

# Diseño e implementación de un codec digital de audio con FPGA, en formato PCM, de 2 canales con interfaz para usuario

## Design and implementation of a digital audio codec with FPGA, PCM format, 2 channel with user interface

VÍCTOR ALFONSO BRAVO GARZÓN

Tecnólogo en Electrónica. Estudiante de Ingeniería en Control de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. vicalfbg@gmail.com

JESÚS JAIR NAVARRO BAREÑO

Tecnólogo en Electrónica. Estudiante de Ingeniería en Control de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. achus1218@hotmail.com

EDWAR JACINTO G.

Ingeniero en Control. Docente de la Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. edwarjg@hotmail.com

Clasificación del artículo: investigación (Conciencias)

Fecha de recepción: agosto 5 de 2009

Fecha de aceptación: febrero 2 de 2010

**Palabras clave:** Filtro digital, PCM, Picoblaze, Procesamiento, VHDL.

**Key words:** Digital filter, PCM, Picoblaze, Processing, VHDL.

### RESUMEN

En este documento se describe el proceso para la creación de un dispositivo de procesamiento digital con codificación y decodificación en formato PCM, con el desarrollo de filtros digitales y el microprocesador *picoblaze*, mediante la implementación sobre la tarjeta de desarrollo de XILINX “Spartan 3E”. Además, se describe el desarrollo realizado sobre algunos de los módulos que contiene dicha tarjeta y otras etapas de no menor importancia, como el

acondicionamiento de señal, amplificación en entrada y salida, al igual que el acople hacia la tarjeta.

### ABSTRACT

This document describes the process to create a device of digital process with encoding process and decoding in PCM format, with the development of digital filters and the *picoblaze* microprocessor by means of the implementation in Spartan-3E Starter KitBoard from XILINX, also it describes the work

out made about some modules that have this one in order to get the best quality in device, other phases with less value as the signal conditioning, amplifi-

cation in phase input and output, as well as joining together toward the board.

\* \* \*

## 1. Introducción

En Colombia el diseño y desarrollo de nuevas tecnologías aplicadas a la investigación avanza lentamente debido a las pocas aplicaciones que se realizan con base a estas tecnologías; son desarrollos que hasta el momento están comenzando a ser explorados con mayor profundidad. Es por esto que en el grupo de investigación Digiti de la Universidad Distrital Francisco José de Caldas, en la Facultad de Tecnológica, se planteó la elaboración de un dispositivo que sea capaz de procesar señales de audio en dos canales diferentes, teniendo en cuenta todos los pasos por seguir, desde la manipulación de la tarjeta de desarrollo Spartan 3E con sus módulos incorporados y todo el tratamiento digital que esto implica. Con base a lo anterior, este proyecto se estructura en cuatro fases de desarrollo. Una primera fase de investigación sobre el desarrollo de FPGA como nueva tecnología aplicada al procesamiento digital de audio; una segunda etapa ya enfocada en la implementación que implica el procesamiento previo y conversión a símbolos digitales; una tercera etapa del bloque de programación con la FPGA para el procesamiento por medio del filtrado e interfaz tarjeta-usuario, y por último la conversión de estos símbolos digitales procesados a señales continuas que permita realizar una etapa final de amplificación y muestra de resultados al procesamiento general.

de la familia SPARTAN y VIRTEX, utilizadas para aplicaciones de dispositivos como embebidos, control, sistemas de adquisición de datos y telecomunicaciones.

Para empezar definimos la estructura básica que nos presenta el FPGA (XC3S500E) incorporada en la tarjeta SPARTAN 3E (ver figura 1 y 2). Estructura lógica de compuertas universales con 500.000 programables, alrededor de 10.000 celdas lógicas y 232 pines de usuario, tanto de entrada como de salida. El FPGA es un dispositivo de fácil programación y mediante Lenguajes de Descripción de Hardware (HDL) se permite la programación concurrente y su principal ventaja es que puede programarse múltiples veces bajando los costos cuando en el diseño se encuentran fallos y requiera ser reprogramado.

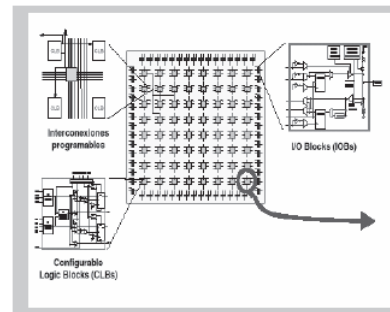


Figura 1. Estructura interna de Fpga [1].

## 2. Estado del arte

Actualmente el procesamiento digital de señales se realiza por medio de dispositivos como FPGA, entre otros, que vienen implementados en tarjetas de desarrollo, como por ejemplo el caso específico

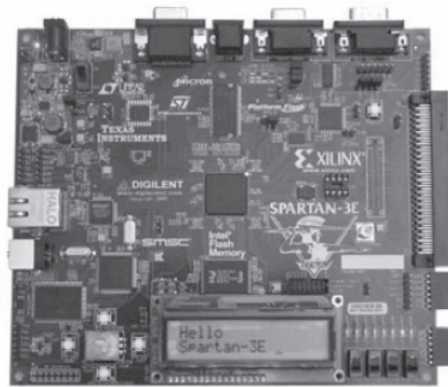


Figura 2. Spartan 3E starter kit board [2].

Las características esenciales de esta tarjeta son las siguientes:

- Reloj oscilador de 50 Mhz
- 64 Mbyte de DDR SDRAM
- *Flash* serial SPI de 16 Mbits
- *Flash* paralelo de 128 Mbits
- Pantalla LCD de dos líneas
- Puerto PS2, VGA, RS-232, Ethernet, interfaz USB, DAC/ADC, LED's, Switches, push-buttons, puertos de expansión, entre otros [2].

Los sistemas diseñados para el tratamiento digital de audio implican unas características en los dispositivos que se manejan como son conversores, filtros digitales encapsulados y de procesamiento, que en la ejecución del tratamiento de la señal puede llegar a ser un trabajo más extenso y menos confiable debido a tasas de muestreo, velocidad de procesamiento, y cantidad de bits a manejar lo que implica pérdida de información y de calidad.

Con este proyecto se procuró aprovechar las ventajas de configurabilidad, y entradas de usuario para

adaptar la interfaz, y la toma de mayor cantidad de datos y su procesamiento mediante el filtrado en un mismo instante de tiempo.

### 3. Metodología

Con el fin de crear unas bases de conocimiento y avance en FPGA, se requiere implementar un dispositivo que procese audio acoplándose a las características de la tarjeta de desarrollo y cumpliendo las necesidades básicas que esto implica, como la fidelidad y calidad del resultado final hacia el usuario, y que para este sea de su mayor agrado. El presente proyecto se dividió en cuatro etapas que permitieran facilitar su ejecución. La figura 3 muestra un bloque de ejecución y la figura 4 un diagrama de bloques del *hardware*.

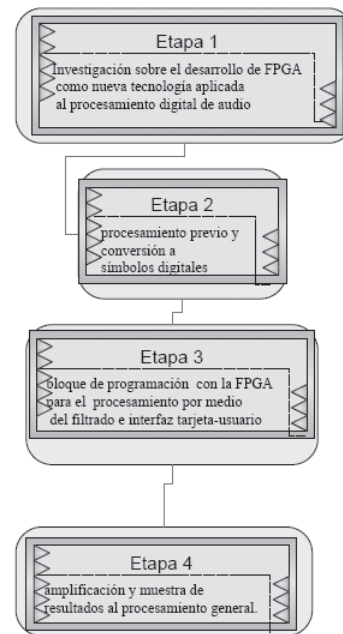


Figura 3. Bloques de ejecución para el desarrollo del Codec Digital.

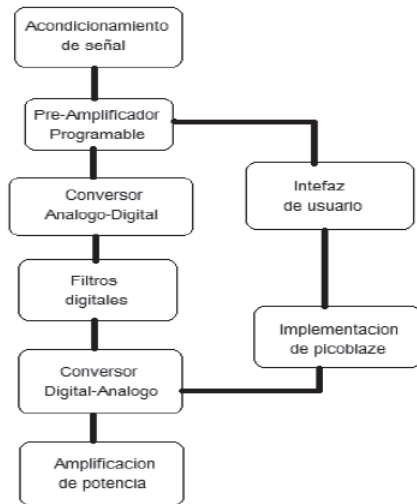


Figura 4. Diagrama de bloques del hardware implementado.

### 3.1 Procesamiento previo y conversión a símbolos digitales

Esta etapa está estructurada con base a una entrada estéreo y el procesamiento de sus dos canales independientes pero con igual proceso de acondicionamiento hasta la conversión análogo-digital. Para no afectar la calidad hasta la conversión de símbolos digitales se efectuaron las siguientes etapas:

#### Etapa de acondicionamiento por JFET

Configurado en *drain* común con el objetivo principal de disminución de ruido, alta impedancia de entrada y una mayor velocidad de conmutación, además del consumo de potencia que es menor a diferencia de los BJT.

#### Circuito seguidor de voltaje

Hace una buena etapa de aislamiento entre la fuente de voltaje y la carga, adapta impedancias con alimentación (5,-5).

#### Etapa de filtrado

Se realizan dos filtros, uno pasa-bajo a 100 Hz con el propósito de eliminar las fuentes de ruido de red que se presentan a esas frecuencias, y otro pasa-alto a 15 KHz., evitando señales indeseadas y por supuesto inutilizadas en frecuencias audibles. Estos fueron realizados con OPAMs con una ganancia de tan solo 2 respecto a la entrada, y su salida hacia un seguidor para acople nuevo de impedancias.

#### Etapa de offset (sumador)

Ante la salida de los filtros por características de entrada en el conversor (especificadas en la siguientes etapas) se tiene que realizar un circuito de *offset* sobre la señal de entrada. Dicho nivel es de 1.65 V., la mitad del voltaje de alimentación de la tarjeta (3.3 v). Esta etapa es realizada por medio de un sumador donde la señal de entrada es aplicada a un voltaje desde un zener referenciado a 1.65v, teniendo la señal de salida esperada.

#### Etapa de Ventana de voltaje

Esta etapa se presenta antes de llevar la señal hacia el conversor que exige un voltaje mínimo y máximo (dependiendo la ganancia interna, en este caso la mínima (-1)) de entrada, el cual es de +/- 1.25v., sumado al *offset*; esto implica un circuito que limite el voltaje. Para esto se utilizaron dos reguladores de voltaje positivo y negativo, respectivamente, con voltaje de (5,-5) y con potenciómetros que limitan la señal y recortan su ventana máxima de entrada desde 0.4 hasta 3.3. Ver los valores mínimos y máximos según ganancia específica (tabla 1) [2].

#### Preamplificador programable

La tarjeta de desarrollo Spartan 3e trae incorporado un preaplicador programable serial LTC6912-1 con dos entradas análogas. Las cuatro líneas SPI que son controladas son de vital importancia para el funcionamiento del dispositivo, el cual es controlado por el reloj del bus (SCK). El preamplificador soporta periodos máximo de 10 MHz., es activado

por un habilitador (amp\_cs) que funciona durante los ciclos de reloj que necesita recibir el dispositivo, que en este caso son cuatro bits por cada canal de entrada, ocho en total; esto es para enviar el dato predeterminado de cada ganancia en tan solo 8 ciclos de reloj. Las ganancias se presentan en la tabla 1. Esta ganancia es enviada desde la FPGA por el bus MOSI (master out- slave input) y la última línea SPI MISO (master input-slave output), que en este caso puede ser ignorada debido a que verifica el dato que se le envió previamente al preamplificador.

las cuatro líneas SPI y su forma de trabajo durante el envío de la trama.

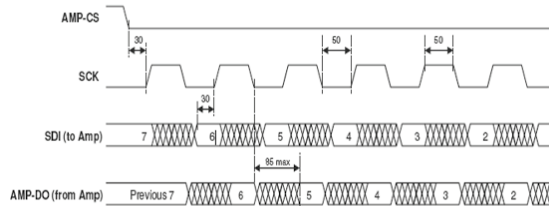


Figura 6. Pulsos de activación en protocolo spi [2].

Tabla 1. Configuración del servidor.

Gain	A3	A2	A1	A0	Input Voltage Range	
	B3	B2	B1	B0	Minimum	Maximum
0	0	0	0	0		
-1	0	0	0	1	0.4	2.9
-2	0	0	1	0	1.025	2.275
Gain	A3	A2	A1	A0	Input Voltage Range	
	B3	B2	B1	B0	Minimum	Maximum
-5	0	0	1	1	1.4	1.9
-10	0	1	0	0	1.525	1.775
-20	0	1	0	1	1.5875	1.7125
-50	0	1	1	0	1.625	1.675
-100	0	1	1	1	1.6375	1.6625

Como muestra la tabla 1 esta ganancia programable es negativa, el dato debe ser enviado primero desde el más significativo, que en este caso es el bit 3 del canal B hasta el bit 0 del canal A.

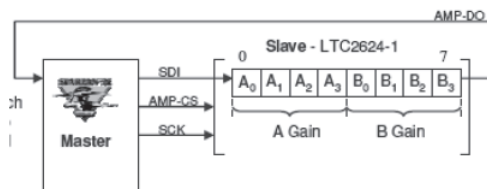


Figura 5. Trama de comunicación [2].

Inmediatamente después de enviar los 8 datos el AMP\_CS vuelve a nivel alto y el preamplificador interpreta el dato cargado.

El bus MOSI captura el dato siempre en flanco de subida del reloj SCK. En la figura 6 se especifican

### 3.1.1. Conversor analógico-digital

En el proceso de codificación de señal esta etapa hace parte fundamental del resultado final del proyecto. Al igual que todos los dispositivos desarrollados este también utiliza el protocolo SPI. Las características principales al momento de conversión son: el conversor LTC1407A-1 posee doble canal de operación a una tasa máxima de muestreo de 1.5 MHz por cada canal con un voltaje de operación de 3V, cada conversión es representada digitalmente con 14 bits por cada canal y una periodo de reloj de hasta 50 MHz. Inmediatamente cargado el dato desde el preamplificador el conversor toma la señal y espera el pulso de activación (AD\_CONV), que en este caso es activo por tan solo un instante a diferencia del habilitador del preamplificador que se mantiene en alto durante los ciclos que no trabaja. Cada salida digital está representada en complemento a 2, esto quiere decir que el bit más significativo de los 14 bits representa el signo del dato, teniendo en cuenta el nuevo nivel de señal que se aplicó en etapas pasadas de 1.65v y que en este caso sería el dato (0x00).

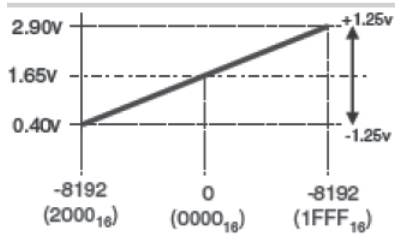


Figura 7. Intervalo de Conversión [3].

Del tipo de conversión podemos especificar el dato de salida según la ecuación 1, lo que representa valores desde  $-2^{13}$  hasta  $2^{13} = 8192$ .

$$D[13:0] = GAIN \frac{(V_{IN} - 1.65V)}{1.25} \times 8192 \quad (1)$$

En este dispositivo la trama de comunicación consiste básicamente en el pulso de habilitación seguido de 34 ciclos de reloj SCK, que permiten iniciar la conversión de los dos canales uno tras otro. De estos 34 datos digitales convertidos el dato es representado para el canal 1 desde (bit 2 hasta bit 15) y para el canal 2 desde (bit 18 hasta bit 31). Los otros bits pueden ser tomados sin importancia para el dato digital.

El dato muestreado hacia la FPGA por el bus MISO solo es presentado hasta el siguiente pulso de activación AD\_CONV.

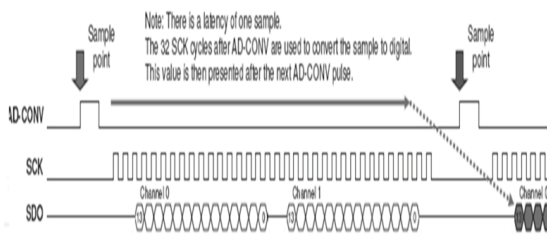


Figura 8. Intervalo de comunicación por muestra [2].

El dato es enviado hacia el maestro desde el bit más significativo, el MISO captura el dato a diferencia del anterior en flanco de bajada del SCK.

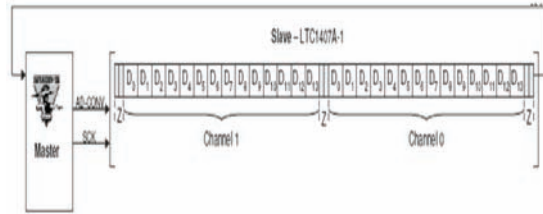


Figura 9. Trama de comunicación [2].

Cada muestra tomada desde el conversor se codifica por medio de PCM (Modulación por Pulso Codificado), basada en conversión a longitud fija y con base a números binarios que varían dependiendo de la amplitud de la señal analógica. Esta codificación nos da ventajas como confiabilidad y estabilidad; además, usa códigos eficientes y evita redundancias.

Finalmente, en la figura 10 podemos observar la fase principal de la conversión a símbolos digitales con los dos dispositivos controlados por spi (preamplificador y conversor).

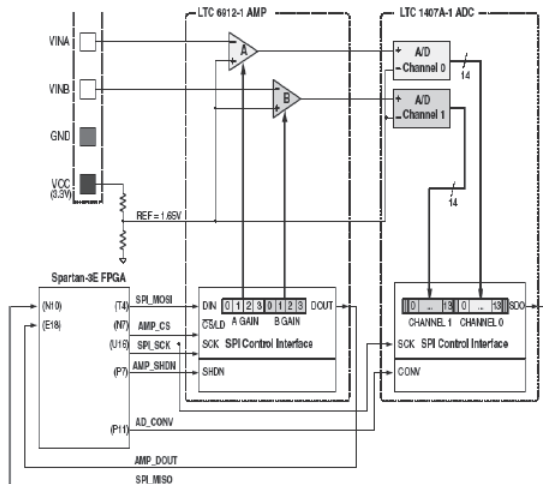


Figura 10. Unión etapa de conversión de la señal a símbolos digitales [2].



### 3.2. Bloque de programación con la FPGA para el procesamiento por medio de filtros e interfaz tarjeta usuario

En esta etapa se estructuran los procesos realizados a partir de la obtención de símbolos digitales en la etapa anterior. El usuario tiene la posibilidad de guiarse por mensajes que le presenta (tipo menú) la LCD, donde le especifica básicamente los tipos de filtros que quiere aplicar a la señal de audio que ingresó. De allí puede utilizar las entradas digitales tipo *switch* para la escogencia de la configuración deseada. Para llegar a esto se necesita desarrollar:

#### Filtros digitales

La herramienta XILINX 9.2 como aplicación al desarrollo de FPGAs permite varias formas de diseño, así como se utiliza el lenguaje estructurado VHDL y se programa por una serie de instrucciones basadas en ciclos de reloj y máquinas de estado, diseñando en diagramas de flujo. También se puede diseñar por medios esquemáticos diseñados con base a dispositivos lógicos: compuertas, *flipflops*, contadores, etc., y en representación del lenguaje como esquemático. Por último, para el diseño de filtros digitales es necesario utilizar el generador de CORE, que es una herramienta útil generadora de diferentes tipos de núcleos para aplicaciones como procesamiento de señal, comunicaciones y funciones matemáticas. Para nuestro caso se utiliza el CORE DA FIR (Distributed Arithmetic FIR filter), el cual consiste en el diseño de un esquemático con base a la previa configuración de características de un filtro tipo FIR.

#### Configuración DA FIR

Esta herramienta se basa en una secuencia realizada por ciclos de reloj, durante una tasa de muestreo que depende de factores como tamaño de entrada, orden del filtro y simetría de las muestras por realizar, y está mecanizada con base a adiciones, sustracciones y registros. La DA es una algorítmica computacional que permite la implementación eficiente de sumas ponderadas de productos, también denominados productos tipo *dot*, y que define la

importancia de las operaciones de procesamiento de la señal. El producto *dot* es una operación intensiva en cálculo, cuya velocidad es limitada por el circuito multiplicador. La implementación se detalla por los siguientes aspectos:

- Cada término del producto consiste en una variable (señal) y una constante (coeficiente), ambos en formato binario de punto fijo, pero no necesariamente de la misma longitud de palabra [4].
- En vez de computar el producto de un término con base a cada término en sí mismo, los productos parciales de todos los términos son calculados de forma simultánea y en el tiempo que lleve el cálculo de un único producto parcial. Esto se consigue precalculando todas las posibles salidas de los productos parciales acumulados y almacenándolos en una tabla de búsqueda (*look up table* o LUT) que sea direccionada por los bits del multiplicador (variable). Todas las variables de entrada son secuenciadas simultáneamente, en modo bit serie antes de direccionar la tabla LUT. Cada producto acumulado parcial es escalado de forma binaria y añadido a los productos parciales acumulados [4].
- El cálculo del producto *dot* completo tarda B ciclos de reloj, donde B es el número de bits de la variable de entrada [4].
- La herramienta permite la posibilidad de escoger entre diferentes tipos de filtros. Para nuestro caso se utiliza el tipo single rate FIR, diseñado para responder a una respuesta a un pulso de activación y proceso de muestreo cada cierto ciclo de reloj. Durante este tiempo los registros acumulan la convolución de las señales de entradas X(n) con la respuesta al impulso h(n), según la ecuación 2.

$$y_n = \sum_{k=0}^{N-1} b_k x(n-k) \quad (2)$$

Aplicado a la siguiente estructura:

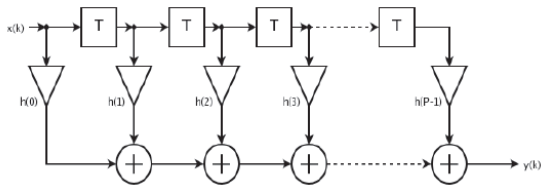


Figura 11. Estructura de filtros FIR [5].

El valor de la muestra de la entrada actual y algunas muestras anteriores (que previamente habían sido almacenadas) son multiplicados por unos coeficientes definidos. También podría tomar valores de la salida en instantes pasados y multiplicarlos por otros coeficientes. Finalmente, todos los resultados de estas multiplicaciones son sumados, y se da una salida para el instante actual.

Se utilizó el método de la ventana de KAISER para el diseño de los coeficientes, los cuales fueron generados en la herramienta FIR toolbox, configurando frecuencias de muestreo, orden del filtro y tamaño de bit de los coeficientes generados, que serán cargados en el CORE de XILINX archivos.coe. La figura 12 tomada del programa “FIRToolbox” muestra la respuesta en fase y frecuencia de un filtro pasa-bajo.

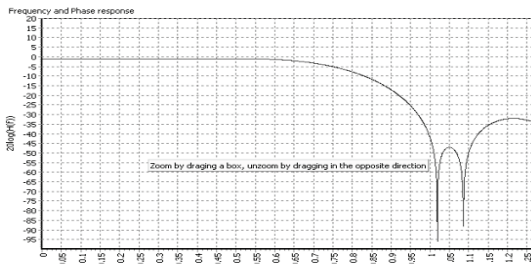


Figura 12. Respuesta en frecuencia pasa bajo.

Así mismo, los coeficientes generados según el orden del filtro.

Tabla 2. Coeficientes generados por FIRtoolbox.

Coef #	Theoretical	Practical
0	0.000000000000	0
1	-0.001459167975	0
2	0.006950530242	0
3	0.000000000000	0
4	-0.021459451902	-1
5	0.015270814486	1
6	0.042942001439	2
7	-0.067571361370	-4
8	-0.063103085111	-3
9	0.303460138027	16
10	0.571428571429	31
11	0.303460138027	16
12	-0.063103085111	-3
13	-0.067571361370	-4
14	0.042942001439	2
15	0.015270814486	1
16	-0.021459451902	-1
17	0.000000000000	0
18	0.006950530242	0
19	-0.001459167975	0
20	0.000000000000	0

Con cada uno de los esquemáticos generados por el CORE para cada filtro, que en este caso fue diseñado a frecuencias bajas, medias y altas audibles, obteniendo a la salida la señal filtrada proveniente desde el conversor.

### Interfaz tarjeta usuario

Para la interfaz con el usuario se utilizó la LCD, 4 switch y 3 pulsadores contenidos en la tarjeta. Para el desarrollo de esta etapa se necesita comunicar la FPGA por medio de picoblaze hacia el modulo LCD.

### 3.3. Conversión digital-análogo, amplificación de salida

En esta etapa se estructura la salida del procesamiento con el conversor DAC que también utiliza la comunicación SPI, con cuatro canales de salida.

#### 3.3.1. Conversor digital-análogo

El ltc2624 tiene una resolución de 12 bits sin signo por cada canal y puede trabajar a una frecuencia de reloj hasta de 50 Mhz con cuatro canales de conversión configurados por dirección, dos canales por cada referencia de voltaje y un nivel de operación de hasta 3V.



El pulso de habilitación SPI, llamado (DAC\_CS), espera ser puesto en nivel bajo para operar durante 32 ciclos de reloj SCK. La trama enviada desde la línea MOSI con el bit más significativo enviado primero consta de ocho bits de no importancia, cuatro bits de comando que específicamente y por defecto debe ser “0011”. Los siguientes cuatro bits de dirección determinan el canal que se va a utilizar (tabla 4) y doce bits del dato sin signo, el cual es el voltaje de salida determinado por la ecuación 3.

$$V_{out} = \frac{D[11:0]}{4096} * V_{REFERENCE} \quad (3)$$

Por último, cuatro bits de no importancia. Todos los bits son tomados en flanco de subida del SCK. En la tabla 3 se observan las direcciones de los conversores [2].

Tabla 3. Direcciones de conversores.

a <sub>3</sub>	a <sub>2</sub>	a <sub>1</sub>	a <sub>0</sub>	ADDRESS
0	0	0	0	DAC A
0	0	0	1	DAC B
0	0	1	0	DAC C
0	0	1	1	DAC D
1	1	1	1	All

Después de los 32 ciclos de reloj el habilitador vuelve a nivel alto.; en este caso la línea MISO puede ser ignorada y tan solo utilizada para la verificación de los datos enviados. Este dispositivo es un *reset* interno (DAC\_CLR) que debe estar en alto cuando el conversor esté funcionando.

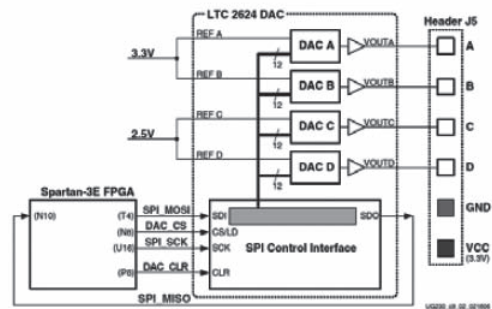


Figura 13. Interfaz salida de conversión digital-análogo [2].

### 3.3.2. Etapa de amplificación de potencia

Circuito diseñado para amplificar la señal de audio entregada desde los dos canales de conversión. Consta de un circuito integrado (TDA 2009) con las salidas estéreo. Sus características de operación son: amplificación de hasta 10W por canal, protección de corto circuito, amplificador tipo AB y necesita fuente de alimentación de hasta 28v y 3.5 A para adaptarlo a cualquier tipo de parlante y mantener alta fidelidad de la señal. [7]

## 4. Resultados

Para cada etapa se realizaron las pruebas correspondientes a su correcto funcionamiento individual.

### 4.1. Conversor análogo-digital

Se hicieron pruebas con un nivel de voltaje DC desde 0.4 hasta 2.9, comprobando la entrega de los datos digitalizados y visualizados sobre las señales SPI y leds, tanto para la configuración del preamplificador y del conversor. El amplificador está conectado de forma serie con el conversor análogo-digital, de tal forma que es vital primero programar la ganancia antes de convertir la primera muestra.

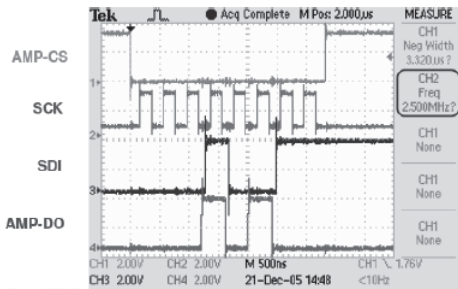


Figura 14. Prueba amplificador.

Se realizó el cambio de voltaje en la entrada para comprobar la sensibilidad y la resolución del convertor.

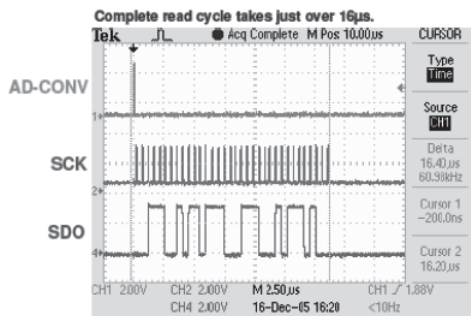


Figura 15. Prueba amplificador.

Según pruebas y visualización el dato convertido es entregado en complemento a 2 e, invertida la señal debido a la ganancia programable negativa. Para esto es necesario realizar un nuevo proceso de tratamiento de señal.

#### 4.2. Etapa de filtrado

Se configuraron diferentes tipos de filtros a varias frecuencias de muestreo y los respectivos barridos de frecuencia para escoger cuál configuración era la más adecuada para este proyecto.

A continuación se muestra un barrido de diferentes frecuencias en una misma simulación

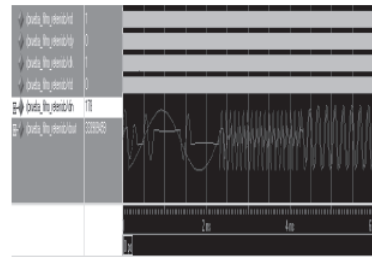


Figura 16. Prueba amplificador.

Se comprobó lo obtenido en las simulaciones en el momento del diseño de los coeficientes en la herramienta FIR toolbox, como frecuencia de corte y caída en dBs.

A continuación se muestra la verificación del filtro a una frecuencia de muestreo superior (una década arriba).

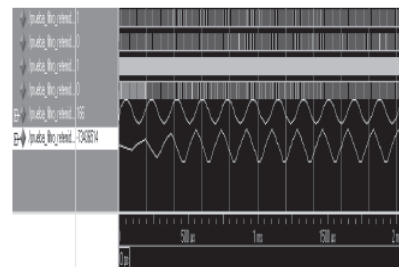


Figura 17. Segunda prueba filtrado.

#### 4.3. Diseño de interfaz por LCD

Se realizó el programa para la LCD en VHDL, el cual no fue realmente adecuado debido a que se convierte en un proceso muy extenso, ya que para publicar un mensaje se necesitaría enviar carácter por carácter y se ocuparían muchos ciclos de reloj. Por este motivo fue necesaria la utilización del *picozable* con comunicación hacia la LCD, creando tan solo en VHDL una memoria ROM de caracteres con los mensajes presentados al usuario y con un contador de recorrido dentro de esta memoria, puesto que la memoria del microprocesador es muy limitada y no fue posible enviar la cadena de caracteres.

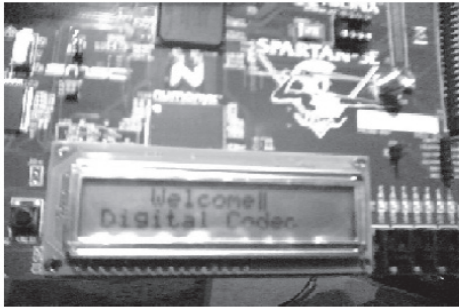


Figura 18. Funcionamiento LCD.

El puerto de salida de microprocesador es de ocho bits. Como resultado tenemos 256 posibles posiciones, es decir, 256 caracteres. Para una ampliación sería necesario la implementación de una concatenación con datos en el puerto de salida.

#### 4.4. Protocolo de comunicación SPI

El Serial Peripheral Interface necesita un dispositivo maestro, que en nuestro caso es la FPGA, y varios dispositivos esclavo. Tiene tres líneas principales compartidas entre todos los dispositivos y la línea de habilitadores para cada uno de los esclavos. Por esta razón se necesita sincronizar los habilitadores teniendo en cuenta que solo debe funcionar un dispositivo a la vez. Al no deshabilitar un dispositivo se creará interferencia entre los datos de comunicación, aunque las líneas SPI funcionen correctamente.

#### 4.5. Conversor digital analógico

La primera prueba que se realizó fue convertir datos digitales en una señal diente de sierra, lo cual se logró mediante el incremento de un dato por cada muestra, ya que el conversor analógico digital entrega los datos en complemento 2 y además la señal viene invertida desde el preamplificador. Es necesario diseñar un algoritmo que permita cambiar los datos de forma que el conversor digital analógico reciba los datos sin signo y no invertida respecto al original, como lo especifica la trama de comunicación.

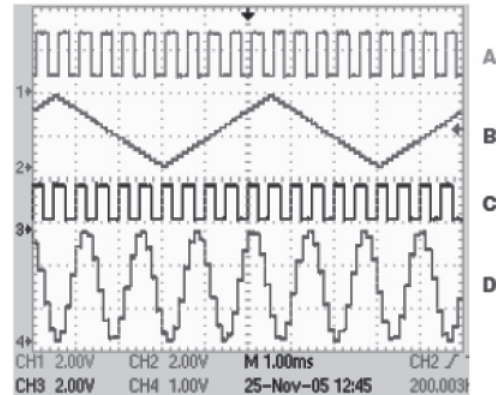


Figura 19. Muestreo de varias señales a la salida de los canales de conversión.

El conversor digital analógico no puede trabajar con distintos datos en la salida de los cuatro canales en una misma muestra, lo que implica la pérdida de la muestra de alguno de los dos canales del conversor analógico digital por cada conversión.

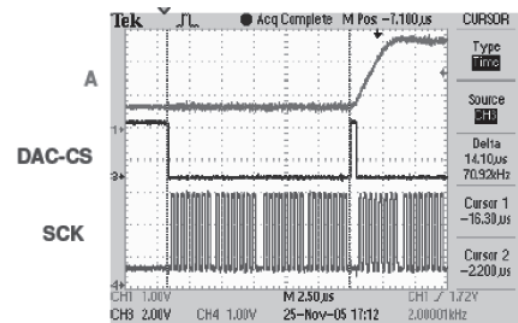


Figura 20. Conversión analógica digital.

## 5. Conclusiones

La implementación de un códec digital de audio por medio de FPGA tiene ventajas como la velocidad de procesamiento, almacenamiento de datos y versatilidad en cuanto al algoritmo de procesamiento según la topología de implementación, y por ende el ancho de los coeficientes utilizados para calcular la ecuación en diferencias en el caso de los filtros.

La manipulación de la tarjeta Spartan 3E presenta los retos de adquirir conocimientos en el manejo de nuevos dispositivos y tecnologías que pueden ser aplicadas al grupo de investigación DIGITI. El desarrollo de la tarjeta Spartan 3E con este proyecto permite la manipulación de tan solo una parte de los módulos contenidos. La utilización de los demás módulos permitiría que el proyecto tuviera mayores aplicaciones.

El desarrollo sobre el lenguaje VHDL motiva al programador a diseñar de forma concurrente, teniendo en cuenta los procesos en paralelo con señales jerárquicas como el reloj y *reset*; además de facilitar la estructuración de procesos digitales con herramientas como máquinas de estados, esquemáticos y otras aplicaciones contenidas en el *software* Xilinx.

Es de vital importancia cumplir específicamente las tramas de comunicación sobre un protocolo como el SPI para el correcto funcionamiento de los dispositivos, y adaptarse a estas para realizar

la correcta sincronización del proyecto y no ver afectados los resultados esperados.

La implementación de un *software* como el *picoblaze* o incluso un procesador de más alto nivel como *microblaze* permitiría el desarrollo de sistemas embebidos con múltiple propósitos y procesos que pueden ser aplicados en nuevas investigaciones para el beneficio del grupo de investigación y la universidad.

Se observa que el rendimiento de los filtros digitales en FPGA tiene unas prestaciones bastantes altas en cuanto a cantidad de compuertas utilizadas dentro del dispositivo, en comparación con la cantidad de ciclos de reloj necesarios y posibles canales análogos que se puedan trabajar, lo cual hace que estas tareas se vean simples en comparación con las mismas características en otros dispositivos. El implementar más de un filtro digital en un dispositivo no concurrente como un DSP o un DCS se convierte en una tarea dispendiosa, por no decir titánica.

---

## Referencias bibliográficas

---

- |   |  |
|---|--|
| <p>[1] G. Sánchez, “Microelectrónica”, Universidad Francisco de Paula Santander, vol. 20, agosto, 2002.</p> <p>[2] Xilinx. “Datasheet User Guide Spartan 3E”. [En línea]. Disponible: <a href="http://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf">http://www.xilinx.com/support/documentation/boards_and_kits/ug130.pdf</a></p> <p>[3] K. Chapman, “Initial Design for Spartan-3E Starter Kit”. [En línea]. Disponible: <a href="http://www.xilinx.com/products/boards/s3estarter/files/s3esk_startup.pdf">http://www.xilinx.com/products/boards/s3estarter/files/s3esk_startup.pdf</a></p> <p>[4] R. Bautista. “Investigación y educación”. [En línea]. Disponible: <a href="http://www.tesisenxarxa.net/TESIS_UB/.../11.LSB_BIBLIO_PUB_ANEXOS.pdf">www.tesisenxarxa.net/TESIS_UB/.../11.LSB_BIBLIO_PUB_ANEXOS.pdf</a></p> | <p>[5] E. Jacinto, “Tutorial for the design of fir digital filters over fpga”, Revista Ingenieria, vol. 14, No. 1, pp. 13-17, junio 2009.</p> <p>[6] K. Chapman. Senior Staff Engineer – KCPSM3 8 bits microcontroller for Spartan 3 and virtex. [En línea]. Disponible: <a href="http://www.eng.auburn.edu/~strouce/class/elec4200/ug129.pdf">http://www.eng.auburn.edu/~strouce/class/elec4200/ug129.pdf</a></p> <p>[7] STMicroelectronics, “Datasheet, TDA2009A 10 +10W STEREO AMPLIFIER”. [En línea]. Disponible: <a href="http://www.st.com/stonline/books/pdf/docs/1457.pdf">http://www.st.com/stonline/books/pdf/docs/1457.pdf</a></p> <p>[8] Xilinx, “Spartan-3 SP-305 Development Board (HW-SP305-xx)”. [En línea]. Disponible: <a href="http://www.xilinx.com/xlnx/">http://www.xilinx.com/xlnx/</a></p> |
|---|--|

- xebiz/designResources/ip\_product\_details.jsp?key=HW-SP305-US
- [9] Linear Technology, “LTC2624 Quad DAC Data Sheet”. [En línea]. Disponible: <http://www.linear.com/pc/downloadDocument.do?navId=H0,C1,C1155,C1005,C1156,P2048,D2170>
- [10] Linear Technology, “LTC6912 Dual Programmable Gain Amplifiers with Serial Digital Interface”. [En línea]. Disponible: <http://www.linear.com/pc/downloadDocument.do?navId=H0,C1,C1154,C1009,C1121,P7596,D5359>
- [11] Linear Technology, “LTC1407A-1 Serial 14-bit Simultaneous Sampling ADCs with Shutdown”. [En línea]. Disponible: <http://www.linear.com/pc/downloadDocument.do?navId=H0,C1,C1155,C1001,C1158,P2420,D1295>