

Plataforma flexible de procesamiento de señales para el aprendizaje de sistemas basados en DSP

Antonio F. Díaz, Carlos G. Puntonet, Begoña del Pino, Antonio Cañas

Departamento de Arquitectura y Tecnología de Computadores

Universidad de Granada

18071 Granada

Email: {afdiaz, carlos, bego, acanas}@atc.ugr.es

Resumen

El uso de DSP (Digital Signal Processor) para procesamiento de señales en tiempo real va en progresivo aumento, aplicaciones no sólo en el ámbito del audio o las telecomunicaciones, sino en elementos de seguridad activa en vehículos o en el control de potencia en ascensores donde se utilizan variadores de frecuencia para el funcionamiento suave y preciso de los motores.

En este artículo se presenta la plataforma desarrollada para el estudio de DSP, fácil de utilizar y orientada a las necesidades en la enseñanza de dichos sistemas, donde el alumno pueda aprender todos los elementos necesarios para desarrollar un sistema basado en DSP así como evaluar distintos algoritmos y ventajas e inconvenientes que presentan éstos en una implementación real.

1. Introducción

El estudio de los DSP plantea una serie de retos que son abordados en simposios específicos de IEEE como EDERS-2004 (The European DSP Education and Research Symposium), o bien en revistas especializadas [7]. Se puede considerar muy aconsejable la enseñanza de los DSPs en perfiles de informática orientados principalmente a las áreas de arquitectura de computadores y tratamiento de señales.

El aprendizaje completo de los sistemas con DSP no sólo debe quedar en una exposición teórica de los principales fundamentos sino que requiere un componente importante asociado a las prácticas.

Actualmente existen sistemas de evaluación con DSP pero no acaban por adecuarse al ámbito docente. Si se buscan sistemas sencillos resultan muy genéricos y no permiten desarrollar prácticas algo más complejas. Si se buscan más completas

pasan a ser excesivamente sofisticadas y difíciles de utilizar puesto que éstos están principalmente orientados a la industria y al desarrollo donde prima más el carácter de herramientas muy potentes, con complejas capacidades de depuración pero no orientadas a un entorno didáctico.

Una vez evaluadas las distintas alternativas que pueden utilizarse con DSP para plantear un sistema de prácticas, se ha optado por el desarrollo de una nueva plataforma que permita evaluar de forma práctica diversos aspectos relacionados con estos procesadores y el tratamiento de señales.

Aunque un DSP tiene un núcleo basado en un procesador, realmente presenta una gran cantidad de elementos que lo diferencian y hacen dificultoso su comprensión. Igual que orientar el estudio de un microcontrolador de forma exclusivamente teórica conduce a un vacío y a un desconocimiento de los estudiantes sobre que se puede hacer con éstos, puede ocurrir que un estudio poco práctico con los DSP produzca igualmente una sensación al alumnado de “pérdida de tiempo”.

Los DSP son incluso más complejos pues se suele recurrir a cantidad de “trucos” para la optimización del código en la implementación de algoritmos a la arquitectura. Modos de direccionamiento poco habituales como búferes circulares o con acarreo invertido requieren una especial atención para no utilizarlos de forma incorrecta. Igualmente instrucciones que realizan múltiples operaciones simultáneamente no sólo aritméticas sino que afectan a punteros o a bucles deben escogerse con atención para conseguir un código plenamente optimizado. Se pueden conocer la arquitectura interna del procesador, pero se requiere de unas prácticas adecuadas que ofrecen una visión más real.

Las prácticas deben englobar tanto aspectos relacionados con el tratamiento de señales como la arquitectura interna del procesador y su

optimización para la implementación de algoritmos.

También se pueden diseñar laboratorios virtuales para la evaluación de estos sistemas, pero la experiencia muestra que es muy recomendable combinarlos con un contacto más directo con sistemas reales. Para el tratamiento de señales, Matlab es una herramienta muy potente y completa para la simulación de sistemas. Su "toolbox" de tratamiento de señales permite el diseño y simulación de filtros, FFT, DCT, procesamiento complejo de señales, visualización de resultados, espectrogramas, etc.

El estudio de la arquitectura de un DSP debe basarse en un sistema que no sea inicialmente excesivamente complejo, pero que permita abordar problemas en tiempo real de forma razonable mediante prácticas de una complejidad media.

Partiendo de la experiencia previa de otros DSP utilizados en prácticas en años anteriores y en proyectos fin de carrera como la implementación del estándar europeo de codificación de voz, GSM 06.10 en un sistema en tiempo real basado en el DSP TMS320C26 [4], se han localizado ciertos elementos interesantes que se han tenido en cuenta en el desarrollo de esta nueva plataforma.

Resulta adecuado utilizar un DSP de punto flotante para analizar los distintos efectos de la precisión aritmética utilizada para el procesamiento numérico. Este debe disponer memoria interna y potencia suficiente para tratar señales con frecuencias de muestreo del orden de KHz.

Los sistemas con DSP suelen ser "cajas negras" que procesan señales y no puede observarse su dinámica interna. En este sentido, un elemento que puede facilitar la motivación es incluir una interfaz más "amigable" no sólo para el usuario final, sino también para el desarrollador, que en este caso son los estudiantes confeccionando sus prácticas.

Se podrían plantear unas prácticas alternativas para el tratamiento de señales basada en ordenadores portátiles, sin utilizar DSP, pero presentan ciertos aspectos negativos:

- Coste elevado.
- Autonomía limitada a 2 -3 horas en sistemas autónomos.
- Peso excesivo.
- Tiempos de inicio / parada.

Métodos innovadores aplicados a distintas disciplinas

- Interfaz de usuario no adaptada a las necesidades específicas.
- No muestra una visión real de una implementación portátil como pueden ofrecer un DSP.

No obstante los procesadores actuales como la familia Pentium o AMD cada vez incorporan repertorios de instrucciones orientados a multimedia que resultan muy atractivos para la implementaciones de algoritmos asociados a tratamiento de señales.

Los DSP tienen un lugar bien distinto, puesto que se orientan a diseños donde:

- Se requiere un sistema de bajo coste.
- Posibilidad de ser autónomo.
- Se pueden abordar procesamiento de señales con algoritmos complejos que plantean serias dificultades a los microcontroladores convencionales de bajo coste.

Un aspecto muy motivador para el aprendizaje de los DSP es buscar aplicaciones prácticas que el alumno pueda desarrollar con cuestiones actuales. Hace algún un tiempo desarrollamos unas prácticas utilizando un DSP para decodificación de sonido encriptado [2]. Siguiendo esta línea y tratando de actualizar los sistemas utilizados para el estudio de DSP se han completado nuevas prácticas que se comentan posteriormente.

El principal objetivo de esta plataforma es el facilitar el estudio de los DSP y su utilización para el tratamiento de señales en tiempo real.

2. Descripción del sistema

El sistema propuesto permite implementar equipos autónomos con procesamiento avanzado de señales en tiempo real. Las principales características de éste son:

- Sistema basado en DSP.
- Completamente autónomo.
- Interfaz gráfica de usuario, pantalla LCD.
- Mandos giratorios y botones.
- Sistema de entrada/salida (Analógicas y Digitales).
- Posibilidad de aumentar el número de entradas y salidas.

2.1. Hardware del sistema

Se parte de un sistema de evaluación de Analog Devices muy interesante, el EZ-LITE [1].

Este kit integra algunos elementos como son:

- DSP ADSP-21061 .
 - Codec AD1847 .
 - Interfaz básica de Entrada / Salida.
- El elemento principal es el DSP de Analog Devices ADSP-21061 que ofrece las siguientes características:
- Super Harvard Architecture Computer (SHARC).
 - Procesador de 32 bits orientado a aplicaciones DSP de altas prestaciones.
 - Procesador de punto flotante con aritmética IEEE de 32 bits, extendida de 40 bits y formatos de datos flexibles.
 - 120 MFLOPs de pico, y 80 MFLOPs sostenidos.
 - Memoria interna: 128 Kbytes configurable.
 - Direccionamiento externo: 4 Gwords.
 - 6 canales de DMA.
 - Arquitectura escalable multiprocesador.
 - Velocidad de transferencia externa: 300 Mbytes/s.

El DSP es el núcleo del sistema pero requiere de una serie de elementos que lo complementan para poder disponer de un equipo operativo. Para la captura de datos en tiempo real se utiliza el CODEC AD1847 que incluye como entrada un convertor sigma-delta de 16 bits de dos canales que permite muestrear hasta 48 Kmuestras/segundo, y simultáneamente como salida otros dos canales a la misma frecuencia de muestreo.

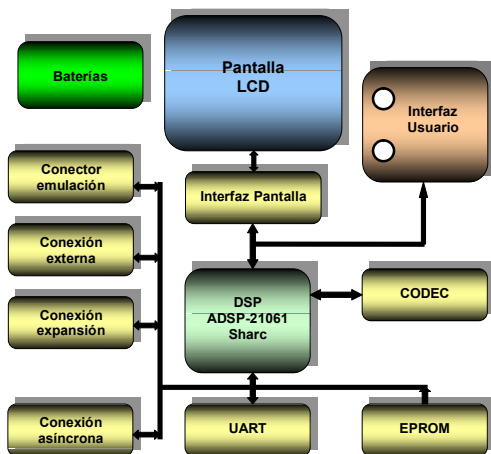


Figura 1. Diagrama general del sistema

Una de las novedades que incluye la plataforma desarrollada es incluir una pantalla LCD controlada directamente por el DSP lo que permite analizar los datos de forma directa.

La pantalla LCD de Hitachi es la LM6733 con 320 x 240 pixel. Esta pantalla es muy interesante para poder observar las señales en tiempo real y contrastar los parámetros obtenidos con los algoritmos geométricos.

La Figura 1 muestra un diagrama general del sistema. En la Figura 2 puede observarse el aspecto del prototipo desarrollado.



Figura 2. Prototipo desarrollado

La interfaz de usuario está compuesta de 2 mandos sin fin, codificados angularmente, para realizar funciones de movimientos ilimitados como desplazamientos. 4 botones permiten el acceso a menús y funciones rápidas.

El sistema puede operar con batería e implementar un sistema completamente autónomo lo que se puede aproximar a modelos como puede ser el de la telefonía móvil digital.

2.2. Herramientas de programación

La programación de DSP se realiza generalmente en C o ensamblador. En el caso de los procesadores convencionales, suelen ser muy limitados los casos en los que se requiere la programación en ensamblador, pero en el caso de los DSP, a veces es necesario para conseguir un código plenamente optimizado a la arquitectura.

La forma más usual de desarrollar aplicaciones para DSP es escribir el código en C y utilizar el ensamblador para escribir las rutinas que deban estar optimizadas.

Esta plataforma desarrollada no sólo incluye elementos hardware sino también elementos software para facilitar su uso.

Se incluyen funciones que facilitan el acceso al nuevo hardware como son las relacionadas con el acceso a la pantalla gráfica, trazado de líneas, escritura de texto, captura de bitmaps, y acceso a la entrada de botones y señales. Estas funciones se han escrito en ensamblador para optimizar al máximo el funcionamiento, minimizar el consumo adicional de CPU.

El kit que se utiliza como base de esta plataforma incluye para su la programación un conjunto práctico de herramientas, ensamblador, compilador de C y el entorno de depuración, Visual DSP.

Éste está compuesto de dos herramientas básicas, un simulador y un emulador. Como primer paso para la depuración, el simulador permite verificar algunas partes de código sin necesidad de disponer de la plataforma.

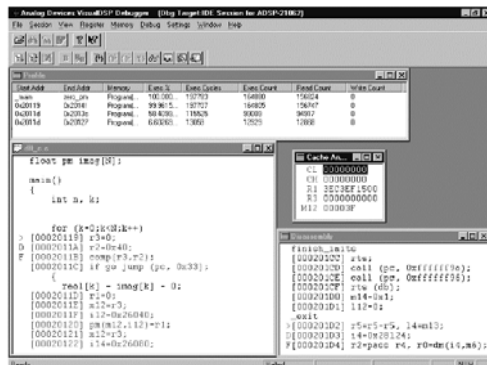


Figura 3. Visual DSP

El emulador permite la carga de programas en la plataforma para poder ejecutar en tiempo real. Adicionalmente, se incluye un conjunto de funciones que lo hace incluso más práctico pues éstas ya están optimizadas en ensamblador y simplifica inicialmente la labor de aprendizaje. También permiten desarrollar un sistema complejo que pueda funcionar en poco tiempo.

Además de las funciones matemáticas “básicas” que podemos encontrar en bastantes implementaciones de C, (funciones trigonométricas, exponenciales y logaritmos), también podemos encontrar un conjunto de funciones implementadas en ensamblador muy prácticas como son filtros FIR, IIR, procesamiento

Métodos innovadores aplicados a distintas disciplinas

FFT y estadística como medias, varianzas y valores rms.

Además, se ha tratado de homogeneizar el funcionamiento interno de las aplicaciones de forma que el código puede adaptarse fácilmente a la plataforma.

Para poder desarrollar código internamente se divide en dos partes:

- Tiempo real.
- Pseudo tiempo real.

El código de tiempo real generalmente está asociado con las interrupciones y tiene que ser código reducido o bien requieren de un procesamiento, en el dominio del tiempo, por cada muestra como el caso de los filtros FIR e IIR.

El código de pseudo tiempo real permite poder combinar la ejecución de algoritmos y el tiempo restante con la visualización de señales en tiempo real, FFTs, etc. En esta zona también se ejecutan las rutinas asociadas a procesamiento por ventanas, principalmente en el dominio de la frecuencia, en las que se requiere un conjunto de muestras para ser tratadas.

3. Aspectos didácticos

Esta plataforma se utiliza en diversos ámbitos, en asignaturas optativas como arquitectura de sistemas basados en microprocesador donde se estudian el uso de DSP, y como otras asignaturas de arquitecturas especializadas. Adicionalmente los alumnos de tercer ciclo también cuentan con ella para la evaluación de nuevos algoritmos y su implementación en DSPs.

Este sistema resulta atractivo a los alumnos porque el poder visualizar señales con el DSP le muestra un sistema con el que pueden interactuar. Muchos sistemas de evaluación están basados en un procesador, un codec y una conexión vía serie a un ordenador personal pero esto también ofrece bastantes limitaciones porque:

- Implica la transmisión de los datos al ordenador por el puerto serie que suele ser lento.
- Los sistemas de depuración a veces ofrecen conflictos al compartir el puerto serie.

La interfaz de usuario es especialmente práctica puesto que permite modificar en tiempo real parámetros que afectan al procesado de la señal. Como ejemplo sencillo, pueden modificarse la frecuencia de corte, con un giro de uno de los

mandos o características de un filtro con una pulsación.

Además, el disponer de un sistema autónomo permite que el alumno pueda observar como puede realizarse un sistema completamente independiente, sin que tenga que estar conectado un PC, para aplicaciones en tiempo real.

Los alumnos evalúan así todos los detalles de la implementación como son el estudio de los recursos utilizados tanto de memoria como de tiempos de ejecución. También se estudian todas las interfaces utilizadas.

4. Utilización de la plataforma en prácticas

Las prácticas están compuestas de un conjunto de trabajos orientados al aprendizaje por una parte de los elementos básicos del procesamiento de señales, como son la implementación de filtros FIR e IIR, y transformadas para extracción de información.

En el sistema se han incluido rutinas básicas para procesamiento de señales como filtrado, FFT, DCT y algunas otras que ayudan a una composición rápida de un procesado de señales. Otra de las novedades que introduce es la posibilidad de evaluar algoritmos de forma real. A veces puede ocurrir que si queremos utilizar el sistema en un entorno real, debemos dejar de utilizar muestras e introducir el sistema en el entorno real y ver como se comporta.

Las prácticas no quedan sólo en un proceso de simulación. Esta plataforma permite avanzar un paso más para facilitar la implementación de aplicaciones en tiempo real.

La familia ADSP-2106X de Analog Devices se adapta muy bien a las necesidades docentes, tanto para implementar sencillas prácticas como son implementación de filtros FIR como algoritmos más sofisticados de tratamiento de señales.

4.1. Prácticas básicas

Con estas prácticas se trata de que el alumno tome contacto con los DSP y la implementación e inconvenientes que plantea este tipo de sistemas.

Cuando se evalúan los filtros FIR se puede apreciar los diferentes efectos que tienen el

utilizar aritmética de punto flotante o punto fijo, como puede observarse en la Figura 4.

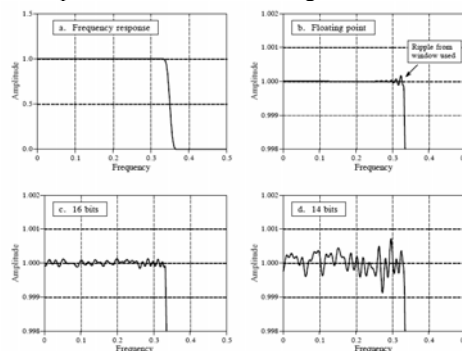


Figura 4. Comportamientos de un filtro FIR

En otras prácticas se hace síntesis y decodificación de tonos DTMF, como una aplicación que requiere adaptarse a las especificaciones originales de la señal.

4.2. Prácticas avanzadas

Los alumnos que continúan con el estudio de los DSP pasan a un nivel más complejo del manejo de éstos.

Para este siguiente nivel no es necesario tener que pasar a una nueva familia de DSPs, puesto que con la misma plataforma se puede tanto implementar algoritmos más sofisticados en C, o bien tratar de optimizar éstos a la arquitectura interna del DSP, que en algunas ocasiones no resulta trivial. Como prácticas más complejas se realiza el filtrado adaptativo y cancelación de eco.

4.3. Implementando algoritmos ICA

Los DSP no tienen que quedar meramente reducido en su uso a aplicaciones "clásicas" como el filtrado, FFTs, o DCTs. Trabajando con una de las líneas de investigación de nuestro grupo, los alumnos pueden evaluar algoritmos ICA para la separación ciega de señales [7]. Los métodos geométricos se adaptan plenamente a esta plataforma puesto que:

- Requieren procesamiento en tiempo real de señales digitales.
- La aproximación geométrica puede analizarse en la pantalla de sistema.

- La interfaz permite una fácil optimización de los parámetros.
- La complejidad de los algoritmos permiten que se pueda implementar en el DSP utilizado.

Para los estudiantes resulta muy interesante ver como los DSP pueden utilizarse para otros tipos de algoritmos distintos a los algoritmos tradicionales de tratamiento de señales. Este método no requiere estadística de alto orden e inicialmente parten de señales estadísticamente independientes.

La idea básica consiste en el concepto de independencia desde el punto de vista geométrico. En efecto, para el caso de la separación ciega de 2 canales de entrada, x_1 y x_2 , una visualización donde se representa una señal frente a otra para cada muestreo recibido presenta una distribución en forma de paralelogramo como puede apreciarse en la Figura 5.

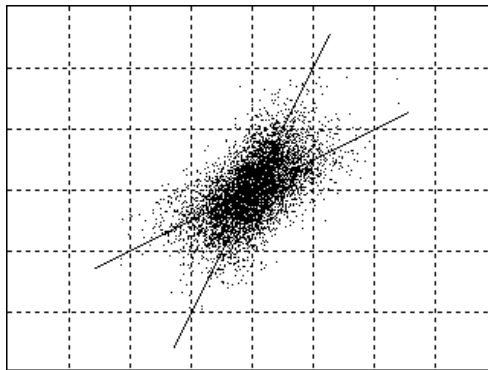


Figura 5. Captura de señales en tiempo real

El algoritmo contiene dos pasos básicos, como muestra la figura 6:

- Trasladar el paralelogramo al primer cuadrante.
- Encontrar la pendiente de los bordes Δ_1 y Δ_2 .

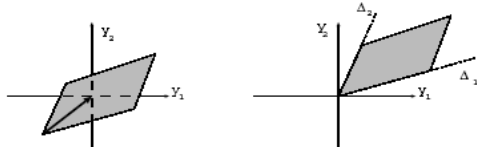


Figura 6. Desplazamiento y separación

A partir de aquí puede encontrarse la matriz de mezcla y calcular su inversa para obtener la separación de la señales de entrada.

4.4. Valoración de uso del sistema en prácticas

Según el planteamiento original, se trataba de desarrollar un prototipo y evaluarlo en diversas asignaturas para analizar diversos aspectos como son:

- Comprensión de los elementos que componen un sistema basado en DSP por los alumnos.
- Facilidad de manejo.
- Adaptabilidad a los requerimientos de las diversas asignaturas.

Obviamente, el contar sólo con un prototipo limita su utilización a pocas personas, por lo que inicialmente se ha orientado a prácticas optativas realizadas por alumnos que estaban interesados en profundizar en las arquitecturas basadas en estos procesadores.

Para los alumnos contar con esta plataforma les ha aportado una visión mucho más clara del funcionamiento de los DSPs. Una enseñanza de procesadores basada sólo en una componente exclusivamente teórica conlleva el rápido olvido y el no asentar los elementos clave.

Hemos comprobado que el prototipo desarrollado satisface las necesidades básicas para poder adentrarse en estos procesadores de forma amena y sencilla. Por otra parte se estudia la posibilidad de ampliar el número de unidades para su uso en los laboratorios.

5. Mejoras futuras

Esta plataforma demuestra ser muy interesante para implementar algoritmos rápidamente, por lo que se estudia como combinar varios de estos sistemas para crear un sistema multiprocesador utilizando los puertos de comunicaciones que incluye el ADSP-21061. Los canales serie permiten escalar la arquitectura sin coste alguno pues múltiples DSPs pueden conectarse mediante estos buses que ofrecen 40 Mbit/s.

Por otra parte, continuamente aparecen nuevos DSPs con mayores prestaciones. Las últimas arquitecturas presentan sistemas VLIW como el procesador TMS320C6711 de Texas Instruments [9] que abren nuevas posibilidades para abordar problemas más complejos. Se propone el desarrollo una nueva plataforma más sofisticada basada en este procesador y poder evaluar y comparar frente a la presentada actualmente.

6. Conclusiones

En este artículo se presenta una plataforma para el estudio de sistemas basados en DSPs. Se pretende que los estudiantes interesados en estos procesadores puedan estudiarlos no sólo desde el punto de vista teórico, sino también poder realizar prácticas en aplicaciones de tiempo real sin que para ello les suponga un esfuerzo excesivo.

El ir "a ciegas" en estos sistemas puede suponer horas innecesarias de depuración por no poder evaluar fácilmente que ocurre internamente. La visión en tiempo real de las señales y extracción de parámetros ofrece innumerables ventajas, no sólo permite la localización más rápida de errores sino también la optimización más adecuada del código y un ajuste más preciso del comportamiento del sistema que se desarrolla.

Agradecimientos

Este sistema se ha desarrollado y financiado gracias al Proyecto de Innovación Docente "Módulo docente para separación de señales y análisis en componentes independientes" del Vicerrectorado de Planificación, Calidad y Evaluación Docente de la Universidad de Granada

Referencias

- [1] Analog Devices, ADSP-2106X Family Data Sheet. 1997
- [2] A. Díaz García, L. Parrilla Roure, M. Rodríguez Alvarez, C. G. Puntonet. "Decodificación de un sonido encriptado mediante la utilización de un procesador digital de señal, Seminario Anual de Automática, Electrónica Industrial e Instrumentación, (SAAEI'98). Universidad Publica de Navarra, pp. 127-128, 16 a 18 de Septiembre de 1998
- [3] DSPGuru. <http://www.dspguru.com>. 2005
- [4] Llamuza Castell, R. Implementación del algoritmo de codificación de voz GSM 06.10 en un sistema de tiempo real basado en DSPs. Proyecto fin de carrera Ingeniería Electrónica, Dpto. Arquitectura y Tecnología de Computadores. Universidad de Granada. 1999
- [5] Matlab. <http://www.mathworks.com>, The MathWorks Inc., Natick, Massachusetts, USA, 2005.
- [6] J. Picone, J. Hamaker, R. Brown, R.A. Cole J.H.L. Hansel. Modern DSP Education. The Story of Three Greek Philosophers. IEEE Signal Processing Magazine, vol.16. no.5, pp. 48-56. Sep. 1999
- [7] C. G. Puntonet, A. Prieto, C. Jutten, M. Rodríguez Alvarez, and J. Ortega, "Separation of sources: A geometry-based procedure for reconstruction of n-valued signals," Signal Processing, vol. 46, no. 3, pp. 267-284, 1995.
- [8] Smith, S.W. The Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing, 1997. ISBN 0-9660176-3-3
- [9] Texas Instruments. TMS320C6711 /11B /11C /11D Floating-Point Digital Signal Processors (Rev. L) Data Sheet.. Mayo 2004.