

ENTORNO DE SIMULACION Y TEST PARA PLDs

L.F. Marsal, N. Cañellas, J. Pallarés, J.L. Ramírez y F. Hurtado
Escola Tècnica Superior d'Enginyeria, Universidad Rovira i Virgili
Carretera de Salou, s/n 43006, Tarragona
Tif: 977 55 96 25
Fax: 977 55 97 10
e-mail: lmarsal@etse.urv.es

RESUMEN.- El objetivo final de todo diseño es su implementación y correcto funcionamiento. En el caso de los PLDs, la verificación del correcto funcionamiento es muy importante dada la naturaleza de estos dispositivos donde su funcionalidad no está definida hasta que se programa. La problemática que se plantea entonces, es la realización de una serie de pruebas de test para verificar su correcto funcionamiento. El presente trabajo presenta un sistema software-hardware de test para dispositivos lógicos programables (PLDs). Esta herramienta está pensada como soporte docente a las prácticas de PLDs que se realizan en el laboratorio de electrónica digital.

1.- INTRODUCCIÓN

La verificación de los dispositivos programados tiene un coste en tiempo que puede traducirse en dinero. Sin embargo, se estima que los costes para detectar un PLD defectuoso montado en placa de circuito impreso son diez veces mayores que los de detección del PLD defectuoso en el propio programador. Si el defecto se descubre más tarde, cuando el equipo ha sido instalado y el propietario observa un comportamiento anómalo, el coste de la reparación puede superar en mil veces a los costes de detección del PLD defectuoso en el programador de PLDs.

En las aulas también nos podemos encontrar en situaciones donde el uso de sistemas de testeo puede ser de gran ayuda. Por ejemplo: El alumno bajo la supervisión de un profesor y las directrices de una práctica, debe diseñar, dimensionar e implementar un sistema secuencial con fines didácticos. Se le proporciona el material necesario, en el que se incluye un dispositivo programable tipo PLD. Esta PLD, que le ha sido prestada por el responsable del laboratorio, se utilizó anteriormente en otras prácticas, y realmente no se conoce su fiabilidad en estos momentos. El estudiante reprograma la matriz de configuración con sus ecuaciones y se dispone a implementar su sistema ... 2 horas más tarde el alumno desesperado y tras varias modificaciones y correcciones in situ, todavía no ha conseguido hacer funcionar correctamente el sistema. Y si aún le quedan ánimos en el cuerpo, se preguntará: ¿dónde está el fallo?, ¿en el software?, ¿en el hardware?. Posteriormente todo se soluciona cuando se prueba el diseño con otra PLD. El fallo de sistema lo provocaba el dispositivo programable, que estaba dañado físicamente.

No son del todo inhabituales este tipo de frustrantes experiencias. Ya que estamos hablando de dispositivos que son utilizados y reprogramados por muchos alumnos, los cuales pueden

someterlos a regímenes de funcionamiento extremos. Ello conduce a la larga que un porcentaje de PLD acaben estropeándose.

Para evitar quebraderos de cabeza con respecto al material defectuoso, el alumnado debería disponer de medios de diseño y testeo adecuados. Además, este test puede ser muy engorroso y de extrema dificultad para el alumno, siendo de gran utilidad una herramienta que permita simular y verificar, en un entorno agradable, fácil de manejar y rápido, los valores reales del PLD. En este sentido, se ha desarrollado un sistema de test para PLDs que complementa la formación práctica del alumno. Estas prácticas se desarrollan en la Escuela Técnica Superior de Ingeniería de la Universidad Rovira i Virgili, dentro de las titulaciones de Ingeniería Técnica en Informática de Sistemas y Ingeniería Técnica Industrial especialidad Electrónica Industrial en las asignaturas de Sistemas Digitales. El software utilizado es el paquete ST-CUPL 4.0a [1]. Para implementar los diseños propuestos a los alumnos, utilizamos PLDs del tipo GAL (Generic Array Logic) [2]: GAL16V8, GAL20V8, GAL22V10 y GAL6001. Estos tienen la ventaja de ser unos dispositivos clásicos y con una complejidad baja-media, que los hace muy adecuados para la primera toma de contacto del alumno con un PLD. Para facilitar el test, se ha considerado utilización de los ficheros de simulación generados por el CUPL y utilizarlos como vectores de test.

2.- EL TESTER DIGITAL DE PLDs

El sistema está dividido en 2 partes tal como podemos observar en la figura 1:

-El módulo PC: Un PC-XT o superior con puerto de comunicaciones RS-232 y ratón. Dicho PC ejecutará la aplicación TESTER (Software), que permite desde un entorno gráfico (640 x 480 a 16 colores) y por ventanas, controlar la generación, simulación y análisis de vectores de test, además de llevar a cabo la comunicación (con control de errores) entre el módulo del PC y el módulo TESTER a través de un cable serie.

-El módulo TESTER: Placa de test (Hardware), está compuesta de una caja de dimensiones reducidas, donde se halla instalado un zócalo de inserción nula (aquí se conectan los PLDs), un puerto delta-9 de comunicaciones, un led testigo de funcionamiento, interruptor de encendido, conmutador modo 20/24 pins, botón de reset y un jack mono para la alimentación.

Los requerimientos mínimos del sistema son:

- Un Ordenador PC-XT o superior
- Un ratón
- 2 puertos de comunicaciones RS-232 (uno de ellos para el ratón)
- Una placa de video VGA
- MS-DOS versión 3.1 o superior
- 640 KB de memoria.

3.- MODOS DE FUNCIONAMIENTO DEL TESTER

El sistema tiene 2 modos de funcionamiento:

-El modo Tester: En este modo el usuario debe proporcionar al sistema el fichero fuente (*.PLD) utilizado para programar el PLD. Este creará un archivo *.DOC a través del compilador CUPL y mostrará gráficamente una ventana de estado del PLD, con los nombres

de las variables y el estatus (0/1) de cada pin. Mediante el ratón podemos modificar y reflejar el valor real de los pins de la GAL en el mismo instante en que se modifique alguna entrada. El sistema también funciona proporcionando directamente el archivo *.DOC.

-El modo Simulador-Tester: Este modo, aprovecha los vectores de test creados a partir de la simulación del CUPL (*.SO) y presenta gráficamente, en forma de cronogramas, los resultados de una simulación y los resultados del test físico. Estos vectores de test se pueden crear independientemente del programa CUPL mediante un editor de texto. Al mismo tiempo se indican el nombre de los pins de entrada y salida, y las ecuaciones que las relacionan. También se presentan una serie de controles como son un zoom, posicionado en un ciclo concreto del cronograma y captura de errores. Cuando existe una discrepancia entre un resultado de la simulación y del test físico, este valor se representa en el cronograma de color rojo.

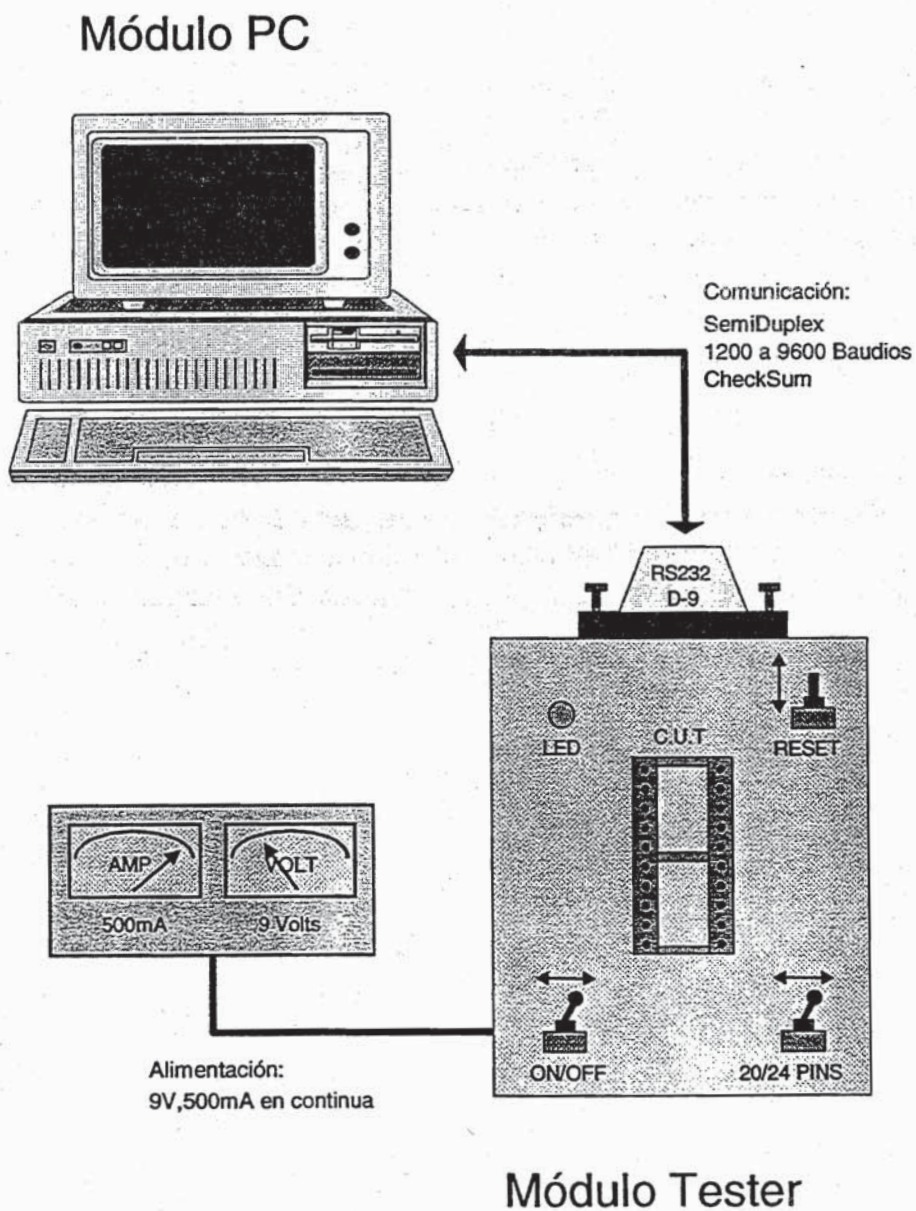


figura 1.- Esquema general del sistema de test para PLDs.

4.- EJEMPLO DE TEST PARA UNA GAL16V8

Como primera toma de contacto del alumno con el programa, se plantea un sencillo ejemplo de verificación del correcto funcionamiento de una GAL16V8. Previamente el alumno habrá compilado y programado la GAL mediante el programa ST-CUPL. Las funciones que se han programado son de dos tipos: combinatoriales (AND, OR y XOR) y secuenciales (biestables tipo D). La figura 2 muestra un esquema de las funciones implementadas en la GAL junto con el fichero tester.pld.

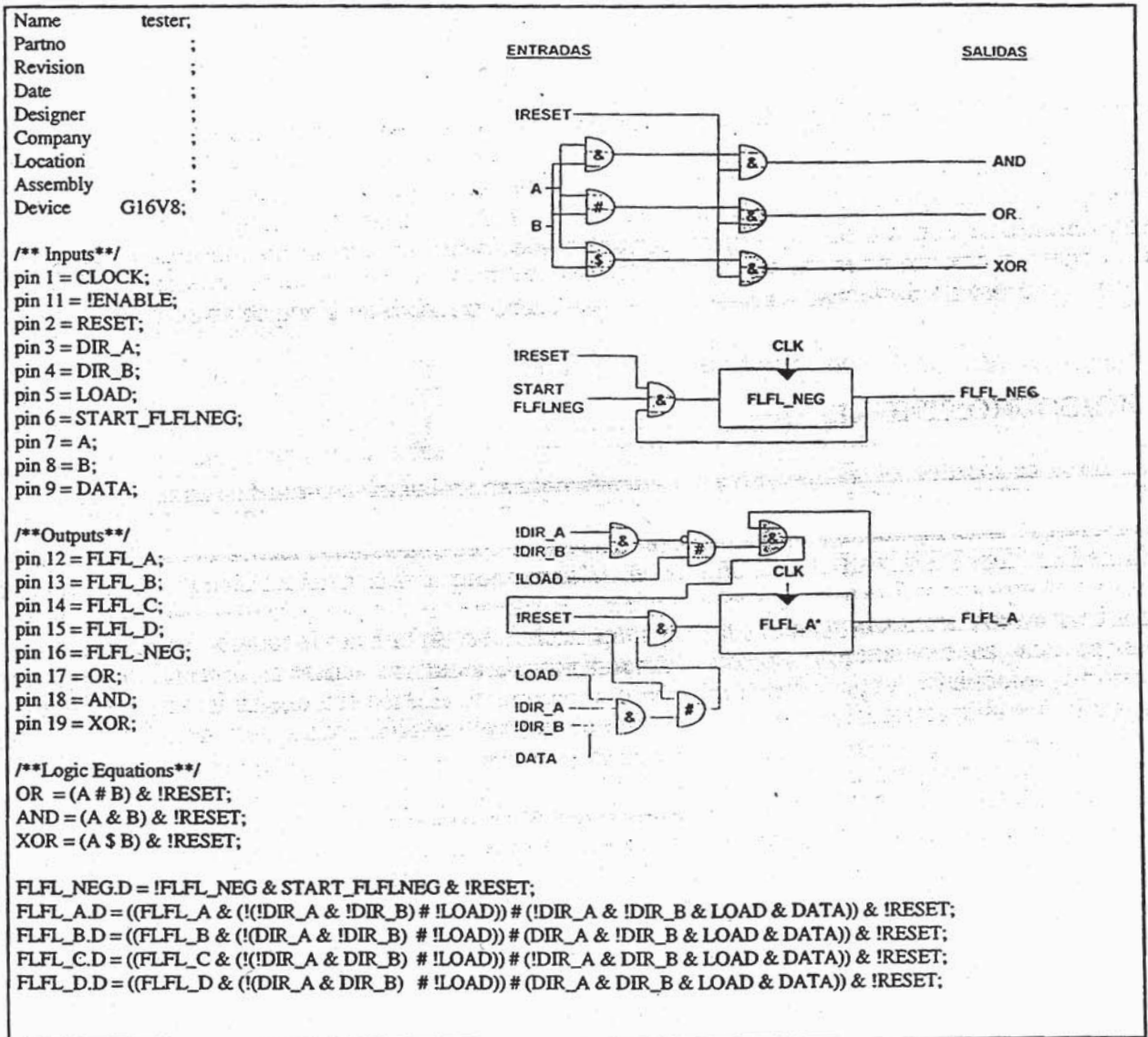


figura 2.- Esquema del circuito implemetado junto con el fichero tester.pld

La figura 3 muestra la pantalla de trabajo en el modo Tester. Como se puede observar, se presenta un esquema de la GAL utilizada con las variables asociadas a cada pin. Ahora no tenemos nada más que ir con el ratón y cambiar a voluntad los pines de entrada y observar los resultados de la salida. Este modo además de comprobar si el resultado esperado es el correcto o incorrecto, tiene la ventaja de no tener que montar el PLD sobre una placa protoboar y realizar el consiguiente cableado.

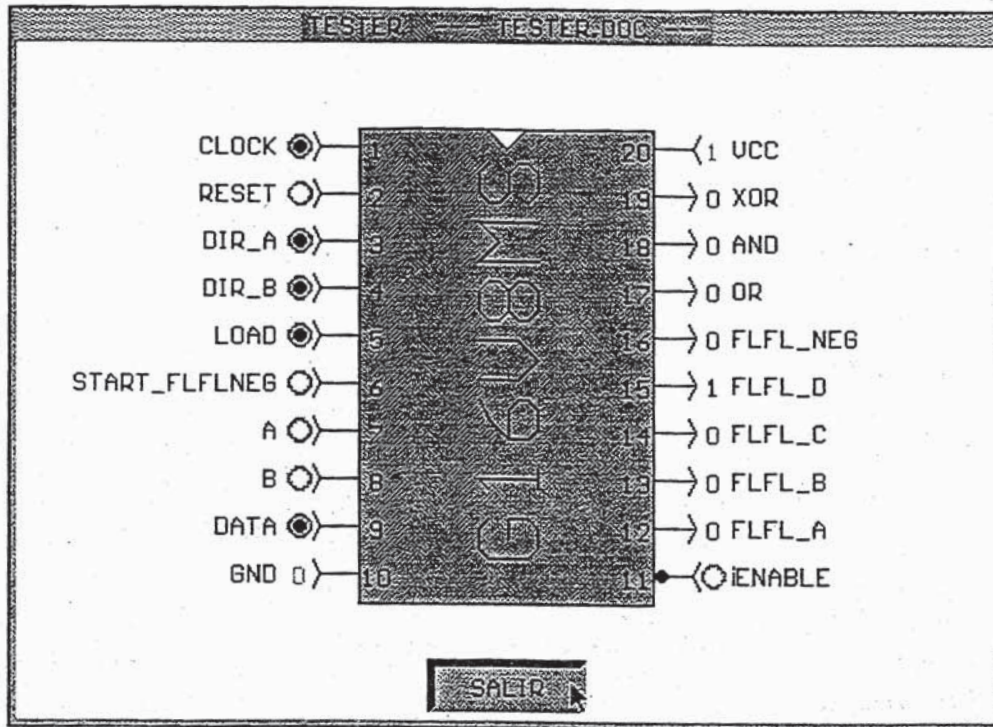


figura 3.- Representación de la GAL16V8 en el modo Tester.

```

1: Name      tester;
2: Partno   ;
3: Revision ;
4: Date     ;
5: Designer ;
6: Company  ;
7: Location ;
8: Assembly ;
9: Device   G16V8;
10: ORDER:  !ENABLE,RESET,A,B,CLOCK,LOAD,DIR_B,DIR_A,START_FLFLNEG,DATA,
            FLFL_A,FLFL_B,FLFL_C,FLFL_D,FLFL_NEG,AND,OR,XOR;

```

Simulation Results

	!ENABLE	RESET	A,B	CLK	LOAD	DIR _B,A	START_ _DATA	FLFL_ A,B,C,D	FLFL_ NEG	AND	OR	XOR
0001: 1	0		XX	0	X	XX	XX	ZZZZ	Z	X	X	X
0002: 0	1		00	C	0	00	00	LLLL	L	L	L	L
0003: 0	0		00	0	0	00	00	LLLL	L	L	L	L
0004: 0	0		01	0	0	00	00	LLLL	L	L	H	H
0005: 0	0		10	0	0	00	00	LLLL	L	L	H	H
0006: 0	0		11	0	0	00	00	LLLL	L	H	L	L
0007: 0	0		00	C	0	00	10	LLLL	H	L	L	L
0008: 0	0		00	C	0	00	10	LLLL	L	L	L	L
0009: 0	0		00	C	1	00	01	HLLL	L	L	L	L
0010: 0	0		00	C	1	01	01	HHLL	L	L	L	L
0011: 0	0		00	C	1	10	01	HHHL	L	L	L	L
0012: 0	0		00	C	1	11	01	HHHH	L	L	L	L
0013: 0	0		00	C	1	00	00	LHHH	L	L	L	L
0014: 0	0		00	C	1	01	00	LLHH	L	L	L	L
0015: 0	0		00	C	1	10	00	LLLH	L	L	L	L
0016: 0	0		00	C	1	11	10	OLLLL	L	L	L	L

figura 4.- Fichero de resultados de la simulación (tester.so). Los resultados son comparados con los valores reales obtenidos en la placa de test.

Una vez simulado el dispositivo, los vectores de test del fichero tester.so son utilizados como vectores de entrada en el modo Simulador-Tester. Los resultados reales obtenidos en la placa de test son comparados con los obtenidos en la simulación. La figura 5 presenta la pantalla de trabajo en este modo. Como se observa, la pantalla esta dividida en tres partes principales. A la izquierda y en una columna se presentan los nombres de las variables de entrada y salida de la PLD. En la parte superior-derecha se muestra el cronograma de los vectores de entrada y en la parte inferior-derecha se muestra el cronograma de los vectores de salida simulados. Si existe alguna discrepancia entre los valores simulados y los reales, el cronograma mostrara el error cambiando el color del cronograma en el instante de tiempo donde se produce.

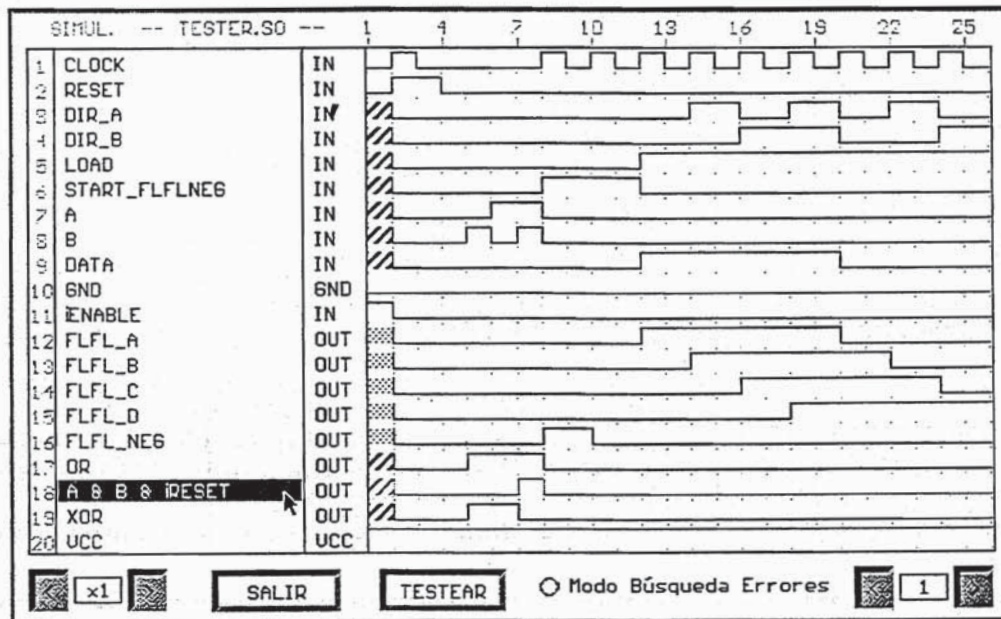


figura 5.- Pantalla de trabajo en el modo Simulador-Tester.

5.- CONCLUSIONES

Para aumentar el nivel docente de nuestras clases, se ha desarrollado un equipo de test, en un entorno agradable y de fácil uso, que complementa la formación del alumno en las prácticas de Sistemas Digitales. Por otro lado, se presenta de una manera práctica e intuitiva, la importancia de la simulación y test en el diseño digital, de manera que el alumno pueda experimentar con los conceptos aprendidos en las clases de teoría.

El uso de este sistema de test de PLDs, nos ayudará en el área del software (a través del simulador comprobaremos si la programación se ajusta a nuestras expectativas), y en el área del hardware (testando el correcto funcionamiento del dispositivo PLD).

Dentro del área de las prácticas en el laboratorio, el sistema de test de PLDs, se convierte en una herramienta sencilla y útil para el alumno. Con este sistema de testeo, el alumno entrará en contacto con la tecnología programable PLD de una manera cómoda y segura.

6.- REFERENCIAS

- [1] ST-CUPL™ Universal Compiler For Programmable Logic, SGS-Thomson & Logical Devices, Inc. 1990.
- [2] GAL Programmable Logical Devices, Databook, SGS-Thomson Microelectronics, 1992.