

## Lógica Combinacional y Secuencial

Laboratorio de Electrónica 2019

### Práctica 6

La electrónica digital, o los circuitos digitales, son componentes electrónicos que operan con señales digitales, es decir señales que toman valores discretos. En la mayoría de los casos esta señal puede tomar dos valores; a esto se llama señal binaria o lógica. Estas dos señales corresponden a los valores “0” y “1” (o “verdadero” y “falso”) del dominio de la lógica Booleana. Las técnicas digitales son prácticas porque es mucho más fácil producir un componente electrónico que pueda conmutar entre estados discretos que reproducir con exactitud un rango continuo de valores. En la práctica, estos dos estados se generan a partir de voltajes. El rango de voltajes que representa cada estado depende de la familia lógica de circuitos utilizados.

Los circuitos de electrónica digital están contruidos a partir de arreglos de gran cantidad de compuertas lógicas. La mayoría de los circuitos digitales se dividen en dos clases: combinacionales y secuenciales. Los circuitos de lógica combinacional son un tipo de circuitos digitales que implementan operaciones de lógica Booleana, en los que la salida es función exclusivamente del valor de la(s) entrada(s), “1”, “high” o “0”, “low” en ese mismo momento. Como resultado, los circuitos combinacionales no tiene feedback, y cualquier cambio en las señales de entrada tendrá un efecto inmediato en la salida. Dependen en todo momento de **combinaciones** de sus entradas. Son circuitos sin memoria. Esos circuitos entran también en la categoría de “asincrónicos”, debido a que no precisan un reloj ni interno ni externo para operar.

Por el contrario, en los circuitos de lógica secuencial las salidas dependen no sólo del valor actual de las entradas sino también de la historia de la entrada. Casi la totalidad de los circuitos de sistemas digitales prácticos usan una mezcla de lógica combinacional y secuencial (el oscilador 555 que vimos hace unas semanas es un ejemplo). La lógica secuencial se puede usar para construir máquinas de estado, que son componentes fundamentales de la electrónica digital. Si bien se pueden hacer componentes de lógica secuencial asincrónicos, la mayoría son “sincrónicos”, es decir que los cambios de estado del circuito sólo se llevan a cabo en tiempos discretos dados por este reloj.

#### Compuertas combinacionales:

En general los componentes se esquematizan mediante símbolos y a partir de su tabla de operación (*truth table*). En la figura 1 se esquematiza una compuerta positiva AND de 3 entradas. Los símbolos estándar de las compuertas Booleanas combinacionales se pueden ver en [este link](#).

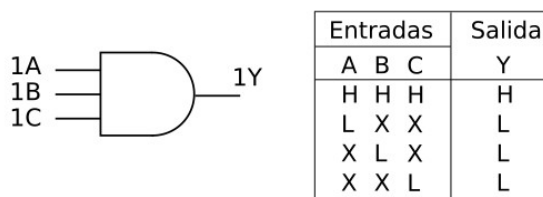


Figura 1: Símbolo lógico y tabla de operación de una compuerta AND de tres entradas

Las distintas familias lógicas de circuitos integrados digitales se construyen usando un diseño particular, con niveles lógicos y alimentación compatibles entre sí. La tabla I ejemplifica niveles lógicos binarios de tres familias lógicas; TTL (*transistor-transistor logic*), CMOS (*Complementary metal-oxide-semiconductor logic*) y ECL (*emitter-coupled logic*)

Tabla I: características básicas de algunas familias lógicas de circuitos integrados

Tecnología	Voltaje "0"	Voltaje "1"	Alimentación	Velocidad
TTL	0 V a 0.8 V	2 V a $V_{CC}$	$V_{CC} = 5 V \pm 10\%$	10-100 MHz
CMOS	0 V a $0.333 V_{DD}$	$0.667 V_{DD}$ a $V_{DD}$	$V_{DD}$ hasta 16V	50-125 MHz
ECL	$V_{EE}$ a -1.75 V	-0.9 V a 0 V	$V_{EE} = -5.2 V$	500 MHz

Una de las familias lógicas comerciales más comunes y utilizadas es la serie 7400. Presentada en 1964 por Texas Instruments, los primeros componentes eran chips lógicos TTL en cápsulas cerámicas y plásticas. Posteriormente evolucionaron para incluir tecnología CMOS de bajo consumo, voltajes de alimentación más bajos, y encapsulados de montaje superficial. En [este link](#) hay una lista de los componentes de la serie. El esquema de la figura 1 muestra uno de las tres compuertas triples de un circuito integrado 74XX11.

### Ejercicio 1:

El integrado 74XX86 agrupa cuatro compuertas XOR, mientras que los 74XX00 conforman cuatro compuertas NAND. Encontrar la tabla de funcionamiento de cada uno de los arreglos de compuertas de la figura 2.

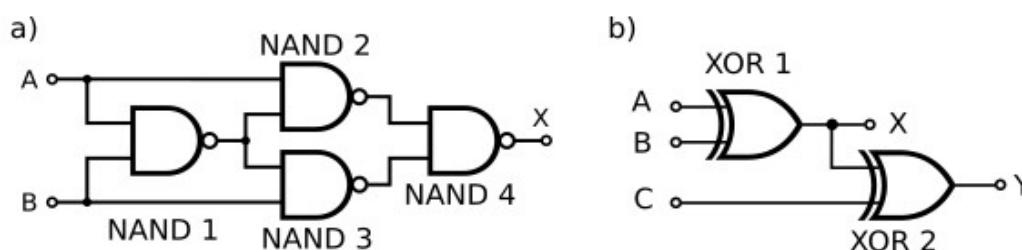


Figura 2: Qué hacen estos arreglos de compuertas?

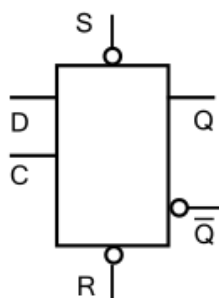
Qué compuerta implementa el arreglo a)?

### Flip-Flop. El secuencial más simple:

Un flip-flop (o latch) es un circuito que tiene dos estados estables (al contrario de un monoestable), y puede ser usado para almacenar información de un estado digital. El circuito puede producir cambios de estado a partir de señales aplicadas a una o más entradas de control, y tiene una o dos salidas. Ya vimos un tipo de flip-flop cuando estudiamos el oscilador 555. Estos componentes son los constituyentes fundamentales de la electrónica digital, ya que se pueden usar como memorias. Un flip-flop puede guardar un bit de datos; uno de sus estados representa un "1" y el otro un "0". La salida de los mismos depende no sólo del estado actual de sus entradas, sino que también de estados anteriores. Es el ejemplo más simple de componentes de lógica secuencial. También se pueden usar para contar eventos, y para sincronizar señales de entrada de tiempos de llegada variables con un reloj de referencia. Los flip-flops se pueden disparar en forma asincrónica, por nivel de señal, o en forma sincrónica, con reloj, a partir de eventos disparados por flancos (ascendentes o descendentes).

Hay una gran variedad de circuitos flip-flop. Para no extendernos demasiado vamos a concentrarnos en un tipo de ellos, el D-flip flop (D-FF, la D es por "data" o "delay"): el D-FF captura el valor de la entrada D (datos) sólo al momento de detectar un flanco positivo en la entrada C (clock). En otros momentos las salidas no cambian.

Además, se puede forzar el estado de salida con entradas más directas, llamadas (S) Set y (R o  $C_D$ , Reset o Clear). La tabla de posibles estados y acciones y el símbolo lógico se muestran en la Fig. 3.



Inputs				Outputs	
S	R	C	D	Q	$\bar{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H	H
H	H	↗	H	H	L
H	H	↘	L	L	H
H	H	L	X	Q <sub>0</sub>	$\bar{Q}_0$

Figura 3: Símbolo lógico y tabla de operaciones de un flip flop D del integrado 74XX74

El circuito de la figura 4 muestra un detector de coincidencias armado a partir de flip-flops D de la serie 74XX74. Vamos a analizar el comportamiento de cada parte. El sistema debería emitir un pulso en la salida “COINC OUT” toda vez que haya dos pulsos simultáneos (en coincidencia) en las entradas “A IN” y “B IN”. Las otras dos salidas, “A OUT” y “B OUT” generan copias de los pulsos de entrada. Comenzando por el FF de más arriba, tiene  $D=low$ ,  $R=high$ , y al encenderse el componente  $S=low$ , por lo que Q pasa a “high” y un tiempo  $RC$  más tarde S pasa a “high”. En esta condición, el componente queda esperando eventos en C, con “A OUT”= $low$ . Cuando llega un pulso a C, el componente reacciona a la transición (flanco de subida), pone la salida “A OUT”= $high$ , y Q pasa a “low”. Esto descarga el capacitor y en un tiempo  $RC$ , S pasa a  $low$ , con lo que “A OUT” pasa de nuevo a  $low$ , y Q a  $high$ . El sistema emitió un pulso de duración  $RC$  en respuesta a un pulso a la entrada, por lo que funciona como monoestable.

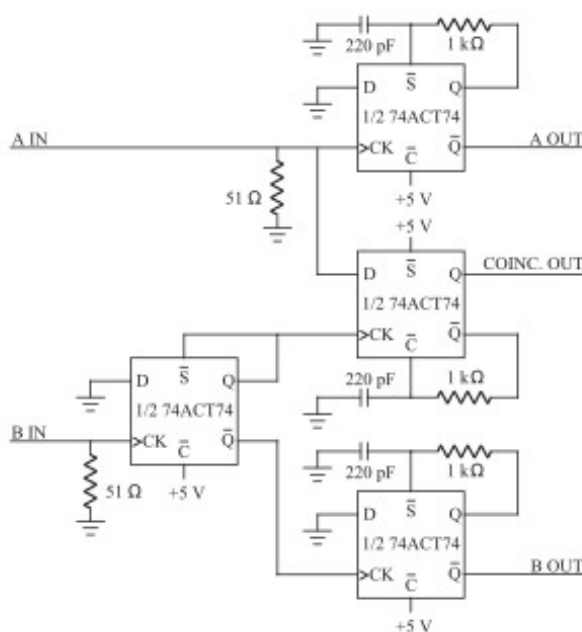


Figura 4: Detector de coincidencias usando Flip-Flops D [Dehlinger & Mitchell, Am J. Phys, 70, 898 (2002)].

Veamos qué ocurre con eventos en B: El primer D-FF hace exactamente lo mismo que el anterior, pero como Q y S están conectados directamente, la conmutación es “instantánea”, y el pulso a la salida está limitado por los tiempos de tránsito y capacidades parásitas del integrado (es el pulso más corto posible que puede emitir el componente, alrededor de 10 ns). El pulso en Q es negado (Q está todo el tiempo en “high”, salvo en el pulso que pasa a “low”, por lo que el flanco ascendente se obtiene cuando el pulso termina). En el tercer flip flop, el retardo está armado sobre la pata R. El estado a  $t=0$  es  $S=high$ ,  $R=low$ , por lo que la salida “COINC OUT” queda en “low”,  $\bar{Q}$  en “high” y al cargarse el capacitor llega al estado de espera con  $S=R=high$ . En esta condición, si mientras

recibe el flanco ascendente del componente anterior hay un estado “high” en D (un pulso sobre la entrada A), la salida “COINC OUT” cambia a “high”, durante un tiempo  $RC$ . Es decir, se emite un pulso siempre y cuando haya un flanco ascendente en C y un estado alto en D, es decir dos pulsos simultáneos en A y en B.

### Experimento 1:

Armar el circuito de la figura 4 y estudiar qué ocurre con las salidas. Verificar que la duración de los pulsos se puede controlar con las resistencias y capacitores. Usar un generador de funciones para medir el ancho de la ventana de coincidencias (cuán “separados” pueden estar los pulsos para que siga midiendo eventos en coincidencias).

### Ejercicio 2:

Estudiar qué comportamiento se espera en las salidas  $V_{out1}$  y  $V_{out2}$  en el circuito de la figura 5, si a la entrada se conecta una señal cuadrada de frecuencia  $f$ .

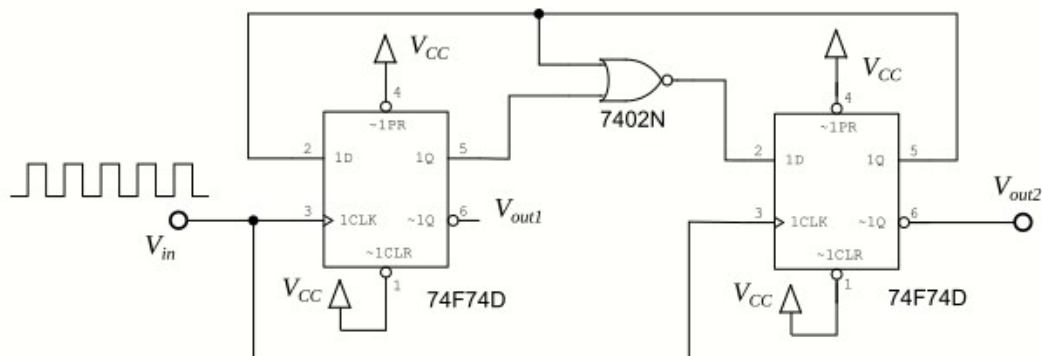


Figura 5: Estudiar la función del circuito

### Experimento 2:

Armar el circuito de la figura 5 y verificar el funcionamiento. Si se agregan más etapas qué ocurre?

#### Shift registers:

Un shift register es una serie de flipflops conectados en cascada que comparten un mismo reloj, en la cual la salida de cada flipflop está conectada a la entrada D (data) del siguiente en la cadena, logrando un circuito que a cada pulso de reloj desplaza en una posición la lista de bits almacenados, ingresando el bit del estado a la entrada y emitiendo el último bit de la lista a la salida (también se puede, armando un arreglo de varias de estas cadenas en paralelo, construir un shift register que puede manejar una matriz de bits). Los shift registers pueden tener salidas y entradas en serie o paralelo. Generalmente son *serial-in/parallel-out* (SIPO) o *parallel-in, serial-out* (PISO).

Uno de los usos de los shift registers es como contadores, incrementando un registro en cada ciclo. Sirven para contar ocurrencias de ciertos eventos, por ejemplo la cantidad de instrucciones de un proceso, llevar registro del tiempo entre eventos, y generar señales de control y temporización, por ejemplo para producir señales cuyas frecuencias son múltiplos de la frecuencia de reloj original.

### Ejercicio 3:

Estudiar el funcionamiento del arreglo de la figura 6. Completar la secuencias temporales de la figura. Para qué sirve? Qué valores de señal debe haber en S y R para que funcione?

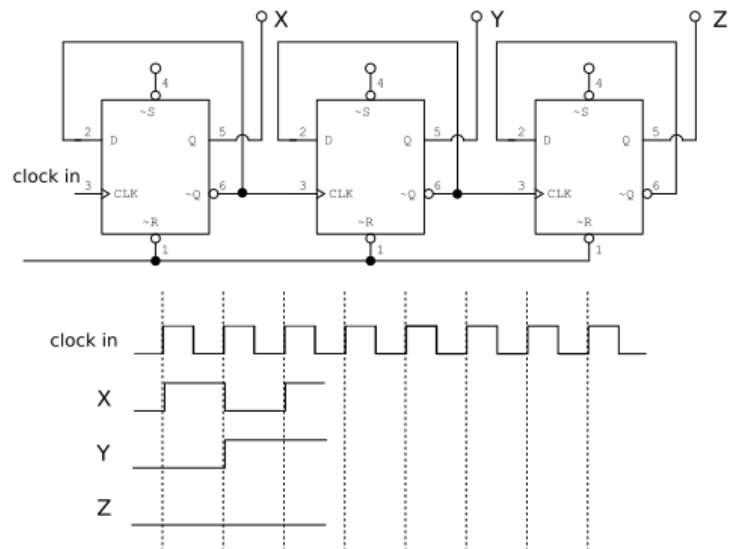


Figura 6: una aplicación de un shift register

### Experimento 3:

Implementar el circuito de la figura 6. Agregar un LED y una resistencia de  $2.2k\Omega$  en cada salida para limitar la corriente y comprobar el funcionamiento a partir de la secuencia de salida. En qué estado deben estar las patas S (set) y las R (reset o clear)? Qué pasa si se cambia el estado de todas las R simultáneamente?