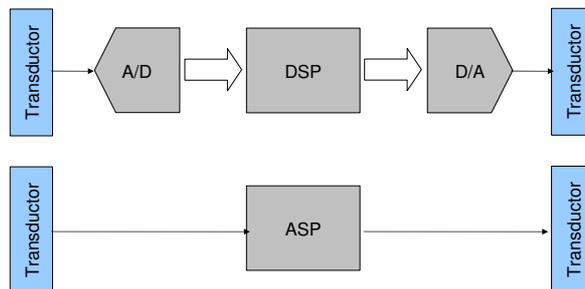


Arquitecturas DSP

Phil Lapsley, Jeff Bier, Amit Shoham, Edward A. Lee
DSP Processor Fundamentals. Architectures and Features
 IEEE Press 1997

Arquitecturas DSP

Sistemas de procesamiento de señales



Arquitecturas DSP

Ventajas de los sistemas digitales

- Pueden resolverse, en forma económica, problemas que en el campo analógico serían muy complicados.
- Insensibilidad al entorno.
- Insensibilidad a la tolerancia de los componentes.
- Predictibilidad y comportamiento repetitivo (consecuencia de las dos anteriores).
- Reprogramabilidad.
- Tamaño constante.

Arquitecturas DSP

Principales aplicaciones

Clases de aplicaciones

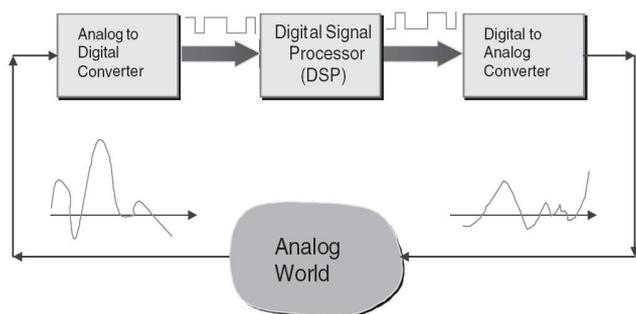
- Embedded de bajo costo (celulares, control de motores HD)
- Aplicaciones de alta performance (algoritmos demandantes)
- PC Multimedia

Problemas típicos

- Demanda de procesamiento en tiempo real con altas frecuencias de muestreo.
- Muestreo y generación de señales analógicas.
- Aplicaciones de bajo consumo y pobres niveles de señal.
- Compresión de datos en tiempo real.

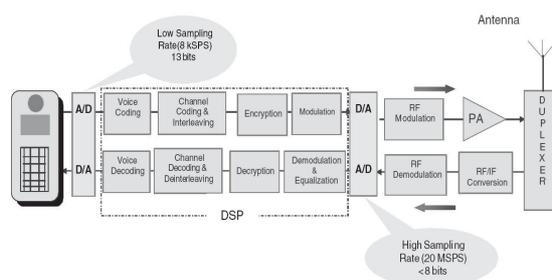
Arquitecturas DSP

Principales aplicaciones

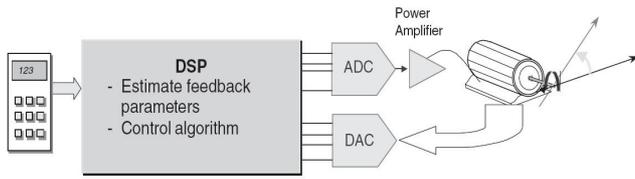


Arquitecturas DSP

Principales aplicaciones

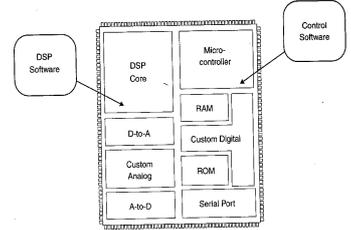


Arquitecturas DSP Principales aplicaciones

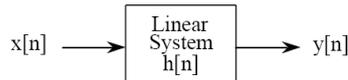


Arquitecturas DSP Características principales

- Algoritmo a ser ejecutado
- Frecuencia de muestreo f_m
- Frecuencia de clock f_c
- Relación f_c/f_m (cantidad de hardware)
- Tipos de representación numérica (punto fijo versus punto flotante)



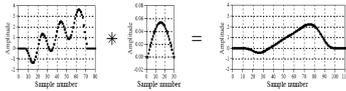
Problemas típicos de DSP Convolución de secuencias discretas



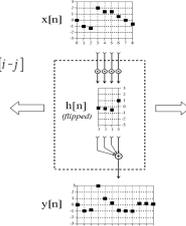
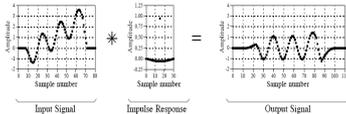
$$x[n] * h[n] = y[n]$$

$$y[i] = \sum_{j=0}^{M-1} h[j]x[i-j]$$

a. Low-pass Filter



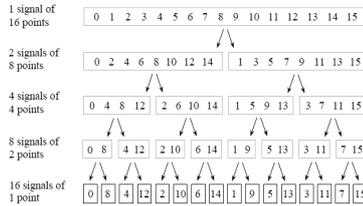
b. High-pass Filter



Problemas típicos de DSP DFT, FFT

$$ReX(\omega) = \sum_{n=-\infty}^{\infty} x[n] \cos(\omega n)$$

$$ImX(\omega) = - \sum_{n=-\infty}^{\infty} x[n] \sin(\omega n)$$



Sample numbers in normal order

Decimal	Binary
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Sample numbers after bit reversal

Decimal	Binary
0	0000
8	1000
4	0100
12	1100
2	0010
10	1010
6	0110
14	1110
1	0001
9	1001
5	0101
13	1011
3	0011
11	1011
7	0111
15	1111

Problemas típicos de DSP Suma de productos

La operación típica de los algoritmos más frecuentemente utilizados en Procesamiento Digital de Señales es la **suma de productos**. Los Procesadores Digitales de Señal están optimizados para realizar estas operaciones en la forma más eficiente. Esto tiene implicancias directas en la arquitectura física del procesador, y en el modelo de programación asociado.

$$y[i] = \sum_{j=0}^{M-1} h[j]x[i-j]$$

Una forma de acelerar su ejecución podría ser acceder simultáneamente a los operandos y al código de operación. Para ello es necesario disponer de bancos de memoria separados para datos e instrucciones.

Problemas típicos de DSP Suma de productos

Es inmediato ver que la sumatoria representa un lazo. Si este tipo de cálculo es el más frecuente, puede brindarse soporte hardware para realizar lazos cortos más rápido. En algunos casos, por ejemplo para implementar eficientemente la FFT, se brinda como modo de direccionamiento el bit-reversal.

Si consideramos que en $y[i]$ se acumulan los resultados parciales de la sumatoria, el cálculo se mapea bien en una arquitectura de tipo acumulador. Por otra parte, si la mayor parte del tiempo el procesador estará realizando lazos de este tipo, es razonable que el multiplicador y el shifter se encuentren cableados directamente en el camino de datos (favorecer el caso más frecuente), y que exista una instrucción MAC (multiply and accumulate) que realiza la multiplicación de dos registros y suma el resultado al acumulador. Esto evidentemente va en detrimento de otro tipo de programas: los DSP no son procesadores de propósito general.

Problemas típicos de DSP

Impacto en la arquitectura

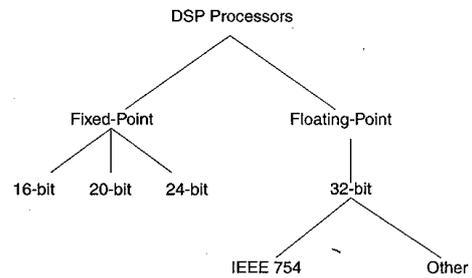
- **Representación numérica** -> Punto fijo [-1,+1] vs. punto flotante.
- **Aritmética** -> Datapath incluye acumulador, shifter normalizador, saturación y redondeo.
- **Sistema de memoria** -> Harvard, multiple access.
- **Acceso a los datos** -> AGU (address generation unit) y modos de direccionamiento especiales (circular, bit reversed).
- **Control** -> Set de instrucciones especializado (MAC, loop, etc.).

Incluyen **segmentación profunda**, características **superescalares** y **VLIW**

Compiladores (C)

Arquitectura de DSP

Operaciones aritméticas y el datapath



Arquitectura de DSP

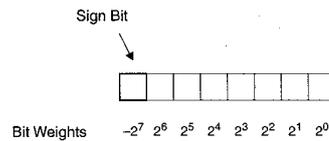
Operaciones aritméticas y el datapath

Los procesadores de **punto fijo** son los mas baratos y, comparativamente los mas rápidos. Sin embargo, son los que mas inconvenientes acarrear al programador, ya que éste debe tener en consideración su menor **rango dinámico**. En un procesador de punto fijo los números están representados ya sea como **enteros** o como **valores fraccionales**.

Los procesadores de **punto flotante** son, por el contrario, caros y en general mas lentos. Su mayor rango dinámico hace que en general no sea necesario preocuparse por problemas de saturación o de overflow.

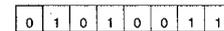
En general, todos los DSP modernos, ya sean de punto fijo o flotante, poseen un ALU entera dedicada al cálculo de direcciones efectivas (AGU: Address Generation Unit).

Representación Entera Binaria



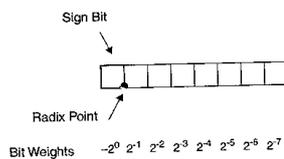
To determine the equivalent decimal value, add up the bit weights for each bit that is a "1."

Example 1:



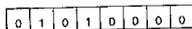
$$= 2^6 + 2^4 + 2^1 + 2^0 = 64 + 16 + 2 + 1 = 83$$

Representación Fraccional



To determine the equivalent decimal value, add up the bit weights for each bit that is a "1."

Example 1:



$$= 2^{-1} + 2^{-3} = 0.5 + 0.125 = 0.625$$

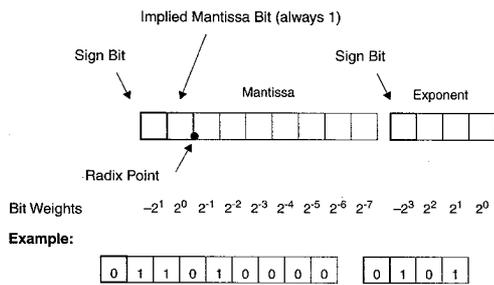
Representación Fraccional

En la representación fraccional, los números (desde el punto de vista de un filtro, por ejemplo, muestras de las señales y coeficientes) son siempre menores a la unidad, por lo tanto el producto **no puede tener overflow** (solo puede aparecer en las sumas). Limitar el overflow es importante por la **fase** del resultado. Por otra parte, siempre será necesario **escalar** adecuadamente a fin de obtener la salida deseada. En general esto se hace mediante simulaciones.

Los números fraccionales se multiplican como si fueran enteros. Esto quiere decir que se necesitan 2n bits en el acumulador para expresar el producto de dos números de n bits. Además se requieren algunos bits adicionales si se quiere prevenir el overflow en las sumas. Por ejemplo:

DSP56000: registros de 24 bits – Acumulador de 56 bits (24+24+8)
 DSP96002: registros de 32 bits – Acumulador de 96 bits (32+32+32)

Representación en punto flotante



Mantissa = $2^0 + 2^{-1} + 2^{-3} = 1 + 0.5 + 0.125 = 1.625$

Exponent = $2^2 + 2^0 = 4 + 1 = 5$

Decimal Value = $1.625 \times 2^5 = 52.0$

Rango Dinámico

Es la relación entre el mayor y el menor número de una representación. Supongamos que la longitud de palabra es de 32 bits.

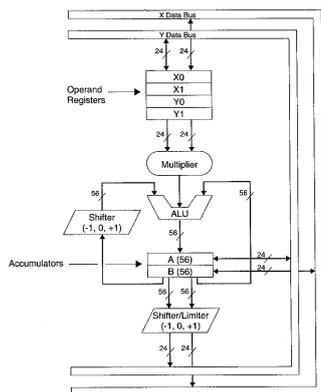
En representación de punto fijo, el número mas pequeño es 2^{-31} y el número mas grande es $1-2^{-31}$. El cociente es aproximadamente 2.15×10^9 , lo que representa unos 187db

En representación de punto flotante (24 bits de mantisa y 8 de exponente), el número mas chico es 5.88×10^{-39} y el mayor 3.40×10^{38} , lo que da un rango dinámico de 5.79×10^{76} , unos 1535db.

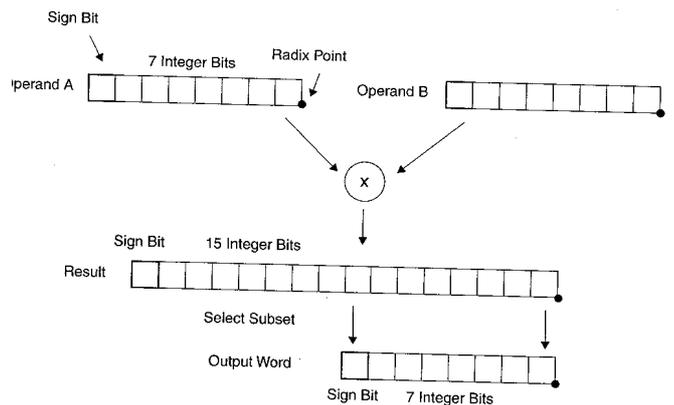
La necesidad de rango dinámico está impuesta por la aplicación. En telecomunicaciones suele alcanzar con 50db. Audio de alta fidelidad requiere unos 90db.

Un costo adicional aparece si la memoria debe estar también en formato de punto flotante.

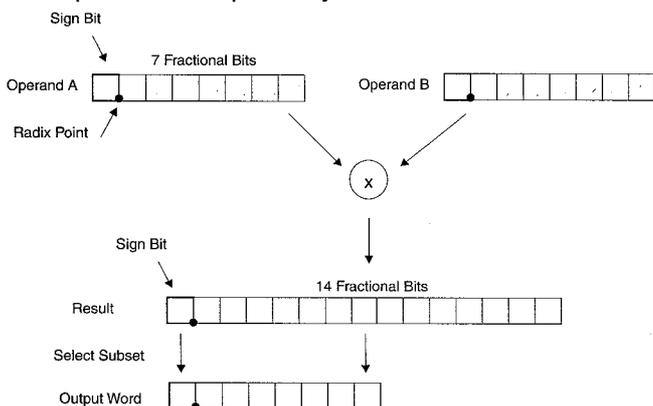
Datapath típico en punto fijo



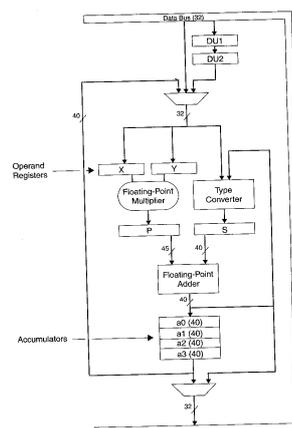
Multiplicación entera



Multiplicación en punto fijo



Datapath de punto flotante



Arquitectura de DSP Subsistema de memoria

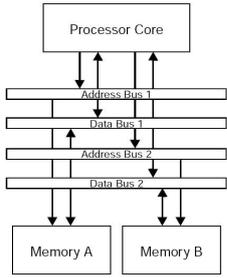


FIGURE 3. A Harvard architecture, common to many DSP processors. The processor can simultaneously access the two memory banks using two independent sets of buses, allowing operands to be loaded while fetching instructions.

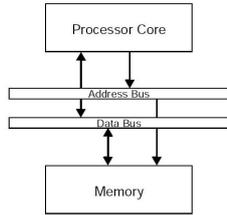
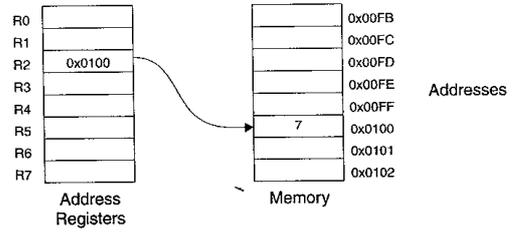


FIGURE 4. The Von Neumann memory architecture, common among microcontrollers. Since there is only one data bus, operands cannot be loaded while instructions are fetched, creating a bottleneck that slows the execution of DSP algorithms.

Arquitectura de DSP Modos de direccionamiento

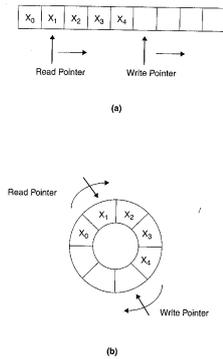
1. Indirecto via registro con post decremento



MOVE (R2), A

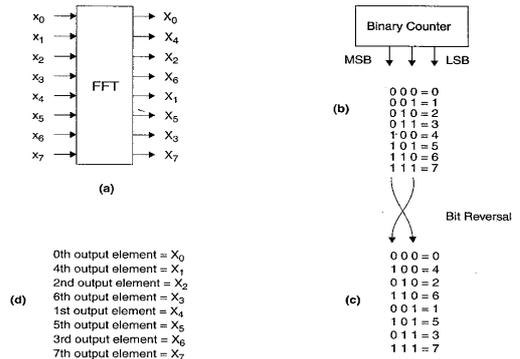
Arquitectura de DSP Modos de direccionamiento

2. Acceso a "buffers" circulares. Direccionamiento "módulo n"



Arquitectura de DSP Modos de direccionamiento

3. Direccionamiento "bit reversal"



Arquitectura de DSP Control

Hardware loops

```

MOVE #16, B
LOOP: MAC (R0)+, (R4)+, A
      DEC B
      JNE LOOP
    
```

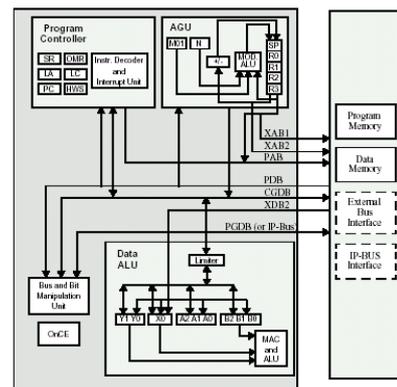
(a)

```

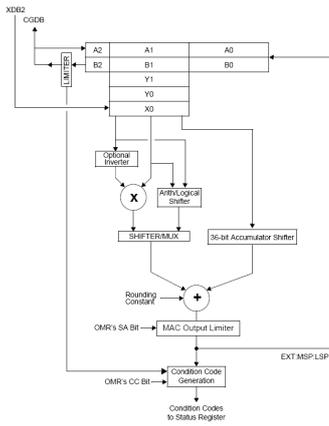
RPT #16
MAC (R0)+, (R4)+, A
    
```

(b)

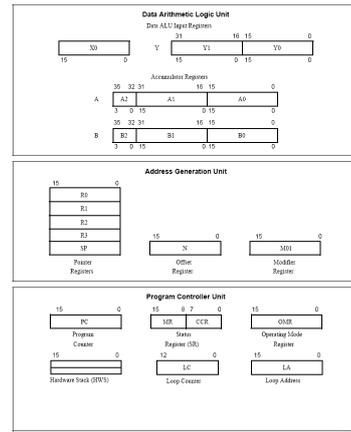
Motorola DSP56F800



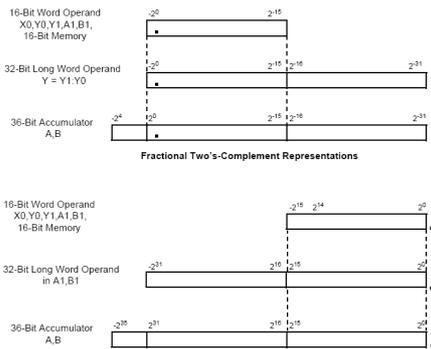
Alu de datos 56800



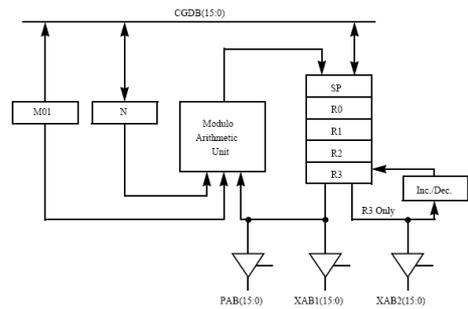
Modelo de programación 56800



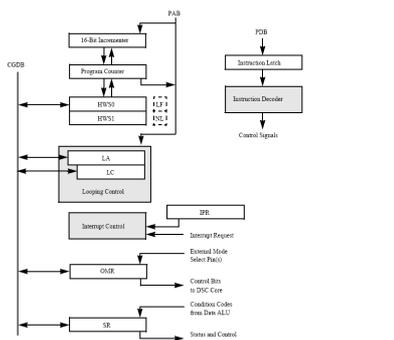
Registros de datos 56800



Alu de direcciones (AGU) 56800

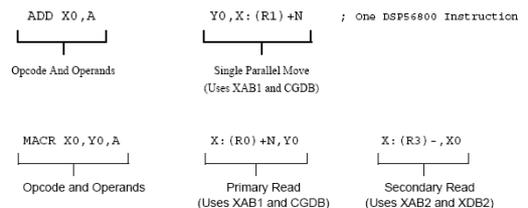


Unidad de Control 56800



Arquitectura de DSP

Instrucciones paralelas



Texas TMS320C62xx

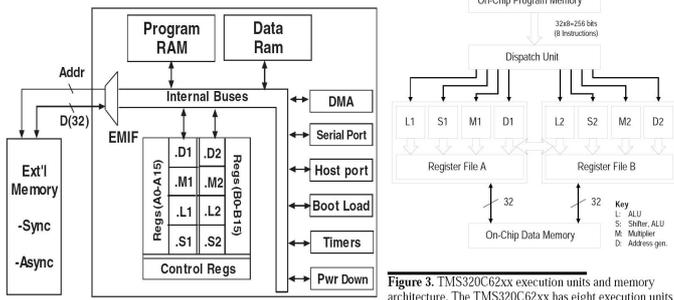


Figure 3. TMS320C62xx execution units and memory architecture. The TMS320C62xx has eight execution units, grouped in two sets of four.

