

Procesadores digitales de señal (DSP)

Arquitecturas y criterios de selección

COMPONENTES. Los DSP o procesadores digitales de señal son microprocesadores específicamente diseñados para el procesamiento digital de señal. Algunas de sus características más básicas como el formato aritmético, la velocidad, la organización de la memoria o la arquitectura interna hacen que sean o no adecuados para una aplicación en particular, así como otras que no hay que olvidar, como puedan ser el coste o la disponibilidad de una extensa gama de herramientas de desarrollo.

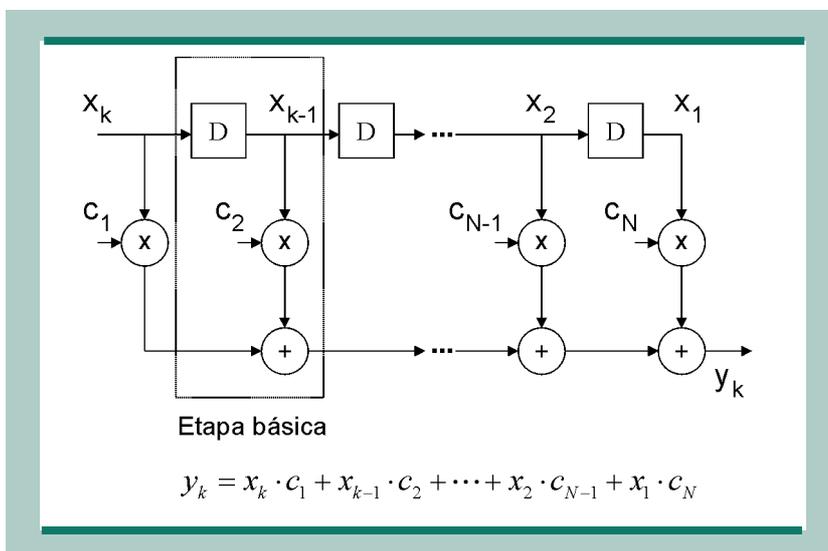
JORDI SALAZAR.

DPTO. INGENIERÍA ELECTRÓNICA. CENTRO DE SISTEMAS Y SENSORES ELECTRÓNICOS,
UNIVERSIDAD POLITÉCNICA DE CATALUÑA.

Jsalazar@eel.upc.es

Un sistema de procesamiento digital de señal puede definirse como cualquier sistema electrónico que realice procesamiento digital de señal, entendiéndose por él la aplicación de operaciones matemáticas a señales representadas de forma digital. Las señales son representadas de forma digital mediante secuencias de muestras. A menudo, estas muestras se obtienen de señales físicas (por ejemplo, señales de audio) utilizando transductores (un micrófono en este caso) y convertidores analógico-digitales. Después del procesamiento matemático, las señales digitales pueden volver a convertirse en señales físicas mediante convertidores digital-analógicos.

Figura 1. Estructura de un filtro de respuesta impulsional finita (FIR)



Si bien, en principio, el corazón de un sistema de procesamiento digital puede ser un microcontrolador, un procesador de propósito general o un procesador digital de señal (DSP), en sistemas en los cuales la carga computacional es extremadamente intensa la solución óptima pasa por escoger a un DSP.

En la actualidad, los cuatro grandes fabricantes de DSP son Texas Instruments, con la serie TMS320; Motorola, con las series DSP56000, DSP56100, DSP56300, DSP56600 y DSP96000; Lucent Technologies (anteriormente AT&T), con las series DSP1600 y DSP3200; y Analog Devices, con las series ADSP2100 y ADSP21000.

¿QUÉ ES UN DSP?

Estrictamente hablando, el término DSP se aplica a cualquier chip que trabaje con señales representadas de forma digital. En la práctica, el término se refiere a microprocesadores específicamente diseñados para realizar procesamiento digital de señal. Los DSP utilizan arquitecturas especiales para acelerar los cálculos matemáticos intensos implicados en la mayoría de sistemas de procesamiento de señal en tiempo real. Por ejemplo, las arquitecturas de los DSP incluyen circuitería para ejecutar de forma rápida operaciones de multiplicar y acumular, conocidas como MAC. A menudo poseen arquitecturas de memoria que permiten un acceso múltiple para permitir de forma simultánea cargar varios operandos, por ejemplo, una muestra de la señal de entrada y el coeficiente de un filtro simultáneamente en

DSP

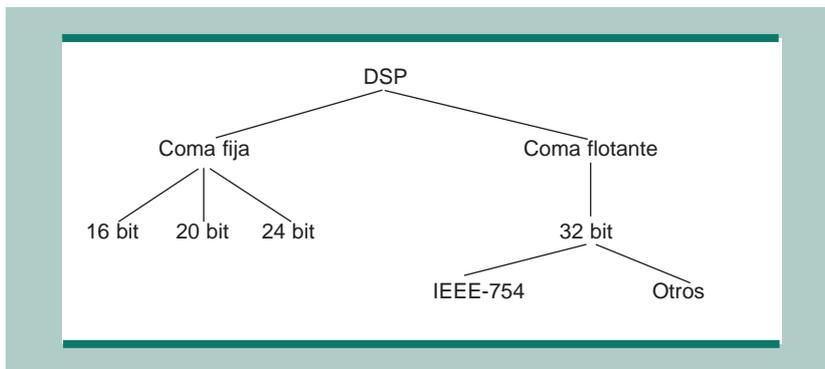


Figura 2. Representaciones numéricas comunes en los DSP comerciales

paralelo con la carga de la instrucción. También incluyen una variedad de modos especiales de direccionamiento y características de control de flujo de programa diseñadas para acelerar la ejecución de operaciones repetitivas. Además, la mayoría de los DSP incluyen en el propio chip

periféricos especiales e interfaces de entrada salida que permiten que el procesador se comunique eficientemente con el resto de componentes del sistema, tales como convertidores analógico-digitaes o memoria.

La diferencia esencial entre un DSP y un microprocesador es que el DSP tiene características diseñadas para soportar tareas de altas prestaciones, repetitivas y numéricamente intensas. Por contra, los microprocesadores de propósito general o microcontroladores no están especializados para ninguna aplicación en especial; en el caso de los microprocesadores de propósito general, ni están orientados a aplicaciones de control, en el caso de los microcontroladores.

Aunque el ejemplo del filtro de respuesta impulsional finita (FIR) ha sido ampliamente utilizado en el entorno DSP, es quizás el más simple que permite ilustrar la necesidad de estas prestaciones en los DSP, las cuales permiten concebir muchas de las funciones de procesado en tiempo real.

La mecánica del algoritmo del filtro FIR es bastante sencilla. Los bloques D en la figura 1 son retardos unitarios; su salida es una copia de la entrada retardada en un período de muestreo. El filtro FIR se construye a partir de una serie de etapas básicas que se van repitiendo. Cada etapa básica incluye una operación del tipo multiplicación y acumulación. La señal de entrada x_k es un conjunto de valores discretos obtenidos mediante muestreo de una señal analógica. El valor x_{k-1} es en realidad el valor que tenía x_k en un período de muestreo anterior. De forma similar x_{k-2} sería la x_k de dos períodos de muestreo anteriores. Cada vez que una nueva muestra llega, las que estaban almacenadas previamente se desplazan una posición hacia la derecha y una nueva muestra de salida se calcula después de multiplicar la nueva muestra y cada una de las anteriores por sus correspondientes coeficientes. En la figura, c_n representa a los coeficientes, donde n es el número del coeficiente. Los resultados de cada multiplicación se suman para formar la nueva muestra de salida y_k . El algoritmo de cálculo se basa en operaciones del tipo multiplicación y acumulación ($A=B*C+D$), siendo la multiplicación el verdadero cuello de botella en el cálculo de la mayoría de los algoritmos de procesado digital.

APLICACIONES

Los DSP se utilizan en muy diversas aplicaciones, desde sistemas radar hasta la electrónica de consumo. Naturalmente, ningún procesador satisface todas las necesidades de todas o la mayoría de aplicaciones. Por lo tanto, la primera tarea para el diseñador al elegir un DSP es ponderar la importancia relativa de las prestaciones, coste, integración, facilidad de desarrollo, consumo y

Glosario de términos

DSP: (Digital Signal Processor). Procesador digital de señal.

ALU: (Arithmetic/Logic Unit). Unidad aritmético-lógica. Unidad de ejecución en un procesador responsable de la aritmética (sumar, restar, desplazar, etc.) y de la lógica (y, o, no, o-exclusiva, etc.).

MIPS: (Million instructions per second). Número de instrucciones que el procesador puede ejecutar por segundo. La clave está en cuánto puede hacer cada instrucción. El parámetro MFLOPS es normalmente una medida más fiable de las prestaciones del procesador.

MOPS: (Million operation per second). Número total de operaciones que el procesador puede realizar por segundo. Se incluye accesos DMA, transferencias de datos, operaciones de entrada salida. Este parámetro proporciona una idea aproximada de la capacidad de procesado y de entrada salida del procesador.

MFLOPS: (Million floating-point operations per second). Número de multiplicaciones, sumas, restas, etc. en coma flotante que el procesador puede realizar. Algunas veces este parámetro hace referencia a valores de pico en lugar de valores sostenidos.

MBPS: (Mega-bytes per second). Proporciona una medida del rendimiento total de procesamiento de datos.

MMACS: (Million multiply-accumulate per second). Número de multiplicaciones y acumulaciones que el procesador puede realizar por segundo.

VLIW: (Very long instruction word). Formato muy largo de palabra de instrucción.

FFT: (Fast Fourier Transform). Transformada de Fourier rápida. Método computacional eficiente para estimar el espectro frecuencial de una señal.

FIR: (Finite impulse response). Respuesta impulsional finita. Una categoría de filtros digitales.

IIR: (Infinite impulse response). Respuesta impulsional infinita. Una categoría de filtros digitales.

PGA: (Pin grid array). Un tipo de encapsulado para circuitos integrados. Las conexiones externas se realizan con terminales de conexión dispuestos en forma de cuadrícula.

PQFP: (Plastic quad flat pack). Un tipo de encapsulado para circuitos integrados.

QFP: (Quad flat pack). Un tipo de encapsulado para circuitos integrados. Los circuitos integrados con este encapsulado son típicamente más baratos que el mismo circuito con encapsulado PGA.

TQFP: (Thin quad flat pack). Tipo de encapsulado similar, pero más delgado, al PQFP. El encapsulado TQFP normalmente se utiliza en sistemas portátiles pequeños.

DSP

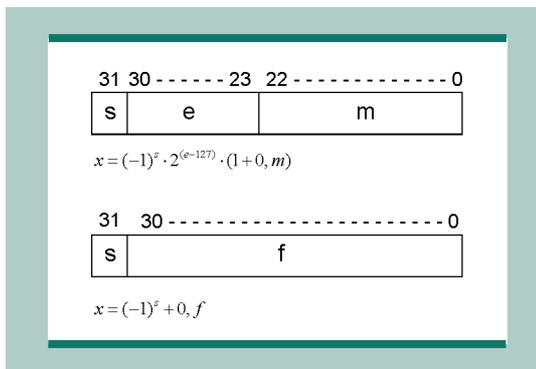


Figura 3. (Arriba) Formato de coma flotante IEEE-754; 1 bit de signo, 8 de exponente y 23 de mantisa. (Abajo) Formato en coma fija; 1 bit de signo y 31 bits significativos

otros factores para las necesidades de la aplicación en particular.

Las grandes aplicaciones, en términos de dinero que mueven sus productos, se realizan para los sistemas pequeños, baratos y con un gran volumen de producción como los de telefonía celular, disqueteras y modems, en donde el coste y la integración son de

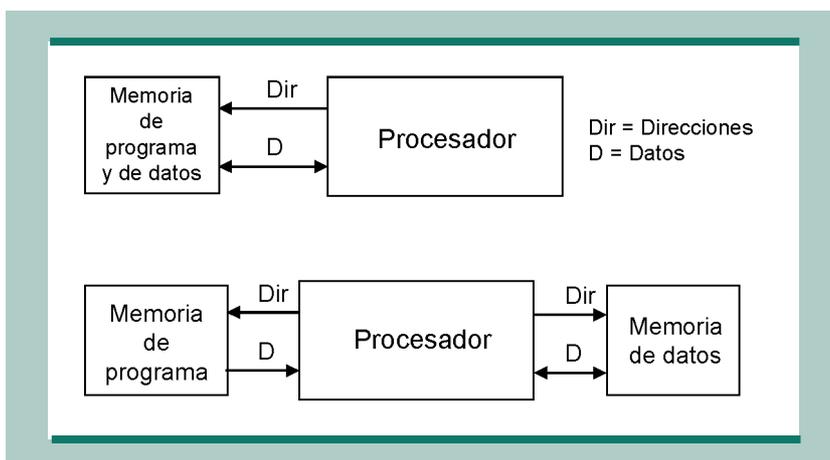
la mayor importancia. En sistemas portátiles, alimentados por baterías, el consumo es crítico. Sin embargo, la facilidad de desarrollo es generalmente en estas aplicaciones menos importante para el diseñador. A pesar de que estas aplicaciones casi siempre implican el desarrollo de hardware y software a medida, el enorme volumen de producción justifica el esfuerzo extra de desarrollo.

Una segunda clase de aplicaciones englobaría a aquellas que procesan un gran volumen de datos mediante algoritmos complejos. Ejemplos incluyen la exploración sonar y sísmica, donde el volumen de producción es bajo, los algoritmos más exigentes y el diseño del producto más largo y complejo. En consecuencia, el diseñador busca un DSP que tenga máximas prestaciones, buena facilidad de uso y soporte para configuraciones multiprocesador. En algunos casos, más que diseñar el propio hardware y software, el sistema se construye a partir de placas de desarrollo de catálogo y el software a partir de librerías de funciones ya existentes.

CARACTERÍSTICAS DE LOS DSP

La elección de un DSP que posea unas ciertas características estará muy condicionada a la aplicación que se quiera destinar. En este apartado se presenta un conjunto de aspectos carac-

Figura 4. (Arriba) Arquitectura Von Neumann. (Abajo) Arquitectura Harvard



terísticos de los DSP sin que se pretenda con ello hacer una lista exhaustiva. Dichos aspectos deberán tenerse en cuenta a la hora de su elección para una aplicación en particular.

Formato aritmético

Una de las características fundamentales de los DSP es el tipo de formato aritmético utilizado por el procesador. La figura 2 muestra la estrecha relación entre formato numérico y número de bits del DSP.

La figura 3, arriba y abajo, muestra los formatos de coma flotante y coma fija, respectivamente. En el formato IEEE-754 de coma flotante la “s” indica que el bit más significativo es el signo, donde un 1 indica que se trata de un número negativo. La “e” indica exponente, formado por 8 bits y la “m”, de 23 bits, la mantisa del número.

Al carecer de exponente el formato en coma fija, éste puede representar números con más bits significativos que el formato en coma flotante del mismo tamaño en bits. En este ejemplo, 31 bits son significativos, “f”, comparados con los 23 del formato IEEE-754.

Para un mismo tamaño en número de bits, el formato en coma fija proporciona una mejor resolución que el formato en coma flotante. Sin embargo, es este último quien posee un margen dinámico superior. Así, por ejemplo, si se considera una representación en coma fija de 32 bits, el mínimo valor que puede ser representado es 2^{-31} siendo el mayor $1-2^{-31}$. La relación entre ambos, la cual resulta en el margen dinámico, es de $2,15 \cdot 10^9$, aproximadamente 187 dB. En cambio, con una representación en coma flotante, con 24 bits de mantisa y 8 de exponente, pueden representarse números desde $5,88 \cdot 10^{-39}$ hasta $3,40 \cdot 10^{38}$, resultando en un margen dinámico de $5,79 \cdot 10^{76}$ o de 1535 dB.

La aritmética de coma flotante es más flexible que la de coma fija. Con coma flotante, los diseñadores de sistemas tienen acceso a un margen dinámico más amplio. En consecuencia, los DSP de coma flotante son generalmente más fáciles de programar que los de coma fija, pero son usualmente más caros. El mayor coste es resultado del requisito de una mayor complejidad circuital que se traduce en un mayor tamaño de chip. Sin embargo, el mayor margen dinámico facilita su programación pues el programador no debe preocuparse por el margen dinámico ni por la precisión. Por el contrario, en los DSP de coma fija el programador a menudo debe escalar las señales en sus programas para asegurar una adecuada precisión numérica con el limitado margen dinámico del DSP de coma fija.

Por lo general, las aplicaciones con un gran volumen de unidades y/o bajo consumo utilizan los DSP de coma fija al ser la prioridad en este tipo

DSP

de aplicaciones el bajo coste. Los programadores determinan el margen dinámico y la precisión necesarias de la aplicación, ya sea analíticamente o a través de simulaciones, y entonces aplican operaciones de escalado dentro del código de la aplicación en los puntos en donde sea necesario. En aplicaciones donde el coste no sea un requisito crítico o que demanden un margen dinámico y precisión elevadas, o donde la facilidad de desarrollo sea vital, los DSP de coma flotante poseen ventaja.

Mediante rutinas software es posible emular el comportamiento de un dispositivo de coma flotante con uno de coma fija. Sin embargo, tales rutinas resultan generalmente caras en términos de ciclos del procesador. En consecuencia, raramente se suele emular la aritmética de coma flotante.

Ancho de palabra

Los DSP de coma flotante utilizan un bus de datos de 32 bits. En los DSP de coma fija, el tamaño más común es de 16 bits. Sin embargo, las familias DSP5600x y DSP563xx de Motorola utilizan un formato de 24 bits, mientras que la familia ZR3800x de Zoran utiliza 20 bits. El tamaño del bus de datos tiene un gran impacto en el coste, ya que influye notablemente en el tamaño del chip y el número de patillas del encapsulado, así como en el tamaño de la memoria externa conectada al DSP. Por lo tanto, se intenta utilizar el integrado con el menor tamaño de palabra que la aplicación pueda tolerar.

De la misma forma que ocurre con la elección entre coma fija y coma flotante, existe un compromiso entre tamaño de palabra y complejidad. Una aplicación que requiera 24 bits puede ser desarrollada por un DSP de 16 bits a costa de un aumento de complejidad en el software. Por ejemplo, con un DSP de 16 bits se pueden realizar operaciones con aritmética de doble precisión y 32 bits combinando las instrucciones adecuadas. Naturalmente, la doble precisión será mucho más lenta que la precisión simple. Si el grueso de la aplicación puede desarrollarse en precisión simple, puede tener sentido emplear la doble precisión únicamente en aquellas partes del programa que lo necesiten. Pero si la mayoría de la aplicación requiere más precisión, entonces un DSP con un tamaño de palabra mayor sería la opción adecuada.

La mayoría de los DSP utilizan un ancho de la palabra de instrucción igual a la de los datos, pero no todos lo hacen. Así por ejemplo, la familia ADSP-21xx de Analog Devices utiliza 16 bits para los datos y 24 bits para las instrucciones mientras que la familia ZR3800x

Figura 5. Ejecución de instrucciones sin pipeline. I1 y I2 representan la instrucción 1 y la 2, respectivamente

	Ciclo de reloj							
	1	2	3	4	5	6	7	8
Obtención Instrucción	I1				I2			
Descodificación		I1				I2		
Lecturas/Escritura operando			I1				I2	
Ejecución				I1				I2

Figura 6. Procesador que utiliza la técnica del pipelining

	Ciclo de reloj (instrucción)							
	1	2	3	4	5	6	7	8
Obtención Instrucción	I1	I2	I3	I4	I5	I6	I7	I8
Descodificación		I1	I2	I3	I4	I5	I6	I7
Lectura/Escritura operando			I1	I2	I3	I4	I5	I6
Ejecución				I1	I2	I3	I4	I5

Figura 7. Efecto en la pipeline ante la llegada de una instrucción de salto

	Ciclo de reloj (instrucción)							
	1	2	3	4	5	6	7	8
Obtención Instrucción	Salto	I2	-	-	N1	N2	N3	N4
Descodificación		Salto	-	-	-	N1	N2	N3
Lecturas/Escritura operando			Salto	-	-	-	N1	N2
Ejecución				Salto	NOP	NOP	NOP	N1

de Zoran utiliza un formato de instrucción de 32 bits.

Velocidad

La medida clave para saber si un DSP es o no apropiado para una aplicación es su velocidad de ejecución. Existen varias formas para medir la velocidad de un procesador, aunque quizás el parámetro más usual es el tiempo de ciclo de instrucción: tiempo necesario para ejecutar la instrucción más rápida del procesador. Su inverso dividido por un millón da lugar a la velocidad del procesador en millones de instrucciones por segundo o MIPS. En la actualidad todos los DSP ejecutan una instrucción por ciclo de instrucción.

Un problema que se presenta cuando se comparan los tiempos de ejecución de instrucciones de varios procesadores es que la cantidad de trabajo realizado por una instrucción varía significativamente de un procesador a otro. Por ejemplo, algunos DSP disponen de desplazadores combinatorios ("barrel shifters") que permiten hacer desplazamientos de múltiples bits en los datos con sólo una instrucción, mientras que otros DSP requieren que el dato sea desplazado con repetidas instrucciones de desplazamiento de un solo bit. De forma similar, algunos DSP permiten el movimiento de datos en paralelo (carga simultánea de datos mientras se ejecuta una instrucción) que no están relacionados con la instrucción que la ALU está ejecutando, pero otros DSP sólo soportan movimientos en paralelo que

DSP

estén relacionados con los operandos de la instrucción que esté ejecutando la ALU.

El parámetro MIPS, al igual que MFLOPS, MOPS, MBPS y otros, se miden de forma muy precisa aunque su valor no necesariamente dice mucho de lo que un determinado DSP es capaz de hacer. Dichos valores sirven para ubicar cada DSP en una categoría amplia en cuanto a prestaciones pero uno se debería preguntar si se trata de valores de pico o sostenidos. Además, también nos deberíamos preguntar cómo se ven afectadas las prestaciones del DSP cuando muchos de los datos se encuentran fuera del chip, en la memoria externa.

Una solución a estos problemas consiste en decidir una operación básica y utilizarla como referencia al comparar distintos DSP. La operación que suele tomarse como referencia es la MAC. Desafortunadamente, los tiempos de ejecución de la MAC proporcionan, a veces, poca información para poder diferenciar entre distintos DSP, ya que en la mayoría de ellos esta instrucción se ejecuta en un solo ciclo de instrucción, y como se ha mencionado anteriormente, algunos DSP pueden hacer mucho más que otros en una simple instrucción MAC. Además, los tiempos de ejecución de la MAC no suele reflejar las prestaciones de otro tipo importante de operaciones como los bucles que están presentes en todas las aplicaciones.

Un enfoque mucho más general consiste en definir un conjunto algoritmos o funciones, como un filtro FIR o IIR, e implementarlo en distintos DSP y de esta forma ver cuál de ellos proporciona unas mejores prestaciones. Sin embargo, la implementación de estos algoritmos para distintos DSP puede resultar una tarea ardua. En este sentido, una buena referencia pueden ser los tests que efectúa la Berkeley Design Technology, Inc., pionera en utilizar distintas porciones de algoritmos y funciones para medir las prestaciones de los diferentes DSP.

Organización de la memoria

La organización del subsistema de memoria de un DSP puede tener un gran impacto en sus prestaciones. Como se ha mencionado anteriormente, la instrucción MAC, así como otras, son fundamentales en muchos de los algoritmos de procesado de señal. Una ejecución rápida de la instrucción MAC requiere que la lectura en memoria del código de la instrucción y de sus dos operandos se haga en un ciclo de instrucción. Existe una variedad de formas de hacerlo, utilizando memorias multipuerto para permitir múltiples accesos a memoria en un ciclo de instrucción, mediante memorias de datos e instrucciones separadas (arquitectura Harvard), y memo-

rias caches de instrucciones para permitir el acceso a la memoria para la obtención de datos mientras que las instrucciones se obtienen de la cache en lugar de la memoria. La figura 4 muestra las diferencias entre la arquitectura Harvard y la Von Neumann, esta última utilizada en la mayoría de procesadores de propósito general.

Otro punto importante a tener en cuenta es la cantidad de memoria que soporta el DSP, interna y externamente. Atendiendo a las características de la aplicación, la mayoría de los DSP de coma fija poseen memorias internas, en el propio chip, de tamaño pequeño medio, entre 256 y 32k palabras, y un bus externo de direcciones pequeño. Así por ejemplo, la mayoría de los DSP de coma fija de Analog Devices, Lucent Technologies, Motorola y Texas Instruments tienen buses de direcciones de 16 bits o menos, lo que limita la cantidad de memoria externa de acceso directo.

Por el contrario, la mayoría de los DSP de coma flotante proporcionan poca o ninguna memoria interna, pero se caracterizan por tener buses de direcciones externos de gran tamaño, para soportar una gran cantidad de memoria externa. Por ejemplo, el ADSP-21020 de Analog Devices no tiene memoria interna pero posee un bus externo de direcciones de 24 bits. De forma similar, el TMS320C30 de Texas Instruments posee 6k palabras de memoria interna y dos buses externos de direcciones, uno de 24 bits y el otro de 13 bits. Además, estos DSP poseen memorias cache para permitir un uso más eficiente de memorias externas lentas.

Segmentación (“pipelining”)

“Pipelining” es una técnica para incrementar las prestaciones de un procesador, que consiste en dividir una secuencia de operaciones en otras de más sencillas y ejecutar en lo posible cada una de ellas en paralelo. En consecuencia se reduce el tiempo total requerido para completar un conjunto de operaciones. Casi todos los DSP del mercado incorporan el uso de la segmentación en mayor o menor medida.

Para ilustrar de qué forma la técnica de la segmentación mejora las prestaciones de un procesador, considérese un hipotético procesador que utiliza unidades de ejecución separadas para la ejecución de una única instrucción:

- *Obtención de la instrucción de la memoria
- *Descodificar la instrucción
- *Leer o escribir un operando de la memoria
- *Ejecutar la parte de la instrucción relacionada con la ALU o MAC.

La figura 5 muestra la temporización de varias instrucciones ejecutadas de forma secuencial. Si se supone que cada etapa o unidad de ejecución

DSP

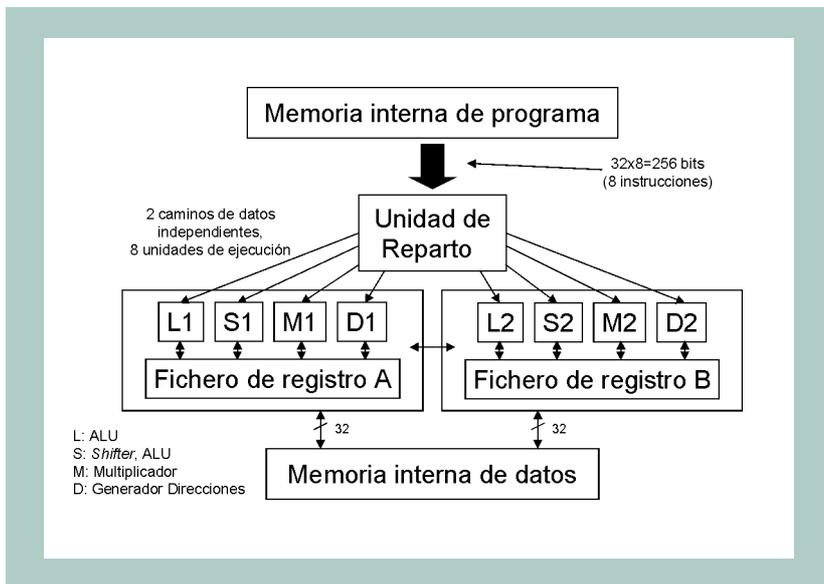


Figura 8. Núcleo de los TMS320C62xx de Texas Instruments

tarda 20 ns en ejecutar su parte de la instrucción, entonces el procesador ejecuta una instrucción cada 80 ns. Sin embargo, también se observa que el hardware asociado a cada etapa de ejecución está inactivo el 75% del tiempo. Esto ocurre porque el procesador no empieza a ejecutar una nueva instrucción hasta que finaliza la ejecución de la instrucción en curso.

Un procesador que implementara la técnica de “pipelining” obtendría una nueva instrucción inmediatamente después de haber obtenido la anterior. De forma similar, cada instrucción sería descodificada después de haber terminado la descodificación de la instrucción anterior. Con esta filosofía, las instrucciones se ejecutan de forma solapada, tal y como se ilustra en la figura 6. Las unidades de ejecución trabajan en paralelo, mientras una obtiene el código de una instrucción otra está descodificando la anterior y así sucesivamente. En consecuencia, una vez que la “pipeline” está llena, cada 20 ns se ejecuta una instrucción, lo cual representa un factor de mejora de prestaciones de cuatro respecto a un procesador que no incorpore dicha técnica.

Aunque la mayoría de los DSP utilizan la técnica de segmentación, su profundidad o número de etapas varía de un procesador a otro. En general, cuanto mayor sea el número de etapas menor tiempo tardará el procesador en ejecutar una instrucción.

En el ejemplo anterior se ha supuesto un procesador con una eficiencia en el uso de la “pipeline” del 100%. En realidad, esto no siempre ocurre así. La eficiencia se ve disminuida por varias causas, entre las cuales se encuentra el hecho de que un procesador necesite dos ciclos para escribir en memoria, se obtenga el código de una instrucción de salto de programa o bien la

petición de una interrupción. La figura 7 muestra qué es lo que pasa cuando una instrucción de salto llega a la “pipeline”. En el momento en que el procesador detecta la llegada de una instrucción de salto en la descodificación del segundo ciclo de reloj, la “pipeline” se vacía y detiene la obtención de nuevas instrucciones. Esto provoca que la instrucción de salto se ejecute en cuatro ciclos. Posteriormente, el procesador comienza la obtención de las instrucciones (N1-N4) a partir de la dirección de salto y del quinto ciclo de reloj. A causa de este tipo de situaciones, casi todos los DSP incorporan algún tipo de mejora en el uso de la segmentación con el propósito de reducir su posible ineficiencia temporal.

Consumo

El uso cada vez más extendido de los DSP en aplicaciones portátiles como la telefonía celular hace que el consumo sea un factor a tener muy en cuenta en el momento de decidirse por un DSP u otro. Conscientes de esta necesidad, los fabricantes de DSP ya fabrican DSP para tensiones bajas de trabajo (3,3 V -3 V) que incorporan prestaciones para la gestión de energía, como pueden ser los modos “sleep” o “idle” que inhiben el reloj del DSP a todas o sólo algunas partes del mismo, divisores programables del reloj para permitir la realización de determinadas tareas a velocidad inferior o en control directo de periféricos, lo que permite la desactivación de algunos de ellos si no se prevé su aplicación.

Coste

Generalmente el coste del DSP es el principal parámetro en todos aquellos productos que se van a fabricar en grandes volúmenes. En tales aplicaciones, el diseñador intenta utilizar el DSP con coste inferior y que satisfaga las necesidades de la aplicación aun cuando ese dispositivo pueda ser considerado poco flexible y más difícil de programar que otros DSP más caros. De entre las familias de DSP, el más barato será aquel que tenga menos características funcionales, menos memoria interna y probablemente menos prestaciones que otro más caro. Sin embargo, una diferencia clave en el precio está en el encapsulado. Los encapsulados PQFP y TQFP son usualmente bastante más baratos que los PGA.

ARQUITECTURAS DE ALTAS PRESTACIONES

Hablar de DSP obliga a hacer referencia a las nuevas arquitecturas VLIW (Very Long Instruction Word) que están siendo adoptadas por los DSP de muy altas prestaciones. Las Tecnologías de la Información y las Comunicaciones (TIC) demandan cada vez más recursos para poder procesar grandes volúmenes de datos. Hasta ahora, los avances en la capacidad de cómputo de los procesadores se ha basado en el aumento de

DSP

la velocidad del reloj y en innovaciones en la planificación, por parte del hardware, de la ejecución de instrucciones. Este modelo actual presupone que cada nueva generación de procesadores es capaz de ejecutar más instrucciones y será difícil que las arquitecturas tradicionales continúen doblando prestaciones cada 12-18 meses sin que se emigre a una nueva tecnología. El número de instrucciones por ciclo aumenta y también lo hace el número de interdependencias entre instrucciones a comprobar para determinar qué instrucciones pueden ejecutarse de forma simultánea. La lógica compleja requerida para la correcta planificación de instrucciones ocupa una gran parte del silicio del procesador y empieza a no tener sentido dedicar una gran parte de los recursos del procesador a la planificación de instrucciones.

En su lugar, parece tener más sentido utilizar ese silicio para poder ejecutar más instrucciones por ciclo, incorporando más unidades funcionales y aumentar así el paralelismo de ejecución, mientras que la planificación de instrucciones se realiza por el compilador. Éste es el principio en que se basa la arquitectura VLIW. Esta reducción de complejidad, hardware más sencillo y menor número de transistores, permite incrementar la velocidad del reloj y al mismo tiempo reducir el consumo.

El concepto de arquitectura VLIW no es nuevo ya que desde 1975 han ido apareciendo procesadores VLIW pero siempre más a un nivel de prototipo que a nivel comercial. Sin embargo, no ha sido hasta hace relativamente poco, aproximadamente en 1997, que los esfuerzos en la mejora del compilador, en aspectos de paralelismo ha hecho que estos procesadores sean realmente eficientes. Tradicionalmente, las ventajas asociadas con la arquitectura VLIW eran difíciles de conseguir y su futuro era cuestionable. La falta de compiladores eficientes a menudo significaba que el programador tuviera que pasar muchas horas tratando de optimizar, la mayoría de las veces sin éxito, su extenso código para mejorar las prestaciones de la aplicación.

Las arquitecturas VLIW están estrechamente relacionadas con los procesadores superescalares. Ambos tratan de aumentar la velocidad computacional mediante paralelismo a nivel de instrucciones en el que se utilizan múltiples copias de algunas etapas de la "pipeline" o unidades de ejecución trabajando en paralelo. Las dos diferencias principales recaen en cómo se formulan las instrucciones y en cómo se realiza su planificación o secuenciamiento.

En una arquitectura VLIW, las instrucciones poseen un formato grande de palabra compues-

to por múltiples instrucciones independientes que incluye un campo de control para cada una de las unidades de ejecución. El tamaño de la instrucción depende de dos factores: el número de unidades de ejecución disponibles y la longitud de código requerida para cada una de ellas. Una consecuencia de ello es que los buses internos de datos y de instrucciones son de mayor tamaño.

Por otro lado, a diferencia de los procesadores superescalares en los que la planificación de las instrucciones para buscar el máximo paralelismo la realiza el propio procesador, en las arquitecturas VLIW esta tarea la realiza el compilador. Esta planificación es conocida como estática ("static scheduling"). Una ventaja inmediata de este tipo de planificación es que permite dedicar más tiempo a encontrar la mejor optimización, aunque esto hace que el compilador sea más complejo ya que sobre él recae la responsabilidad de agrupar de la mejor forma posible las instrucciones.

La figura 8 muestra el núcleo de los TMS320C62xx de Texas Instruments. En particular, el TMS320C6202 trabaja a 250 MHz y es capaz de ejecutar 2000 MIPS y 500 MMACS. El mismo fabricante ya está anunciando la salida al mercado de dos nuevas familias, la TMS320C64x de altas prestaciones y la TMS320C55x de bajo consumo, también basadas en una arquitectura VLIW.

El TMS320C64x tendrá una frecuencia máxima de trabajo de hasta 1,1 GHz que permitirá obtener prestaciones de 8800 MIPS y 4400 MMACS. Sus prestaciones son 10 veces superiores a las que presenta el actual líder, el TMS320C62xx. Esta mejora en prestaciones se ha conseguido haciendo que las unidades funcionales sean más flexibles, lo que permite ejecutar un mayor número de instrucciones, añadiendo instrucciones especiales para aplicaciones de comunicaciones digitales y vídeo que permitan una reducción del código y un incremento de la eficiencia del compilador, e incrementando el nivel de paralelismo. Por todo ello, puede realizar simultáneamente 2 operaciones de 16 bits o bien 4 de 8 bits. Además, el TMS320C64x cuenta con un mayor número de registros, 64 en total, y un camino de datos interno, "datapath", de 64 bits, que dobla en ambos casos las cifras del TMS320C62xx.

Para que el lector pueda hacerse una idea de lo que representan estos valores, alrededor de 1997 los valores para las frecuencias de trabajo y los MIPS eran de 10 MHz a 100 MHz y de 10 a 80, respectivamente.

Por su parte, el TMS320C55x se presenta como el DSP con menor consumo en el mercado, tan solo 0,05 mW/MIPS a 0,9 V y con unas prestaciones que van de los 140 a 800 MIPS.

DSP

Comparado con su más directo competidor, el TMS320C54x, también de Texas, su factor de consumo/MIPS es 6 veces inferior a la vez que consigue multiplicar por 5 sus prestaciones. Este bajo consumo se debe a la incorporación de un sistema de gestión automática avanzada de la energía. El sistema continuamente supervisa las partes del chip que se están utilizando y procede a desactivarlas cuando no son necesarias. Además, se ofrece al diseñador la posibilidad de poder configurar a su conveniencia 6 componentes del DSP, cada uno representado por un bit en uno de sus registros, lo que origina un total de 64 modos configurables distintos.

SOPORTE TÉCNICO Y FACILIDAD DE DESARROLLO

En el momento de decantarse por un DSP u otro será necesario conocer completamente los requisitos de procesamiento del sistema. Muchos DSP pueden ser eliminados previamente con sólo tener en cuenta consideraciones de falta de potencia de cálculo, resolución insuficiente, coste, etc. Esto probablemente deje todavía a un número de posibles candidatos para los cuales será preciso realizar otro tipo de análisis.

En el apartado anterior se han visto aquellas características más técnicas de los DSP y que están estrechamente relacionadas con los algoritmos de la aplicación a implementar. Sin embargo, no se han considerado para nada aspectos relacionados con el desarrollo de la aplicación. El DSP que finalmente se elija deberá disponer de un amplio conjunto de herramientas de desarrollo. Algunos requerimientos básicos son:

- *Documentación de diseño detallada
- *Herramientas de desarrollo de código en ensamblador y/o en lenguaje de alto nivel
- *Herramientas para el test de la funcionalidad del diseño
- *Notas de aplicación u otro tipo de ayuda al diseño

El objetivo será seleccionar el DSP que permita terminar el proyecto en el tiempo previsto y que la solución alcanzada sea la que presente la mejor relación coste-eficiencia. En aplicaciones de gran volumen de producción, esto probablemente signifique que el DSP escogido será el más barato que pueda realizar la aplicación. Para aplicaciones con un volumen bajo-medio existirá el compromiso entre el coste de las herramientas de desarrollo y el coste y eficiencia del DSP. En cambio, para aplicaciones con un volumen bajo de producción tendrá más sentido utilizar un DSP que facilite el diseño o que tenga las herramientas de desarrollo más baratas.

Cabe la posibilidad que la elección del DSP sea un proceso iterativo. En otras palabras, puede no haberse escogido el dispositivo correcto. Podría ser que aparecieran problemas imprevistos en la fase de desarrollo y prueba del código o incluso que se encontrara que un DSP más barato y menos potente pudiera ser el elegido. Comúnmente, las especificaciones del diseño alterarán y forzarán a replantear la solución escogida. Los dos primeros casos pueden evitarse haciendo más minuciosa la búsqueda del DSP que más se adecúe a la aplicación en particular. Algunas veces merece la pena la compra de herramientas de desarrollo tales como los simuladores software para algunos DSP y ejercitar el código antes de comprometerse a un solo DSP.

CONCLUSIONES

Los DSP poseen arquitecturas especialmente diseñadas para acelerar los cálculos matemáticos intensos utilizados en la mayoría de sistemas de procesamiento de señal en tiempo real. Se ha visto que el DSP está muy estrechamente ligado al tipo de aplicación. La tendencia es que vayan apareciendo DSP con arquitecturas que estén cada vez más adaptadas a las particularidades de las diferentes aplicaciones. En este sentido, aunque a nivel de prestaciones varios DSP puedan reunir los requisitos necesarios exigidos por una aplicación, otras consideraciones como el coste o el consumo pueden ayudar a disminuir el número de posibles candidatos.

Por otro lado, la reciente aparición de compiladores realmente eficaces en extraer el paralelismo de un programa ha propiciado la recuperación de las arquitecturas VLIW y con ello la obtención de DSP de elevadas prestaciones. La arquitectura VLIW posee una gran ventaja frente a los procesadores superescalares, y es que la extracción del paralelismo se realiza por el compilador y ello permite dedicar más tiempo para obtener la mejor optimización. En consecuencia, el procesador resulta ser mucho más simple y con un número mucho menor de transistores, lo que permite trabajar con velocidades de reloj más elevadas y con un menor consumo.

REFERENCIAS

- [1]. Phil Lapsley, Jeff Bier, Amit Shoham and Edward A. Lee, "DSP Processor Fundamentals: Architectures and Features", Berkeley, California: Berkeley Design Technology, Inc., 1996.
- [2]. Christopher Inacio and Denise Ombres, "The DSP decision: fixed point or floating?", IEEE Spectrum, vol. 33, no. 9, pp. 72-74, sept. 1996.
- [3]. Nikitas Alexandridis, "Design of Microprocessor-Based Systems", Prentice-Hall, Inc., 1993.
- [4]. Dezsó Sima, Terence Fountain, Péter Kacsuk, "Advanced Computer Architectures. A Design Space Approach", Addison-Wesley, 1997. **ME**