

# Desarrollo de Instrumentos basados en Instrumentación Virtual Reconfigurable

## Development of instruments based on Reconfigurable Virtual Instrumentation

---

*Miguel A. Risco Castillo (\*)*

---

### RESUMEN

Un equipo con el cual se pueda realizar una medición o adquisición de información para un experimento o trabajo específico ha sido siempre un requerimiento constante en un laboratorio de investigación, los instrumentos convencionales de bajo costo no siempre permiten una personalización adecuada, y sólo los equipos más costosos admiten una programación o configuración suficientemente personalizable; por ello muchas veces los investigadores de centros con bajos recursos se ven en la necesidad de confeccionar ellos mismos pequeños 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. Estos equipos personalizados pocas veces son basados en un hardware normalizado que permita su intercambio con otros investigadores o incluso su reutilización. El presente trabajo propone una plataforma versátil basada en un hardware de bajo costo que ha sido previamente validado y sobre el cual se pueden emular diversos instrumentos, su arquitectura hace posible su reutilización e inclusive se promueve el intercambio del firmware y software que lo transforman en diversos instrumentos electrónicos de laboratorio..

### ABSTRACT

An equipment, which can be used to make a measurement or acquisition of data for an experiment or specific work has always been a constant requirement in an investigation laboratory, the conventional low cost instruments don't always allow an appropriate personalization, and only the most expensive equipments permit a programming or enough level of customization; for it many times the investigators of scientific centres with low resources have the necessity of making themselves small specific electronic systems. The development of a personalized instrument is a task that consumes time and resources, not only in the design and construction of this but also in its validation. These equipments few times are based on a normalized hardware that allows their exchange with other investigators or even their reuse. The present work proposes a versatile platform based on a low cost hardware that has been previously validated and on which diverse instruments can be emulated, its architecture makes possible its reuse and inclusive it is promoted the exchange of the firmware and software that transform it in diverse electronic laboratory instruments.

### INTRODUCCIÓN

El Laboratorio de Microprocesadores (MLAB) del Centro Internacional de Física Teórica (ICTP) Abdus Salam, desarrolló hacia el año 2006 un interesante proyecto basado en el diseño de un sistema denominado Instrumentación Virtual Reconfigurable «RVI»

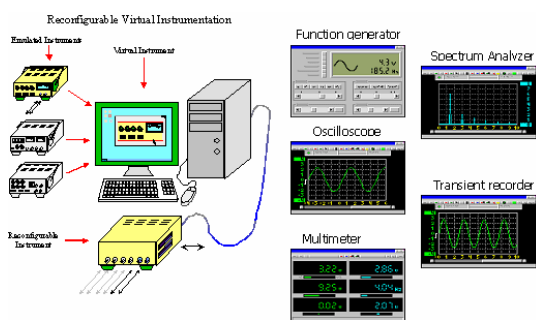
[1] el cual tiene como componente hardware una tarjeta electrónica programable y el software adecuado que hace de interface entre el usuario y el sistema.

La idea principal es que la tarjeta electrónica posea la suficiente flexibilidad como para implementar distintos tipos de instrumentos,

---

(\*) Docente Investigador del Centro de Investigación y Desarrollo en Ingeniería (CIDI), Facultad de Ingeniería Electrónica y Mecatrónica de la Universidad Tecnológica del Perú. Visitor Scientists and collaborator of the RVI Project from the Microprocessor Laboratory of the International Centre of Theoretical Physics (ICTP), Trieste, Italy

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 experiencia y desarrollo. Desde el punto de vista del usuario final, el hardware debería considerarse como una caja negra en donde se cargaría el firmware necesario para la emulación de determinado instrumento, usando una Computadora (PC) como interface a la que previamente se le ha instalado el software adecuado para manipular el hardware y el instrumento asociado, de esta forma, esta caja negra podría comportarse como un generador de ondas arbitrario, un osciloscopio digital, un analizador de espectro, un sistema de adquisición y almacenamiento de datos, filtros digitales, sistema de medición y control genérico, etc. (ver Fig. 1)



**Fig. 1** Concepto del RVI – fuente: *Introducción al proyecto RVI* – <http://mlab.ictp.it/rvi/relevance.html>

Para asegurar la flexibilidad del hardware se le dotó como componente principal una FPGA (Field Programmable Gate Array), este dispositivo es un enorme arreglo de bloques lógicos los que, al ser correctamente interconectados mediante programación, pueden generar complejos sistemas digitales de alta performance [2].

Si bien la inclusión de un microprocesador o microcontrolador hubiese hecho el sistema mucho más sencillo de programar, la decisión de incluir una FPGA en su reemplazo asegura un alto nivel de personalización, permitiendo desarrollar rutinas que al estar directamente implementadas en hardware mediante la interconexión de bloques lógicos superan ampliamente en performance a su

correspondiente en software de microcontrolador, así mismo una FPGA posee inherentemente la capacidad de desarrollar distintas tareas en simultáneo (paralelismo) y en tiempo real, algo casi imposible de obtener con un solo microcontrolador; por ejemplo puede implementarse filtros digitales para distintos canales analógicos que funcionen en paralelo, muestreo en simultáneo (no secuencial) de varias señales analógicas y digitales.

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 entonces definir sistemas tan simples como funciones lógicas unidas a una máquina secuencial, o tan complejos como el de múltiples microprocesadores trabajando en paralelo interconectados mediante buses personalizados y unidos al exterior por interfaces hechas a medida, todo dentro de una única FPGA. De lo último es fácil deducir que la principal limitación para elaborar un complejo diseño digital la constituye la cantidad de celdas lógicas disponibles en la FPGA.

## Hardware de la RVI

Para el diseño del hardware de la RVI los desarrolladores utilizaron modernas técnicas de elaboración de circuitos, pensando principalmente en un diseño versátil y adaptable que contara desde el inicio con adecuadas capacidades de interconexión y de adquisición/generación de señales. (Ver Fig. 2). El componente más importante en el Hardware de la RVI es la FPGA, ésta define el nivel de complejidad de los instrumentos que podrían ser emulados.

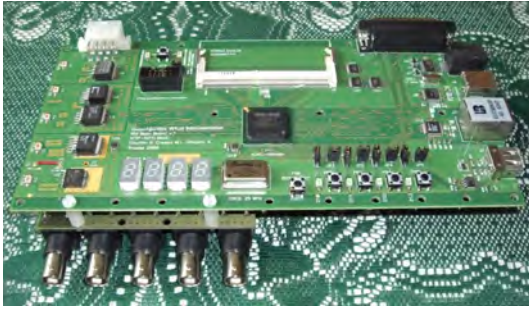


Fig. 2 Primer plano de la tarjeta RVI

Gracias al nivel de colaboración existente entre la empresa Actel [3] y el MLAB se optó por la familia ProASIC3E montándose los primeros prototipos con la FPGA A3PE 600 y luego con la A3PE 1500, FPGA's con 600K y 1.5M compuertas lógicas equivalentes respectivamente, con programación no volátil (basada en memoria Flash), estructura I/O avanzada, bloques de memoria SRAM, FIFO y PLL integrados, etc.

En el área de comunicación, la tarjeta RVI posee un rico abanico de posibilidades, desde una puerta paralela con conexión directa a varios de los puertos I/O de la FPGA hasta conectores de red Ethernet y USB (Ver Fig. 3), apoyados por pequeños circuitos integrados de interface que sólo brindan una funcionalidad básica dependiendo de la FPGA para la labor de gestión y elaboración de los paquetes de datos en sí.



Fig. 3 Conexiones disponibles en la RVI

En cuanto a capacidades de expansión, la RVI posee un sócalo para la colocación de un módulo de memoria SDRAM SODIMM que permitiría almacenar información a una velocidad mayor a lo 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

las señales de alimentación. Para estos conectores existen actualmente dos tarjetas adicionales (daughter boards) que añaden la capacidad de adquirir y generar señales analógicas:

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

Aunque se le ha denominado «Low Performance» a la primera tarjeta (LPDB), sus características son más que suficiente para muchas aplicaciones, ver Fig. 4. La RVI también cuenta con 5 botones y 4 displays de LED de 7 segmentos pensados inicialmente para apoyar al debuggin.

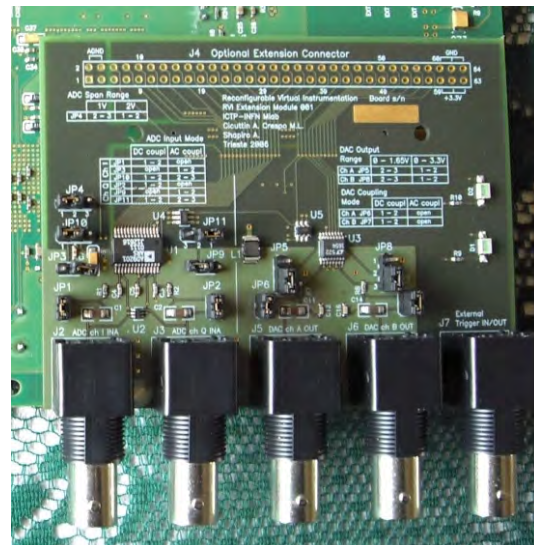
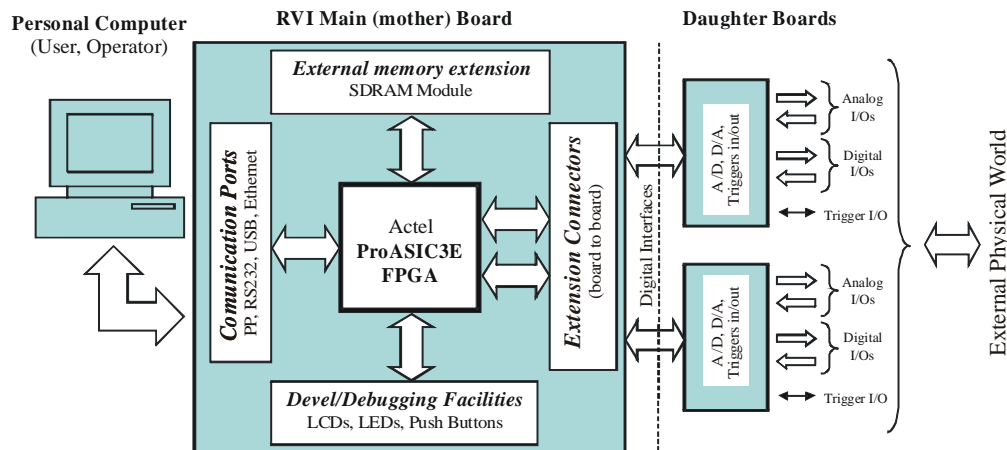


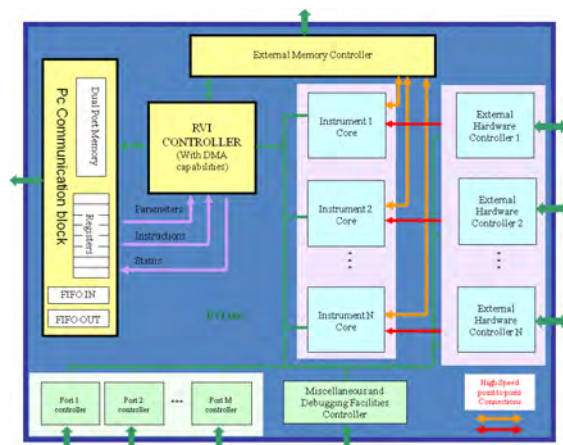
Fig. 4 Detalle de la LPDB

En cuanto al modo de programación, la RVI posee un interface JTAG compatible con los programadores USB de Actel y también la capacidad de ser programada vía el interface paralelo. En la Fig. 5 puede apreciarse un resumen de los bloques que constituyen la arquitectura hardware de la RVI, para una revisión más detallada se incluyen como anexos los planos esquemáticos de la tarjeta principal y de la LPDB.



**Fig. 5** Diagrama de bloques del hardware de la RVI – fuente: RVI System Architecture <http://mlab.ictp.it/rvi/system.html>

Para usar cada uno de los bloques de hardware se necesita construir el código que permita la configuración de la FPGA de manera que esta establezca las conexiones y mecanismos de interacción adecuados que determinen la funcionalidad del instrumento final. Este código que consiste en una serie de instrucciones en un lenguaje HDL sintetizable es tan importante como el hardware que controla y su correcta implementación definirá la capacidad del instrumento final, otro aspecto igual de importante lo constituye el software que se encargará del interface entre el usuario final y el equipo y del cual dependerá la usabilidad del mismo.



**Fig. 6** Bloques funcionales del SDHC en la arquitectura RVI – fuente: A. Cicuttin, Reconfigurable Virtual Instrumentation Platform.

## Arquitectura del Software de la RVI

En la arquitectura original propuesta por los desarrolladores de la RVI [4], se establece la posible implementación del código sintetizable para la descripción del hardware (SHDC) en la FPGA, que básicamente consiste en código VHDL, separándolo en diferentes secciones funcionales (ver Fig. 6): módulo de comunicaciones, controlador RVI, núcleos de instrumentos, controladores de hardware y los archivos de asignación de puertos en la FPGA, adicionalmente se sugieren módulos para la depuración, etc.; construyendo en su conjunto un poderoso y complejo sistema que podría llegar a implementar en simultáneo diferentes instrumentos que serían seleccionados y configurados mediante el software que sirve de interface en la PC.

El controlador RVI sería el módulo definiría las ubicaciones de cada bloque en un mapa de direcciones y también se encargaría de regular y dirigir el flujo de datos entre los módulos internos y la gestión con el interface hacia la PC vía el módulo de comunicaciones; para ello establecería múltiples enlaces o canales DMA según una serie de instrucciones y parámetros recibidos del software en la PC, asegurando así la más alta velocidad en la transferencia de datos entre las secciones que constituyen una determinada funcionalidad o instrumento. El SDHC establecería interfaces normalizadas de comunicación entre los distintos módulos y mecanismos comunes de interacción, con la finalidad de que cada uno pueda ser mejorado o actualizado

independientemente y que a su vez pueda ser reutilizado en diferentes contextos. Como Bus de interface entre los bloques funcionales se ha elegido la especificación Wishbone propuesta por OpenCores [5] por ser una implementación libre y abierta. La modularidad permitiría el desarrollo independiente a partir de la necesidad de los grupos de desarrollo, enriqueciendo la plataforma. A nivel del software en la PC la propuesta del grupo de desarrollo de la RVI consiste en la implementación de una serie de rutinas que leen y escriben contenido en varios registros o en la memoria de doble puerto, implementados en el bloque de comunicaciones del SDHC, agrupando estas rutinas en instrucciones más complejas que puedan ser interpretadas por el SDHC y el software controlador que finalmente servirá de interface entre la aplicación de alto nivel y el hardware.

Actualmente siguiendo este esquema se han elaborado el bloque de comunicación en la FPGA y las librerías de comunicación en la PC, que han permitido una validación inicial de la arquitectura, aunque el grueso de la misma aun esta por implementarse.

Alternativamente al sistema SDHC expuesto se ha trabajado en un sistema simplificado que permite la operación de la tarjeta RVI y su interacción con software para PC. Este sistema al que se le ha denominado SBA por sus siglas en inglés (Simple Bus Architecture) consta de un controlador único que arbitra un conjunto de buses de comunicación y direccionamiento y aunque cumple funciones similares a un microcontrolador en realidad no es más que una máquina de estado especializada que permite el flujo de información entre bloques más complejos del sistema, de esta manera los recursos de la FPGA se pueden destinar en su mayoría a la creación de bloques altamente especializados. El SBA también especifica librerías de comunicación que pueden utilizarse para interconectar software escrito para PC en una estructura de capas, de esta forma la interface de hardware que se use, ya sea RS232, paralelo (SPP, EPP, ECP), USB, Ethernet, etc. queda encapsulada por la librería y esta brinda funciones estándares de comunicación hacia la arquitectura RVI.

## Ejemplos de Instrumentos desarrollados en la arquitectura RVI



Fig. 7 Adquisición usando un módulo USB FT245



Fig. 8 Adquisición usando el puerto paralelo en modalidad EPP

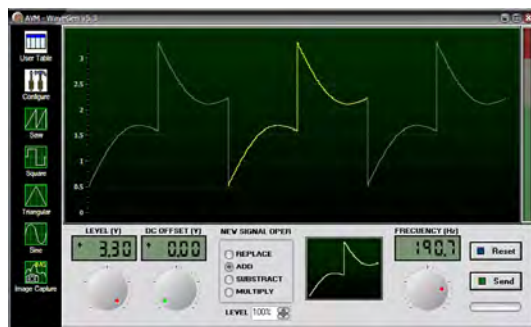


Fig. 9 Generador arbitrario de ondas y señales

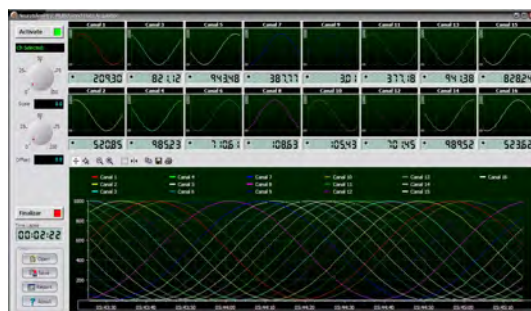
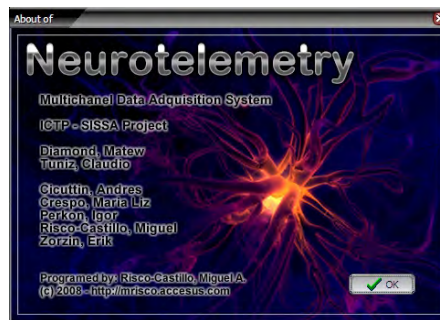


Fig. 10 Prototipo de software de adquisición de datos multicanal del Proyecto Neurotelemetry

## CONCLUSIONES

Si bien en términos de hardware, la tarjeta RVI presenta posibilidades de uso realmente interesantes, estos no son suficientes para poder hacer uso inmediato de la misma, se necesita igualmente del software tanto a nivel de configuración de la FPGA como del interface hacia la PC.

La arquitectura SBA presenta una alternativa a corto y mediano plazo para el uso de la RVI aunque el desarrollo del sistema original SDHC propuesto por el MLAB brindaría a ésta de la potencia y capacidades de reconfiguración máximas.

La RVI ha probado ser una herramienta efectiva, eficiente y versátil con la que se pueden desarrollar trabajos serios de investigación.

Para que lograr que la RVI brinde todo su potencial se necesita de una comunicad que colabore con la construcción de módulos especializados siguiendo una norma y metodología única que permita que su trabajo sea fácilmente integrable y reutilizable.

## AGRADECIMIENTOS

Al laboratorio de Microprocesadores del Centro Internacional de Física Teórica (ICTP) de Trieste, Italia; por la oportunidad de trabajar junto a ellos y colaborar con el desarrollo de la RVI.

A los diseñadores de la RVI:

A. Cicuttin  
M.L. Crespo  
A. Shapiro

Por su excelente y duro trabajo.

A la Escuela Internacional de Estudios Avanzados (SISSA) que con sus proyectos permitieron validar a la RVI como un instrumento efectivo y versátil.

Al Centro de Investigación y Desarrollo en Ingeniería de la Facultad de Electrónica y Mecatrónica de la Universidad Tecnológica del Perú, que me permiten continuar con el desarrollo de la arquitectura RVI y difundirla entre sus estudiantes.

A mi familia por toda su gran paciencia y apoyo durante los meses lejos de casa.

## REFERENCIAS BIBLIOGRÁFICAS

- [1]. N. Abdallah, A. Cicuttin, M. L. Crespo, A. Shapiro. A Block-Based Open Source Approach for a Reconfigurable Virtual Instrumentation Platform Using FPGA Technology. San Luis Potosi : Reconfigurable Computing and FPGA's, 2006. ReConFig 2006. IEEE International Conference on, 2006.
- [2]. Definición FPGA, Wikipedia en español. [En línea] <http://es.wikipedia.org/wiki/FPGA>.
- [3]. Actel (r), portal web. [En línea] <http://www.actel.com>
- [4]. RVI System Architecture. [En línea] <http://mlab.ictp.it/rvi/system.html>
- [5]. SoC Interconnection: Wishbone. [En línea] <http://www.opencores.org/?do=wishbone>.

**E-mail:** [mrisko\\_c@ictp.it](mailto:mrisko_c@ictp.it); [mrisko@accesus.com](mailto:mrisko@accesus.com)