ESTUDIO DEL DSP ADZS-BF561 DE ANALOG DEVICES, FAMILIA BLACKFIN Y DESARROLLO DE SUS APLICACIONES

Franklin Andrés Rodríguez Játiva
ESCUELA POLITECNICA DEL EJÉRCITO
Departamento de Eléctrica y Electrónica
fandresrj@hotmail.com

Resumen— *El presente artículo describe el estudio de la tarjeta ADZS-BF561 de la familia de procesadores "Blackfin". Se verifican las características generales de hardware y de software del procesador ADSP-BF561 como son la arquitectura de su procesador, set de registros, set de instrucciones, interfaces y puertos, además de su conexión e instalación. Se revisa la utilización del software de desarrollo "VisualDSP++" y sus herramientas a través de la creación y depuración de aplicaciones como: operaciones lógicas, aritméticas y con vectores, accesos hacia y desde memoria y utilización de archivos externos. Se desarrollan aplicaciones en lenguaje C y en ensamblador de: Control y Manipulación de Banderas Programables (Leds y Pulsadores), Adquisición y Almacenamiento de Señales de Audio, Aplicación de Filtros digitales a las señales de audio, Manipulación de imágenes y Aplicación de filtros digitales a imágenes.*

Los filtros digitales de audio diseñados son de Tipo FIR, con una frecuencia de muestreo de 48 KHz y de varios tipos según su respuesta en frecuencia y se ha utilizado la herramienta "fdatool" de Matlab para obtener los coeficientes del filtro (se utilizó 32, 64, 128, 256, 512 y 1024 coeficientes). En la manipulación de imágenes se aplicaron operaciones orientadas al pixel y a la región.

Índice de Términos—DSP, FIR, procesamiento digital de audio e imagen.

I. INTRODUCCIÓN

Hoy en día, se tiene al alcance procesadores especializados en el análisis y procesamiento de señales, que se los conoce con el nombre de DSP o "Procesador Digital de Señales".

Estos sistemas poseen la capacidad para realizar tareas y aplicaciones que antes solo se las realizaba de forma analógica; pero ahora, gracias a ellos, se las realiza de manera digital con las ventajas que esto representa; obteniendo una mayor velocidad de procesamiento a un menor costo y tamaño, una mayor flexibilidad y facilidad en el tiempo de desarrollo e implementación, dado que con solo cambiar unas líneas del código del programa, se obtiene una aplicación diferente con el mismo hardware. Además de que brindan una mayor velocidad de transmisión, facilidad de almacenamiento, una mayor precisión en los datos y un menor deterioro de la señal procesada.

Los sistemas DSP están constituidos principalmente por un microprocesador con un set de instrucciones, un hardware y un software optimizados, para permitir el desarrollo de aplicaciones que requieren operaciones numéricas a muy alta velocidad.

"Analog Devices" ha introducido en el mercado de los DSP a la familia de procesadores "BlackFin", que destacan por su alto rendimiento para aplicaciones de convergencia de varios formatos como audio, vídeo, voz y procesamiento de imágenes [1]. El sistema de desarrollo ADSP-BF561 EZ-KIT Lite es uno de los modelos más potentes de la familia "Blackfin", con un procesador de doble núcleo asegurando un alto rendimiento. Además posee un subsistema mejorado de acceso directo a memoria "DMA" y una administración dinámica de energía "DPM"; por lo que el DSP ADSP-BF561 puede soportar las más complejas tareas de control y procesamiento de señales, mientras mantiene una alta tasa de transferencia de datos y un consumo energético eficiente.

Las características del DSP ADSP-BF561 resultan ideales para una amplia gama de aplicaciones

industriales, de instrumentación, de carácter médico, y también para aplicaciones de multimedia. Es por esto que este sistema DSP ha sido escogido por el "DEE" en el área de procesamiento digital de señales, para mantenerse a la vanguardia en este campo de investigación y como una herramienta para el desarrollo de futuros proyectos de nuestra universidad.

II. ESTUDIO DEL PROCESADOR DIGITAL DE SEÑALES ADZS-BF561

El procesador ADSP-BF561 pertenece a la familia de procesadores "Blackfin" de "Analog Device", cuyas principales características son ofrecer un alto rendimiento y control en el procesamiento de señales, además de un bajo consumo de energía.

A. Hardware del ADSP-BF561

La arquitectura del procesador digital de señales ADSP-BF561 se muestra en la figura 1 y consta principalmente de dos segmentos básicos que son: su doble núcleo (donde se ejecutan las instrucciones) y los periféricos de E/S (donde se guardan o cargan los datos).

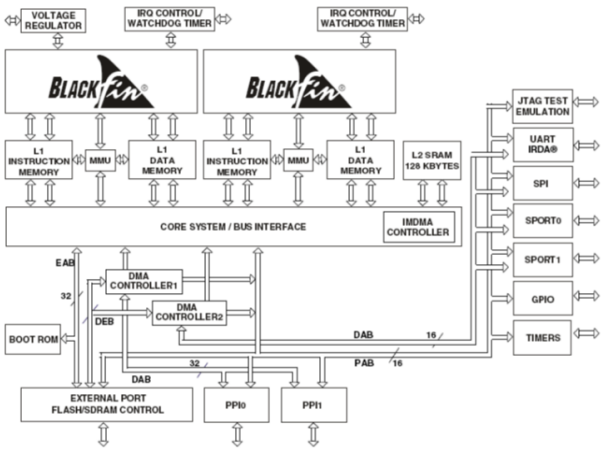

Figura 1. Diagrama de Bloques del ADSP-BF561

El procesador ADSP-BF561 posee dos núcleos "Blackfin" idénticos [1]. La arquitectura de cada núcleo se muestra en la figura 2, y está formado por 4 bloques que son:

La Unidad Aritmética: que permite realizar operaciones tanto aritméticas como lógicas, de multiplicación, desplazamientos lógicos y aritméticos, empaquetamiento y extracción de bits.

La Unidad de Direccionamiento: que permite un adecuado direccionamiento hacia los datos requeridos en las operaciones, desde y hacia la memoria.

La Unidad de Control: que controla el flujo del código de un programa, es decir, provee la dirección de la siguiente instrucción a ser ejecutada por el procesador.

Los Archivos de Registro: son los encargados del almacenamiento de datos o direcciones. Incluyen los registros acumuladores.

Estos cuatro bloques se muestran en la figura 2

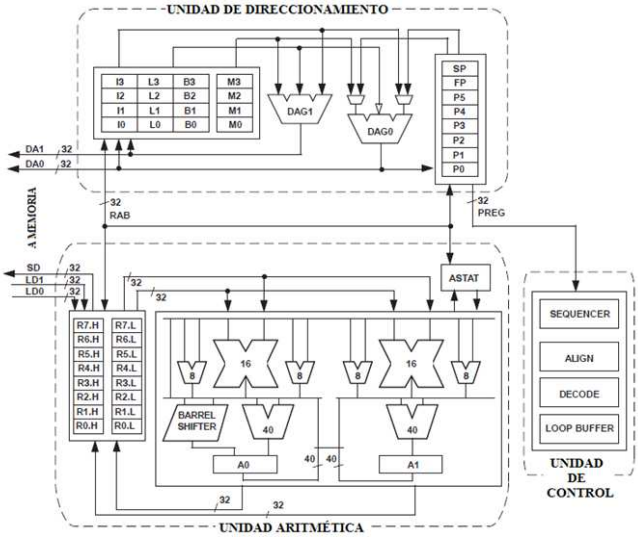

Figura 2. Arquitectura del Núcleo del ADSP-BF561

En cuanto a su memoria, posee una organización niveles, para conseguir el rendimiento de una memoria de gran velocidad, al costo de una memoria de baja velocidad.

Su sistema de acceso directo a la memoria le permite al procesador mover datos entre la memoria y un periférico o entre memoria y memoria sin intervención del núcleo.

Cuenta con un sistema de administración dinámica de energía, que le permite cambiar dinámicamente la frecuencia de operación y el voltaje del núcleo, para optimizar el consumo de energía de acuerdo a cada aplicación.

El procesador ADSP-BF561 posee doce temporizadores de 32 bits de propósito general, usados para generar interrupciones al núcleo del procesador. Además cada núcleo tiene un temporizador de núcleo y un temporizador "guardian".

El ADSP-BF561 tiene 48 banderas programables bidireccionales, de propósito general como I/O. Algunas banderas programables controlan los periféricos.

En cuanto a su sistema de interrupciones, posee dos subsistemas que trabajan juntos: el

Controlador de Interrupciones del Sistema (SIC) y el Controlador de Eventos del Núcleo (CEC). El CEC soporta nueve interrupciones de propósito general (IVG7-IVG15) además de las interrupciones dedicadas y eventos de excepciones.

Entre los periféricos que posee el ADSP-BF561, están la Interfaz para Periféricos Serie, la Interfaz para Periféricos Paralela, los Controladores de Puerto Serial y el Controlador de Puerto UART.

B. Software del ADSP-BF561

El procesador soporta los siguientes tres modos de operación: de Usuario (aplicaciones), Supervisor y Emulación. Además cuenta con el estado de inactividad y el estado de reseteo.

Los procesadores "Blackfin" utilizan un modelo de encauzamiento entrelazado "pipeline" para la ejecución de instrucciones, que consiste en descomponer la ejecución de cada instrucción en varias etapas para poder empezar a procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez. El procesador ADSP-BF561 tiene un "pipeline" de 10 etapas.

La entrada y salida de datos desde las unidades funcionales se realiza a través del banco de registros de propósito general R0 a R7. Este tipo de arquitectura se conoce como load/store y es típica de los procesadores RISC.

La Unidad de Direccionamiento Aritmético del ADSP-BF561 soporta los siguientes modos de direccionamientos de memoria:

- Indirecto
- Con Auto incremento-decremento.
- Indexado con offset
- Con pre y post- modificación
- Con buffer circular
- Bit reverso

El set de instrucciones para la familia del procesador Blackfin presenta una sintaxis diseñada de tal manera que brinda una facilidad de codificación y de lectura de los programas. Las instrucciones han sido optimizadas para obtener un tamaño en memoria menor en sus aplicaciones. El set de instrucciones también brinda muchas instrucciones multifunción que permiten al programador usar muchos recursos del núcleo con una sola instrucción.

III. CARACTERISTICAS DEL SISTEMA DE DESARROLLO ADZS- BF561 EZ-KIT LITE

Las principales características de tarjeta ADZS-BF561 EZ-KIT LITE se presentan a continuación:

- Procesador Blackfin ADSP-BF561 600 MHz
- 64 MB SDRAM (16 x 16 bits x 2 chips)
- 8 MB Flash (4MB x 2 chips)
- Interfaz análoga de audio
  - Codec de audio AD1836A multi-canal de 96 Khz de Analog Devices
  - 4 entradas de audio con conectores RCA (2 canales estéreo)
  - 6 salidas de audio con conectores RCA (3 canales estéreo)
- Interfaz análoga de video
  - Decodificador ADV7183A con 3 entradas de video con conectores RCA
  - Codificador ADV7179 con 3 salidas de video con conectores RCA
- Transmisor-Receptor Asíncrono Universal (UART)
- Conector DB9
- LEDs
  - 20 LEDs: 1 de poder (verde), 1 de reset de la tarjeta (rojo), 1 de USB (rojo), 16 de propósito general (ámbar) y 1 de monitoreo USB (ámbar)
- Pulsadores
  - 5 pulsadores con lógica de rebote: 1 de reset y 4 de banderas programables
- Interfaces de expansión
  - PPI0, PPI1, SPI, EBIU, Timers11-0, UART, banderas programables, SPORT0, SPORT1

El códec de audio AD1836A ofrece tres canales de salida de audio estéreo (DAC- convertidor digital análogo) y dos canales de entrada de audio estéreo (ADC – convertidor análogo digital). El códec soporta una frecuencia de muestreo de 48 y 96 kHz y resoluciones de 16, 20 y 24 bits. La interfaz SPORT0 del procesador se enlaza con los datos de audio estéreo de entrada y los pines de

salida del códec AD1836A [2]. El procesador es capaz de transferir datos a los códecs de audio en el modo multiplexación por división en el tiempo (TDM) o con el modo de interfaz serie de dos cables (TWI).

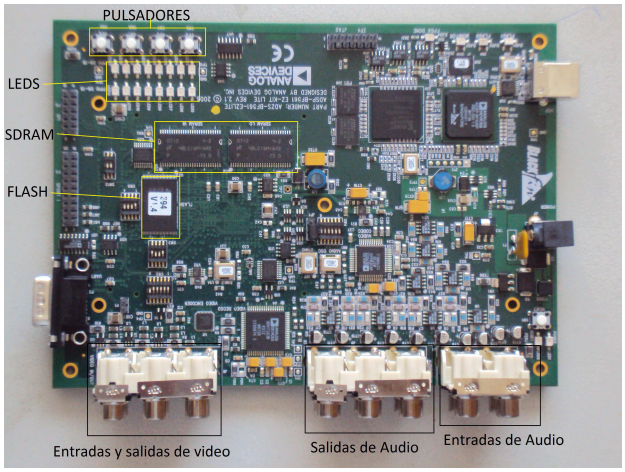

Figura 3. Vista superior de la tarjeta ADZS-BF561

IV. DESCRIPCION Y USO DEL SOFTWARE DE DESARROLLO "VISUALDSP++"

El software de desarrollo "VisualDSP++" [3] presenta las siguientes características:

- Gran capacidad de edición
- Flexibilidad en la gestión de proyectos
- Flexibilidad del Espacio de trabajo y fácil acceso a las herramientas de desarrollo
- Facilidad de cambio entre actividades de depuración y construcción
- Soporte de múltiples lenguajes
- Control efectivo de depuración
- Herramientas para mejorar el rendimiento de las aplicaciones

El software VisualDSP++ incluye las siguientes herramientas de desarrollo:

- Compilador C/C++ (Compiler)
- Librerías de rutinas de Matemáticas, DSP y C
- Ensamblador (Assembler)
- Enlazador (Linker)
- Separador (Splitter)
- Cargador (Loader)
- Simulador (Simulator)

El software de desarrollo VisualDSP++ posee una interfaz fácil de utilizar para los programadores. Como se muestra en la figura 4, consta de una ventana de Gestión de Proyectos, una venta del Editor, la ventana de Salida y la ventana de depuración. La ventana de proyectos nos permite trabajar en un orden jerárquico con nuestros archivos y proyectos, la ventana del Editor nos permite ver y editar los archivos, la ventana de Salida nos muestra los mensajes de E/S y entrada de scripts y la ventana de depuración nos permite seguir los procesos de depuración que llevemos a cabo.

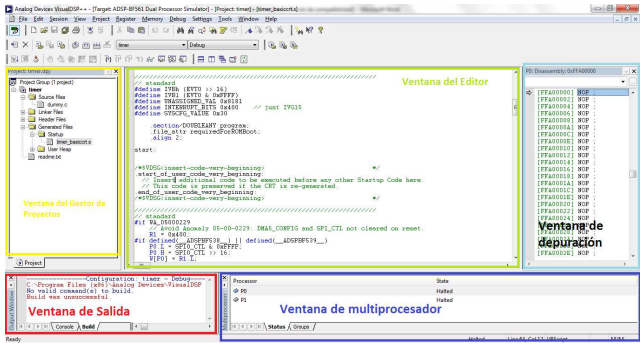

Figura 4. Software de desarrollo VisualDSP++

Para crear un proyecto nuevo en Visual DSP++, se realiza los siguientes pasos:

Del menú File, escoger New y después Project. En la siguiente ventana seleccionar "Standard application", seleccionar el nombre del proyecto y directorio y darle click a "Next". A continuación, desmarcar la opción "Add template source code to the application", ya que se va a agregar los archivos al proyecto manualmente. Darle click a Next.

En las opciones de configuración del uso de los núcleos del procesador para la aplicación. Seleccionar Single core: Single Application, ya que para la siguiente aplicación el procesador va a trabajar como un procesador de núcleo simple. Darle click a Next.

En las opciones para añadir el código de arranque o un archivo de descripción del linker (.LDF), marcar "Don´t add an LDF" or "startup code" y dar click en "Finish" y aparecerá la ventana del proyecto.

Para crear o incluir los archivos del proyecto, ir al menú "File" escoger New y después "File". En el archivo creado ingresar el código de programación e ir al menú "File", escoger "Save

as" y después "File" y guardarlo dentro de la carpeta del proyecto con el nombre deseado.

Para compilar el programa presionar F7 para comprobar q no haya errores en el código. En caso de no existir errores en la compilación, se procede a ejecutar el programa a presionando F5.

En el CD y en el documento escrito del presente proyecto de grado, se encuentran códigos de aplicaciones que incluyen sumas y ordenamiento de vectores, manipulación de registros y memoria, uso de archivos externos. Todos ellos resultan ideales para familiarizarse con la programación de la tarjeta ADSP-BF561.

V. DESARROLLO DE APLICACIONES CON EL ADZS-BF561

A. Control y manipulación de banderas programables

Estas aplicaciones muestran los registros necesarios para la inicialización, configuración y funcionamiento de las banderas programables, tanto para los pulsadores; como para los leds, sus configuraciones, su funcionamiento y su control. Además muestran el uso y configuración de los temporizadores y de las interrupciones del sistema.

Las aplicaciones realizan secuencias de encendido y apagado de los leds. El estado de los leds y de los pulsadores se controla de dos formas: por software usando lazos de programación y por hardware usando los temporizadores y las interrupciones del sistema. Para comprobar el funcionamiento de los pulsadores se les ha asignado funciones para alterar las secuencias de encendido/apagado de los leds, por ejemplo en duración o dirección.

Como se observa en la figura 5, los 4 botones (SW6 - SW9), corresponden a las banderas PF5 - PF8 respectivamente y por lo tanto se encuentran en el primer registro de banderas (o registro 0, por ejemplo: FIO0_FLAG_D). Los Leds de propósito general (LED5 – LED20), corresponden a las banderas PF32 - PF47 respectivamente y ocupan por completo el tercer registro de banderas (por ejemplo: FIO2_FLAG_S).

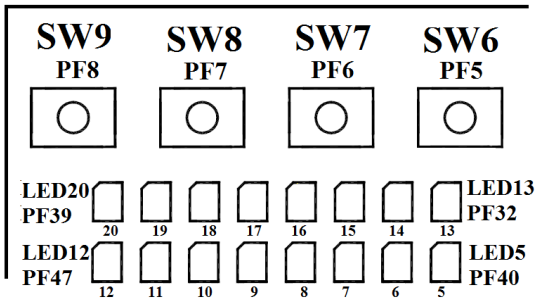

Figura 5. Asignación de Pulsadores y Leds a banderas programables

El siguiente código muestra la forma de configurar todas las banderas (PF0 – PF15) como entradas, en donde están incluidos los botones:

```
p0.l=lo(FIO0_DIR);
p0.h=hi(FIO0_DIR);
r0.l=0x0000;
w[p0]=r0;
```

Y su correspondiente código en C sería el siguiente:

```
*pFIO0_DIR  = 0x0000;
```

Las banderas programables cuando son configuradas como entradas pueden generar interrupciones, para ello se deben configurar los registros FIOn_INEN y FIOn_MASKA_D para habilitar la generación de interrupciones en esa bandera y quitar la máscara de interrupción respectivamente, además configurar el registro FIOn_POLAR para la polaridad, FIOn_EDGE para configurar la sensibilidad y que la interrupción se active por nivel o por flancos y el registro FIOn_BOTH para que se active con flanco simple o en ambos flancos. Un ejemplo de configuración de los botones para generar interrupciones seria:

```
P0.L = LO(FIO0_INEN);
R0 = 0x01e0(z);
w[p0] = r0.l;
P0.L = LO(FIO0_MASKA_D);
R0= 0x01e0(z);
w[p0] = r0.l;
P0.L = LO(FIO0_EDGE);
R0 = 0x01e0(z);
w[p0] = r0.l;
```

Y su código equivalente en C seria:

```
*pFIO0_INEN          = 0x01e0;
```

```
*pFIO0_MASKA_D   = 0x01e0;
*pFIO0_EDGE      = 0x01e0;
```

Un ejemplo para configurar el Timer0 con 0,5 segundos y con generación de interrupciones sería:

```
// PWM, conteo al fin del periodo, interr.
P1.L = LO(TIMER0_CONFIG);
P1.H = HI(TIMER0_CONFIG);
R1 = 0x0019;
w[ P1 ] = R1.l;
//configura el periodo con 0,5 segundos
P1.L = LO(TIMER0_PERIOD);
P1.H = HI(TIMER0_PERIOD);
R0 = 0;
BITSET ( R0 , 26 );
//carga 0x04000000 en R0
[ P1 ] = R0;
//configura el ancho de pulso
P1.L = LO(TIMER0_WIDTH);
P1.H = HI(TIMER0_WIDTH);
R0 = 1;
[ P1 ] = R0;
//habilita el timer
P1.L = LO(TMRS8_ENABLE);
P1.H = HI(TMRS8_ENABLE);
w[ P1 ] = R0.l;
```

Su código equivalente en C seria:

```
*pTIMER0_CONFIG = 0x0019;
*pTIMER0_PERIOD = 0x04000000;
*pTIMER0_WIDTH  = 0x00000001;
*pTMRS8_ENABLE  = 0x0001;
```

B. Manipulación de señales de audio

Estas aplicaciones muestran los conceptos necesarios para trabajar con señales de audio en tiempo real con el ADSP-BF561 Ez Kit Lite. Estos conceptos incluyen el estudio del códec de audio AD1836, los formatos de datos utilizados, el diseño de filtros y generación de señales en Matlab y el uso de herramientas que complementan la manipulación de señales de audio con el ADSP-BF561 EZ-KIT Lite.

A través de una rutina de inicialización de audio, se muestra como conectar (fig. 6) y configurar el códec de Audio AD1836, sus características y su funcionamiento en la adquisición de señales de audio. Se ha programado la misma aplicación en lenguaje ensamblador y en lenguaje C, para apreciar las diferencias.

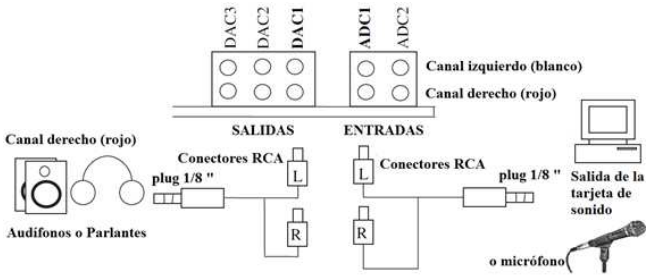

Figura 6. Conexión de la interfaz de audio del ADZS-BF561

A la rutina de inicialización se le añade las siguientes funciones: almacenamiento de audio en la memoria de la tarjeta ADZS-BF561, borrado y reproducción del audio almacenado y paso del audio en tiempo real. También se realiza un procesado de la señal de audio añadiéndole un control de volumen. Todas estas funciones están acompañadas de su led indicador. De esta manera se incluyen los conocimientos adquiridos en las aplicaciones anteriores, para obtener las funciones de un reproductor multimedia.

Los coeficientes de los filtros FIR, se diseñan en Matlab para implementarlos en el ADSP-BF561 EZ-KIT Lite [4].

Para el diseño de los filtros se ha escogido los filtros FIR (pasa-bajos, pasa-altos, pasa-banda y rechaza-banda. Por ejemplo, para el filtro pasa-bajos, se escogió que anule las frecuencias menores a 4000Hz y que no altere al resto, la frecuencia de muestreo será 48000Hz, además se quiere fase lineal. Las especificaciones de este filtro son las siguientes:

- Tipo de Respuesta: Pasa-bajos
- Método de diseño: FIR (Constrained Equiripple)
- Orden del Filtro (# de coeficientes - 1) = 128
- Frecuencia de muestreo Fs: 48000 Hz
- Frecuencia de corte Fc: 4000 Hz
- Atenuación parada Astop: 50 dB

Las figuras 7a y 7b muestran la configuración y respuesta en frecuencia del filtro pasa-bajos en la herramienta “FDATOOL” de matlab.

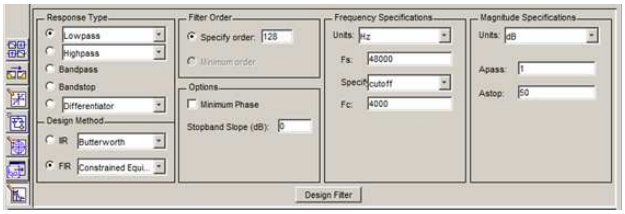

Figura 7a. Configuración filtro pasa-bajos en "fdatool"

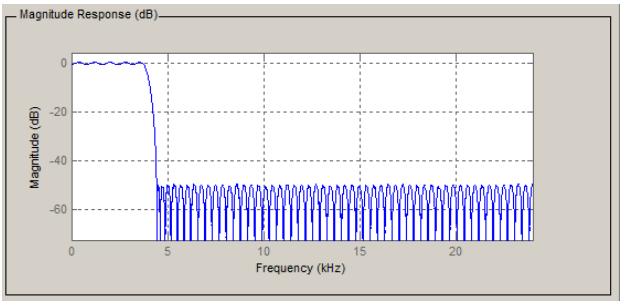

Figura 7b. Respuesta en frecuencia del filtro pasa-bajo

En la figura 8 se muestra el algoritmo implementado en la aplicación en el ADZS-BF561.

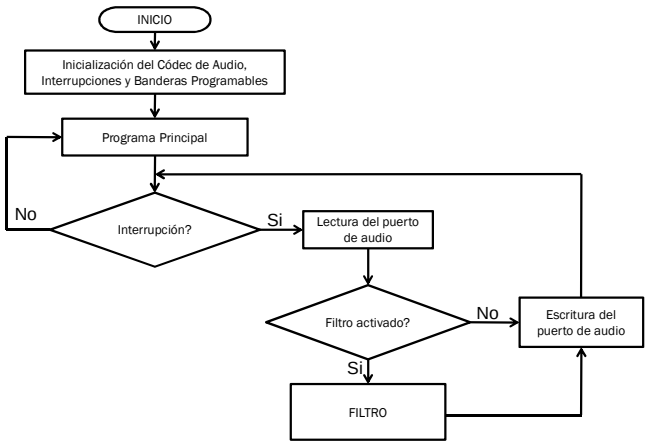

Figura 8. Algoritmo de implementación de la aplicación de filtros

La aplicación permite aplicar 4 tipos de filtros FIR a la señal de entrada. Un filtro pasa-bajo, un pasa-alto, un pasa-banda y un filtro rechaza-banda. Todos estos filtros se pueden activar y desactivar a través de los 4 pulsadores (SW6-SW9), además cada filtro posee su LED indicador.

C. Manipulación de imágenes con el ADSP-BF561 EZ-KIT LITE

Una imagen digital es una información visual recibida, representada, procesada, guardada y transmitida por sistemas digitales. El procesamiento digital de imágenes tiene muchas características en común con el procesamiento de señales en una dimensión (1D), como el de las señales de audio.

Se revisa la teoría desde la representación de una imagen, los espacios de color, su conversión y se comienza usando el visor de imágenes del VisualDSP++ para leer y guardar imágenes en la memoria del ADSP-BF561.

Para cargar una imagen desde la computadora hacia la memoria del Blackfin con el software VisualDSP++, usando el BF561 como plataforma para cargar las imágenes hay que hacer click en el menú View→Debug Windows→Image Viewer…, en donde aparecerá una ventana de configuración como se muestra en la figura 9, en donde hay que indicar la localización de la imagen a cargar a la memoria del Blackfin. Se establece en "Start address" a 0x0 y "Memory Stride" a 1. La dirección de inicio indica la posición en la memoria SDRAM del procesador Blackfin.

Figura 9. Configuración para cargar imágenes en el ADSP-BF561

Para visualizar la imagen guardada en la memoria de la tarjeta dsp, cargamos otra ventana del visor de imágenes, pero en la opción "Image Source" seleccionamos "DSP Memory", y en "Image Info" los valores correctos de la imagen guardada en la tarjeta. Dependiendo de la dimensión de la imagen, toma varios minutos en cargar o en transferir la imagen y mostrarse en la ventana del Visor de imágenes.

Se trabaja con los espacios de color, convirtiendo imágenes a RGB, NTSC, YCbCr y escala de grises.

La figura 10 muestra la imagen original a color (RGB) y las tres imágenes en B&W extraídas de diferente forma usando espacios de color diferentes.

Figura 10. Imagen original e imágenes obtenidas

Se procede a procesar imágenes utilizando operaciones orientadas al pixel, en donde se modifica un pixel a la vez, sin importar el estado de los pixeles vecinos. La transformación se puede aplicar a toda la imagen o a una región de ella. El proceso de transformación en la mayor parte de los casos será de esta forma:

Si x = I[i, j] → y = I'[i, j], donde y = f(x)

Las operaciones orientadas al pixel incluyen copia de una imagen, negativo, filtro corrección de luz, filtros de aclarado y filtros de obscurecimiento.

Se procede a usar operaciones orientadas a la región, en donde se modifica a la imagen un pixel a la vez; pero tomando en cuenta para dicha transformación los pixeles vecinos.

Las operaciones de la región incluyen filtros de enfoque (pasa-altos), filtros de suavizado (pasa-bajos), operaciones geométricas como ampliaciones, reducciones o rotaciones de la imagen.

También se implementan algoritmos para la generación de los principales tipos de ruido que se pueden presentar en una imagen como son el ruido gaussiano y el ruido "sal y pimienta" y de esta manera probar mejor el funcionamiento de los filtros.

Las siguientes figuras muestran los resultados de la aplicación con la tarjeta ADZS-BF561:

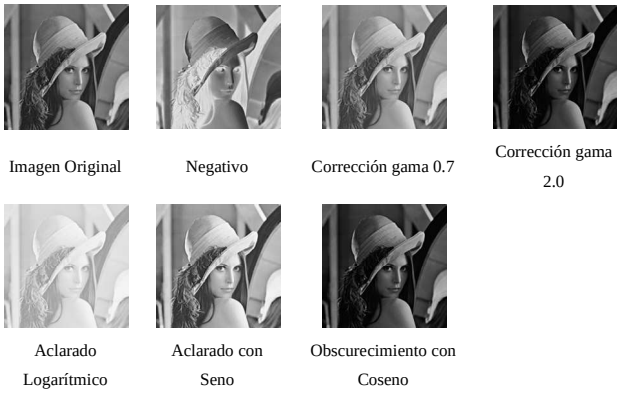

Figura 11a. Operaciones de orientadas al pixel

Figura 11b. Rotaciones

Figura 11c. Ampliaciones y Reducciones

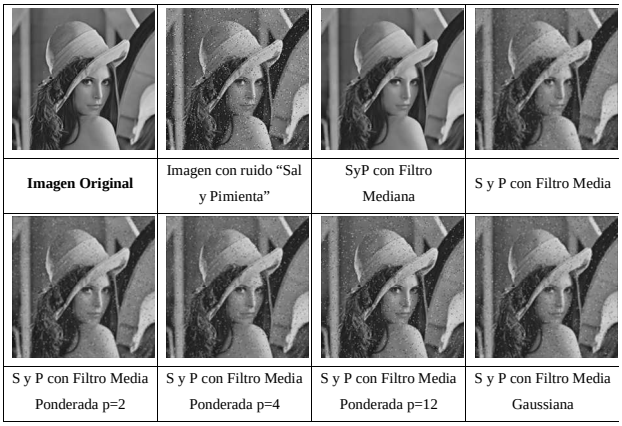

Figura 11d. Aplicación de filtros a imagen con ruido "sal y pimienta"

Figura 11e. Aplicación de filtros a imagen con ruido "gaussiano"

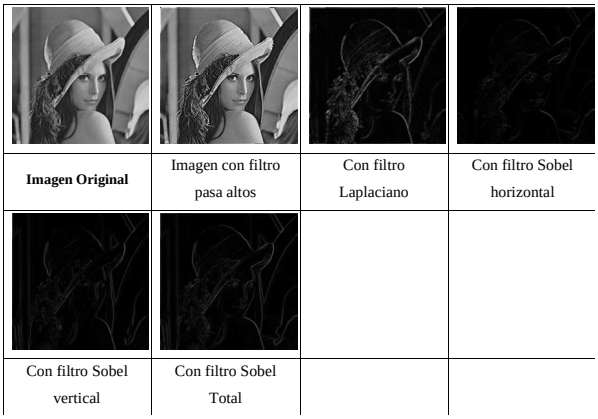

Figura 11f. Aplicación de filtros pasa altos y de detección de bordes

De las figuras 11d y 11e se observa que, el filtro que elimino en mayor medida el ruido "sal y pimienta" fue el filtro "mediana"; mientras que el filtro que eliminó un mayor porcentaje el ruido gaussiano, fue el filtro de media gaussiana [5].

Se verificó el adecuado funcionamiento de la tarjeta mediante la implementación de aplicaciones como por ejemplo, manipulación de LEDS y pulsadores, accesos hacia y desde memoria, grabación de señales de audio, manipulación de imágenes y aplicación de filtros tanto para las señales de audio, como para las imágenes.

El software de desarrollo VisualDSP++ 5.0 es la herramienta principal de programación de la tarjeta ADZS-BF561 y presenta varias utilidades que facilitan la creación, modificación y ejecución de aplicaciones por parte del programador.

El presente estudio de la tarjeta ADZS-BF561, contiene las bases necesarias para el desarrollo de futuros proyectos, gracias a las aplicaciones implementadas, evitando realizar el estudio de la tarjeta desde el inicio y concentrándose en una investigación especifica.

Se constataron las principales ventajas que poseen los DSP, como son la realización de cálculos complejos en tiempo real en tiempos reducidos, debido a que su arquitectura aumenta la capacidad de procesamiento.

Con el desarrollo de las aplicaciones del presente proyecto, se comprobaron las principales capacidades y características que posee la tarjeta ADZS-BF561, constando que las tarjetas de la familia "Blackfin" están orientadas principalmente a aplicaciones multimedia y telecomunicaciones dado que poseen una gran capacidad de cálculo para procesar señales de audio, imagen y video en tiempo real. Además su característica de ahorro de energía, resulta ideal para este tipo de aplicaciones.

REFERENCIAS BIBLIOGRAFICAS

[1] ADSP-BF561 Blackfin Processor Hardware Reference, Revision 1.2, Analog Device, U.S.A., Enero 2010, pp. 47

[2] ADSP-BF561 EZ-KIT Lite Evaluation System Manual, Revision 3.1, Analog Devices, U.S.A., Enero 2007, pp. 1-10 - 1-12

[3] Visual DSP++ 5.0 User's Guide, Revision 3.0, Analog Devices, U.S.A., Agosto 2007, pp. 37 - 43

[4] Diseño de filtros FIR en Matlab. Pagina de Internet www.mathworks.com/help/signal/gs/_f0-35208.html

[5] Gan Woon-Seng, Kuo Sen M., Embedded Signal Processing with the Micro Signal Architecture, Wiley, U.S.A., 2007, pp. 423 – 427

Datos de Contacto

Franklin Andrés Rodríguez Játiva
fandresrj@hotmail.com

Nacido en Quito. Empieza sus estudios formales en la Escuela “Rosario del Alcazar No. 2”, posteriormente estudia el bachillerato en el Colegio Particular La Salle, donde obtiene el título de Bachiller en Ciencias Experimentales. Ingresa a la Escuela Politécnica del Ejército en la cual obtiene el Título de Ingeniera Electrónica y Telecomunicaciones.

Ing. Derlin Morocho Checa
dmorocho@espe.edu.ec

Nacido el 11 de Julio de 1970. Obtuvo el titulo de Ingeniero Electrónico en Telecomunicaciones. Docente a tiempo completo del Departamento de Electrónica. Investigador en el área de Procesamiento Digital de señales y sus aplicaciones. En la actualidad esta cursando un doctorado en Computación y Telecomunicaciones.

Ing. Santiago Puga Benavides
santiago.puga.b@gmail.com

Nacido en Quito el 2 de marzo de 1976. Obtuvo el titulo de Ingeniero Electrónico en Telecomunicaciones. Docente a tiempo parcial del DEEE. Posee una maestría en el Politécnico de Turín-Italia con desarrollo de proyectos de investigación en Motorola Italia.