

DOCENCIA MULTIDISCIPLINAR CON SISTEMAS DE DESARROLLO PARA SoPC

RICARDO JOSÉ COLOM PALERO, RAFAEL GADEA GIRONES, JOAQUÍN CERDÁ BOLUDA,
VICENTE HERRERO BOSCH Y ANGEL TEBAR RUIZ

*Departamento de Ingeniería Electrónica. Escuela Técnica Superior de Ingenieros de
Telecomunicación. Universidad Politécnica de Valencia. España.*
rcolom@eln.upv.es

Este artículo describe las experiencias de los autores en el uso de los “system-on-a-programmable-chip (SoPC)” para la realización y dirección de proyectos final de carrera de los estudiantes de Ingeniería de Telecomunicación. Los alumnos de la Escuela de Telecomunicación reciben una formación en sistemas digitales en la que se desarrollan conceptos de diseño basado en microprocesadores, microcontroladores y procesadores digitales (DSPs), diseño digital con “fiel-programmable-gate-array (FPGAs)” mediante lenguajes de descripción hardware (HDLs) y programación en ensamblador y C/C++. Como proyecto de finalización de los estudios, todos estos conocimientos pueden ponerse conjuntamente en práctica en la realización de un diseño de un SoPC utilizando FPGAs. Los estudiantes deben conjuntar todo lo aprendido y enfrentarse a un nuevo reto en el que se tiene un flujo de diseño nuevo con una nueva herramienta. ¿Serán los alumnos capaces de realizar el diseño propuesto utilizando un SoPC, con los conocimientos actuales? ¿Cuan bueno será el diseño realizado? ¿Por qué no utilizar un SoPC como plataforma única de aprendizaje de sistemas basados en microprocesadores y de diseño mediante FPGAs con HDLs?

1. Introducción

El continuo avance de la tecnología de semiconductores, ha hecho que los fabricantes de FPGAs (Field Programmable Gate Arrays) estén en constante evolución hacia la creación de nuevas familias de dispositivos. El drástico incremento de la complejidad y de la arquitectura interna de los dispositivos ha implicado un avance considerable hacia un nuevo producto que sirve como plataforma para la creación de diseños de elevada complejidad. El surgir de estas nuevas tecnologías han permitido al diseñador disponer de FPGAs de gran tamaño en las que además de tener gran número de elementos lógicos y de memoria se puede poner el “core” de un procesador de IP (Intellectual Property) para implementar un computador con hardware a medida para aplicaciones “System-on-a-Chip” (SoC) [1] - [4]. Esta nueva aproximación del diseño digital se han denominado con el termino de “System-on-a-Programmable-Chip” (SoPC) [5]. En los últimos años diferentes empresas han comercializado y popularizado “cores” de procesadores RISC, lo cual ha provocado que el proceso de diseño de un sistema hardware basado en procesadores entre en el dominio del diseño software. En este artículo los autores realizarán un repaso de los “cores” de procesador que se pueden utilizar para la enseñanza, así como las placas disponibles para el uso de esos “cores”, se presentará el flujo de diseño que se debe utilizar en este proceso, se analizará como la introducción de estas nuevas tecnologías pueden afectar al proceso de aprendizaje y se terminará mostrando un proyecto estudiantil realizado mediante estas técnicas de diseño.

2. Repaso Tecnológico

2.1. Cores de procesador para SoPC

En las FPGAs se pueden distinguir dos tipos de “cores” de procesadores: los “hard” y los “soft”. Los procesadores hard son “cores” de procesadores que se encuentran empotrados en la propia estructura de la FPGA junto con los habituales elementos lógicos. Añadir un procesador hard en una FPGA es una solución híbrida que permite tener un dispositivo con las características de un ASIC

(Application-Specific Integrated Circuit) pero con las ventajas de la reconfigurabilidad de las FPGAs. En el mercado se pueden encontrar diferentes fabricantes con dispositivos con diferente número de procesadores empotrados. Por ejemplo, Xilinx en sus familias Virtex II Pro y Virtex 4 incluye hasta dos “cores” del procesador PowerPC en un “chip”. Altera puso un “core” de un procesador ARM en la familia APEX 20KE, que fue comercializado con el nombre de Excalibur, pero que actualmente ya ha sido descatalogado sin ofrecer una alternativa. Cypress vende una variante de los sistemas SoPC, el denominado “Programmable-System-on-a-Chip” (PSoC), que esta formado básicamente por un “core” del procesador M8C más bloques lógicos configurables para implementar interfaces periféricos, entre los que se encuentran convertidores analógico-digitales y digitales-analógicos, temporizadores, contadores y UARTs.

Los procesadores soft utilizan los elementos lógicos programables de la FPGA para implementar el procesador en el interior de la misma. Tal como lo hacen los procesadores Nios de Altera y MicroBlaze de Xilinx. Los procesadores soft tienen unas muy buenas características y una gran flexibilidad, permiten al diseñador especificar el tamaño del “data path”, la funcionalidad de la ALU, el número y tipo de periféricos y los parámetros del mapa de memoria durante el proceso de compilado. Sin embargo, tanta flexibilidad supone un coste adicional, ya que los procesadores soft trabajan con ciclos de reloj más lentos y requieren de mayor potencia que sus equivalente hard.

Con el precio actual de las FPGAs, añadir un procesador soft representa muy poco más dinero y se tiene suficiente espacio en el dispositivo para realizar diseños de aplicaciones específicas. Tradicionalmente los dispositivos SoC han sido ASIC que ofrecían muy buenas características y prestaciones, pero que requerían un tiempo muy elevado de desarrollo para su realización. Para que los estudiantes puedan realizar una implementación hardware de un diseño, el uso de sistemas SoPC basados en FPGAs es sencillo rápido, pequeño y más económico.

Para poder realizar diseños basados en sistemas SoPC, se requiere de herramientas software adicionales y específicas para cada procesador soft, las cuales permiten configurar las diferentes opciones que tiene el procesador y el hardware de entrada y salida necesario. Estas herramientas generan un modelo HDL sintetizable del “core” del procesador en VHDL o en Verilog. Además, al procesador se le puede añadir un sistema lógico adicional, siendo el diseño resultante sintetizable utilizando herramientas estándares de síntesis para FPGAs. El programa de la aplicación que correrá en el procesador se puede escribir en C o C++, habitualmente, ya que las herramientas del procesador soft disponen del compilador necesario, así mismo también se puede realizar la programación en lenguaje ensamblador.

2.2. Hardware para el Desarrollo de SoPC

Tanto Altera como Xilinx disponen de placas para la realización de diseños SoPC [1], [5] - [6], que incluyen las herramientas software necesarias para el desarrollo de diseños. La “Altera Nios development board” se basa en el uso de la FPGA EP1C20 de la familia Cyclone. El kit de desarrollo incluye un conjunto amplio de herramientas, ejemplos y tutoriales para la realización de diseños SoPC.

Algunos “third-party vendors” disponen de software de ayuda al desarrollo de sistemas basados en el procesador Nios. Para aquellas personas que requieren de sistemas operativos basados en tiempo real, el kit de Nios incluye el sistema operativo MicroC/OS II. Especial interés presenta la posibilidad de correr el kernel del μ C Linux en la placa.

Altera dispone también de versiones de este kit de desarrollo en el que utiliza otras FPGAs como por ejemplo la EP2C35 de la familia Cyclone II, la EP1S10 y la EP1S40 de la familia Stratix o la EP2S60 de la familia Stratix II. Sin embargo, dentro del programa universitario de Altera se encuentra la intención de descatalogar todas estas placa con la finalidad de unificar sus kit educacionales y de desarrollo en una nueva placa. La “Altera development and education board (DE2)” mostrada en la Figura 1 se basa en el uso de la FPGA EP2C35 de la familia Cyclone II con 35.000 elementos lógicos.

Entre sus características se pueden destacar: Un número suficiente de interruptores, LEDs y display 7-segmentos, un display LCD de 16 x 2 caracteres, memorias Flash, SRAM y SDRAM, interface PS/2 y RS232, entrada y salida de audio y microfono (CODEC de audio de 24 bits), Decodificador de TV, DAC de 10 bits para VGA, conector USB 2.0 y Ethernet, Puerto de infrarrojos y tarjeta SD. El kit de desarrollo incluye un conjunto amplio de herramientas (Quartus II Edition y Nios II Embedded Processor), ejemplos y tutoriales para la realización de diseños SoPC.

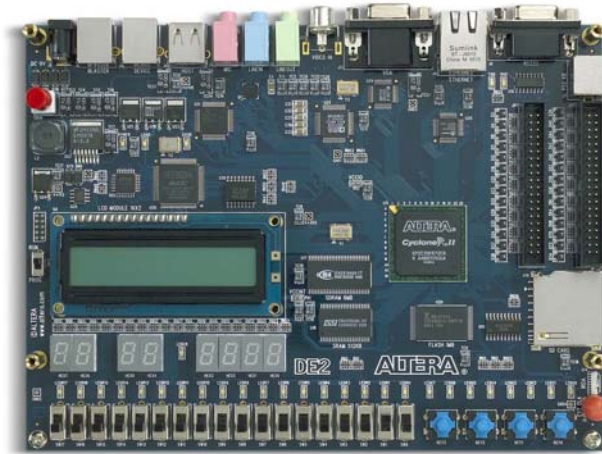


Figura 1. “Altera’s Development and Education board”.

3. Herramientas para el Diseño de SoPC

3.1. Flujo de Diseño Tradicional

El flujo de diseño que tradicionalmente se ha utilizado para el diseño de sistemas digitales basados en FPGAs, comienza con una entrada del diseño mediante esquemáticos, mediante lenguajes de descripción hardware (HDL) o combinada, continua con la síntesis del diseño y el emplazamiento y rutado en el dispositivo, concluyendo en la programación de la FPGA para realizar la verificación del diseño [3], [5], [7]. Los fabricantes de FPGAs proporcionan las herramientas de CAD necesarias que conducen al diseñador paso a paso a través de este flujo de diseño, por ejemplo el software Quartus II de Altera o el ISE de Xilinx. En la Figura 2 se muestra el flujo de diseño tradicional que se utiliza para realizar diseño de sistemas basados en FPGAs. Son necesarias herramientas de síntesis, emplazamiento, rutado y programación, las cuales en el caso de Altera y Xilinx se encuentran implementadas con los mismos nombres en sus herramientas CAD (Quartus II e ISE).

3.2. Herramientas de configuración de “cores” de procesadores

Actualmente existe un gran número de “cores” de procesadores disponibles en varias fuentes. En la Web (por ejemplo: www.opencores.org) se pueden encontrar “cores” de procesadores con licencia pública, mientras que las compañías privadas suministran sus propios procesadores, Altera (Nios) o Xilinx (MicroBlaze). Este artículo se centra en el uso de “cores” de procesadores proporcionados por los fabricantes de FPGAs o en aquellos de características similares suministrados por terceras fuentes.

Los “cores” de procesadores que ofrecen los fabricantes de FPGAs se encuentran optimizados para funcionar adecuadamente en las FPGAs de las familias que ellos recomiendan, con lo cual son mucho mas eficientes que las que un diseñador-estudiante puede llegar a realizar. Además los fabricantes proporcionan una gran cantidad de herramientas y soporte que facilitan considerablemente su uso, incluyendo compiladores de alto nivel personalizados para sus “cores”.

En la Figura 3 se muestra la modificación del flujo de diseño tradicional adaptado para el diseño con SoPC, que se ajusta al caso particular de Altera y Xilinx. Realizado en un “user-friendly graphical user interface (GUI)” el diseñador puede particularizar con facilidad el procesador para una aplicación determinada. Entre los parámetros que se pueden configurar se encuentran: el tamaño del “data path”, la memoria, el espacio de direccionamiento y los periféricos (incluyendo la definición de la arbitración de entrada-salida, UARTs, controladores Ethernet y controladores de memoria). Una vez se han establecido los parámetros en el interfase GUI, la herramienta genera el “core” del procesador en un fichero HDL (en el caso de Altera) o en un fichero netlist (en caso de Xilinx). Estos ficheros se pueden unir junto con diseños HDL tradicionales utilizando las herramientas de CAD. El siguiente paso consistirá en compilar (síntesis, emplazamiento, rutado, etc.) el diseño completo (el “core” del procesador y la lógica adicional del diseñador). Tras lo cual se puede programar la FPGA, a partir del fichero generado, utilizando las herramientas habituales. En este punto el diseño hardware ha concluido.

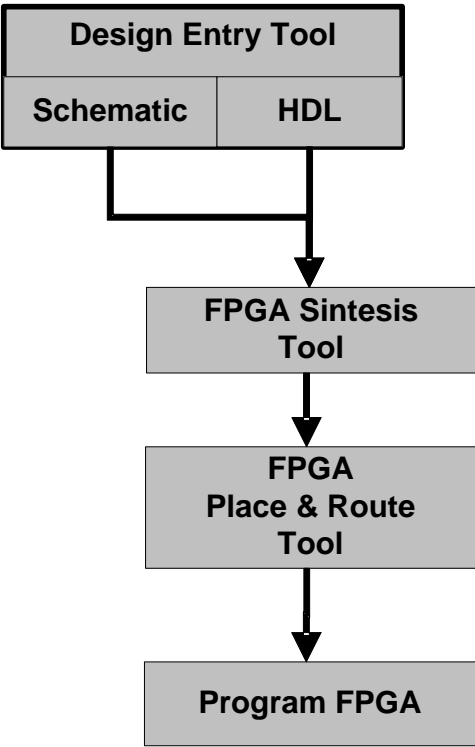


Figura 2. Flujo de diseño tradicional en sistemas basados en FPGAs.

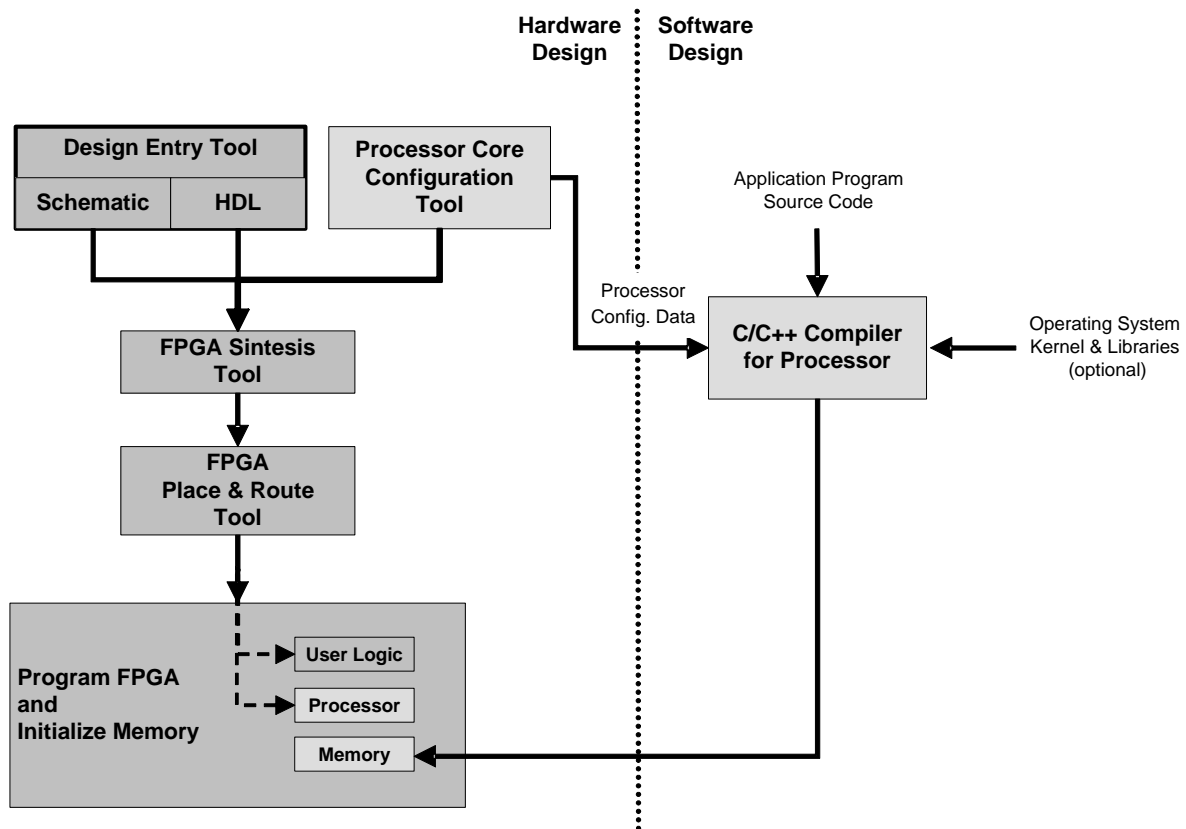


Figura 3. Flujo de diseño para sistemas SoPC.

3.3. Compilador para “cores” de procesadores

Tal como se muestra en el lado derecho de la Figura 3, el siguiente paso consiste en escribir el software que se va a ejecutar en el procesador soft y compilarlo. Cuando la herramienta de configuración del procesador genera el fichero HDL o netlist, también crea unos ficheros de librería con las cabeceras de C particularizadas para el procesador específicamente definido. Los procesadores incluyen un compilador de C/C++ que permiten al diseñador realizar programas “stand-alone”. Sin embargo, el diseñador puede compilar el código para un sistema operativo que funcione con el procesador. La “Altera Nios development board” incluye el sistema operativo MicroC/OS-II, que permite trabajar en tiempo real. También se puede optar por adquirir una versión del μ Clinux u otros operativos suministrados por terceras personas.

3.4. Inicialización de la memoria

Cuando el programa se compila se obtiene un fichero binario que debe ser cargado en la memoria del procesador. El proceso de carga se puede realizar de diferentes maneras dependiendo de la configuración de memoria que tenga el procesador.

- 1) “On-Chip Memory”: Si el programa de aplicación es pequeño, se puede poner en los bloques de memoria que tiene internamente la FPGA, en este caso el programa debe cargarse en la memoria cuando se realiza la configuración hardware del dispositivo. Sin embargo, la memoria interna de las FPGAs no es excesivamente grande y por tanto limita el tamaño de la aplicación, en consecuencia no es una opción que se suele utilizar.
- 2) “Bootloader”: Durante el proceso de realización de un programa, este será modificado y probado muchas veces, hasta alcanzar la versión definitiva. En este caso lo más práctico es disponer de un programa “bootloader” en la memoria interna de la FPGA que inicie el proceso de carga. Este programa es bastante pequeño, con lo cual cabe en la memoria interna de la

FPGA. Su función consiste en recibir el fichero binario, cargarlo en la memoria externa y comenzar la ejecución del código. Así, un nuevo código de programa se puede cargar en la memoria externa a través de un puerto serie u otra interfase, sin necesidad de recargar de nuevo la configuración de la FPGA. Altera incluye un “bootloader” en su procesador Nios, que denomina GERMS, el cual proporciona un entorno con capacidades bastante limitadas de depuración, además de las funciones básicas del “bootloader”. Xilinx tiene un depurador denominado XMDstub, que es capaz de cargar el fichero binario a través del puerto serie, almacenarlo en memoria y comenzar la ejecución.

- 3) “External Nonvolatile Storage”: El código del programa de aplicación se puede almacenar en una memoria externa no volátil, ya sea EEPROM, Flash o de cualquier otro tipo. El programa de la aplicación puede preprogramarse en un módulo de memoria externa o por medio de un “bootloader” cargarse en una memoria no volátil. Para aplicaciones que no requieran funcionar a elevadas frecuencias, se puede realizar la ejecución desde memoria externa. Sin embargo, las aplicaciones que funcionen a frecuencias elevadas se puede optar por utilizar tres tipos de memorias: La memoria interna de la FPGA, memoria externa no volátil o memoria externa volátil. Combinando el uso de estas tres memorias se puede obtener un diseño óptimo en velocidad. La memoria interna de la FPGA se inicializa con un “bootloader” que se encarga de controlar los movimientos del programa de aplicación entre las memorias. La memoria no volátil, por ser más lenta se utiliza para almacenar permanentemente el programa de la aplicación, mientras que la memoria volátil por su velocidad es adecuada para contener el programa de aplicación durante su ejecución. El “bootloader” puede modificarse para que en el “power-up” del sistema, transfiera el programa de la memoria no volátil a la memoria volátil (más rápida) y comience la ejecución desde esta última. Esto tiene las ventajas del almacenamiento permanente, la ejecución rápida y la facilidad para cambiar el programa de la aplicación en cualquier momento.

4. La docencia

El diseño de sistemas digitales es una parte muy importante de la enseñanza recibida en la titulación de Ingeniero de Telecomunicación de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Valencia [8]. A lo largo de los cinco años que duran los estudios, se dedica a la electrónica digital en general un total de seis asignaturas:

- 1) Electrónica digital: centra las bases para el estudio y desarrollo de sistemas digitales utilizando circuitos integrados estándares.
- 2) Sistemas Electrónicos Digitales: asignatura que se dedica al estudio de sistemas digitales basados en microprocesadores comerciales.
- 3) Diseño de Circuitos y Sistemas Electrónicos: diseño de sistemas basados en FPGAs, utilizando el flujo de diseño tradicional mostrado en la Figura 2. Se trata por primera vez el uso de los HDL, concretamente VHDL.
- 4) Circuitos Microelectrónicos: Diseño de sistemas digitales utilizando técnicas full-custom y semi-custom, para la realización de un ASIC.
- 5) Diseño de Circuitos Integrados Digitales. Esta asignatura es una continuación directa de la denominada Diseño de Circuitos y Sistemas Electrónicos, en la cual se profundiza en los conceptos de sistema, verificación y modelización utilizando Verilog como lenguaje HDL.
- 6) Sistemas Electrónicos Avanzados. Continuación de la asignatura de Sistemas Electrónicos Digitales, trata sobre el uso de microcontroladores comerciales y procesadores digitales de señal (DSP).

Estas asignaturas se pueden agrupar en dos bloques, en función del número de alumnos, las tres primeras asignaturas son comunes a todos los alumnos de la titulación y por ello tienen entre 300 y 400 alumnos. Las otras tres son asignaturas correspondientes a la intensificación de electrónica, por tanto el número de alumnos matriculados está alrededor de los 20. Esta clasificación determina distintas filosofías de abordar contenidos y de plantear las asignaturas. Para el caso de pocos alumnos

se opta por la realización de trabajos tutorizados como método de evaluación mientras que cuando hay muchos alumnos se recurre a un examen ya sea teórico, práctico o combinado.

En esta enumeración de asignaturas se puede encontrar una similitud entre ellas y el presente artículo, hay dos asignaturas dedicadas a los microprocesadores (microcontroladores, DSPs) y otras dos dedicadas al diseño con FPGAs, mientras que este artículo trata sobre diseño de sistemas SoPC. Actualmente, estas asignaturas disponen de sistemas de desarrollos independientes e inconexos para el estudio de los diferentes sistemas: Motorola para microprocesadores, Siemens para Microcontroladores, Texas Instruments para DSPs, Quartus de Altera e ISE de Xilinx para FPGAs. El uso de sistemas SoPC en educación permite la unificación de los sistemas de desarrollo en uno, único y común. Mediante un sistema SoPC se puede impartir docencia de diseño basado en FPGAs y de diseño de sistemas basado en un procesador, aunque en este caso se trate de un procesador soft. Esto presenta ventajas para los profesores como por ejemplo: la económica al sólo disponer de un sistema hardware de desarrollo para varias asignaturas, sólo hay que trabajar en un entorno lo cual facilita la labor de preparación de material de apoyo al eliminar la dispersión. También, tiene ventajas para el alumno: al facilitar el aprendizaje del alumno, ya que únicamente tendría que centrarse en un entorno de trabajo, que iría perfeccionando a lo largo del tiempo. Aunque presenta inconvenientes, ya que el uso de un único sistema para todo, actúa como canalizador cerrando la visión generalista hacia otros sistemas que pueden ser mejores o peores, pero que en definitiva existen y que forman parte del mercado.

La “Altera Nios development board”, esta pensada para el uso en sistemas SoPC, lo cual la hace adecuada para poder utilizar únicamente el procesador soft de Nios, que puede ser programado tanto en lenguaje ensamblador como en C/C++, como sustituto de los sistemas basado en microprocesadores y microcontroladores que actualmente se utilizan en la titulación. También, existe la opción de prescindir del procesador soft y utilizar la FPGA para realizar diseños digitales con la lógica interna mediante lenguajes HDL. Así con una sola placa se podría impartir la docencia de las asignaturas que tratan el diseño con FPGAs y lenguajes HDL y las que tratan el diseño con microprocesadores. La “Altera development and education board (DE2)” mostrada en la Figura 1, además de tener las mismas posibilidades que la “Altera Nios development board”, dispone de CODECs de audio y video lo cual permite orientar las aplicaciones hacia los DSPs. Así, con una sola placa y un entorno de trabajo único se puede impartir la docencia de cuatro asignaturas distribuidas a lo largo de la titulación de Ingenieros de Telecomunicación. Además, de este modo se podrían impartir conceptos sobre diseño de SoPC, que actualmente no se imparten.

Para poner en funcionamiento estas propuestas, algunas de las seis asignaturas planteadas anteriormente deben realizar una adaptación de sus contenidos, consistentes en:

- 1) Sistemas Electrónicos Digitales: asignatura que se dedica al estudio de sistemas digitales basados en microprocesadores comerciales. El microprocesador escogido para el estudio sería el NIOS II. Las clases prácticas se desarrollaría utilizando la “Altera development and education board (DE2)” mostrada en la Figura 1. La programación del microprocesador se puede realizar en lenguaje ensamblador. Altera dispone de un software de fácil utilización denominado “Altera Debug Client” que permite realizar el ensamblado de programas y la depuración de los mismos. Este software permite visualizar los registros del microprocesador, la memoria, realizar ejecuciones paso a paso, activar breakpoints para detener la ejecución del programa y otras muchas posibilidades.
- 2) Diseño de Circuitos y Sistemas Electrónicos: diseño de sistemas basados en FPGAs, utilizando el flujo de diseño tradicional mostrado en la Figura 2. Se trata por primera vez el uso de los HDL, concretamente VHDL. En este caso no se requiere ninguna modificación ya que esta asignatura ya utiliza la “Altera development and education board (DE2)” como elemento de verificación de los diseños.

- 3) Diseño de Circuitos Integrados Digitales. Esta asignatura es una continuación directa de la denominada Diseño de Circuitos y Sistemas Electrónicos, en la cual se profundiza en los conceptos de sistema, verificación y modelización utilizando Verilog como lenguaje HDL. Actualmente en esta asignatura se utiliza como entorno de diseño Xilinx, por tanto debería adaptarse a utilizar Quartus II con la “Altera development and education board (DE2)”. Además en esta asignatura se aborda como práctica la realización de un microprocesador, lo cual la convierte en una asignatura ideal para abordar el diseño de sistemas SoPC basados en FPGAs.
- 4) Sistemas Electrónicos Avanzados. Continuación de la asignatura de Sistemas Electrónicos Digitales, trata sobre el uso de microcontroladores comerciales y procesadores digitales de señal (DSP). Centrándose más en el diseño de aplicaciones para procesamiento digital de señal, esta asignatura puede abordar las herramientas que Altera dispone para tal finalidad como es el caso del DSP builder. La combinación de las herramientas de Matlab, Quartus II y el DSP Builder de Altera, puede dar como resultado la realización de diseños que funcionen en la “Altera development and education board (DE2)”, la cual tiene elementos para el tratamiento digital del audio y del vídeo.

Estas modificaciones permiten en cuatro asignaturas compartir recursos y herramientas de diseño, minimizando el impacto que el aprendizaje de una nueva herramienta supone para el alumno. Así mismo los profesores que imparten las asignaturas pueden compartir experiencias y desarrollar conjuntamente material para el desarrollo de las mismas.

5. Desarrollo de proyectos final de carrera con SoPC

Durante el último año los autores han utilizado placas de desarrollo de sistemas SoPC basados en FPGAs, para la realización de proyectos fin de carrera, con la finalidad de analizar, el comportamiento de los alumnos frente a este nuevo sistema y la posibilidad de utilizarla en la docencia de las asignaturas comentadas en el apartado anterior. La realización del proyecto final de carrera, es el trabajo de conclusión de los estudios de Ingeniero de Telecomunicación en el que el alumno debe poner en práctica los conocimientos adquiridos. Estos alumnos tienen una preparación previa en diseño digital, arquitectura de computadores, programación en C/C++, diseño con VHDL y Verilog, diseño con FPGAs y experiencia en las herramientas CAD asociadas con el flujo de diseño. Sin embargo, desconocen los sistemas SoPC y las herramientas de diseño para estos sistemas. Por tanto, será un reto para ellos enfrentarse con un nuevo entorno y una nueva filosofía de abordar los diseños, ya que hasta este momento han realizado diseños digitales utilizando FPGAs, también han realizado diseños basados en microprocesadores, pero nunca han trabajado con una combinación de ambos. De entre los diseños propuestos, el más exitoso ha sido la realización de un “Web Server” en el que los ficheros se ubiquen en una memoria “CompactFlash”.

El “Altera Nios Development Kit” contiene los elementos necesarios para poder realizar diseños basados en sistemas SoPC. Tiene una “Altera Nios development board” y las herramientas de desarrollo software necesarias para poder realizar diseños (Quartus II design software with SOPC builder system development tool, Nios embedded processor, and GNUPro Toolkit). Al instalar el software del kit se descubre que tiene tutoriales de ayuda y ejemplos de referencia, entre los que se puede encontrar un pequeño “Web server”.

Un “Web server” es un proceso capaz de comunicarse con uno o varios clientes remotos y de ofrecer los servicios que éstos le demandan. Este proceso implementa todos los protocolos de cada nivel necesarios de la arquitectura TCP/IP para su correcto funcionamiento. Los niveles de esta arquitectura son:

- 1) Aplicación: proporciona la comunicación entre procesos o aplicaciones de procesadores separados. El protocolo utilizado puede ser HTTP (protocolo para la transferencia de

hipertextos), SMTP (protocolo sencillo para la transferencia de correo electrónico), Telnet (protocolo para conexión remota), FTP (protocolo para la transferencia de ficheros). . .

- 2) Transporte: proporciona un servicio de transferencia de datos extremo a extremo. El protocolo será TCP o UDP.
- 3) Red: relacionada con el encaminamiento de los datos del computador origen al destino a través de una o más redes conectadas por dispositivos de encaminamiento.
- 4) Acceso a la red: relacionada con la interfaz lógica entre un sistema final y una subred.
- 5) Físico: define las características del medio de transmisión, la tasa de señalización y el esquema de codificación de señales.

La implementación del “Web server”, que Altera facilita en el “Nios Development Kit” incorpora una pequeña implementación de los protocolos de nivel de transporte “Transmission Control Protocol (TCP)” y del nivel de red “Internet Protocol (IP)” que se conoce con el nombre de “Lightweight IP (lwIP)”. El objetivo de esta implementación es reducir el uso de la memoria, siendo muy utilizado en sistemas embebidos. Sus características más importantes son:

- Protocolo IP con envío de paquetes a múltiples interfaces de red.
- “Internet Control Message Protocol (ICMP)” para el correcto mantenimiento de la red y “debugging”.
- “User Datagram Protocol (UDP).”
- Protocolo TCP con control de congestión, estimación de RTT, rápida recuperación y retransmisión.
- “Dynamic host configuration protocol (DHCP).”
- “Address resolution protocol (ARP)” para Ethernet.
- Estándar sockets para interfaces de programación del nivel de aplicación (API).

Todas estas opciones son configurables en el entorno de desarrollo de la aplicación. Esto ayudará a diseñar un servidor Web eficiente y eficaz desde el punto de vista de funcionamiento. Sin embargo, para utilizar la pila lwIP que Altera proporciona se necesita el sistema operativo MicroC/OS II, el cual solo se encuentra disponible en los “Nios Development Kit”. Este kernel provee los siguientes servicios: tareas o threads (hilos de ejecución), indicadores de evento, paso de mensajes, control de memoria, semáforos y control de tiempo. Ha sido empleado en cientos de aplicaciones desde su creación en 1992 y ha sido portado sobre 40 arquitecturas de procesadores distintas, entre ellas Nios.

El “Web Server” del ejemplo de Altera realmente se encuentra implementado en el nivel de aplicación. Sin embargo, debido a su sencillez presenta ciertos defectos:

- No tiene una gestión de los buffers de memoria eficientes.
- No soporta navegación por varias páginas. No hay una gestión eficaz del buffer de la cadena URL.
- La única página que se muestra en el servidor está almacenada en la memoria Flash, cuyo espacio está limitado a 8 Mbytes.
- Sólo puede enviar determinado tipo de archivos básicos. El número de tipos es muy reducido.
- Requiere que el programador de la página Web tenga conocimientos de cómo programar la FPGA, ya que cambiar la página Web almacenada no es trivial y se requieren conocimientos del hardware y de todos los programas utilizados para desarrollar el proyecto (Quartus, SOPC Builder y Nios IDE). Para cambiar la página, es necesario reprogramar la memoria Flash y saber utilizar un sistema de lectura de ficheros en formato zip.
- Cambiar la página requiere, además de lo descrito en el apartado anterior, tiempo para llevarlo a cabo y la existencia insitu del ordenador con el software necesario con las licencias para programar la placa. Hoy en día, el tiempo que se nos exige para un proyecto es cada vez

menor y nos obliga a optimizar la velocidad y los recursos para disminuir el tiempo de exposición al mercado. Por tanto, los recursos y el tiempo que se necesitan para llevar a cabo un proyecto son fundamentales.

- Utilización de funciones que no son seguras en la ejecución de un hilo. Esto es muy importante porque en un determinado momento pueden crear una excepción y que el programa deje de funcionar, siendo necesario realizar un reset. Este puede ser el caso de la función 'strtok'.

Una vez analizados los defectos de la aplicación “Web server” de Altera, se planteo la posibilidad de mejorarla, proponiendo para ello un proyecto fin de carrera. La principal mejora se centra en poner las páginas Web en una memoria CompactFlash, en lugar de la memoria Flash de la placa. Así, se incrementa la capacidad de almacenamiento de páginas Web, también se mejora la actualización de las páginas Web y la manejabilidad del sistema, ya que una vez programado el sistema SoPC en la placa, sólo hay que guardar la páginas Web en la memoria CompactFlash, se conecta a la placa y se reinicia el sistema. Las tareas que hay que llevar a cabo para alcanzar dicho objetivo son:

- 1) Crear el componente hardware de la CompactFlash para añadirlo a nuestro procesador, de tal manera que el “core” del procesador pueda leer datos en la CompactFlash.
- 2) Crear los drivers necesarios en C para acceder a la CompactFlash, así como el sistema de ficheros.
- 3) Modificación del software de la aplicación del “Web Server”, para mejorar sus características.
- 4) Desarrollar un servidor web de páginas HTML que se ejecute sobre Nios II utilizando los protocolos y el kernel que se proporcionan.

En el proyecto realizado, se han estudiado los niveles hardware y software con mucho detalle. Han sido necesarios conocimientos electrónicos, sobre todo de sistemas electrónicos digitales para crear las funciones de lectura y escritura de la CompactFlash. Al mismo tiempo, se ha requerido ciertos conocimientos de lenguajes de descripción hardware como puede ser VHDL o Verilog. La implementación del sistema de ficheros, así como los drivers de acceso a la CompactFlash y el servidor web, se han realizado en lenguaje C, estudiando en profundidad además un sistema de ficheros estándar como es FAT16. Este sistema de ficheros ha requerido la creación de un descriptor de ficheros propio para la CompactFlash. Por último, la creación de un servidor Web en lenguaje C ha requerido de conocimientos telemáticos a diversos niveles. Dado que el nivel de red y el nivel de transporte eran suministrados, la dificultad ha residido en completar el nivel de aplicación, junto a los sockets, de la manera más óptima posible.

6. Conclusiones

Este artículo pone de manifiesto las grandes posibilidades de las FPGA en la tecnología. Según avanzan los años, las principales empresas de diseño y fabricación de FPGA mejoran la velocidad y las prestaciones de estos dispositivos. En este caso, se ha podido realizar una aplicación de alto nivel, como es el servidor Web, gracias a la multitud de librerías y el “core” suministrados por el fabricante. Todo esto provoca al mismo tiempo la necesidad de incrementar los conocimientos de los programadores de FPGA. Hoy en día, las FPGA pueden ser utilizadas no sólo como prototipo para simular chips, sino para desarrollar aplicaciones y productos comerciales basados en esta tecnología.

El uso de las herramientas hardware y software basadas en SoPC, permiten unificar el desarrollo de los conocimientos y minimizar los costes de laboratorios, ya que los mismos recursos pueden ser compartidos por diversas asignaturas. Esto permite además centrar todos los esfuerzos del alumno en aprender solo una herramienta de trabajo, pero quizás también le impida tener una visión global del mercado que le pueda servir para analizar las diferentes opciones y elegir por sí mismo en un futuro.

Con la docencia actual un alumno tiene capacidad demostrada para enfrentarse exitosamente con la realización de un sistema basado en SoPC que utiliza un entorno de diseño a priori desconocido directamente por él. Esto podría considerarse como una justificación para permanecer como estamos, pero la tecnología avanza y el uso de sistemas SoPC se extiende, por tanto tenemos la obligación de preparar a nuestros alumnos lo mejor posible y por lo tanto intentar avanzar actualizando la docencia a las exigencias del mercado laboral.

Referencias

- [1] J.O. Hamblen, "Using an FPGA-based SOC Approach for Senior Design Projects", Proceedings of the 2003 IEEE International Conference on Microelectronic Systems Education, pp. 18-19, Jun. 2003.
- [2] J.W. Bruce, J.C. Harden, and R.B. Reese, "Cooperative and Progressive Design Experience for Embedded Systems", IEEE Transactions on Education, vol. 47, pp. 83-92, Feb. 2004.
- [3] S. Nooshabadi and J. Garside, "Modernization of Teaching in Embedded Systems Design – An International Collaborative Project", IEEE Transactions on Education, vol. 49, pp. 254-262, May. 2006.
- [4] Dong Lin and Shiyuan Yang, "An Implementation of Rapid Prototyping Platform of Embedded Systems", 2006 IEEE Tenth International Symposium on Consumer Electronics, pp. 1-4, Jun. 2006.
- [5] T.S. Hall and J.O. Hamblen, "System-on-a-Programmable-Chip Development Platforms in the Classroom", IEEE Transactions on Education, vol. 47, pp. 502-507, Nov. 2004.
- [6] J.O. Hamblen, "Using Second Generation SOPC Boards for Student Design Projects", Proceedings of the 2005 IEEE International Conference on Microelectronic Systems Education, pp. 69-70, Jun. 2005.
- [7] Matjaz Finc and Andrej Zemva, "Rapid HW/SW co-design of softcore processor systems", EUROCON 2003, The IEEE Region 8 Computer as a Tool, vol. 1, pp. 104-108, Sept. 2003.
- [8] J. Cerdá, M.A. Martinez, M.A. Larrea, R. Gadea, and R.J. Colom, "An Active Methodology for Teaching Electronic Systems Design", IEEE Transactions on Education, vol. 49, pp. 355-359, Aug. 2006.