

HERRAMIENTAS DE CAD PARA LA SÍNTESIS DE SISTEMAS DE INFERENCIA DIFUSOS MEDIANTE FPGAS*

S. SÁNCHEZ-SOLANO¹, A. CABRERA², C. J. JIMÉNEZ¹ Y D. R. LÓPEZ¹

¹ Instituto de Microelectrónica de Sevilla. Centro Nacional de Microelectrónica, Avda. Reina Mercedes s/n, 41012-Sevilla, España.

² Dpto. Automática y Computación. Facultad de Ingeniería Eléctrica. Instituto Superior Politécnico “José Antonio Echeverría”, Ciudad de la Habana, Cuba.

En esta comunicación se describe un flujo de diseño que permite automatizar el proceso de síntesis sobre FPGAs de sistemas de inferencia basados en lógica difusa. El entorno de CAD utilizado combina herramientas específicas desarrolladas por los autores con versiones educativas de herramientas comerciales de diseño de sistemas digitales. Todas las herramientas pueden ser ejecutadas sobre entornos MS-Windows, lo que facilita su utilización en aulas informáticas.

1. Introducción

Al crecer la complejidad de los sistemas electrónicos y convertirse en críticos los tiempos de desarrollo de nuevos productos, cobra una importancia capital el uso de herramientas informáticas de ayuda al diseño que faciliten las tareas de descripción, verificación y síntesis de sistemas de propósito general o específico. Por este motivo, los programas de numerosas asignaturas relacionadas con las tecnologías de la información incluyen entre sus objetivos el estudio de herramientas de CAD similares a las que encontrará el alumno en el desarrollo de su futura actividad profesional.

El número de aplicaciones electrónicas que utilizan soluciones basadas en lógica difusa se ha incrementado considerablemente en los últimos años y, de forma paralela, se han propuesto nuevas técnicas de implementación para este tipo de sistemas. De entre ellas, el uso de arquitecturas específicas de procesamiento implementadas sobre FPGAs presenta como principales ventajas una buena relación “coste/rendimiento” y un ciclo de desarrollo extremadamente corto [1, 2].

En esta comunicación se describe un flujo de diseño de sistemas difusos sobre FPGAs que se apoya en el uso combinado de herramientas específicas para el desarrollo de sistemas basados en lógica difusa [3] y herramientas genéricas de simulación y síntesis a partir de lenguajes de descripción de hardware. El uso de versiones educativas de las herramientas comerciales y la ejecución del entorno en ordenadores personales permite reducir el coste de cada puesto de trabajo con vistas a su empleo en aulas informáticas.

* Este trabajo ha sido parcialmente financiado por el proyecto CICYT TIC98-0869.

2. Flujo de diseño de sistemas difusos mediante FPGAs

Siguiendo el diagrama de flujo ilustrado en la Figura 1, el diseño de un sistema difuso comienza con la definición de su especificación XFL dentro del entorno *Xfuzzy*. Las distintas herramientas que incluye este entorno de desarrollo de sistemas difusos permiten ajustar la definición del sistema a las características de la aplicación considerada.

Una vez obtenida una especificación funcionalmente correcta, se realiza la fase de síntesis VHDL. La herramienta *xfvhdl* toma como entrada la especificación XFL de un sistema difuso y genera un conjunto de archivos VHDL que implementan el motor de inferencia utilizando a una serie de componentes predefinidos.

La descripción VHDL constituye el punto de partida de las etapas de simulación (simulador *ModelSim* de *Mentor Graphics*) y síntesis lógica (*FPGA Express* de *Synopsys*). Con objeto de acelerar la realización de estas dos etapas de diseño, *xfvhdl* proporciona dos salidas adicionales: 1) un *testbench* para facilitar la simulación del sistema difuso; y 2) procedimientos de comandos que permiten realizar en *batch* los procesos de síntesis e implementación de la FPGA (sobre dispositivos *Xilinx*).

Las herramientas de simulación VHDL, síntesis lógica e implementación sobre FPGAs pueden obtenerse a través de los “Programas Universitarios” de las distintas compañías [4]. En concreto, la versión 3.1i de *Foundation Express* incluye, además del software de *Xilinx*, versiones operativas de *FPGA Express* y *ModelSim XE*.

3. Adaptación del entorno *Xfuzzy* y la herramienta *xfvhdl*

La versión 2.0 de *Xfuzzy* fue concebida y desarrollada como un sistema abierto, tanto por la utilización de componentes software de libre distribución, como por el hecho de que el propio entorno está disponible bajo licencia GNU [5]. Esta versión de *Xfuzzy* puede compilarse sobre sistemas operativos tipo Unix (en particular, sobre las distintas distribuciones de Linux). Las diferentes herramientas pueden ejecutarse de forma independiente desde la línea de comandos o a través de una interfaz gráfica de usuario, basada en X-window, que proporciona menús con las opciones y comandos más habituales.

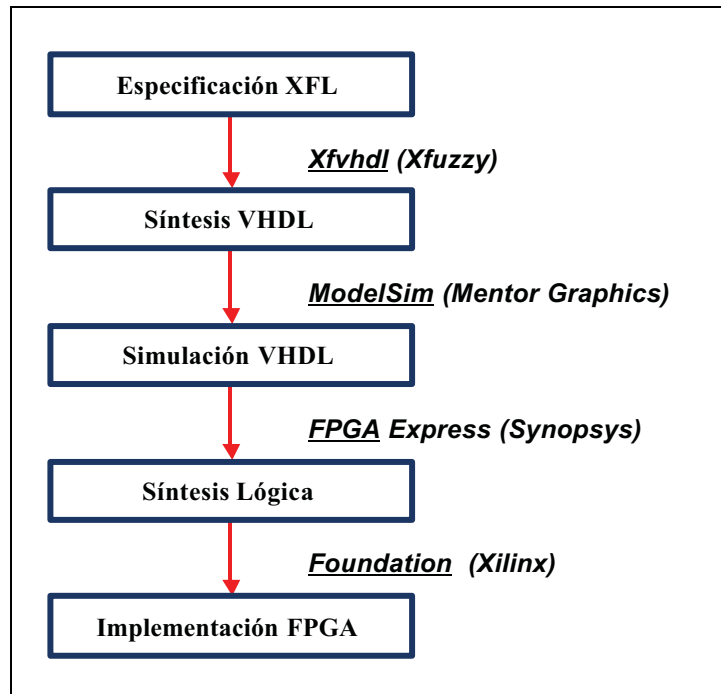


Figura 1: Flujo de diseño y herramientas CAD.

La utilización de versiones MS-Windows del software de diseño de FPGAs nos obligó, sin embargo, a desarrollar una versión de *Xfuzzy* para dicho sistema operativo, con objeto permitir un flujo de diseño unificado. Por este motivo, la última actualización (2.1) de este entorno de CAD permite generar versiones de *Xfuzzy* totalmente operativas sobre MS-Windows con ayuda de las herramientas *Cygnus Windows (Cygwin)* [6]. *Cygwin* proporciona un entorno Unix sobre Windows consistente en dos partes: una librería dinámica que actúa como un emulador Unix y una colección de herramientas que aportan la apariencia de los sistemas Unix/Linux. *Cygwin* funciona con todas las versiones de MS-Windows a partir de Windows 95.

Al seguir esta estrategia se obtienen dos grandes ventajas. Por una parte, se minimiza el esfuerzo necesario para portar la aplicación al nuevo sistema operativo. El 99% del código es totalmente compatible ya que sólo es preciso considerar algunas diferencias como la estructura de los sistemas de ficheros o los nombres de dispositivos. Por otra parte, la aplicación Windows aprovecha todas las utilidades de los entornos Unix (comandos, utilidades como *make* o *configure*, etc) sin cambiar en absoluto la filosofía de operación con respecto a la versión original.

Además de los comandos y utilidades ya mencionados, *Cygwin* proporciona una implementación del entorno de ventanas X-Window (*Xfree86*) que incluye un servidor X, las librerías correspondientes (*Xlib*) y la mayoría de los clientes X tradicionales (*xterm*, *xhost*, *xon*, *xset*, etc). Utilizando la librería *Xlib* es posible compilar una versión MS-Windows de *Xfuzzy* cuya apariencia y funcionalidad son 100% compatibles con las de las versiones Unix.

Coincidiendo con la adaptación de *Xfuzzy* al nuevo sistema operativo, también se ha procedido a la actualización de sus herramientas de síntesis hardware a las últimas versiones de las herramientas de Synopsys y Xilinx. La Figura 2 muestra la apariencia de la interfaz gráfica de usuario de *xfvhdl*. En la parte superior de la ventana se definen las opciones que controlan la generación del código VHDL (dimensionado de los buses, técnica de almacenamiento de antecedentes y generación de código VHDL para simulación y/o síntesis). Los distintos campos de la parte inferior de la ventana permiten definir las opciones de síntesis: nivel de esfuerzo, tipo de optimización, generación de reports, herramienta de síntesis utilizada (*FPGA Express* para MS-Windows ó *FPGA Compiler 2* para Unix), dispositivo elegido y localización de la librería de módulos.

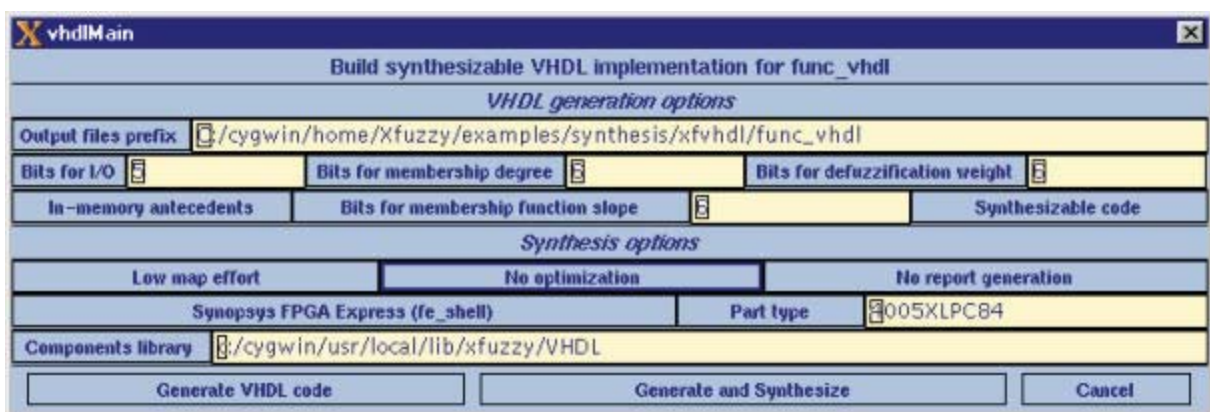


Figura 2: Interfaz gráfica de usuario de la herramienta *xfvhdl*.

A modo de ejemplo, la Figura 3 muestra distintas etapas del flujo de diseño de un controlador difuso: 1) la representación gráfica de la superficie de control del sistema proporcionada por las herramientas de *Xfuzzy*; 2) el fichero de comandos generado por *xfvhdl* para dirigir el proceso de síntesis llevado a cabo por *FPGA Express*; y 3) una visualización gráfica del contenido de la FPGA proporcionada por la herramienta de *floorplanning* de Xilinx.

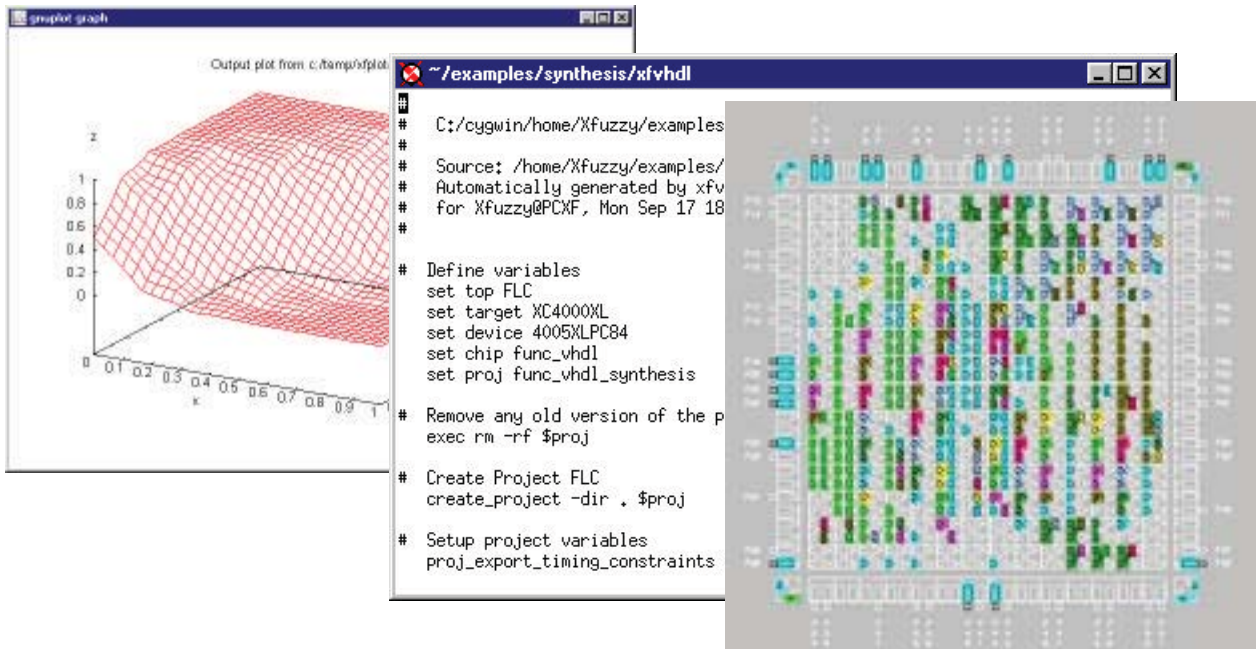


Figura 3: Interfaz gráfica de usuario de la herramienta *xfvhdl*.

4. Conclusiones

Combinando una versión actualizada del entorno de desarrollo *Xfuzzy* (generada con ayuda de las herramientas *Cygwin*) con versiones educativas de herramientas comerciales de simulación y síntesis, se ha definido un flujo de diseño sobre MS-Windows que permite automatizar el proceso de síntesis de sistemas difusos mediante FPGAs.

Referencias

- [1] I. Baturone, A. Barriga, S. Sánchez-Solano, C.J. Jiménez, y D. López, *Microelectronic Design of Fuzzy Logic-Based Systems*, CRC Press (2000).
- [2] E. Lago, M. A. Hinojosa, C. J. Jiménez, A. Barriga y S. Sánchez-Solano, *FPGA implementation of fuzzy controllers*, Conference on Design of Circuits and Integrated Systems (1997).
- [3] D. R. López, C. J. Jiménez, I. Baturone, A. Barriga y S. Sánchez-Solano, *Xfuzzy: A design environment for fuzzy systems*, IEEE Int. Conference on Fuzzy Systems (1998).
- [4] <http://university.xilinx.com/>
- [5] <http://www.fsf.org/copyleft/gpl.html/>
- [6] <http://sources.redhat.com/cygwin/>