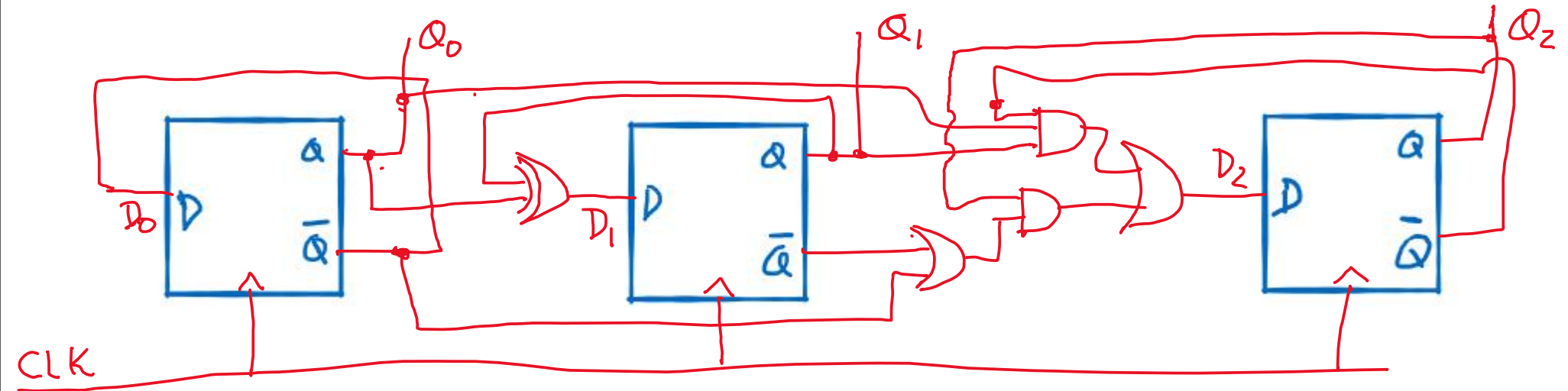
$Q_2 Q_1 Q_0$	$Q_2' Q_1' Q_0'$	$D_2 D_1 D_0$
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 0
0 1 0	0 1 1	0 1 1
0 1 1	1 0 0	1 0 0
1 0 0	1 0 1	1 0 1
1 0 1	1 1 0	1 1 0
1 1 0	1 1 1	1 1 1
1 1 1	0 0 0	0 0 0



$$D_2 = \bar{Q}_0 Q_2 + \bar{Q}_1 Q_2 + Q_0 Q_1 \bar{Q}_2$$

$$D_2 = Q_2 (\bar{Q}_0 + \bar{Q}_1) + Q_0 Q_1 \bar{Q}_2$$

Ejemplo: Contador síncrono 0 a 7 (3)



$$\begin{aligned} D_0 &= \overline{Q_0} \\ D_1 &= Q_1 \text{ EXOR } Q_0 \\ D_2 &= Q_2(\overline{Q_0} + \overline{Q_1}) + Q_0 Q_1 \overline{Q_2} \end{aligned}$$

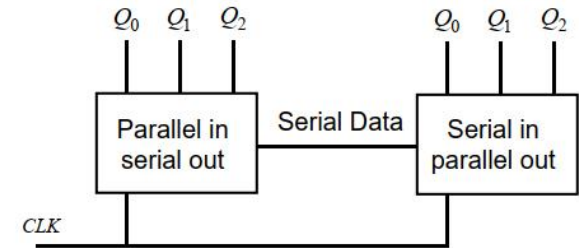
Aplicaciones de los Flip-Flops. Memorias

Registros de desplazamiento (*Shift Register*) -> *Aplicación muy frecuente.*

No es exactamente lo mismo que los desplazadores vistos en circuitos combinacionales. Aquí los valores pasan por elementos de memoria y quedan actualizados en el circuito. Antes era un simple resultado de una operación combinacional. Ahora es un estado secuencial **que se actualiza**. El circuito **CAMBIA DE ESTADO**.

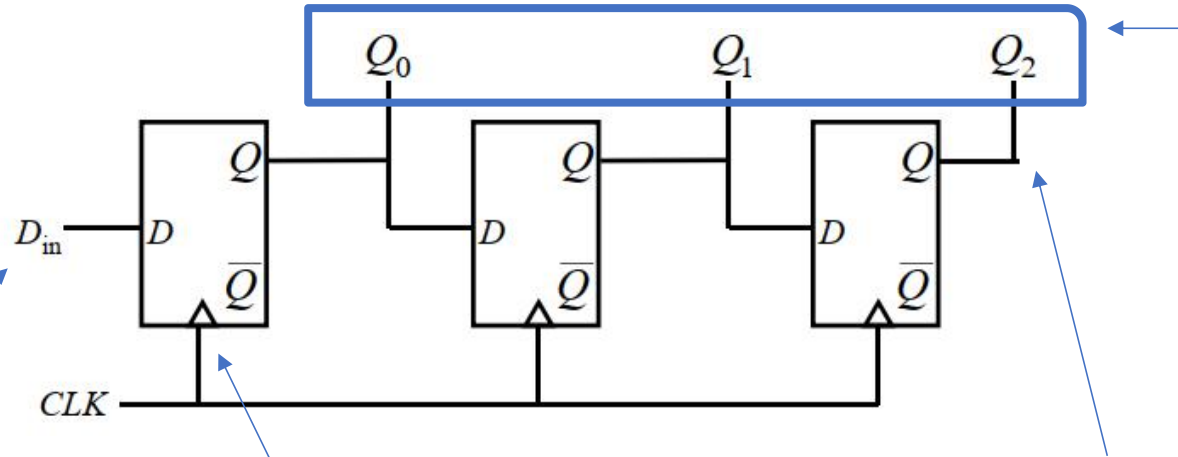
Usos:

- Usados frecuentemente en conversión serie-paralelo / paralelo-serie en sistemas de comunicaciones.



Registros de desplazamiento (*Shift Registers*)

Ejemplo de 3 bits con una cadena de 3 FF tipo D: (paso una cadena serial a una disposición del dato paralela)



3. Pero la salida está disponible en paralelo

1. Los datos se introducen por una entrada "serie"

2. El reloj va marcando la carga y el desplazamiento

4. Si quisiésemos cargar los FF en paralelo (como hacíamos en el caso combinacional) tendríamos que usar las señales de PRESET y CLEAR del FF

5. Si quisiésemos descargar el registro en "serie" tendríamos que ir obteniendo los datos todos a través de Q_2 en cada flanco de CLK

6. Este circuito así planteado puede ser una base para generar un enlace de datos serie-paralelo.