Placa Prototipo para el Desarrollo de Test Automatizados de Circuitos en FPGAs

Sánchez M, Jiménez R.

Escuela Politécnica Superior, Huelva, España,
msraya@diesia.uhu.es

Resumen. Se ha llevado a cabo el desarrollo de una placa de prototipo basada en *FPGA* debido a la eficacia presentada por estos dispositivos para el desarrollo de prototipos que implementen nuevos diseños y algoritmos. Como principal novedad, en la placa se ha incluido un sistema autónomo de prueba que, además de la verificación funcional, permite la medición de parámetros importantes para el diseño. Como aplicación se ha implementado un banco de prueba para la validación de circuitos aritméticos.

1 Introducción

En la actualidad existe un elevado número de placas prototipo de *FPGA* comerciales. No obstante, estas placas están dedicadas básicamente a verificación funcional de prototipos [1]. Por otro lado y desde un punto de vista práctico, una vez realizada la implementación de un circuito en una *FPGA* resulta importante la medición de ciertos parámetros que nos ofrecen valores de mérito de nuestro diseño, con la finalidad de una posible implementación final en *ASIC*. Entre estos parámetros se encuentran la latencia del circuito (ns), el *throughput* (MB/s), el consumo (W) y el ruido generado (dBm). En cualquier caso, en la mayoría de placas existentes, la medida de ciertos parámetros de mérito del diseño se ve dificultada por la imposibilidad de variar la frecuencia de la señal de reloj, o por la inexistencia de puntos de medida de consumo de potencia o de señales de alta velocidad [2]. La principal novedad que aporta el diseño de esta placa es la posibilidad de extraer parámetros dinámicos del funcionamiento del diseño sobre las *FPGAs*.

Dada la cantidad de medidas que se pueden necesitar se ha visto la necesidad de una automatización de éstas. Esta automatización se lleva a cabo mediante un microcontrolador. Este microcontrolador mantiene un sistema de archivos donde se almacenan los ficheros de configuración de las *FPGAs* y las memorias *RAM*, así como los patrones de prueba y los resultados de las pruebas. Este sistema de ficheros se mantiene mediante una memoria de tipo *FLASH EPROM*. Tradicionalmente las *FPGAs* han sido orientadas al campo síncrono. Los circuitos asíncronos están aumentando en interés por las potenciales ventajas de las que gozan frente a los síncronos. Este creciente auge ha llevado a algunos investigadores a implementar estos sistemas asíncronos en *FPGAs*, con fines de construcción de prototipos y no de producto final [3]. La medición de los parámetros de funcionamiento de estos sistemas resulta especialmente complicada, debido a que la metodología de prueba resulta ligeramente diferente de la habitual empleada en circuitos síncronos [4].

La serie de *FPGAs* empleada en esta placa prototipo es la XC4000XL de *XILINX* que podría considerarse obsoleta, pues ya no se incluye en las últimas revisiones de software del fabricante. Sin embargo, consideramos que para entender las posibilidades de las *FPGA* en general y en particular en el campo de la docencia, los dispositivos de la serie XC4000 todavía juegan un papel muy importante y no están obsoletos. No obstante, también se puede emplear la familia *Spartan* que es compatible a nivel eléctrico.

En el apartado 2 se describe la descripción física del hardware incluido en la placa, su estructura y su funcionamiento básico. En el apartado 3 se describe la estructura y funcionamiento del circuito de prueba implementado en la *FPGA*. En el apartado 4 se describe la estructura y manejo del *firmware* incluido en la placa prototipo. En el apartado 5 se ofrecen comentarios sobre los resultados obtenidos con el prototipo. En el apartado 6, por último, se comentan futuras mejoras y líneas de trabajo.

2 Placa prototipo

La implementación de la placa ha dado lugar a un trabajo en dos áreas. En primer lugar el desarrollo de un sistema hardware, entre cuyos componentes se encuentran *FPGA*, microcontrolador y sensores de medida. En segundo lugar, el desarrollo de un software para los procedimientos relativos al control y gestión del sistema.

2.1 Descripción Hardware

En la figura 1 se observa una foto de la placa desarrollada y en la figura 2 se refleja el diagrama de bloques del hardware integrado en la placa prototipo. Se han empleado las herramientas "*OrCAD Cadence*" [5] y el compilador "*PICC*" de *HI-TECH* [6] para el desarrollo del prototipo.

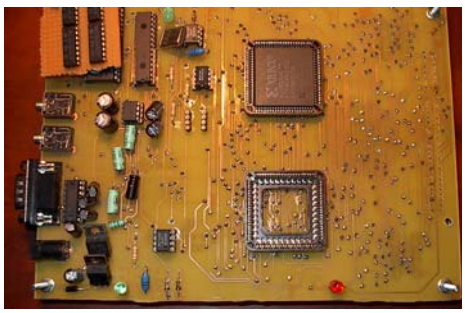

fig. 1. Fotografía de la placa prototipo

Las *FPGAs* instaladas, XC4010XL-PC84 disponen de 950 macroceldas con 10.000 puertas lógicas equivalentes [7]. Estos dispositivos disponen de 60 patillas de E/S que se encuentran conectados a los periféricos y microcontrolador, existiendo además varias líneas que interconectan ambas *FPGAs*. Las *FPGAs* disponen de interfaz serie para la programación en el sistema. La memoria *SRAM* de alta velocidad se puede emplear para guardar

tablas, para almacenaje intermedio de coeficientes o de *cache* de datos/instrucciones. Un oscilador programable en placa proporciona una base de tiempo. El oscilador es programable y puede generar una señal de reloj comprendida entre 200KHz y 100MHz con el fin de realizar las pruebas a diversas frecuencias de funcionamiento.

La placa incluye también un microcontrolador *RISC* [8] que es el encargado de programar la *FPGA* y de realizar tareas de control. El microcontrolador se encuentra conectado a una memoria *FLASH EPROM* que almacena el *bitstream* de configuración de las *FPGAs* y contenido de las memorias *SRAM*. El microcontrolador es el encargado de programar las *FPGAs* y puede monitorizar su estado para realizar medidas de rendimiento, programar la *SRAM* o modificar la frecuencia del generador de reloj. Por último, el *CODEC* de audio incluido con dos canales de entrada y dos de salida simultáneos, junto con un amplificador, habilita la placa para prueba de sistemas de procesado de señal de audio.

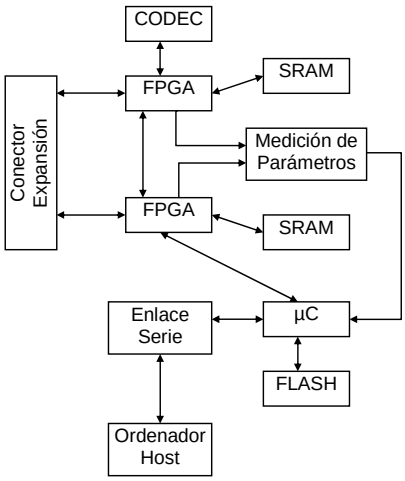

fig. 2. Estructura del sistema de procesamiento

2.2 Funcionamiento básico

El funcionamiento básico del sistema se establece mediante la comunicación con el ordenador *HOST* (generalmente un *PC*) del cual se descargan los ficheros de configuración de las memorias y los ficheros de entrada y salida del circuito a comprobar. Estos ficheros quedan finalmente almacenados en el sistema de ficheros proporcionado por la *FLASH EPROM*.

La descarga suele ser un proceso lento, la programación de la *FLASH EPROM* consume al menos 10 microsegundos por byte, siendo el tamaño medio de los archivos unos 32Kbytes, el proceso puede llegar a tardar unos 3 minutos. Además, el proceso de grabación de un byte requiere la realización de tres ciclos de escritura en direcciones diferentes, lo que ralentiza aun más la carga del registro de desplazamiento que genera las direcciones.

Sin embargo la programación de la *FPGA* es un proceso más rápido puesto que la lectura de la *FLASH EPROM* se realiza en un solo acceso y tiene un tiempo de ciclo máximo de unos 140 ns. El microcontrolador de la placa prototipo programa a continuación la o las *FPGAs* que se encuentran conectadas en "*daisy-chain*".

Un fichero especial contiene un listado de las pruebas a realizar con el nombre del fichero de configuración de *FPGA* y *RAM*, velocidad de reloj y fichero de salida. Este fichero de salida se compara con el generado por la *FPGA* en su proceso para considerar la ejecución del circuito bajo prueba como correcta.

2.3 Sensores incorporados

Existen sensores diferentes para cada *FPGA*, por lo que es posible dividir un diseño en dos componentes asimétricos y evaluar el consumo y la señal de ruido generado para cada dispositivo. La figura 3 muestra un diagrama esquemático de los sensores y su conexión.

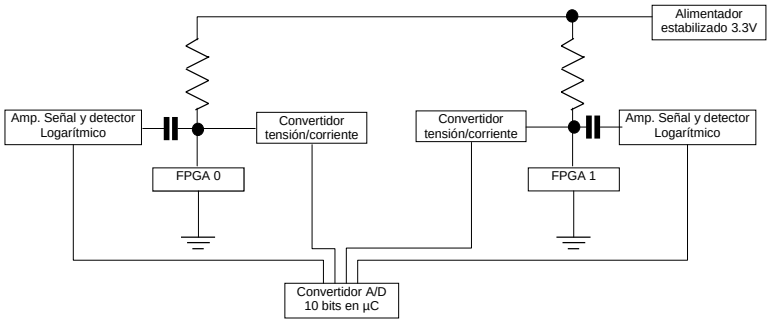

fig. 3. Diagrama de sensores analógicos

En la línea de alimentación de las *FPGAs* se inserta una resistencia con un pequeño valor óhmico que provoca la caída de una tensión proporcional al consumo de corriente del componente. Este valor debe ser lo suficientemente pequeño para que no perturbe el funcionamiento del dispositivo. Medidas experimentales aconsejan valores del orden de unos pocos ohmios. La tensión que cae en esta resistencia se amplifica y se mide mediante el convertidor A/D integrado en el microcontrolador. Además, es en el punto de alimentación de los dispositivos donde se acumula la mayoría del ruido de conmutación proveniente de los amplificadores de corriente y *CLBs* internos, cuya amplitud es dependiente de la actividad del circuito en pruebas.

La señal extraída del punto de alimentación consiste en pulsos de corta duración y tiene componentes de frecuencias elevadas, múltiplos de la señal de reloj. Una vez acoplada en continua se procesa mediante un circuito amplificador y medidor logarítmico de precisión que extrae una señal continua introducida en el convertidor A/D del microcontrolador. El rango medido es de -95 dBm a +20 dBm aproximadamente hasta unos 500MHz [9].

3 Banco de Test implementado en FPGA

La automatización de las medidas implica el diseño de un circuito independiente al que se quiere medir. Este circuito obtendrá los datos necesarios para las pruebas de los ficheros de configuración o "*bitstream*". Cada uno de estos contiene la descripción del circuito a probar junto con un pequeño circuito encargado de la carga inicial de los punteros de las zonas de memoria de la *RAM* a procesar por el diseño y la longitud de los vectores de prueba.

Una vez configurada la *FPGA*, el microcontrolador accede a través de un puerto de 8 bits y realiza la carga mediante la escritura de sucesivos *nibbles* de cuatro valores de 16 bits que almacenan los punteros de inicio de dos vectores de prueba y el puntero destino del vector de salida del circuito y por último de la longitud de los vectores de prueba, ver las figuras 4 y 5.

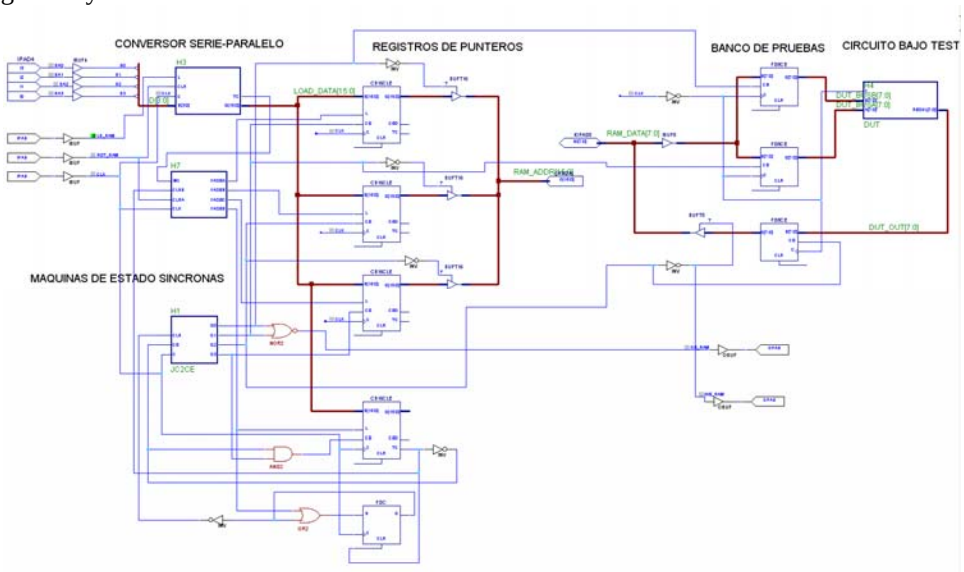

Fig. 4. Banco de pruebas para la carga de punteros de vectores de prueba en RAM y ejecución de la operación del circuito bajo medida

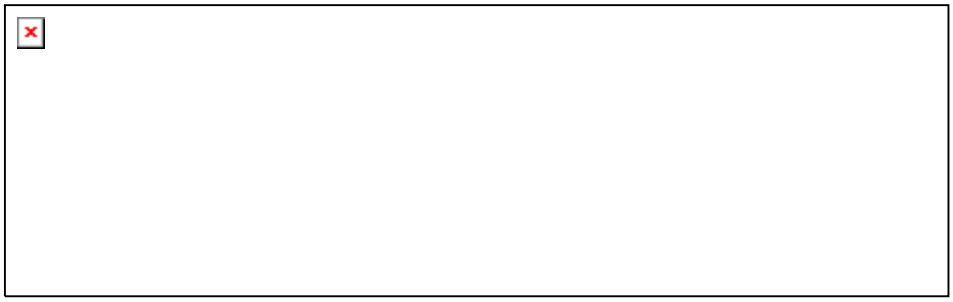

Fig. 5. Simulación de carga de 4 dobles palabras y ejecución para tres vectores de prueba

En la figura 5 se observa un cronograma, donde a partir de las señales LD_RAM y RST_RAM se cargan sucesivos *nibbles* para formar tres punteros de 16 bits donde se encuentran los vectores de prueba de entrada y de salida del circuito que se está probando, además de un valor de 16 bits que contiene el número de muestras que se procesan restado de FFFFH. En el caso de la figura, este valor es FFFCH para tres muestras.

Una vez cargados estos valores, la maquina secuencial empieza a leer de la memoria RAM los datos a procesar y entregarlos al dispositivo bajo prueba (DUT), escribiendo a continuación los datos procesados en la memoria RAM, a través de las señales OE_RAM y WE_RAM. Al finalizar con los vectores de prueba, el dispositivo queda a la espera de realizar una nueva prueba.

Se pueden calcular los siguientes parámetros: *throughput*, corriente media consumida, nivel de ruido de la línea de alimentación en dBm medio y tiempo empleado en el cálculo. Una vez finalizada la prueba puede reprogramar el generador de reloj, de modo que se pueda realizar una nueva prueba con el mismo circuito. Cuando el microcontrolador finaliza las pruebas con el circuito escribe los resultados en un fichero y pasa al siguiente circuito si lo hay.

El sistema de prueba es completamente autónomo una vez programados los ficheros de configuración en la *FLASH*. Esta característica hace que el prototipo pueda emplearse para otras aplicaciones que requieran la posibilidad de cambiar el circuito sin necesidad de un sistema *HOST*, de forma independiente en función de algún parámetro que controle el microcontrolador, o sea, en aplicaciones donde sea necesario reconfigurar el dispositivo.

4 *Firmware* del Microcontrolador

La figura 6 muestra el diagrama de bloques del *software* incluido en el microcontrolador. El funcionamiento es totalmente autónomo, su programación se realiza a través de un interprete de comandos disponible mediante la interfaz serie *RS-232*. Conectándonos en modo terminal es posible interactuar con el sistema de ficheros de forma que se pueda gestionar su mantenimiento, la carga desde el *HOST* y la descarga. El desarrollo del software se ha realizado a partir de notas de aplicación de *XILINX* [10] y del software de control de las placas *XS40* de *X Engineering Software Systems Corp* [11].

Para llevar a cabo las transferencias de ficheros se emplea un protocolo de tramas de 16 *bytes* junto con un *byte* de *CRC* en formato binario. El programa empleado para la transferencia de ficheros ha sido implementado empleando *C++ BUILDER*. Este programa también tiene la posibilidad de reprogramar el microcontrolador en el sistema, si es necesario actualizar su *firmware*.

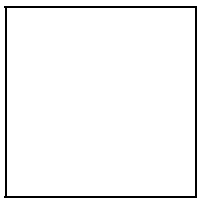

Fig. 6. Estructura del *firmware* del microcontrolador

Las funciones de programación se pueden introducir en un fichero de procesamiento por lotes o "*script*" que el microcontrolador se encargará de ejecutar. De esta forma es posible automatizar los comandos de programación de la *FPGA*, *RAM* y GENERADOR de RELOJ.

5 Conclusiones

Nos encontramos en las últimas fases de desarrollo del prototipo que hemos empleado principalmente para la evaluación de consumo de potencia de varios tipos de circuitos aritméticos con la intención de evaluar modelos de consumo de potencia [12] [13] y también de generación de ruido en función de la actividad del circuito.

La tabla 1 muestra el consumo del circuito para tres circuitos básicos de complejidad pequeña, mediana y grande. El consumo en reposo y en programación del dispositivo es muy similar, siendo de unos 4 mA. De la tabla se deduce que el escalado de consumo no es proporcional a la frecuencia de reloj. Este consumo se ha medido mediante una resistencia de 2 ohmios, manteniéndose la tensión de alimentación en los márgenes que recomienda el fabricante.

Ocupación	20 Mhz	50Mhz	80Mhz
Baja (10%)	16 mA	24 mA	32 mA
Media (58%)	31 mA	58 mA	80 mA
Alta (97%)	42 mA	70 mA	120 mA

Tabla 1. Medida experimental de consumo de corriente de alimentación para circuitos de diferente ocupación implementados en el dispositivo XC4010PC84-3. El número de transistores que conmutan a la velocidad de reloj es aproximadamente proporcional a la ocupación del circuito.

Se han desarrollado unas 2000 líneas de código C, para el programa que ejecuta el microcontrolador y 800 líneas de C++ para el programa de descarga de ficheros y se ha comprobado que la velocidad de programación de la FLASH EPROM es muy lenta, lo cual hace poco práctica la descarga de muchos ficheros de configuración. Se pretende cambiar este tipo de memoria por otro más rápido que facilite la carga de ficheros. Se ha cambiado el tipo de microcontrolador varias veces para acomodar el número creciente de funcionalidades que se le han incluido a la placa prototipo. Actualmente se emplea un PIC 18F252 al

50% de uso de código (32Kbytes totales) y 30% de RAM (1500 bytes totales) funcionando a una velocidad de reloj de 40MHz.

6 Líneas futuras

Se plantea una actualización de la FPGA integrada en la placa y ya estamos realizando gestiones para incorporar algún miembro de la familia VIRTEX que aportarían entre otras mejoras el aumento de la densidad de la lógica interna y el aumento del número de patillas externas permitiendo añadir más memoria y buses de interconexión más anchos. Podemos pensar en un diseño mejorado que contenga más sensores, más sensibles y mejor desacoplo de alimentación, para realizar medidas más fiables. También se ha pensado en la posibilidad de añadir un CODEC de video para poder procesar imágenes de video y así trabajar con algoritmos de procesamiento de señal de mayor nivel.

Otras aplicaciones futuras abarcan el concepto de hardware reconfigurable y sistemas en chip (systems on chip, "SoC"). Se pretenden realizar implementaciones de procesadores sencillos de tipo RISC integrados en FPGA como el xr16 [14] para la evaluación de coprocesadores en aplicaciones de procesado de señal.

Referencias

1. Optimagic Inc: FPGA Boards. http://www.optimagic.com/boards.html
2. H.W.Johnson, M. Graham: High-Speed Digital Design – A Handbook of Black Magic. Prentice-Hall (1993)
3. S.W.Moore P. Robinson: Rapid Prototyping of Self-timed Circuits. Proceedings of ICCD'98, (1998)
4. E. Brunvand: Using FPGAs to Implement Self-Timed Systems. Journal of VLSI Signal Processing nº 6. (1993) 173-190
5. Cadence : Orcad Cadence User's Manual. Publicado por Cadence (2000)
6. HI-TECH Inc.: PICC Compiler, User's Manual. Publicado por HI-TECH (2002)
7. Xilinx Inc: The Programmable Logic Data Book. Publicado por Xilinx (2000)
8. Microchip Inc: Microcontrolers Data Book. Publicado por Microchip (1999)
9. Analog Devices: AD8307 Data Sheet, Rev A. Publicado por Analog Devices (1999)
10. Xilinx Inc: Application Note XAPP502, Using a Microprocessor to Configure Xilinx FPGAs via Slave Serial or SelectMAP Mode. Publicado por Xilinx (2002)
11. XESS Corporation: XSTOOLs Source Documentation, v 3.0. Publicado por XESS (2000)
12. Xilinx Inc: Application Brief XBRF014, A Simple Method of Estimating Power in XC4000XL FPGAs. Publicado por Xilinx (1997)
13. Eduardo Boemo, José A. Herrera, Juan Meneses: Logic Depth and Power Consumption in Self-Timed Circuits: A Case-Study. Proc. XIII DCIS Conference (Design of Circuit and Integrated Systems). (1998)
14. Jan Gray: Designing a Simple FPGA-Optimized RISC CPU and System-on-a-Chip. DesignCon' 2001. (2001)