

PROXIL: Programa de apoyo a la enseñanza del diseño de sistemas digitales basados en circuitos configurables

M.D. Valdés, E. Mandado y M.J. Moure
Instituto de Electrónica Aplicada "Pedro Barrié de la Maza"
Universidad de Vigo
Apartado Oficial de Correos.36200 Vigo
Teléfono: (986) 812143 Fax: (986) 469547
e-mail: mvaldes@uvigo.es

RESUMEN.- Se presenta un software que permite la programación de Conjuntos Configurables de Puertas de Xilinx basados en memorias de acceso aleatorio estáticas, a partir de datos almacenados en un ordenador. Se basa en generar un fichero de configuración adecuado y las señales eléctricas necesarias en los terminales del dispositivo durante la programación. La utilización de esta herramienta acelera el proceso de desarrollo de prototipos, facilita las operaciones de verificación en tiempo real de los sistemas digitales, y permite la reprogramación de un dispositivo tantas veces como se desee y la implementación de diferentes estructuras circuitales, sin necesidad de utilizar memorias EPROM externas. Desde el punto de vista docente es una herramienta útil para el desarrollo de prácticas de laboratorio puesto que permite al estudiante centrarse en el diseño y verificación de sistemas digitales sin dedicar un tiempo considerable a los aspectos de la programación.

1.-INTRODUCCIÓN

Las técnicas de diseño de sistemas electrónicos digitales han evolucionado siguiendo los importantes cambios ocurridos en la tecnología. El paso desde el diseño con transistores individuales hasta el uso de circuitos integrados de gran escala de integración (VLSI), establece una tendencia hacia el desarrollo de sistemas cada vez más compactos y con mayores velocidades de operación.

Paralelamente a estos cambios han surgido herramientas de diseño asistido por ordenador (CAD) que permiten acelerar la fase de diseño de dispositivos y sistemas. Entre ellas destacan los capturadores de esquemas, las herramientas de síntesis lógica automática que facilitan el posicionamiento y conexionado (routing) sobre el área de silicio, las herramientas de simulación funcional y de tiempo, etc. No obstante, el ciclo de vida de un producto es cada vez menor respecto a su ciclo de diseño, por lo que se hace necesario disminuir el tiempo dedicado a la creación de prototipos. Es en este aspecto en el que los dispositivos lógicos programables asumen un papel importante, pues en muchos casos reducen considerablemente el tiempo requerido para el desarrollo de prototipos. La amplia utilidad de estos circuitos los convierte en materia obligada en la enseñanza de la electrónica digital [1].

2.- ESTRUCTURAS Y MÉTODOS DE PROGRAMACIÓN DE LOS CONJUNTOS CONFIGURABLES DE PUERTAS

Entre los diferentes circuitos configurables digitales se distinguen los conjuntos configurables de puertas [FPGAs (Field Programmable Gate Arrays)] [2], que permiten diseñar e implementar fácilmente sistemas digitales de alta complejidad. Consisten en una estructura de células lógicas básicas y un conjunto de conexiones programables que se configuran para realizar diferentes funciones lógicas, de acuerdo a la arquitectura del dispositivo.

Existen diferentes familias de FPGAs que se distinguen entre sí por su arquitectura, la densidad de puertas lógicas (capacidad), los recursos de interconexión disponibles, el tipo de unidad lógica sobre la que se implementan las células básicas y los métodos de programación, entre otros parámetros. Este trabajo se centra en las series XC3000 y XC4000 [3] de la familia Xilinx, cuya estructura se caracteriza por un conjunto de bloques lógicos dispuestos en forma de filas y columnas delimitadas por los recursos de interconexión programables. La estructura lógica está rodeada por los bloques de entrada/salida que permiten el acceso a los terminales del dispositivo.

Xilinx utiliza la tecnología de memorias de acceso aleatorio estáticas [SRAM (Static Random Access Memory)] para su configuración [4]. Esta característica permite que los dispositivos sean reprogramables, pero a su vez implica la utilización de algún otro elemento de almacenamiento que mantenga el mapa de configuración del dispositivo cuando el sistema no está alimentado. Una memoria EPROM suele ser el dispositivo típico empleado para realizar dicha función.

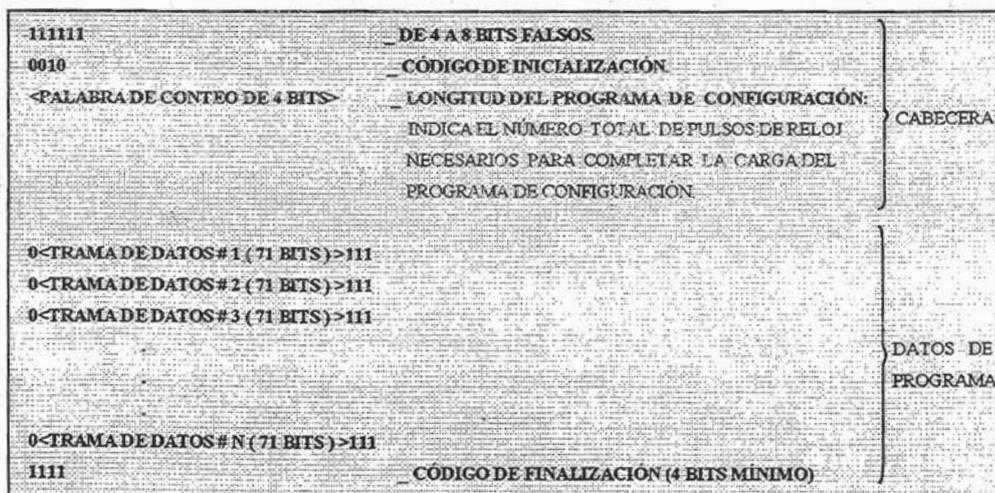


Figura 1.- Estructura de datos necesaria para la programación de una FPGA XC3000 de la familia Xilinx.

2.1.- Programación en modo periférico

Una de las características más útiles de las FPGAs bajo estudio es su capacidad de reprogramación. En aquellas aplicaciones en las que se desee cambiar la configuración de un dispositivo, resulta imprescindible disponer de una determinada capacidad de memoria que almacene los datos de todas las configuraciones. Si la FPGA se utiliza en una aplicación asociada a un PC, éste puede actuar como un dispositivo de memoria, con lo que se evita el uso de varias EPROM externas. Por ello, resulta interesante disponer de una

herramienta de programación que permita configurar una FPGA a partir de ficheros almacenados en un PC. Por este motivo se decidió elaborar PROXIL, un software de propósito específico que permite programar FPGAs de las series XC3000 o XC4000 de la familia Xilinx a partir de datos disponibles en un computador.

PROXIL utiliza el modo de programación periférico de los dispositivos y realiza la comunicación a través del puerto paralelo. Se basa en conformar el fichero de configuración de la FPGA y proporcionar las señales eléctricas adecuadas en cada momento, que aseguren los niveles lógicos necesarios en los terminales de programación. El fichero de configuración se obtiene a partir del que generan las herramientas de diseño del paquete WorkView para una memoria EPROM (extensión .EXO). El fichero se modifica para lograr una estructura de datos similar a la que se muestra en la Figura 1.

La programación en modo periférico se realiza de acuerdo con la secuencia de pasos descrita en la Figura 2. El proceso comienza con un ciclo de inicialización, puesta a cero y verificación. Durante este ciclo se comprueban los niveles adecuados del voltaje de alimentación (inicialización), se pone a cero la memoria de configuración (CLEAR) y se verifica que no está activa la señal de RESET. Seguidamente se identifica el modo de programación deseado y se procede a ejecutar la operación de acuerdo con las características de cada caso. El modo de programación se establece mediante los terminales M0, M1 y M2. Por último, se comprueba si la programación ha concluido con éxito mediante el cambio del estado lógico "0" al "1" del terminal DONE/PROGRAM.

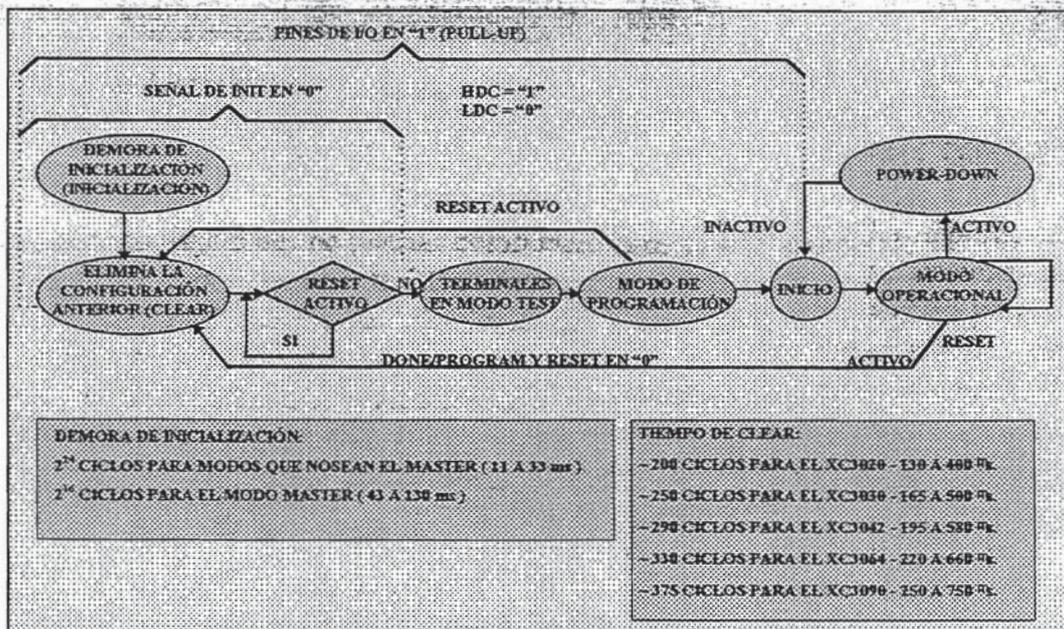


Figura 2.- Diagrama de estados del proceso de programación de una FPGA.

Durante el modo de operación normal la señal DONE/PROGRAM permanece en el estado lógico "1". Si se desea reprogramar un dispositivo hay que poner esta señal en nivel "0". La conexión del dispositivo durante la programación en modo periférico se corresponde con el diagrama de la Figura 3.

El software diseñado combina adecuadamente las señales antes mencionadas para lograr la configuración deseada en la FPGA. La generación y muestreo de estas señales requiere 15 bits en modo salida (incluye las señales antes mencionadas y las 8 del bus de datos) y 2 en

modo entrada (READY/BUSY e INIT). El ordinograma de la forma de operación de PROXIL se muestra en la Figura 4.

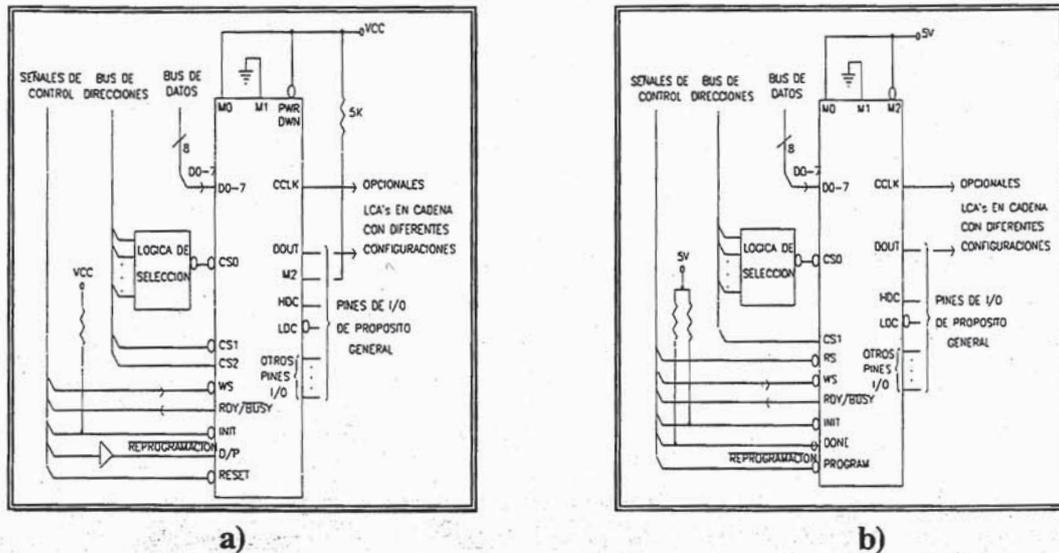


Figura 3.- Modo de programación Periférico. a) XC3000 y b) XC4000.

3.- PARTICULARIDADES DEL SOFTWARE Y DE LA TARJETA DE CONEXIÓN

Esta herramienta está orientada, fundamentalmente, al desarrollo de prácticas de laboratorio sobre el diseño y verificación en tiempo real de sistemas basados en FPGAs. Para ello se ha desarrollado una tarjeta que contiene un zócalo en el que se coloca la FPGA, resistores de pull-up de aproximadamente $4.7k\Omega$ conectados a los terminales INIT, RESET y M2, y un conector DB25, que permite la comunicación con el puerto paralelo del PC.

El software desarrollado se ejecuta sobre plataforma Windows y posee un entorno de trabajo interactivo, por lo que el tiempo requerido para su aprendizaje es muy breve.

Entre sus opciones permite elegir:

- I.- Selección del tipo de dispositivo a programar.
- II.- Información gráfica sobre las características topológicas que debe cumplir la tarjeta en la que se coloca la FPGA. Se especifica cuáles son los terminales que deben ir conectados a alimentación, y a masa y a qué terminal del conector DB25 debe ir conectada cada una de las señales que intervienen en el proceso de programación.
- III.- Configuración de los dispositivos.
- IV.- Información sobre el resultado de la configuración.
- V.- Impresión de la pantalla.

En la Figura 5 se muestra una pantalla del programa.

Aunque algunos fabricantes proporcionan sus propias herramientas para la programación de los dispositivos en el modo periférico, generalmente están orientadas a trabajar sobre kits de desarrollo, lo cual limita considerablemente su funcionalidad. Entre las ventajas que ofrece PROXIL frente a estas herramientas se distinguen:

- Puede utilizarse indistintamente para cualquier aplicación.
- La generación de los protocolos de comunicación y programación son transparentes para el usuario.
- Ofrece un entorno de trabajo interactivo.
- Teniendo en cuenta su aplicación con fines docentes, proporciona información sobre el dispositivo que se desea programar en cuanto a arquitectura, características de

conexión, distribución de los terminales y otras. Esta información se puede imprimir mediante una función disponible.

4.- UTILIDAD DEL SOFTWARE CON FINES DOCENTES

Este programa resulta de gran utilidad en un laboratorio de prácticas de circuitos digitales configurables porque permite comprobar, en un período de tiempo muy breve, gran número de configuraciones y elimina la necesidad de disponer de un programador de EPROMs, un borrador y memorias EPROMs. Se logra así que la atención del estudiante se centre en el diseño de determinadas topologías y se evita la dedicación de un tiempo excesivo a las tareas de programación del dispositivo para la prueba y puesta a punto del diseño realizado.

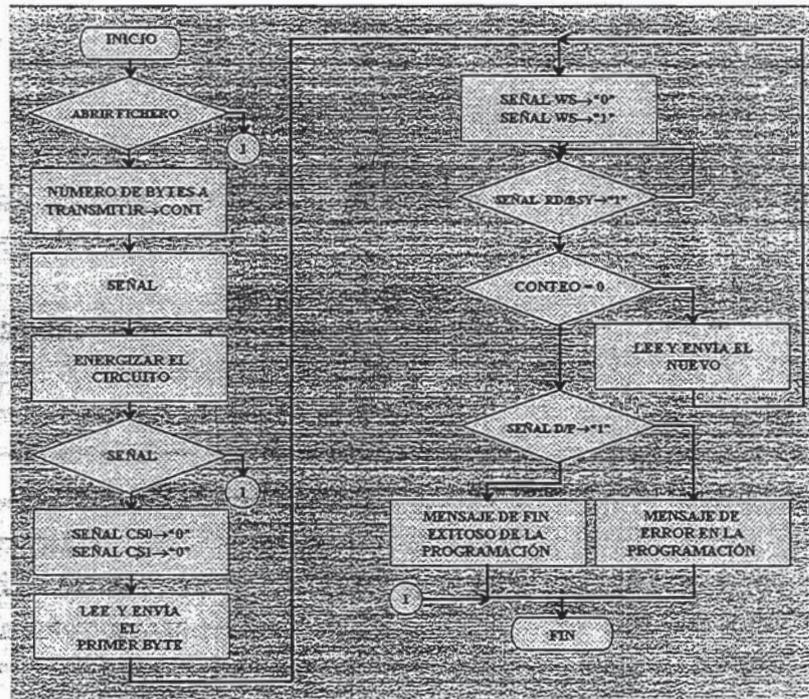


Figura 4.- Ordinograma de la forma de operación del programa PROXIL.

En cuanto a la infraestructura de un laboratorio, se requieren:

- *Tantas placas de prueba como puestos de trabajo existan.*

Las placas de prueba consisten en un zócalo, en el que se coloca el dispositivo a programar y un conector DB25 para establecer la comunicación con el ordenador. Es conveniente disponer de acceso a todos los terminales del zócalo con vistas a la verificación funcional de las configuraciones diseñadas.

- *Uno o varios ordenadores.*

El ordenador permite realizar las funciones de programación. Teniendo en cuenta que los dispositivos FPGAs no conservan la programación una vez retirada la alimentación, es necesario que los puestos de trabajo se coloquen cerca del ordenador. De esta forma también se evitan posibles errores de comunicación debido a la distancia entre el ordenador y las placas. La condición anterior implica utilizar un ordenador cada 3 ó 4 (como máximo) puestos de trabajo. Otra opción consiste en incluir pequeñas baterías en las placas de prueba para permitir el desplazamiento de un lugar a otro sin estar obligatoriamente conectados a

una fuente de alimentación. Esta variante posibilita el uso de un sólo ordenador para la programación de todos los dispositivos.

En la actualidad la mayoría de las Universidades y Centros Docentes disponen de uno o varios ordenadores destinados a actividades de laboratorio, por tanto el material requerido en este caso no se considera costoso ni de difícil adquisición.

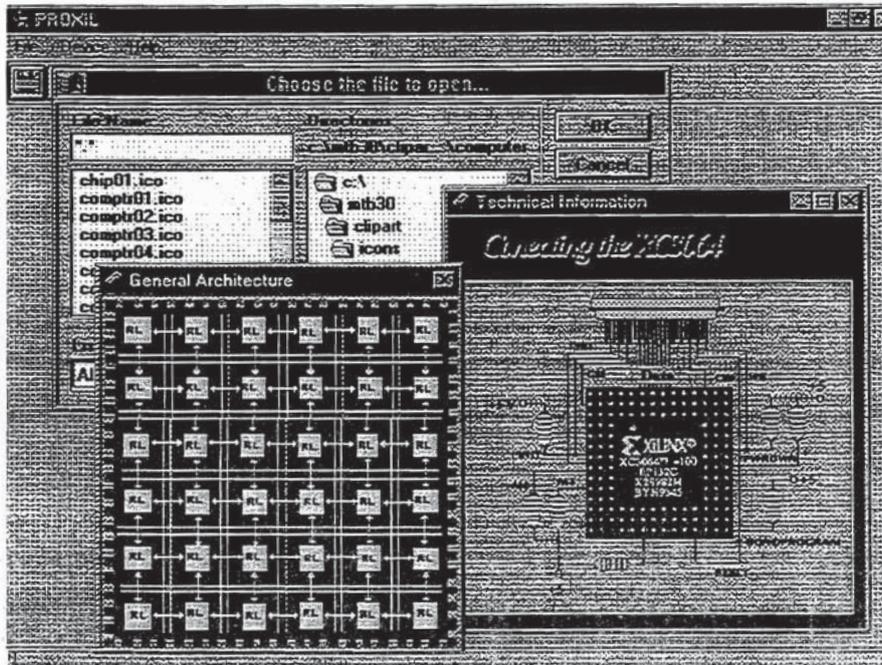


Figura 5.- Pantalla del programa.

5.- CONCLUSIONES

La importancia de los conjuntos configurables de puertas como elementos para el diseño de aplicaciones digitales es cada vez mayor. Sus características los convierten en componentes ideales para el desarrollo de prototipos, por lo que resulta de gran interés el diseño de herramientas que faciliten el trabajo con estos dispositivos.

El software que aquí se presenta permite la programación de FPGAs de Xilinx basadas en memorias de acceso aleatorio estáticas de forma rápida y sencilla, evitando el uso de EPROMs externas y de los dispositivos requeridos para la programación y borrado de estas memorias. La reprogramabilidad de los dispositivos posibilita la verificación y puesta a punto de las configuraciones circuitales de forma dinámica, lo cual acelera el proceso de creación de prototipos.

Por otra parte el uso de este programa puede extenderse al área docente, convirtiéndolo en una herramienta para el aprendizaje del diseño con conjuntos configurables de puertas.

6.- REFERENCIAS

- [1] Bolton M., "Digital System Design with Programmable Logic". Ed. Addison-Wesley, 1990.
- [2] Chan Pak K., Mourad S., "Digital Design Using Field Programmable Gate Arrays". Ed. Prentice Hall, 1994.
- [3] "Xilinx. The Programmable Logic Data Book", San José, California, 1994.
- [4] Conner D., "High-Density PLDs", *EDN*, Enero 1992.