

DESARROLLO DE UN TRAZADOR DE CURVAS COMO UN CASO DE APLICACIÓN DE INSTRUMENTOS BASADOS EN INSTRUMENTACIÓN VIRTUAL RECONFIGURABLE

DEVELOPMENT OF CURVE TRACER AS APPLICATION OF INSTRUMENTS BASED UPON RECONFIGURABLE VIRTUAL INSTRUMENTATION

JUAN VEGA¹ Y MIGUEL RISCO²

¹ GRUPO DE MICROELECTRÓNICA-CENTRO DE INVESTIGACIÓN Y DESARROLLO EN INGENIERÍA (CIDI), FACULTAD DE INGENIERÍA ELECTRÓNICA Y MECATRÓNICA, UNIVERSIDAD TECNOLÓGICA DEL PERÚ (UTP), LIMA 01, PERÚ

² CENTRO DE INVESTIGACIÓN Y DESARROLLO EN INGENIERÍA (CIDI), FACULTAD DE INGENIERÍA ELECTRÓNICA Y MECATRÓNICA UNIVERSIDAD TECNOLÓGICA DEL PERÚ, LIMA 01, PERÚ; COLABORADOR DEL PROYECTO RVI DEL LABORATORIO MULTIDISCIPLINARIO DEL CENTRO DE FÍSICA TEÓRICA (ICTP), TRIESTE, ITALIA.

RESUMEN

El presente trabajo presenta un caso de aplicación de una plataforma versátil basada en una FPGA validada, con una arquitectura que hace posible su reutilización y promueve el intercambio del firmware y software que lo transforma en diversos instrumentos electrónicos de laboratorio. En este caso específico se ha diseñado un instrumento que permite la caracterización mediante el trazado de curvas I vs V de dispositivos semiconductores. Una tarjeta denominada RVI (Reconfigurable Virtual Instrument) diseñada en el MLAB del Centro Internacional de Física Teórica y una tarjeta adaptadora de señal constituyen el hardware del sistema, mientras que el software ha sido desarrollado usando Lazarus. El código para configurar la FPGA se ha realizado en VHDL. Este diseño pretende contribuir con el proyecto RVI, iniciado por el ICTP, en el año 2006, desarrollando un instrumento virtual dentro de esta plataforma.

Descriptores: instrumentación electrónica, laboratorio virtual, instrumentación virtual, trazador de curvas, tarjeta RVI, sistemas reconfigurables

ABSTRACT

The work presents an application case of a versatile platform based on FPGA that has been validated, its architecture makes possible its reuse and promotes the exchange of the firmware and software that transform it in diverse laboratory electronic instruments. This paper describes an instrument that allows the characterization of semiconductor devices by means of its "I vs. V" curve. A board named RVI (Reconfigurable Virtual Instrument) developed at the MLAB of the International Centre for Theoretical Physics and a signal adapter board makes the hardware of the system while the software was developed using Lazarus. The code to configure the FPGA was making using VHDL. This design is intended to contribute to the RVI project initiated by the ICTP in 2006, with developing a virtual instrument in this platform.

Keywords: electronic instrumentation, virtual laboratory, virtual instrumentation, curve tracer, RVI board, reconfigurable system.

INTRODUCCIÓN

En laboratorios de investigación se requiere constantemente un equipo que realice una medición o adquisición de información para un experimento o trabajo específico. Sólo los equipos más costosos admiten una configuración suficientemente personalizable, y por ello muchas veces los investigadores de centros con bajos recursos deben confeccionar sistemas electrónicos a medida.

El desarrollo de un instrumento personalizado es una tarea que consume tiempo y recursos, no sólo en el diseño y construcción del mismo sino también en su validación, y pocas veces se basan en un hardware que permita su reutilización.

El Laboratorio de Microprocesadores (hoy Laboratorio Multidisciplinario MLAB) desarrolló, hacia el año 2006, un interesante proyecto basado en el diseño de un sistema denominado Instrumentación Virtual Reconfigurable, RVI por sus siglas en inglés, el cual tiene como componente hardware una tarjeta electrónica programable y el software adecuado que hace de interface entre el usuario y el sistema. La idea principal es que la tarjeta electrónica posea la suficiente flexibilidad como para implementar distintos tipos de instrumentos, un rico abanico de interfaces y la potencia necesaria para ser considerada seriamente como alternativa a algunos instrumentos de laboratorio. Por otra parte, se estimularía la creación de software de libre disponibilidad y de una comunidad académica de usuarios que intercambie experiencias y desarrollo.

En este trabajo se describe la aplicación de una plataforma versátil basada en una FPGA validada. Su arquitectura hace posible su reutilización, y se promueve el intercambio del firmware y software que lo transforma en diversos instrumentos electrónicos de laboratorio. En este caso específico se ha diseñado un instrumento que permite la caracterización mediante el trazado de curvas I vs V de dispositivos semiconductores. Una tarjeta denominada RVI [1] (Reconfigurable Virtual Instrument), diseñada en el Laboratorio Multidisciplinario (MLAB) del Centro Internacional de Física Teórica (ICTP, Trieste, Italia) y una tarjeta adaptadora de señal, diseñada por los alumnos del Centro de Investigación y Desarrollo en Ingeniería (CIDI), constituyen el hardware del

sistema, mientras que el software ha sido desarrollado en Free Pascal usando la plataforma Lazarus, una herramienta de desarrollo de aplicaciones gratuita disponible para las plataformas Windows, GNU/Linux y Mac OS-X. El código para configurar la FPGA se ha realizado en VHDL, haciendo uso del entorno integrado de desarrollo Libero que la compañía Actel [2] proporciona en su versión gratuita.

MÉTODO EXPERIMENTAL

Hardware RVI

La tarjeta RVI (fig. 1) cuenta con una FPGA de alto rendimiento. La flexibilidad de su hardware la hace ideal para el desarrollo de instrumentos de laboratorio, y cuenta con una variedad de interfaces disponibles como:

- Puerto paralelo,
- Puerto serie,
- Conector de red Ethernet y
- Conector USB

Posee también visualizadores LED de cuatro dígitos a siete segmentos y cinco pulsadores que ayudan en la depuración.



Figura 1: Tarjeta de Instrumentación Virtual (RVI).
Fuente: ECI PERU 2009[4].

“En cuanto a capacidades de expansión, la RVI posee un zócalo para la colocación de un módulo de memoria SDRAM SODIMM que permitiría almacenar información a una velocidad mayor que la posible vía comunicación con la PC. También posee un par de conectores de alta calidad de 54 pines con conexión directa a los puertos de propósito general de la RVI y señales de alimentación. Para estos conectores existen actualmente dos tarjetas adicionales (daughter boards) que añaden la capacidad de adquirir y generar señales analógicas:



Figura 2: Tarjeta Low Performance.
Fuente: ECI PERU 2009[2].

Low Performance Daughter Board (LPDB) [con] dual channel 10-bits 20 MSPS ADC (AD9201, Analog Devices) [y] dual channel 14-bit 1 MSPS DAC (LTC1654, Linear) (fig. 2), y High Performance Daughter Board (HPDB) [con] single channel 14-bits 125 MSPS ADC (LTC2255, Linear) [y] single channel 16-bit 50 MSPS DAC (LTC1668, Linear).

En cuanto al modo de programación, la RVI posee una interface JTAG compatible con los programadores USB de Actel, y también la capacidad de ser programada vía el interface paralelo” [3].

Arquitectura del software de la RVI

A diferencia de un microprocesador o microcontrolador, en una FPGA las estructuras que definen los bloques funcionales dentro del chip son programables, es decir, los registros, memorias, ALU, buses, interfaces, puertos, etc. pueden ser “descritos” mediante el uso de un código que se encarga de unir adecuadamente celdas que poseen determinada funcionalidad lógica y que al ensamblarse forman bloques y funciones más complejas. La sintaxis de dicho código se expresa en un lenguaje de alto nivel de descripción de hardware (HDL); casos particulares de implementación de este lenguaje lo constituyen el VHDL y el Verilog. Mediante un lenguaje HDL es posible definir sistemas tan simples como funciones

lógicas unidas a una máquina secuencial, o tan complejos como múltiples microprocesadores que trabajan en paralelo interconectados mediante buses personalizados y unidos al exterior por interfaces hechas a medida dentro de una única FPGA. Por ello, es fácil deducir que la principal limitación para elaborar un complejo diseño digital la constituye la cantidad de celdas lógicas disponibles en la FPGA.

El diseño realizado para el trazador de curvas se basa en una estructura de bloques (IP Cores) independientes, similar a los SoC (System on Chip). No se ha implementado un microprocesador como maestro del sistema; en su lugar una máquina de estado compleja administra un bus, con la finalidad de ahorrar recursos en la FPGA. Esta máquina de estado compleja que hace las veces de controlador del sistema se encuentra dentro de una disposición a la que se la denomina SBA (Simple Bus Architecture) o Arquitectura Simple de Bus, la cual no es más que una simplificación de las señales y reglas que establece la especificación Wishbone [4].

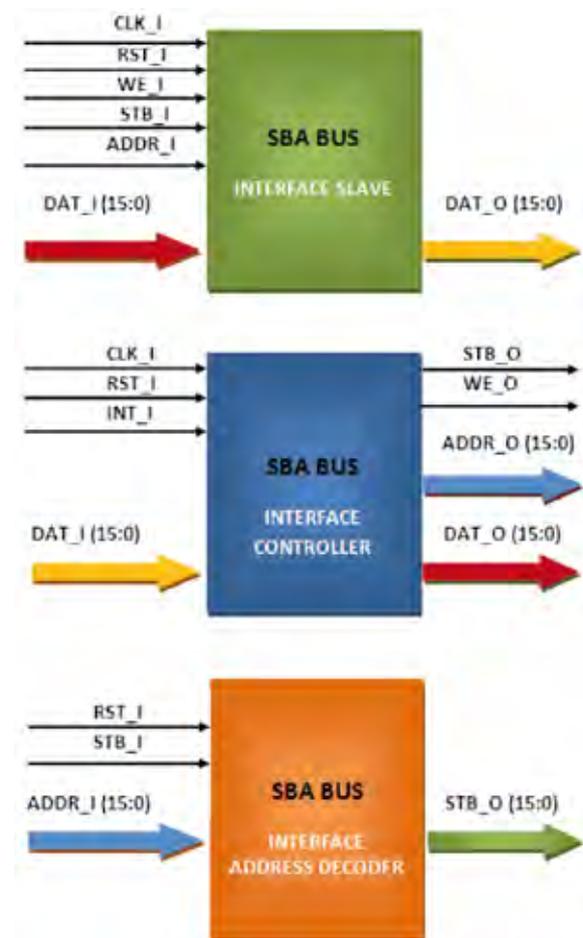


Figura 3: Ejemplo de posibles implementaciones de los núcleos siguiendo SBA.

SBA

La Arquitectura Simple de Bus, o SBA, permite interconectar diferentes componentes o bloques unos a otros, de una manera práctica y sencilla, permitiendo al usuario optimizar tiempo a la hora de realizar sus proyectos. Asimismo, sacrifica las complejidades que muestran otras implementaciones con la finalidad de introducir con facilidad el concepto SoC en la FPGA, por lo que tiene un valor educativo inherente.

La SBA pretende ser una interface de propósito general; como tal, define el intercambio de datos entre módulos estándar de IP Cores.

Se divide en tres tipos de bloques principales: maestro o controlador del sistema (SBA Controller), núcleos esclavos y núcleos de apoyo al bus (decodificador de direcciones, adaptadores de bus, generadores de reloj, etc.).

Descripción funcional de las señales:

- *RST_I*: Recibe la señal de reset principal del sistema;
- *CLK_I*: Recibe la señal de reloj de sincronismo del sistema;
- *ADR_I*: Recibe la dirección del maestro o controlador del sistema (SBA Controller);
- *ADR_O*: Envía la señal a los demás bloques;
- *DAT_I, DAT_O*: Bus de datos;
- *INT_I*: Solicitud de Interrupción;
- *WE_I, WE_O*: Habilitación de escritura;
- *STB_I, STB_O*: Selector de núcleo (*Strobe*).

Descripción funcional de los bloques:

- *Controlador de sistema (SBA Controller)*: Es el bloque principal, pues aquí se asigna una dirección para cada acción a realizar. Se encarga también de decidir el orden de dichas acciones (decide qué se ejecuta primero y qué después). Este direccionamiento se comunica directamente con el Decodificador de direcciones (Address Decoder).
- *Decodificador de direcciones (Address Decoder)*: Asigna a cada núcleo esclavo una posición en el mapa de direcciones y habilita dichos núcleos mediante el STROBE (Chip Select).

- *Esclavos*: Periféricos a usar (componentes, bloques, IP Cores). Envían y reciben datos.

Estructura del sistema descrito en VHDL

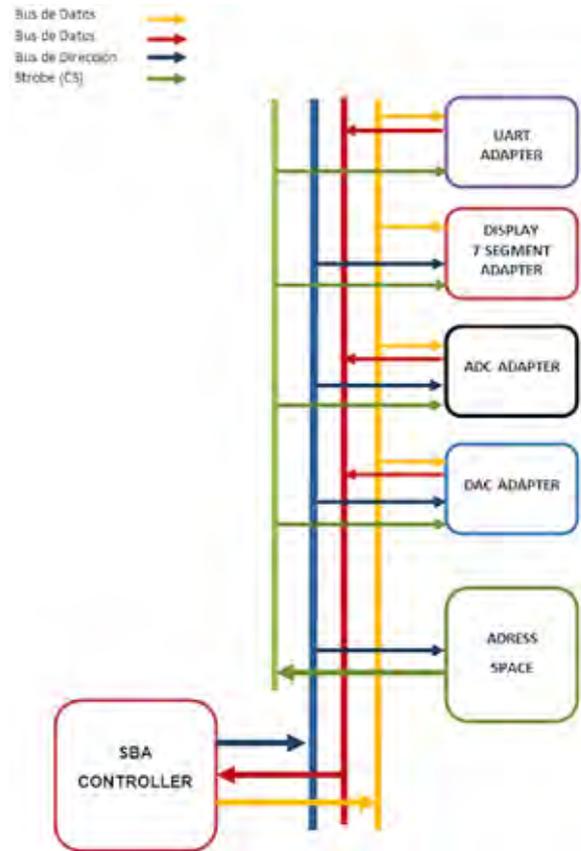


Figura 4: Diagrama de la descripción en VHDL para el trazador de curvas.

Descripción funcional de los bloques:

- *Espacio de direcciones (Address Space)*: Genera el mapeo de la direcciones, y a cada esclavo una línea para su habilitación mediante una señal STROBE (Chip Select).
- *Controlador de sistema (SBA Controller)*: Realiza la administración general del bus. Dirige el orden del flujo de la información según la secuencia determinada por su máquina de estado interna.
- *Adaptador DAC (DAC Adapter)*: Bloque para controlar el Conversor Digital Análogo LTC1654 de 14 bits de resolución que viene integrado en la tarjeta Low Performance Daughter (fig. 2).
- *Adaptador ADC (ADC Adapter)*: Bloque para controlar el Conversor Análogo Digital AD9201

de 10 bits de resolución que viene integrado en la tarjeta Low Performance Daughter (fig. 2).

- *Adaptador de Display de Siete Segmentos*: Bloque para controlar los cuatro displays de siete segmentos que vienen integrados en la RVI.
- *Adaptador UART (UART Adapter)*: Este bloque se encarga de la comunicación serial, tanto para el envío como para la recepción de datos.

Hardware de la placa de adaptación

La placa de adaptación está diseñada para la adquisición de datos de la I_c y V_{ce} , de transistores bipolares tipo NPN y PNP y otros dispositivos semiconductores similares.

La conexión es de Tipo RCA, por ser fácil de manipular y de bajo costo.



Figura 5: Hardware: Placa de adaptación.

Esquema circuital

El diseño del esquemático y el layout de la placa adaptadora se realizó en el programa Proteus, de la empresa LabCenter Electronics. Este software también proporcionó las herramientas necesarias para la simulación del circuito, tomando como base las capacidades de la tarjeta de adquisición de datos de la RVI.

Usando la configuración de OPAM seguidor y ajustando un voltaje de referencia a la entrada no inversora del OPAM ($1/2$ del voltaje máximo de salida del DAC), en la salida del OPAM se tendrá un voltaje que será nuestra tierra flotante, y que se

usará como referencia en el circuito del transistor (fig. 6).

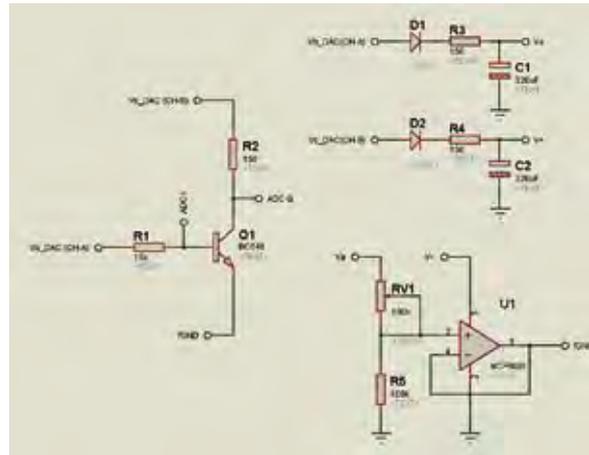


Figura 6: Diagrama circuital de la placa de adaptación.

En el diseño no se está considerando una fuente de alimentación externa para alimentar el circuito del OPAM. En su reemplazo se usará las dos salidas del DAC LTC1654 (CH-A y DAC CH-B) que, conjuntamente con un arreglo de diodos y capacitores, puedan alimentar al circuito que proporcionará la tierra flotante.

Software Curve Tracer

Para obtener las curvas del transistor, se debe entregar un barrido de tensión (V_c) en forma de diente de sierra al transistor en prueba, al mismo tiempo que se fija una corriente en la base (I_b).

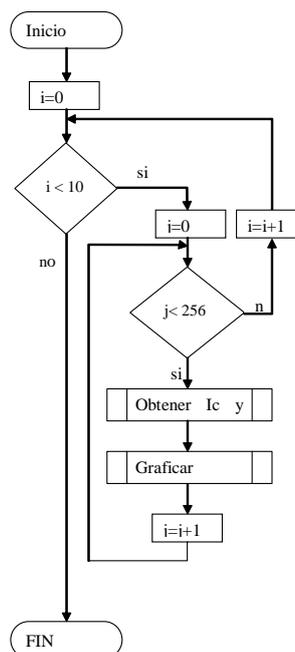


Figura 7: Diagrama de flujo del software "Curve Tracer".

La tensión de barrido para el colector se obtiene incrementando el valor en el DAC de la RVI, con una amplitud que podemos variar entre 0 y 3,3 voltios. Cada barrido se hace con una corriente de base fija.

En la figura 7, la variable *i* representa el número de curvas que podremos graficar, y la variable *j* la tensión de barrido por cada valor de *i*.

Cálculos del voltaje de base

El conversor D-A LTC 1654 tiene una resolución de 14 bits, de los cuales se usará para la aplicación solo los 8 bits más significativos (b13 – b6).

Al trabajar solo con ocho bits, podremos variar de 0X00 a 0XFE, esto quiere decir que podremos tener un voltaje de 0 a 3,3V en 256 pasos.

En el circuito de la figura 6, la salida del DAC CH-A, al estar conectada en serie con una resistencia a la base del transistor, genera una corriente de base (*I_b*). Para distintos valores de corrientes de base, se tendría que cambiar el valor programado en el DAC CH-A (tabla 1).

Los valores de voltaje de base y su valor equivalente en Hexadecimal serán programados en el DAC, para generar las diferentes corrientes de base.

Tabla 1: Valores de voltaje de base.

I_b (μ A)	R_b (K Ω)	V_{max} DAC	Valor de voltaje	Valor en HEX
4	15	3,3	0,63	31
6	15	3,3	0,66	33
8	15	3,3	0,69	36
10	15	3,3	0,72	38
12	15	3,3	0,75	3A
14	15	3,3	0,78	3D
16	15	3,3	0,81	3F
18	15	3,3	0,84	41
20	15	3,3	0,87	43
22	15	3,3	0,9	46

Implementación del sistema

Una vez que los bloques del sistema han sido

implementados y verificados, el siguiente paso es obtener los valores de corriente y voltaje para graficar las curvas *I* vs. *V* del dispositivo semiconductor. Esos valores serán obtenidos de la tarjeta Low Performance Daughter Board de la RVI.

Descripción del funcionamiento

El dispositivo semiconductor elegido como ejemplo y del cual se obtendrán sus curvas características, será un transistor bipolar tipo NPN.

Dentro de la FPGA, existe una estructura física descrita en lenguaje VHDL (fig. 4). El programa realizado en el entorno de Lazarus, denominado “Curve Tracer”, enviará los valores de *V_b* (tabla 1) y *V_c* por el puerto serial a la FPGA.

El Adaptador UART recibe los valores de *V_b* y *V_c* (8 bits) provenientes del puerto serial de la RVI, y luego el Controlador SBA lee los datos y manda las señales al Espacio de direcciones para habilitar al Adaptador DAC para escritura. Los datos de *V_b* y *V_c* se escribirán en los dos DAC (CH-A y CH-B) del LTC1654, el valor equivalente del voltaje en bits.

Fijado los valores en los DAC CH-A y CH-B, se esperará un tiempo para la estabilidad de la salida de los DAC; luego el Controlador SBA mandará las señales al Espacio de Direcciones (Address Space) para habilitar al Adaptador ADC y proceder con la lectura de los valores del ADC Q y el ADC I, que contienen los datos de la *I_c* y *V_{ce}* del transistor.

El Controlador SBA mandará las señales al Espacio de direcciones para habilitar el Adaptador UART, quien enviará los valores por el puerto serial al programa “Curve Tracer”, que graficará el punto.

Para obtener una curva completa se tendría que fijar una *V_b* (este generará una corriente de base) y hacer variar el *V_c* de 0 hasta su valor máximo.

RESULTADOS

Con los datos de *V_b* obtenidos para cada corriente de base, se realizó la adquisición de datos con la RVI, el hardware del trazador de curvas y el programa “Curve Tracer”, y se obtuvo las curvas del transistor (fig. 8).

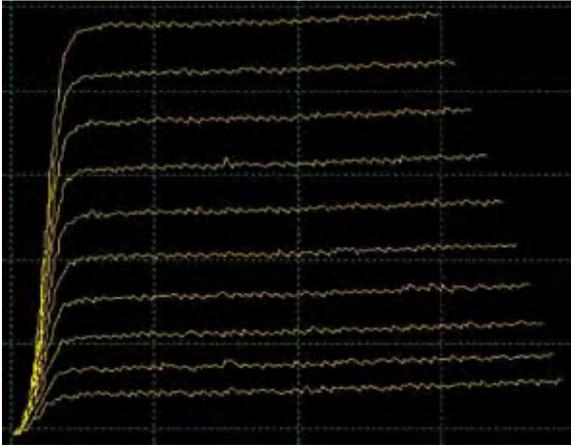


Figura 8: Curva característica del transistor bipolar BC547.

La figura 8 corresponde a la variación de I_c respecto al V_{ce} de un transistor bipolar tipo NPN. Se puede apreciar que a partir de la segunda curva, el voltaje de colector-emisor medido por el conversor A-D no llega al final del rango del eje x, y que esto va creciendo de curva en curva hasta la décima. Esto se debe a que a medida que se incrementa la corriente en la base, se incrementa de manera proporcional la corriente en el colector Eq. (1).

$$\beta \cdot I_b = I_c \quad (1)$$

Esto ocasiona el aumento de la caída de potencial en la Resistencia del Colector (R_c), lo cual implica una disminución en el valor del Voltaje colector-emisor (V_{ce}), que efectivamente se puede comprobar mirando la gráfica.

Se puede apreciar que las curvas tienen pequeñas deformaciones a lo largo del gráfico, pero aún así se puede ver la forma característica de las curvas del transistor. Estas deformaciones se deben a la existencia de ruido en el momento de la adquisición. Actualmente se está trabajando en métodos de *oversampling* y filtrado digital, así como el análisis del modelo SPICE del transistor con la finalidad de extraer los parámetros cualitativos y cuantitativos de los resultados obtenidos y realizar un ajuste de curva adecuado, lo que permitiría en el futuro obtener curvas con mayor precisión y detalle.

CONCLUSIONES

La placa Adaptadora de señales ha sido construida con dispositivos comunes que no son difíciles de encontrar en el mercado. Su diseño simple permitió reducir su tamaño y ahorrar en los costos de fabricación.

La flexibilidad del diseño en las FPGAs hace que la Tarjeta RVI, junto con el Bus SBA, sea ideal para poder diseñar distintos tipos de Instrumentos Virtuales en corto tiempo.

Aunque el sistema trazador de curvas es aún un trabajo en progreso, los datos adquiridos nos han permitido validar la arquitectura empleada y demostrar la versatilidad del sistema.

AGRADECIMIENTOS

Al Ing. Miguel A. Risco Castillo, docente investigador del Centro de Investigación y Desarrollo en Ingeniería (CIDI), por su paciencia y excelente trabajo.

A los diseñadores de la RVI: A. Cicuttin, M.L. Crespo y A. Shapiro

Al Centro de Investigación y Desarrollo en Ingeniería de la Facultad de Electrónica y Mecatrónica de la Universidad Tecnológica del Perú, que nos permite continuar con nuestra formación en investigación y desarrollo.

REFERENCIAS

- [1] RVI. Instrumentación Virtual Reconfigurable [En línea] <http://mlab.ictp.it/rvi/system.html>
- [2] Actel (r), portal web. [En línea] <http://www.actel.com>
- [3] M.Risco, Desarrollo de instrumentos basados en Instrumentación Virtual Reconfigurable, ECI PERU 2009 Vol. 6, Número 2 pp. 22-27
- [4] SoC Interconnection: Wishbone. [En línea] <http://www.opencores.org/?do=wishbone>

E-mail: juan.vega25@gmail.com