

BLACKFIN: DSP PARA LA ERA MULTIMEDIA

Jaime Andrés Aranguren Cardona

SanJaaC Electronics, jaac@sanjaac.com, Medellín

Abstract: Se hace una presentación de los novedosos DSPs de la familia Blackfin de Analog Devices. Son procesadores de doble MAC – 16 bits / doble ALU – 32 bits, con aplicaciones embebidas en audio, video y comunicaciones. Se describe la arquitectura, algunas de sus características mas destacadas, sus alcances y beneficios y se presentan algunos *benchmarks* en algoritmos claves de compresión de audio, análisis de imágenes y compresión de video.

Keywords: Blackfin, DSP, Multimedia, Tiempo Real, Desempeño.

1. INTRODUCCIÓN

La familia de DSPs Blackfin de Analog Devices introduce un conjunto de procesadores específicamente orientados a las aplicaciones y mercados multimedia y de telecomunicaciones, las cuales son aplicaciones que requieren de gran capacidad computacional, bajo costo, bajo consumo de potencia, facilidad de uso y facilidad para ejecutar tanto tareas de procesamiento de señal como tareas de control.

Teniendo en mente este tipo de aplicaciones, que requieren de gran capacidad computacional para procesar señales de audio y video en tiempo real, y además usualmente demandan de interfaces gráficas para interactuar con usuarios (GUIs), *stacks* de comunicaciones para acceso a redes / Internet y simultáneamente bajo consumo de potencia para prolongar la duración de las baterías en dispositivos portátiles, se ha desarrollado la familia Blackfin de

DSPs, conocida como BF-53x, la cual ofrece las siguientes características para afrontar los retos planteados: arquitectura *Micro Signal Architecture*, desarrollada en conjunto con Intel, Corp., de doble MAC – 16bits / doble ALU – 32 bits, inicialmente a 350MHz/700MMACS, con desempeños actuales de hasta 600MHz/1200MMACS y con proyección a más de 1GHz; *Dynamic Power Management* que permite controlar tanto el voltaje como la frecuencia internos de operación, y la activación o desactivación selectivas de módulos o periféricos internos; variedad de periféricos *on-chip*, tales como: Controlador SDRAM, PPI con soporte para formatos de video ITU-R 656, puertos seriales (SPORT), controlador DMA de 14 canales, SPI, Timers/PWM, UART, Real-Time Clock, Watchdog Timer, PCI 2.2, USB 1.1.

Los DSPs Blackfin incorporan dentro de un mismo dispositivo ambos tipos de funcionalidad, no como dos *cores* separados, tal como es el caso de la familia *OMAP* desarrollada por Texas Instruments, la cual

incorpora un *core* DSP basado en el TMS320C55xx y un controlador ARM925 dentro de un mismo chip, sino como un único dispositivo con múltiple funcionalidad, evitando así los inconvenientes de la comunicación entre procesadores, el manejo de memorias compartidas, y el uso de herramientas diferentes para procesadores diferentes, con arquitecturas diferentes, los cuales inherentemente resultan en mayores costos, un *time-to-market* más elevado y un más alto consumo de potencia. Para los procesadores Blackfin, con un kernel de RTOS (*Real Time Operating System*, Sistema Operativo de Tiempo Real) las tareas de procesamiento de señal se pueden configurar como las de mayor prioridad y menor latencia, mientras las tareas de control, tradicionalmente desempeñadas por microcontroladores / microprocesadores, se ejecutan como tareas de mejor prioridad y mayor latencia.

En este artículo se hace una descripción de las características de la arquitectura de los DSPs Blackfin, sus alcances y beneficios y se presentan algunos *benchmarks* desarrollados tanto en simulación como en hardware, de algunos algoritmos fundamentales en las áreas de compresión de audio, análisis de imágenes y compresión de video.

2. ARQUITECTURA BLACKFIN – MICRO SIGNAL ARCHITECTURE

En esta sección se describirá la arquitectura interna de los DSPs de la familia Blackfin, describiendo cinco frentes en los cuales se puso el mayor esfuerzo para lograr el balance entre poder computacional, alto desempeño, flexibilidad y facilidad de uso.

2.1. Bloques computacionales – DSP Core.

Los procesadores Blackfin combinan un arquitectura DSP de doble MAC con la facilidad de programación de un microcontrolador tipo RISC y capacidad de procesamiento en modo SIMD (*Single Instruction, Mutiple Data*). El *core* del procesador consta de las siguientes unidades: Unidad Aritmética de Datos, Unidad Aritmética de Direcciones, y Unidad de Control, interconectadas como se ilustra en la Figura 1.

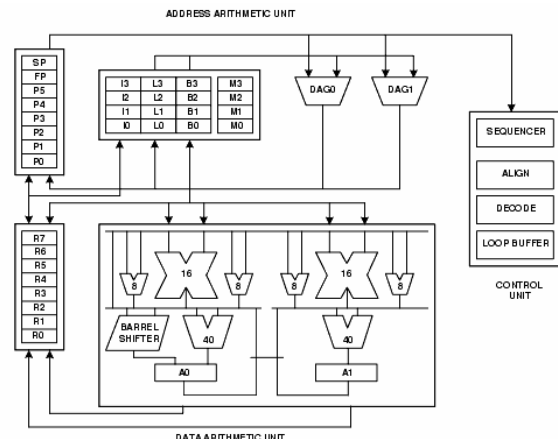


Figura 1. DSP Core.

Unidad Aritmética de Datos. Esta unidad contiene los siguientes bloques: dos MAC de 16 bits, dos acumuladores de 40 bits, cuatro ALUs de video y un *shifter*. Las unidades computacionales pueden procesar datos nativos en 8, 16 o 32 bits desde el archivo de registros. Cada uno de los registros de 32 bits puede ser empleado como un registro doble de 16 bits, para ejecutar operaciones dobles, por ALU, en un mismo ciclo de máquina. En esta unidad se ejecutan todas las operaciones aritméticas y lógicas, además de las operaciones del *shifter*, tales como rotación, normalización, etc. Las ALU de video ejecutan operaciones tipo SIMD en operandos específicos de 8 bits.

Las operaciones primarias de la ALU ocurren en ALU0, mientras las operaciones en paralelo se ejecutan en ALU1 como un subconjunto de las operaciones en la ALU0. Para cada una de las ALU, dos o cuatro operandos de 16 bits resultan en uno o dos resultados de 16 bits; dos operandos de 32 bits resultan en un resultado de 32 bits y tomando como entrada el resultado en 32 bits del multiplicador resulta en una combinación de un resultado en 32 bits del multiplicador con un resultado en 40 bits del acumulador. En resumen, y ejecutando en paralelo las operaciones de la ALU0 y la ALU1 se obtienen cuatro resultados de 16 bits, dos resultados de 32 bits o dos resultados de 40 bits, todo esto en una única instrucción, con ejecución en un único ciclo de máquina.

Como ejemplo, tómese la siguiente instrucción, la cual se ejecuta en un único ciclo:

$R3 = R0 + | + R1, R2 = R0 - | - R1 (S);$

Esta instrucción equivale a las siguientes cuatro instrucciones:

$R3.H = R0.H + R1.H (S);$
 $R3.L = R0.L + R1.L (S);$
 $R2.H = R0.H - R1.H (S);$
 $R2.L = R0.L - R1.L (S);$

La Figura 2 ilustra el proceso ejecutado. Es de mencionar que además de las operaciones aritméticas, en este ejemplo se está ejecutando aritmética saturada, lo que se indica por la (S) en la instrucción.

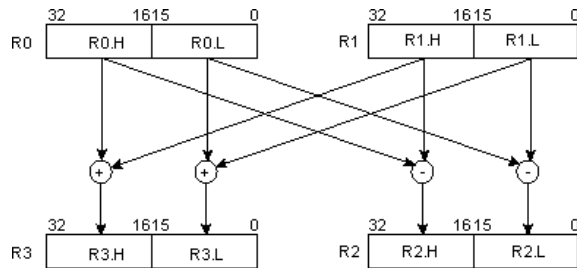


Figura 2. Operaciones ALU.

Cada multiplicador tiene dos entradas de 32 bits, de las cuales se derivan los dos operandos de 16 bits. Cada una de estas entradas puede ser cualquier registro del banco de registros de datos. Los resultados del acumulador se pueden almacenar en el registro acumulador A1 o A0, o alguno de los registros de 32 bits, con redondeo opcional.

La utilización de la doble MAC se ilustra con el siguiente ejemplo, cuya operación se esquematiza en la Figura 3.

$A1 += R1.H * R2.H, A0 += R1.L * R2.L;$

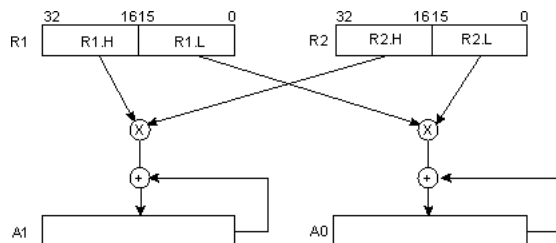


Figura 3. Operaciones MAC.

El *shifter* ejecuta operaciones orientadas a bits sobre operandos de 16, 32 o 40 bits, con resultados de 16, 32 o 40 bits. Las operaciones incluyen desplazamiento lógico, desplazamiento aritmético, rotación, *bit-test*, *bit-set*, empaquetamiento, des-empaquetamiento, detección de exponentes (muy útil en la implementación de algoritmos en punto flotante) y extracción y depósito de campos en registros.

Unidad Aritmética de Direcciones. En esta unidad dos DAGs (*Data Address Generator*, Generadores de Direcciones de Datos) se encargan de generar las direcciones para el doble acceso simultáneo a datos en la memoria del DSP. El par de DAGs comparte un archivo de registros de 32 bits que contienen cuatro conjuntos de archivos de índice, base, modificador y longitud. De manera análoga, existen 8 registros más que pueden ser utilizados como punteros de propósito general y *stack pointer*.

Los DAGs son ampliamente utilizados para operaciones con direccionamiento indirecto; las funciones disponibles en los DAGs son:

- Generación de direcciones con post-modificación.
- Generación de direcciones con *offset*.
- Modificación de direcciones.
- Direccionamiento de bit invertido (*bit reversed*).
- Búferes circulares.
- *Frame* y *stack pointer*.

2.2. Memoria

Los procesadores Blackfin se basan en una arquitectura Harvard modificada en combinación con una estructura jerárquica de memoria. La memoria de nivel 1 (L1) opera a la velocidad interna del DSP sin latencia alguna. En este nivel la memoria de instrucciones almacena únicamente instrucciones, mientras los dos bloques de memoria de datos almacenan únicamente datos y un segmento dedicado de memoria almacena los datos de la pila (*stack*) y variables locales.

Adicionalmente se proveen múltiples bloques de memoria L1, los cuales se pueden configurar como una mezcla de memoria SRAM y cache. La MMU (*Memory Management Unit*, Unidad de

Administración de Memoria) facilita la protección de datos en memoria para tareas individuales que deben mantenerse operando en el *core*, y la protección de acceso restringidos a registros del sistema ante accesos indebidos.

En relación con lo anteriormente expuesto, existen tres modos de operación: usuario, emulación y supervisión. El modo de usuario provee acceso restringido a un grupo de recursos del DSP, facilitando de esta manera la implementación de ambientes protegidos de software en la operación del DSP; los modos de emulación y supervisión tienen acceso no restringido a los recursos del DSP, pero a niveles diferentes, pues están orientados a operaciones diferentes. El modo de usuario es el que se emplea en los programas de aplicación, los otros dos usualmente se reservan para el *kernel* de un sistema operativo.

Los procesadores Blackfin tienen un espacio unificado de direccionamiento de hasta 4 GB. La porción de memoria interna incluye:

- Memoria SRAM de niveles 1 y 2 (L1 y L2).
- Registros mapeados en memoria (MMR, *Memory Mapped Registers*).
- Memoria de arranque de solo lectura (*Bootup ROM*).

En la Figura 4 se ilustra la arquitectura de memoria de los procesadores de la familia Blackfin.

La Tabla 1 resume las capacidades de memoria interna de los procesadores Blackfin.

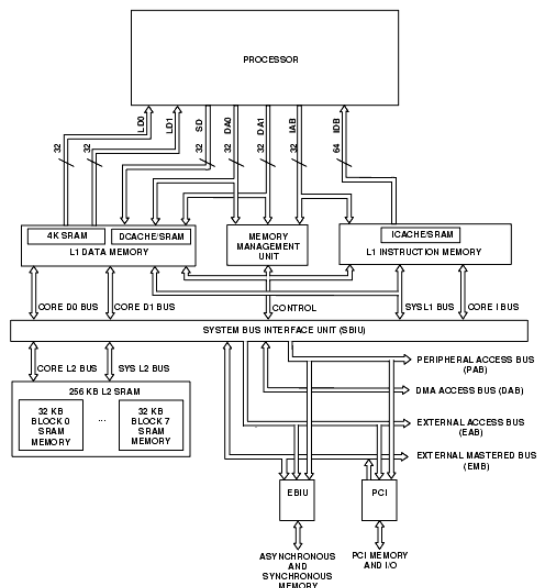


Figura 4. Jerarquía de memoria.

Tabla 1. Capacidad de memoria

Tipo de Memoria	ADSP-BF535	ADSP-BF533	ADSP-BF532	ADSP-BF531
SRAM de Instrucciones / Caché	16 kB	16 kB	16 kB	16 kB
SRAM de Instrucciones	-	64 kB	32 kB	16 kB
ROM de Instrucciones	-	-	32 kB	32 kB
SRAM de Datos / Caché	32 kB	32 kB	32 kB	16 kB
SRAM de Datos	256 kB	32 kB	-	-
SRAM de Datos "Scratchpad"	4 kB	4 kB	4 kB	4 kB
Total	308 kB	148 kB	116 kB	84 kB

La memoria SRAM de nivel 1 (L1) puede ser configurada completa o parcialmente como memoria caché. También cuenta con soporte para memoria externa asincrónica y para SDRAM (*Synchronous Dynamic RAM*) por medio de los controladores *on-chip* correspondientes.

Algunas características destacables de la memoria caché L1 son: arquitectura Harvard modificada que permite hasta cuatro accesos simultáneos al *core* del DSP (traída de una instrucción de 64 bits, carga de dos datos de 32 bits, un almacenamiento "pipelined"

de datos de 32 bits); realización simultánea de operaciones DMA, mantenimiento de caché y accesos al *core*, todas estas funciones operando a la velocidad interna del *core* (CCLK); protección de memoria.

2.3. Pipeline.

Los procesadores Blackfin utilizan un modelo de encauzamiento entrelazado (*intelocked pipeline*) para la ejecución de instrucciones. Cuando se ejecuta una instrucción de carga (*load*) a un registro, éste queda marcado como “ocupado” hasta que se obtenga el valor requerido desde el sistema de memoria. Si una instrucción subsiguiente trata de acceder a éste registro antes de que el nuevo valor esté cargado en él, el encauzamiento se encasilla hasta que se complete la operación de acceso a la memoria. De ésta manera se garantiza que las operaciones que requieren el uso del nuevo valor en memoria no utilicen valores previos o inválidos, incluso aunque haya instrucciones que deben comenzar su proceso de ejecución antes de que el acceso a memoria se haya completado.

Este mecanismo permite la ejecución de instrucciones independientes entre los procesos de carga (*load*) y almacenamiento (*store*) que utilizan el registro al que se escribirá, sin necesidad de que el compilador o el programador conozcan con exactitud el número de ciclos que tomará la operación de lectura y escritura de la memoria. De esta manera, se pueden aprovechar esos ciclos “muertos” para la ejecución de instrucciones no dependientes del resultado de la operación de memoria sin incurrir en errores lógicos o de cálculo, y obteniendo el máximo desempeño del DSP.

La Figura 5 ilustra la *pipeline* de los procesadores BF-53x.

	Inst Fetch 1	Inst Fetch 2	Inst Decode	Address Calc	Ex1	Ex2	Ex3	WB
Inst Fetch 1	Inst Fetch 2	Inst Decode	Address Calc	Ex1	Ex2	Ex3	WB	

Figura 5. Instruction Pipeline.

2.3. Dynamic Power Management

La familia de procesadores Blackfin ofrece

funcionalidad para la administración dinámica de potencia, dicha funcionalidad incluye:

- Manejo de relojes.
- Bucle de enganche de fase (*PLL, Phase Locked Loop*).
- Controlador de Administración Dinámica de Potencia.
- Modos de operación.
- Control de voltaje.

La entrada externa de reloj *CLKIN*, proporciona la frecuencia, ciclo de trabajo y estabilidad necesarias para multiplicar la frecuencia interna de operación del DSP por medio del PLL. La señal que se obtiene opera sobre el *core* del procesador, y se denomina *CCLK*. A continuación se divide la frecuencia de *CCLK* por medio de un valor definido por el usuario, obteniendo la señal de reloj del sistema *SCLK*, la cual opera sobre el Bus de Acceso a Periféricos (*PAB, Peripheral Access Bus*), el Bus de DMA (*DAB*), el Bus de Direcciones Externas (*EAB, External Address Bus*), el Bus Maestro Externo (*EMB, External Master Bus*) y la Unidad de Interfase de Bus Externo (*EBIU, External Bus Interface Unit*). Una característica muy importante y destacable de los procesadores ADSP-BF53x es la posibilidad de modificar las frecuencias tanto del *core* como del sistema, con el fin de optimizar el desempeño y el consumo de potencia.

El Controlador de Administración Dinámica de Potencia (*DPMC, Dynamic Power Management Controller*) permite cuatro modos de operación diferentes, cada uno con sus propias características de desempeño y consumo de potencia. La Tabla 2 resume los modos de operación y sus características.

Tabla 2. Modos de operación DPMC.

Modo de Operación	Ahorro en consumo	PLL		CCLK	SCLK	DMA (5 / 3,2,1)
		Estado	Desv.			
Full On		Hab.	No	Hab.	Hab.	L1, L2/L1
Active		Hab.	Si	Hab.	Hab.	L1, L2/L1
Sleep		Hab.	Dep.	Desh.	Hab.	L2 / -
Deep Sleep		Desh.	-	Desh.	Desh.	- / -

La Figura 6 ilustra las posibles transiciones en los modos de operación para el módulo DPMC, las cuales afectan el consumo de potencia y el desempeño por medio de control en la frecuencia de operación del DSP. Adicionalmente se puede reducir aun más el consumo de potencia al deshabilitar selectivamente el reloj de un periférico específico, cuando este no sea utilizado; esta señal de relo cual puede ser re-habilitada en caso de necesitar nuevamente la operación del periférico en cuestión.

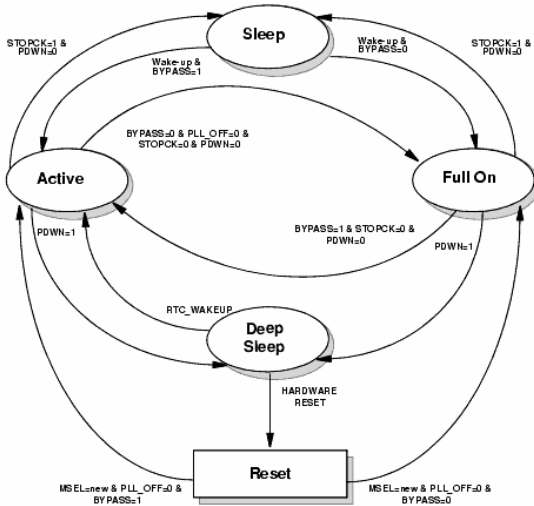


Figura 6. Transiciones en bs modos de operación DPMC.

Adicionalmente al control de la frecuencia de operación, ésta familia de DSPs permite hacer un control dinámico del voltaje de operación del dispositivo, con el fin de optimizar el consumo de potencia según los requerimientos específicos de la aplicación.

El ADSP-BF535 tiene cinco dominios de consumo de potencia; los ADSP-BF531,2,3 tienen tres dominios de consumo de potencia. Cada dominio de potencia tiene su propia fuente de alimentación, que puede variar entre 0.9V y 1.5V. La información para cada tipo de DSP se resume en la Tabla 3.

2.5 Set de instrucciones ampliado

Los procesadores Blackfin agregan a su *set* de instrucciones un grupo de avanzadas características para el procesamiento eficiente de señales de video/

Tabla 3. Dominios de consumo de potencia.

ADSP-BF535		ADSP-BF531/2/3	
Dominio de Potencia	Rango de Vdd	Dominio de Potencia	Rango de Vdd
Lógica interna analógica del PLL	Fijo	Toda la lógica interna excepto el RTC	Variable
Toda la lógica interna, excepto el PLL y el RTC	Variable	I/O (Xtal) y lógica interna del RTC.	Variable
I/O (Xtal) y lógica interna del RTC	Fijo	Toda la demás I/O	Variable
I/O PCI	Fijo	-	-
Toda la demás I/O	Variable	-	-

imagen y el procesamiento vectorizado. A continuación se describirán algunas de las instrucciones más relevantes dentro de este grupo:

Operaciones de video / imagen. Este tipo de instrucciones permite alinear *bytes*, deshabilitar excepciones que se presentan por accesos de 32 bits a memoria sin tener los datos correctamente alineados, y realizar operaciones de suma, resta y promedio en datos dobles o cuádruples de 16 u 8 bits, respectivamente. Algunas de ellas son:

- **ALIGN8, ALIGN16, ALIGN24.** Estas instrucciones copian una palabra no alineada de cuatro *bytes* a partir de una combinación de dos registros de datos. Este tipo de instrucciones son empleadas para el alineamiento de datos previo a instrucciones tipo *SIMD* subsiguientes.
- **DISALGNEXCPT.** Esta instrucción permite deshabilitar excepciones que se pueden presentar al acceder a posiciones de 32 bits en memoria, que se encuentren desalineadas. Normalmente se utiliza previo a instrucciones que involucren operaciones cuádruples sobre datos de 8 bits en modo *SIMD*.
- **BYTEOP3P (Doble suma / recorte a 16 bits).** Esta instrucción suma dos valores de 8 bits no signados con dos valores signados de 16 bits y luego recorta el resultado a 8 bits (entre 0 y 255). Esta instrucción es

supremamente útil en algoritmos de compensación de movimiento en video.

- Doble extracción de acumulador con suma, 16 bits. Esta instrucción suma las mitades superior (bits 31 a 16) e inferior (bits 0a 15) de cada acumulador, y la almacena en un registro de 32 bits. Esta instrucción se emplea frecuentemente en algoritmos de estimación de movimiento en video, en conjunto con la instrucción de Sustracción – Valor Absoluto – Acumulación cuádruple de 8 bits.
- BYTEOP16P. Suma cuádruple de valores de 8 bits (*bytes*).
- BYTEOP1P, BYTEOP2P. Promedio aritmético de cuatro pares de valores de 8 bits (*bytes*). Es una instrucción de gran utilidad en algoritmos de interpolación binaria para compensación de movimiento y búsqueda fraccional de movimientos, en el área de video.
- BYTEOP16M. Sustracción de cuatro pares de valores de 8 bits (*bytes*).
- SAA. (*Subtract – Absolute – Accumulate*). Esta instrucción sustrae cuatro pares de valores, toma el valor absoluto de cada diferencia y acumula el resultado en una mitad de 16 bits de un acumulador. Esta instrucción se emplea en los algoritmos de estimación de movimiento en video que emplean la Suma de Diferencias Absolutas (*SAD, Sum of Absolute Difference*) como métrica de la distorsión.

Operaciones sobre vectores. Este tipo de instrucciones permite ejecutar operaciones sobre grupos de valores de 16 bits simultáneamente, tales como suma, sustracción, negación, búsqueda, etc.

- *Add-On-Sign.* Esta instrucción multiplica el signo de un par de valores de 16 bits (alojados en un registro de 32 bits) por otro par de valores de 16 bits (alojados en un registro de 32 bits). Luego suma las mitades superior e inferior, y almacena el par de resultados de 16 bits en las dos mitades del registro utilizado como destino. Esta instrucción se emplea extensivamente para calcular la métrica de ramificación empleada en las mariposas del algoritmo de Viterbi.

- VIT_MAX (*Compare – Select*). Esta instrucción selecciona los valores máximos de pares de operandos de 16 bits y almacena en una de las secciones acumulador un *footprint* que permite identificar la fuente del máximo valor. Esta es una instrucción clave en la implementación de la función *Add – Compare – Select* para decodificadores Viterbi. Se utiliza en combinación con la instrucción de suma vectorial (*Vector Add*) para el cálculo de mariposas Trellis.
- Suma / Resta / Multiplicación / MAC vectorial. Estas instrucciones simultáneamente suman y / o sustraen, o multiplican o ejecutan una operación MAC, sobre dos pares de valores, almacenando el resultado como en registros de 32 bits o en mitades como valores de 16 bits.
- Máximo / Mínimo vectorial. Compara las mitades superiores de los operandos y almacena el mayor / menor de esos valores en la mitad superior del registro destino, simultáneamente ejecutando la misma operación para las mitades inferiores de los registros en cuestión.
- Búsqueda vectorial. Esta instrucción compara dos valores de 16 bits con valores almacenados en el acumulador. Luego actualiza condicionalmente el acumulador y un puntero de destino, basado en los resultados de la comparación. Dentro de un ciclo, esta instrucción se emplea para hallar el máximo o mínimo o un valor específico dentro de un arreglo.

Adicionalmente se pueden ejecutar algunas instrucciones en paralelo, basados en el esquema de una instrucción de 32 bits para la ALU / MAC y un par de instrucciones de 16 bits. A continuación, un ejemplo:

```
/* Subtract-Absolute-Accumulate issued in parallel with
the memory
access instructions that fetch the data for the next SAA
instruction. This sequence is executed in a loop to flip-
flop
back and forth between the data in R1 and R3, then the
data in R0
and R2. */
saa (r1:0, r3:2) || r0=[i0++] || r2=[i1++] ;
saa (r1:0, r3:2)(r) || r1=[i0++] || r3=[i1++] ;
mnop || r1 = [i0++] || r3 = [i1++] ;
```

2. MERCADOS Y APLICACIONES

Todos los miembros de la familia comparten el mismo *core*, diferenciándose en su capacidad de memoria *on-chip*, en su velocidad máxima de operación, en las características de control de consumo de potencia y en los periféricos integrados, lo cual se traduce en un balance - precio desempeño que permite incluir estos procesadores tanto en equipos electrónicos de consumo, a bajo costo, como en equipos de infraestructura de telecomunicaciones y multimedia del más alto desempeño.

Las diferencias en capacidad de memoria y modos de consumo de potencia ya se han reseñado y especificado en las secciones previas, por tanto no se incluirán en las secciones subsiguientes.

3.1. BF-535

Este fue el primer procesador de la familia, lanzado al mercado en junio del 2001, como resultado de un desarrollo denominado *Micro Signal Architecture* en conjunto con Intel, Co. en un programa que inició a principios de 1999.

Características.

- 300 MHz / 600 MMACs.
- Puerto PCI 2.2, 32 bits, 33 MHz, 3.3V Maestro / Esclavo.
- Puerto USB 1.1.
- Dos UARTs, una de ellas con soporte para IrDA.
- Dos puertos SPI.
- Dos SPORTs.
- Cuatro Timers.
- Puerto de emulación JTAG.
- Encapsulado PBGA de 260 pines.

Mercados y aplicaciones. Con su conjunto de periféricos e interfaces integrados, el ADSP-BF535 se convierte en una solución tipo *system-on.chip* para aplicaciones de próxima generación en las áreas de *networking*, procesamiento de imágenes e infraestructura de telecomunicaciones.

- Internet e infraestructura de telecomunicaciones. El procesamiento de señales con alta desempeño y eficiencia en el consumo de potencia es la clave para la

convergencia del tráfico de voz, fax, datos y video sobre la Internet; estos factores son fundamentales en la implementación de *gateways* de Voz sobre IP (VoIP) y Servidores de Acceso Remoto (RAS, *Remote Access Servers*). Los *gateways* VoIP permiten el tráfico de información de voz a través de redes de datos por paquetes; los RAS permiten el tráfico de datos y video a través de redes conmutadas. Aunque con orientaciones diferentes, estos dos tipos de aplicaciones tienen un criterio de diseño común: maximizar la densidad de canales en un área determinada de circuito impreso. El ADSP-BF535 permite cuadruplicar la densidad de canales manejados respecto a un procesador de otro tipo. Adicionalmente, sus características de MCU RISC permiten la eliminación de un procesador externo para ejecutar tareas de detección / corrección de errores y funciones de control.

- Dispositivos / herramientas para internet / *networking*. Se espera que Internet siga creciendo como un importante medio de distribución de información combinada de voz, datos y video, por medio de una gran variedad de productos que se conectan a Internet, transfieren y procesan datos y ofrecen esta información al usuario final. Entre este tipo de productos se encuentran PDAs Inteligentes, *screenphones*, dispositivos de juegos sobre Internet, *web phones / terminals*, *NetTV*. Gran cantidad de estos y otros productos similares están diseñados para ser portátiles, lo cual implica que el consumo de potencia es un factor clave en el diseño y aceptación del producto; el ADSP-BF535 incorpora las funciones de control dinámico de consumo para ofrecer soluciones a estos retos de diseño. Adicionalmente, las avanzadas interfaces de comunicación (PCI, USB) hacen del ADSP-BF535 un procesador apropiado para dispositivos *wireless LAN*, *gateways* residenciales, etc.

3.2. BF-531, BF-532, BF-533.

Este grupo de miembros de la familia Blackfin ofrece

algunas características adicionales, como más alto desempeño (600 MHz / 1200 MMACs, BF533) y bajo costo (US\$ 4.95) y algunos periféricos especiales para establecer comunicación con *codecs* de video a través de interfaces ITU-R 656, además de DMA bidimensional mejorada para aplicaciones de procesamiento de video e imágenes.

Características

- 600 MHz / 1200 MMACS (BF533) ; 400 MHz / 800 MMACs (BF531, BF-532)
- PPI/GPIO con soporte a interfaces ITU-R 656.
- Dos SPORTs, con soporte para ocho canales I²S.
- 12 canales DMA.
- Tres canales PWN/Contador con soporte para PWM.
- UART con soporte para IrDA.
- Encapsulado PBGA de 160 pines, o LQFP de 176 pines.

Mercados y aplicaciones.

- Procesamiento digital multimedia. Los datos de imagen y video se han convertido en los medios principales de comunicación, dando un mayor alcance a la información escrita y hablada. Los recientes desarrollos en las técnicas de compresión de datos han acelerado la transición de video analógico a video digital, reduciendo los costos del procesamiento, la transmisión y almacenamiento de video en formato digital. Esta reducción de costos hace que el video digital se convierta en pieza clave de nuevos productos, tales como las cámaras digitales, las videograbadoras digitales, dispositivos de entretenimiento para el hogar, seguridad automotriz y equipos de seguridad / vigilancia.
- Dispositivos portátiles de información. El deseo y la necesidad de nuestra sociedad por un rápido y fácil acceso a la información de voz, datos, imágenes y video, continúan en constante aumento. Los dispositivos para acceso, bien sea por cable o inalámbrico, a esas fuentes de información, son dispositivos de mano alimentados por baterías, para los cuales es de suma importancia tener un excelente control del

consumo de potencia, el cual se realiza en los procesadores Blackfin por medio de las técnicas DPMC explicadas en el numeral 2.4. Las múltiples interfaces para ADCs hacen de los procesadores Blackfin aptos para aplicaciones tales como equipos portátiles de prueba y equipos de instrumentación médica e industrial.

- Entretenimiento y telemática automotriz. Muchos factores hacen que cada vez más se empleen técnicas de telemática y entretenimiento digital en automotores. Algunos de ellos tienen relación con disposiciones legales y de seguridad, como el uso de teléfonos celulares “manos libres”, otros provienen de los avances en tecnologías de presentación de video, y otros más encuentran sus raíces en la necesidad del consumidor de ser más “productivo” en el automóvil. La arquitectura Blackfin permite implementar eficientemente, a bajo costo, y con reducido consumo de potencia, algoritmos claves para este tipo de aplicaciones, tales como cancelación de ruido / eco, descompresión de audio y video y reconocimiento de voz. Las características de microcontrolador tipo RISC embebidas en los DSPs Blackfin, sus capacidades de DMA de 12 canales, puertos seriales sincrónicos y asincrónicos, controlador de SDRAM, PPI, *watchdog timer*, RTC (*Real Time Clock*) y temporizadores de propósito general, hacen de los procesadores ADSP-BF531/2/3 apropiados para el diseño de dispositivos que incluyen módulos tales como GPS, terminales celulares, CD, DVD y Bluetooth.

4. BENCHMARKS

Resumen de pruebas ejecutadas en simulador y en hardware de algunos algoritmos críticos en las áreas de compresión de audio, análisis de imágenes y compresión de video.

4.1. Compresión de audio.

La MDCT (*Modified Discrete Cosine Transform*, Transformada Coseno Discreta Modificada) en sus

versiones directa e inversa, es uno de los bloques mas importantes dentro del proceso de codificación / decodificación de señales de audio, especialmente en formatos como MP3 y MP3Pro. Es básicamente la cuantización de los componentes obtenidos como resultado de esta transformación en su versión directa (MDCT), de acuerdo a ciertos parámetros perceptuales, la información que finalmente se transmite en un flujo de bits MP3 en el proceso de codificación o compresión; el proceso de decodificación de audio en formato MP3 (y similares) consiste fundamentalmente en la reconstrucción de la señal de audio a partir de los componentes MDCT cuantizados, por medio de la IMDCT.

El algoritmo mas eficiente reportado hasta la fecha en la literatura especializada, para la implementación en tiempo real de la MDCT / IMDCT se debe a Rao y Britanak. En la Figura 7 se ilustra el diagrama de flujo para la MDCT-12, la cual es la versión utilizada dentro de la codificación MP3 para bloques cortos.

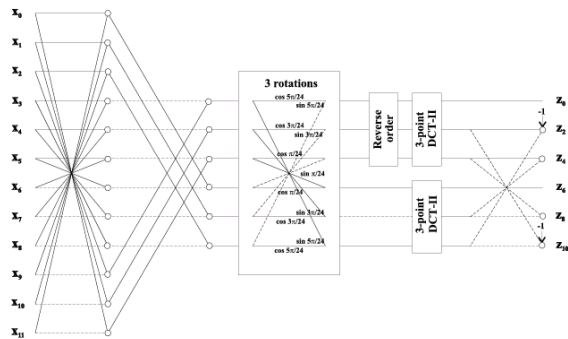


Figura 7. MDCT-12

Los tiempos de ejecución de la subrutina desarrollada para este algoritmo, y junto con otras relacionadas con la MDCT / IMDCT, para los procesadores ADSP-BF535 y ADSP-BF533 están resumidos en la Tabla 4.

Tabla 4. MDCT / IDCT en BF-53x.

Algoritmo	Ciclos	BF-535 (300 MHz)	BF-533 (600 MHz)
MDCT-12	54	180 nS	90 nS
MDCT-36	239	796.66 nS	398.33 nS
IMDCT-12	52	173.33 nS	86.66 nS
IMDCT-36	258	860 nS	430 nS

4.2 Análisis de imagen.

Dilatación. Este es un operador morfológico del procesamiento de imágenes, empleado para imágenes binarias o en escala de grises, cuyo efecto es “agrandar” los bordes de regiones con píxeles de tipo “anterior” (usualmente de color blanco, o de valor 1, o 255 en escala de grises de 8 bits). Por tanto, las regiones con píxeles de tipo “anterior” crecen en área, mientras que los huecos dentro de estas regiones decrecen en área.

Para este operador se necesitan dos conjuntos de datos de entrada, a saber, la imagen original y otro conjunto de puntos conocido como *elemento de estructura*; este ultimo es precisamente el que determinara el efecto del operador de dilatación. Sea X el conjunto de coordenadas euclidianas correspondientes a la imagen de entrada binaria, y K el conjunto de coordenadas del elemento de estructura; denótese con Kx la translación de K , tomando a x como el origen. La dilatación de X con K es entonces el conjunto de todos los puntos x tales que la intersección de Kx con X no es diferente del conjunto vacío.

Se implemento una subrutina en lenguaje ensamblador de los DSPs de la familia BlackFin para ejecutar la erosión de una imagen binaria de tamaño dado, con un elemento de estructura de 3x3.

La metodología de calculo es la siguiente, la cual permite explotar el paralelismo en las instrucciones de los procesadores de la familia BlackFin. Supóngase que el elemento de estructura (mascara) tiene la siguiente configuración:

	h00	h01	h02	
	h10	h11	h12	
	h20	h21	h22	

Y que la entrada en consideración es la siguiente (como ejemplo, de 3x3):

	x00	x01	x02	
	x10	x11	x12	
	x20	x21	x22	

De esta manera, el elemento de salida en la posición (1,1) es el siguiente:

```

Y11 == ((x00&h00) | (x01&h01) | (x02&h02)
        | (x10&h10) | (x11&h121)
        | (x12&h12) | (x20 &h20) | (x21&h21) | (x22
        &h22))

```

Se corrieron pruebas sobre las subrutinas, con pequeñas imágenes de entrada de 8x8 y 8x16, con los resultados resumidos en la Tabla 5.

Tabla 5. Desempeño Dilatación.

Imagen de entrada	Ciclos	BF-535 (300 MHz)	BF-533 (600 MHz)
8 x 8	753	2.51 μ S	1.255 μ S
8 x 16	1449	4.83 μ S	2.415 μ S

Detección de bordes. Una de las metodologías utilizadas para la detección de bordes en imágenes es el uso de operadores gradiente, los cuales se representan por un par de mascarar, H_1 y H_2 , las cuales miden el gradiente de la imagen $u(m,n)$ en dos direcciones ortogonales. La magnitud del vector gradiente y su dirección están dadas por:

$$g(m,n) = \sqrt{g_1^2(m,n) + g_2^2(m,n)}$$

$$q_g(m,n) = \tan^{-1} \left(\frac{g_2(m,n)}{g_1(m,n)} \right) \quad (1)$$

Usualmente, la magnitud se calcula como:

$$g(m,n) \triangleq |g_1(m,n)| + |g_2(m,n)| \quad (2)$$

El píxel con coordenadas (m,n) se asume como un borde si $g(m,n)$ supera algún umbral t .

Las mascarar horizontal y vertical utilizadas para el algoritmo de detección de bordes por medio de gradientes con el operador de Sobel, son respectivamente (* indica el punto origen de la mascarar):

$$\begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0^* & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \quad \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0^* & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Se implemento una subrutina en lenguaje ensamblador de los procesadores BlackFin para ejecutar la el algoritmo de detección de bordes por medio de gradientes. Dicha subrutina bien podría ser empleada para ejecutar esta tarea con diferentes operadores de mascarar 3x3, como el operador Prewitt, por ejemplo.

La Tabla 6 resume los resultados de desempeño del algoritmo de detección de bordes, con operador Sobel, para imágenes de diferentes tamaños.

Tabla 6. Desempeño Sobel.

Imagen de entrada	Ciclos	BF-535 (300 MHz)	BF-533 (600 MHz)
8 x 16	4071	13.57 μ S	6.785 μ S
32 x 32	33224	110.746 μ S	55.373 μ S

4.3. Compresión de video.

Una de las funcionalidades claves del estándar MPEG-4 para la compresión de video es la capacidad de poder codificar objetos de video con formas arbitrarias diferentes a las cuadros de video tradicionales de forma rectangular. Dado que múltiples objetos de video se pueden codificar independientemente de los demás y se pueden además multiplexar en un único flujo de datos, se ofrece al usuario la posibilidad de componer una escena de video a su propio antojo colocando objetos de video decodificado en cualquier posición arbitraria. Para poder reconstruirse dentro de esquemas de codificación híbridos basados en bloques, cada plano de objetos de video es embebido dentro de un rectángulo con una estructura de bloque regular. Aquellos bloques que estén completamente contenidos dentro del objeto de video se codifican con una DCT tradicional de 8x8, mientras que aquellos bloques que no están completamente contenidos dentro del objeto de video son codificados separadamente utilizando bien sea técnicas de relleno de imagen o transformadas adaptativas a la forma.

Dentro de las transformadas adaptativas a la forma del objeto a codificar, ha sido de gran relevancia para el tema de la codificación de imágenes basada en objetos la Transformada Discreta de Coseno Adaptativa a la Forma (*SADCT, Shape Adaptive Discrete Cosine Transform*), por dos razones básicas: supera a las técnicas de relleno de imagen en términos de eficiencia de codificación para altas tasas de bits, y además es separable en filas y columnas, lo que implica una carga computacional baja.

La idea básica de la SA-DCT es transformar un objeto de forma arbitraria por medio de transformadas en cascada, por filas y por columnas, manteniendo de esta manera la característica bien conocida de separabilidad, de las transformadas basadas en bloques. Sea la matriz DCT $\mathbf{A} = \{a_{jk}\}$ con:

$$a_{ik} = g \cos \frac{(2i+1)pk}{2N}; j, k = 0, \dots, N-1 \quad (3)$$

$$k = 0 \Rightarrow g = \frac{\sqrt{2}}{2}; g = 1 \forall k \neq 0 \quad (4)$$

Cada columna de los píxeles de imagen \mathbf{x}_j se alinean primero horizontalmente de acuerdo a la regla:

$$\mathbf{c}_j = \frac{4}{N} \mathbf{A} \mathbf{x}_j$$

Los coeficientes resultantes están, por tanto, alineados verticalmente, y cada fila es sometida a una segunda transformada horizontal siguiendo la regla anterior de transformación. Para la SA-DCT inversa (*ISADCT*), se efectúan los pasos respectivos, aplicando como regla de transformación la siguiente:

$$\mathbf{x}_j = \frac{1}{2} \mathbf{A}^T \mathbf{c}_j$$

Los factores de escala en la ecuación de la SA-DCT garantizan que cada coeficiente de DC obtenido a partir de la transformación vertical sea proporcional al valor medio de la columna correspondiente, y su rango de amplitud independiente de la longitud de la columna N y en el mismo rango que una DCT de orden 8. Por tanto, la subsiguiente transformación

horizontal de los valores DC de todas las columnas produce un único coeficiente de DC en el caso de un segmento de imagen monocromática. Factores de escala diferentes en dicha ecuación resultarían en coeficientes adicionales de alta frecuencia que simplemente representan la forma del segmento de imagen.

Se implementó una subrutina en lenguaje ensamblador, para la función SA-DCT sobre matrices 8x8, tal como lo establece el estándar MPEG-4, aprovechando las características de los procesadores Blackfin. La Tabla 7 resume el desempeño de la subrutina implementada.

Tabla 7. Desempeño SA-DCT.

Algoritmo	Ciclos	BF-535 (300 MHz)	BF-533 (600 MHz)
SA-DCT	1904	6.346 μ S	3.173 μ S

5. CONCLUSIONES

Se presentaron los procesadores digitales de señal de la familia Blackfin desarrollada por Analog Devices. Son poderosos DSPs de punto fijo, concebidos para lograr alto desempeño en aplicaciones multimedia y de telecomunicaciones, a la vez que facilitando el diseño del sistema por su gran integración con periféricos y espacio de memoria interna, y prolongando la vida útil de la batería en dispositivos portátiles.

Algunas de estas características se ilustraron por medio de la simulación en software e implementación en hardware de modernos algoritmos clave en aplicaciones multimedia.

RECONOCIMIENTOS

Agradecimientos especiales al Dr. Singaram Jayakumar y su equipo técnico de Epigon Media Technologies, Inc., en Bangalore, India, por su orientación y contribución en la implementación de los algoritmos de *benchmarking* para los procesadores Blackfin.

Agradecimiento también al personal del grupo DSP en Analog Devices, Inc., por facilitar documentación y

material (hardware y software) para el desarrollo del presente trabajo.

REFERENCIAS

- Analog Devices (2002). *ADSP-21535 Blackfin DSP Hardware Reference*. Norwood, Mass.
- Analog Devices (2002). *Blackfin DSP Instruction Set Reference*. Norwood, Mass.
- Aranguren Cardona, J. A. (2002). *Codificador MP3 en Tiempo Real Implementado en Hardware*". VII STSIVA, Bucaramanga.
- Britanak, V. y Rao, K. R. (2001) An efficient implementation of the forward and inverse MDCT in MPEG audio coding. En: *IEEE Signal Processing Letters*. Vol. 8, No. 2. p. 48 – 51.
- Gonzalez, R. y Woods, R. (1996) *Tratamiento Digital de Imágenes*. Addison-Wesley. Madrid.
- Jain, A. (1989) *Fundamentals of Digital Image Processing*. Prentice Hall. Englewood Cliffs.
- Koenen, R.; Pereira, F y Chiariglione, L (1997). *MPEG-4: Context and Objectives*. Signal Processing: Image Communication. Vol. 9, no. 4, pp. 295 – 304.