

# Instrumentación virtual en tiempo real con FPGA: Analizador de Espectros

Emanuel Trabes; Carlos Sosa Páez; Víctor Yelpe; Diego Costa (\*)

Laboratorio de Electrónica, Investigación y Servicios  
Facultad de Ciencias Físico, Matemáticas y Naturales / Universidad Nacional de San Luis  
San Luis, Argentina  
e-mail: {3024306, sosapaez, vayelpe, dec}@unsl.edu.ar

**Resumen**—La instrumentación virtual reconfigurable es un campo de aplicación de la lógica programable en el que el usuario y el desarrollador pueden definir la funcionalidad del sistema por software y modificar las características del equipo reprogramando el hardware. Esto brinda la posibilidad de contar con instrumentos “a medida”, con altas prestaciones y relativo bajo costo. En el trabajo se realizó un analizador de espectros digital en tiempo real en FPGA descrito en VHDL, con una interfaz USB para conexión a una PC con presentación gráfica de los resultados y control de los parámetros desarrollados con software libre y código abierto. La integración de estos tres elementos se hizo sobre una plataforma de desarrollo para otras aplicaciones en la temática.

**Palabras clave:** Instrumentación virtual reconfigurable, tiempo real, analizador de espectros, FPGA, VHDL, software libre, código abierto.

## I. INTRODUCCIÓN.

En un instrumento tradicional, el elemento que determina la funcionalidad es el *hardware* definido por el fabricante. En un instrumento virtual, es el *software* el que determina las características, bajo el control del usuario, con las limitaciones impuestas por el fabricante. En un instrumento virtual reconfigurable, el uso de la lógica programable es el elemento que brinda la reconfigurabilidad, ya que permite que el usuario y particularmente el desarrollador determinen el *hardware* y el *software*, y por ende, definan la funcionalidad permitiendo ampliar características, agregar funciones, brindar fácil accesibilidad al uso de datos para su posterior almacenamiento y procesamiento, incorporar más instrumentos con el mismo circuito e integrarlos a sistemas más complejos.

En la Figura 1 se muestra un esquema de un instrumento virtual reconfigurable [1] para medición o generación. Cuenta con un *hardware* para la etapa de adquisición (sensores y actuadores, MUX y DEMUX, acondicionamiento, y conversión A/D y D/A) pero la etapa de procesamiento (cálculo, temporización, memoria, etc.) y la etapa de presentación y ajuste (pantallas, indicadores, potenciómetros, llaves, etc.), se realizan combinadamente en *software* y en *hardware*. La articulación de estos elementos requiere de una interfaz externa estándar para conexión del circuito con la PC que corre el programa (paralelo, RS-232, Ethernet o USB) y de interfaces internas para la vinculación entre los módulos del *hardware* (CoreConnect, AMBA y crecientemente WISHBONE).

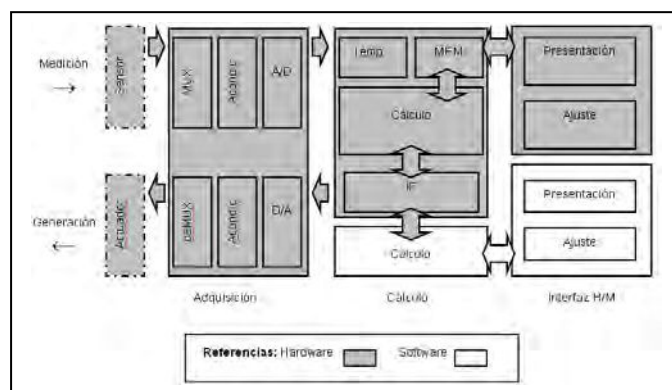


Figura 1. Diagrama en bloques de un instrumento virtual reconfigurable.

En este trabajo se desarrolló un analizador de espectros en tiempo real completo, sobre una plataforma de desarrollo para instrumentación virtual reconfigurable [2] con sus bibliotecas *software* y *hardware*. El circuito se describió con diseño modular [3], bajo el estándar WISHBONE [4] de conexión interna y con interfaz externa USB hacia la PC. La interfaz gráfica se realizó en Qt 4.5, software que posee licencia LGPL.

## II. HARDWARE.

### A. Dispositivos utilizados.

Se utilizó una placa Fusion Embedded Development que contiene un FPGA M1AFS1500 con encapsulado 484FBGA de Actel Microsemi [5] con su placa de programación, circuito de configuración, indicadores y pulsadores para control y depuración, puertos de comunicación USB y Ethernet 10/100. La misma se muestra en la Figura 2. Para la adquisición de señales, a la placa mencionada se le acopló la LP Data Conversion Daughter Board desarrollada por el ICTP (\*\*). La misma cuenta con un convertor A/D (AD9201) y un convertor D/A (LTC1654), cada uno con conectores y selector de rango de convertor, además de conectores para incorporar a la placa principal y sensores de temperatura.

### B. Metodología de diseño y herramientas utilizadas.

La descripción de los circuitos se hizo generando los códigos en VHDL [6] [7] usando diferentes técnicas. Se utilizó un bloque Analizador [8] generado mediante herramienta (\*\*\*) de conversión de bloques esquemáticos de modelado y simulación a código HDL, al cual se le realizaron

(\*) Pertenecientes al Proyecto “Instrumentación Virtual Reconfigurable” financiado por Ciencia y Técnica de la UNSL.

(\*\*) Esta placa ha sido provista por el ICTP (International Center For Theoretical Physics) y fue desarrollada por A. Cicuttin, M. L. Crespo y A. Shapiro.

(\*\*\*) Las herramientas de *software* fueron recibidas como premio Designer Forum Mention del III Southern Conference on Programmable Logic.

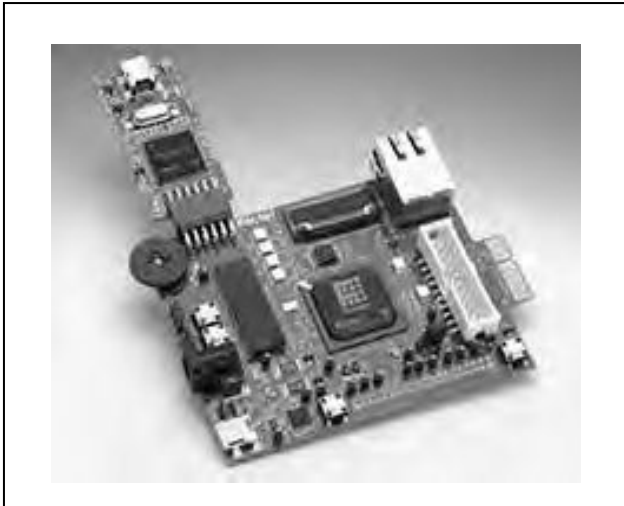


Figura 2. Placa de desarrollo Fusion de Actel con programador.

múltiples optimizaciones para reducir el número de celdas requeridas, según se muestra en Figura 3. El bloque de reloj y la interfaz serie con la PC se realizaron con un generador de *cores* que implementa los circuitos mediante librerías de diseños ya realizados, parametrizados, y optimizados para el FPGA elegido mediante la invocación de primitivas. El resto de los módulos de adquisición, temporización, reinicio y la lógica que enlaza todas las partes se realizaron describiendo directamente el código en VHDL.

En el presente trabajo, para depuración se utilizó el Libero 9.1 (que posee la herramienta Smart Gen para generar *cores*), Synplify DSP AE para la síntesis, Designer para posicionamiento-ruteo, Flash Pro para programación y ModelSim para simulación pre-síntesis, post-síntesis y *post-layout* [9]. Para la conversión de modelo de simulación a código HDL se usó Synplify DSP [10].

### C. Diseño en el FPGA.

El diseño programado en el FPGA consta de diferentes partes.

Un bloque generador de reloj con un PLL provee una señal de 20MHz a partir de un oscilador a cristal de 50 MHz

ubicado en la placa.

El convertor A/D es controlado por un bloque ADC que adecua el formato de datos a 8 bits de ancho de palabra para el resto de los bloques.

El bloque que realiza el cálculo de la Transformada Discreta de Fourier (DFT) posee un puerto de entrada con registros, seguido de un generador de ventana LUT con una secuencia de 1024 puntos de una función de Blackman que se multiplica con la señal de entrada para minimizar el efecto de Gibbs. La secuencia enventanada es procesada en la sección central que implementa el algoritmo de la Transformada Rápida de Fourier (FFT) [11], tiene una latencia de 2061 ciclos. Al resultado en forma directa con sus partes real e imaginaria (en tramas de 1024 muestras en dos vectores de 13 bits con formato entero y con signo) se le calcula el valor absoluto mediante un CORDIC (COordinate Rotation DIGital Computer) eficiente que usa sumas, restas, desplazamientos y LUTs. Para utilizar los valores en todo el rango del espectro, se permuta el orden de la segunda mitad de las muestras de la salida de la sección anterior. El puerto de salida de 14 bits en formato entero y sin signo registra los datos. Para la sincronización con el bloque de la interfaz con la PC, se genera una señal con forma de onda rectangular cuyo período es de 512 muestras, estableciendo un mecanismo de disparo automático.

La interfaz con la PC se realizó a través de un puerto USB manejado por un controlador que contiene la placa de desarrollo Fusion. Si bien la comunicación física se realiza por medio de ese bus, el controlador permite que los datos presentados para la transmisión, y los datos leídos desde la recepción, se presenten en formato RS-232. Esto facilita el diseño dado que para el presente trabajo, sólo fue necesario describir un controlador de puerto RS-232 que fue implementado con el generador de *cores*, dado que el *hardware* se encarga de la conversión de formatos y el *software* provee los drivers necesarios para un funcionamiento armónico del sistema. La desventaja de este puente de buses reside en la reducción de la velocidad de transmisión, que para el presente desarrollo no constituyen un problema dado que resultaría inútil tratar de presentar más de 50 frames por segundo, tasa mucho menor a la que se alcanza a pesar de la limitante mencionada.

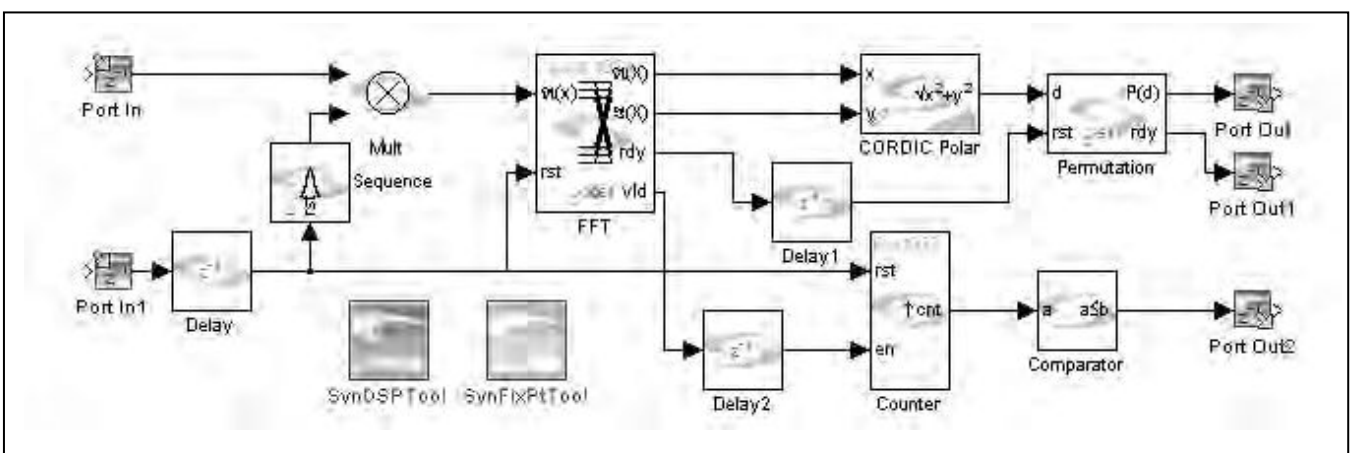


Figura 3. Esquemáticos del bloque Analizador en el ambiente de modelado y simulación.

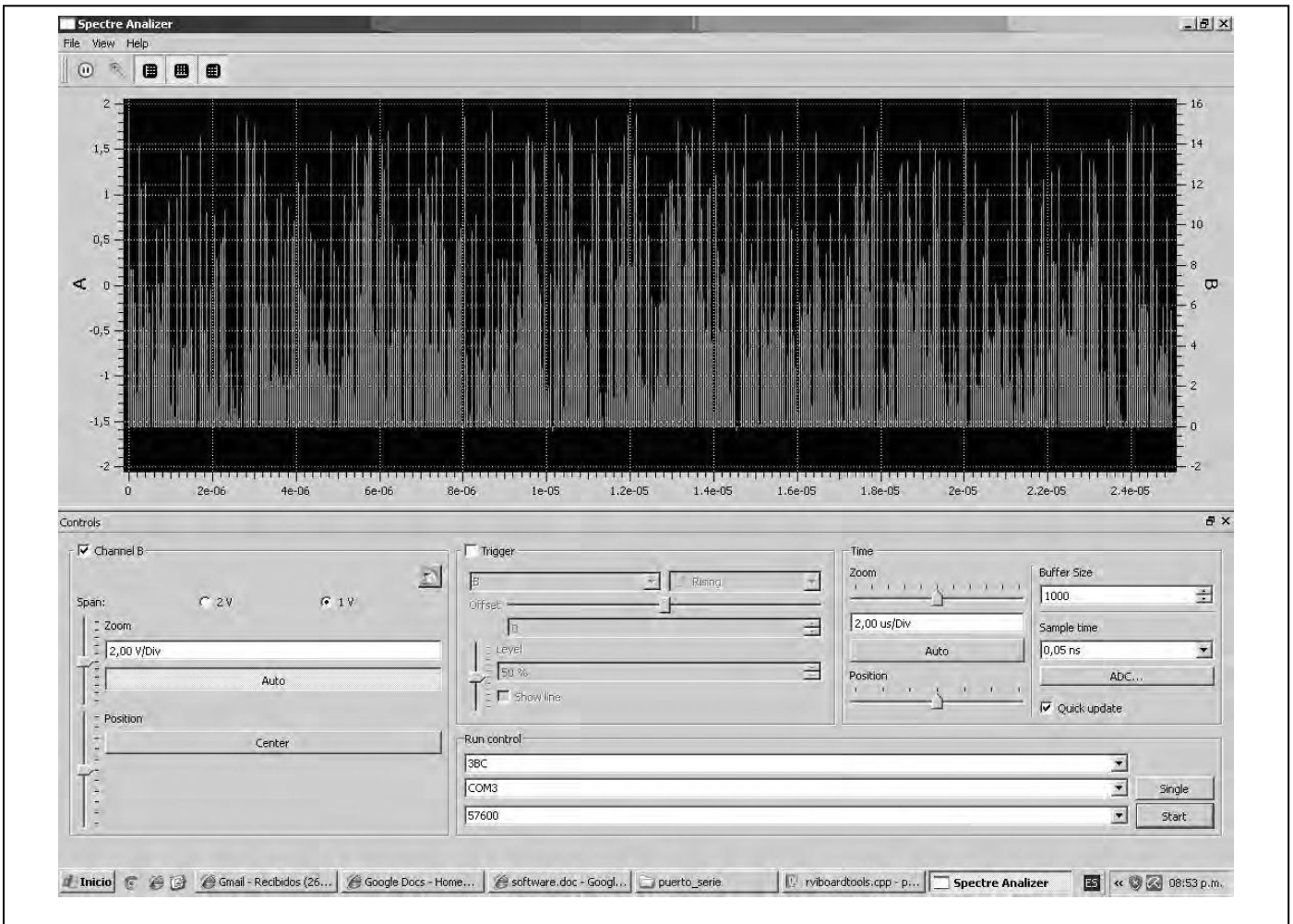


Figura 4. Interfaz gráfica del software desarrollado para la presentación de resultados y ajuste de parámetros.

### III. SOFTWARE.

Se diseñó una interfaz gráfica en Qt, simple e intuitiva que permite operar el instrumento de manera similar a uno tradicional. Para el desarrollo del *software*, se han utilizado las bibliotecas Qt y el IDE Qt Creator. Las bibliotecas Qt están disponibles para varias plataformas (Windows, MAC OS, Linux y Unix) y el código está íntegramente escrito en un lenguaje portable (C++) y estructurado adecuadamente para que sea fácil su lectura, realización y traspaso a otra plataforma.

En la Figura 4 se muestra la interfaz gráfica del *software* compilada para una plataforma Windows. La misma se configuró para utilizar un canal, pudiéndose agregar otro sin dificultad. Se definieron controles de atenuación y posición horizontal y vertical con escala ajustable. Si bien el disparo es realizado automáticamente por el *hardware* que señala el inicio de la trama de datos para presentar en pantalla, se pueden programar otras modalidades de disparo.

El programa está estructurado de manera que existen dos *threads*:

- La *thread* principal, que básicamente se encarga de dibujar los datos en pantalla, hacer la interfaz de usuario, etc.
- La *thread* de comunicación, que se encarga de tomar los datos por el puerto serie y ponerlos adecuadamente en memoria para luego ser dibujados por pantalla.

La placa Fusion que se utiliza, no solamente provee el *hardware* y los IP *cores* adecuados para poder implementar una comunicación equivalente a la de un puerto serie normal pero a través de un USB, sino que también dispone de los drivers necesarios para crear un puerto serie virtual del lado de la PC, de manera que, tanto en el *hardware* como en el *software*, la comunicación es tratada como si se estuviera realizando mediante un puerto serie.

Para la versión Windows, se utilizan las bibliotecas de comunicación serie que ya están embebidas en el sistema operativo, por lo que no se necesita librería externa alguna.

La librería utilizada permite dos opciones para la comunicación serie:

- *Non overlapped io*, en la cual la función espera el dato hasta que este disponible para ser utilizado, y

- *overlapped io*, que es más sofisticada, y se basa en preguntar cada determinado tiempo si ha llegado un dato que luego es leído.

Debido a que se utiliza una *thread* aparte para la comunicación y dado que la forma *overlapped* de la función no es compatible con muchos sistemas operativos, se optó por la forma *non-overlapped*.

El programa en sí, consta de cuatro capas, implementadas cada una en una clase para su fácil lectura.

- Capa física, en la cual se implementa la forma en la que el programa lee los datos del puerto. Esta sería la capa a modificar si se quisiera cambiar la forma de interconexión física.
- Capa de comunicación, en la que se toman los datos y se los arregla en tramas para luego ser mostrados en pantalla como un conjunto.
- Capa de placa, en la que se usan las utilidades que posee Qt para dibujar los datos e hacer la GUI con la cual se comunica el usuario con el programa.
- Capa principal, que engloba a todas las demás.

#### IV. RESULTADOS.

En el presente trabajo se logró implementar un instrumento virtual reconfigurable para análisis espectral completo, con altas prestaciones y bajo costo, sobre una plataforma de desarrollo que integra todas las etapas desde la adquisición, el procesamiento de cálculo y la presentación del resultado. La metodología de diseño y el tipo de dispositivo utilizado permite integrar nuevas funciones al presente y nuevos instrumentos para conformar un sistema de medición. Esto, se ve favorecido porque el *software* usado para la interfaz gráfica es de código abierto, lo que facilita el trabajo cooperativo para futuras ampliaciones.

El rango dinámico se definió entre -0,5 V a 0,5 V aunque es ajustable según la configuración de DIP switches dispuestos en la LP Data Conversion Daughter Board.

La resolución en amplitud en la entrada está dada por el ancho de palabra de 8 bits que optimiza el diseño. La resolución interna se limitó a la mejor precisión posible de obtener con los recursos que insume el bloque que calcula la FFT. Una resolución mayor insumiría más celdas lógicas que las disponibles. Con esto, la salida se fijó en 14 bits.

La resolución en frecuencia es de 512 muestras, limitada por la máxima precisión posible que se le pudo configurar al bloque que calcula la FFT capaz de ser sintetizado en el FPGA disponible. Una FFT con más de 512 puntos requeriría más recursos de los disponibles y una FFT de 256 puntos implicaría una degradación notable de la precisión para la transformada calculada.

El ancho de banda del instrumento, está limitado por la máxima frecuencia a la que puede funcionar el bloque Analizador que, según el reporte temporal, es de 128,75 Mhz, por lo que se puede lograr un ancho de banda de 64,375 MHz.

Para verificar el funcionamiento de la interfaz con la PC a diferentes frecuencias, se tomó una señal analógica de

salida conformada por las tramas del espectro para su visualización en una pantalla de osciloscopio a fin de contrastarla con la presentación en el monitor de la PC. El conversor D/A que posee la LP Data Conversion Daughter Board es un conversor serie de 25 bits de palabra con 14 bits de carga útil. Usando un reloj de 20 MHz se logró una tasa de reconstrucción de 800 kHz para los 25 bits por lo que fue posible verificar la interfaz con señales hasta los 400 kHz. Para la verificación con señales de frecuencia hasta 15 MHz se utilizó un generador de formas de onda arbitrarias Hewlett Packard 33120A.

En la Tabla I se detalla el reporte de recursos. En la Tabla II se consigna el informe de tiempos. Cabe señalar que el retardo entrada-salida oscila entre 3,111 y 6,791 ns. En la Tabla III se muestra el reporte de consumo.

TABLA 1. REPORTE DE RECURSOS UTILIZADOS

Recurso	Utilizados	Disponibles	Porcentaje
CORE cells	30030	38400	(78.20%)
IO(W/clocks)	21	223	(9.42%)
GLOBAL(Chip+Quadrant)	6	18	(33.33%)
PLL	1	2	(50.00%)
RAM/FIFO	40	60	(66.67%)

TABLA 2. REPORTE DE TEMPORIZACIÓN

Nombre	Período (ns)	Frecuencia (MHz)	Máx Clock Out (ns)	Min Clock Out (ns)
Ext Clk	4	250	8,111	4,133
U0/U1/Core: GLA	7,764	128,750	17,503	3,490
U2/present_state[1]:Q			15,906	5,420
U4/Clk_0:Q	57,015	17,539	9,733	3,864

TABLA 3. CONSUMO DE POTENCIA

	Potencia [mW]	Porcentaje
Potencia dinámica	0,496	2,5%
Potencia estática	19,320	97,5%
Potencia total	19,816	100%

La interfaz HM para presentación de resultados y ajustes de control del instrumento realizada con Qt permitió resultados óptimos, destacándose la metodología de desarrollo cómoda provista por el ambiente de diseño IDE Qt Creator, su uso versátil e intuitivo, la flexibilidad y la fácil validación, etc. Si bien existe buena documentación y la misma es abundante, existieron algunos problemas a la hora de depurar el diseño y no se encontraron todas las soluciones con la facilidad esperada. Sin embargo, las ventajas obtenidas superan ampliamente los problemas presentados, máxime si se tiene en cuenta el costo que poseen otras herramientas similares de uso comercial.

REFERENCIAS

- [1] A. Manuel, D. Biel, J. Olivé, J. Prat. F. J. Sánchez, "Instrumentació virtual. Adquisició, processament i anàlisi de senyals", Ediciones UPC, junio 2001, ISBN 84-8301-473-4
- [2] F. Aguilera, C. Sosa Páez, D. Costa, "Implementación de un osciloscopio en una plataforma de instrumentación virtual reconfigurable", Congreso de Microelectrónica Aplicada 2010, Tercer Milenio. ISBN 978-978-9374-65-8, pp. 110 a 112.
- [3] A. Cicuttin, M. L. Crespo, A. Shapiro y N. Abdallah, "A block-based open source approach for a reconfigurable virtual implementation platform using FPGA technology", IEEE International Conference on Reconfigurable Computing and FPGA's ReConFig 2006, 2006, pp 1 a 8.
- [4] Specification for the WISHBONE System-on-Chip (SoC), Interconnection Architecture for Portable IP Cores; Revision: B.3, Released: September 7, 2002
- [5] Documentos técnicos de Actel Corporation, <http://www.actel.com>
- [6] K. Skahill, "VHDL for programmable logic", Addison-Wesley, 1996.
- [7] J. Bhasker, "VHDL primer", 3th edition, Prentice Hall, 1999.
- [8] V. Yelpo, D. Costa, C. Sosa Páez, "Módulo de cálculo de la Transformada Rápida de Fourier para analizador de espectros en tiempo real en FPGA", Congreso de Microelectrónica Aplicada 2010, Tercer Milenio. ISBN 978-978-9374-65-8, pp. 23 a 28.
- [9] Manuales de uso del Libero, Designer, Flash Pro y ModelSim.
- [10] Synplify® DSP User Guide
- [11] J. G. Pproakis y D. G. Manolakis, "Tratamiento digital de señales", 3ra ed., Prentice-Hall, 2003, pp 457 a 507.