

Sistema de co-diseño hardware/software basado en FPGA para la captura de video analógico a través del bus serie USB

Angel Jimenez Fernandez, Rafael Paz Vicente, Manuel Dominguez Morales, Elena Cerezuela Escudero, Miguel Angel Rodriguez Jodar

Dpto. Arquitectura y Tecnología de Computadores
Universidad de Sevilla

E.T.S. Ingeniería Informática. Avda. Reina Mercedes s/n
41012 Sevilla
ajimenez@atc.us.es

José Ignacio Villar de Ossorno

Dpto. Tecnología Electrónica

Resumen

Con la adaptación del Espacio Europeo de Educación Superior (EEES) y la creación de los nuevos títulos de grado, se plantea una nueva visión metodológica en la enseñanza superior.

En el caso de titulaciones relativas a las ingenierías informáticas y electrónicas resulta muy interesante el uso de metodologías orientadas al Aprendizaje Basado en Proyectos (ABP).

En esta exposición se presenta un proyecto en el que se ilustra a los alumnos como diseñar sistemas hardware / software. En particular proponemos el codiseño de un sistema capaz de capturar video analógico mediante el uso de un convertidor analógico/digital (ADC) y una FPGA, y además envíe la información capturada a través del bus USB usando un microcontrolador.

De esta manera esperamos que los participantes de la experiencia ABP adquieran conocimientos relativos a las metodologías para codiseño hardware/software, uso de FPGA, microcontroladores y ADC. Así como entrenar habilidades relativas a la descripción de circuitos lógicos mediante el lenguaje VHDL.

Summary

By adapting the European Higher Education Area (EEES) and the creation of new graduate degrees, there is proposed a new methodological view in higher education. In the case of studies related to computer and electronic engineering result very interesting to use the methodologies for Project Based Learning (PBL). In this paper, we present an exposition that shows students how to design systems hardware / software. In particular we propose the co-design a system capable of capturing analog video using an analog / digital converter (ADC) and a FPGA, and

also send the information captured through the USB bus using a microcontroller. In this way we hope that the participants of the PBL experience acquired knowledge of the co-design methodologies for hardware / software, use of FPGA, microcontroller and ADC, and also training skills related to the description of logic circuits using VHDL.

Palabras clave

Aprendizaje basado en problemas, trabajo en equipo, aprendizaje colaborativo, programación de microcontroladores, lenguaje C, ARM.

1. Motivación

La adaptación del Espacio Europeo de Educación Superior y, en particular, la implantación de las nuevas titulaciones de grado, el modelo educativo está experimentando un gran cambio. Evolucionamos desde un modelo centrado en la "enseñanza del profesor" a uno basado fundamentalmente en el "aprendizaje del alumno" [1].

Debido a esta evolución, la mayoría de las universidades han promovido planes de innovación docente que faciliten la implantación de nuevas metodologías por parte del profesorado.

En la Universidad de Sevilla, en el curso 2007, se puso en marcha el Plan de Renovación de las Metodologías Docentes 2007 [2]. Este plan estaba dividido en varias líneas promoviendo, entre otras cuestiones, la formación del profesorado en nuevas las metodologías docentes, implantación de dichas metodologías en las asignaturas, dotación de espacios docentes, etc.

Con la implantación de los nuevos grados, en muchos casos nos hemos encontrado con la novedad de la carga docente no presencial por

parte del alumno. Esto nos ha supuesto el planteamiento de actividades y tareas que ocupen esta carga por parte del alumno, así como la elaboración de mecanismos para su seguimiento y evaluación por parte del profesor. Las universidades han invertido una elevada cantidad de recursos en formar al profesorado en nuevas metodologías didácticas y de aprendizaje, sin embargo, poco se ha invertido en la formación en nuevos contenidos. Siendo este un aspecto crucial en los estudios de base tecnológica, ya que la tecnología está en continuo cambio y evolución, resultando crucial para el desarrollo tecnológico que los futuros profesionales posean conocimientos de la tecnología más actual posible, estando esta responsabilidad casi exclusivamente delegada al profesor.

En particular, en el ámbito de las ingenierías, el uso de las metodologías del Aprendizaje Basado en Proyectos (ABP) resulta muy adecuado para su uso en las partes no presenciales de las asignaturas. Está tratado ampliamente en numerosos trabajos [3,4,5] y ha demostrado ser eficaz en la enseñanza de las más diversas ramas de conocimiento [5]. Sin embargo, puede que no todo sea ventajoso. El uso del ABP en la carga no presencial de las asignaturas demanda un elevado nivel de motivación por parte del alumno, ya que es el propio alumno el que tiene que “obligarse” a participar en el ABP [7]. Para tratar de conseguir esa motivación extra, pensamos que se puede conseguir proponiendo proyectos que les sean muy atractivos.

2. El problema del material y los laboratorios

Gracias al I Plan Propio de Docencia de la Universidad de Sevilla en el curso 2008-2009 (línea 13: infraestructuras y equipamiento de espacios docentes) se adaptó un laboratorio para el desarrollo hardware para su libre uso por parte del alumnado, creando así un espacio docente adaptado a las necesidades de la implantación de metodologías basadas en ABP en materias que versan acerca del desarrollo de sistemas hardware.

Este laboratorio se equipó, entre otras medidas de seguridad, con un control de acceso controlado por la tarjeta identificativa de los estudiantes, de manera que los alumnos pertenecientes a las asignaturas participantes de experiencias ABP

están autorizados a su uso, pudiendo sólo acceder los alumnos adecuados. Además este sistema nos permite hacer un seguimiento acerca del uso individual de cada alumno del laboratorio. A diferencia de las prácticas simuladas y eminentemente software, en las que todo el material necesario es un ordenador, para el desarrollo de prácticas hardware suele ser necesario el uso de un material mucho más específico y costoso, como son fuentes de alimentación, multímetros, osciloscopios, generadores de señales, analizadores lógicos o de espectro. El alumno puede encontrar disponible todo este equipamiento y puede hacer un uso libre del mismo, acercando y familiarizando al alumno al equipamiento de laboratorio necesario para el desarrollo hardware que se encontrará en su vida laboral. Es muy importante destacar que el alumno no sólo desarrolla la práctica propuesta en el laboratorio, si no que en su propio desarrollo se instruye en el uso del equipamiento de test y medida existente. Ya que el desarrollo de sistemas hardware no sólo implica los conocimientos teóricos o fundamentales subyacentes, sino que implica un fuerte conocimiento de los instrumentos necesarios para la fabricación y desarrollo, no siendo esta implicación tan fuerte en los desarrollos software. Los alumnos están comprometidos a hacer un uso responsable del laboratorio, siendo advertidos, que un mal uso del equipamiento repercutirá en una calificación, ya que demuestran así su falta de conocimiento en la materia.

3. Material proporcionado para la elaboración de la práctica

En esta sección pasamos a exponer el material que será proporcionado para la elaboración de la práctica.

3.1. La plataforma USB-AER

La tarjeta USB-AER es una tarjeta basada en una FPGA y diseñada para por el grupo de investigación de Robótica y Tecnología de Computadores (RTS) de la Universidad de Sevilla para permitir la comunicación entre diferentes tipos de dispositivos basado en la comunicación AER. Nos hemos decantado por el uso de esta plataforma porque es muy adecuada para el desarrollo de la práctica, y además los profesores

cuentan con una elevada experiencia en su uso y manejo, así como diversos desarrollos software relativos a la comunicación USB. La plataforma USB-AER dispone de 2MB de memoria RAM estática, con un tiempo de acceso de de 12 nseg, y estructurada en palabras de 32 bits. Además posee una interfaz para tarjetas SD/MMC, en las se puede almacenar el firmware que se cargara en la FPGA, y el contenido inicial de la memoria RAM, permitiendo así que el dispositivo pueda configurarse de forma autónoma sin necesidad de estar conectado a un PC. Adicionalmente, por medio de un puerto USB puede reconfigurarse la tarjeta desde un PC, con fines de depuración, e intercambiar datos con el PC. Usando esta funcionalidad puede usarse la tarjeta USB-AER como puente entre cualquier información que fluya por la FPGA y el bus USB.

La versión de la interfaz USB tiene un ancho de banda de 11 Mbps, aunque el valor exacto depende del tipo de transferencia que se use, la sobrecarga del protocolo, etc. Siendo el ancho de banda efecto suficiente para la mayoría de las aplicaciones. Dado que la plataforma USB-AER se basa en una FPGA, el sistema es completamente reconfigurable, cargando diferentes firmwares para la FPGA, los cuales permiten diferentes usos para el mismo hardware.



Figura 1: Fotografía de la plataforma USB-AER.

En la figura 1 mostramos una fotografía de la plataforma USB-AER, en ella aparecen identificados los diferentes componentes que la componen:

1. FPGA Spartan II 200.

2. Memoria SRAM. 512Kx32bits.
3. Microcontrolador Cygnal C8051F320.
4. Conector USB.
5. Conector para tarjeta de memoria MMC/SD.
6. Reloj de cuarzo de 50 MHz
7. Conector de alimentación (6V-12V).
8. Fuente de alimentación (2.5V, 3.3V y 5V).
9. Puerto AER de entrada (No usado en esta práctica).
10. Puerto AER de salida (No usado en esta práctica).
11. Puerto VGA de salida (sin DAC) (No usado en esta práctica).
12. Puerto de programación del Cygnal.
13. Conector JTAG de la FPGA.
14. Diodos LED de FPGA.

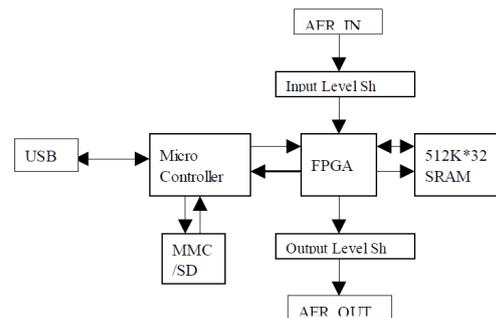


Figura 2: Diagrama de bloques de la plataforma USB-AER.

En la figura 2 mostramos el diagrama de bloques de la plataforma USB-AER. La tarjeta USB-AER se basa fundamentalmente en una FPGA Xilinx Spartan 2, conectada a 4 chips de memoria RAM conformando un total de 512Kx32 bits, con 4 señales de escritura independiente de forma que pueden escribirse palabras de 32 bits, o bytes independiente. Con un reloj de 50Mhz, es posible realizar un acceso cada ciclo a estas memorias (tiempo de acceso de 12 nseg). Por otra parte, un microcontrolador se encarga de cargar el contenido del firmware de la FPGA simulando una memoria EEPROM serie conectada a la FPGA. Aunque como hemos comentado con anterioridad, el firmware podrá cargarse utilizando un fichero almacenado en la tarjeta MMC o bien por el puerto USB. El microcontrolador, Cygnal C8051F320, está

basado en un núcleo 8051 con un interfaz USB 1.1 Full Speed, se encarga del interfaz con la MMC (protocolo SPI) y con un PC a través del puerto USB. Adicionalmente, un bus asíncrono compuesto por 8 líneas de datos, además de algunas señales adicionales de control, interconectan la FPGA con el microcontrolador para el intercambio de datos. Por medio de este bus, el microcontrolador, una vez cargado el firmware en la FPGA, cargará el contenido de la RAM, desde la MMC/SD o el USB. El firmware cargado irá recibiendo los datos desde el microcontrolador, y almacenarlo en la RAM externa de la placa, en la interna de la FPGA o realizar la operación que se necesite. Las señales por las cuales se carga el firmware en la FPGA son distintas a las que se utilizan posteriormente para transferir datos. Un reloj de cuarzo de 50 MHz genera una señal de reloj estable para la FPGA, mientras que el microcontrolador utiliza un oscilador interno de 12 MHz, que mediante un PLL interno se multiplica por 4 para conseguir 48MHz para el USB, y 24 MHz para el procesador 8051. La interconexión entre el microcontrolador y la FPGA se sincroniza por medio de dos señales: strobe y busy. El microcontrolador activa strobe para enviar o recibir datos, mientras que la FPGA puede activar la señal busy para congelar el envío de datos, aunque dado la mayor velocidad de la FPGA sobre el microprocesador esto raramente es necesario.

3.2. Comunicación a través del bus USB

Cuando la tarjeta USB-AER se encuentra conectada a un PC por medio del USB, desde el PC se podrá modificar el firmware contenido en la FPGA, además de enviar o recibir datos desde esta. Puesto que las aplicaciones que pueden cargarse en la FPGA son muy diversas, el protocolo debe ser lo bastante genérico para no tener que ser modificado al desarrollar nuevos diseños. El protocolo está orientado en forma de paquetes o comandos. Antes de cualquier transferencia hay que enviar desde el PC un paquete de control de 64 bytes al microcontrolador (escribir en el dispositivo). El formato de dicho paquete es el siguiente:

Offset	Bytes	Campo	Descripción
00h	3	"ATC"	"Magic Number" para identificar un paquete de control.
03h	1	Command	Comando.
04h	4	Longitud	Numero de bytes que se van a transferir
08h	16	Parámetros	16 bytes de parámetros que se enviarn a la FPGA.

Figura 3: Formato de las tramas USB.

Actualmente hay definidos 3 comandos para el control de la FPGA. Los comandos implementados son los siguientes:

Código	Comando	Descripción
00h	Update Firmware	Se utiliza para subir un nuevo firmware a la FPGA
01h	Write FPGA	Se utiliza para enviar datos a la FPGA.
02h	Read FPGA	Se utiliza para recibir datos de la FPGA.
03h	SetDevName	Se utiliza para darle un nombre al dispositivo.

Figura 4: Comandos implementados para la comunicación USB.

El campo longitud indica el tamaño de los datos que se van a transferir. En el caso del comando '00' indica el tamaño del firmware que se va a transferir. En los comandos '01' y '02' el tamaño de los datos que se van a transferir. A continuación se transfieren 16 bytes que serán parámetros del comando que se le enviarán a la FPGA. Hay una señal de control entre el microcontrolador y la FPGA que se activará para distinguir si los datos que se transfieren son parámetros o datos. El significado es dependiente del diseño cargado en la FPGA. Si no se necesitan parámetros pueden ser ignorados por la FPGA, aunque el microcontrolador siempre los reenviara a la FPGA. Estos parámetros siempre son enviados por el PC a la FPGA, independientemente de si se trata de un comando de lectura o de escritura. Una vez transferido este paquete de control, dependiendo de la dirección del comando especificado, se transferirán desde el PC a la FPGA o viceversa tantos datos como se indicaron en el campo longitud. Estos datos serán troceados y enviados por el USB en fragmentos de 64 bytes.

3.3. El conversor Analógico Digital: ADC08100

Para el desarrollo de esta práctica se necesita una fuente de video compuesto, ya sea una cámara de video CCD o la salida de video un PC. Para digitalizar la señal de video compuesto se los alumnos conectan la salida de video compuesto producido por una cámara al ADC. Una maquina de estados en el interior de la FPGA se encargará de detectar los sincronismos horizontales y

verticales (sincronismo de frame), para ir tomando las muestras en los instantes adecuados.

Describiendo a grandes rasgos las características de una señal de video compuesto, consiste en transmitir una imagen como niveles de tensión. La imagen se escanea por filas, de manera que el nivel de tensión es proporcional al nivel de luminosidad. Una vez terminada una fila, se envía un pulso de sincronismo, indicado por un nivel de tensión negativo, y se procede con la siguiente fila. Al terminar toda la imagen compuesta por 625 líneas de las cuales solo 575 son visibles, se envía un pulso de sincronismo negativo de mayor duración que el que se empleaba al final de cada fila para indicar el final de un frame o imagen.

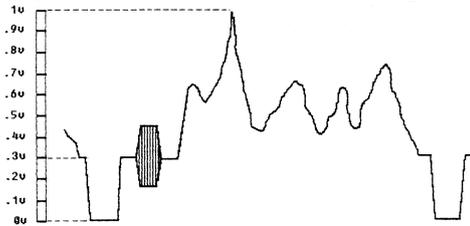


Figura 5: Señal de video compuesto.

El ADC08100 es un conversor analógico/digital que incorpora un circuito de track-and-hold, el cual le permite operara entre 20 y 100Mega muestras por segundo, ofreciendo una excelentes respuestas tanto en AC como en DC, y funcionando con una alimentación de simple rail de 3.3V. Este conversor ADC ha sido diseñado especialmente para aplicaciones de video, imagen y comunicaciones, siendo ideal para el desarrollo de la práctica.

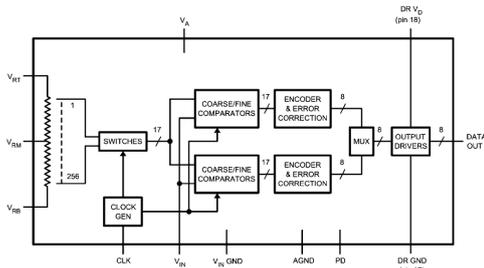


Figura 6: Diagrama de bloques del ADC08100.

El ADC08100 proporciona una salida digital compuesta por 8 bits en paralelo y una señal de reloj, actualizando el valor de cada muestra en el flanco de bajada de una señal de reloj.

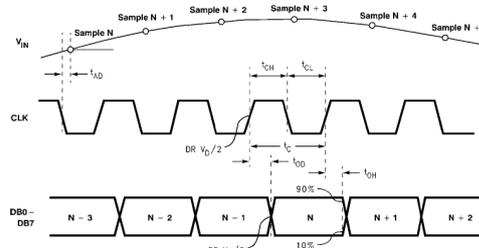


Figura 6: Señal de video compuesto.

En la figura 7 mostramos una fotografía del montaje final de la práctica. En ella puede observarse como se ha conectado la plataforma USB-AER al ADC08100, esta conexión se realiza mediante un bus paralelo de 40 bits, aunque sólo 9 son utilizados. Finalmente como fuente de video los alumnos disponen una pequeña cámara CCD, mediante la cual podrán experimentar y desarrollar satisfactoriamente la práctica.

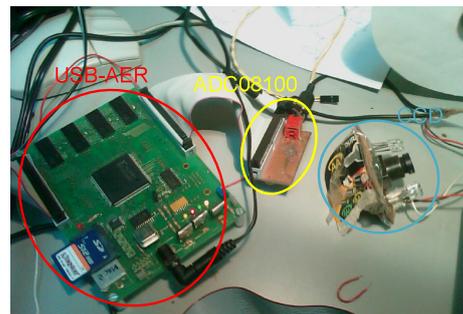


Figura 7: Fotografía del montaje final.

4. Metodología de elaboración y seguimiento

La asignatura se divide en sesiones teóricas y sesiones prácticas, primando fundamentalmente estas últimas. Durante las primeras sesiones (teóricas), se estudia los fundamentos del codiseño hardware-software, donde se separan las funcionalidades del problema que se quiere resolver, asignando responsabilidades a ambas partes. Se hace hincapié en las diferencias entre los lenguajes de programación, como una secuencia de instrucciones que son ejecutadas por un procesador, y los lenguajes de descripción de hardware (HDL), donde lo que se definen son

circuitos digitales, y que por lo tanto no describen un algoritmo, sino un circuito lógico.

En estas sesiones se explica las nociones del lenguaje VHDL, la diferencia entre qué partes del VHDL son sintetizables, y por tanto trasladables a un circuito, y cuáles no, sino que sirven únicamente a la hora de realizar testbench (que son las que se parecen más a un lenguaje de programación, precisamente).

Una vez se les ha enseñado la sintaxis y demás aspectos del VHDL, se pondrán a prueba dichos conocimientos en el laboratorio. Las primeras prácticas se centrarán en la realización de testbench (entidades de prueba utilizadas en la simulación), y al manejo del simulador de vhdl.

El simulador permite realizar varios tipos de simulaciones: behavioural (de comportamiento, sin tener en cuenta los retrasos de la circuitería), post-síntesis y post-placement, donde se tienen en cuenta los retrasos de la circuitería (en el primero), teniendo en cuenta el modelo concreto de FPGA que se va a utilizar, y el más completo de todos, el post-placement, donde se tendrá en cuenta también los retrasos introducidos por las conexiones dentro de la FPGA, y que por tanto dependerá de la colocación física dentro de la FPGA, y la separación entre los distintos elementos.

Una vez se han familiarizado con el simulador, y aprendido a identificar y corregir fallos, basándose en los cronogramas obtenidos con ayuda del simulador, se pasará a realizar las primeras síntesis y a probar los diseños sobre una FPGA real. Una vez se han completado satisfactoriamente estas primeras sesiones prácticas, se pasará a realizar un diseño completo, que es el que se describe en este artículo.

El objetivo de la práctica es desarrollar con ayuda de una FPGA y un convertidor analógico digital externo (capaz de muestrear a una velocidad de hasta 100Mhz), un diseño capaz de capturar la señal proveniente de una cámara de video y almacenarla dentro de la memoria existente en la FPGA. Para ello se les describe las características de la señal de video compuesto, y la funcionalidad de las señales de sincronismo, horizontal y vertical.

Mediante el diseño de una máquina de estados, deberán detectar dichos sincronismos, para realizar las capturas en unos instantes de tiempo determinados. Para simplificar el

contenido de la práctica, se supondrá que la señal de video es monocromática, obviándose la parte de la demodulación de la sub-portadora de color.

Para ayudarles en el desarrollo de esta parte, se les facilitará junto con la documentación de la práctica, un diagrama con el funcionamiento de la máquina de estados que deben implementar, así como unos esqueletos de algunas partes del diseño ya planteados, que ellos deberán completar. Para esta parte del ejercicio se cuenta con dos sesiones de prácticas, durante las cuales podrán consultar al profesor las dudas que les vayan surgiendo.

Durante la semana que transcurre entre las dos sesiones de prácticas, los alumnos podrán acceder a un laboratorio de prácticas habilitado para tal fin, donde podrán continuar el trabajo con el mismo material que se ha utilizado en las prácticas. A diferencia de otras asignaturas, donde al alumno se le ofrece a modo de préstamo el material necesario para que puedan trabajar en casa, debido al número de elementos necesarios para un puesto de trabajo (y por tanto, al elevado coste que representa), se ha estimado más conveniente en que los alumnos pudieran acceder a él en el laboratorio. Entre el material que necesitan podemos señalar la placa con la FPGA, la placa con el ADC externo, programador USB de Xilinx, fuente de alimentación, osciloscopio, etc.

En la siguiente sesión de prácticas, deberán interconectar la FPGA con el microcontrolador, utilizando un protocolo prefijado, que permitirá una vez terminada la captura, el ir leyendo secuencialmente el contenido capturado en la memoria e ir transmitiéndolo al ordenador utilizando el bus USB. El microcontrolador empleado, un Silabs C8051F320 (basado en el núcleo de Intel del 8051), tiene integrado un controlador USB. La parte del código del microcontrolador empleado que se encarga de la gestión de los mensajes del bus USB se le proporciona al alumno, como parte del material con el que contarán para realizar la práctica, permitiéndoles centrarse en el aspecto de interconexión entre el microcontrolador y la FPGA.

Para poder comprobar que el diseño que están haciendo se comporta satisfactoriamente, se les proporcionará un programa de ejemplo con el que podrán mostrar en la pantalla del ordenador el contenido descargado por el puerto USB. Aunque

en una práctica posterior deberán ellos mismos realizar un pequeño programa de ejemplo que realice estas mismas funciones, utilizando unas librerías que se les suministra. Se ha creído conveniente que contarán en un primer momento con un programa ya funcional para facilitar la detección de errores en la parte que están desarrollando, evitando la frustración de no conseguir los resultados esperados, sin saber en cuál de las componentes se encuentra localizado el fallo.

De esta forma, al ir desarrollando la práctica de manera progresiva, contando en cada paso con las herramientas necesarias para probar el diseño que están realizando de forma aislada, se facilita enormemente la localización de fallos, evitando que el alumno se sienta frustrado y pierda la motivación al emplear mucho tiempo buscando errores sin saber en qué parte de la cadena se encuentra. Esto permite que se pueda plantear este tipo de prácticas, con un nivel de dificultad que de otra forma, si tuvieran que empezar todo el diseño desde cero, no podrían plantearse, o que en caso de plantearse, requerirían un número de sesiones prácticas y un trabajo añadido del alumno fuera de las mismas en horas de laboratorio, demasiado alto. Además, el número de alumnos que perderían la motivación al no ir obteniendo resultados visibles de forma progresiva, sería también muy elevado, repercutiendo negativamente en su proceso de aprendizaje.

Esta práctica ha sido positivamente acogida por parte de los alumnos, en el momento que se encuentra estructurada como distintas prácticas con hitos intermedios. Así mismo, fomenta el intercambio de ideas, a la hora de encontrar soluciones a los problemas que se van encontrando.

Esta metodología de aprendizaje basado en problemas asegura además un mejor aprendizaje por parte del alumno, respecto al modelo tradicional de enseñanza donde las sesiones prácticas tenían una importancia superior a las sesiones prácticas.

5. Conclusiones

En este trabajo se ha presentado el enfoque adoptado para la impartición de la asignatura de co-diseño en VHDL, intentando realizar una aproximación progresiva y amena al contenido de

la práctica, para intentar despertar la curiosidad del alumno, a la vez que se profundiza en el estudio del co-diseño software-hardware. Parte de la práctica se desarrolla realizando el desarrollo de un circuito descrito en VHDL, sobre una FPGA, que se encargará de la captura de imágenes proveniente de una cámara de video. Una vez terminada la parte hardware, deberán interconectarlo con una aplicación en MATLAB, o cualquier lenguaje de alto nivel, utilizando unas librerías que manejan el bus USB.

Se pretende así que el alumno tenga una visión completa de lo que sería una cadena completa entre un circuito electrónico (FPGA), su interfaz con un microcontrolador, que hace de interfaz con el ordenador por medio del USB, y por último, del manejo de dicha información desde un lenguaje de alto nivel. Algunas capas, como el driver del dispositivo USB, no se han tratado durante esta práctica, debido a que se escapaba de los objetivos del curso. Sin embargo, no siempre es necesario el desarrollo de un nuevo driver. En este caso, se ha utilizado un driver proporcionado por el fabricante del microcontrolador.

También se ha conseguido que el alumno entienda y se familiarice con la señal de video compuesto y sus características, ayudándoles a entender cómo funcionan las señales de video analógicas, que describen una secuencia de imágenes mediante su recorrido por filas.

A pesar de la aparente complejidad de dicha práctica, se les ha dado en forma de una lista o guía, con diversos puntos que debían ir resolviendo paso a paso, hasta su completa resolución. Se ha hecho hincapié en los aspectos que podían tener mayor dificultad, aportándole pistas o información adicional para superar satisfactoriamente el ejercicio. Ha sido bien acogida por los alumnos y la mayoría ha resuelto sin demasiada dificultad la totalidad de la práctica.

Agradecimientos

Este proyecto ha sido posible en parte gracias al proyecto VULCANO (TEC2009-10639-C04-02).

Referencias

- [1] M. Valero-García, J. Navarro, FAQ sobre la docencia centrada en el aprendizaje.

- <http://epsc.upc.edu/~miguel%20valero/contingut3.html>.
- [2] Universidad de Sevilla. Plan de Renovación de Metodologías Docentes 2007. http://www.institucional.us.es/plan_renovación.
- [3] T. Markham Project Based Learning, a guide to Standard-focused Project based learning for middle and high school teachers. *Buck Institute for Education, (2003)*.
- [4] G. Solomon. Project Based Learning: a Primer, *Technology Learning*, vol. 23, num. 6 (Jan 2003).
- [5] H. Dolmans, et al. Problem-based learning: future challenges for educational practice and research. Blackwell Publishing Ltd 2005. *Medical Education* (2005); 39.
- [6] M. Hedley, S. Barrie. An undergraduate microcontroller systems laboratory. *IEEE Transactions on Education*, Volume: 41, Issue: 4, (Nov.1998).
- [7] M. Valero-García. Las dificultades que tienes cuando haces PBL. Cap. 8 La Educación Superior hacia la Convergencia Europea.