UNA NUEVA METODOLOGÍA EN EL DISEÑO DE FILTROS DIGITALES FIR SOBRE FPGA

A NEW METHODOLOGY IN THE DESIGN OF DIGITAL FILTERS FIR ON FPGA

César A. Hernández S.[1]
Edwar Jacinto G.[2]

Recibido: Marzo 2009
Aprobado: Junio 2009

Resumen

Este artículo presenta una metodología diferente y sencilla en el diseño e implementación de filtros digitales con respuesta de impulso finita, sobre una arquitectura reconfigurable como la FPGA XC3S200. En este desarrollo se emplearon herramientas CAD como FIR Toolbox e ISE de Xilinx y se obtuvieron resultados en tiempo récord, optimización en cuanto a los recursos de hardware utilizados y un buen desempeño en frecuencia, características deseables en el desarrollo de sistemas orientados al procesamiento digital de señales.

Palabras clave

Filtros, filtros digitales, FIR, FPGA, VHDL, procesamiento digital de señales.

Abstract

This article presents a different and simple methodology used in design and implementation of digital filters with finite impulse response (FIR), on an architecture reconfigurable as FPGA XC3S200. On this development was used CAD tools such as "FIR Toolbox" and "ISE" from Xilinx, achieving results in record time, optimizing the hardware resources used, and a good performance in frequency. These characteristics are desirable on development of systems oriented to digital signal processing.

Key words

Filters, Digital Filters, FIR, FPGAs, VHDL, Digital Signal Processing.

1 M.Sc. En Ciencias de la Información y las Comunicaciones con énfasis en Teleinformática e Ingeniero Electrónico de la Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia. Investigador del grupo ARMOS. Correo: cahernandezs@udistrital.edu.co

2 Ing. en Control Electrónico e Instrumentación, Universidad Distrital Francisco José de Caldas, Bogotá D.C., Colombia. Investigador del grupo ARMOS. Correo: edwarjg@hotmail.com

1. INTRODUCCIÓN

El procesamiento digital de señales es un área de la electrónica que ha tenido grandes avances en los últimos años debido a los desarrollos tecnológicos del hardware. Hoy en día juega un papel importante en ramas tan diversas como las comunicaciones, el diseño de circuitos, la acústica, el procesamiento de voz y los sistemas de regeneración y distribución de energía. Por esta razón, se ha concebido una arquitectura de filtro digital de alta capacidad y velocidad de cómputo, con la cual se obtiene una mayor precisión en comparación con los circuitos analógicos y los sistemas de procesamiento de señales analógicas [9]. Otras ventajas de los filtros digitales con respecto a sus similares analógicos son su pequeño tamaño, eficiencia y las posibilidades de una rápida reconfiguración en sus labores, al ser soportados por las tecnologías de alta escala de integración [7].

Los procesadores digitales de señales (DSP) son usados para implementar muchas de las aplicaciones de DSP, dentro de las cuales se encuentran los filtros digitales. Sin embargo, aunque los DSP son programables a través de software, la arquitectura del hardware de estos no es flexible, sino fija; esto se evidencia, por ejemplo, en el número fijo de bloques multiplicador-acumulador (MAC) o el ancho del bus de datos. Lo anterior constituye una limitante para los DSP con respecto a arquitecturas reconfigurables con Complex Programmable Logic Device (CPLD) y Field Programmable Gate Array (FPGA) [11]. Los circuitos CPLD-FPGA suministran una solución reconfigurable y eficiente para implementar aplicaciones DSP como los filtros digitales. Estos circuitos pueden alcanzar un más alto rendimiento (*throughput*) y una mayor potencia de procesamiento de datos que los DSP [11].

En este artículo se expone una metodología diferente en el diseño e implementación de filtros digitales con respuesta de impulso finita (FIR) mediante la utilización de herramientas CAD como FIR Toolbox e ISE de Xilinx. La metodología que se plantea pretende reducir considerablemente el tiempo necesario para realizar un filtro digital FIR, además de optimizar los recursos de *hardware* disponibles en la FPGA XC3S200, seleccionada como dispositivo lógico programable para implementar dicha solución.

2. DISEÑO DEL FILTRO FIR

El filtro digital a diseñar es un filtro FIR. Los filtros FIR tienen dos ventajas fundamentales con respecto a los filtros IIR (de respuesta infinita) [6]. La primera es que se pueden diseñar con fase lineal, por lo que no presentan distorsión de fase. Además, son incondicionalmente estables, ya que su función de transferencia tiene solo ceros. Sin embargo, un filtro IIR requiere un menor número de coeficientes que un filtro FIR [6]. Un filtro digital FIR tiene una respuesta al impulso de finita duración, la cual se puede expresar matemáticamente como:

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i) \tag{1}$$

Donde N es el orden del filtro. Esta expresión indica que el valor de salida actual de un filtro FIR es función de la entrada actual y de las N entradas anteriores. En efecto, el sistema se comporta como una ventana. Equivalentemente, un filtro FIR puede ser descrito mediante su función de transferencia como:

$$H(z) = \sum_{i=0}^{N-1} h(i)z^{-i} \tag{2}$$

Es decir:

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3} + ... + b_{N-1} z^{-N+1} \quad (3)$$

Las raíces de este polinomio son los ceros del filtro. Por lo tanto, el problema de diseño de filtros FIR es simplemente determinar los N coeficientes *h(n)* a partir de una especificación de la respuesta en frecuencia deseada [HD(ω)].

En esta metodología el software libre FIR Toolbox, desarrollado por la empresa Mediatronix, es la herramienta encargada de determinar los coeficientes *h(n)* del filtro, a partir de parámetros configurados por el diseñador. Esta herramienta facilita el desarrollo de este proceso, además de proporcionar compatibilidad total con dispositivos lógicos programables.

En las pestañas superiores de la hoja de trabajo se puede elegir el tipo de filtro deseado: pasa bajos, pasa altos, pasa banda o rechaza banda, con sus respectivas frecuencias de corte a menos 3 dB.

A continuación se determina la frecuencia de muestreo, el orden del filtro y la cantidad de bits. Para seleccionar la frecuencia de muestreo el diseñador debe tener en cuenta las características del conversor análogo-digital (ADC) y la aplicación final del filtro. Para la selección del orden del filtro es necesario tener en cuenta que este puede afectar el ancho de banda del filtro. El número de bits determina la precisión de los cálculos matemáticos, por lo cual el diseñador basado en la experiencia seleccionará la cantidad de bits según el rendimiento deseado del filtro (Figura 1).

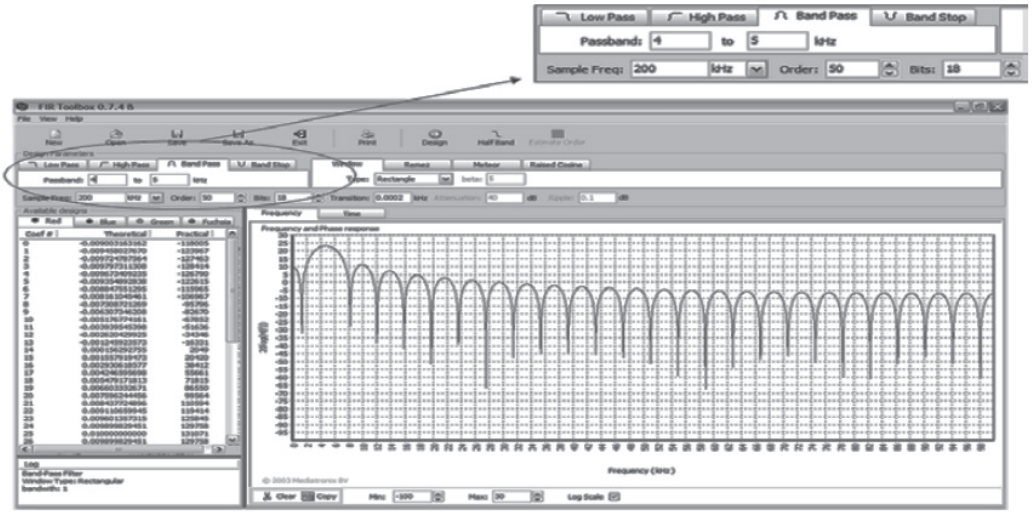

Figura 1. de la frecuencia de muestreo, orden del filtro y número de bits.

A continuación se selecciona el método de diseño a partir del cual se calcularán los coeficientes del filtro FIR (Figura 2).

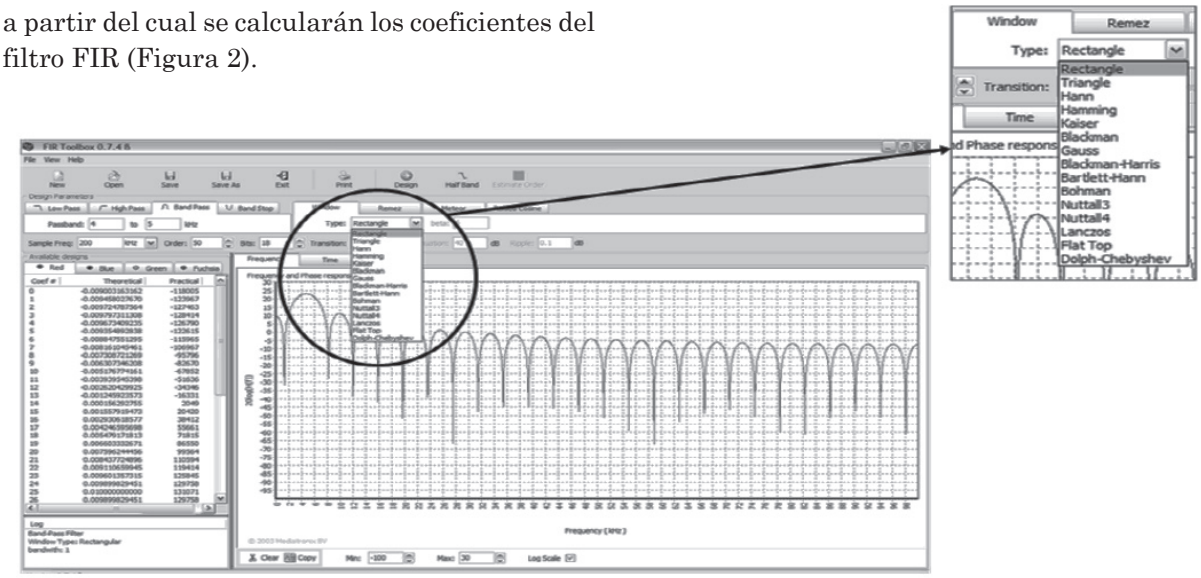

Figura 2. Selección del método de diseño.

Finalmente, al presionar la opción "Design", se obtienen los coeficientes en punto fijo, para lograr una implementación simple que facilite los cálculos dentro del dispositivo lógico programable. Sin embargo, la herramienta proporciona un listado de los coeficientes con su representación en punto flotante y su respectiva representación en punto fijo, cuya transformación depende de la cantidad de bits seleccionados en el procedimiento anterior. (Figura 3).

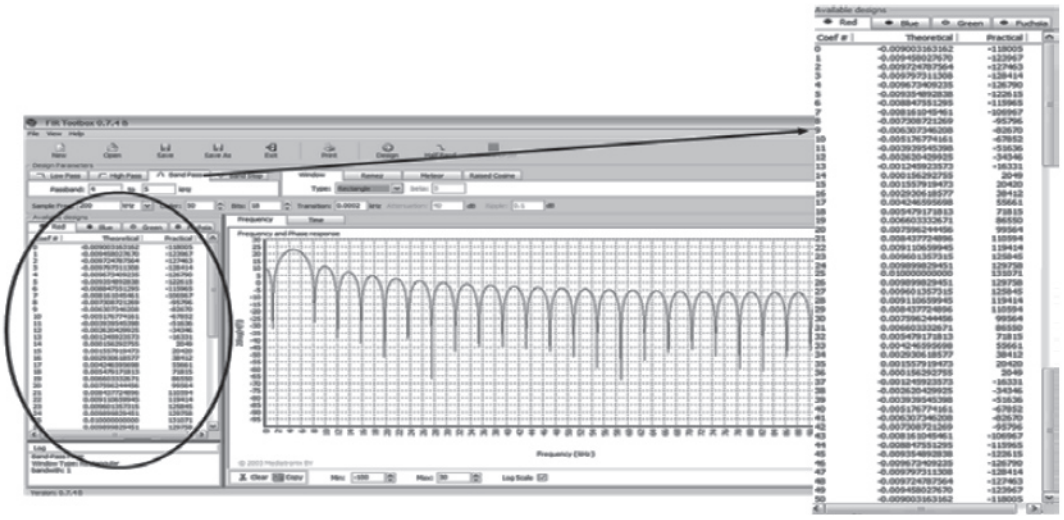

Figura 3. Obtención de los coeficientes del filtro.

La herramienta también proporciona la opción de verificar el comportamiento en tiempo y frecuencia del filtro diseñado, como sus frecuencias de corte y su fase (Figura 4).

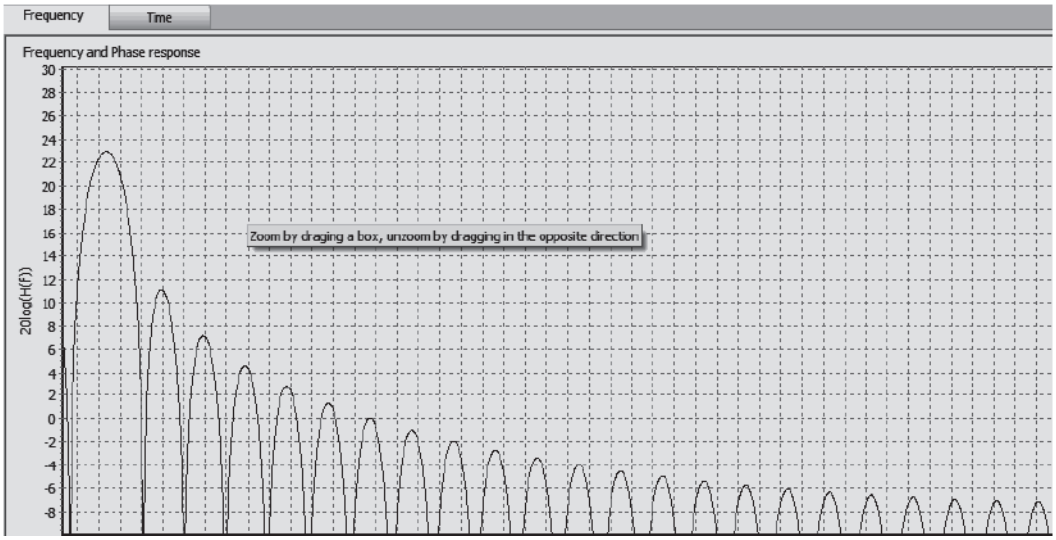

Figura 4. Respuesta en frecuencia del filtro diseñado.

Por último, se exporta el archivo de los coeficientes del filtro diseñado con extensión COE. Este archivo es compatible con la herramienta ISE de Xilinx, lo cual permite una implementación directa y sencilla.

3. REALIZACIÓN DEL FILTRO EN VHDL

Para la implementación del filtro diseñado con la herramienta FIR Toolbox, se seleccionó la FPGA Spartan3 XC3S200 y el software ISE 8.1 de Xilinx. La programación de esta FPGA a partir del archivo de extensión COE, generado anteriormente, fue muy sencilla. En primera instancia, se creó un nuevo proyecto. Una vez creado el proyecto se agregó una nueva fuente, pero esta vez no se seleccionó la opción "VHDL Module", sino la "IP (Coregen & Architecture Wizard)".

La fuente IP (*Intellectual Property*) es un modulo de propiedad intelectual que permite realizar diseños rápidos optimizando los sistemas a medida para aplicaciones específicas, como la que nos ocupa en este caso. Dentro de la gran variedad de módulos que ofrece la fuente IP, se selecciona "Distributed Arithmetic FIR Filter", en otras palabras, un filtro FIR con un hardware paralelo y masivo, el cual utilizará más compuertas que un filtro FIR realizado con una arquitectura basada en MAC (como un DSP), pero con un nivel de procesamiento superior.

Después de esto, se configuran la cantidad de canales del filtro y la interpolación de los datos de entrada del mismo. Según el tipo de conversor análogo digital utilizado como entrada del sistema y su modo de funcionamiento, se eligió el tipo de acondicionamiento de los datos más apropiado.

Luego del paso anterior, según el grado del filtro, se configura la cantidad de elementos de almacenamiento (Taps), el formato de los datos de salida y el ancho de la palabra de trabajo del

filtro, además de la carga del archivo de los coeficientes. Todo lo anterior depende de los parámetros con que se calculó el filtro digital en la herramienta FIR Toolbox.

También es de importancia resaltar que en este punto se da la opción de elegir entre coeficientes fijos o coeficientes reconfigurables, lo cual daría la opción de hacer filtros adaptativos o aplicar estrategias de control, como hardware evolutivo o cualquier otro tipo de implementación que tenga como base una ecuación en diferencias con coeficientes reconfigurables.

Por último, se elige el ancho de los datos de entrada, si vienen del conversor con o sin signo, y si se realizará una implementación, serial o paralela, y si tendrá un *reset* general para reiniciar el sistema en caso de ser necesario, además se da la opción de poner un registro de salida del sistema.

Después de tener el *core* del filtro totalmente configurado y cargado con los coeficientes indicados, se incluye dentro del proyecto creado en el ISE. En este caso se insertó como un bloque en la herramienta grafica de esta *suit*.

4. RESULTADOS

Al realizar la implementación dentro del dispositivo, se muestra un reporte del uso del mismo. En la Figura 5 se puede observar que tan solo se utilizó el 17% de los recursos disponibles en la FPGA Spartan3 de 200 mil compuertas.

FILTRO Project Status			
Project File:	FILTRO.ise	Current State:	Placed and Routed
Module Name:	FILTRO_GRAFICO	• Errors:	No Errors
Target Device:	xc3s200-4ft256	• Warnings:	1 Warning
Product Version:	ISE 8.1i	• Updated:	Tue Sep 19 21:09:03 2006

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	633	3,840	16%	
Number of 4 input LUTs	519	3,840	13%	
Logic Distribution				
Number of occupied Slices	343	1,920	17%	
Number of Slices containing only related logic	343	343	100%	
Number of Slices containing unrelated logic	0	343	0%	
Total Number 4 input LUTs	601	3,840	15%	
Number used as logic	519			
Number used as a route-thru	7			
Number used as Shift registers	75			
Number of bonded IOBs	44	173	25%	
IOB Flip Flops	33			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	14,952			
Additional JTAG gate count for IOBs	2,112			

Performance Summary

Figura 5. Reporte del estado del proyecto finalizado.

Es importante resaltar que el orden del filtro diseñado es 50, que es muy alto y trae como consecuencia el gran número de compuertas utilizadas en la implementación. Este número de compuertas se podría disminuir considerablemente al reducir el orden del filtro.

Otra de las características de este filtro, de gran importancia, es la velocidad de trabajo máxima, que limita la frecuencia de entrada de la señal a filtrar. Se muestra mediante una matriz de retardos, en la que se hace un listado de los máximos tiempos de transporte de las distintas señales en la implementación. En este caso, el máximo retardo es 3,2 nanosegundos, que da una frecuencia máxima de trabajo de aproximadamente 312 megas, lo que divididos por los 16 ciclos que se demora en hacer cada cálculo de la nueva salida del filtro nos da la máxima frecuencia de muestreo, de 19,5 megahertzios, prueba que realizaremos posteriormente.

Después de la implementación, se procede a realizar una simulación del sistema, para lo cual se genera un archivo con cientos de vectores de simulación, que pueden ser creados en Excel o en Matlab con las señales que entran a nuestro sistema. Este archivo se llama Test Bench. Cuando

se trata de simular un sistema digital complejo, se tiene el problema de simular y verificar los resultados del sistema en tiempos grandes, para lo cual en este caso utilizamos una herramienta llamada ModelSim que nos permite hacer simulaciones con grandes ventajas, como por ejemplo un tipo de simulación que convierte los datos digitales en análogos, lo que da una idea de lo que realmente está pasando con el sistema diseñado.

En la Figura 6 se puede observar una primera simulación. En la parte superior se ven las señales de reloj y control de nuestro sistema digital, que en este caso son de tan alta frecuencia que no se pueden visualizar sus cambios; en la parte inferior tenemos debidamente visualizadas (en formato análogo dentro de la simulación) la señal de entrada de nuestro sistema y la de salida, y se puede observar cómo se atenúa la señal de más alta frecuencia y pasa la señal que está dentro de los parámetros de funcionamiento del filtro pasabanda; también observemos el detalle del tiempo de simulación, de casi 7 milisegundos, que para la simulación de un sistema digital configurable es un tiempo considerable.

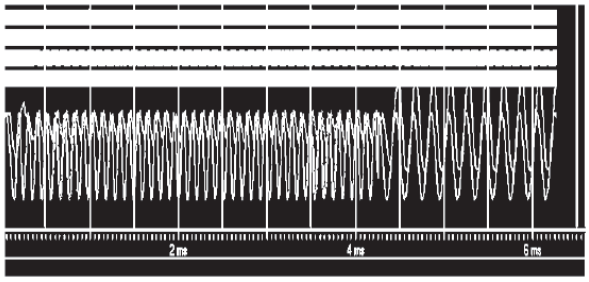

Figura 6. Primera simulación.

En la Figura 7 se observa una simulación mucho más interesante. Al igual que en la anterior, las primeras cuatro señales no nos muestran información importante, debido a su alta frecuencia, pero las últimas dos: la señal de entrada y la señal de salida, muestran un comportamiento interesante. En primer lugar, se inyectó una señal de baja frecuencia, en la cual, al existir cruces por cero, se ve una típica respuesta a impulso dentro de nuestro filtro, la cual se estabiliza en tan solo un semiciclo de oscilación, lo que es una respuesta bastante buena; en el siguiente tramo, se inyecta una señal de alta frecuencia, casi totalmente atenuada, y por último, un período de tiempo en donde se inyecta una frecuencia igual a la resonancia del filtro, que pasa totalmente.

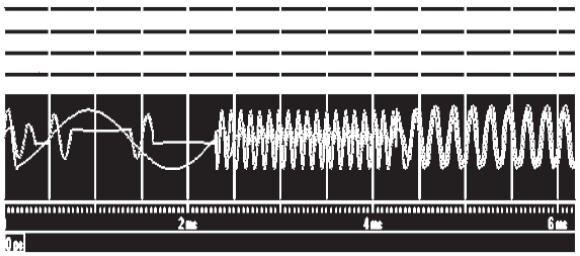

Figura 7. Segunda simulación.

En la Figura 8 se muestra una prueba corriendo el período de muestreo del filtro a una frecuencia mayor, para demostrar que el filtro puede funcionar a frecuencias mucho más altas. A pesar de ser de tan alto grado, la prueba es satisfactoria, aunque está en el límite, ya que se notan pequeñas deformaciones en la señal, al no poder reconstruir con la debida exactitud la señal de entrada. Esta prueba se hizo con una señal de entrada que coincidiera con la nueva frecuencia de resonancia del filtro.

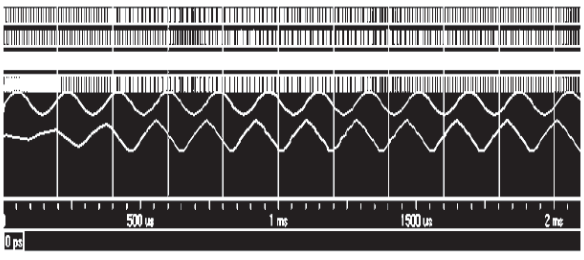

Figura 8. Tercera simulación.

Por último, en la Figura 9 se realizó una simulación que pusiera a prueba realmente al filtro. En la señal de entrada se hizo una multiplicación entre una señal que correspondiera con la frecuencia de resonancia del filtro con ruido blanco (señal pseudoaleatoria), dando como resultado una señal totalmente deformada, pero que implícitamente tiene una frecuencia que hace que el filtro reaccione, tratando de reconstruir la señal original, de frecuencia adecuada para nuestro propósito. Es una prueba bastante significativa, ya que el sistema es capaz de reconstruir señales deformadas por errores sistemáticos y totalmente aleatorios, lo cual es difícil de encontrar en un filtro y mucho menos en uno digital.

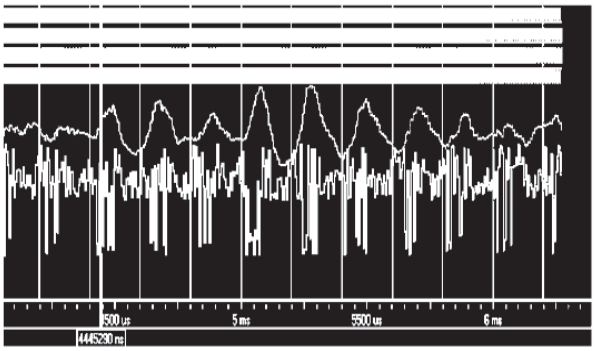

Figura 9. Cuarta simulación.

5. CONCLUSIONES

El desarrollo del filtro digital FIR a través de la metodología descrita en este artículo ha permitido realizar una solución digital satisfactoria en términos de tiempos de respuesta, selectividad del filtrado, optimización los recursos de hardware disponibles en la FPGA seleccionada y en el tiempo requerido para su desarrollo, el cual fue bastante reducido. Los resultados obtenidos en las pruebas del filtro FIR con respecto a tiempos de respuesta son muy satisfactorios, destacando el nivel de procesamiento de las arquitecturas reconfigurables.

REFERENCIAS

[1] Altera Corp. "FPGAs Provide Reconfigurable DSP Solutions". White Paper, August 2002, version 1.0.

[2] Altera Corp. "Implementing FIR Filters in FLEX Devices". Application Note 73, February 1998, version 1.01.

[3] Altera Corp. "Using PLDs for High-Performance DSP Applications". White Paper, February 2002, version 1.0.

[4] G. Proakis and D. Manolakis. "Tratamiento digital de señales", 1 ed. Prentice-Hall. 1998.

[5] K. Ogata. "Sistemas de control en tiempo discreto", 2 ed. Prentice-Hall. 1996.

[6] Eduardo Lahuerta Aguilar, Carlos Medrano Sánchez, y Pedro Ramos Lorente. Implementación de un filtro digital en VHDL. Universidad de Zaragoza.

[7] Miguel Melgarejo y Alexis Pirajan. "Filtro FIR adaptativo sobre celdas lógicas programables FPGA". Revista Ingeniería (2000). Universidad Distrital Francisco José de Caldas, Bogota.

[8] R. Hamming. Digital Filters. Prentice-Hall, 1999.

[9] R. N. Vera. Sistemas de transmisión, 2 ed. McGraw-Hill, 1998.

[10] S. Haykin. Señales y sistemas, 1 ed. Prentice-Hall, 1997.

[11] Mario Vera Lizcano, Gustavo Vejarano y Jaime Velazco Medina. "Diseño de funciones DSP usando VHDL y CPLDs-FPGAs". Ingeniería y Competitividad (2006). Universidad del Valle.

[12] "Distribuited Arithmetic FIR Filter". Xilinx Corp (abril 28 de 2005).