

## **Una Herramienta de Diseño Virtual de Circuitos Digitales para un Aprendizaje Basado en Proyectos de Ingeniería**

**Ing. Arturo J. Miguel de Priego Paz Soldán**

*www.tourdigital.net*

*arturo@tourdigital.net*

*amiguel@pucp.edu.pe*

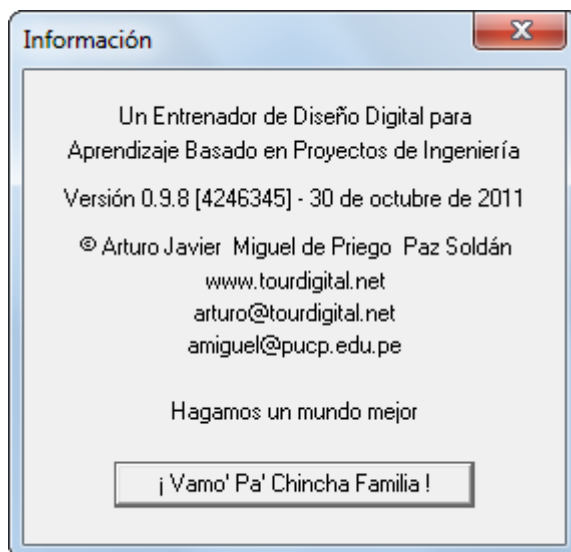
*Chincha, Perú*

*30 de octubre de 2011*

Este documento describe las características y utilidad de un programa de computadora especialmente diseñado para servir de apoyo en un curso de introducción al diseño lógico con circuitos digitales. El programa provee los elementos básicos para abordar el diseño digital desde varias representaciones de diseño y ensayar las soluciones con circuitos integrados SSI/MSI, de lógica programable, diagramas esquemáticos y escenarios virtuales. El usuario puede insertar tableros (protoboards, interruptores, leds, temporizadores), circuitos integrados TTL (puertas lógicas, selectores, sumadores, registros, contadores, etc.), osciloscopios, circuitos de lógica programable (PLA, PLD), funciones lógicas abstractas (diagramas de flujo, símbolos lógicos) y escenarios virtuales (cruce de avenidas, depósitos de agua) para interconectarlos con cables de colores.

Esta herramienta sirve para el aprendizaje experimental, inductivo y abstracto de los fundamentos del diseño digital en un ambiente tecnológico virtual que puede adaptarse para una metodología de aprendizaje basada en proyectos de ingeniería. Asimismo, el profesor puede preparar tutoriales en la medida y detalle que piense conveniente, y los alumnos pueden aprender con más ejemplos y retos de diseño.

La parte final menciona una metodología y un ejemplo de taller de diseño digital utilizando esta herramienta. Al finalizar el taller los alumnos están en capacidad de abordar el diseño de sistemas digitales completos desde una perspectiva del codiseño de hardware y software.



## Introducción

---

En agosto de 2006 apareció la primera versión gratuita, de libre distribución, de un programa para construir virtualmente circuitos digitales y simularlos con escenarios virtuales. Tal versión contaba con algunos tutoriales que demostraban el funcionamiento básico de circuitos típicos. El programa fue mejorado y llegó hasta la versión 0.9.5 (julio de 2008). A finales de 2010 se desarrolló una versión con varios protoboards y edición mejorada de las interconexiones. La versión final es 0.9.7a (julio de 2011) mas no cuenta con escenarios ni con tutoriales. También fue una versión gratuita y de libre distribución. Ambos programas pueden descargarse desde un enlace en la página [www.tourdigital.net](http://www.tourdigital.net).

Este documento describe la versión 0.9.8 del programa, el cual está especialmente diseñado para servir de apoyo en un curso de introducción al diseño lógico con circuitos digitales, tanto en el nivel escolar como en las primeras clases de un curso en un instituto o universidad. Con la experiencia adquirida a lo largo de muchos años de ensayos, aciertos, errores y equivocaciones, se ha creado una herramienta que puede ayudar a los docentes y alumnos en su primer contacto con los circuitos digitales. Esta nueva versión, al momento de redactar este documento, no es gratuita ni de libre distribución. Por más detalles e información escriba al correo electrónico [amiguel@pucp.edu.pe](mailto:amiguel@pucp.edu.pe).

A continuación se describen las características del software y sus aplicaciones para la enseñanza y aprendizaje del diseño lógico. Este documento es preliminar y todas las sugerencias, críticas, observaciones, reportes de errores, etc. son bienvenidos.

1. Circuitos Integrados TTL y CMOS
  2. Tableros de Entrada y Salida
  3. Diagramas Eléctricos y Lógicos
  4. Interconexiones con Cables
  5. Osciloscopio
  6. Circuitos de Lógica Programable: PLA y PLD
  7. Símbolos Lógicos
  8. Funciones Lógicas con Diagramas de Flujo
  9. Escenario para Experimentos de Cinemática (*en preparación*)
  10. Escenario de Tanque de Agua
  11. Escenario de Tanque de Agua con Cisterna
  12. Escenario para Control de Tráfico en un Cruce de Avenidas
  13. Aprendizaje Basado en Proyectos de Ingeniería
  14. Ampliaciones Futuras
  15. Referencias
- Apéndice A. Menú de Circuitos Integrados TTL
- Apéndice B. Construcción y Simulación de un Circuito con 7400
- Apéndice C. Análisis del Flip-flop JK
- Apéndice D. Contador BCD de cuatro dígitos con selector de frecuencia

## 1. Circuitos Integrados TTL y CMOS

El programa provee más de 80 modelos lógicos de circuitos digitales de la familia TTL LS. Estos modelos no consideran ningún aspecto físico (eléctrico, mecánico, térmico, cuántico, etc.) presente en los circuitos reales. Tampoco se han incluido modelos con salidas de tres estados, colector abierto, ni de pines bidireccionales, a excepción del 7447 donde el pin bidireccional está modelado únicamente como salida. El apéndice A muestra todos los modelos de chips disponibles en este programa.

El programa también contiene un pequeño conjunto de modelos de circuitos CMOS de la serie 4000. Actualmente estos modelos están en la fase de depuración y extensión.

## 2. Tableros de Entrada y Salida

Los circuitos integrados se colocan sobre protoboards para interconectarlos entre ellos y con los tableros y demás escenarios del programa. Las entradas lógicas pueden provenir de tableros de interruptores, pulsadores y temporizadores, mientras que los estados lógicos pueden ser observados en tableros de leds y de visualizadores de siete segmentos. Los tableros de entradas y de salidas deben conectarse apropiadamente con VCC y GND (figura 1). Ellos no funcionan cuando están conectados incorrectamente. Cada tablero tiene unas marcas que sirven para trasladarlos presionando con el botón izquierdo del ratón o para eliminarlos pulsando con el botón derecho del ratón. Para evitar conexiones incorrectas deben evitarse las superposiciones de tableros.

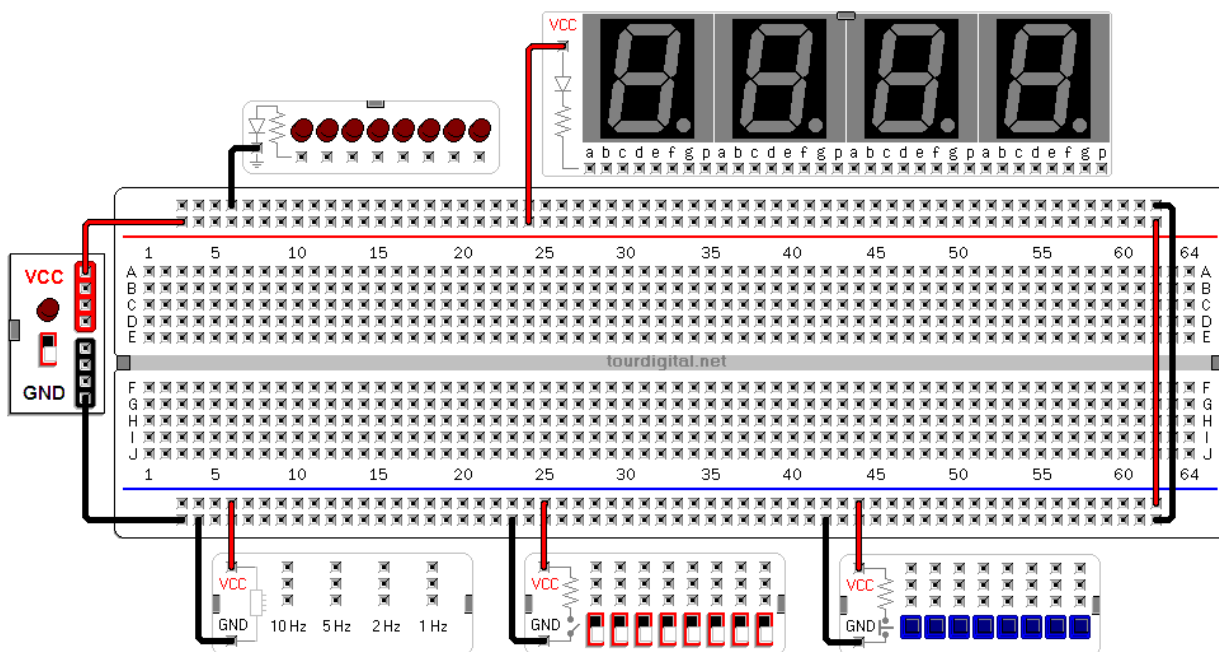


Figura 1. Tableros de entrada y salida.

En el protoboard se insertan circuitos integrados y otros componentes para interconectarlos mediante cables. Los agujeros donde se insertan los cables se llaman casillas o puertos.

El tablero de fuente DC simula la potencia eléctrica necesaria para el funcionamiento de los demás tableros y de los componentes del circuito. Este tablero aparece en todos los circuitos. El programa evita ediciones o modificaciones del circuito cuando el tablero de fuente DC está encendido.

El tablero de interruptores provee ocho interruptores cuyos estados pueden ser conmutados al pulsar sobre ellos con el botón izquierdo del ratón. Con la pestaña hacia abajo los interruptores proveen una señal lógica baja (0); con la pestaña hacia arriba proveen una señal lógica alta (1). Similarmente, el tablero de pulsadores tiene ocho pulsadores activos en alta. Cuando no están presionados la salida es baja. Solamente un pulsador puede activarse en un momento dado.

El tablero de leds contiene ocho leds activos en alta. El color de los leds puede cambiarse pulsando sobre ellos (los colores actuales son rojo, azul, verde y amarillo). El tablero de visualizadores de siete segmentos contiene cuatro visualizadores de ánodo común.

El tablero de temporizadores provee cuatro señales periódicas de frecuencias aproximadas a 10 Hz, 5 Hz, 2 Hz y 1 Hz. Solamente existe un tablero temporizador por cada circuito.

### 3. Diagramas Eléctricos y Lógicos

La figura 2 muestra un circuito del programa, el diagrama eléctrico del circuito de prueba y un circuito típico real usando una puerta NAND. En el diagrama eléctrico los números 13, 12 y 11 identifican a los pines utilizados en el circuito integrado 7400. En el circuito real las resistencias conectadas a VCC son de  $1K\ \Omega$  y la conectada en serie con el led es de  $330\ \Omega$ .

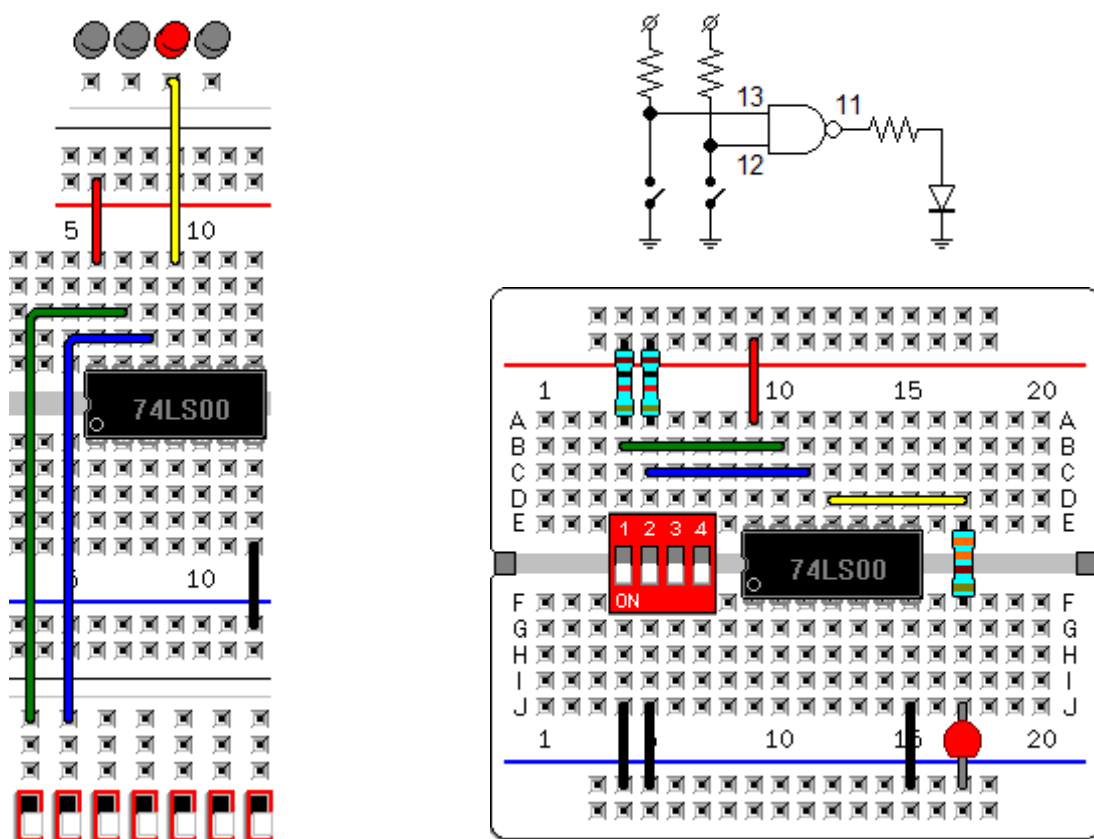



Figura 2. Equivalencia entre circuitos del programa y circuitos reales típicos.

## 4. Interconexiones con Cables

Para unir dos casillas con un cable, se lleva el cursor del ratón sobre una de las casillas, se presiona el botón izquierdo, se arrastra el cursor hasta la otra casilla y se libera el botón. Para ajustar las conexiones se presiona sobre el cable y se arrastra un segmento. Los cables pueden tener hasta cinco segmentos, de los cuales hasta tres pueden ser modificados. El programa evita conexiones entre casillas del mismo nodo eléctrico. También evita cortocircuitos entre VCC, GND y puertos de salida.

La figura 3 muestra una secuencia de edición de un cable. En (a) se ha formado un cable conectando las casillas I-2 y B-9. La dirección original del primer segmento queda determinada por el sentido del movimiento inicial del puntero del ratón. En (b) el segmento horizontal del cable fue presionado y arrastrado hacia abajo. En (c) el segmento vertical inferior fue presionado y arrastrado hacia la derecha. Similarmente, en (d) el segmento vertical superior ha sido desplazado hacia la derecha. Los segmentos horizontales superior e inferior de (d) no pueden ser arrastrados, mientras que los otros tres siguen siendo editables. Para eliminar un cable se pulsa el botón derecho del ratón sobre uno de los segmentos editables.

Un color de cable puede elegirse pulsando en el botón  de la barra de herramientas. Los colores de los cables dibujados ya no pueden ser modificados.

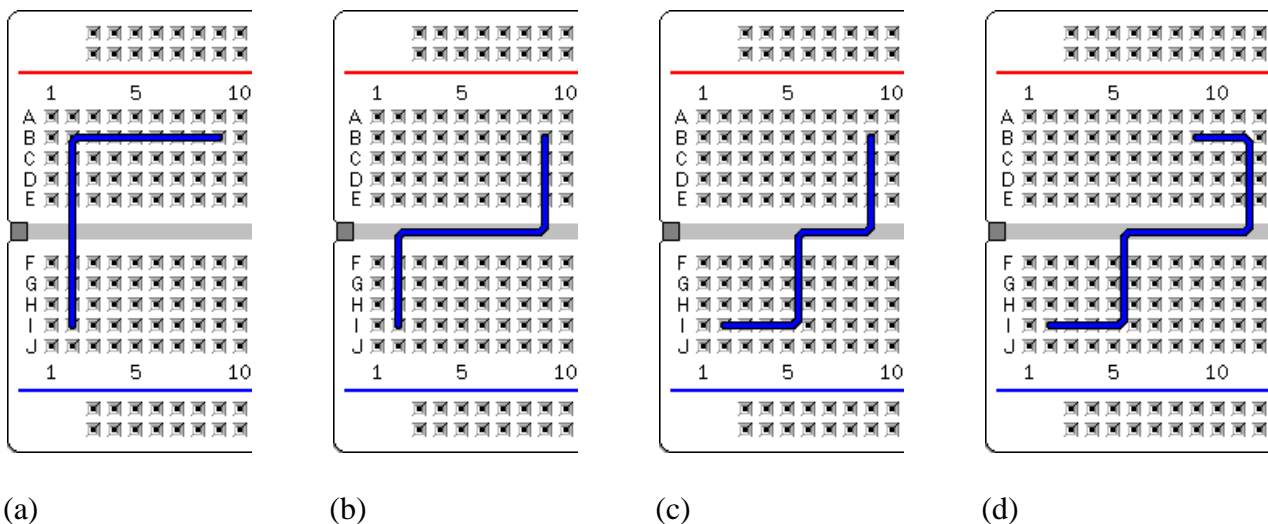


Figura 3. Secuencias en la formación de un cable.

## 5. Osciloscopio

El osciloscopio (figura 4) consta de nueve canales y puede mostrar 40 datos a la vez en cada canal. Tiene dos modos de captura: síncrono y asíncrono. En el modo asíncrono los nueve canales se actualizan cuando hay un cambio de valor en cualquier canal. En el modo síncrono los canales se actualizan cuando ocurre un cambio en el canal extremo inferior. Con **Reinicio** se vacía la memoria de datos. Con **Pausa** se suspende la captura de datos hasta que vuelva a conmutarse. Las flechas permiten avanzar o retroceder para explorar los datos almacenados (paso, página de 40 pasos, extremo).

La figura 5 muestra un ejemplo de uso del osciloscopio en modo síncrono. Note que los valores mostrados justo después de la señal de sincronía corresponden a los valores actuales de las señales en el circuito.

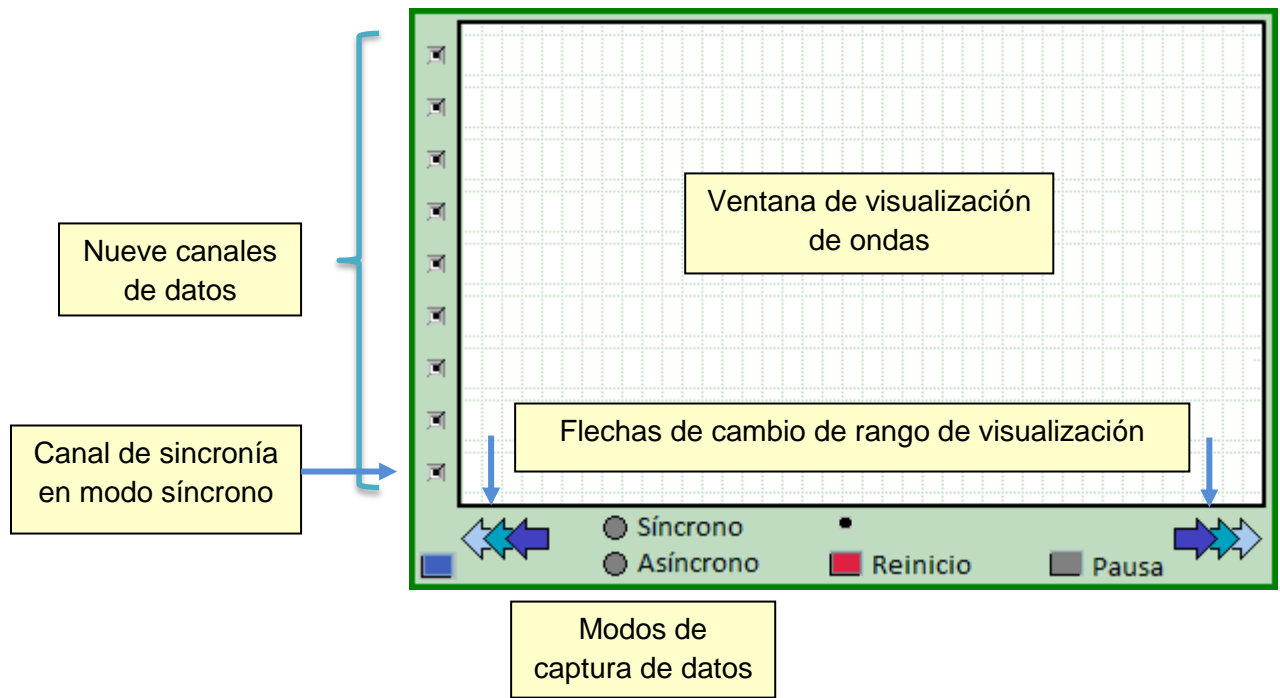


Figura 4. Interfaz del osciloscopio digital.

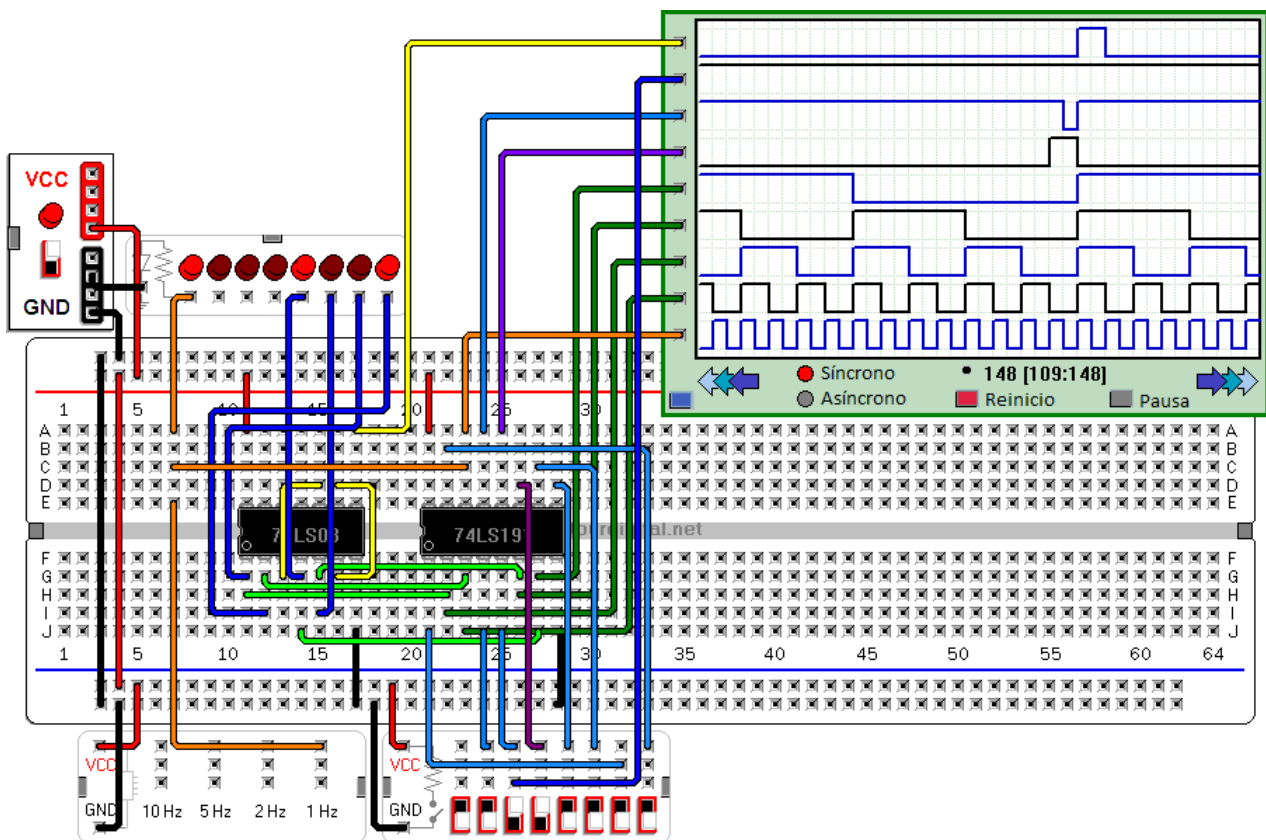


Figura 5. Prueba de un contador 74191 con detección de cuenta en 15.

## 6. Circuitos de Lógica Programable: PLA y PLD

La figura 6 muestra el escenario de un PLA (arreglo de lógica programable) de cuatro entradas, cuatro salidas y ocho productos. Este PLA contiene dos planos programables AND y OR para la formación de funciones lógicas como sumas de productos. Para unir y separar líneas se pulsa sobre las intersecciones entre las líneas de los planos programables.

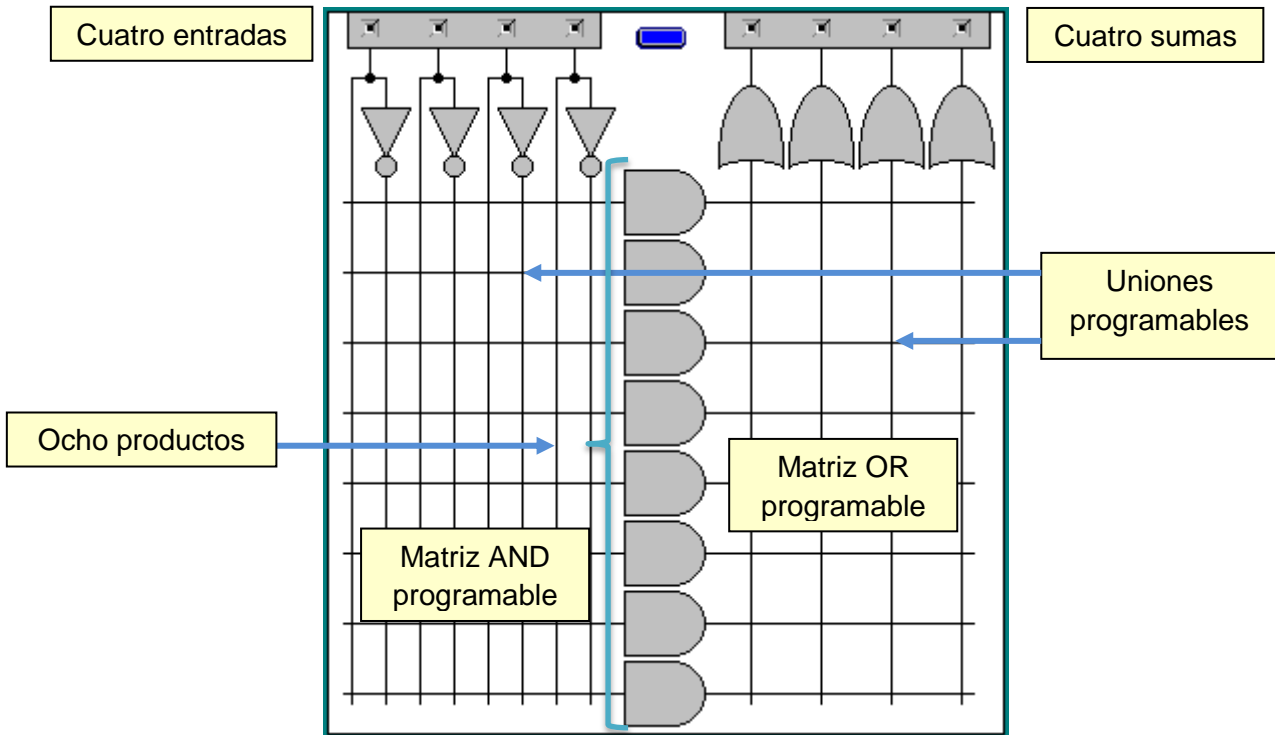


Figura 6. Elementos del circuito PLA.

La figura 7 muestra la configuración de un latch SR con puertas NAND de dos entradas. Cada NAND se configura en la PLA utilizando el teorema de De Morgan  $(ab)' = a' + b'$ . Note que se utilizan dos sumas extras solo para mostrar las salidas directamente en el osciloscopio (configurado en modo asíncrono). Observe también el retardo de las señales. En este programa todos los retardos de entrada a salida de los escenarios son de 50 ms.

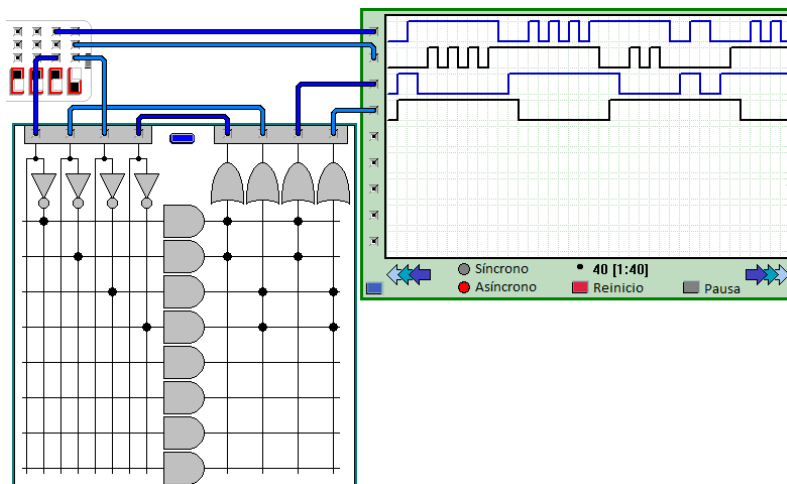


Figura 7. Pruebas del PLA configurado como latch SR.

La figura 8 muestra el escenario de un PLD (dispositivo de lógica programable) que consta de cuatro entradas que alimentan a una matriz programable AND y dos entradas (clk, arst) que se conectan a cuatro registros de tipo D. Los productos alimentan se suman predeterminadamente. Las salidas de las puertas OR van a las entradas de los flip-flops D y a los multiplexores. Las salidas del PLD pueden seleccionarse entre las salidas de las puertas OR y las salidas de los flip-flops.

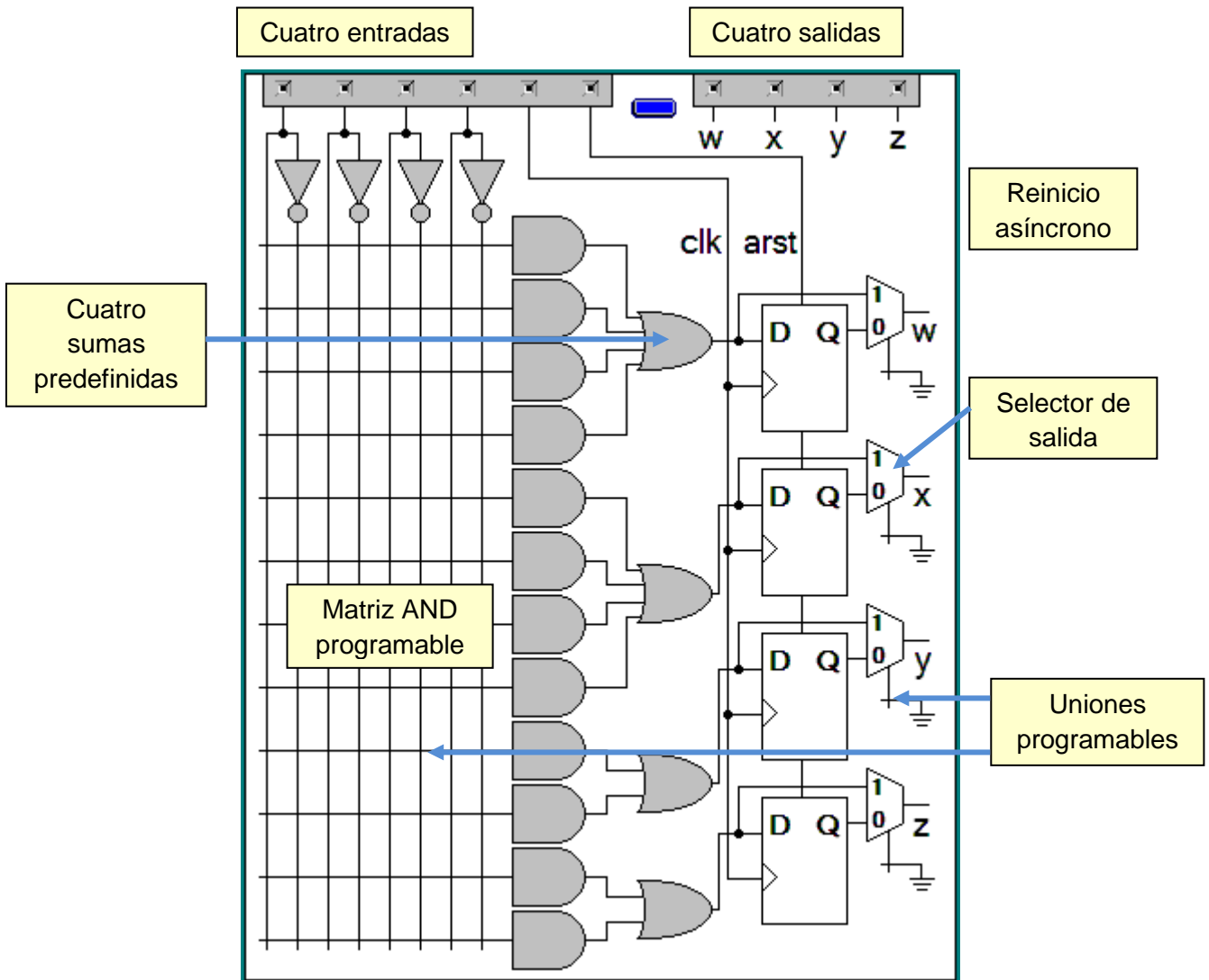


Figura 8. Elementos del circuito PLD.

Tanto el PLA como el PLD son elementos didácticos muy importantes. Por ejemplo, pueden servir para demostrar las equivalencias de los postulados de Huntington del álgebra de conmutación, para comprobar los teoremas de simplificación de funciones y para realizar bloques lógicos típicos, tales como decodificadores, selectores, codificadores, sumadores, comparadores, contadores, registros de desplazamiento, etc. También pueden servir para evaluar funciones combinacionales y de máquinas de estados antes de construirlas con circuitos integrados sobre un protoboard. Asimismo, este primer contacto con los PLD prepara a los estudiantes para abordar el diseño de circuitos digitales empleando las modernas tecnologías de lógica programable.

Como ejemplos, la figura 9 ilustra el circuito de un contador ascendente de cuatro bits y la figura 10 un registro de desplazamiento de cuatro bits. Note que los productos innecesarios deben hacerse 0 y para ello se aplica la propiedad  $a \cdot a' = 0$ .



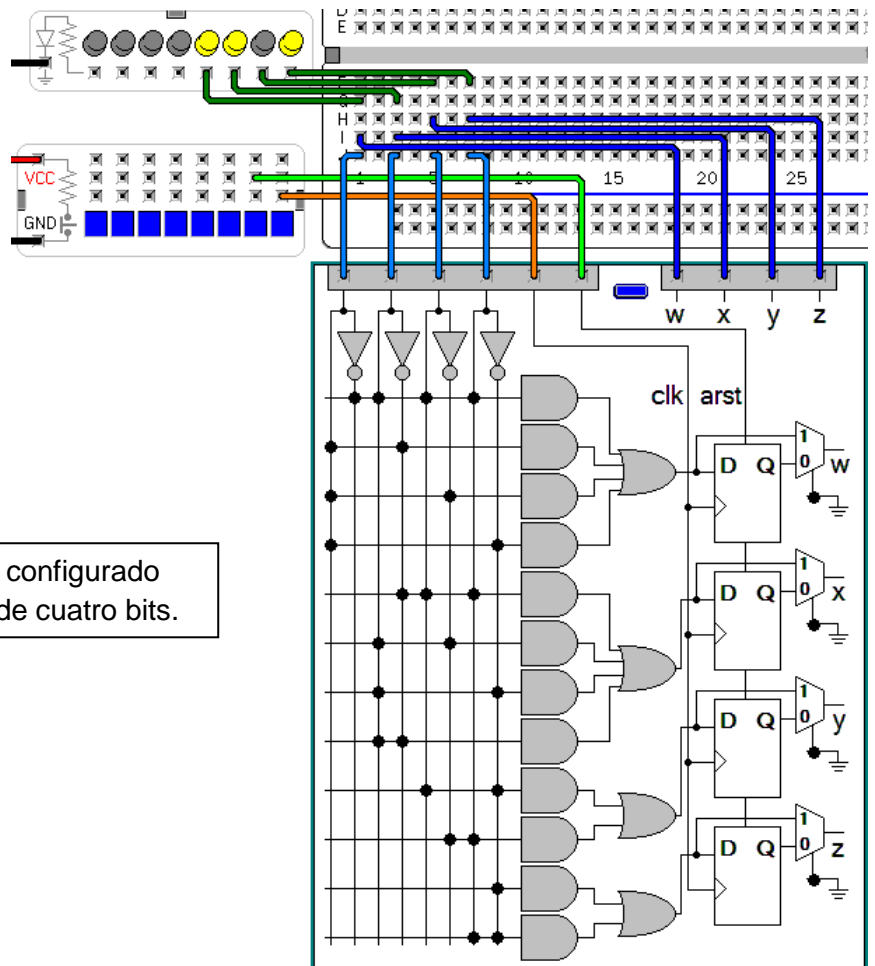


Figura 9. Pruebas del PLD configurado como contador ascendente de cuatro bits.

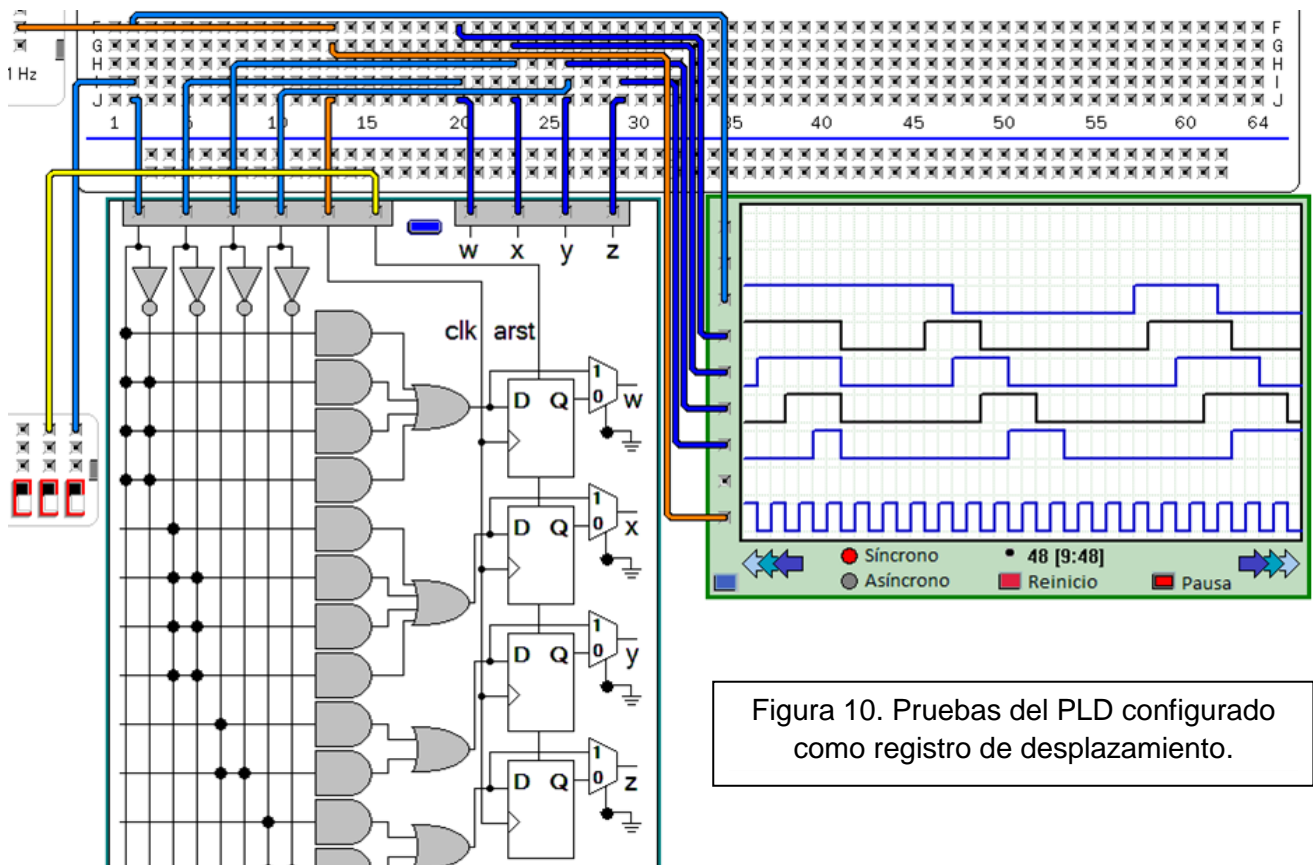


Figura 10. Pruebas del PLD configurado como registro de desplazamiento.

## 7. Símbolos Lógicos

El programa provee varios tipos de símbolos lógicos. Ellos pueden ser utilizados en combinación con los circuitos integrados y demás elementos del programa. La figura 11 muestra entradas y sensores lógicos de varios tipos. Las entradas lógicas proveen los estados lógicos (0, 1) y los sensores visualizan los estados lógicos de los nodos del circuito.

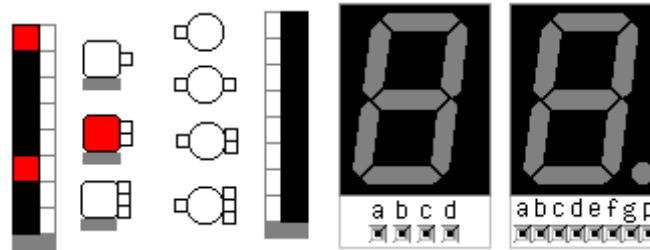


Figura 11. Entradas y sensores lógicos.

Existen siete símbolos lógicos correspondientes a las funciones lógicas AND, OR, NOT, NAND, NOR, XOR y XNOR. Salvo la función NOT, todas las funciones son de dos entradas. La figura 12 muestra un flip-flop D maestro esclavo hecho de puertas NAND. Observe cómo los sensores lógicos permiten analizar con detalle el funcionamiento de este circuito.

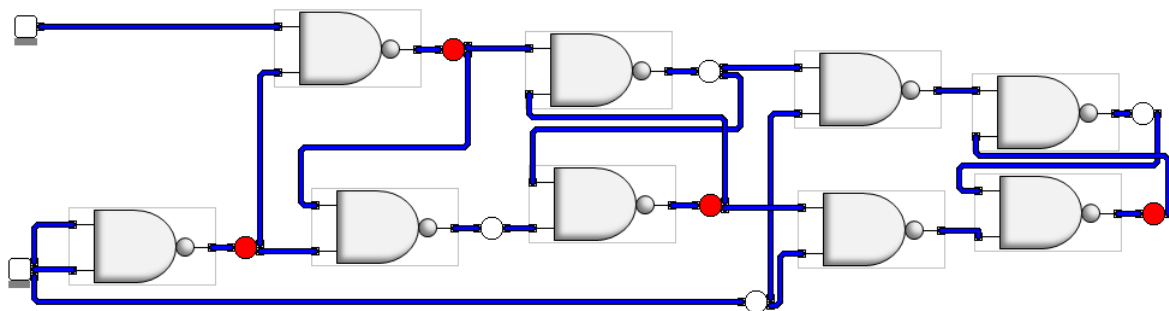


Figura 12. Flip-flop D hecho con puertas NAND.

El programa trata a los símbolos lógicos como a los demás escenarios. Esto significa que existe un retardo de propagación que puede llegar a notarse visualmente en los sensores lógicos y en el osciloscopio. Esta característica ayuda a entender el diseño de los latches y de flipflops así como de circuitos asíncronos.

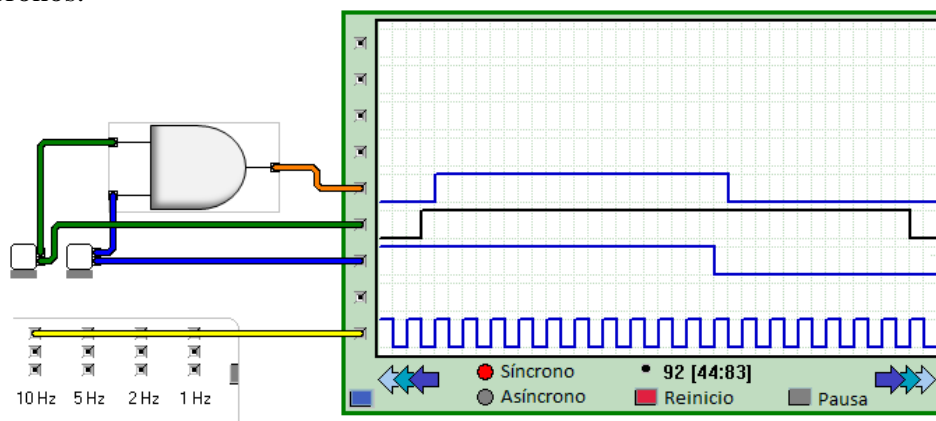


Figura 13. Respuesta temporal del símbolo lógico AND.

La figura 14 muestra cuatro símbolos programables. De izquierda a derecha aparecen una OR de ocho entradas, una NAND de ocho entradas, un selector de 8 a 1 y un decodificador 1 de 8. Estos símbolos ayudan a aprender el concepto de funcionalidad universal y a realizar prototipos de funciones lógicas rápidamente.

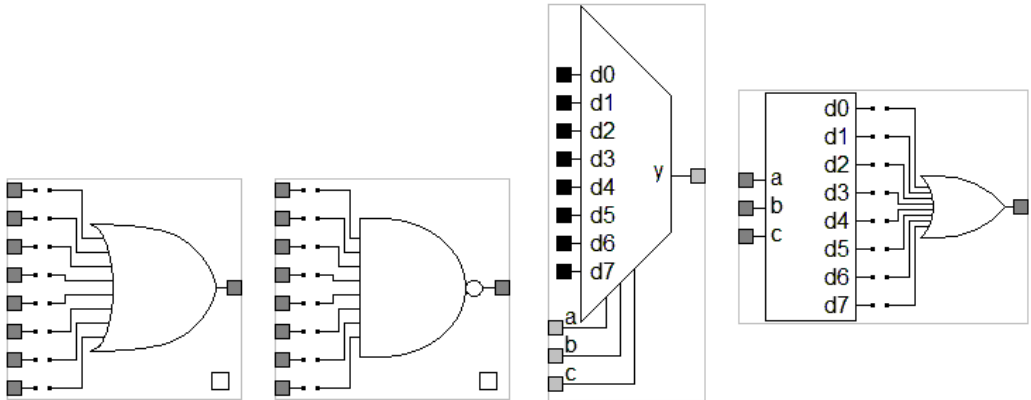


Figura 14. Símbolos programables.

Existen siete símbolos de funciones de complejidad MSI mostrados en la figura 15. Ellos son: multiplexor de 8 a 1, decodificadores 1 de 8 (en versiones con salidas en baja y en alta), registro D de 4 bits, contador de 4 bits, registro universal de 4 bits, sumador de 4 bits y comparador de 4 bits. La figura 16 ilustra un registro universal configurado para contar en anillo.

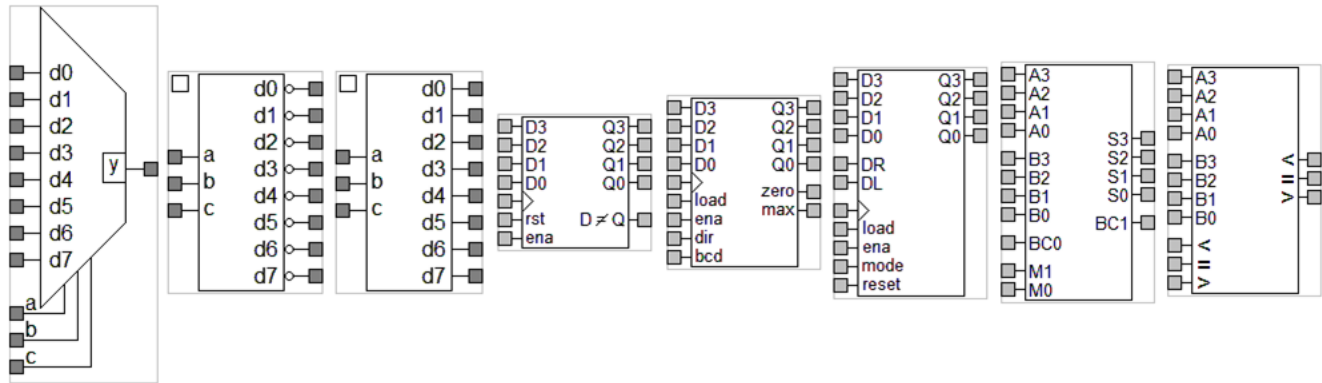


Figura 15. Símbolos MSI.

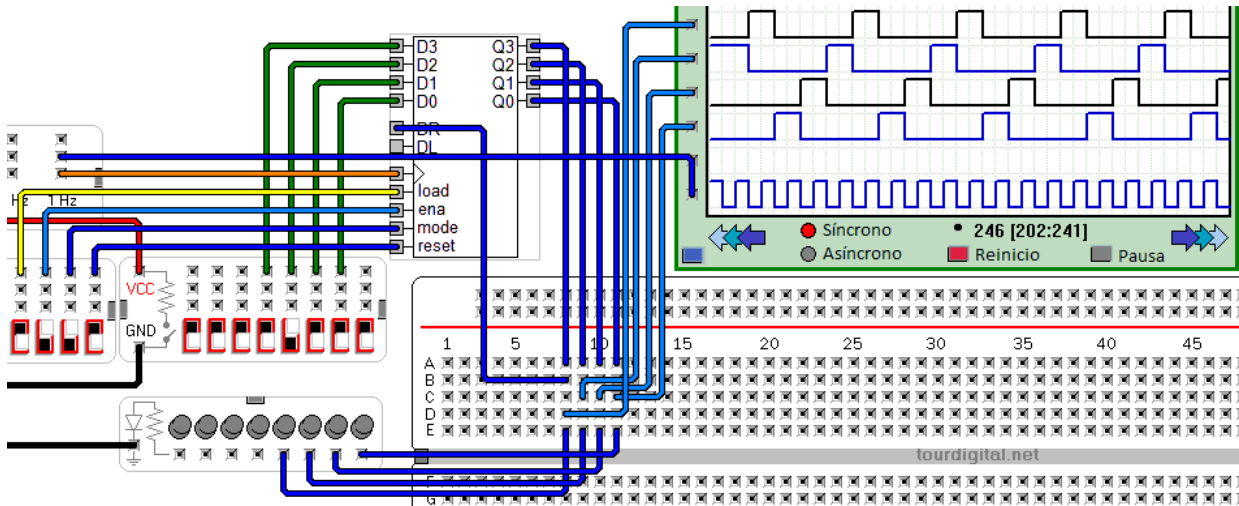


Figura 16. Registro universal configurado como contador en anillo.

## 8. Funciones Lógicas con Diagramas de Flujo

Las figuras 17 y 18 muestran dos diagramas de flujo para describir funciones lógicas. En la casilla de decisión se evalúa una afirmación. Dependiendo del valor de verdad del resultado se efectúa la asignación de la casilla correspondiente. En estos diagramas el operador de la condición puede elegirse entre = (igualdad), + (OR), y \* (AND). Por ejemplo, la expresión  $A.B$  significa A es 1 y B es 1;  $A + B$  significa A es 1 ó B es 1; y  $A = B$  significa A y B son iguales. Los valores de cada operando pueden ser 0, 1, A, A', B y B'. Para cambiar un operador o un operando se pulsa sobre ellos repetidamente. Si la afirmación es cierta entonces se ejecuta la asignación de la rama Sí, caso contrario se realiza la asignación de la rama No. Este tipo de diagramas ayuda a aprender las descripciones algorítmicas de circuitos y facilita la transición hacia las descripciones de hardware de alto nivel como VHDL y Verilog.

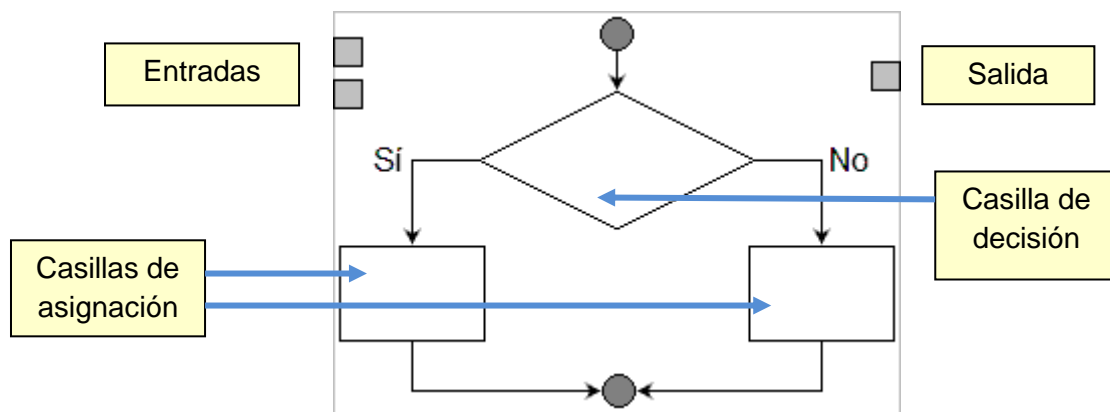


Figura 17. Diagrama de flujo de tipo IF-ELSE.

*En preparación: el diagrama servirá para describir un latch D y un flip-flop D.*

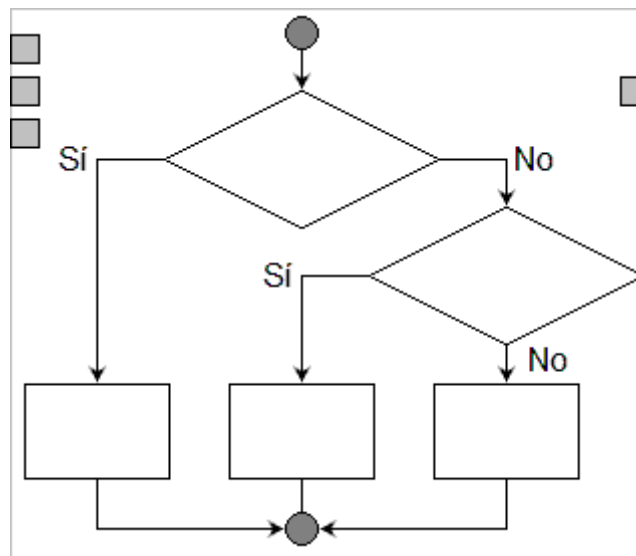


Figura 18. Diagrama de flujo de tipo IF-ELSIF-ELSE.

*En preparación: el diagrama servirá para describir varios tipos de latches y flip-flops.*

En la figura 19 se emplean tres diagramas de flujo individuales para realizar la función  $f = as' + bs$ . Note que se han empleado propiedades del álgebra de conmutación para expresar las funciones intermedias de varias maneras.

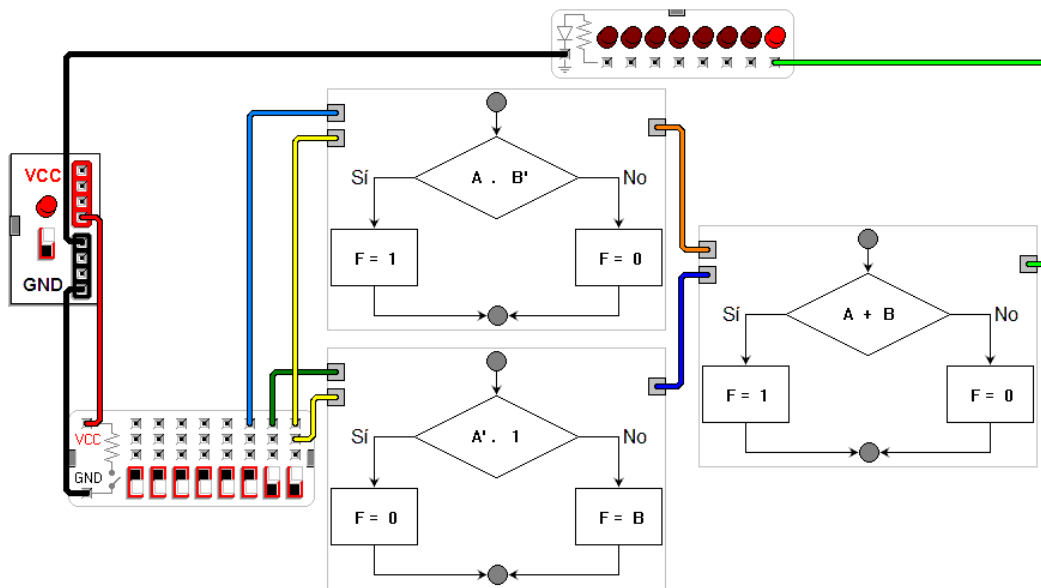


Figura 19. Multiplexor de 2 a 1 utilizando tres diagramas de flujo.

## 9. Escenario para Experimentos de Cinemática *(en preparación)*

Este escenario sirve para realizar experimentos de movimiento en caída libre, movimiento semiparabólico y movimiento parabólico. Es excelente para aprender el método científico tomando como caso de estudio el movimiento de proyectiles bajo el efecto de la aceleración de la gravedad. El experimento inicia cuando la entrada verde va de 0 a 1. Si esta entrada vuelve a cero el proyectil retorna a la posición de disparo inicial. El sensor s2 se posiciona automáticamente en el lugar de impacto. El botón superior derecho sirve para borrar las trayectorias del proyectil.

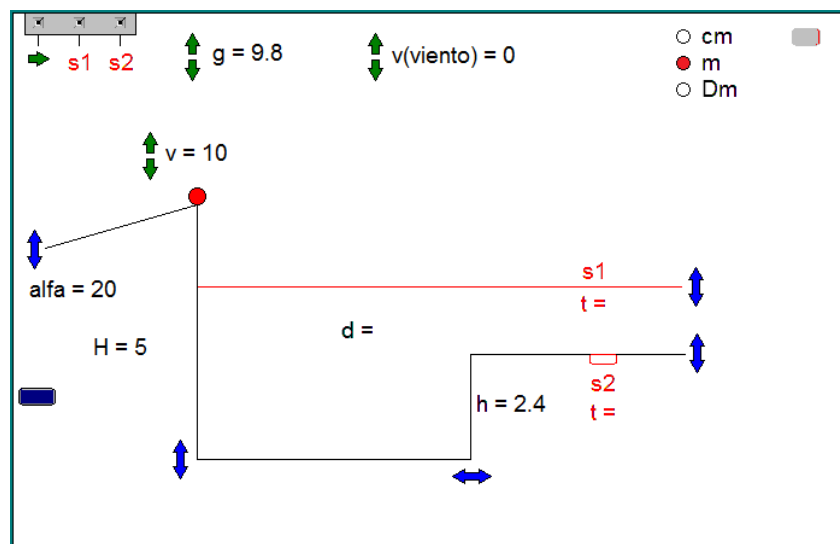


Figura 20. Experimento virtual de cinemática.

## 10. Escenario de Tanque de Agua

Este escenario provee ocho sensores del nivel de agua en un tanque; un sensor de flujo de agua por la tubería de ingreso; una llave para desalojar el agua del tanque; y una bomba para el suministro de agua (figura 21). Tanto la llave como la bomba pueden ser accionadas por el usuario y por las señales externas. El usuario puede cortar el suministro de agua al tanque para simular escasez de agua. El sensor de flujo permite optimizar el funcionamiento de la bomba.

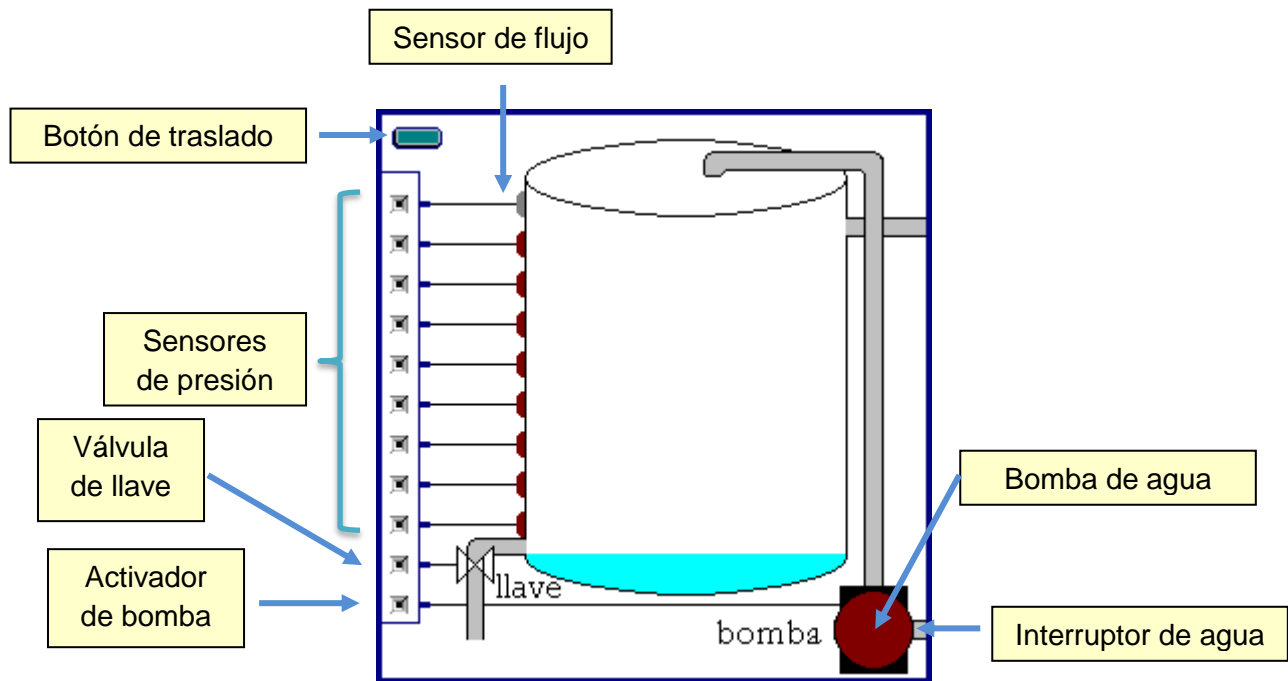


Figura 21. Elementos del tanque de agua.

El circuito de la figura 22 mide el nivel de agua en el tanque y maneja el estado de la bomba.

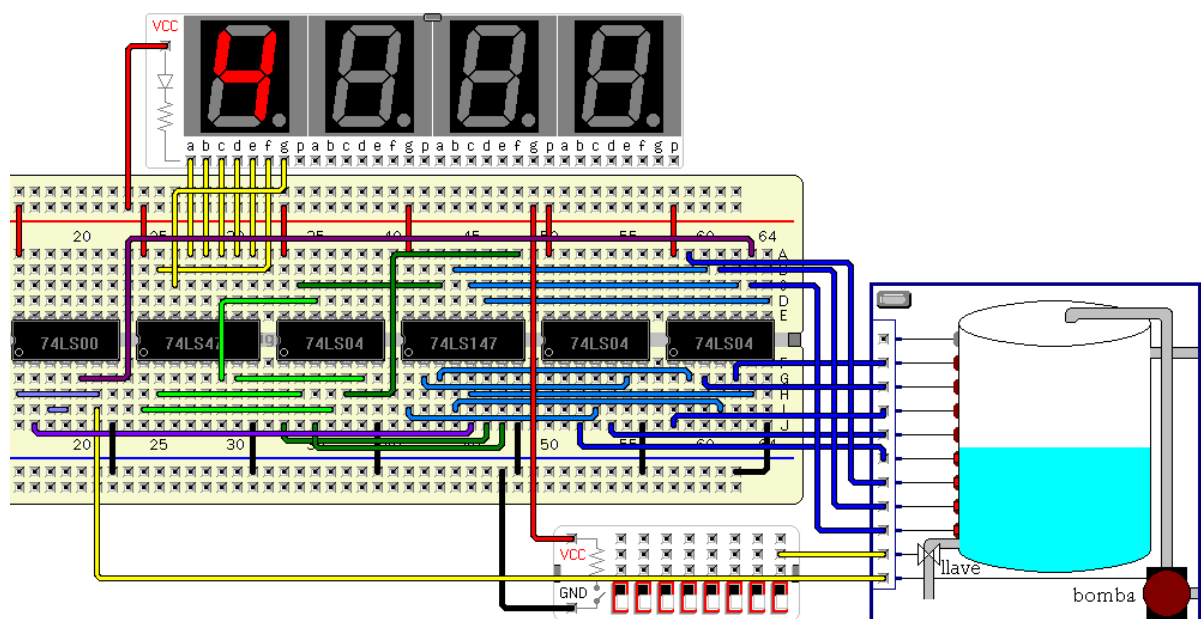


Figura 22. Control automático del nivel de agua en un tanque.

## 11. Escenario de Tanque de Agua con Cisterna

El escenario de tanque y cisterna contiene dos depósitos de agua donde una cisterna provee agua para un tanque. Como ejemplo de diseño práctico y de introducción al diseño en ingeniería, los estudiantes pueden diseñar un circuito para medir el flujo de agua en cada tanque a lo largo de un periodo para estimar el consumo y las necesidades de agua a mediano plazo. Para resolver este problema, el estudiante deberá utilizar contadores, comparadores, sumadores, latches, temporizadores, flip-flops, registros y también puede utilizar máquinas de estados para una solución más sofisticada.

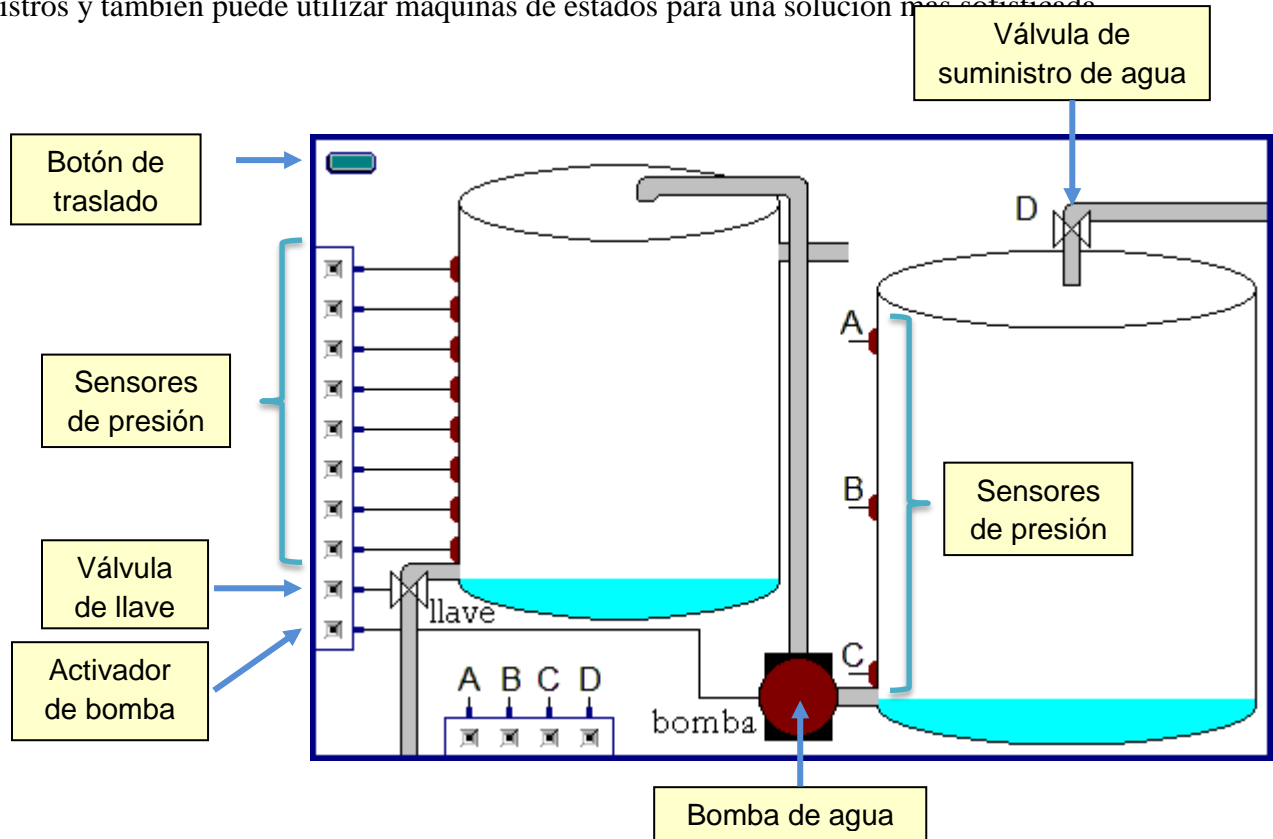


Figura 23. Elementos del tanque de agua con cisterna.

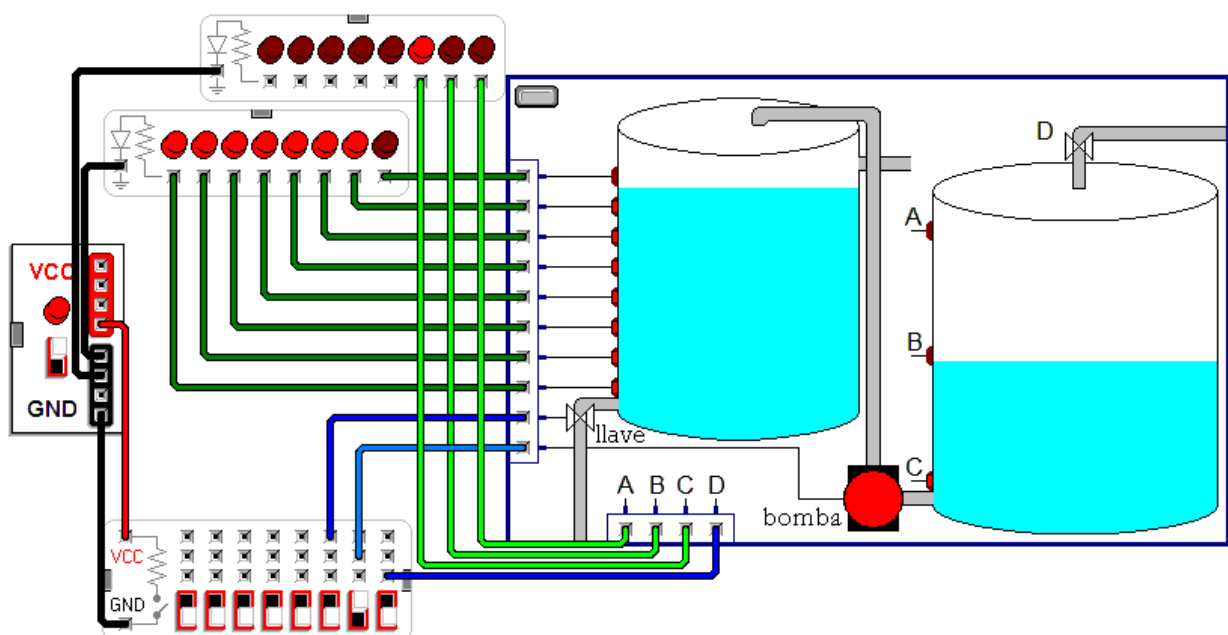


Figura 24. Control manual del escenario de tanque de agua con cisterna.

## 12. Escenario para control de tráfico en un cruce de avenidas

Este escenario provee cuatro sensores de paso y cuatro semáforos individuales de cuatro luces. Los sensores se activan cuando un vehículo pasa sobre ellos o cuando el usuario pulsa sobre las flechas azules. Los autos del escenario aparecen cíclicamente, obedecen solamente a la luz verde principal y siguen trayectorias rectilíneas. La figura 25 muestra un control sencillo de las luces de los semáforos.

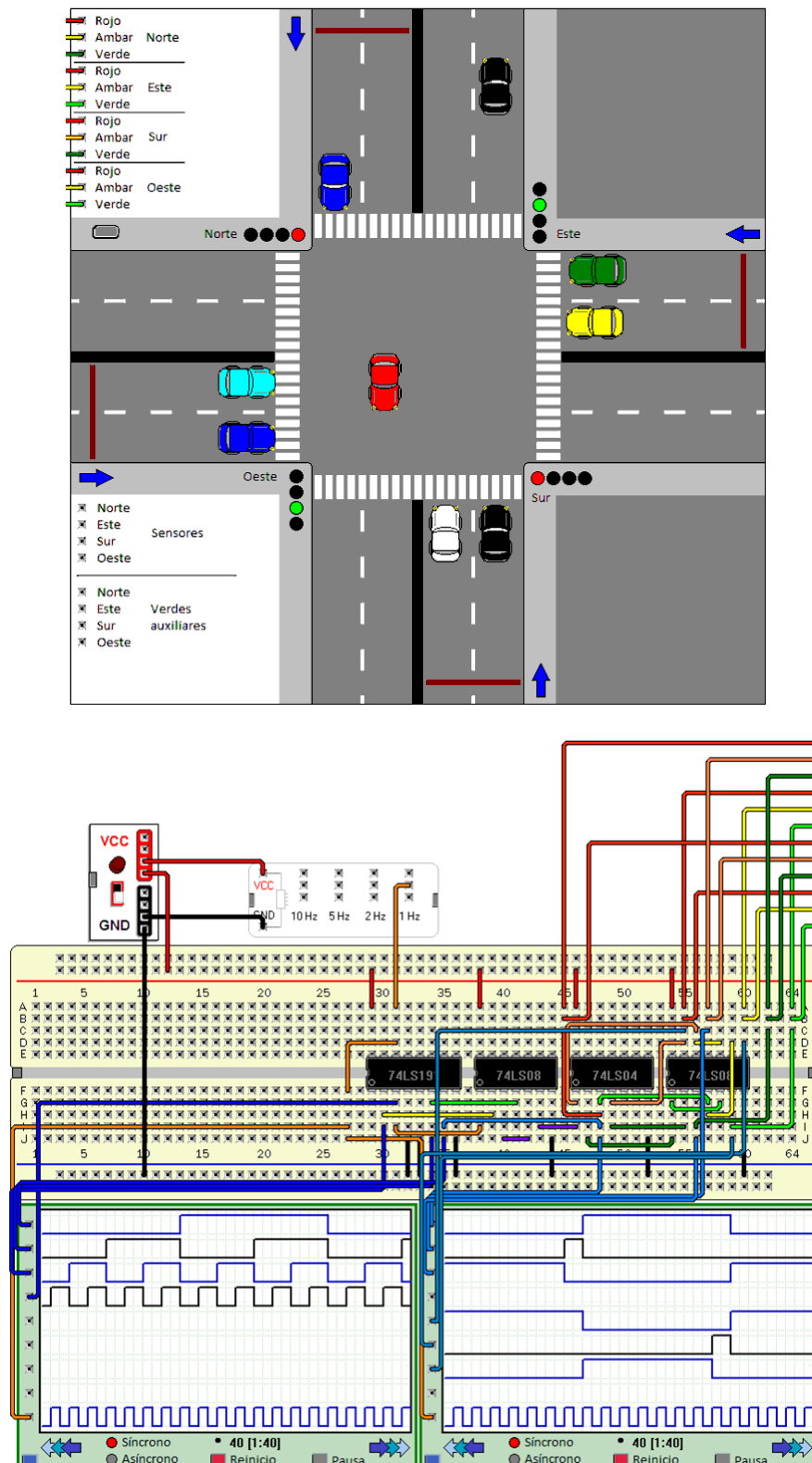


Figura 25. Prueba del escenario de semáforos en un cruce de avenidas.



### 13. Aprendizaje Basado en Proyectos de Ingeniería

---

El mundo tecnológico actual, dominado por los computadores digitales, requiere que los estudiantes de ingeniería posean “un conocimiento práctico de los fundamentos de los sistemas digitales y fluidez en el empleo de sistemas y herramientas de cómputo modernas” [1]. En general, existen muchos programas exitosos de educación en ingeniería [2] y conocimientos científicos de cómo aprendemos [3] que pueden ayudar a mejorar los sistemas de educación escolar [4] y universitario, especialmente en las disciplinas STEM (acrónimo formado por las iniciales en inglés de ciencia, tecnología, ingeniería y matemática).

Las implicaciones para la enseñanza sobre lo que sabemos acerca de cómo aprendemos [3] son: conocer los preconceptos de los estudiantes; enseñar algunas materias en profundidad, mencionando los fundamentos científicos y con muchos ejemplos de aplicación; integrar el desarrollo de habilidades metacognitivas en varias áreas del programa curricular. La misma referencia destaca que un ambiente de enseñanza debe centrar la instrucción en los estudiantes; atender a su contenido, razón y profundidad; evaluar formativamente; y tener en cuenta el lugar del aprendizaje. Indica además, que la transferencia debe estar sostenida por aprendizajes previos, ser activa y dinámica, para generar representaciones abstractas del conocimiento contextualizado.

En correspondencia con estas recomendaciones, se ha diseñado y ensayado en varias ocasiones talleres de diseño digital y programación con una aproximación basada en proyectos de ingeniería y alineamiento constructivo. En las últimas oportunidades se han considerado la indagación científica, el método de instrucción 5E y los estilos de aprendizaje junto a varios otros métodos recomendados y aprendidos de la literatura y de vídeos de universidades norteamericanas, europeas y asiáticas.

Sobre el contexto de una metodología de diseño basada en proyectos, un uso recomendado de este programa es comenzar con aprendizaje inductivo y experimentación científica para descubrir el funcionamiento de las puertas lógicas básicas y combinaciones sencillas de ellas. Es una aproximación adaptada del programa Diseño y Descubrimiento de Intel. Este método es diferente al método clásico de describir para el estudiante las propiedades de los circuitos. Además, en cuanto al orden de los temas, todos los circuitos SSI son estudiados antes que los circuitos MSI, es decir, los latches y flip-flops se descubren antes que los sumadores y multiplexores. Asimismo, los contadores se estudian antes que los sumadores. Algunos ejemplos básicos se ilustran en los apéndices B, C y D.

En un primer módulo de contacto con los circuitos digitales, los estudiantes analizan las propiedades y características del circuito integrado 7400 y las representan mediante varios algoritmos, expresiones lógicas, tablas, cronogramas y diagramas. Los estudiantes utilizan las propiedades de la función lógica NAND para descubrir nuevas funciones y aprenden a usar chips dedicados a esas funciones. Paralelamente, el profesor puede crear tutoriales y ajustarlos a los estilos de aprendizaje de los estudiantes. Por ejemplo, con un estudiante de pensamiento global puede mostrarle varias representaciones para clasificar, y con un estudiante de aprendizaje serial puede mostrarle gradualmente los tipos de representaciones desde niveles concretos (circuitos integrados) a abstractos (símbolos lógicos).

Como ejemplo de instrucción, la figura 26 muestra varias representaciones de la función lógica NAND de dos entradas y circuitos típicos para demostrar su funcionalidad universal. Así, el programa permite aprender varias formas de hacer lo mismo y entender con profundidad los conceptos fundamentales del diseño lógico. En el mismo sentido, este tutorial en particular cubre seis de los siete niveles de la taxonomía de Bloom: el estudiante **conoce** la función lógica NAND, **entiende** su funcionamiento, **aplica** sus conceptos, **analiza** circuitos compuestos, **evalúa** desempeños y **valora** su aprendizaje. Únicamente el nivel de **síntesis** no se cubre en este tutorial. Para completar este nivel se han proporcionado escenarios, como el de tanque y cisterna de agua, que el estudiante puede analizar para crear y evaluar soluciones y diseños.

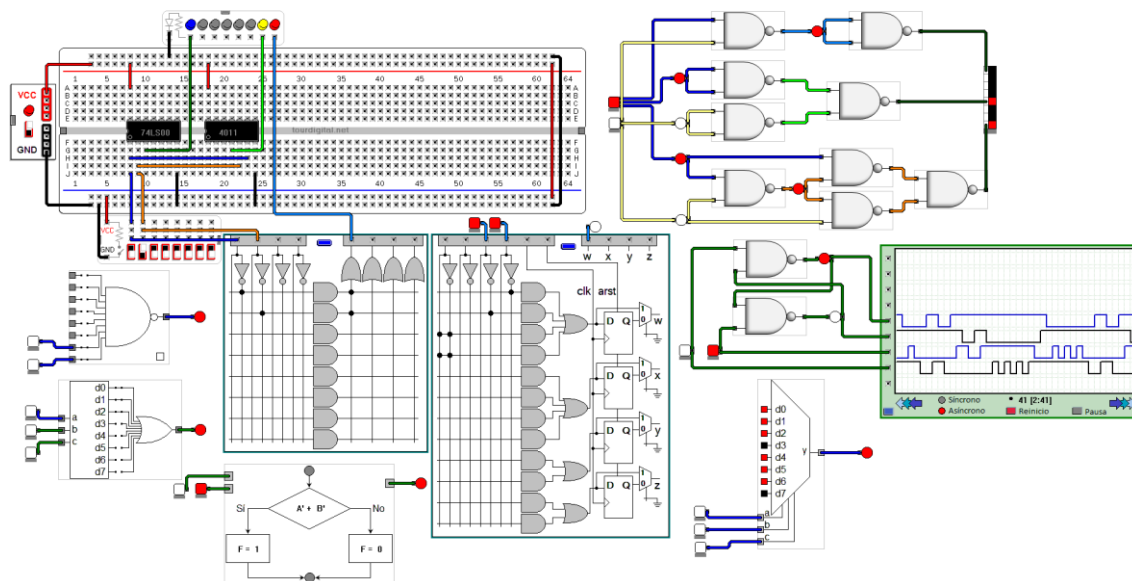


Figura 26. Tutorial de puertas NAND.

Un segundo módulo presenta el álgebra de conmutación. Los estudiantes conocen las técnicas tradicionales de transformaciones y síntesis manual de funciones booleanas, pero se presta más atención a los varios tipos de circuitos MSI con el fin de que diseñen funciones lógicas modularmente.

En el tercer módulo los estudiantes aprenden a describir circuitos algorítmicamente. Ellos realizan descripciones VHDL que son sintetizadas con Quartus II [5] para síntesis sobre chips de lógica programable y crean programas orientados a objetos en C++ utilizando el compilador Dev-C++ [6].

En un cuarto módulo, los estudiantes diseñan microprocesadores y microcontroladores a partir de secuenciadores hechos con máquinas de estados finitos y máquinas de estados algorítmicas. Los estudiantes aprenden a diseñar la unidad de control y el camino de datos extrayendo información de las sentencias lógicas y transformando algoritmos.

Al finalizar este taller los alumnos están en capacidad de abordar el diseño de sistemas digitales completos desde una perspectiva del codiseño de hardware y software.

## 14. Ampliaciones futuras

---

Actualmente se está desarrollando la biblioteca de componentes para chips CMOS y se están diseñando más escenarios: ascensor de cinco pisos, estacionamiento de vehículos, motores DC y motores a pasos. También se está culminando la programación del escenario para experimentos de cinemática y más tableros de especificaciones de circuitos (tablas de verdad, cronogramas, pseudocódigos). Más adelante se crearán interfaces con el puerto paralelo y el puerto serial y modelos de tarjetas de entrenamiento para lógica programable y microcontroladores.

## 15. Referencias

---

1. The Engineer of 2020: Visions of Engineering in the New Century. National Academy of Engineering, 2004, <http://www.nap.edu/catalog/10999.html>
2. Educating the Engineer of 2020: Adapting Engineering Education to the New Century. National Academy of Engineering, 2005, <http://www.nap.edu/catalog/11338.html>
3. How People Learn: Brain, Mind, Experience, and School: Expanded Edition. National Academy of Sciences, 2000, <http://www.nap.edu/catalog/9853.html>
4. OECD (2010), PISA 2009 at a Glance, OECD Publishing. <http://dx.doi.org/10.1787/9789264095298-en>
5. Quartus II (versión 9.1), 2011, <http://www.altera.com>
6. Dev-C++ (versión 0.9.7a), 2011, <http://www.bloodsheet.net>

## Apéndice A. Menú de Circuitos Integrados TTL

<b>And</b>	▶	7408 - 2 entradas (x4)	<b>Or</b>	▶	7432 - 2 entradas (x4)
Nand	▶	7411 - 3 entradas (x3)	..		
Not	▶	7421 - 4 entradas (x2)	<b>Nor</b>	▶	7402 - 2 entradas (x4)
<b>Nand</b>	▶	7400 - 2 entradas (x4)	Or	▶	7427 - 3 entradas(x3)
Not	▶	7410 - 3 entradas (x3)	Xor	▶	74260 - 5 entradas (x2)
Nor	▶	7420 - 4 entradas (x2)	<b>Not</b>	▶	7404 - Not (x6)
Or	▶	7430 - 8 entradas	<b>Xor</b>	▶	7486 - 2 entradas (x4)
Xor	▶	74133 - 13 entradas	And - Or - Invert	▶	74386 - 2 entradas (x4)
<b>And - Or - Invert</b>	▶	7451 - 2-ancho 2-3-entradas AND-OR-INVERT (x2)			
		7454 - 3-2-2-3-entradas AND-OR-INVERT			
		7455 - 2-ancho, 4-entradas AND-OR-INVERT			

<b>Codificadores</b>	▶	74147 - de prioridad, 10 líneas a 4
Descodificadores	▶	74148 - de prioridad, 8 líneas a 3
..		
<b>Descodificadores</b>	▶	7442 - 1 de 10 líneas (BCD a decimal)
Multiplexores	▶	7447 - BCD a 7 Segmentos
ALU	▶	74137 - 1 de 8 líneas con latches de dirección
Generador de paridad	▶	74138 - 1 de 8 líneas
Comparador	▶	74139 - 1 de 4 líneas (x2)
Sumadores	▶	74155 - 1 de 4 líneas (x2)
		74247 - BCD a 7 Segmentos

<b>Multiplexores</b>	▶	74151 - 8 líneas a 1
ALU	▶	74153 - 4 líneas a 1 (x2)
Generador de paridad	▶	74157 - 2 líneas a 1 (x4)
Comparador	▶	74158 - 2 líneas a 1 (x4)
Sumadores	▶	74298 - 2 líneas a 1 con registro (x4)
		74352 - 4 líneas a 1 (x2)
		74398 - 2 líneas a 1 con registro (x4)
		74399 - 2 líneas a 1 con registro (x4)

<b>ALU</b>	▶	74181 - 4 bits
Generador de paridad	▶	74280 - Generador/revisor de paridad par/impar de 9 bits
Comparador	▶	7485 - 4 bits
Sumadores	▶	7483A - 4 bits
		74283 - 4 bits

<b>Memoria</b>	▶	74170 - Memoria de lectura y escritura 4 x 4
----------------	---	--

<b>Flip-flops</b>	▶	7473A - JK flanco negativo(x2)
Registros	▶	7474A - D, preset, clear, flanco positivo (x2)
Contadores	▶	7476A - JK, preset, clear, flanco negativo (x2)
		74107A - JK flanco negativo (x2)
		74109A - JK flanco positivo (x2)
		74112A - JK flanco negativo (x2)
		74113A - JK flanco negativo (x2)
		74114A - JK flanco negativo (x2)

Flip-flops ▶	
<b>Registros ▶</b>	con Latches ▶
Contadores ▶	con Flip-flops ▶
	de desplazamiento ▶

<b>con Latches ▶</b>	7475 - 4 latches D
con Flip-flops ▶	7477 - 4 latches D
de desplazamiento ▶	74256 - Latch direccionable de 4 bits (x2)
	74259 - Latch direccionable de 8 bits
	74279 - 4 latches con set y reset
	74375 - 4 latches D

<b>con Flip-flops ▶</b>	74174 - 6 flip-flops D
de desplazamiento ▶	74175 - 4 flip-flops D
	74273 - 8 flip-flops D con clear
	74377 - 8 flip-flops D con enable
	74378 - 6 flip-flops D con enable
	74379 - 4 flip-flops D con enable

<b>de desplazamiento ▶</b>	7495B - 4 bits
	74164 - Entrada serie, salida paralela
	74165 - 8 bits, paralelo a serial
	74166 - Entrada paralela, salida serie
	74194A - bidireccional, 4 bits
	74195A - 4 bits, universal

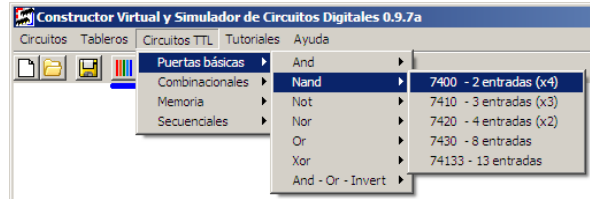
<b>Contadores ▶</b>	Asíncronos ▶
	Síncronos ▶

Asíncronos ▶	
<b>Síncronos ▶</b>	74160A - Módulo 10, reset asíncrono
	74161A - Módulo 16, reset asíncrono
	74162A - Módulo 10, reset síncrono
	74163A - Módulo 16, reset síncrono
	74168 - Bidireccional, módulo 10
	74169 - Bidireccional, módulo 16
	74190 - Módulo 10
	74191 - Módulo 16
	74192 - Bidireccional, BCD
	74193 - Bidireccional, módulo 16
	74669 - Bidireccional, módulo 16

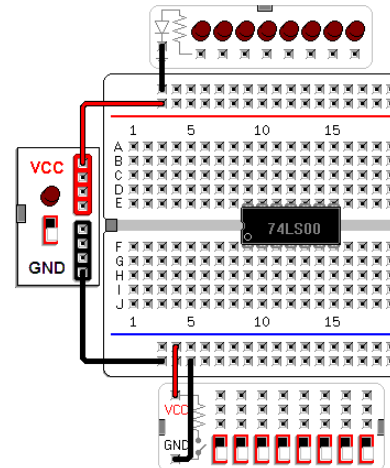
<b>Asíncronos ▶</b>	7490 - Divisor por 2 y 5
Síncronos ▶	7492 - Divisor por 2 y 6
	7493 - Divisor por 2 y 8
	74196 - Divisor entre 2 y 5
	74197 - Divisor entre 2 y 8
	74290 - Divisor entre 2 y 5
	74293 - Divisor entre 2 y 8
	74390 - Divisor entre 2 y 5 (x2)
	74393 - Contador binario de 4 bits (x2)
	74490 - Contador de décadas (x2)

## Apéndice B. Construcción y Simulación de un Circuito con 7400

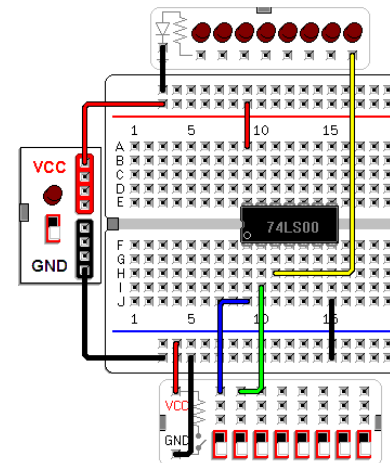
Inserte un protoboard, un tablero de leds y un tablero de interruptores. Conecte las líneas de VCC y GND e inserte un modelo del circuito integrado 74LS00. Recuerde guardar el circuito editado periódicamente.



El chip aparece en la parte superior izquierda de la ventana del programa. Arrástrelo sobre el protoboard para que quede insertado entre las filas E y F. El programa valida la inserción de los chips evitando conexiones de pines de salida con casillas conectadas a otros pines y puertos de salida. Un chip puede ser retirado con el botón derecho del ratón y trasladado arrastrándolo con el botón izquierdo.



Ahora conecte los pines de entrada 1 y 2 del chip a dos interruptores y la salida 3 del chip a un led. También conecte VCC al pin 14 y GND al pin 7. Para modificar un cable presione el botón izquierdo sobre un segmento y arrastre el ratón para dar al cable una nueva forma. Para eliminar un cable pulse el botón derecho del ratón cuando el cursor esté sobre uno de los segmentos modificables del cable.



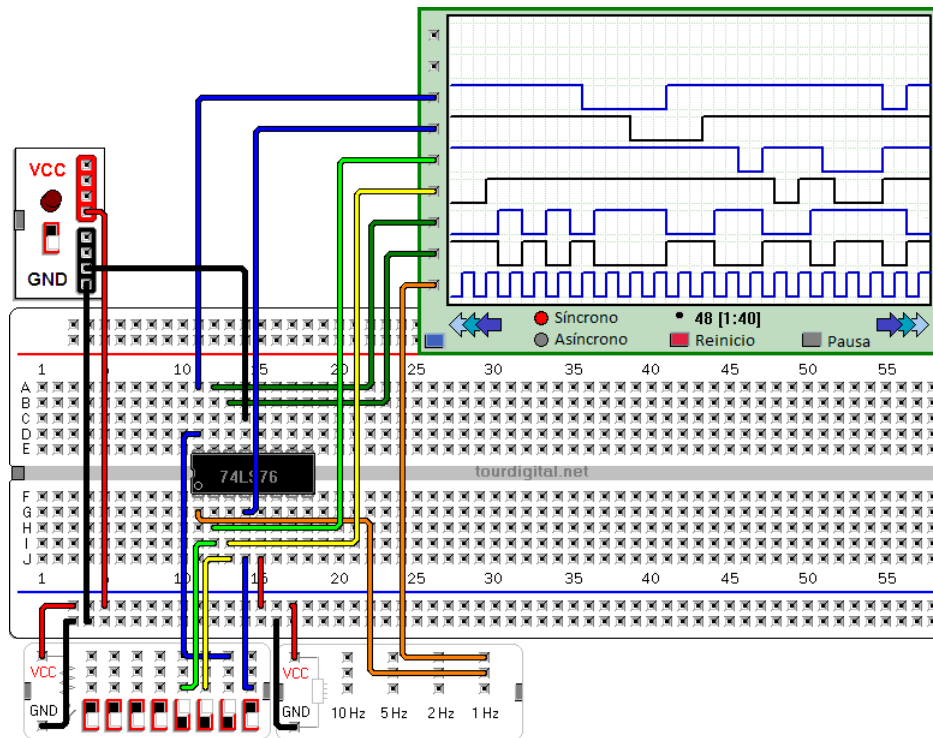
Encienda el módulo y observe cómo funciona el circuito conectado. Puede observarse que el led enciende cuando uno o dos de los interruptores están hacia abajo y se apaga cuando ambos interruptores están hacia arriba. Asignando los símbolos 0 y 1 para cada estado de los interruptores y del led se puede formar una tabla de verdad. Verifique que las otras puertas del circuito integrado producen la misma función lógica.



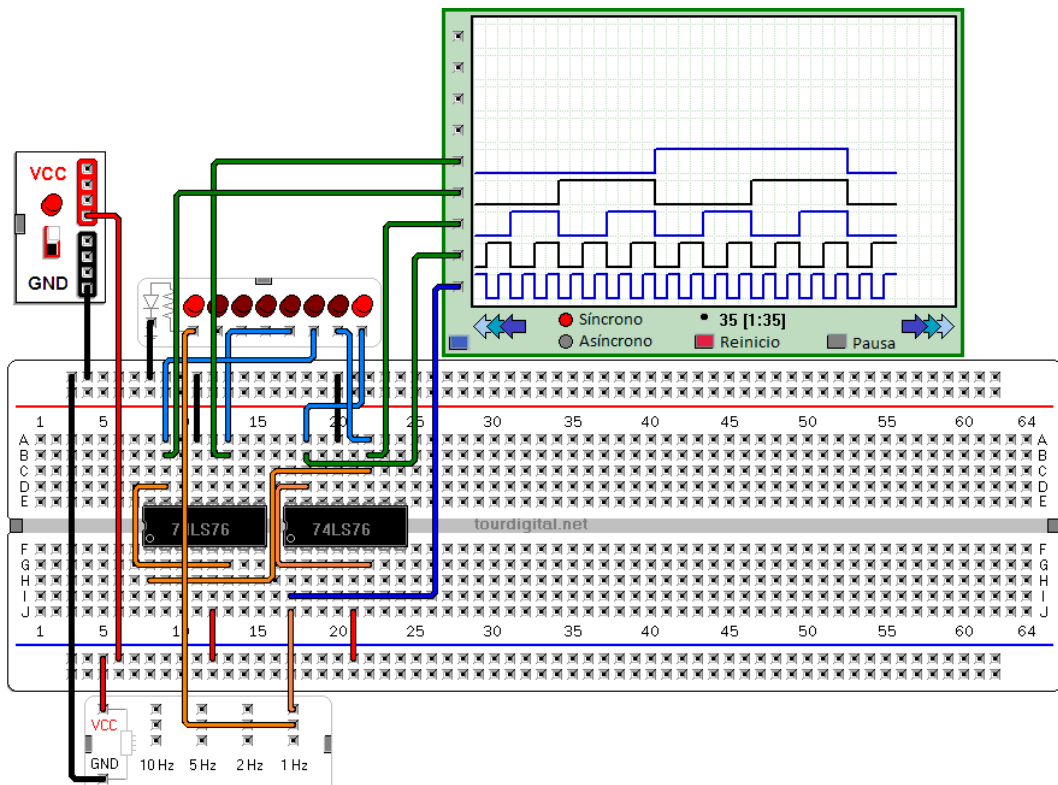
a	b	s
0	0	1
0	1	1
1	0	1
1	1	0

## Apéndice C. Análisis del Flip-flop JK

Consulte en las hojas de datos las especificaciones del chip 7476A y luego forme el circuito de la figura siguiente.



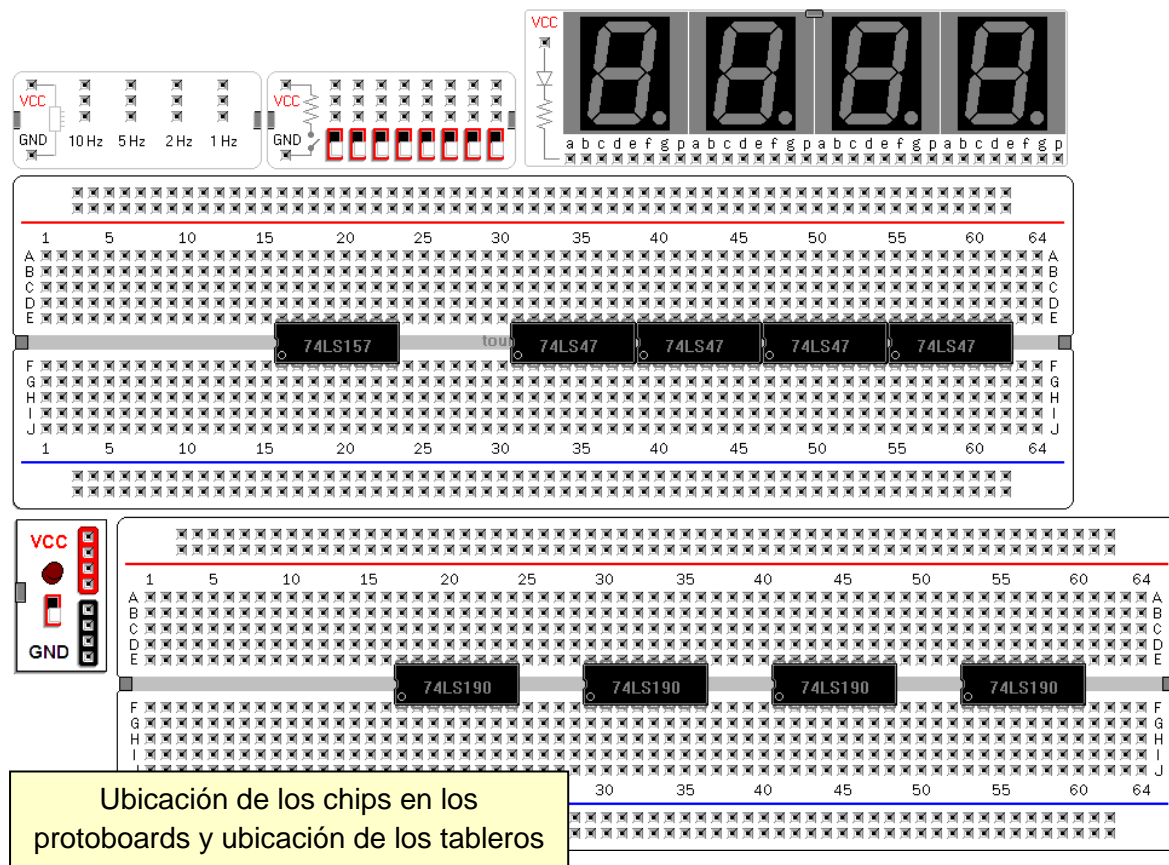
Note que las entradas flotantes son consideradas por el programa como 1 lógico. En la práctica, a veces será necesario conectar una resistencia ( $1K\Omega$ ) entre VCC y los pines de entrada de control. Cuando culmine el circuito anterior forme el circuito de la figura siguiente para analizar el comportamiento de un contador.



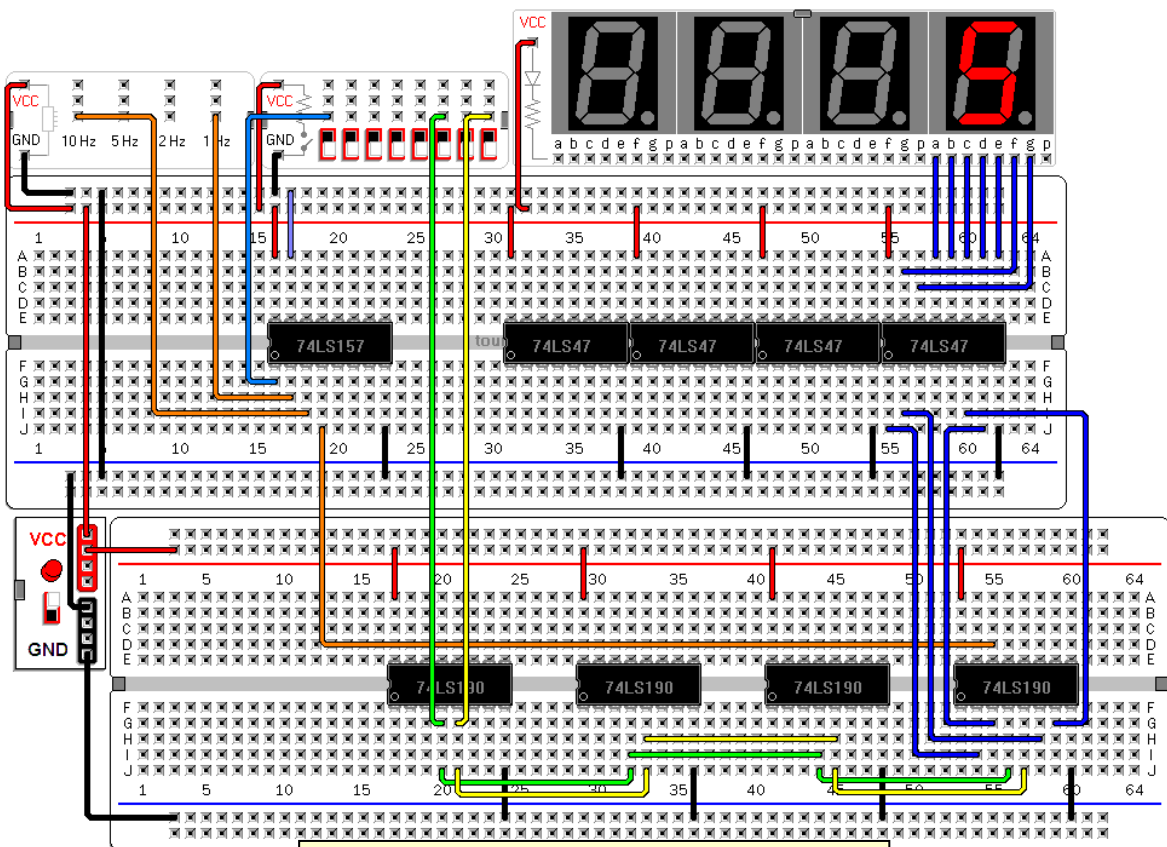
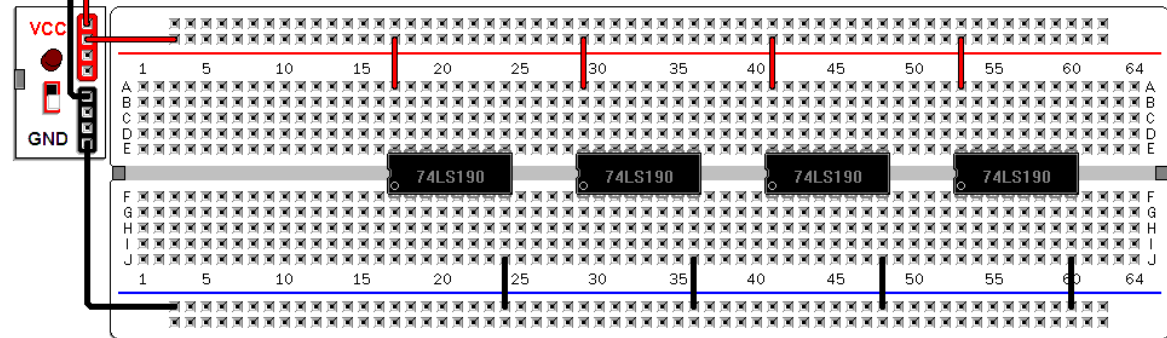
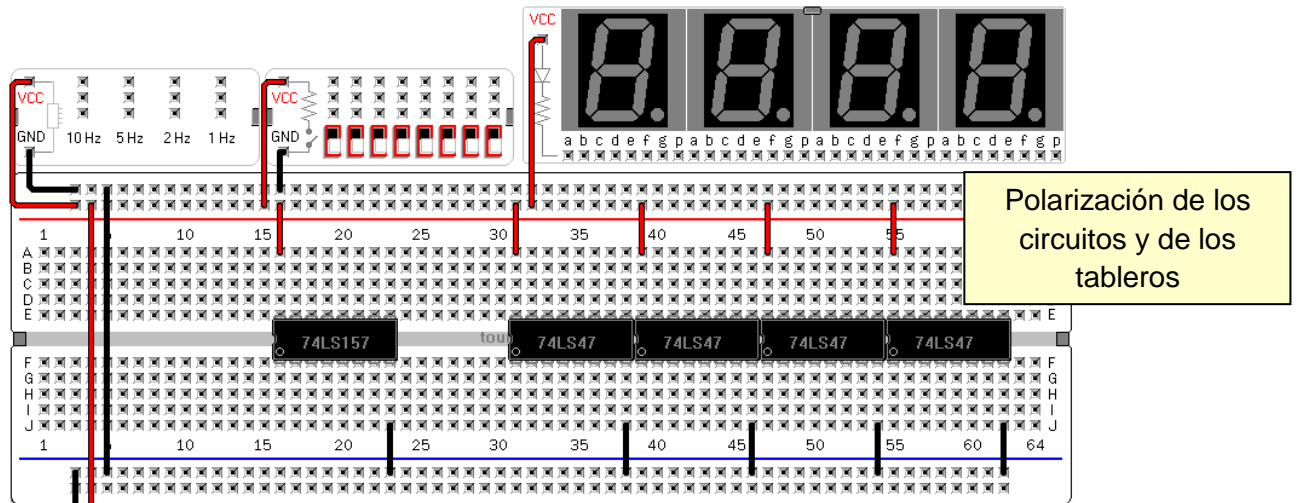
## Apéndice D. Contador BCD de cuatro dígitos con selector de frecuencia

Este apéndice muestra los pasos para construir un circuito de mediana complejidad. Primero debe hacerse un diagrama de bloques donde aparezcan todos los nombres de los nodos del circuito. Luego se construye el circuito por fases, comprobando cada fase antes de continuar con la siguiente. En el caso de un contador de cuatro dígitos, se prueba primero el funcionamiento de la primera etapa de cuenta (unidades) antes de construir la siguiente etapa (decenas) y así hasta las decenas de millar.

El circuito consta de contadores 74190, decodificadores 7447 y un multiplexor 74157. Consulte las hojas de datos de estos chips antes de comenzar la construcción del circuito.







Conexiones del selector, del visualizador  
y del contador de unidades

