

DISEÑO DE PLACAS DE PROTOTIPADO CON FPGAs VIRTEX Y APLICACIÓN PARA LABORATORIOS DOCENTES

Gerardo Leyva¹, Felipe Rizo², Carlos Carreras¹, Octavio Nieto-Taladriz¹

¹*Universidad Politécnica de Madrid. {gleyvah, carreras, nieto}@die.upm.es*

²*Universidad Autónoma de Aguascalientes. frizo@correo.uaa.mx*

RESUMEN

En este documento presentamos una propuesta para el desarrollo de placas entrenadoras de bajo coste y de fácil uso para el prototipado rápido de circuitos digitales basadas en las familias de FPGAs Xilinx VIRTEX y VIRTEX-E con una capacidad máxima de 400K puertas. La configuración de las placas se realiza en modo serie a través del puerto de comunicaciones RS-232 en sistemas operativos Windows o Linux y también a través de puerto JTAG. Las placas tienen previsto puertos de encadenamiento serie durante la descarga del archivo de configuración para diseños cuya magnitud impliquen el uso de más de una FPGA. Los costos de las placas entrenadoras que se presentan son en promedio 50% del costo de las placas comerciales similares. Con esta propuesta se ha puesto en funcionamiento un laboratorio con 30 placas de bajo coste y sin necesidad del cable de configuración del fabricante.

1. INTRODUCCIÓN

Los avances tecnológicos demandan de las Universidades la continua actualización de los planes de estudio así como de los materiales didácticos de sus laboratorios. En este documento presentamos una propuesta para el desarrollo de placas entrenadoras de FPGA de bajo costo y de fácil uso para el prototipado rápido de circuitos digitales. Las placas entrenadoras propuestas son una plataforma de prototipado de diseño electrónico digital basado en HDL, orientado a estudiantes o docentes que deseen experimentar con FPGAs Xilinx [1] sin grandes necesidades de recursos.

Las placas entrenadoras contienen un microcontrolador de bajo costo AT89C2051 [5,6], el cual realiza la interfaz entre el ordenador y la FPGA. El proceso de descarga del archivo de configuración de la FPGA puede ser efectuado por cualquier software de comunicaciones serie (RS-232) a una velocidad de 57600 Baudios. La metodología de configuración que se propone ha sido empleada en dos modelos de placas entrenadoras con FPGAs Xilinx basadas en las familias XC4000 y SPARTAN con capacidad máxima de 20K [10,11] y también en placas basadas en FPGAs VIRTEX Y VIRTEX-E hasta 400k puertas con 150 I/O con niveles de voltaje que pueden ser seleccionados por el usuario (3,3 voltios y 2,5 voltios) y un costo aproximado de 200 euros. El método aquí descrito es válido también para las familias más recientes de FPGAs Xilinx, como las familias VIRTEX-II/PRO [14] y SPARTAN-III [15].

Con esta propuesta se ha puesto en funcionamiento un laboratorio para dar servicio a 200 estudiantes, con 30 placas con FPGA de muy bajo precio y sin necesidad del costoso cable JTAG [9] de configuración del fabricante. Además, la configuración de la placa puede ser realizada desde sistemas operativos Linux o Windows ya que la descarga del fichero de configuración se realiza a través de la interfaz RS-232 resolviendo con ello también la problemática de configuración a través del puerto paralelo en ordenadores personales.

2. DESCRIPCIÓN DE LA PLACA

La placa basada en FPGAs VIRTEX y VIRTEX-E está diseñada para ser configurada en varios modos. En este modelo de placa se puede emplazar cualquier FPGA con capacidad menor o igual a 400K puertas de las familias antes mencionadas con encapsulado TQFP 240.

Las partes principales que integran la placa son: FPGA Xilinx, microcontrolador, memoria de configuración EEPROM, chip de interfaz RS-232, LEDs de estatus, cristal oscilador y puertos serie de configuración. La figura 1 muestra una fotografía real de la placa entrenadora.

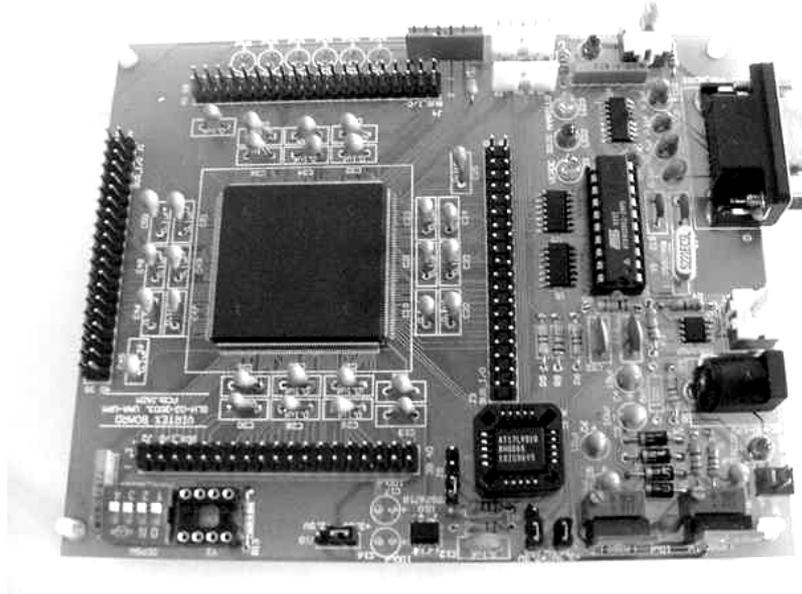


Fig. 1. Placa entrenadora de FPGAs VIRTEX y VIRTEX-E.

El fichero de configuración necesario para programar las placas debe ser generado en el sistema de desarrollo de Xilinx en formato MCS86 [13]. El microcontrolador incluido en la placa tendrá como tarea principal interpretar el fichero y configurar la FPGA además de facilitar al usuario el uso de la placa. En la placa se ha dispuesto de un “bus” serie entre el microcontrolador, la memoria EEPROM y la FPGA que además permite su interconexión con las FPGAs de otras placas a través de sus puertos serie de configuración.

2.1. El microcontrolador

La razón principal por la cual se ha puesto un microcontrolador en la placa entrenadora es para disponer de un elemento que sirviera de intérprete entre el ordenador y la FPGA, haciendo de este modo transparente para el usuario la descarga de ficheros. Los recursos del microcontrolador (puerto serie, temporizadores, puertos externos de ocho bits) son esenciales en esta placa para controlar la funcionalidad de la FPGA y de la memoria EEPROM, para exhibir siempre el estatus de la placa y para facilitar al usuario el manejo de la misma.

El programa del microcontrolador está escrito en lenguaje ensamblador para aprovechar al máximo los ciclos de ejecución del programa y, por lo tanto, para programar la FPGA a la máxima velocidad que permite el puerto serie interno.

En resumen, Las funciones del microcontrolador son las siguientes:

- Recibe el fichero de configuración de la FPGA vía puerto serie RS-232.
- Monitoriza la transmisión.

- Convierte bytes ASCII a formato binario.
- Monitoriza los bits de estatus de la FPGA durante la programación.
- Programa y borra la EEPROM
- Configura la FPGA
- Borra la FPGA y se prepara para la descarga de un nuevo fichero con solo presionar un pulsador.

2.2. Formato del fichero de configuración

El formato del fichero de configuración aceptado por las placas entrenadoras es el MCS86 [13], que puede ser generado por el Sistema de desarrollo de Xilinx [12], tanto en su versión de estudiante como en la versión completa. Cada una de las líneas de los ficheros en formato MCS86 se compone de las siguientes partes: inicio de línea, número de bytes, dirección de memoria, separador, código y “checksum”. En la figura 2 se muestra un ejemplo del formato MCS86.

```
:020000020000FC
:10000000FFFFFFFF5599AA660C000180000000E089
:0CA5C00000000000000000000000000008F
:00000001FF
```

Fig. 2. Ejemplo de formato MCS86

Estos elementos son analizados por el microcontrolador lo que le permite, por ejemplo, identificar errores de comunicación en cualquier momento del proceso de descarga a la FPGA. La figura 3 muestra el algoritmo para procesar el formato MCS86 y obtener los bytes de configuración de la FPGA.

- 1) Inicio
- 2) Detecta inicio de línea.
- 3) Identifica número de datos.
- 4) Desecha dirección.
- 5) Desecha byte de separación.
- 6) Procesa datos de configuración:
 - a) Recepción del carácter (“nibble” alto).
 - b) Conversión ASCII – Hexadecimal.
 - c) Recepción del carácter (“nibble” bajo).
 - d) Conversión ASCII – Hexadecimal.
 - e) Concatena ambos “nibbles” para obtener el byte de configuración.
 - f) Envío vía serie síncrona de byte de configuración a FPGA.
- 7) Identifica y calcular checksum
- 8) Si existe error en el estatus de FPGA o checksum notificar y terminar.
- 9) Identifica secuencia de fin de fichero. En caso de presentarse ir a punto 10, de lo contrario ir a punto 2.
- 10) Fin

Fig. 3. Algoritmo para obtener los bytes de configuración de la FPGAs Xilinx.

2.3. La memoria de configuración serie para modo maestro

La placa entrenadora incluye una memoria de configuración para la FPGA. La memoria seleccionada para esta aplicación es la EEPROM AT17LV010 [7] del fabricante ATMEL, que permite programar y borrar la memoria en la placa sin necesidad de un circuito programador extra. Para facilitar el funcionamiento de la placa, se ha diseñado un “bus” serie entre el microcontrolador, las líneas de configuración de la FPGA y la memoria de configuración. Cuando la placa es configurada en modo esclavo, el microcontrolador obliga a la memoria a colocar su patillaje en alta impedancia con lo cual ésta no interfiere en el proceso de configuración de la FPGA. Cuando la placa es configurada para trabajar en modo maestro, el microcontrolador es puesto en alta impedancia y la FPGA busca automáticamente su código de configuración en la memoria EEPROM. En el momento que se aplica voltaje a la placa, el usuario puede seleccionar si el fichero que se recibirá del ordenador será enviado a la FPGA o a la memoria EEPROM.

La programación (escritura) de la memoria EEPROM se realiza por páginas de 128 bytes de longitud. Para llevar a cabo la escritura es necesario formar una instrucción serie que deberá contener las siguientes partes: condición de inicio, selección de lectura – escritura, dirección de escritura, 128 bytes de datos y finalmente condición de paro. La dirección de escritura está contenida en el fichero de descarga en formato MCS86 descrito en secciones anteriores y el proceso es transparente para el usuario.

2.4. Monitorización

Una de las características especiales de las placas que se muestran en este artículo es la facilidad que brinda a los usuarios de monitorizar su estado de operación. Para este fin dispone de un conjunto de LEDs que indican en todo momento el estado de operación de la placa. Los indicadores de estado son: READY, ERROR, RX y DONE.

2.4.1. Ready

Cuando está encendido indica que la FPGA está lista para ser configurada; cuando está apagado y el LED RX está intermitente indica que esta recibiendo el fichero de configuración de la FPGA; cuando está intermitente y el LED DONE encendido indica que la FPGA fue configurada correctamente.

2.4.2. Rx

Muestra la actividad del puerto serie durante la descarga de un fichero de configuración de la FPGA.

2.4.3. Error

Sólo se enciende si se ha efectuado un fallo en la transmisión de datos del ordenador a la placa o cuando la FPGA notifica que los bits de configuración no son correctos

2.4.4. Done

Sólo se enciende cuando la FPGA está configurada apropiadamente.

3. PROGRAMACIÓN DE LA FPGA

Las FPGAs Xilinx pueden ser configuradas de diversos modos. Aunque el JTAG es uno de los modos de configuración más ampliamente usado para prototipado, se han aprovechado para la configuración las características de los modos esclavo – serie, maestro – serie y encadenamiento – serie (“*serial chain*”), todos ellos soportados por las familias de FPGAs Xilinx [1]. El microcontrolador de la placa proporciona las señales CCLK, PROG, INIT y DIN necesarias para configurar la FPGA en modo serie. Durante todo el proceso de configuración, la señal INIT de la FPGA es monitorizada permanentemente para asegurar que el proceso de descarga es correcto.

Además, las placas que aquí se presentan están preparadas para aprovechar la programación de FPGAs en modo encadenamiento serie. Cada placa tiene un puerto de configuración de entrada serie y otro de salida serie (adicionales al puerto RS-232) de forma que si se tiene la necesidad de segmentar un diseño cuya magnitud implique el uso de más de una FPGA, pueden ser encadenadas varias placas para programarse simultáneamente en una sola descarga de código y con una sola conexión al puerto RS-232 del ordenador, tal como se muestra en el diagrama de bloques de la figura 4. En este modo de conexión es necesario generar un fichero único que contenga la información de todas las FPGAs que serán configuradas. Este proceso deberá efectuarse en el Sistema de desarrollo de Xilinx (PROM FILE FORMATTER).

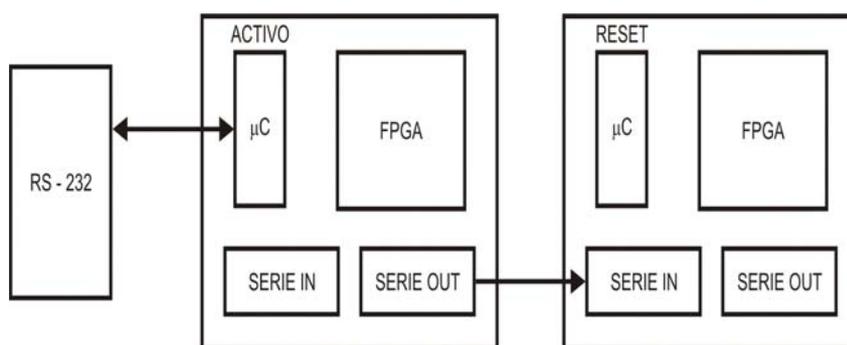


Fig. 4. Encadenamiento serie de las placas entrenadoras.

La placa basada en FPGAs VIRTEX y VIRTEX-E está diseñada para ser configurada en modo JTAG, además de los modos de configuración anteriormente descritos. En este modo es necesario poner el microcontrolador de la placa en modo “*reset*” y conectar el cable JTAG en el puerto correspondiente.

El método aquí presentado para configuración serie también es válido para los modelos más recientes de FPGAs Xilinx, como lo son VIRTEX-II/PRO [14] y SPARTAN-III [15], debido a que éstas poseen las características de programación descritas anteriormente y además porque la interfaz eléctrica entre el microcontrolador de la placa y las FPGAs está preparada para soportar niveles de voltaje entre 5 y 1.2 voltios [16].

4. OTROS MODELOS DE PLACAS DE PROTOTIPADO

En esta sección comparamos nuestra propuesta con otros modelos comerciales conocidos.

Quizá [8] sea la más completa de las placas de prototipado (y la de mayor coste) porque incluye hardware adicional como LEDs, exhibidores de 7 segmentos, circuitos codificadores

de audio y vídeo; conectores de audio, vídeo y comunicaciones y memoria FLASH y SRAM. Sin embargo, su manejo no es el más sencillo en comparación con las otras placas porque su configuración se realiza en varios pasos (configuración de CPLD, configuración de FPGA, borrado de CPLD) y porque hay varios “buses” compartidos entre diversos dispositivos. En nuestra placa la configuración se lleva a cabo en un solo paso y para realizar una nueva descarga basta con presionar un pulsador. Además, todo el hardware adicional es externo a la placa, logrando con ello más flexibilidad.

En [3] se presenta una placa prácticamente sin hardware adicional a la FPGA y en formato PC-104. Sus interfaces de entrada – salida son de 3,3 voltios y puede ser configurada en diversos modos. La ventaja de nuestra propuesta es que el usuario puede configurar las interfaces de entrada – salida a 3,3 voltios ó a 2,5 voltios, se pueden utilizar FPGAs de las familias VIRTEX o VIRTEX-E y tiene mucho menos coste.

La placa presentada en [4] está diseñada para mantenerse conectada al “bus” PCI del ordenador, con lo cual no es fácil realizar en ella prácticas de laboratorio que requieran resultado visibles por el usuario.

Finalmente la placa presentada en [2] contiene hardware básico adicional, se configura a través de puerto JTAG y aunque tiene un precio similar a nuestra propuesta, sin embargo, sólo tiene la mitad de puertas lógicas equivalentes (200K).

En la tabla 1 se resumen los costes en Euros y el número de puertas de los modelos de placas mencionados anteriormente.

| Modelo | FPGA | Puertas | Coste | Página WEB |
|------------------|----------|---------|-------|---|
| ADS-XLX-SP2E-EVL | Xc2s200e | 200K | 220 | http://www.silica.com/eval_kits/adx-20020215.html |
| Aps-v240-xcv300e | Xcv300e | 411K | 687 | http://www.associatedpro.com/v240revb.PDF |
| Xc2s200_eval | Xc2s200 | 200K | 349 | http://www.cesys.com/english/ebene2/product_p.htm |
| XSB-300E | Xc2s300e | 300K | 781 | http://www.xess.com/prod032.php3 |
| Diseño Propio | Xcv300e | 411K | 200 | |

Tabla 1. Modelos de placas comerciales.

5. EXPERIENCIA EDUCATIVA

En este apartado se cuenta como se desarrolló el proceso de actualización de los contenidos de las asignaturas y las prácticas de laboratorio relacionadas con sistemas digitales. El proceso se llevó a cabo en dos etapas, comenzando con placas de bajo coste y con asignaturas básicas. En una segunda etapa se abordó el diseño de placas más complejas y la actualización de asignaturas de diseño digital avanzado.

5.1. Antecedentes

El objetivo principal que motivó a la realización de este trabajo fue la modernización de la enseñanza de las asignaturas relacionadas con los principios y prácticas de diseño digital combinacional y secuencial. Estos temas son abordados en dos cursos separados con un laboratorio asociado para la realización de prácticas en cada uno de los cursos. La tecnología de soporte para dichos cursos estaba basada en las familias estándar TTL de circuitos lógicos y en la tecnología de PLDs borrables y programables eléctricamente (GAL). En cada laboratorio se realizan alrededor de 10 prácticas y un proyecto final por curso. En cuanto a los circuitos estándar TTL la problemática estaba relacionada con el alto número de componentes que se requerían para las prácticas, así como el gran espacio de prototipado que era necesario

en placas de inserción o placas de “*wire-wrapping*”. En cuanto los PLDs, el lenguaje hardware seleccionado para el diseño fue CUPL, de Logic Devices (actualmente distribuido de forma gratuita por ATMEL). El uso de PLDs disminuía considerablemente el número de circuitos integrados necesarios para las prácticas, pero el problema principal que se tenía era la programación de estos dispositivos ya que requerían de un programador especial y nuestro laboratorio solo disponía de un número limitado de ellos.

5.2. Actualización de contenidos de asignaturas y laboratorios

Tres factores nos motivaron para intentar realizar un cambio en la enseñanza de circuitos digitales: la posibilidad de adquirir con relativa facilidad y bajo coste circuitos integrados FPGAs, la abundante información en las nuevas ediciones de libros de texto y la posibilidad de obtener de forma gratuita el entorno de desarrollo de fabricante. En una primera etapa desarrollamos placas de prototipado basadas en FPGAs Spartan de Xilinx de bajo coste [10,11] intentando que fuesen fáciles de usar por los estudiantes y que éstas pudieran ser configuradas en cualquier ordenador que tuviese instalado el sistema de desarrollo. La fotografía de nuestras primeras placas se muestran en la figura 5.

Paralelamente al diseño de los prototipos se actualizaron los programas de estudio, intentando mantener los conceptos y contenidos básicos de las asignaturas pero modificando los apartados referentes a la tecnología a emplear en las prácticas y en los ejercicios de clase. Se incluyó en los cursos la información necesaria sobre el lenguaje de programación VHDL y las nociones básicas de tecnologías basadas en FPGAs. Asimismo fueron rediseñadas las prácticas de laboratorio basando su realización sólo en el lenguaje de programación VHDL y en FPGAs. Esta primera fase de actualización de las asignaturas fue satisfactoria porque los estudiantes asimilaban muy bien la nueva metodología de diseño y además porque utilizaron adecuadamente los nuevos materiales diseñados para tal fin. Después de un año de uso, sólo fue necesario reparar el 15% de las placas, lo cual es aceptable.

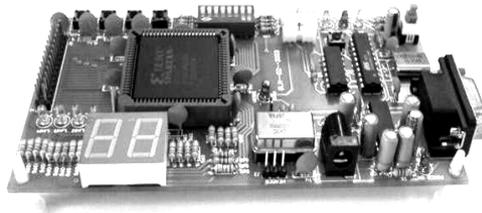


Fig. 5. Placa de prototipado modelo Spartan.

5.3. Fase final de actualización de asignaturas

Una vez concluida la primera etapa de actualización de asignaturas y probados los beneficios de las nuevas tecnologías, hemos desarrollado nuevas placas que permiten la realización de prácticas de diseño de sistemas digitales complejos que requieren varios cientos de miles de puertas equivalentes, como por ejemplo, pequeños microprocesadores o circuitos de procesamiento digital de señal. Así, considerando también que los estudiantes han desarrollado previamente habilidades en el manejo de FPGAs, se han seleccionado dispositivos de tecnología más avanzada (VIRTEX/VIRTEX-E), de mayor coste y que implican un mayor cuidado en su manejo. Se tiene planeado que las nuevas placas que se presentan en este trabajo, sean utilizadas en cursos universitarios a partir de septiembre en las asignaturas de Arquitectura de Ordenadores y Diseño Digital Avanzado. En estas nuevas placas se ha mantenido el concepto de facilidad de uso y la posibilidad de ser utilizadas en

cualquier ordenador que tenga el sistema de desarrollo necesario. Para un futuro se planea tener un solo tipo de placa que de servicio a todas las asignaturas de diseño digital, un plan que por supuesto merece ser estudiado con detalle.

6. CONCLUSIONES

La elaboración de placas entrenadoras de bajo costo es una opción viable para implementar laboratorios de FPGAs en las Universidades y también lo es para estudiantes, profesores y público en general que deseen experimentar con FPGAs Xilinx sin grandes necesidades de recursos.

A pesar de que existen en el mercado placas entrenadoras de alta tecnología con FPGAs de alta densidad [2-4, 8] consideramos que la placa entrenadora que aquí se presenta, en conjunto con sistemas de desarrollo de Xilinx en versiones gratuitas, ofrece el soporte necesario para el prototipado de sistemas digitales de densidad baja - media en un entorno amigable. Con esta propuesta se ha puesto en funcionamiento un laboratorio con 30 placas con FPGA, de muy bajo precio y sin necesidad del costoso cable JTAG [9] de configuración del fabricante. Además, la configuración de la placa puede ser realizada desde sistemas operativos Linux o Windows ya que la descarga del fichero de configuración se realiza a través de la interfaz RS-232 resolviendo con ello también la problemática de configuración a través del puerto paralelo en ordenadores personales.

7. BIBLIOGRAFÍA

- [1] Xilinx Databook DS060(V1.7) June 27, 2002.
- [2] http://www.silica.com/eval_kits/adx-20020215.html
- [3] www.associatedpro.com/v240revb.PDF
- [4] <http://www.cesys.com/english/ebene2/productp.htm>
- [5] Atmel Data Handbook, 2002.
- [6] http://www.atmel.com/dyn/resources/prod_documents/DOC0368.PDF
- [7] http://www.atmel.com/dyn/resources/prod_documents/doc2321.pdf
- [8] <http://www.xess.com/prod032.php3>
- [9] <http://www.toolbox.xilinx.com/docsan/xilinx4/data/docs/pac/cables6.html>
- [10] G. Leyva *et al*, "Tarjeta didáctica para prácticas de circuitos digitales en FPGA", Proc. Computación Reconfigurable & FPGA's, UPS Universidad Autónoma de Madrid, Eds. Boemo, Gómez, López-Buedo, Sutter, pp. 457-462, España, Septiembre de 2003.
- [11] G. Leyva *et al*, "Tarjeta didáctica para implementación de circuitos digitales en FPGA" Proc. Reseach on Computing Science, CIC Instituto Politécnico Nacional, Eds. Díaz de León, González, Figueroa, pp. 291-294, México, Octubre de 2003.
- [12] The Programmable logic data book, Xilinx, 2000.
- [13] http://toolbox.xilinx.com/docsan/xilinx6/books/data/docs/dev/dev0126_19.html
- [14] Xilinx VIRTEX-II PRO, Platform FPGA Handbook, pp. 68-71, 2002
- [15] Xilinx Databook DS099-2(V1.2), pp.32-39, July 2003.
- [16] G. Leyva *et al*, "Design of prototyping boards with Xilinx FPGAs", Proc. European Workshop on Microelectronics Education (EWME'04), Suiza, Abril de 2004.