

DOCENCIA CON SOPC PARA EL DESARROLLO DE PROYECTOS DE TITULACIÓN

Julio Cesar Sosa Savedra¹, Rubén Hernández Tovar², Víctor Hugo García Ortega³

1,2,3 ESCOM, Instituto Politécnico Nacional. Departamento de Posgrado. Av. Juan de Dios Batiz s/n, Col. Lindavista, Del. Gustavo A. Madero, c.p. 07738. México, D.F. Tel +52 55 57296000, Ext. 56867 2UPIITA, Instituto Politécnico Nacional. Av. Instituto Politécnico No. 2580. Col. Ticomán, Del. Gustavo A. Madero, c.p. 07340. México, D.F. Tel +52 55 57296000, Ext. 52064

1 jcsosa@ipn.mx , 2 rhtovar@ipn.mx , 3 vgarciao@ipn.mx

RESUMEN

En este trabajo es presentada la experiencia docente en el diseño de sistemas en un chip programable (SoPC, de *System-on-a-Programmable-Chip*). El diseño con SoPC es empleado para los proyectos de fin de carrera, también llamados trabajo terminal, por los estudiantes de la Escuela Superior de Cómputo (ESCOM), del Instituto Politécnico Nacional. En la ESCOM es posible titularse mediante la opción curricular, que consiste en que el estudiante deberá acreditar dos unidades de aprendizaje y el desarrollo de un proyecto que utilice los conocimientos adquiridos durante su carrera. Los estudiantes reciben una formación en diseño de sistemas digitales en donde adquieren las habilidades necesarias para resolver problemas específicos. Dentro de esta formación están: conceptos de diseño basados en microprocesadores, microcontroladores, procesador digital de señales (DSPs) y diseño digital con dispositivos lógicos programables (PLDs), entre los que destacan los "*fiel-programmable-gate-array* (FPGAs). Todos estos dispositivos pueden ser programados mediante: un lenguaje de descripción hardware (HDLs) o programación en ensamblador y/o C/C++. Aquí se presenta un trabajo de titulación donde el estudiante conjunta todo lo aprendido e implementa el protocolo H.225 en un FPGA.

Las ITU presenta la recomendación H.225.0 como un conjunto de protocolos de señalización de llamada y empaquetado de flujos de medios por los sistemas de comunicación multimedia basados en paquetes. La implantación en SoPC de este esquema de control de llamada es de suma importancia, si se quiere realizar un enlace de VoIP, ya sea fijo o móvil, que sea independiente de una computadora personal.

PALABRAS CLAVES: SoPC, PLD, FPGA, H.225, VoIP.

TEACHING WITH SOPC FOR DEVELOPMENT OF DEGREE PROJECT

ABSTRACT

In this paper is presented the teaching experience in the design of a system on a programmable chip (SOPC). The SoPC design is used for the final degree project, also called terminal work, by students of the Escuela Superior de Computo, of Instituto Politecnico Nacional. It is possible to obtain the engineering degree by curricular option. The curricular option consist on the student must approve two courses with developing a project using the knowledge acquired during their stay, in the university. The students have a training in digital systems design where acquired the skills to solve specific problems. Within this training are: design concepts based on microprocessors, microcontrollers, digital signal processors (DSPs)

and programmable logic devices (PLDs). Within the PLD stand the field-programmable-gate-array (FPGA). All these devices can be programmed by: a hardware description language (HDL) or assembler and/or C / C + + programming. Here it is presented a final degree project where the student used all the knowledge acquired and implement the H.225 protocol on a FPGA.

The ITU H.225.0 recommendation is a set of call signaling protocols and package for data packet multimedia communication systems. The SoPC implementation of this control mechanism is essential when a VoIP link is required without the using of a personal computer.

KEY WORDS: SoPC, PLD, FPGA, H.225, VoIP.

INTRODUCCIÓN

El avance tecnológico ha hecho que los fabricantes de semiconductores, como los dispositivos lógicos programables (PLD's) y en particular los FPGA's (Field Programmable Gate Arrays), deben estar en constante evolución y hacia la creación de nuevas familias de dispositivos [1]. Por otro lado, el desarrollo tecnológico en ambos sentidos; tanto en la complejidad de los sistemas requeridos, que cada día son más ambiciosos, como la posibilidad de integrar un sistema en un mismo circuito integrado, orillan a buscar nuevas estrategias de diseño [2]. De esta manera surgen nuevos elementos que facilitan el diseño de sistemas digitales. Estos elementos son conocidos como herramientas de diseño asistido por computadora y automatización de diseño electrónico, también conocidas como herramientas CAD- EDA [3].

El surgir estas nuevas herramientas tecnológicas, como es el caso de los FPGA's que contienen una gran cantidad de memoria, elementos lógicos, núcleos de procesadores, multiplicadores y otros elementos capaces de desarrollar funciones hasta cierto punto complejas hace posible la implementación de todo un sistema en un mismo chip programable (SoPC) [4].

En la Escuela Superior de Cómputo, del Instituto Politécnico Nacional de México, los estudiantes pueden titularse mediante la opción curricular. Dicha opción de titulación, consiste en que el estudiante deberá acreditar dos unidades de aprendizaje desarrollando un proyecto/sistema en el cual emplee los conocimientos adquiridos durante su carrera. Los estudiantes reciben, durante su carrera, una formación en sistemas digitales en la que desarrollan conceptos de diseño basados en microprocesadores, microcontroladores, procesador digital de señales (DSPs) y diseño digital con dispositivos lógicos programables (PLDs). Entre los PLD's empleados están los SPLD o PLD Simples, y los FPGA o Arreglos de Compuertas programables en campo. Todos los dispositivos son programados ya sea con un lenguaje de descripción hardware (HDLs) o programación en ensamblador y/o C/C++, según sea el tipo de dispositivo.

En el proyecto de titulación, llamado trabajo terminal (TT), además de que el estudiante debe conjuntar todo lo aprendido, también debe enfrentarse a un nuevo reto en el que se tiene un flujo de diseño nuevo y una herramienta de desarrollo más completa. Como una experiencia de éxito se presenta un trabajo que expone la implementación del protocolo H.225 en un dispositivo lógico programable, del tipo FPGA. El protocolo es uno del conjunto H.323 que es comúnmente utilizado para la señalización de llamada capaz de integrar redes de datos, audio y video.

DOCENCIA DE SISTEMAS DIGITALES

Desde el surgimiento de nuevas herramientas para el diseño de sistemas digitales se inició un proceso de actualización en las bibliografías del área digital [5] y [6], entre otros. Inicialmente se consideraban esos temas, PLD's y herramientas CAD, como un capítulo totalmente independiente. Recientemente varios autores han realizado modificaciones más profundas.

Existen dos tendencias bien definidas y marcadas en las nuevas bibliografías: las que exponen al final de cada unidad temática el diseño empleando un lenguaje de descripción de hardware [7] y [8], a la que se nombra enseñanza modular [2]. La otra tendencia es en donde se muestra el diseño de circuitos digitales empleando un lenguaje de descripción de hardware [9], llamada enseñanza con un HDL. Sin

embargo, es importante resaltar que en la práctica docente realmente existen tres enfoques: el método de enseñanza tradicional o clásica, la enseñanza modular y la enseñanza con un HDL.

Enseñanza tradicional

Consiste en mostrar al estudiante las estrategias de diseño digital usando tecnología estándar TTL (serie 74XX) o tecnología CMOS (serie 4000). Las estrategias de diseño utilizan circuitos integrados de mediana o alta escala de integración que presentan varias funciones específicas.

Este tipo de enseñanza tiene una tendencia a fusionarse pues ya no se emplean circuitos integrados TTL como base fundamental en la implantación de sistemas digitales. La razón por la que puede permanecer es debido a que no se ha actualizado el plan de estudios de la carrera, algo poco común, pero que puede pasar.

Enseñanza modular

Consiste en instruir al estudiante con la metodología de diseño tradicional (enseñanza tradicional) y adicionalmente, al finalizar cada tema o unidad de aprendizaje, se le enseña el diseño digital utilizando un lenguaje de descripción de hardware. Morris Mano [7] especifica que la principal modificación, de la 3ª edición, es la inclusión de secciones sobre HDL, en particular Verilog HDL, e indica que introdujo el material insertando secciones aparte, de modo que el profesor pueda decidir cómo incorporarlo a su curso. De aquí podemos desprender que deja abierto, por cuestiones didácticas y de enfoques, el cómo introducir las herramientas de desarrollo para la enseñanza de un curso que requiere una herramienta.

Otra obra [8], muestra los conceptos fundamentales del diseño digital manual clásico e ilustra cómo se diseña hoy en día los circuitos digitales mediante herramientas CAD, en particular utilizando el lenguaje de programación VHDL (por sus siglas en inglés Very High Speed Integrated Circuits Hardware Description Language). En la obra se hace hincapié en que los diseñadores modernos ya no siguen las técnicas manuales, sin embargo deja muy claro de que el diseñador no comprenderá dichas herramientas si no tiene la teoría implícita en ello.

Enseñanza con un HDL

Esta estrategia de enseñanza parte como base fundamental de un lenguaje de descripción de hardware enfocado en el diseño de circuitos digitales combinacionales, secuenciales o bloques de la arquitectura de un computador según sea la asignatura, ejemplos de esta estrategia tenemos [9] y [10].

Actualmente cualquier proceso de ingeniería dispone de un soporte software, para el desarrollo de sistemas complejos, que asiste al ingeniero de aplicaciones o sistemas. Por ello, es fácil pensar que con el solo hecho de poseer conocimientos básicos de programación es más que suficiente para dominar el diseño digital con lógica programable. Si bien las herramientas CAD-EDA presentan muchas bondades, también es cierto que para aprovecharlas al máximo y sobre todo para realizar diseños que puedan ser sintetizables y programables es necesario dominar dicha herramienta y poseer los conceptos teóricos del diseño lógico digital [3].

El principal inconveniente en aprender directamente un Lenguaje de Descripción de Hardware (HDL) es: que comúnmente el alumno sabe un lenguaje de programación de alto nivel, y al realizar el programa con un HDL, suele cometer el error de comenzar a describir el circuito como si fuera un programa para

computadora, en vez de describir un circuito para un dispositivo lógico programable. De esta manera, la principal restricción para describir y poder sintetizar fácilmente un circuito hardware digital es pensar en Términos de compuertas y registros y no en función de variables y subrutinas [3].

HERRAMIENTAS CAD-EDA

Las necesidades actuales y el continuo avance tecnológico han permitido fabricar y diseñar sistemas electrónicos cada vez más complejos. Por tal razón es necesario emplear Herramientas de Diseño Asistido por Computadora y Automatización de Diseño Electrónico, conocidas como herramientas CAD-EDA.

Las herramientas EDA son todo el software y el hardware que sirven para el diseño de sistemas electrónicos. Dentro de las herramientas EDA, están las herramientas CAD, que desempeñan un papel muy importante en el diseño de sistemas. El concepto de CAD proviene del proceso de diseño que emplea sofisticadas técnicas gráficas por computador, apoyadas en paquetes de software para la resolución de problemas de cálculo, de desarrollo, de costos, etc. asociados con el trabajo de diseño [3].

De esta forma queda de manifiesto que para realizar un desarrollo tecnológico es ineludible el uso de las herramientas CAD y está claramente establecido que su enseñanza es necesaria, como se ha expuesto en [1], [11] y [12]. En los trabajos se deja de manifiesto el empleo de herramientas CAD durante la enseñanza y aprendizaje en las carreras de tecnologías de la información. Asimismo, se habla de la actualización docente y de metodologías activas para la enseñanza de los sistemas digitales, empleando herramientas CAD-EDA.

La Escuela Superior de Cómputo (ESCOM), del Instituto Politécnico Nacional de México, posee laboratorios que cuentan con las herramientas CAD-EDA necesarias, para el desarrollo de sistemas digitales en un chip programable (SoPC). Dentro de dichas herramientas, destacan 30 tarjetas de desarrollo Spartan 3, de Xilinx y 30 tarjetas de desarrollo DE2-115, de Altera. Todas las tarjetas poseen un dispositivo programable del tipo FPGA, además de que cada tarjeta se programa con una PC que posee un software de desarrollo y simulación. El software cuenta con una licencia gratuita para emplear un softcore de cada uno de los fabricantes, por ejemplo: el microprocesador MicroBlaze es posible implantarlo en la tarjeta Spartan 3 y el Nios II, en su versión básica, es posible implantarlo en la tarjeta DE2-115. Adicionalmente la ESCOM pertenece a los programas universitarios de Xilinx como de Altera.

Antes de la llegada de los dispositivos del tipo FPGA, los equipos de diseño que requerían crear un sistema en un circuito integrado (SoC), tenían dos alternativas: el diseño de un circuito full-custom VLSI (Very Large Scale Integration) o mediante los Circuitos Integrados de Aplicación Específica (ASIC). Ambas tecnologías son costosas y presenta tiempos relativamente largos de fabricación, alargando los procesos de diseño, sin contar con la cantidad de grandes equipos de ingenieros y miles de dólares en herramientas de diseños y equipamiento. Una comparación entre las ventajas y desventajas en el diseño con un SoPC & ASIC (SoC) & VLSI (full-custom) se muestra en la Tabla 1. Además, un error en un diseño con decenas de millones de transistores (VLSI o ASIC) puede llevar a cambios que obligan a un rediseño, lo que repercutiría en meses de retraso del producto y otro cargo millonario de ingeniería no recurrente. Por esta razón el diseño de sistemas sobre dispositivos programables o SoPC abre las puertas a la innovación, un bajo costo de diseño y en un rápido prototipado.

Tabla 1: Comparación en la modalidad de diseño entre SoPC & ASIC & VLSI.

Características	SoPC	ASIC	VLSI
S/W flexibilidad	Buena	Buena	Nula
H/W flexibilidad	Buena	Buena	Buena
Reconfigurabilidad	Buena	Nula	Nula
Costo/tiempo desarrollo	Buena	Malo	Malo
Rendimiento	Medio	Bueno	Bueno
Costo producción*	Medio	Bueno	Bueno
Consumo de energía	Malo	Bueno	Bueno

* Para producción en serie.

Flujo de Diseño

En un flujo de diseño típico se parte de la idea del sistema, posteriormente se introduce la descripción del circuito para realizar una primera simulación, llamada simulación funcional. Esta simulación es para saber si el circuito diseñado realmente hace la función para la cual fue concebido, sin importar los retardos de propagación. Una vez pasada esta etapa del diseño se procede a sintetizar el circuito y se hace una segunda simulación, llamada simulación digital, también conocida como timing simulation. En esta segunda simulación se consideran los retardos de propagación y se fija una estrategia para mejorar dichos tiempos, así mismo se hacen ajustes más finos en el sistema.

En ciertos diseños, sobre todo cuando es implantado un microprocesador embebido en el mismo chip (soft-core), es necesario un ajuste en la estrategia de diseño. Este ajuste es debido a que ahora existen en el mismo chip, un dispositivo que se debe programar con un conjunto de instrucciones. La programación se hace mediante ensamblador o con un programa de alto nivel como C/C++. En la figura 1 se muestra la metodología de diseño para implantar un sistema con un soft-core, en donde se hace la modificación y se diferencia entre el diseño software y hardware.

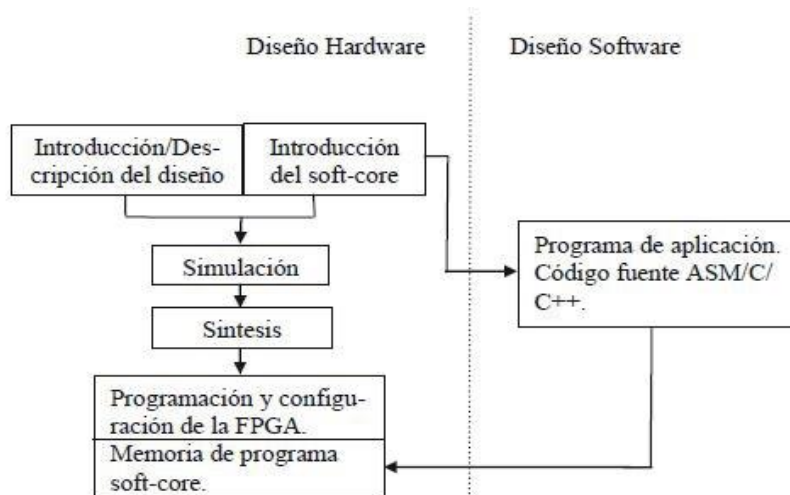


Figura 1: Diagrama a flujo de diseño para un SoPC que posea un microprocesador embebido.

DESARROLLO DEL TRABAJO TERMINAL

La Unión Internacional de Telecomunicaciones (Internacional Telecommunication Union - ITU), bajo la recomendación H.225, ha planteado un esquema para la señalización de la llamada de un enlace de VoIP en una conexión de 2 puntos finales. En dicha recomendación establece el proceso de control de llamada que puede agruparse en 4 fases: conexión, suspensión, reanudación y desconexión. Las fases permiten ubicar el estado del proceso de control de llamada y consisten en un intercambio de mensajes con los cuales se realizan peticiones y se dan respuesta a estas peticiones [13].

En la fase de conexión se realiza o recibe una petición de llamada, la cual al ser aceptada, da comienzo al intercambio de mensajes para establecer la llamada. Si la petición es rechazada se pasa a la fase de desconexión. Una vez alcanzada la conexión de la llamada se llega a un estado de hablando (Talking) en el cual se puede realizar o recibir una petición de suspensión de llamada, que al ser rechazada nos traslada de nuevo al estado de “hablando” para continuar llamando. Si la petición es aceptada se alcanza un estado de suspendido (Suspend) en el cual la llamada se coloca en espera. Cuando nos encontramos en un estado de suspensión, es posible recibir o realizar una petición de reanudación, que al ser rechazada nos deja de nuevo en el estado de suspensión, para una reanudación de llamada futura; pero si la petición es aceptada, nos movemos al estado de hablando en el cual podemos continuar con nuestra llamada de forma normal. Finalmente, durante cualquier etapa del control de llamada, se puede realizar o recibir una petición de desconexión, esta petición no puede ser rechazada, por lo que simplemente al enviar o recibir una petición de este tipo se liberan los canales establecidos y se da por terminada la llamada, con previo acuse de recibo de desconexión.

Este esquema de control de llamada permite realizar conexiones, intercambio de mensajes de servicios suplementarios y cancelaciones de llamada. El listado y sintaxis del mensaje de control de llamadas H.225 se toma directamente de las recomendaciones Q.931 y Q.932 de la ITU.

Para la implementación, del control de llamadas H.225, se realizó mediante una máquina de estados que consiste en cada uno de los estados por lo que pasa el control de la llamada en una conexión de VoIP a través de H.323. Cada uno de estos estados corresponde a una máquina de estado mayor que realiza el intercambio de mensajes de cada fase. Para un mayor detalle de la implementación del protocolo H.225 ver la referencia [13].

La implementación final, del esquema de control de llamada descrito por la recomendación H.225, puede concebirse como un dispositivo con las señales de entrada y salida mostradas en la figura 2.

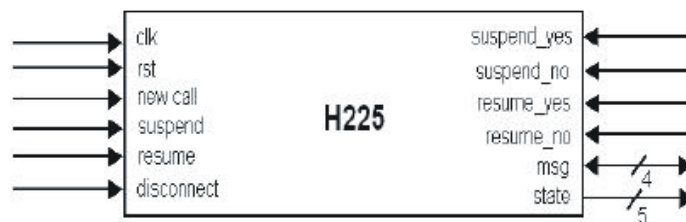


Figura 2: Entidad del componente de la recomendación H.225 desarrollado.

Donde:

clk: señal de reloj para la máquina de estado

rst: señal de reset

new call: señal de petición de nueva llamada

suspend: señal de petición de suspensión de llamada

resume: señal de petición de reanudación de llamada

disconnect: señal de desconexión de llamada

suspend_yes: señal para la aceptación de la suspensión de llamada

suspend_no: señal para el rechazo de la suspensión de llamada

resume_yes: señal para la aceptación de la reanudación de llamada

resume_no: señal para el rechazo de la reanudación de llamada

msg: señal con el mensaje recibido o mensaje a enviar

state: señal que nos indica en que estado nos encontramos

El diseño de esta arquitectura fue realizado con la herramienta StateCAD 9.2i para el modelado de la máquina de estados, llevando acabo los procesos de optimización con: Register outputs, Guarantee coverage, Enable buffering, Optimize port I/O y Retain unassigned outputs between clocks. La implementación y puesta en marcha del sistema fue hecha con la herramienta de desarrollo ISE 9.2.03i llevando acabo los procesos de síntesis y programando en la FPGA Spartan 3E, en particular la XC3S500E-5FG320, de Xilinx. Así mismo se empleo la herramienta de simulación con ModelSim XE III 6.2c. La cantidad de recursos empleados de la FPGA son los mostrados en la Tabla 2.

Tabla 2: Recursos utilizados de la FPGA XC3S500E-5FG320.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	227	9,312	2%
Number of 4 input LUTs	2,835	9,312	30%
Logic Distribution			
Number of occupied Slices	1,610	4,656	34%
Number of Slices containing only related logic	1,610	1,610	100%
Number of Slices containing unrelated logic	0	1,610	0%

Total Number of 4 input LUTs	2,841	9,312	30%
Number used as logic	2,835		
Number used as a route-thru	6		
Number of bonded IOBs	32	232	13%
IOB Flip Flops	11		
Number of GCLKs	3	24	12%
Total equivalent gate count for design	20,990		
Additional JTAG gate count for IOBs	1,536		

La frecuencia máxima de operación fue de 50MHz.

CONCLUSIONES

En este trabajo se ponen de manifiesto las grandes ventajas que poseen los FPGA en la tecnología para el desarrollo tecnológico pero sobre todo para la innovación. En este caso, se ha podido realizar una aplicación de alto nivel, como es el protocolo H.225 gracias a la flexibilidad del dispositivo y a las herramientas CAD-EDA. Además se pone de manifiesto la necesidad de incrementar los conocimientos de los ingenieros de desarrollo en sistemas reconfigurables con la finalidad de desarrollar aplicaciones y productos comerciales basados en esta tecnología.

Sin embargo, tomar ventaja de las inmensas capacidades de los FPGA no es fácil pues requiere una manera especial de pensar, y un ojo vigilante para estar al día con las nuevas herramientas, técnicas, tecnologías y tendencias en la siempre cambiante área de los FPGA. Convertirse en un experto diseñador en FPGA requiere dominar lenguajes descriptivos de hardware, herramientas de simulación y síntesis, software de colocación y ruteo, integración de sistemas, optimización de potencia e integridad de la señal, programación en C/C++ o ensamblador y un número considerable de otras habilidades técnicas.

Finalmente, el uso de las herramientas CAD-EDA, basadas en SoPC, permiten unificar el desarrollo de los conocimientos y minimizar los costes de laboratorios. Esto debido a que los mismos recursos pueden ser compartidos por diversas asignaturas, como son: Arquitectura de Computadoras, Fundamentos de Diseño Digital, Diseño de Sistemas Digitales y Sistemas Embebidos.

Referencias

1. COLOM PALERO, Ricardo José et al. "Docencia multidisciplinar con sistemas de desarrollo para SoPC" En resúmenes de los trabajos del VIII Congreso de tecnología aplicada a la enseñanza de la electrónica TAE08. Julio 2008, pp. 42.
2. SOSA SAVEDRA, Julio Cesar. "Propuesta para la enseñanza de sistemas digitales empleando herramientas CAD-EDA" En actas del Congreso Internacional en Informática y Computación (ANIEI 2009). Ensenada, Baja California, México. Del 21 al 23 de Octubre de 2009.
3. PARDO, Fernando y BOLUDA José A. VHDL: Lenguaje para síntesis y modelado de circuitos. España: Editorial RA-MA. 2011. 308 pp. ISBN 978-84-9964-040-2.
4. HALL, Tyson y HAMBLEN, James. "System-on-a-Programmable-Chip Development Platforms in the Classroom" IEEE Tansaction on Education, 2004. Vol. 47, No. 4, pp. 502-507.
5. TOCCI, Ronald. Sistemas Digitales: principios y aplicaciones. México: Editorial Prentice Hall, 1996. 833 pp. ISBN: 968-880-737-0.
6. NELSON, Victor P. et al. Análisis y Diseño de Circuitos Lógicos Digitales. México: Editorial Prentice Hall, 1996. 842 pp. ISBN: 968-880-706-0.
7. MANO, Morris. Diseño digital. México: Editorial Prentice Hall/Pearson, 2003. 521 pp. ISBN: 970-26-0438-9.
8. BROWN, Stephen y ZVONKO, Vranesic. Fundamentos de Lógica Digital con Diseño VHDL. México: Editorial Prentice Hall, 2006. 939 pp. ISBN:-13: 978-970-10-5609-7.
9. PEDRONI, Volnei A. Circuit Design and Simulation with VHDL. USA: Editorial The MIT Press. 2010. 610 pp. ISBN-13: 978-0262014335.
10. ROTH, Charles H. Digital Systems Design Using VHDL. México: International THOMSON EDITORES. SA CV. 2008. 591 pp. ISBN: 0-495-24470-8 ISBN-13: 978-0262014335.
11. SOSA, Julio Cesar et al. "El posgrado como cimiento del desarrollo tecnológico: los FPGAs como motores de innovación". En actas del 8º Congreso Internacional de Educación Superior. La Habana, Cuba. Del 13 al 17 de febrero de 2012.
12. OLIVERAN, Juan P. Y FIORELLA, Haim. "Lab at Home: Hardware Kits for a Digital Design Lab" IEEE Transaction on Education, 2009, Vol.52, No.1, pp. 46-51.
13. SOSA, Julio C. et al. "Señalización de llamada H.225 implantada sobre un FPGA". Congreso Internacional de Telemática y Telecomunicaciones, CITTEL'08. La Habana, Cuba. Del 1 al 5 diciembre de 2008.