

Taller de
Diseño de Circuitos Digitales y Programas de Computadoras

Parte I: Circuitos Electrónicos y Algoritmos Básicos

Módulo 1: Introducción a los circuitos lógicos

Versión preliminar 1204121010

Ing. Arturo Miguel de Priego Paz Soldán

amiguel@pucp.edu.pe

www.tourdigital.net

Chincha, Perú, 12 de abril de 2012

Este módulo enseña a construir circuitos digitales virtualmente y a determinar sus propiedades mediante simulaciones lógicas. Los circuitos estudiados en este documento son puertas lógicas y biestables básicos. Este módulo utiliza un programa de construcción y simulación virtual, versión 0.9.8a. Las versiones anteriores de este programa (0.9.5 y 0.9.7) pueden descargarse desde la página www.tourdigital.net. El lenguaje utilizado en este documento es informal, con menciones en primera persona del plural y en segunda persona. Así que bienvenido, en este módulo aprenderás a encontrar las funciones que realizan varios circuitos integrados, a deducir circuitos a partir de tablas que especifican sus funciones, a dibujar diagramas lógicos y cronogramas, y a analizar circuitos para deducir sus funciones. También verás aplicaciones útiles de estos circuitos, como por ejemplo el control de nivel de agua en un tanque mediante el uso de sensores y accionamiento de una bomba de agua. Entonces, adelante, que te diviertas y aprendas mucho en este primer módulo del taller.

Contenido:

1. Análisis del Circuito Integrado 74LS00

Ejercicio 1. Circuito NOR

2. Un caso especial de la puerta NAND

3. Primer diseño de un circuito digital

Ejercicio 2. Circuito OR

4. Análisis de un circuito biestable

Ejercicio 3. Más circuitos integrados

Ejercicio 4. Control de nivel de agua en un tanque

Ejercicio 5. Función OR con puertas NAND

Ejercicio 6. Función XOR con puertas NAND

Ejercicio 7. Latch SR con habilitador

Este primer taller se divide en siete partes:

- I. Circuitos electrónicos y algoritmos básicos
- II. Circuitos digitales y programas típicos
- III. Álgebra de conmutación y programación orientada a objetos
- IV. Lógica programable y codiseño de hardware y software
- V. Máquinas de estados y sistemas embebidos
- VI. Diseño de controladores lógicos e interfaces de usuario

VII. Microprocesadores y microcontroladores

La parte I contiene siete módulos de introducción a:

- 1. Circuitos lógicos**
- 2. Circuitos eléctricos
- 3. Descripciones algorítmicas de circuitos digitales
- 4. Lenguaje de descripción de hardware
- 5. Programación orientada a objetos
- 6. Indagación científica y diseño en ingeniería
- 7. Diseño de sistemas digitales

1. Análisis del Circuito Integrado 74LS00

Antes de continuar es importante que observes en vídeos o imágenes de Internet cómo lucen los circuitos integrados, los protoboards, leds, interruptores y cables. Busca con las palabras claves y filtra los resultados de búsqueda primero por imágenes y después por vídeos. Dedicar un tiempo prudencial a esta actividad.

Ahora vamos a construir un circuito de prueba lógica del circuito integrado 74LS00 ejecutando los siguientes pasos (en cada etapa aprenderemos algunas características del programa junto con prácticas comunes de trabajo con circuitos digitales):

1. Ejecución del programa
2. Almacenamiento del circuito
3. Inserción de tableros
4. Interconexión con cables
5. Construcción de la red de polarización
6. Estados lógicos y tabla de verdad
7. Inserción del chip 7400
8. Conexión del circuito de prueba
9. Simulación lógica

1. Ejecución el programa

Ubica el programa en tu computador y ejecútalo. La versión utilizada en este documento es 0.9.8a pero puedes emplear la versión 0.9.7 para esta sección.

La figura 1 muestra la ventana inicial del programa. Tiene tres regiones principales: el menú, la barra de herramientas y el espacio de edición de circuitos. En todos los circuitos siempre habrá un tablero de fuente de voltaje, la cual sirve para energizar los circuitos digitales.

2

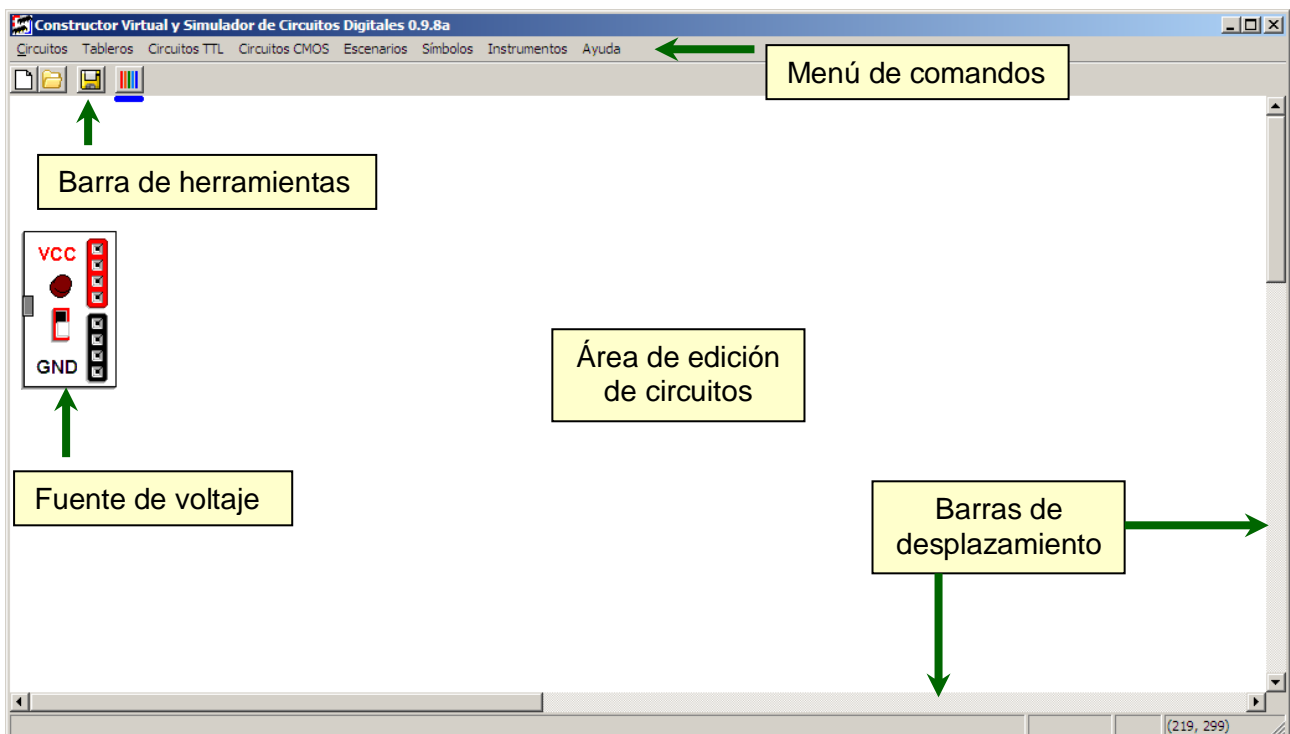


Figura 1. Ventana inicial del programa.

2. Almacenamiento del circuito

Guarda el circuito con el nombre *Test de Nand2*. La extensión automática de los archivos es **txt**.

El circuito puede ser almacenado en cualquier momento pero es recomendable hacerlo frecuentemente desde el inicio de la edición. Procura que los nombres de los circuitos representen la función lógica que realizan.

3. Inserción de tableros.

Inserta un tablero Protoboard, un tablero Interruptores y un tablero Leds (figura 2).

En este primer circuito utilizaremos un protoboard, interruptores y leds. Sobre el protoboard se insertan los circuitos integrados. Los interruptores sirven para variar el estado lógico de las entradas de los circuitos. Los leds se emplean como monitores de los estados lógicos de los nodos del circuito.

Para **desplazar un tablero** presiona el botón izquierdo del ratón sobre una de sus marcas, traslada el tablero y cuando esté en la posición elegida libera el botón izquierdo. En la figura 3 las flechas verdes señalan las marcas para mover y retirar los tableros.

Para **retirar un tablero** presiona el botón derecho del ratón sobre una marca. Un tablero puede retirarse solamente cuando está libre de cables y de circuitos integrados.

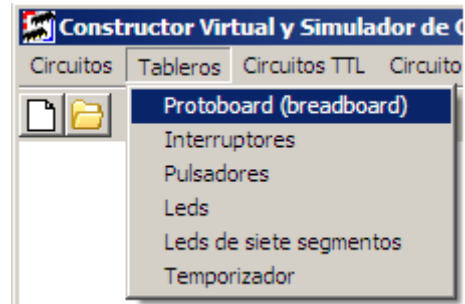


Figura 2. Menú Tableros.

3

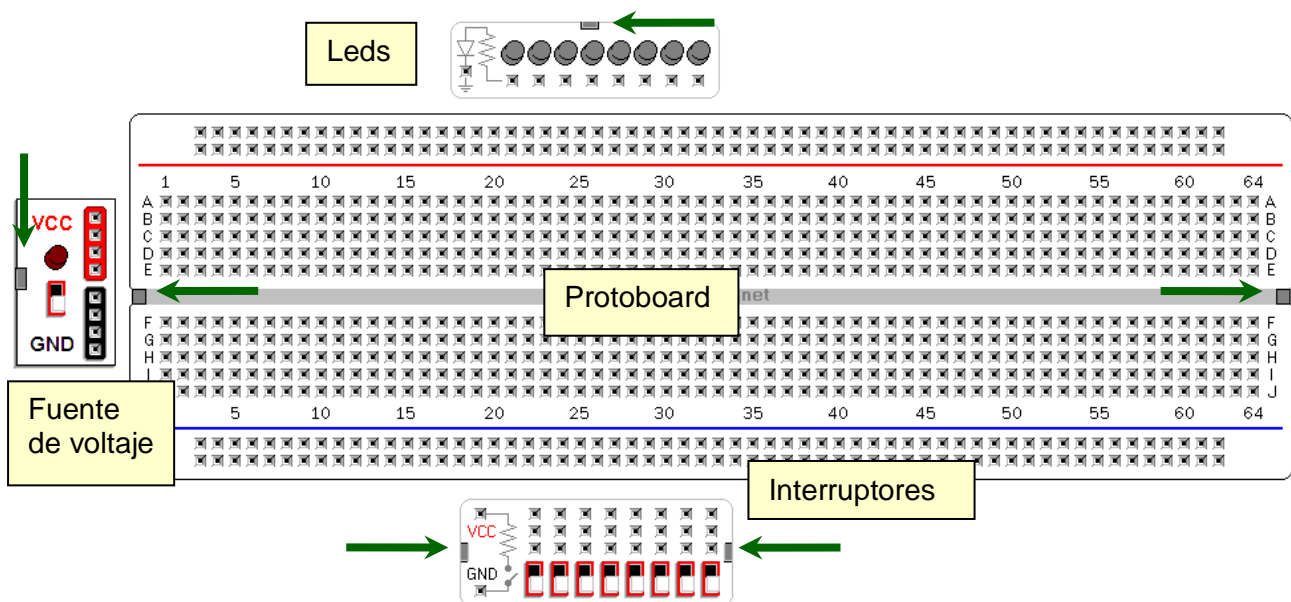


Figura 3. Algunos tableros del programa.

4. Interconexiones con cables

Los tableros contienen casillas donde pueden insertarse las puntas de los cables para hacer interconexiones. Las casillas están organizadas de varias maneras formando nodos eléctricos. En la fuente de voltaje, protoboard e interruptores las casillas de una misma columna están conectadas entre sí. El protoboard contiene 128 columnas de cinco casillas (64 en la parte superior y 64 en la parte inferior), la fuente de voltaje tiene dos columnas de cuatro casillas y el tablero de interruptores tiene ocho columnas de tres casillas. El tablero de leds contiene ocho casillas desligadas entre sí. Además, el protoboard contiene cuatro filas de casillas. Las casillas de cada fila están unidas entre sí y las casillas de filas diferentes están separadas.

Para unir casillas de nodos eléctricos diferentes se emplean cables. Los **cables se forman directamente** presionando el botón izquierdo del ratón sobre una casilla origen, moviendo el ratón hacia una casilla destino y soltando el botón izquierdo. **Este programa evita cortocircuitos**, es decir, no permite conexiones entre VCC, GND y nodos de salida. Por ejemplo, intenta hacer una conexión entre dos interruptores, entre un interruptor y VCC o entre VCC y GND.

Para **establecer el color del cable** actual presiona el botón de colores de la barra de herramientas. Para **cambiar la forma de un cable** presiona el botón izquierdo del ratón sobre un segmento y arrastra el ratón. Hasta tres segmentos del cable pueden ser modificados. Para **retirar un cable** pulsa el botón derecho del ratón sobre uno de los segmentos editables.

Las ediciones del circuito solamente pueden ser hechas con la fuente de voltaje apagada.

5. Construcción de la red de polarización

Establece los nodos de VCC y GND sobre el protoboard y sobre los tableros de entrada y salida.

La figura 4 muestra una conexión típica de VCC y GND. Es recomendable utilizar cables de color negro para GND y de color rojo para VCC. Nota que la línea de GND se encuentra envolviendo a la línea de VCC. Esta configuración ayuda a blindar al circuito contra ruidos eléctricos. Observa que el tablero de leds debe conectarse a GND y el de interruptores a VCC y GND.

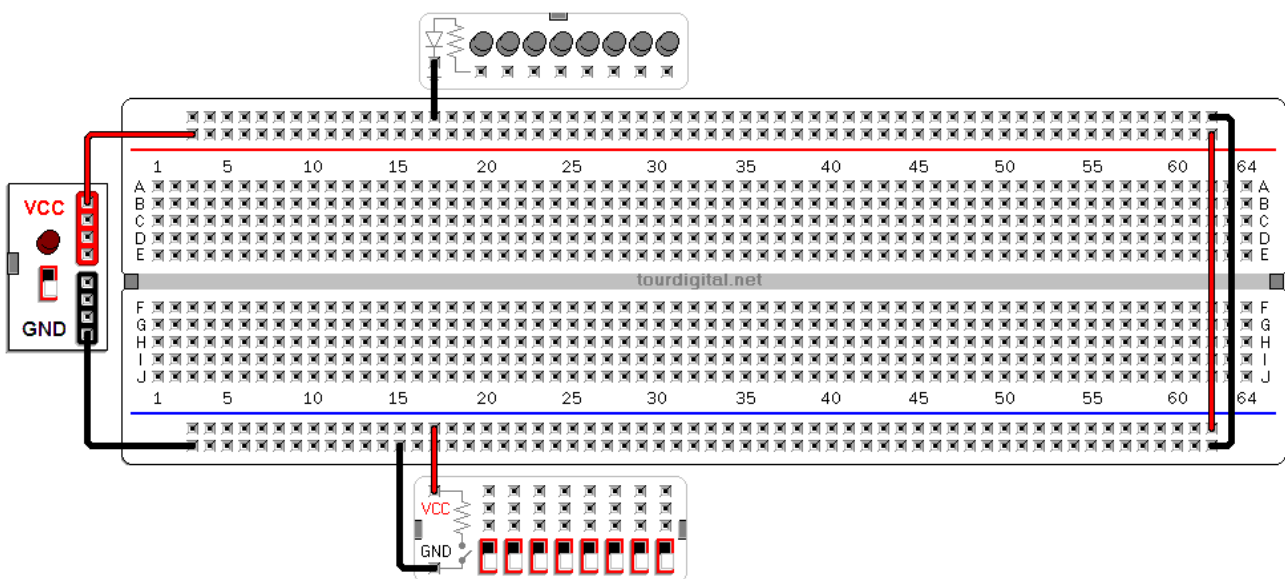


Figura 4. Una configuración de una red de polarización.

6. Estados lógicos y tabla de verdad

Para verificar las conexiones conecta un interruptor con un led.

La figura 5 ilustra una conexión. Enciende el módulo pulsando sobre el **interruptor de la fuente de voltaje**. Para **cambiar el color de un led** pulsa el botón izquierdo sobre el led elegido.

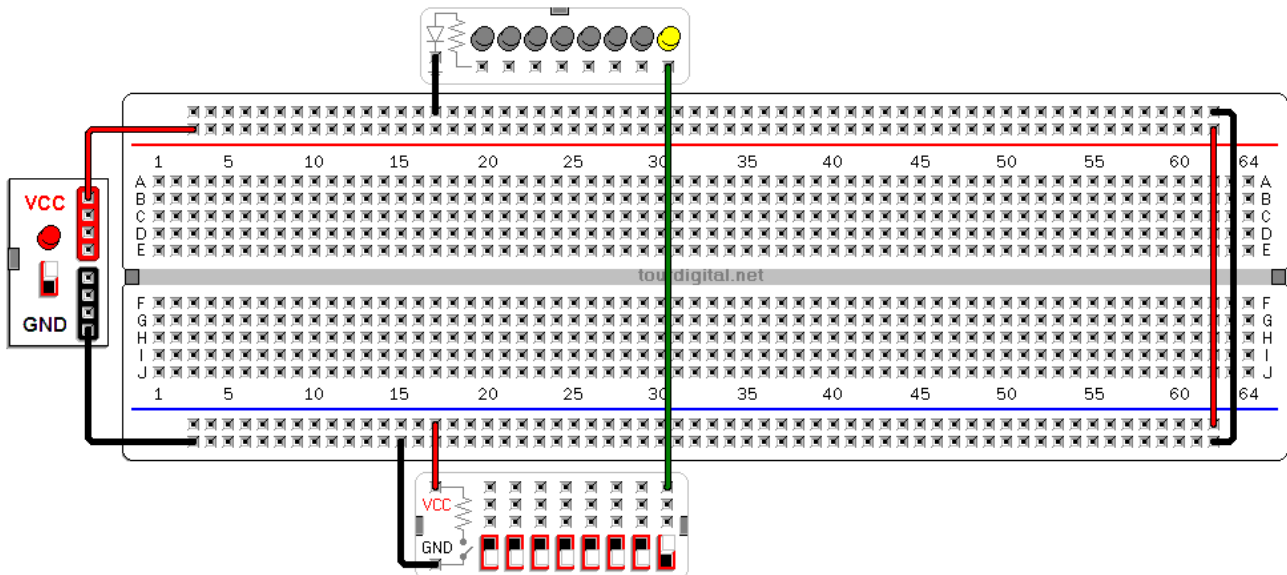


Figura 5. Circuito de prueba de conexión entre un interruptor y un led.

5

Observando los estados del led y los estados del interruptor podemos elaborar la tabla 1. En diseño lógico esta tabla recibe el nombre de **tabla de verdad**. En la tabla de verdad se especifica el valor de la salida para cada combinación de las entradas. Cuando el interruptor está con la pestaña hacia abajo se dice que está en el **estado lógico OFF** o 0. Con la pestaña hacia arriba está en el **estado lógico ON** o 1. Cuando el led está encendido se dice que está en el estado ON o 1, y cuando está apagado está en el estado OFF o 0. En la tabla el interruptor aparece con el nombre SW y el led se llama LED.

SW		LED	
OFF, 0		OFF, 1	
ON, 1		ON, 1	

Tabla 1. Tabla de verdad mostrando las relaciones entre los estados lógicos de un interruptor y de un led.

7. Inserción de un circuito integrado 7400.

Selecciona el circuito integrado 7400 (figura 6) e insértalo en el protoboard.

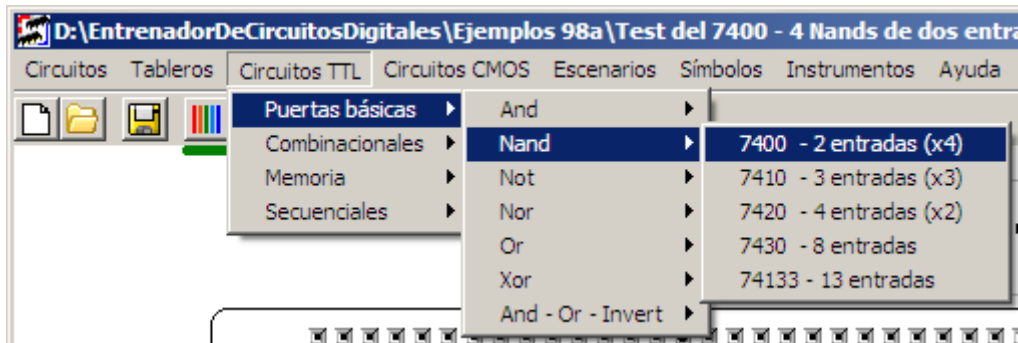


Figura 6. Selección de un chip 7400.

El circuito integrado (chip) aparecerá en la parte superior izquierda de la ventana. Para **trasladar un chip** presiona el botón izquierdo sobre el chip, mueve el ratón y ubica el chip muy cerca de sus casillas de inserción; luego suelta el botón izquierdo. Si el chip se coloca lejos de las casillas retornará a la esquina superior izquierda. Para **retirar un chip** pulsa el botón derecho del ratón sobre el chip.

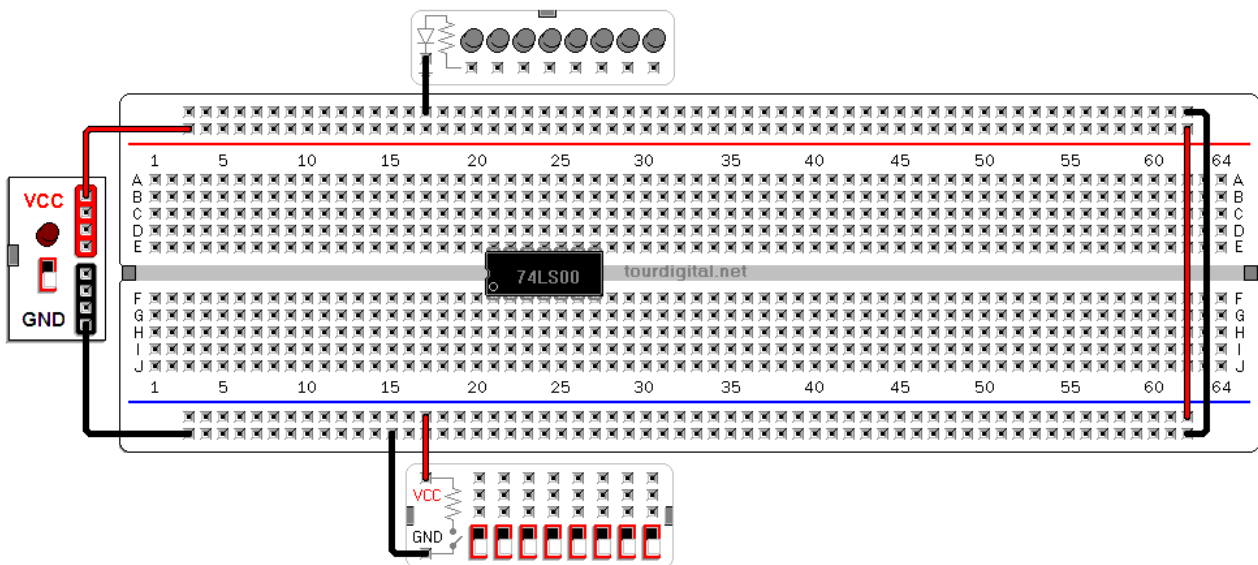


Figura 7. Inserción en el protoboard de un chip 74LS00.

8. Conexión del circuito de prueba

Conecta las líneas de VCC, GND y las entradas y salidas del chip con los nodos correspondientes en los tableros.

Para conectar el chip con los elementos externos es necesario conocer su **diagrama de conexiones**. Este diagrama se encuentra en las hojas de datos provistas por el fabricante del chip, y pueden obtenerse en Internet con las palabras de búsqueda *datasheet 74LS00*.

La figura 8 muestra un diagrama de conexiones del chip 74LS00. Este chip contiene cuatro **puertas lógicas** que realizan la función **NAND**, cuyo **símbolo lógico** aparece dibujado en el mismo diagrama. Una de las puertas lógicas tiene entradas en los pines 1 y 2, y salida en el pin 3. Ello significa que podemos cambiar el estado lógico de los pines 1 y 2 para modificar a través de ellos el estado lógico del pin 3. Similarmente, los pines 4 y 5 son las entradas de la NAND con salida 6. De igual modo, 9 y 10 dan la salida 8 mientras que los pines 13 y 12 originan la salida 11. El pin 14 va a VCC y el pin 7 a GND. En general, cada tipo de chip tiene su propio diagrama de conexiones.

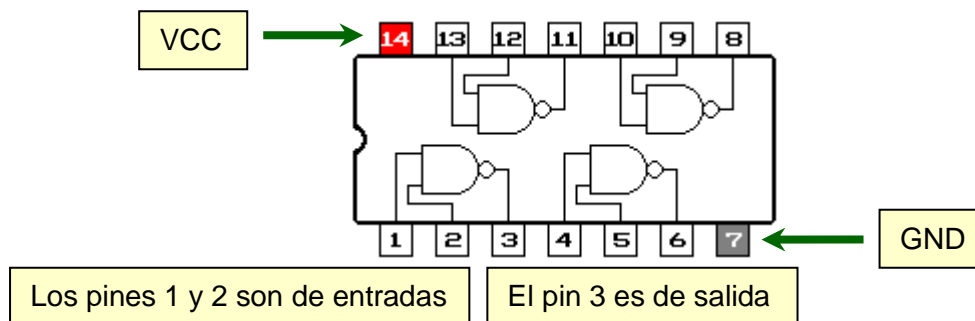


Figura 8. Diagrama de conexiones del chip 7400.

Un pin de entrada puede conectarse a un dispositivo externo que provea señales al circuito (interruptor, pulsador) o a otro nodo o terminal del mismo circuito donde forma parte. Un pin de salida puede conectarse con un dispositivo que reciba señales (led) o con un nodo que alimente a otro pin de entrada del circuito. En este ejemplo, los interruptores cambian los estados lógicos de las entradas del chip y observamos el estado lógico resultante a través del led.

La figura 9 muestra una conexión para examinar el funcionamiento de la puerta lógica que está conectada con los pines de entrada 1 y 2 y el pin de salida 3. Los pines 1 y 2 están conectados con interruptores y el pin 3 está conectado con un led. Asimismo, el pin 14 está conectado a la línea de VCC, mientras que el pin 7 está conectado a la línea de GND.

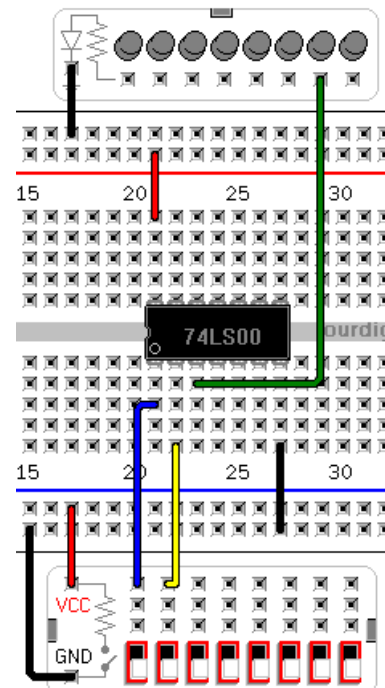


Figura 9. Un circuito de prueba del chip 7400.

Simula el funcionamiento de este circuito y define su tabla de verdad.

Este circuito está conectado a dos interruptores. Cada interruptor tiene dos estados físicos (pestaña arriba o abajo) que producen dos estados lógicos (ON, OFF o 1, 0). El led tiene dos estados físicos (encendido, apagado) que pueden representarse también con ON, OFF o con 1, 0. El número de combinaciones diferentes de estados de los interruptores se calcula como 2 (número de estados por interruptor) x 2 (número de interruptores), que da un total de 4 combinaciones. Al simular este circuito notamos que cuando ambos interruptores están OFF el led está ON; cuando los interruptores están en estados opuestos (ON-OFF y OFF-ON) el led está ON; y cuando ambos interruptores están ON el led está OFF. Con estos datos definimos la tabla de verdad (tabla 2).













SW1, pin 1	SW2, pin 2	LED, pin 3
OFF, 0 	OFF, 0 	ON, 1 
OFF, 0 	ON, 1 	ON, 1 
ON, 1 	OFF, 0 	ON, 1 
ON, 1 	ON, 1 	OFF, 0 

Tabla 2. Tabla de verdad de la función lógica NAND de dos entradas.

Algebraicamente la función NAND entre el led y los interruptores se escribe:

$$LED = SW1 \text{ nand } SW2$$

En este caso, el orden de las señales no interesa, es decir, podemos expresar:

$$LED = SW2 \text{ nand } SW1$$

Ahora es tu turno. Construye el circuito de la figura 10 y verifica que las otras tres puertas del chip cumplen la misma función. Seguidamente, simula el circuito de la figura 11 para observar todas las combinaciones juntas, al mismo tiempo.

Es importante que verifiques el funcionamiento de cada puerta antes de utilizar el chip en un circuito más grande.

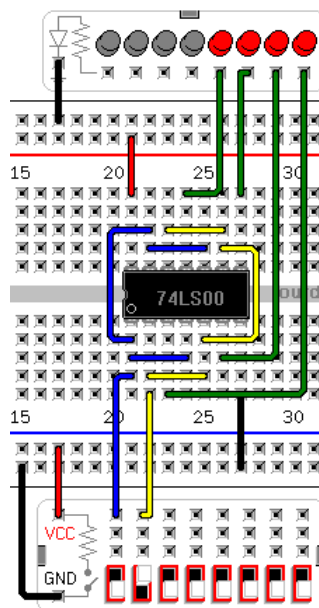


Figura 10. Circuito de verificación de todas las puertas lógicas del 7400

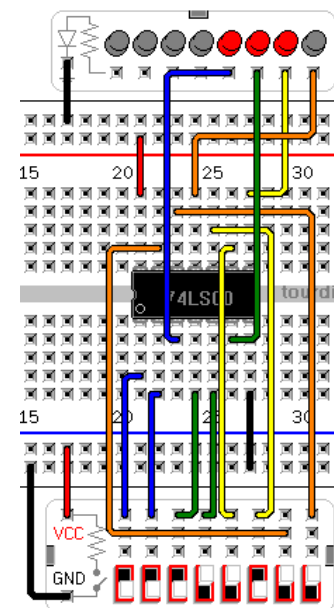


Figura 11. Circuito de verificación simultánea de la función NAND.

Ejercicio 1

Verifica el funcionamiento del circuito 7402 con la tabla 3. Las puertas lógicas de este circuito se llaman NOR de dos entradas. Construye tres circuitos similares a los que hemos visto para el 7400.

En este chip el pin 14 se conecta con VCC y el 7 con GND (figura 12). Nota que el sentido de orientación entre salidas y entradas de las puertas NOR del 7402 es contrario al de las puertas NAND del 7400. Todos los circuitos integrados tienen un diagrama de conexiones que debe respetarse, caso contrario el chip puede dañarse o causar daño a otros componentes.

En todos los ejercicios debes explicar textualmente cada paso que realizas. Es importante que escribas todo tu razonamiento e inclusive tus dudas y observaciones para que podamos seguir de cerca tus avances y progreso.

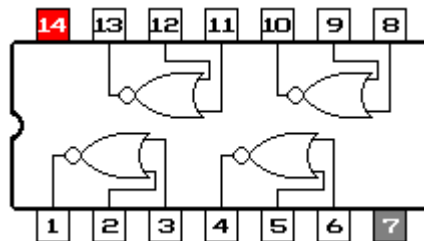


Figura 12. Diagrama de conexiones del chip 7402.

SW1	SW2	LED
OFF, 0	OFF, 0	ON, 1
OFF, 0	ON, 1	ON, 1
ON, 1	OFF, 0	ON, 1
ON, 1	ON, 1	OFF, 0

Tabla 3. Especificación de la función lógica NOR.

2. Un caso especial de la puerta NAND

En el mundo real todas las señales eléctricas tardan un tiempo en propagarse por los sistemas electrónicos. En el siguiente análisis vamos a despreciar esos retardos considerando que son muy pequeños comparados con la señal de periodo de un segundo (frecuencia de 1Hz).

Construye el circuito de la figura 13. En la versión 0.9.7 no existe el osciloscopio, pero puedes seguir el análisis a partir del **cronograma** (gráfica temporal de la secuencia de señales).

Observa el comportamiento del led y de la señal de 1 Hz. Con el pin 1 (conectado al interruptor) en 0 el led permanece encendido, mientras que con el pin 1 en 1 el led oscila opuestamente a la señal de 1 Hz, es decir, si la señal está en 1 entonces el led está en 0 y viceversa. Esta función, la de invertir el valor de una señal, se llama NOT. El cronograma que aparece en el osciloscopio muestra la secuencia en que cambian las señales de entrada y de salida. Observa que la salida solamente depende del estado actual de la señal de entrada (otros estados pasados no la afectan).

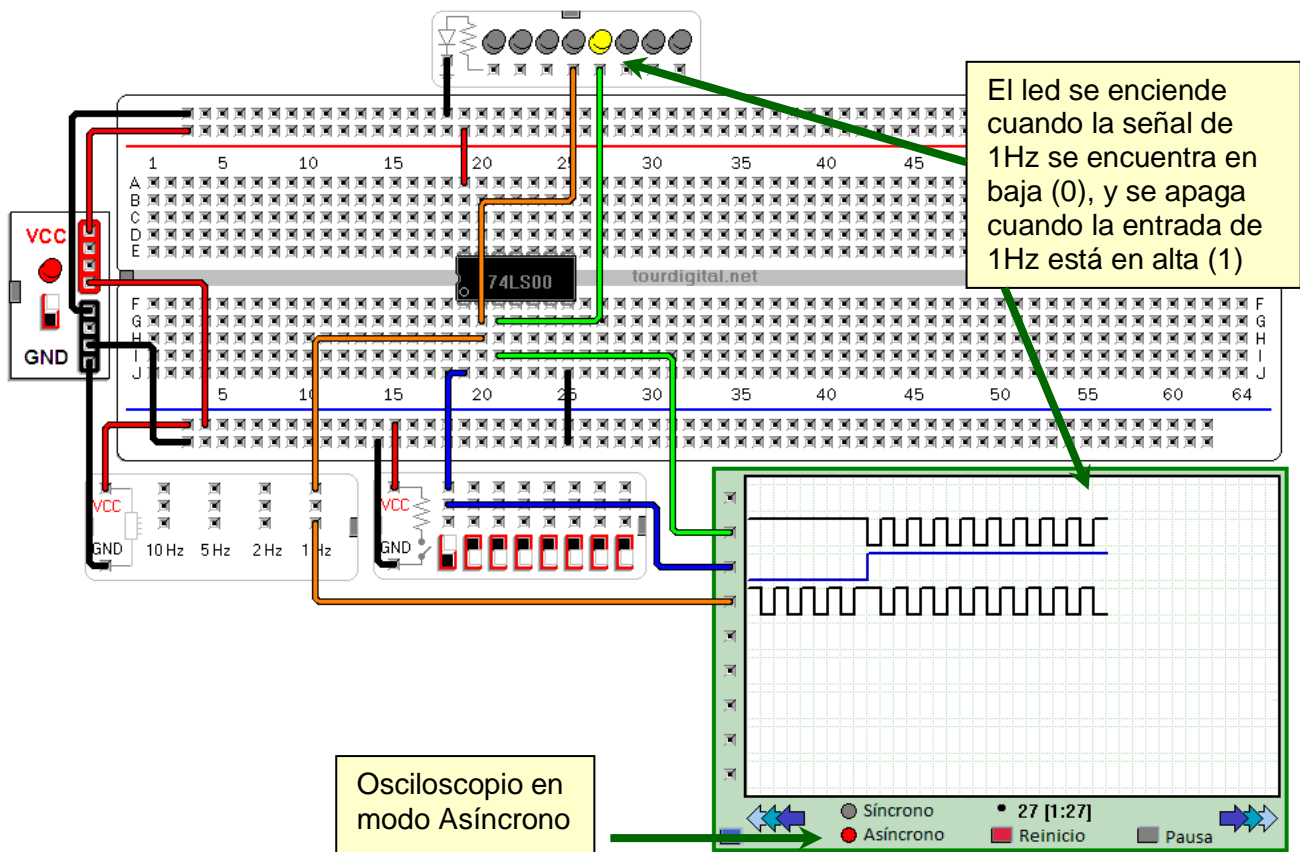














Figura 13. Simulación del circuito NAND con una entrada funcionando como habilitadora.

Modifica el circuito de la siguiente manera: elimina la conexión del pin 1 al interruptor y conecta este pin 1 a la línea de VCC. Al simular el circuito verás que se comporta como el caso anterior cuando el interruptor estaba en 1. Si llamamos a la señal de entrada T1HZ y a la salida LED, podemos expresar algebraicamente esta función como:

$$\text{LED} = \text{not T1HZ}$$

Este resultado pudo ser deducido desde la tabla de verdad, como indica la figura 14.

SW1, pin 1	SW2, pin 2	LED, pin 3
OFF, 0 	OFF, 0 	ON, 1 
OFF, 0 	ON, 1 	ON, 1 
ON, 1 	OFF, 0 	ON, 1 
ON, 1 	ON, 1 	OFF, 0 

Cuando una entrada es cero, la salida es 1

Cuando una entrada es uno, la salida tiene el valor opuesto al de la otra entrada

Figura 14. Análisis de casos especiales de la función NAND.

Podemos hacer el análisis separando los casos especiales. Si SW1 se mantiene fijado a 1 (si nunca cambia a 0) resulta la tabla 4. En este caso se dice **LED = not SW2**.







SW1, pin 1	SW2, pin 2	LED, pin 3
ON, 1 	OFF, 0 	ON, 1 
ON, 1 	ON, 1 	OFF, 0 

Tabla 4. Caso especial cuando SW1 permanece en 1 y SW2 puede cambiar.

Similarmente, si SW2 permanece fijado a 1 (si nunca cambia a 0) resulta la tabla 5 y entonces la expresión algebraica es **LED = not SW1**.







SW1, pin 1	SW2, pin 2	LED, pin 3
OFF, 0 	ON, 1 	ON, 1 
ON, 1 	ON, 1 	OFF, 0 

Tabla 5. Caso especial cuando SW2 permanece en 1 y SW1 puede cambiar.

3. Primer diseño de un circuito digital

A continuación abordaremos el diseño de nuestro primer circuito lógico. ¿Cómo podemos deducir un circuito que produzca la función lógica AND definida en la tabla 6? Dedica un tiempo para reflexionar sobre la pregunta y para imaginar unas respuestas.

























SW1	SW2	LED
OFF 	OFF 	ON 
OFF 	ON 	ON 
ON 	OFF 	ON 
ON 	ON 	OFF 

Tabla 6. Especificación de la función lógica AND.

12

Comparando la tabla 6 con la tabla de la función NAND vemos que las salidas correspondientes a cada combinación de las entradas están en estados lógicos opuestos. Por lo tanto, para obtener la salida requerida bastará con invertir (**complementar**) la salida de la función NAND. Para invertir una señal hemos visto que debemos colocar una entrada de una puerta NAND en 1 y conectar la otra entrada con la señal. Una forma diferente de obtener una función NOT consiste en colocar la misma señal a ambas entradas de la puerta NAND como se demuestra en la figura 15:

SW1, pin 1	SW2, pin 2	LED, pin 3
OFF, 0 	OFF, 0 	ON, 1 
OFF, 0 	ON, 1 	ON, 1 
ON, 1 	OFF, 0 	ON, 1 
ON, 1 	ON, 1 	OFF, 0 

Cuando las entradas son iguales, la salida tiene el valor opuesto al de ellas.

Figura 15. La función lógica NOT se obtiene al unir las entradas de una puerta NAND.

La figura 16 muestra el diagrama lógico del nuevo circuito. La primera puerta NAND, conectada a los interruptores, realiza la función NAND normalmente, pero la segunda puerta NAND tiene sus entradas A y B conectadas al mismo nodo X que es salida de la primera puerta NAND. Esta configuración produce la salida deseada en el LED, pues cada valor de X es invertido por la segunda puerta. Algebraicamente esta nueva función entre la salida LED y los interruptores SW1 y SW2 se expresa así:

$$\text{LED} = \text{SW1 and SW2} = \text{not} (\text{SW1 nand SW2})$$

SW1	SW2	X	A	B	LED
0	0	1	0	0	1
0	1	1	0	0	1
1	0	1	1	1	0
1	1	0	1	1	0

$$X = \text{SW1 nand SW2}$$

$$\text{LED} = \text{not X}$$

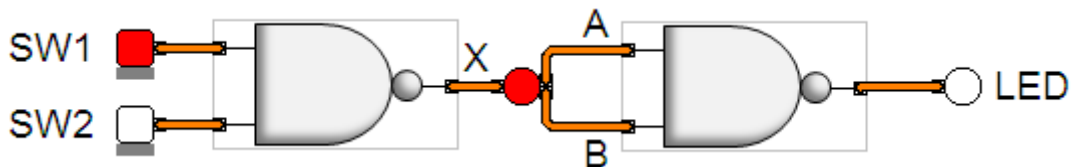


Figura 16. Un circuito lógico de una función AND.

Para construir este circuito sobre el protoboard necesitamos especificar todas las conexiones: desde los interruptores hacia los pines de entrada, los nodos intermedios y desde los pines de salida hacia los leds. Como el chip 7400 contiene cuatro puertas lógicas, pueden utilizarse dos de ellas para esta función. En general, la elección de las puertas depende de sus posiciones relativas y del espacio de interconexión. En la figura 17 se eligieron las puertas que están ubicadas en la zona baja del chip.

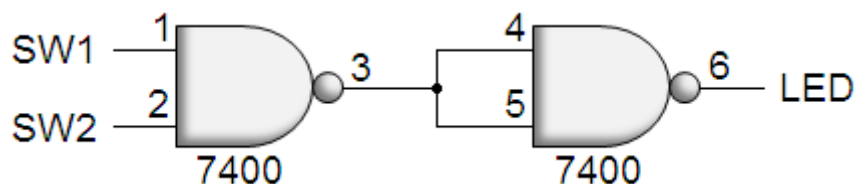


Figura 17. Un diagrama lógico de un circuito AND hecho con puertas NAND

El diagrama de la figura 17 indica lo siguiente:

- ⇒ El interruptor SW1 se conecta con el pin 1 del 7400.
- ⇒ El interruptor SW2 se conecta con el pin 2 del 7400.
- ⇒ El pin 3 del 7400 se conecta con el pin 4 y con el pin 5 del 7400.
- ⇒ El pin 6 del 7400 se conecta con el led.

El circuito lógico resultante se muestra en la figura 18. Ahora simula el circuito y verifica que se cumplan las condiciones de la tabla 6.

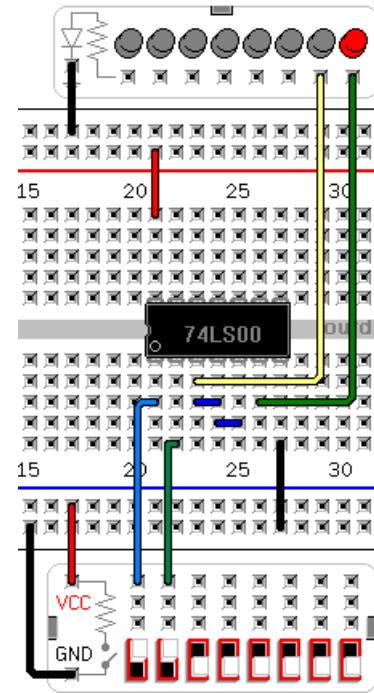


Figura 18. Función lógica AND utilizando puertas lógicas NAND.

Ejercicio 2

Construye un circuito lógico para producir la tabla 7 utilizando solamente puertas NOR.













SW1	SW2	LED
OFF 	OFF 	ON 
OFF 	ON 	ON 
ON 	OFF 	ON 
ON 	ON 	OFF 

Tabla 7. Especificación de la función lógica OR de dos entradas.

4. Análisis de un circuito biestable

Observa el circuito lógico de la figura 19. Es un circuito biestable de tipo *latch* SR con puertas NAND. En este circuito las salidas de las NAND son realimentadas recíprocamente hacia las entradas de las NAND. Esta característica brinda memoria al circuito: cuando las señales que cambian las salidas dejan de actuar, el circuito se mantiene en el último estado registrado.

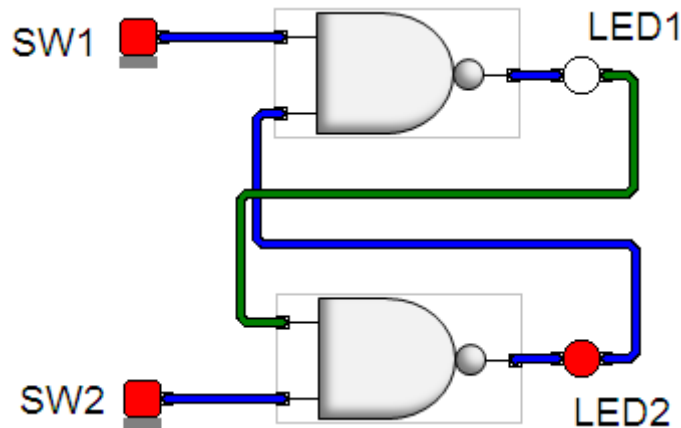


Figura 19. Un circuito lógico de un latch SR.

Analicemos este circuito. Recordemos de la sección 3 (página 11) que cuando una entrada de la puerta NAND está en 0 la salida resulta en 1 (independientemente del valor de la otra entrada), y si una entrada está en 1 la salida tiene el complemento (valor contrario) al de la otra entrada. Entonces podemos decir:

15

- ⇒ Cuando los dos interruptores están en 0 ambas salidas están en 1.
- ⇒ Si SW1 está en 0 y SW2 en 1 entonces LED1 está en 1 y así el pin 4, por lo tanto LED2 = not SW2 = not 1 = 0
- ⇒ Si SW1 está en 1 y SW2 en 0 entonces LED2 está en 1 y así el pin 9, por lo tanto LED1 = not SW1 = not 1 = 0
- ⇒ Si ambos interruptores están en 1, entonces LED1 = not LED2 y LED2 = not LED1. Por lo tanto LED1 = not (not LED1) = LED1 (pues invirtiendo dos veces una misma señal resulta la propia señal). Este resultado indica que LED1 y LED2 no cambian cuando los interruptores están en 1.

Este análisis podemos expresarlo en una tabla de verdad (tabla 8).

SW1	SW2	LED1	LED2
0	0	1	1
0	1	1	0
1	0	0	1
1	1	LED1	LED2

Tabla 8. Tabla de verdad de un latch SR.

¿Cuáles son los estados lógicos específicos de LED1 y LED2 cuando SW1 y SW2 están en 1? Es importante mencionar que este tipo de circuito se utiliza en aplicaciones donde las señales de entrada no cambian simultáneamente. Dicho eso, si SW1 y SW2 están en 1 significa que antes ocurrió uno de dos casos: SW1 en 0 y SW2 en 1 o SW1 en 1 y SW2 en 0. Si ocurrió el primer caso entonces LED1 antes tenía 1 y LED2 tenía 0 y esos son los valores actuales, pero si ocurrió el segundo caso entonces LED1 tenía 0 y LED2 tenía 1 y esos son sus valores actuales.

	Estados anteriores				Estados actuales			
	Entradas		Salidas		Entradas		Salidas	
Casos	SW1	SW2	LED1	LED2	SW1	SW2	LED1	LED2
1	0	1	1	0	1	1	1	0
2	1	0	0	1	1	1	0	1

Tabla 9. Estado de los leds de salida ante dos secuencias de entradas.

Para comprobar que este razonamiento es correcto vamos a construir el circuito. Primero, dibujamos un diagrama lógico para indicar cómo vamos a conectarlo en el chip. Esta vez usaremos las NAND con pines de entradas 4, 5 y 9, 10 (figura 20). Una vez definidas las interconexiones, el circuito en el protoboard resulta similar al de la figura 21 donde SW1 y LED1 están ubicados a la izquierda de SW2 y LED2, respectivamente. Simula este circuito, pero antes de cambiar el estado de los interruptores deduce que ocurrirá con los leds.

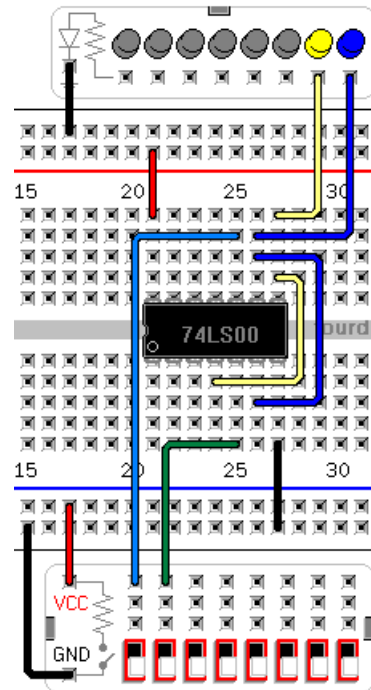
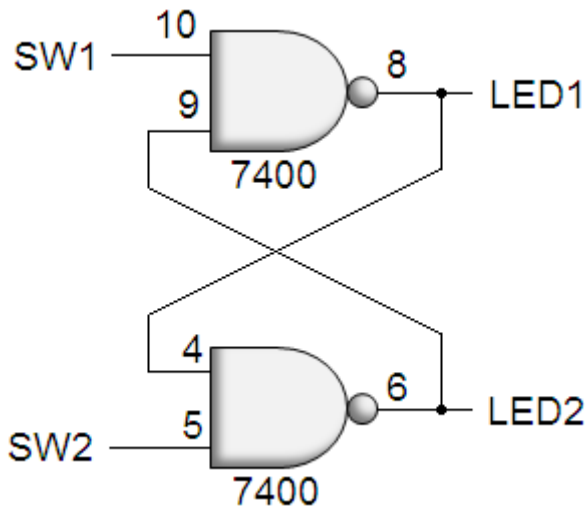


Figura 20. Un diagrama de conexiones del latch SR. Figura 21. Circuito de prueba del latch SR.

La figura 22 ilustra el empleo de un osciloscopio para registrar las señales en un cronograma conforme avanza el tiempo. Analiza los patrones de las señales en el osciloscopio y comprueba que están en armonía con las tablas 8 y 9.

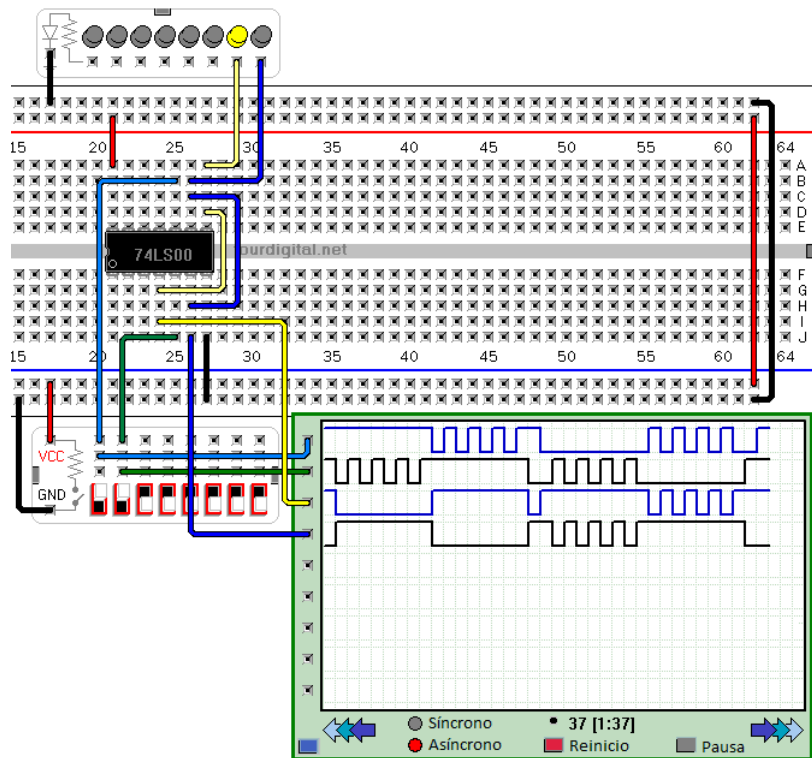


Figura 22. Cronograma para análisis del latch SR.

La figura 23 muestra una aplicación práctica de este circuito. Inicialmente, ambos interruptores están en el estado 1. De pronto, un interruptor baja por un momento a 0 y retorna a su posición original. Un tiempo después el otro interruptor hace lo mismo.

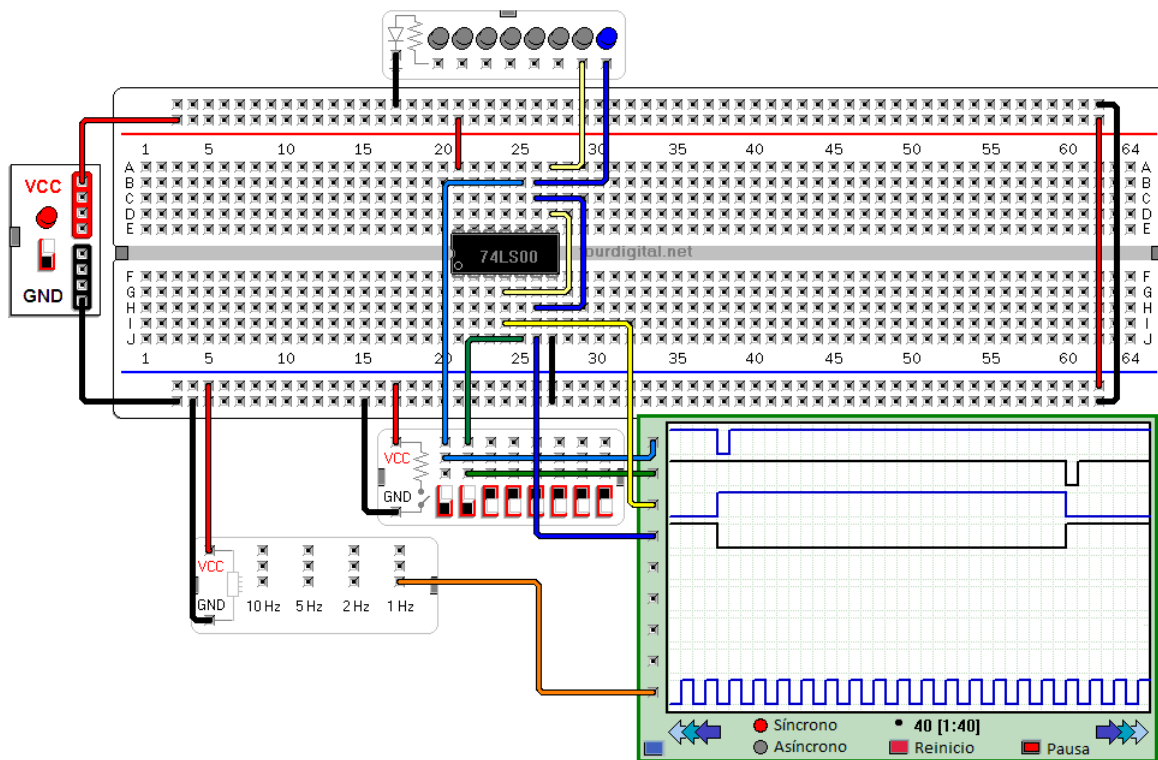


Figura 23. Señales de activación y desactivación temporal del latch SR.

Estos eventos pueden aplicarse para detectar si una puerta fue abierta y activar una alarma (figura 24). Al abrirse una **puerta** se activa el estado lógico 0 que enciende la **alarma**, la cual permanecerá encendida hasta que la señal **control** se active en 0. ¿Qué ocurre si la puerta permanece abierta? En el segundo módulo aprenderemos a fabricar interruptores y pulsadores así como a utilizar leds y fuentes de voltaje para que construyas y utilices este circuito en tu casa.

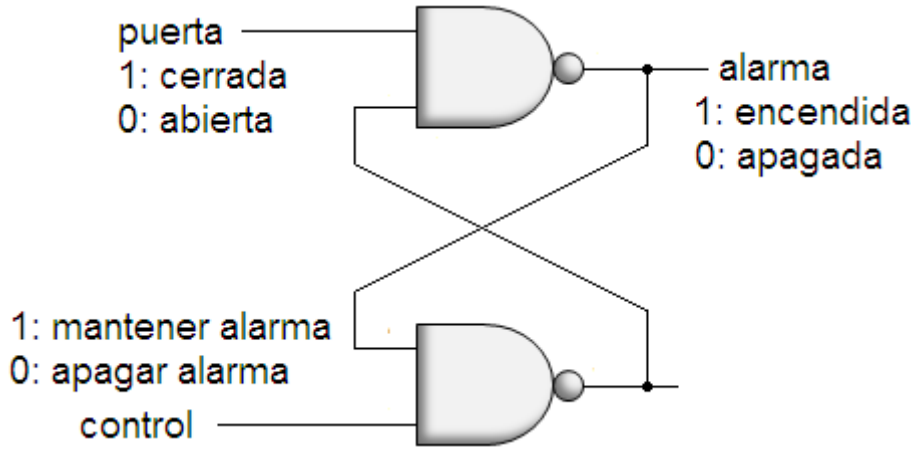


Figura 24. Aplicación del latch SR como circuito de alerta.

La figura 25 muestra una aplicación para controlar el nivel de agua en un tanque apagando o enciende una bomba. Los sensores de nivel se activan en el nivel lógico 1. La señal que controla la llave de agua no forma parte del circuito de control. La versión 0.9.5 de este programa tiene un escenario que permite simular el llenado y vaciado del tanque. Antes de encender el módulo en tal versión asegura que el tanque esté al tope de agua.

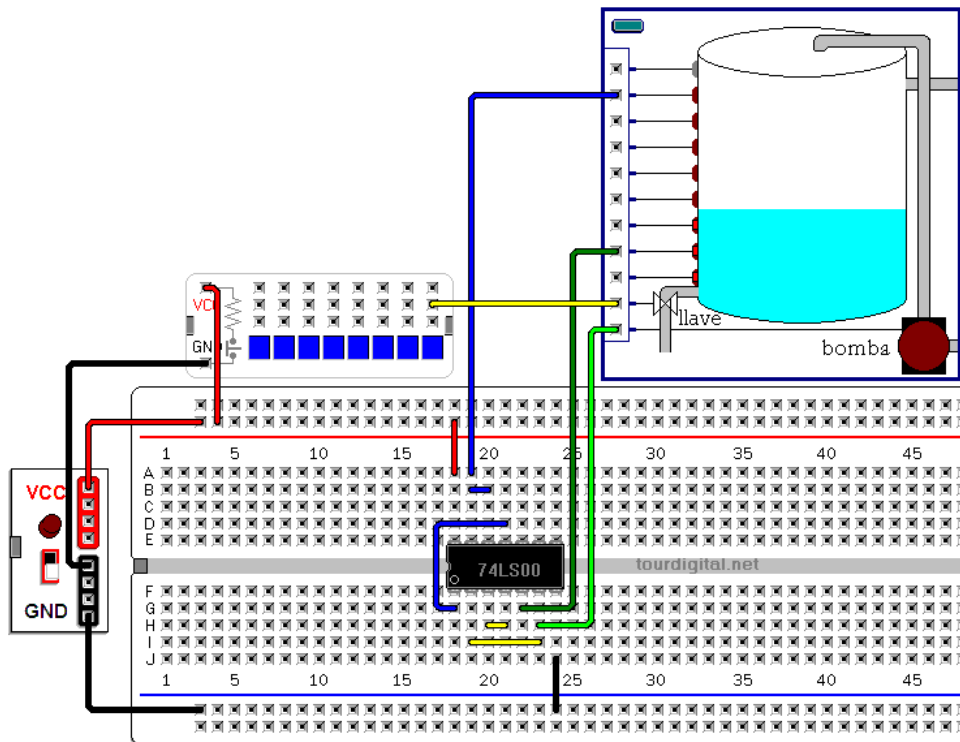


Figura 25. Circuito de control del nivel de agua en un tanque utilizando un latch SR y un inversor.

Ejercicio 3

En este ejercicio vas a conocer más funciones lógicas realizadas por otros circuitos integrados. Obtén la tabla de verdad de cada uno de los circuitos siguientes: 7404, 7408, 7432 y 7486. Todos ellos se encuentran en el mismo menú del 7400. Salvo el 7404, los demás chips tienen la misma configuración de pines, por lo tanto, para ahorrar tiempo de edición, solamente retira un chip y replázalo por otro en el circuito de prueba.

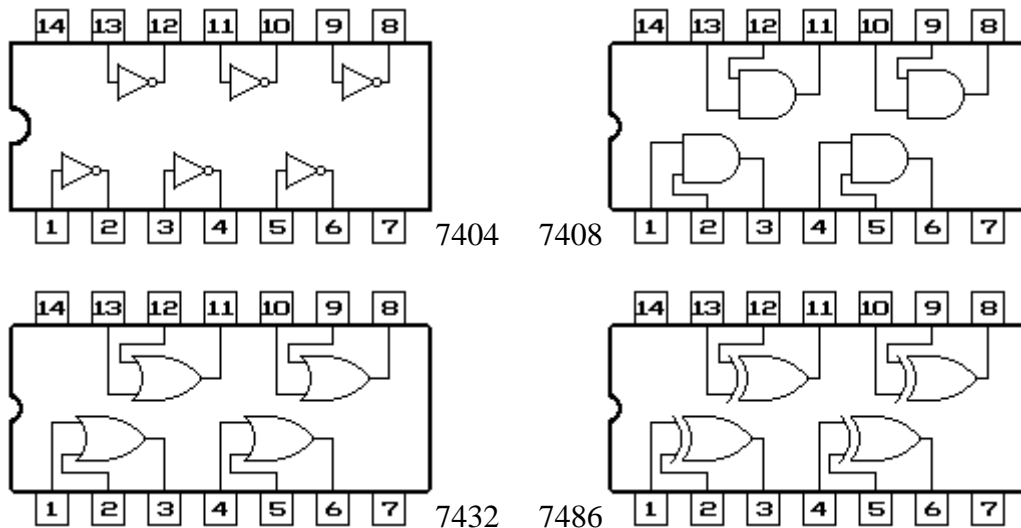


Figura 26. Otros circuitos integrados SSI

Ejercicio 4

Este ejercicio requiere que analices el circuito de control de la bomba del tanque de agua (figura 25) para obtener el diagrama lógico y la tabla de verdad que lo definen, así como un cronograma que ilustre su funcionamiento temporal. Explica la utilidad de cada parte del circuito en relación a la función que realizan en conjunto.

Ejercicio 5

Analiza el siguiente circuito lógico para determinar su tabla de verdad. Luego dibuje un diagrama lógico y construya un circuito de prueba, simule el circuito y verifique la tabla de verdad con los resultados de la simulación. ¿Cómo podría deducir este circuito a partir de la tabla de verdad de la función NAND?

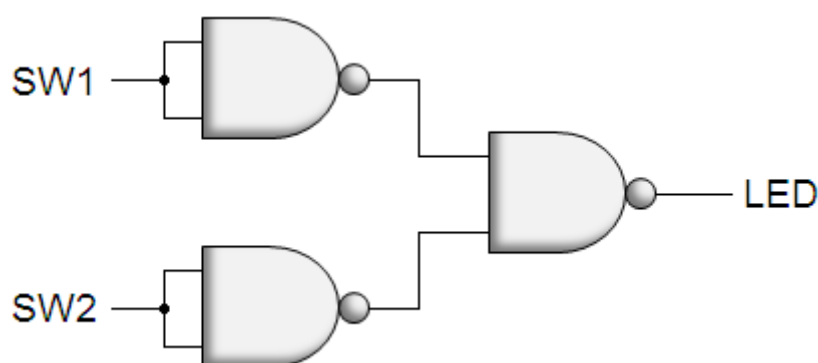


Figura 27. Un circuito lógico para una OR de dos entradas.

Ejercicio 6

La función que realiza el siguiente circuito lógico se llama XOR de dos entradas. Analízalo para determinar su tabla de verdad. Dibuja un diagrama lógico y construye un circuito de prueba, simula el circuito y verifica la tabla de verdad con los resultados de la simulación. ¿Cómo podrías deducir este circuito a partir de la tabla de verdad de la función NAND?

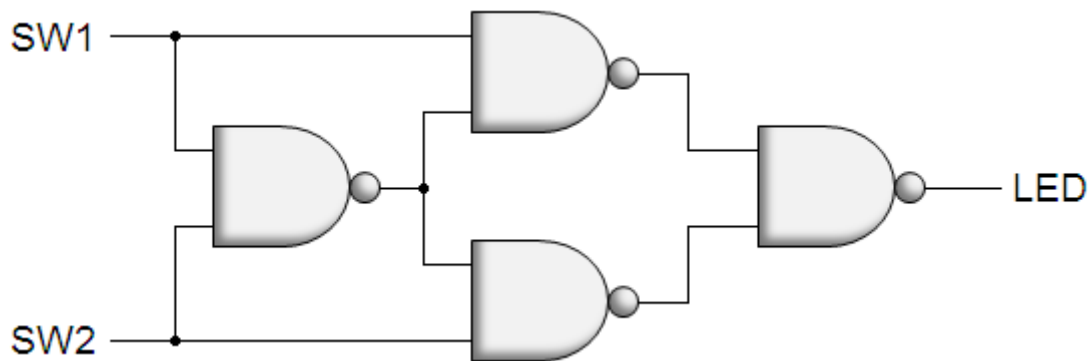


Figura 28. Circuito lógico para una función XOR.

Ejercicio 7

El siguiente circuito es un latch SR con habilitador. Haz lo siguiente:

- ⇒ Analiza este circuito para deducir su tabla de verdad.
- ⇒ Dibuja un cronograma para mostrar su funcionamiento temporal.
- ⇒ Dibuja un diagrama lógico y construye un circuito de prueba.
- ⇒ Simula lógicamente el circuito y verifica que la tabla de verdad y el cronograma se correspondan con los resultados de la simulación.

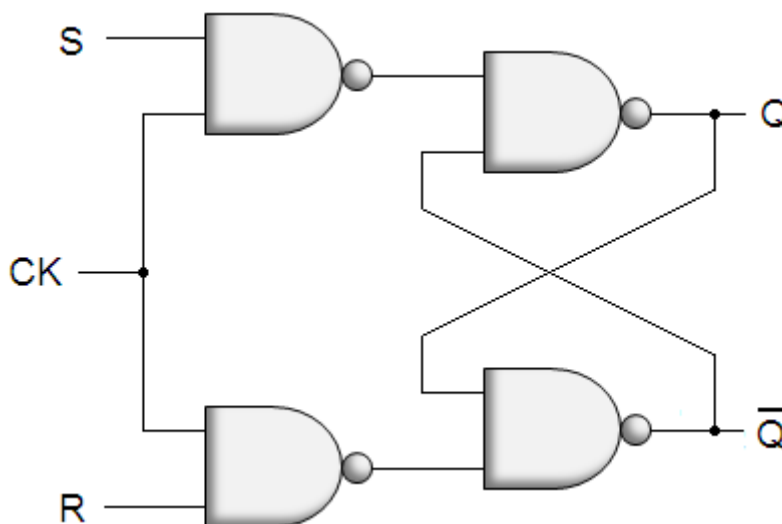


Figura 29. Latch SR con señal de habilitación.