



Ingeniería y Competitividad

ISSN: 0123-3033

inycompe@gmail.com

Universidad del Valle

Colombia

Vera, Alexander; Zúñiga, Nathan; Bernal, Álvaro
Herramienta en línea para la programación y depuración remota de funciones lógicas digitales
Ingeniería y Competitividad, vol. 15, núm. 1, 2013, pp. 79-91
Universidad del Valle
Cali, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=291329165007>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Herramienta en línea para la programación y depuración remota de funciones lógicas digitales

Online tool software to remote programming and debugging of digital logic functions

Alexander Vera*, Nathan Zúñiga, Álvaro Bernal*§**

**Grupo GADyM, Escuela de Ingeniería Eléctrica y Electrónica, Universidad del Valle, Cali, Colombia.*

***Grupo GDSPROC, Ingeniería Electrónica, Universidad del Quindío, Armenia, Colombia*

alexander.vera@correounivalle.edu.co, nathan@uniquindio.edu.co,

§ alvaro.bernal@correounivalle.edu.co

(Recibido: Enero 06 de 2012 - Aceptado: Mayo 16 de 2013)

Resumen

En el diseño de circuitos electrónicos digitales, los ejercicios y prácticas deben integrarse a los procesos de enseñanza. La mayoría de prácticas se realizan en espacios locales, como aulas de laboratorio especializadas; sin embargo, la educación en línea se ha fortalecido y actualmente es posible experimentar desde ubicaciones remotas. En este trabajo se describen los principales aportes realizados por grupos de investigación en el campo de los laboratorios en línea, enfocándose en los sistemas electrónicos digitales. Se describe también una herramienta software desarrollada para la programación remota de funciones lógicas digitales, que apoya la enseñanza de los Sistemas Digitales. Esta herramienta fue diseñada para trabajo local o en red y permite a los usuarios interactuar con un conjunto de herramientas para la configuración de un FPGA Cyclone II sobre una tarjeta de desarrollo DE2-70.

Palabras Clave: *Entorno Integrado de Desarrollo, laboratorio en línea, FPGA, Cliente/Servidor*

Abstract

Digital electronic circuits design requires that both exercises and practices be integrated to teaching processes. Most practices are executed over local areas, as specialized laboratory classrooms; however, online education has been strengthened and currently is possible to experiment from remote locations. Some main contributions about online laboratories made by research groups were described in this work; these were focused on digital electronic systems. A developed software tool to remote programming of digital logic functions was described too, which supports the Digital Systems teaching. This tool was designed for local or net-working, and allows users to interact with a tool set for configuring a Cyclone II Field-Programmable-Gate-Array on a DE2-70 development board.

Keywords: *Integrated Development Environment, oOnline Laboratory, FPGA, Client/Server*

1. Introducción

La virtualización de la educación ha cobrado actualmente un interés general importante, dadas las dinámicas sociales y didácticas educativas que representan en este aspecto una consecuencia positiva de la evolución tecnológica. Así mismo, el aprendizaje en línea se manifiesta como una de las tendencias más fuertes en los usos educativos de la tecnología, según el Centro Nacional para Estadísticas de la Educación (2008) en los Estados Unidos de América (U.S DEPARTMENT OF EDUCATION (2009)), lo que revela la aceptación de este esquema en el sector de la educación y particularmente en el nivel superior.

Los laboratorios en línea se han convertido en un apoyo muy útil para los componentes prácticos y experienciales de los métodos de enseñanza en todo el mundo, desde que se tiene el soporte técnico y la disponibilidad tecnológica para su implementación. Particularmente en Colombia, un gran número de instituciones de educación superior presenta dificultades en la adquisición y disponibilidad de los equipos de prueba adecuados para el apoyo didáctico en los procesos de enseñanza/aprendizaje de metodologías de diseño, como ocurre con el área de los sistemas digitales para la Ingeniería Electrónica, debido a los costos que éstos representan y a la infraestructura física que esto demanda. En torno a esta temática se han desarrollado varios proyectos de investigación, planteando la implementación de modelos cliente/servidor para la experimentación remota con hardware reconfigurable, como en Pájaro et al. (2003), Olivares et al. (2005), y Drutarovský et al. (2009), que aportan significativamente en las áreas de diseño digital avanzado y sistemas digitales en general. En el primer caso Pájaro et al. (2003) se utiliza un analizador lógico y un generador de estímulos de 16 canales para una frecuencia máxima de entrada de 25 MHz. Así mismo, para la implementación de circuitos lógicos se utiliza un FPGA Spartan II de Xilinx, donde su configuración e interconexión se controlan a partir de una aplicación Web básica.

Similarmente, en Olivares et al. (2005) se presenta una propuesta de un sistema que permite al estudiante la realización de sus prácticas de laboratorio en forma remota sin consideraciones de calendario. Este sistema utilizaría una placa de experimentación XSA-50 con un FPGA de Xilinx XC2S50 y una cámara web que permite la retroalimentación al estudiante de la información suministrada por el display de 7 segmentos incluido en la placa.

La inspección de los laboratorios en línea existentes actualmente, permite afirmar que las soluciones disponibles pueden mejorarse mediante una óptima utilización de las tecnologías web disponibles, aumentando la usabilidad y la introducción de este tipo de laboratorios en el proceso educativo, como lo establece Dietrich et al. (2005).

Este trabajo se enfoca en el desarrollo de estos sistemas de laboratorio basados en web, orientado a fortalecer los procesos de enseñanza/aprendizaje en el área de sistemas digitales, mediante la programación remota de funciones lógicas; esto lo hace un recurso esencial en las didácticas orientadas a la experimentación en ingeniería. En este artículo se presenta la propuesta de un sistema constituido de componentes software y hardware, donde el principal elemento a acoplar para su configuración es una placa de desarrollo DE2-70 basada en FPGA Cyclone II de Altera, con interfaz USB como vía de comunicación con un PC. El usuario podría compilar, programar y depurar sus diseños en tiempo real y en forma remota a través de una aplicación web amigable, que permita la interacción y presentación de resultados; además, estarían disponibles en servidor los archivos correspondientes al planteamiento de las prácticas de laboratorio, que el docente con permisos de administrador ha subido previamente.

El proyecto presentado explota las herramientas de depuración que posee el IDE Quartus II de Altera, para la configuración y acceso al FPGA, lo que permite, a diferencia de otros trabajos reportados, disminuir costos relativos a la adquisición de equipos externos y desarrollo.

Este artículo consta de cinco secciones: en la sección siguiente se plantea una breve discusión sobre los proyectos más relevantes que han aportado en este campo, en materia de desarrollo y su pertinencia; en la tercera sección se describe la arquitectura general del sistema propuesto, seguido de una sección de resultados a nivel de presentación e implementación y por último, en la quinta sección, se presentan las conclusiones.

1.1 Arquitecturas para laboratorios en línea de sistemas digitales

En el área de los sistemas digitales, para la ingeniería electrónica, informática y afines, el aprendizaje experiencial es fundamental y el diseño de prácticas orientadas al desarrollo de habilidades en los estudiantes de esta línea debe ser dinámico, de tal forma que soporte el rediseño y adaptación de laboratorios y proyectos posteriores, así como el manejo de grupos de trabajo. El componente práctico de esta área sugiere mínimamente el diseño y la implementación de funciones lógicas que determinan el comportamiento de un circuito combinacional y/o secuencial, lo cual requiere del apoyo sobre paquetes computacionales de simulación. Por esta razón, para el aprendizaje en línea con experimentación remota es esencial la integración de una aplicación web que soporte tales condiciones y administre funciones de acuerdo con un servidor de dispositivos donde se conecta la plataforma hardware mencionada.

1.1.1 Proyectos relacionados

El interés de algunos sectores, particularmente el de la educación, en fortalecer e implementar estrategias didácticas para la enseñanza en modalidad a distancia o semipresencial en educación superior, ha promovido la investigación en la proposición de modelos para su implementación en diferentes disciplinas. Así, el acceso remoto a recursos de hardware ha sido objeto de estudio en proyectos como aparece en Kämmerling et al. (2007), Rodríguez (2005) y Callaghan et al. (2005), en los que se plantean arquitecturas Cliente-Servidor para interactuar con un ambiente de aprendizaje integrado que

soporte la experimentación remota de sistemas. En Drutarovský et al. (2009) el sistema se desarrolla específicamente para cursos de diseño digital avanzado y procesamiento de señal usando plataformas FPGA, permitiendo a los estudiantes acceso total a equipos de laboratorio y software licenciado usando una aplicación con interfaz gráfica de usuario basada en LabVIEW a través de un navegador web en un PC remoto. Por otra parte, en Callaghan et al. (2005) se desarrolla mediante el proyecto DIESEL (Distance Internet-Based Embedded System Experimental Laboratory), un modelo de laboratorio de acceso remoto general para sistemas embebidos. Otros proyectos como Fabregas et al. (2011) de la Universidad de Murcia, destacan la interacción de plataformas de adquisición de datos y otras comerciales de propósito específico, con entornos integrados de desarrollo o paquetes software especializados como MATLAB.

El desarrollo de los laboratorios remotos en diversos campos de la ciencia y tecnología ha permitido también la adopción de múltiples arquitecturas para fines similares, razón por la cual ha surgido la iniciativa de adoptar el estándar IEEE1451.0 para las arquitecturas de laboratorios en Web con base en FPGAs (Costa, Alves, & Zenha-Rela, 2011), la cual se constituye en una alternativa muy viable, pero susceptible de ser validada en cada una de las áreas específicas.

El trabajo más reciente lo ha realizado el WebLab-Deusto Research Group (2009) de la universidad de Deusto, con un proyecto de laboratorio remoto que usa tecnología web 2.0, permitiendo alta conectividad, universalidad, seguridad, utilidad y sostenibilidad. Esta investigación parte de la concepción del laboratorio remoto como una infraestructura que combina software, hardware y estrategias didácticas para permitirle al estudiante o usuario completar experimentos físicos (eléctricos, electrónicos, mecánicos, entre otros) desde un lugar distante como si estuviera físicamente en el laboratorio presencial.

El proyecto WebLab-DEUSTO representa un gran aporte en el desarrollo de plataformas software y

de comunicación para el soporte de los laboratorios remotos en diferentes disciplinas de la ingeniería, dado el importante trabajo realizado a nivel de redes y su validación con diferentes experimentos con hardware, lo que implica mucho más trabajo por realizar a este nivel en otras áreas de la ciencia y la ingeniería.

Por otro lado, en el Blekinge Institute of Technology (BTH) se desarrolló un laboratorio remoto para enseñanza de ingeniería eléctrica (Gustavsson, 2002), usando una arquitectura cliente/servidor e instrumentos basados en computador y donde el servidor administra los procesos de medición de acuerdo a lo solicitado desde cliente, retornando posteriormente los datos obtenidos. La conexión de fuentes de poder e instrumentos con computadores soportada por esta arquitectura se hace mediante una antigua interfaz GPIB, mientras los programas del servidor están escritos en LabView 6i y los de clientes en Visual Basic 6. La administración de estos recursos disponibles para los clientes, requiere de múltiples módulos acoplados a cada computador, según la cantidad de instrumentos utilizables, lo cual disminuye la eficiencia del sistema.

Varios de los proyectos incorporan necesariamente una arquitectura cliente/servidor, pero entre los recursos de hardware que requieren están los instrumentos de medida como unidades o tarjetas independientes; en consecuencia, los servidores de recursos deben contar con suficiente soporte en puertos de expansión E/S (tantos como instrumentos sean vinculados al laboratorio), potencia de cómputo, además de que cada uno de los módulos insertados debe suministrar su respectivo driver y aplicación de usuario para proveer sus servicios al cliente.

En Hashemian y Riddley (2007), Indrusiak et al. (2007), Fujii y Koike (2005), Fujii y Koike (1989) y El-Medany (2008) se presentan también aportes significativos con topologías relacionadas y ligeramente similares, destacando que sus plataformas hardware involucran generalmente arquitecturas reconfigurables para programar

bloques funcionales. Así mismo, cada una de los proyectos presentados considera la adopción de un software entorno integrado de desarrollo para la edición, compilación y soporte de aplicaciones de acuerdo con los servicios ofertados, aunque se ilustra poco del diseño e implementación de esta importante fase que define la interacción directa con el usuario.

1.1.2 Arquitectura adoptada para la herramienta propuesta

El sistema propuesto en este artículo está constituido básicamente por un componente hardware y uno software, este último con cinco módulos distribuidos entre interacción con el usuario, configuración del hardware y manejo de datos.

La componente física se basa fundamentalmente en una tarjeta de desarrollo de *Terasic Technologies Inc.*- DE2-70 (2009). El elemento principal de esta placa, que le da el carácter de hardware reconfigurable, es un FPGA de la familia *Cyclone II* de Altera, sobre la cual un estudiante usuario del sistema implementaría el resultado de la etapa de síntesis del diseño que ha desarrollado; esto se hace mediante descarga del archivo de configuración resultante del proyecto, desde la herramienta hacia la tarjeta. La conexión entre esta placa de desarrollo y el equipo servidor se hace mediante USB, soportada por el fabricante de la tarjeta y que garantiza una comunicación bidireccional entre estos elementos, conforme a la interacción del usuario con la aplicación.

La placa de experimentación estaría controlada desde el nivel de aplicación mediante un conjunto de páginas web dinámicas que dan al usuario la posibilidad de compilar, programar y depurar los diseños lógicos que se plantean como parte de los cursos de sistemas digitales y que se definen mediante proyectos a nivel de la herramienta. La componente software de este sistema se encuentra conformada por varios módulos y está desarrollada con los lenguajes de programación PHP, HTML y TCL. La Figura 1 muestra la composición de esta fase.

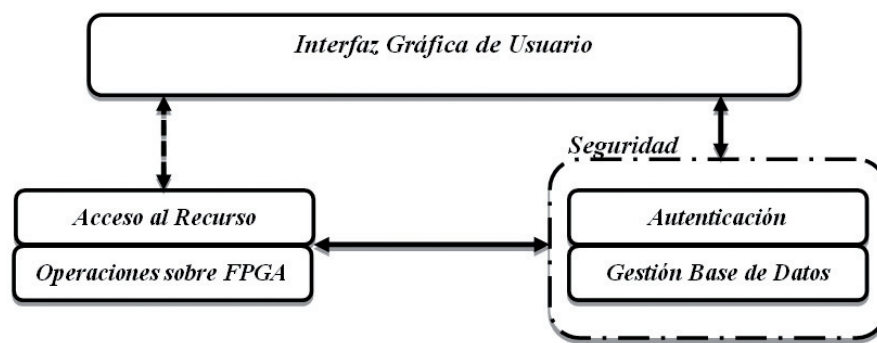


Figura 1. Subsistemas de la componente software del sistema

Es necesario ratificar que el modo empleado para el éxito de esta propuesta es el Cliente/Servidor, pues es en este último donde reside la base de datos con información importante del curso y de los usuarios y es el que dispone del principal recurso hardware que sería compartido a los clientes. Para complementar la perspectiva de la contribución de esta iniciativa en el marco de los laboratorios remotos, en la Tabla 1 se ilustran los aspectos

más relevantes de los aportes realizados por otros trabajos en este contexto y los correspondientes a la herramienta propuesta.

2. Descripción de los módulos de software desarrollados

Tal como se menciona en la sección anterior, se identifica una composición modular en el

Tabla 1. Principales Trabajos en Laboratorios Remotos y sus Aspectos más Relevantes

Referente	Área	Tecnología Hardware	Característica relevante	Base desarrollo Software
[Pájaro et al.(2003)]	Sistemas Digitales	FPGA, Instrumento Embebido	Tarjeta propia, frec. Máx. 25 MHz, 16 canales de muestreo	Java, Delphi, Web - HTML
[Drutarovský et al. (2009)]	Diseño Digital Avanzado	FPGA, Instrumentación externa	Programación gráfica didáctica	Labview
[Callaghan et al. (2005)] - DIESEL	Sistemas Embebidos	Múltiples Tarjetas e Instrumentos con interfaces GPIB	Estructura en red LAN, interfaces RS-232 y paralelo	No especificado
[WebLab-Deusto Research Group]	Varias - Ingeniería	FPGA, uC, CPLD e Instrumentos Virtuales	Tecnología Web 2.0, Servidores de Experimentación	No especificado
[Gustavsson (2002)]	Ingeniería Eléctrica	Instrumentos y fuentes con interfaces GPIB	Múltiples módulos por computador	LabView, Visual Basic
Propuesto	Sistemas Digitales	Tarjetas de Desarrollo existentes o comerciales	Solución de Bajo costo: Adapta aplicación software a recursos existentes	PHP, MySQL, TCL

software desarrollado. Cada elemento que será descrito a continuación tiene unas funciones muy particulares que resultan indispensables para garantizar al usuario el éxito del ciclo de diseño de sus funciones lógicas.

2.1 Interfaz gráfica de usuario

El conjunto de formularios y scripts que conforman la interfaz gráfica de usuario están escritos en lenguajes HTML y PHP. El diseño de este módulo basa su presentación en menús y controles ubicados principalmente sobre el lado izquierdo de la pantalla de cliente, ofreciéndole dos grupos esenciales:

Administración de usuarios y base de datos: a través de formularios se suministra información como código, nombre, contraseña y otros campos que facilitarían el proceso de identificación del usuario en el sistema.

Operaciones con el FPGA: se soportan las opciones de compilación, programación y depuración.

2.2 Seguridad e identificación

Para el acceso al recurso, cada usuario debe identificarse ante el sistema y validar la información respectiva. El mecanismo de autenticación se basa en formularios, lo que consiste básicamente de la existencia de una página de autenticación en el lado del servidor y otra que indique error, si lo hubiere, para informarlo al cliente. Este procedimiento se ilustra con detalle en la Figura 2. Sobre un servidor WEB Apache (The Apache Software Foundation, 2011) se diseña un script PHP para solicitud de *nombre de usuario* y *contraseña*, lo que se constituye como un mecanismo de identificación clásico. Estos campos son devueltos por el formulario de la página realizada y se almacenan en una base de datos protegida por el sistema de gestión y administración propio de MySQL (Oracle Corporation, 2011).

La base de datos almacena información de estudiantes y docentes, para lo que se consideran

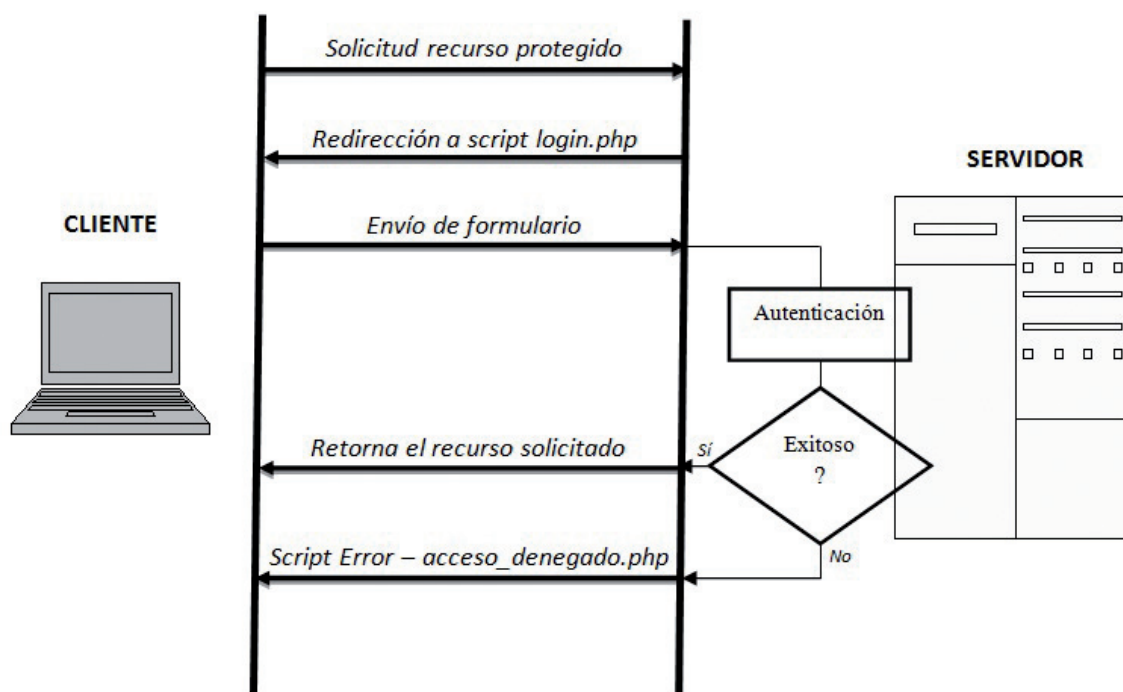


Figura 2. Autenticación basada en formulario

requerimientos mínimos funcionales (Moreno, 2000) relativos a operaciones que se aplican sobre el motor de base de datos y se orientan a la obtención y actualización de datos. Las operaciones que conciernen a este trabajo se dirigen al manejo y administración de citas e incluyen el Ingreso, Búsqueda, Borrado, Edición y Creación de registros. Se utiliza el modelo de esquemas conceptuales Entidad-Relación (E-R) (Moreno, 2000), que describe entidades, vínculos y atributos.

2.3 Acceso al recurso

La administración de citas para los usuarios que acceden al sistema se hace mediante un calendario mensual, que permitiría al docente o administrador la selección de fecha y hora para asignación de la cita correspondiente a cada usuario. Este calendario señala la semana activa, diferenciando los días sábado y domingo con color amarillo. Cada día calendario define un vínculo a una página para solicitar la cita correspondiente al usuario a programar, donde se diligencian campos correspondientes a: nombre de la cita, curso al cual pertenece el estudiante, lista de estudiantes que existen en la base de datos para tal curso, fecha y hora asignada al estudiante para el uso del recurso.

2.4 Operaciones sobre FPGA

Para acompañar el ciclo común de diseño en el área de sistemas digitales, se definen fundamentalmente tres instancias en el manejo del FPGA desde la herramienta diseñada:

Compilación, Programación y Depuración Lógica. El soporte modular de estas instancias es esencial para el manejo de la tarjeta de entrenamiento DE2-70 y el acceso a la configuración del FPGA, con base en las actividades propuestas por el docente para ejecutar y resolver en forma remota desde el diseño hasta la implementación.

Para el soporte de la **Compilación** de un diseño es necesario usar lenguaje TCL ajustado al IDE Quartus (Altera Corporation, 2011); de

esta manera, se utilizan los comandos y rutinas necesarias para realizar una compilación completa a través de una comunicación de fondo con el entorno de desarrollo.

Así, se revisa la sintaxis de los archivos de diseño del usuario y se generan otros tipos de fichero necesarios para programar y depurar el diseño a implementar sobre el FPGA. Uno de los archivos más significativos generados, después del proceso de compilación y síntesis, es el archivo SOF (*SRAM Object File*), el cual es necesario para la etapa de programación y posteriormente la depuración del hardware implementado. Antes de iniciar la rutina de **Programación**, cuyo ingreso se hace mediante el vínculo “programar” de la interfaz de usuario, el sistema verifica con la base de datos si el usuario tiene reservado el acceso al recurso mediante asignación previa de cita. Para esta verificación, se establece inicialmente la zona horaria predeterminada mediante una función en lenguaje PHP, se conecta a la base de datos para hacer la consulta a la tabla “fechas” y luego se restringe la búsqueda en la tabla a aquellas que coincidan con la fecha y hora actual. En tal caso, si la información de cita almacenada en la tabla corresponde con la fecha y hora actual, se concede el acceso al recurso, mientras si no corresponde, el sistema informa el acceso denegado indicando la causa y sugiere el contacto con el docente administrador.

Por otro lado, antes de la programación del FPGA el sistema debe realizar un chequeo general para asegurar las condiciones mínimas de comunicación y disponibilidad del hardware; de esta manera, la aplicación determinaría la existencia de problemas de comunicación u otros inconvenientes con la tarjeta de entrenamiento, antes de permitir la continuidad con el proceso de programación. Consecuentemente, se realiza un conjunto de verificaciones con el fin de determinar el tipo de error que se presenta y así inhabilitar el botón de “Programar” en la interfaz gráfica de usuario e informar las posibles causas de la indisponibilidad del recurso. Si este proceso no encuentra dificultades, el botón de “Programar”

sería habilitado y permitiría al usuario el envío del archivo SOF para programación del FPGA; así mismo, se generaría una indicación al usuario cuando la programación se ha realizado con éxito.

La **Depuración Lógica** permite validar funcionalmente el diseño lógico implementado sobre el FPGA en la etapa de programación. Este módulo interpreta el diseño como un bloque funcional con puertos de entrada y de salida, que intervienen en las operaciones necesarias de conexión y depuración, sin alterar su arquitectura interna, pero revelando al usuario su comportamiento lógico en forma gráfica. El módulo de depuración lógica toma como referencia la entidad de nivel alto del proyecto de diseño del usuario en línea, e identifica los nodos que contiene, como entrada, salida, reloj, reset, entre otros. Una vez los nodos son identificados, el usuario debe seleccionar aquellos a los que debe aplicar estímulos y los que son objeto de observación. Para la generación de estímulos y captura de señales de salida se usa la megafunción *Altsource_Probe* disponible en la IDE

Quartus II (Altera Corporation, 2011). Este módulo dispone de los puertos Fuente (*source*) y Prueba (*probe*), que permiten un tamaño de hasta 256 bits y hasta 128 instanciaciones en el diseño. La interconexión entre el sistema y la megafunción se muestra en la Figura 3. La instanciación de los componentes creados y la conexión con las entradas y salidas definidas por el usuario en su diseño, son indispensables para lograr la generación de estímulos y captura de niveles de salida. Para garantizar este proceso, el sistema genera un script en lenguaje TCL de Quartus II y PHP en el servidor, el cual va a soportar la lectura/escritura en forma remota sobre el FPGA.

3. Resultados y discusión

El equipo servidor que debe atender las solicitudes realizadas por clientes y a su vez reflejar las operaciones que éstas impliquen, debe albergar un sistema de gestión de bases de datos MySQL y el servidor Apache con soporte de PHP. Se obtuvieron datos durante la realización simultánea de un proceso de compilación de un conjunto de ocho

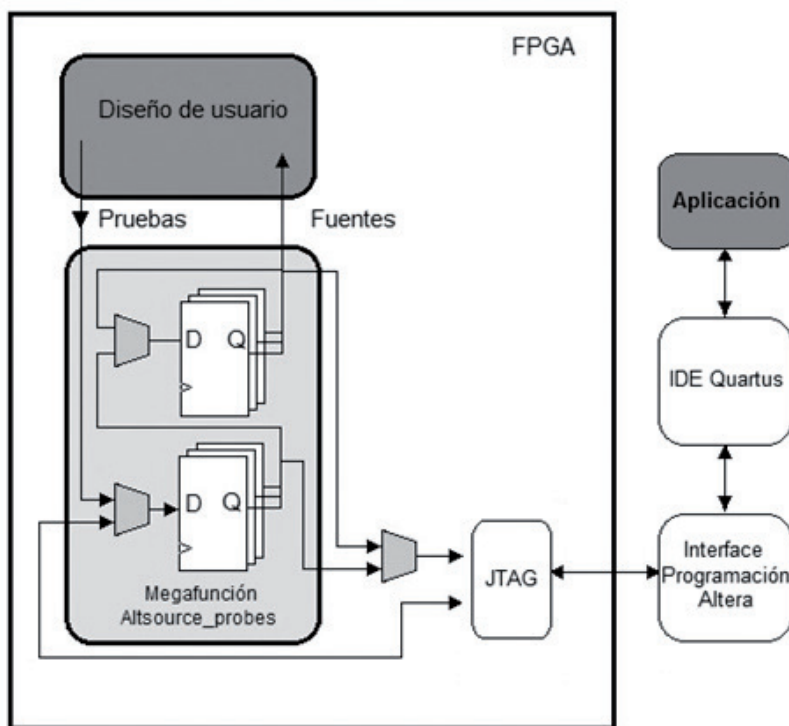


Figura 3. Conexión interna InSystem and Probe de Altera

usuarios, que corresponden a la misma cantidad de clientes abordando el diseño de una Unidad Aritmética y Lógica (ALU) de cuatro bits, el tiempo máximo de compilación por cada usuario para la operación mencionada, fue de 24 segundos. Esto evidencia la necesidad de un equipo de cómputo con excelentes especificaciones de procesamiento y memoria para obtener un buen rendimiento del sistema propuesto, donde se habiliten según la demanda y número de usuarios, los recursos de procesamiento paralelo del servidor.

Si el proceso de autenticación falla, el ingreso y disponibilidad de las operaciones sobre el recurso hardware manejado no es posible, lo cual se informa al usuario. Una vez el proceso de autenticación es validado correctamente, la aplicación presenta al usuario las operaciones disponibles sobre la tarjeta, como son la programación del FPGA, selección de señales o puertos para depuración, entre otras. (Ver Figura 4). Justamente en el proceso de depuración, el usuario selecciona los nodos de entrada y salida necesarios, para establecer la conexión y comunicación con el FPGA; esto se hace con el objetivo de fijar los niveles de estimulación sobre el circuito programado y recibir los resultados lógicos que serían mostrados en un gráfico para

una mejor apreciación por parte del usuario remoto. En la Figura 5, se muestra la página que soporta este procedimiento en la aplicación, con formularios de entrada y la visualización gráfica de los niveles lógicos testeados, para una prueba genérica basada en el diseño de una ALU de 4-bits.

En la Figura 6, se ilustra el formulario de asignación de citas que el docente (o encargado) administraría para otorgar el recurso en forma organizada a un usuario del laboratorio. Los campos que se solicitan son: el título, la selección de área o curso, la descripción de detalles a considerar para la práctica y la asignación de la fecha y hora programada a un usuario que debe estar registrado anteriormente en la base de datos. El proceso de asignación de la cita culmina al presionar el botón “Añadir Cita” y ha sido validado, sin inconvenientes sobre la base de datos, con el acceso simultáneo de hasta siete (7) usuarios en los intervalos críticos cuando el administrador fija estos horarios.

El sistema desarrollado es funcional y actualmente se encuentra en proceso de optimización el desarrollo de un módulo compatible con el analizador lógico interno *SignalTap* de Altera

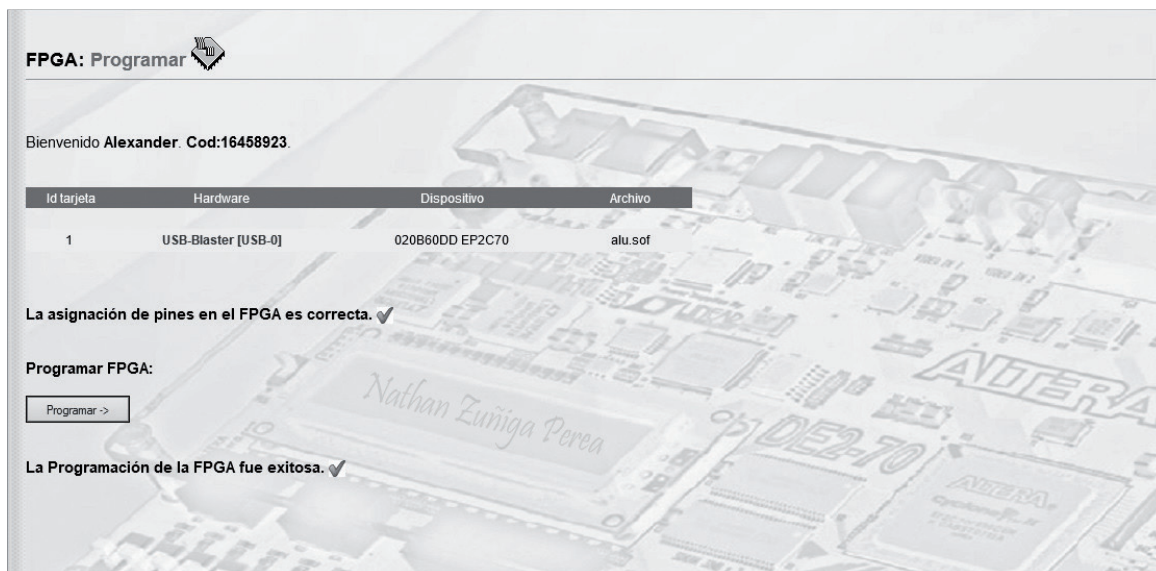


Figura 4. Entorno gráfico de presentación para programación de FPGA

FPGA: Depurador Lógico

Bienvenido Alexander. Cod:16458923.

Nodos encontrados: | alu |

Nodos de Entrada	#Bits
<input checked="" type="checkbox"/> a	4
<input checked="" type="checkbox"/> b	4
<input checked="" type="checkbox"/> cin	1
<input type="checkbox"/> clk	1
<input type="checkbox"/> reset	1
<input checked="" type="checkbox"/> oe	1
<input checked="" type="checkbox"/> s	3

Escoja un nodo de entrada.

Nodos de Salida	#Bits
<input type="radio"/> cout	1
<input checked="" type="radio"/> output	4

Escoja un nodo de salida.

FPGA: Escritura/Lectura de Datos

Indice	Id	Bits	Bits	Nombre
0	0	4	4	insystemprobe:insystemprobe_inst altsource_probe:altsource_probe_component

Nodos de Entrada	#Bits	Datos de Entrada
a	4	1010
b	4	0111
cin	1	1
clk	1	
reset	1	
oe	1	1
s	3	101

Salida	
0	

Salida	
1	
0	
0	
1	

Figura 5. Interfaz de Resultado para depuración - ALU4- bits.

(Altera Corporation, 2011) para su incorporación. De igual manera, se avanza en el proceso de migración a la plataforma Joomla (Portal Joomla Spanish, 2011), la cual como herramienta para gestión de contenidos, ofrece modularidad, usabilidad y seguridad. Este último aspecto fortalecería la herramienta desarrollada como elemento fundamental para la implementación de didácticas del aprendizaje activo en la enseñanza de los sistemas digitales, puesto que se brinda soporte para la integración de foros, actividad en redes sociales, chats y otros servicios orientados a la cooperación en estos procesos.

Finalmente, para evaluar el impacto real del desarrollo de esta herramienta sobre estudiantes y profesores de Sistemas Digitales y hacer un análisis sobre su usabilidad y viabilidad de implementación en las aulas, es necesario establecer un escenario de aplicación para que ellos interactúen con la herramienta y accedan al hardware en forma remota. Dado que la aplicación

está en proceso de finalización, solo se aplicó una prueba parcial, validada mediante encuesta a quince (15) estudiantes y dos (2) profesores de una misma institución y programa académico. En dicha prueba se evaluaron los aspectos técnico, usabilidad de la herramienta y aporte al aprendizaje, con los que se obtuvieron 88.2%, 70.6% y 93.3% respectivamente en favorabilidad. Una vez las mejoras y complementos de la herramienta finalicen y sean adaptadas, será aplicada de nuevo la prueba con un tamaño de muestra más grande y con mayores elementos técnicos para garantizar un entorno más cercano al laboratorio físico presencial.

4. Conclusiones

La arquitectura propuesta incorpora una interfaz cliente/servidor que requiere equipos servidores lo suficientemente robustos a fin de garantizar la demanda de servicios.

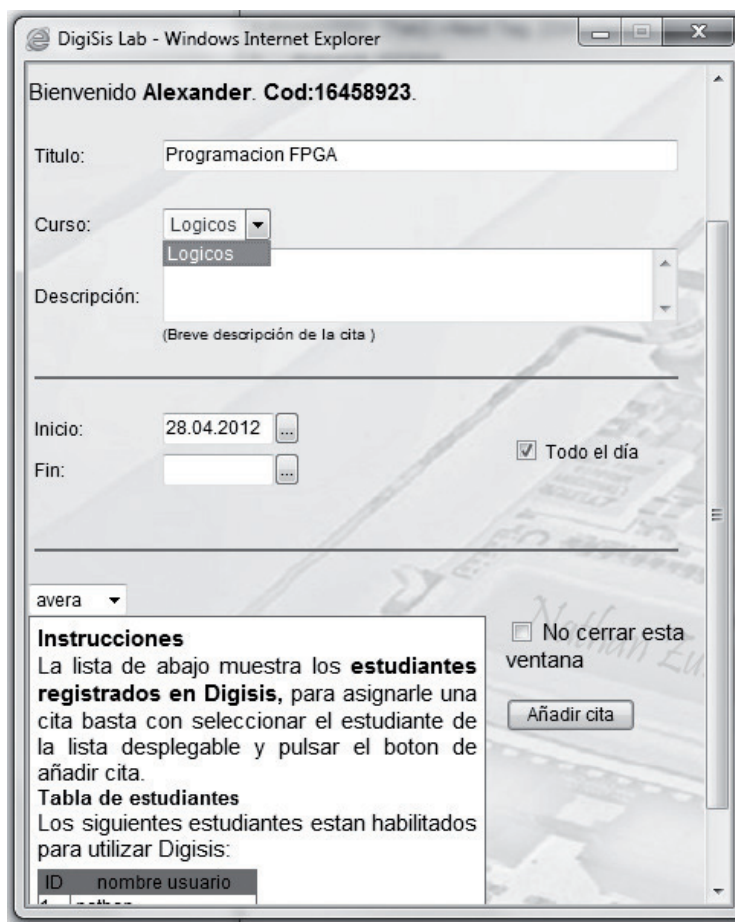


Figura 6. Interfaz para manejo de citas en la herramienta

En este artículo se ha presentado una herramienta computacional para la enseñanza y aprendizaje virtual de los sistemas digitales, que permite motivar al estudiante a realizar sus prácticas de laboratorio de una manera fácil y sencilla, aprovechando las bondades del hardware reconfigurable.

El sistema desarrollado muestra cómo la adaptación de las nuevas tecnologías de la información y las comunicaciones, tales como el hardware reconfigurable, la internet y las aplicaciones software, pueden converger en la implementación de potenciales herramientas pedagógicas.

La herramienta facilita la transformación del modelo de enseñanza presencial tradicional y permite el uso de las NTIC para dinamizar

los procesos de aprendizaje, sin abandonar las prácticas de laboratorio necesarias en la enseñanza de la ingeniería.

Los servicios soportados con la aplicación desarrollada permiten que tanto docentes como estudiantes puedan depurar sus circuitos lógicos en forma remota, con una herramienta que les enfoca en su diseño, sin que deban preocuparse adicionalmente por las conexiones o instancias necesarias que deben ser creadas a nivel de hardware para la fase de depuración, ya que el sistema realiza este tipo de conexiones de manera tal que sea transparente al usuario.

5. Referencias bibliográficas

ALTERA CORPORATION 2010. *Quartus II Handbook Version 9.1* Volume 3, Visto en Enero

de 2010, <<http://www.altera.com/literature/lit-an.jsp>>.

Callaghan, M.J., Harkin, J., McGinnity, M., & Maguire, L. (2005). Client-Server Architecture for Remote Experimentation for Embedded Systems. *iJOE International Journal of Online Engineering*. Intelligent Systems Engineering Laboratory. University of Ulster, Derry, Northern Ireland.

Costa, R., Alves, G., & Zenha-Rela, M. (2011). Work-in-Progress on a Thin IEEE1451.0-Architecture to Implement Reconfigurable Web-lab Infrastructures. Vol. 7, iJOE- International Journal of Online Engineering. p. 5-10.

Dietrich, H., Henke, K., & Ludwig, N. (2005). *Remote Labs versus Virtual Labs for Teaching Digital System Design*. International Conference on Computer Systems and Technologies - CompSysTech.

Drutarovský, M., Šaliga, J., Michaeli, L., & Hroncová, I. (2009). *Remote laboratory for FPGA based reconfigurable systems testing*. XIX IMEKO World Congress Fundamental and Applied Metrology, p. 54-58.

El-Medany, W.M. (2008). *FPGA Remote Laboratory for Hardware E-Learning Courses*. University of Bahrain, Computer Engineering Dept., Sakhir, Bahrain – IEEE Region 8 SIBIRCON.

Fabregas, E., Farias, & G., Dormido, S. (2011) *Developing a Remote Laboratory for Engineering Education*. *Computers & Education*, Vol 57, Issue 2, p. 1686-1697.

Fujii, N., & Koike, N. (2006). *New Virtual Remote Laboratory Environments for Logic Circuit Design Realizing an efficient sharing of Test Equipments and Concurrent User Support*. In proceeding 7th International Conference on Information Technology Based Higher Education and Training (ITHET'06). P. 517-522.

Fujii, N., & Koike, N. (2005) Work in Progress - *A New Time-sharing Remote Laboratory e-Learning System for Hardware Design and Experiment of Digital Circuits*. Hosei University, Faculty of Computer and Information Sciences, Tokyo, Japan. 35th ASEE/IEEE Frontiers in Education Conference, Indianapolis.

Gustavsson, I. (2002). *Remote Laboratory Experiments in Electrical Engineering Education*. Fourth IEEE International Caracas Conference on Devices, Circuits and Systems, Aruba.

Hashemian, R., & Riddley, J. (2007). *FPGA e-Lab, a Technique to Remote Access a Laboratory to Design and Test*. IEEE International Conference on Microelectronic Systems Education (MSE'07), IEEE Computer Society.

Indrusiak, L.S., Glesner, M., Reis, R. (2007). *On Evolution of Remote Laboratory for Prototyping Digital Electronic System*. IEEE Transactions on Industrial Electronics 54 (6).

Kämmerling, P., Ackens, A., Loevenich, H., Borga, A., Wüstner, P., Kemmerling, G., Erven, W., Zwoll, K., Kleines, H., & Drochner, M. (2007). *FPGA Configuration by TCP/IP and Ethernet*. Visto en Diciembre de 2009, <http://conferences.fnal.gov/cgi-bin/rt2007/download.pl?paper_id=PS1C002&wanted_file=PS1C002.PDF>.

Moreno O., Antonio. (2000) *Modelado conceptual de bases de datos relacionales: El modelo E/R* [Online]. Visto en Enero 2010, <<http://elies.rediris.es/elies9/5-1.htm>>.

Olivares, J., Merino, A., Palomares J. M., & Montijano M. A. (2005). “*Laboratorio virtual para la programación de FPGAs*”. VII Simposio Internacional de Informática Educativa-SIIE.

OpenLabs Electronics Laboratory. (2010). *VISIR – Virtual Instrumental Systems in Reality*. OpenLabs Electronics Laboratory. Visto en Enero de 2010, <<http://openlabs.bth.se/index.php?page=ElectroLab>>.

Oracle Corporation. (2011). MySQL Documentation: *MySQL Reference Manuals*. Visto en Enero de 2010, <<http://www.mysql.com/>>.

Pájaro, J., Torres, C., Nieto, R., Duque, E., Bernal, A., & Aedo, J.E. (2003). “*Desarrollo de un sistema para el prototipaje en forma remota de sistemas digitales basados en FPGAs usando Internet*”. CINTEL, COLCIENCIAS, p. 386-392.

Portal Joomla! Spanish. (2011). *Documentación: Manual de usuario*. Visto en Diciembre de 2010, <<http://www.joomlaspanish.org/>>.

Rodríguez -E, Luis A. (2005). *Desarrollo de un sistema de monitoreo y control electrónico remoto vía Internet basado en lenguaje java*. Tesis de Grado, Escuela de Ingeniería Eléctrica y Electrónica Universidad del Valle, Cali, Colombia.

TERASIC (2009). ALTERADE2-70 *Development and Education Board. User Manual*. Visto en Diciembre de 2009, <<http://www.terasic.com.tw>>.

The Apache Software Foundation. (2011). *Apache HTTP Server Documentation: Authentication, Authorization and Access Control*. Visto en enero de 2011, <<http://httpd.apache.org/docs/trunk/howto/auth.html>>.

U.S. DEPARTMENT OF EDUCATION. (2009). *Evaluation of Evidence-Based Practices in Online Learning, Center for Technology in Learning*. Visto en Diciembre de 2009, <<http://www.ed.gov/about/offices/list/oepd/ppss/reports.html>>.

WebLab-DEUSTO Research Group. Universidad de Deusto. *Weblabdeusto, Bilbao*. Visto en enero de 2010, <<https://www.weblab.deusto.es/joomla/>>.