

Verificador de circuitos lógicos

Reyes Ríos Cabrera¹, Ricardo Chaparro Sánchez², Rafael Urretabizkaya Garbus²

¹ CIATEQ A.C. Centro de Tecnología Avanzada,
Manantiales 23^a, Fracc. Ind. B.Q., 76246, Querétaro,
México

² Facultad de Informática, Universidad Autónoma de Querétaro
Cerro de las Campanas, 76010, Querétaro,
México
rchapa@uaq.mx, reyes.rios@ieee.org

Resumen. El artículo presenta un verificador de circuitos lógicos, con capacidad para probar circuitos digitales de las familias estándar TTL y CMOS. Con este verificador se resuelve el problema de diagnóstico de circuitos dañados que existe en los laboratorios de electrónica de las escuelas, así como también en instituciones de investigación, constituyendo con una propuesta que aporta beneficios como bajo costo, fácil mantenimiento, manejo y escalable. Consta de dos partes un dispositivo electrónico y un software. Es de fácil manejo y cuenta con elementos como consulta de diagramas y temas de ayuda, además de ser expandible.

1 Introducción

En los laboratorios de pruebas con el uso y el paso del tiempo, los circuitos lógicos sufren deterioros o daños totales afectando directamente el funcionamiento de prácticas y proyectos realizados. Por ello, es de gran importancia conocer el estado de los circuitos antes de usarse o en el proceso de uso para detectar los errores que provocan, y con esto asegurar un funcionamiento correcto de lo que se esté haciendo.

Se han desarrollado muchos instrumentos y equipos que verifican los circuitos lógicos, sin embargo tienen algunas desventajas importantes como el desarrollado por D. M Brewer [1], donde utiliza dos microprocesadores Zilog y varios periféricos, elevando su complejidad tanto de construcción como de programación, generando que el mantenimiento sea complejo y que sólo alguien especializado pueda repararlo.

La propuesta presentada resuelve el problema de forma sencilla, ya que el programa reside en una PC. Al hacer todos los procesos se ejecuten en la computadora, la complejidad de la programación y del diseño electrónico se reduce considerablemente, aportando ventajas como gran cantidad de memoria, velocidad de procesamiento y un ambiente gráfico amigable.

Algunas de las características que presenta éste verificador son:

- Es seguro, dado que cuenta con protecciones en el hardware que garantizan que no se dañe el verificador por errores en el manejo o descuidos del usuario.

- Es flexible y escalable, que puede adaptarse fácilmente para probar gran cantidad de familias lógicas. Tiene un manejo sencillo, tiene una interfaz de software para el usuario que cuenta con diferentes elementos amigables y sencillos.

2 Desarrollo

El sistema consta de un dispositivo electrónico que se conecta a la computadora y un software que lo controla, en esta sección se describen ambas.

En la figura 1, se muestra un diagrama general del sistema.

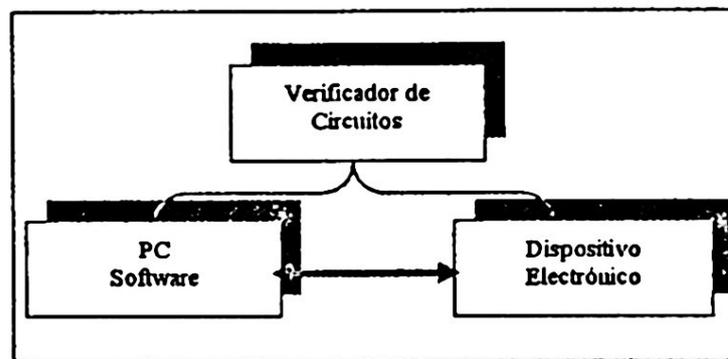


Fig. 1. Verificador de circuitos lógicos

El desarrollo se describe a partir de un planteamiento general de funcionamiento. La idea se basa en trasladar la complejidad de programación y electrónica a la computadora teniendo así mejores herramientas y más beneficios; reduciendo la complejidad y costo, tanto de desarrollo, como de mantenimiento del verificador, haciendo más fácil llevar a cabo versiones futuras, debido a que se traslada el problema a exclusiva programación en la PC, lo que hace posible un fácil crecimiento.

2.1 Dispositivo electrónico

El dispositivo se describe a partir del hardware, de las tareas y la comunicación con la PC.

2.1.1 Hardware. Se le considera una interfaz inteligente y está formada por dos microcontroladores (M1 y M2), los cuales, por medio de comunicación serial, se configuran y controlan por completo desde una PC. La interfaz inteligente tiene la característica de poderse configurar como se desee ya sean entradas o salidas digitales en cada una de sus terminales además de escribirse y leerse de la misma forma sin restricciones, teniendo la función de un puerto configurable.

Una vez hecho el análisis de los circuitos digitales se consideraron los siguientes requerimientos:

- Voltajes de 0V y 5V para marcar el cero y uno lógico respectivamente.
- 30 líneas Configurables como entradas o salidas digitales independientes.

- Puerto de comunicación.

A partir de los cuales se decidió usar el microcontrolador PIC16F877 de la marca Microchip [2], ya que cumple con las características.

El diagrama general del sistema, figura 2, cuenta con dos microcontroladores comunicados con una PC por medio de un bus serial (USART), un arreglo de buffers y una base de fuerza de inserción nula.

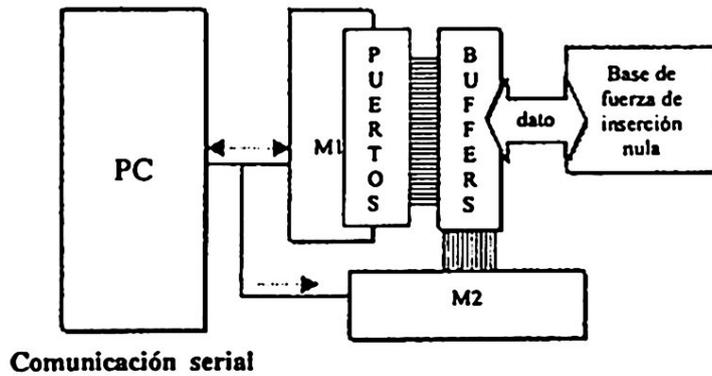


Fig. 2. Bloques del sistema

El arreglo de buffers configurable como entradas o salidas, se colocan donde los circuitos requieren energía, de modo que el circuito de prueba se encuentre siempre energizado por alguno de ellos, esto porque los microcontroladores no proporcionan la suficiente corriente para alimentar los circuitos.

Cada uno de los buffers conectados funcionan de la siguiente forma, figura 3.

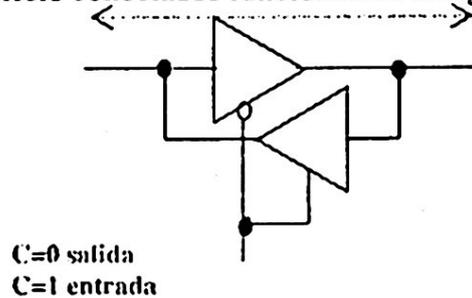


Fig. 3. Arreglo de Buffers

2.1.2 Tareas. M1 se encarga de las lecturas y escrituras en la base de prueba, M2 configura los buffers para determinar la función ya sea salida o entrada, figura 3, con respecto a M1.

De las 33 líneas de E/S digitales, que suman los puertos de M1; 2 se usan para la comunicación serial, 30 se distribuyen 15 y 15 en la base de fuerza de inserción nula y una más para un led indicador.

Los puertos de M1 se ordenan A, B, E, C, D, figura 4. Los puertos de M2 no se usan en su totalidad, sólo se usa el puerto B y D, que es con los que se configuran los buffers, y el resto es para escalamiento del dispositivo.

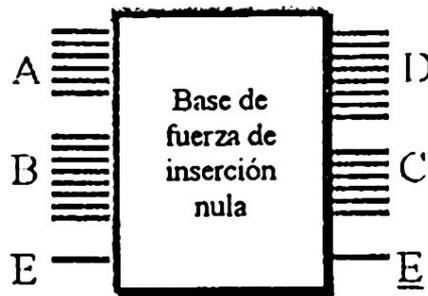


Fig. 4. Base de prueba.

2.1.3 Comunicación y operaciones básicas. La computadora hace la petición de lo que necesita, envía un 0 para hacer una petición de lectura y un 1 para la escritura a la interfaz inteligente. Al escribir M1 recibe dos bytes, el primero indica la dirección y el segundo el dato, figura 5a. Al leer se recibe un byte, que indica la dirección de la cual el M1 debe leer y regresar a la computadora el valor que se encuentre en esa dirección, figura 5b.

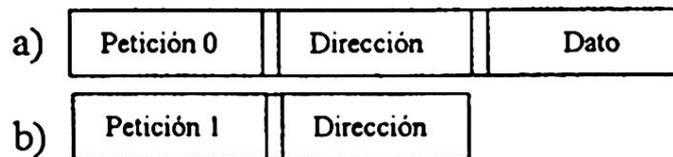


Fig. 5. Gráfica de las operaciones (sin protocolo).

La comunicación se llevo a cabo por medio del puerto USART de la interfaz inteligente, configurada a una frecuencia de 19200 bps, usando la norma RS232. Utiliza un protocolo de tres o cuatro bytes según la función, utilizando como byte de inicio, el código ASCII de la diagonal, los bytes de datos y como fin de trama el código de la diagonal invertida, figura 6.

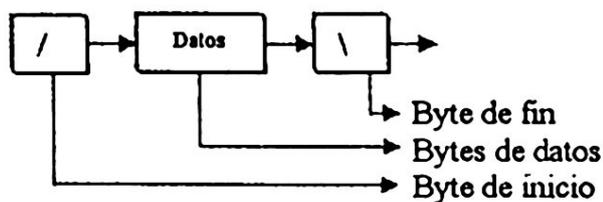


Fig. 6. Protocolo de comunicación

El protocolo es el mismo para los dos microcontroladores M1 y M2, lo que hace la diferencia es un bit de paridad enviado, de esta forma se define cual microcontrolador es el que tiene que hacer la operación, 0 para el M1 y 1 para el M2 (marca y espacio respectivamente).

2.1.4 Seguridad. El hardware está protegido contra cortos circuitos entre el circuito de prueba y la interfaz inteligente por medio de resistencias de 470Ω que mantienen a

los microcontroladores con una impedancia constante, aun si los voltajes no coinciden.

El verificador es el siguiente, figura 7

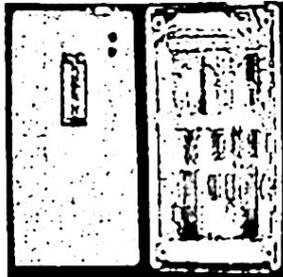


Fig. 7. Verificador de Circuitos lógicos.

2.2 Descripción del software

El software diseñado fue escrito en el lenguaje C++ Builder, [3] y [4], éste cuenta con varias opciones, figura 8.

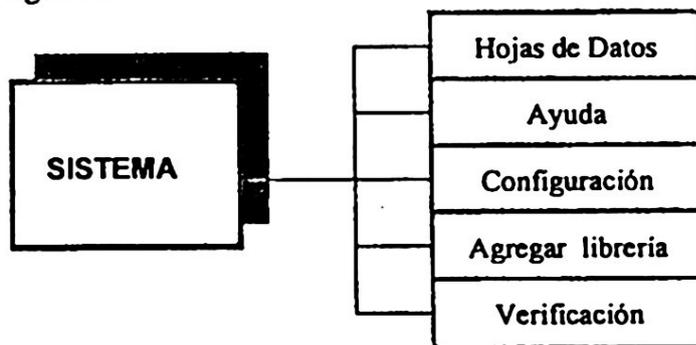


Fig. 8. Contenido de la Interfaz de Software

La primera opción nos muestra las hojas de datos de los circuitos de la familia TTL, en formato PDF, así como otros diagramas de uso común [5] y [6].

La parte de ayuda señala todos los procedimientos importantes en el sistema así como la interpretación de resultados y los pasos para llevar a cabo la reparación del verificador.

La sección de configuración nos permite fijar la frecuencia de trabajo y el puerto de comunicación con el dispositivo verificador (puertos COM1 ó COM2).

2.2.1 Agregar librerías. Las librerías de los circuitos de prueba son archivos que usa el sistema cuando lleva a cabo la verificación del funcionamiento. Maneja una librería por circuito, y esta contiene valores y características particulares.

Las librerías están guardadas de acuerdo a las operaciones que los circuitos necesitan de entrada y lo que se espera de respuesta, distribuidas en forma de vectores, donde si el circuito cuenta con 14 pines será un vector de 14 posiciones. En cada vector se encuentran los valores lógicos que determinan todas combinaciones necesarias para su funcionamiento.

Tabla 1. Contenido de las librerías

Contenido de la librería
Número de terminales
Terminal de voltaje
Terminal de tierra
Vector de configuración de entradas y salidas
Tabla lógica del circuito
...
...
...

Ejemplo para el circuito SN74LS14, figura 9.

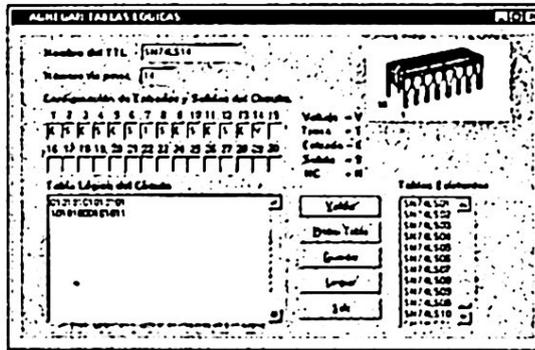


Fig. 9. Interfaz para agregar tablas lógicas

El software necesita como entrada el nombre del circuito y los datos mencionados, tabla 1. Cuando se ingresa la tabla lógica, un cero indica que es entrada en el circuito de prueba y un uno que es salida. Los microcontroladores no cuentan con datos, son únicamente una interfaz inteligente que sirve de intermediario entre el circuito de prueba y la computadora.

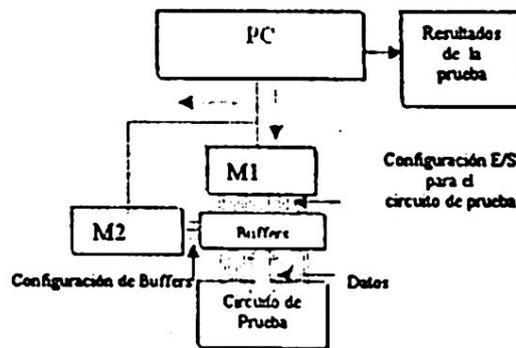


Fig. 10. Bloques de la verificación de un circuito

2.2.2 Verificación de funcionamiento. El primer paso es configurar M2, donde se define el funcionamiento de los buffers (entrada o salida con respecto a M1), figura 10, obteniendo los valores de la librería del circuito que se esté probando.

Después se configura M1, determinando las terminales como entradas o salidas para que coincidan con el circuito de prueba y en base a ello hacer la verificación. Una vez hecha la configuración comienza un ciclo terminando hasta que la tabla lógica del circuito es aplicada a éste en su totalidad. El procedimiento de aplicación de la tabla lógica se basa en las operaciones definidas de lectura y escritura, tomando la tabla lógica por bloques y distribuyéndola en la base de fuerza de inserción nula por medio de M1, escribiendo cada vector, leyendo las respuestas del circuito y comparando con resultados esperados. Finalmente muestra los resultados en la interfaz del software.

2.2.3 Opciones de prueba. Son dos una activa y pasiva; La primera consiste en una verificación constante e indefinida, finaliza cuando el usuario lo decida. La pasiva es sólo una escritura y lectura de la aplicación de la tabla lógica usándose una sola vez, determinando así el estado del circuito.

2.3 Análisis de resultados

El verificador general de circuitos lógicos desarrollado logró obtener de 100 pruebas activas realizadas a diferentes circuitos, 100 aciertos, además de otras 100 pasivas donde se obtuvieron resultados satisfactorios.

En el peor de los casos (donde la tabla es muy grande) se obtiene un tiempo de prueba de 15 segundos, mientras que el tiempo promedio es de 3 segundos por compuerta.

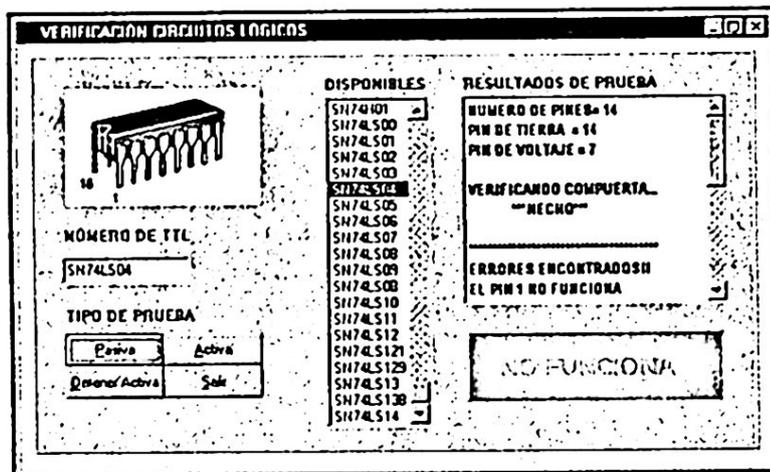


Fig. 11. Interfaz para la de verificación.

Actualmente se está trabajando en una tercera etapa, donde el proyecto, podrá identificar los dispositivos automáticamente; también se modificará para llevar a cabo la prueba y programación de memorias y arreglos lógicos, ya que teniendo las bases actuales, sólo basta agregar librerías en el formato establecido para llevar a cabo la programación o verificación que se desee.

3 Conclusiones

Se ha desarrollado un sistema verificador de circuitos lógicos, que resuelve el mismo problema que otros modelos comerciales [7], pero a un costo mucho menor, además de que para su mantenimiento no se requieren conocimientos avanzados.

Referencias

1. PC BASED TESTER FOR TTL & CMOS LOGIC DEVICES Written & Designed by D. M. Brewer 1996. <http://freespace.virgin.net/darren.brewer/project.htm>
2. Microchip Data Sheet PIC16F87X, 28/40-Pin 8-Bit CMOS FLASH microcontroladores.
3. Chirs H. Pappas, William H. Murray III "Manual de Borland C++ 4.0", McGraw-Hill, 1997.
4. Francisco Charte, "Programación con C++ Builder 5", Anaya Multimedia
5. Data Sheet Book 2 - J. P. M. Steeman, 1988, Elektuur B.V., Beek L, Netherlands, reproduksi dan distribusi Indonesia: 1989, PT. Elex Media Komputindo, Jakarta Indonesia.
6. Texas Instruments, "Logic Selection Guide and Databook CD-ROM", Advanced System Logic Products, January 1997.
7. Jameco Electronics, "Linear IC Tester", Octubre – January 2001-2002, Belmont California U.S.A.