PROCESSADORES DIGITAIS DE SINAIS (DSPs) E SUAS APLICAÇÕES

Rodrigo Coura Torres
Laboratório de Processamento de Sinais (LPS)
COPPE / UFRJ
torres@lps.ufrj.br

Roteiro da Apresentação

1. Processamento Digital de Sinais
2. *Digital Signal Processors* (DSP)
3. Família Sharc e TigerSharc
4. Ferramentas de Desenvolvimento
5. Multiprocessamento com DSPs
6. Fases de um Projeto com DSPs
7. Aplicações

Processamento Digital de Sinais

- É a ciência que estuda as regras que governam o comportamento de sinais discretos, bem como os dispositivos que os processam.
- Surge como uma alternativa ao processamento analógico de sinais, apresentando diversas vantagens.
- Para que o sinal seja processado digitalmente, necessita ser, primeiramente, discretizado.

Tipos de Processamento Digital de Sinais

- *Offline*
  - Não existe restrição de tempo.
  - Os dados a serem processados já foram previamente armazenados.
  - Possibilidade de implementação de sistemas não causais.
- *Online*
  - Os dados são apresentados ao processador, mas o mesmo não precisa terminar o processamento do dado antes que um novo chegue.
  - Necessidade de um elemento de memória.
- *Real Time*
  - Tempo de processamento crítico.
  - O processamento de um dado tem que terminar antes que um novo chegue.

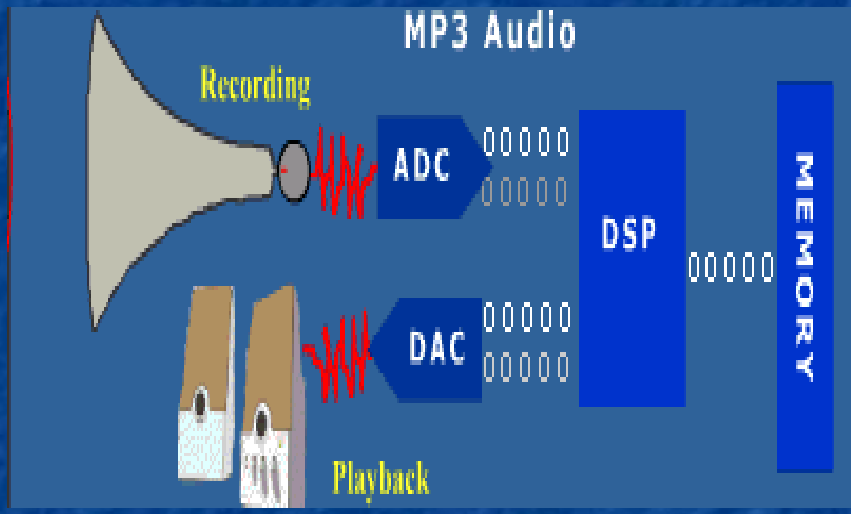
DSP: Aplicação

O processamento digital de sinais é tipicamente usado para:

- Implementar algoritmos de intenso cálculo matemático.
- Operações em tempo real, as quais requerem que o processamento mantenha o fluxo dos sinais de entrada e saída: deve processar os sinais enquanto a tarefa se desenvolve.
- Sistemas flexíveis e adaptativos.

DSP: Exemplo

- Na fase de gravação, o sinal de áudio analógico é convertido para um sinal digital por meio do ADC.
- O DSP recebe este sinal e realiza a codificação MP3, salvando-a em memória.
- Na reprodução, o sinal é decodificado pelo DSP e convertido para analógico, através do DAC.
- A saída pode então alimentar a caixa de som.
- O DSP ainda pode realizar funções como controle de volume, equalização, interface com o usuário, etc.



Vantagens do Processamento Digital

- Programabilidade: um filtro digital pode ser reprogramado, passando de um passa-baixa para um passa-alta, sem troca de hardware.
- Em alguns casos, nem é preciso se reprogramar o dispositivo: podemos apenas melhorar a sua performance.
- Exemplo: guiar mísseis, quando situações de combate podem mostrar deficiências não encontradas durante os testes de simulação. A alteração pode ser feita pela simples troca de um dispositivo de memória, por exemplo.

Vantagens (2)

- Estabilidade
  - Melhoram a confiabilidade.
  - Reduzem os efeitos de envelhecimento de componentes.
  - Reduzem os efeitos de desvio de características com a variação da temperatura.
  - Podem ser programados para detectar e compensar variações das partes analógicas e mecânicas de um projeto completo.

Vantagens (3)

- Redução de custos
  - Reduzem os requisitos de hardware (devido a programabilidade).
  - Reduzem a necessidade de partes de precisão.
  - Reduzem a quantidade de CIs (integração).
  - Reduzem o tempo de desenvolvimento (ferramentas de desenvolvimento, suporte de projeto).

Vantagens (4)

- Facilidades de implementação de algoritmos adaptativos
  - Controle por filtro adaptativo (cancelamento de ruído).
  - Um DSP pode se adaptar facilmente a mudanças nas variáveis de interesse.
  - O algoritmo adaptativo simplesmente calcula os novos parâmetros e os armazena na memória, apagando os antigos valores.

Vantagens (5)

- Realização de funções especiais
  - Filtros de fase linear.
  - Reconhecimento e síntese de voz.
  - Implementação de códigos de correção de erro.
  - Transmissão e armazenamento de dados.
  - Compressão de dados
    - Sem perdas
    - Com perdas (compactação).

Hardwares para Processamento Digital de Sinais

- Microprocessadores
- FPGAs
- Microcontroladores
- DSPs (*Digital Signal Processors*)

Microprocessadores

- Rodam diversas aplicações.
- Otimizados para grandes aplicações.
- Significativo gerenciamento de memória.
- Uma operação por vez.
- Arquitetura CISC *Complex Instruct Set Computers*
  - Blocos computacionais básicos: ALU, *SHIFTER.*
  - Operações como soma, subtração e movimento de dados são facilmente realizadas em poucos ciclos de *clock.*
  - Instruções mais complexas, tais como multiplicações e divisão são construídas de séries de operações simples de adição, deslocamento ou subtração.
  - Instrução de multiplicação pode ser *hard coded in on-chip ROM*, mas leva vários ciclos de *clock.*
  - *A += BC* não é muito adequada, pois não é eficientemente implementada em microcomputadores de uso geral.

FPGAs

- Composta por portas lógicas básicas, como portas *AND,* OR, *INV* e *shift registers*.
- São extremamente rápidas, uma vez que o tempo de apresentar um resultado é igual ao tempo de propagação da entrada pelo integrado.
- A programabilidade é relativamente simples, mas pode tornar-se bastante complicada para processamentos mais complexos.

Microcontroladores

- Dedicados a uma única aplicação e nela inseridos.
- Comparam um sinal externo a um valor conhecido e, então, realizam um movimento de dados ou uma função de chaveamento para controlar um periférico.
- Possui uma unidade lógica e aritmética simples, que realiza todo o processamento.
- Não possui dispositivos internos de otimização.

DSPs

- Realizam funções matemáticas de alto nível.
- Realizam múltiplas operações por ciclo.
- Adequados a aplicações mais complexas, com requisitos matemáticos em tempo real.
- Possuem diversos dispositivos internos de otimização do processamento.

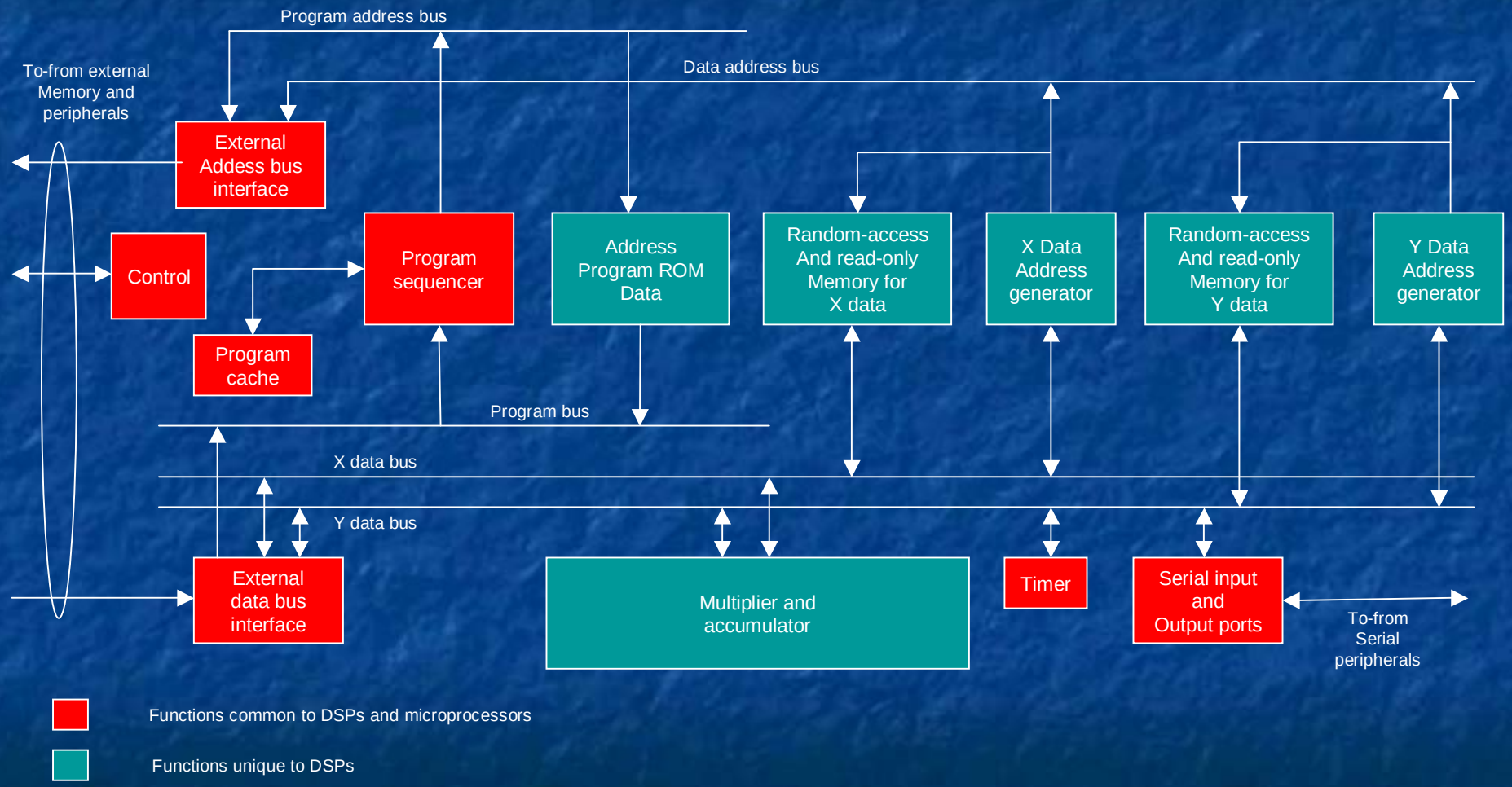
O que é um DSP?

- DSP significa *Digital Signal Processor.*
- Processadores cujo hardware, software e conjunto de instruções são otimizados para aplicações de processamento numérico de alta velocidade.
- São microprocessadores em um único CI que são especialmente projetados para manipular sinais digitais, de acordo com um algoritmo fornecido pelo usuário.
- O objetivo dos DSPs é realizar o máximo de processamento possível antes que um novo dado tenha que ser manipulado.
- O DSP realiza operações como a acumulação de somas parciais resultantes de múltiplos produtos (produtos internos) mais rapidamente que um microprocessador de uso geral.

O que é um DSP (2)

- A arquitetura do DSP é concebida para explorar a natureza repetitiva do processamento digital de sinais, por meio de um *pipeline* do fluxo de dados, de modo a ganhar velocidade.
- Hoje, o engenheiro que deseja tirar partido da capacidade computacional dos DSPs pode escolher de uma lista de dispositivos que cresce sem fim.
- A escolha, porém, requer atenção: capacidade de processamento, endereçamento, precisão aritmética e performance (*benchmarking*) devem ser levadas em conta. O suporte de projeto e a qualidade das ferramentas de desenvolvimento de hardware e software para o dispositivo também são extremamente importantes na decisão.

DSP: Visão Geral Interna



DSP: Tempo Real

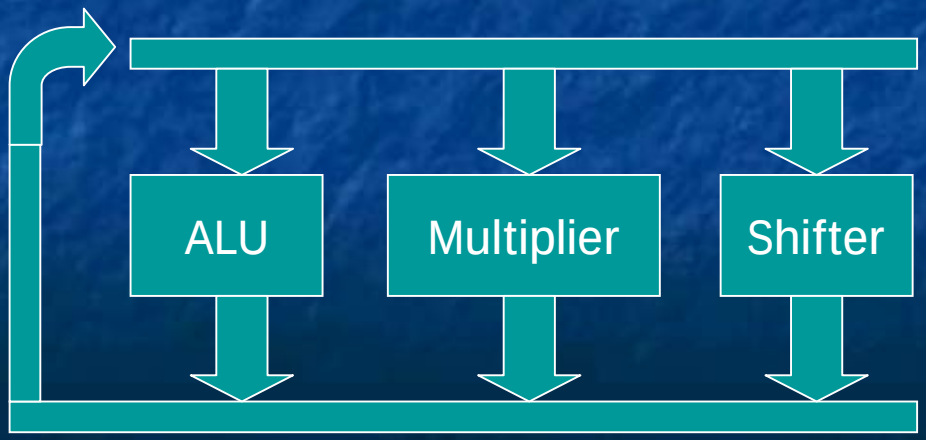
- O sinal chega ao DSP na forma de amostras (ADC).
- Para uma filtragem em tempo real, o DSP deve completar todo o processamento matemático e as operações necessárias sobre uma dada amostra antes que a próxima chegue.
- Em aplicações complexas, alta velocidade e acurácia são essenciais.

Arquitetura Básica

- Unidades computacionais paralelas.
- Memória interna e múltiplos barramentos.
- Geradores de endereços.
- *Cache* de dados e instruções.
- *Pipeline.*
- Canais e DMA.
- Portas externas.
- Portas seriais.
- *Timers.*
- Set especial de instruções.
- *Interface* JTag para emulação.

Unidades Computacionais Paralelas

- Os DSPs possuem unidades separadas para operações lógicas e aritméticas, multiplicação e deslocamento.
- A saída de qualquer unidade pode ser a entrada de outra no ciclo seguinte.
- Arquitetura otimizada para multiplicações com acumulação.



Pipeline

- Consiste em quebrar a cadeia de processamento em blocos independentes *(Fetch, Decode* e Execute*),* separados por registradores.
- Permite que a CPU comece a executar a próxima instrução antes de terminar a instrução atual.
- A principal desvantagem é o efeito “gargalo” ocasionado no início da execução e após alguma descontinuidade do código.

Sem *Pipeline*

Etapa / Ciclo	1	2	3	4	5	6
Fetch	A			B		
Decode		A			B	
Execute			A			B

Com *Pipeline*

Etapa / Ciclo	1	2	3	4	5	6
Fetch	A	B	C	D	E	F
Decode		A	B	C	D	E
Execute			A	B	C	D

Cache de Dados e Instruções

- São elementos de memória que ficam próximos as unidades computacionais, de forma que seus conteúdos sejam acessados rapidamente (sem *overhead*).
- Armazenam dados e instruções que são utilizados com muita freqüência pelo DSP, otimizando o processamento interno.

Memória Interna Dividida e Múltiplos Barramentos

- Possuem uma memória de dados e uma de programa. Cada uma com geradores de endereço e barramento próprios.
- Diferente dos computadores do tipo PC, que usam a arquitetura de memória de Von Neuman, os DSPs utilizam a arquitetura *Harvard*, com barramentos independentes para dados e instruções.
- Com isso, pode-se acessar simultaneamente um dado na memória de dados e uma instrução na memória de programa.
- Se a instrução a ser utilizada estiver no *cache* de instruções, o barramento de instruções pode ser utilizado para o acesso a um dado que possa estar na memória de programa.

Geradores de Endereços

- São utilizados para calcular o endereço do próximo dado (ou instrução) a ser acessado.
- Implementam, sem *overhead*, funções especiais como:
  - *Buffers* circulares.
  - Iteradores em *Hardware.*
  - Operação de Bit-Reverso.

Unidades de DMA *(Direct Memory Access)*

- Unidade interna do DSP que é responsável por realizar transferência, sem intervenção da unidade de processamento, entre a memória e outros recursos do DSP (portas externas, portas seriais, etc).
- Permitem manter o processador núcleo operando sobre um dado, enquanto novos dados são transferidos para a memória interna, para que estejam disponíveis para o processador núcleo no próximo ciclo de processamento.

Portas Externas

- Permitem o interfaceamento do DSP com dispositivos externos como:
  - Memórias.
  - Outros DSPs.
  - Dispositivos customizados.
  - Um computador hospedeiro.

Portas Seriais

- Possibilitam acesso a recursos externos de comunicação serial como:
  - CODECs.
  - Dispositivos customizados.
  - Outros DSPs.

Timers

- Dispositivo que armazena um valor inicial, que é decrementado a cada ciclo de *clock.*
- Quando o valor chega a zero, uma interrupção é gerada, podendo ser tratada por algum dispositivo interno do DSP.
- A valor original é então restaurado, e uma nova contagem se inicia.
- Assim, processamento que precisem de um período fixo podem ser facilmente implementados.

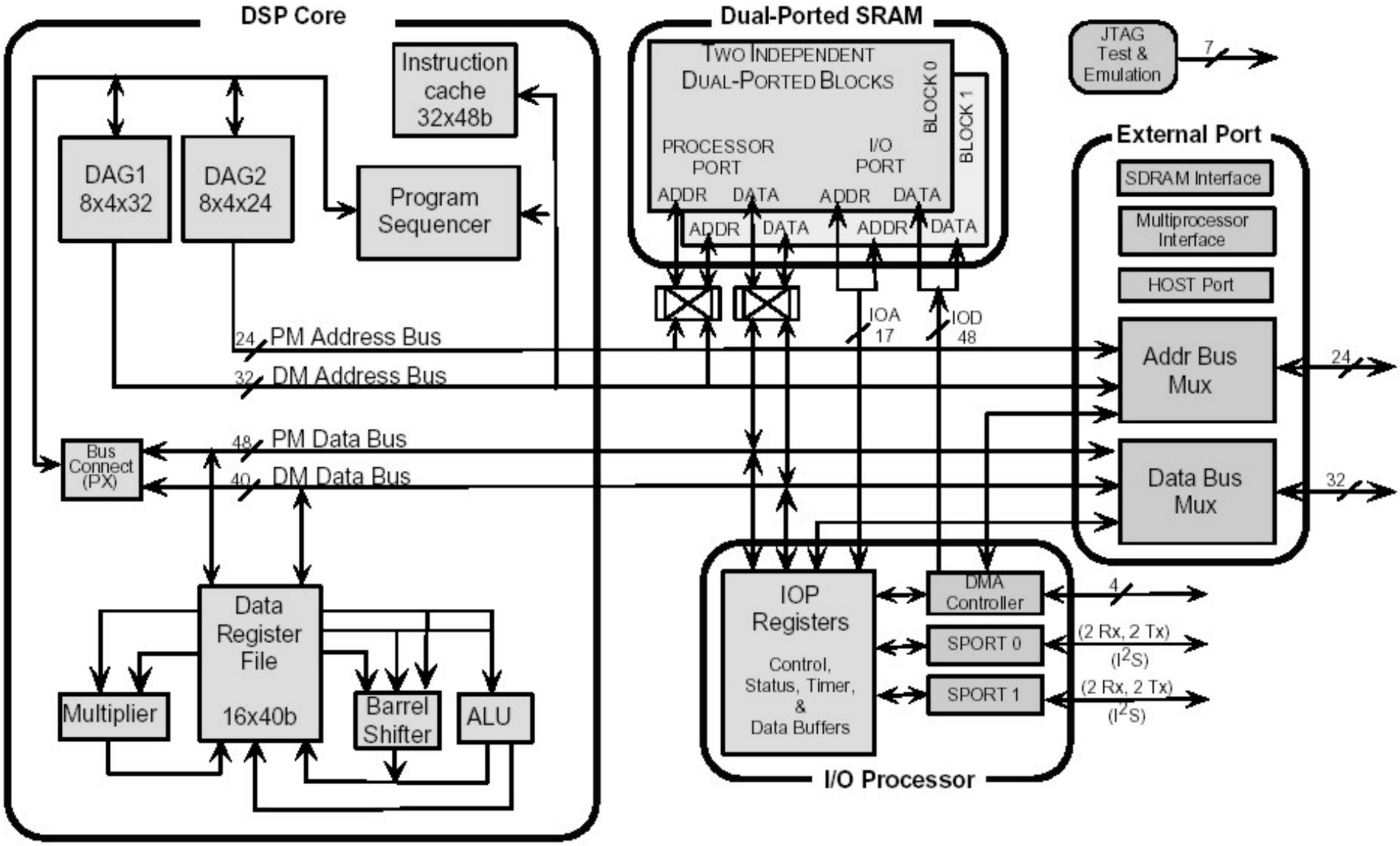
Set de Instruções Especiais

- DSPs possuem instruções próprias para tarefas típicas de processamento digital como multiplicação e acumulação, exponencial, raiz quadrada, entre outras.
- Com isso, enquanto um computador tipo PC gasta em média 10 ciclos para fazer uma soma com acumulação, um DSP pode realizar a mesma tarefa em um único ciclo.

Interface JTag para Emulação

- Sistema padrão IEEE para emulação de sistemas com DSPs.
- Baseia-se num protocolo interno do DSP que possibilita o total acesso a estrutura interna do DSP.
- Assim, é possível ler conteúdo de registradores, memórias, e até mesmo alterá-los, facilitando posteriores *upgrades* de um sistema de processamento digital.

Estrutura Interna: visão geral



Exemplo de Programação: rotina de organização de vetores.

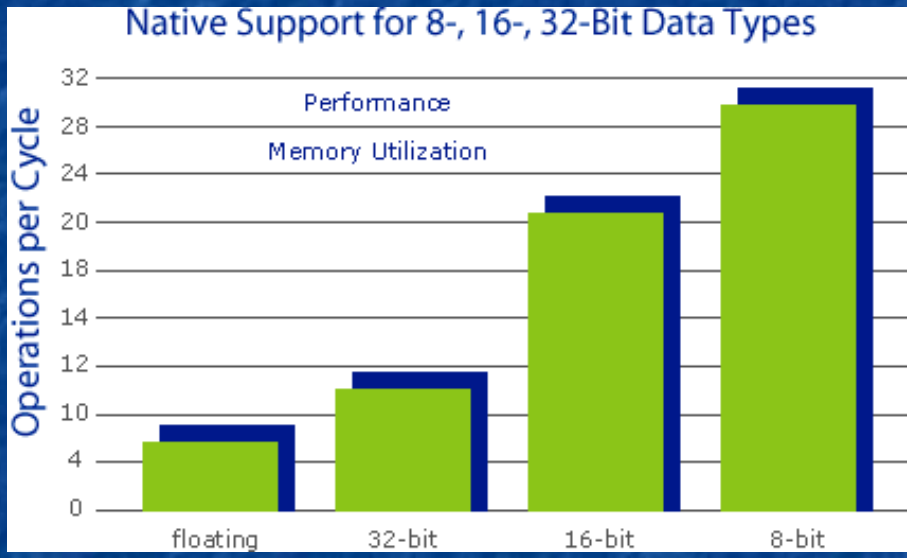
```
; Esta função pega um vetor com 512 amostras e joga as amostras de índices pares para o vetor de nome Ch0, e
      as de índices ímpares, para o vetor Ch1.

; Constantes da função.
NSAMPLES .set    512  ; Tamanho do vetor de entrada.
DATA_ADDR .set   0x03000 ; Endereço do vetor de entrada.
Ch0_ADDR .set   0x04000 ; Endereço do vetor correspondente ao canal 1.
Ch1_ADDR .set   0x05000 ; Endereço do vetor correspondente ao canal 2.

; Atribuição de nomes para os registradores.
.asg AR2, Data
.asg AR3, Ch0
.asg AR4, Ch1

; Processamento
Ch0 = #Ch0_ADDR                    ;(2 ciclos).
Ch1 = #Ch1_ADDR                    ;(2 ciclos).

BRC = #NSAMPLES                    ;Contador de iterações. (2 ciclos).
dblockrepeat(end_proc-1)           ;(3 ciclos (delayed)).
Data = #DATA_ADDR                  ;(2 ciclos)
     *Ch0+ = *Data+                ;(1 ciclo)
      *Ch1+ = *Data+               ;(1 ciclo)
end_proc

; Fim do processo. Total: 519 ciclos.
```

Família SHARC (*Analog Devices*)

- Processadores de 32/40-Bit IEEE em ponto flutuante.
- Arquiteturas SISD e SIMD.
- Velocidades entre 40 e 200 MHz.
- Memórias de 544 kBits a 4 MBits.
- Todas as instruções são realizadas em um único ciclo.
- *Buffers* circulares em *hardware*.
- 32 ponteiros de endereços permitindo 32 *buffers* circulares.
- Iteradores em hardware de até 6 níveis, sem overhead.
- *Assembly* algébrico.
- Set de instruções com aritmética condicional, manipulação de bits, divisão e raiz quadrada, entre outras.
- Canais de DMA operando a velocidade do clock, e sem intervenção do processador núcleo.
- Excelentes para aplicações de multiprocessamento.

Família TigerSharc

- Família de altíssimo desempenho.
- Possui as mesmas características da família SHARC.
- Capaz de manusear dados de 1, 8, 16 e 32 bits em ponto fixo, bem como dados em ponto flutuante (32/40 bits).
- Arquitetura VLIW.
- Utilização de *branch prediction.*
- Permite até 4 instruções de 32 bits simultâneas.
- Clocks de 300 a 600 MHz (2400 a 4800 MMACs).
- Memória interna variando entre 6 e 24 MBits.
- Permite utilizar DSPs em situações onde apenas FPGAs eram possíveis.

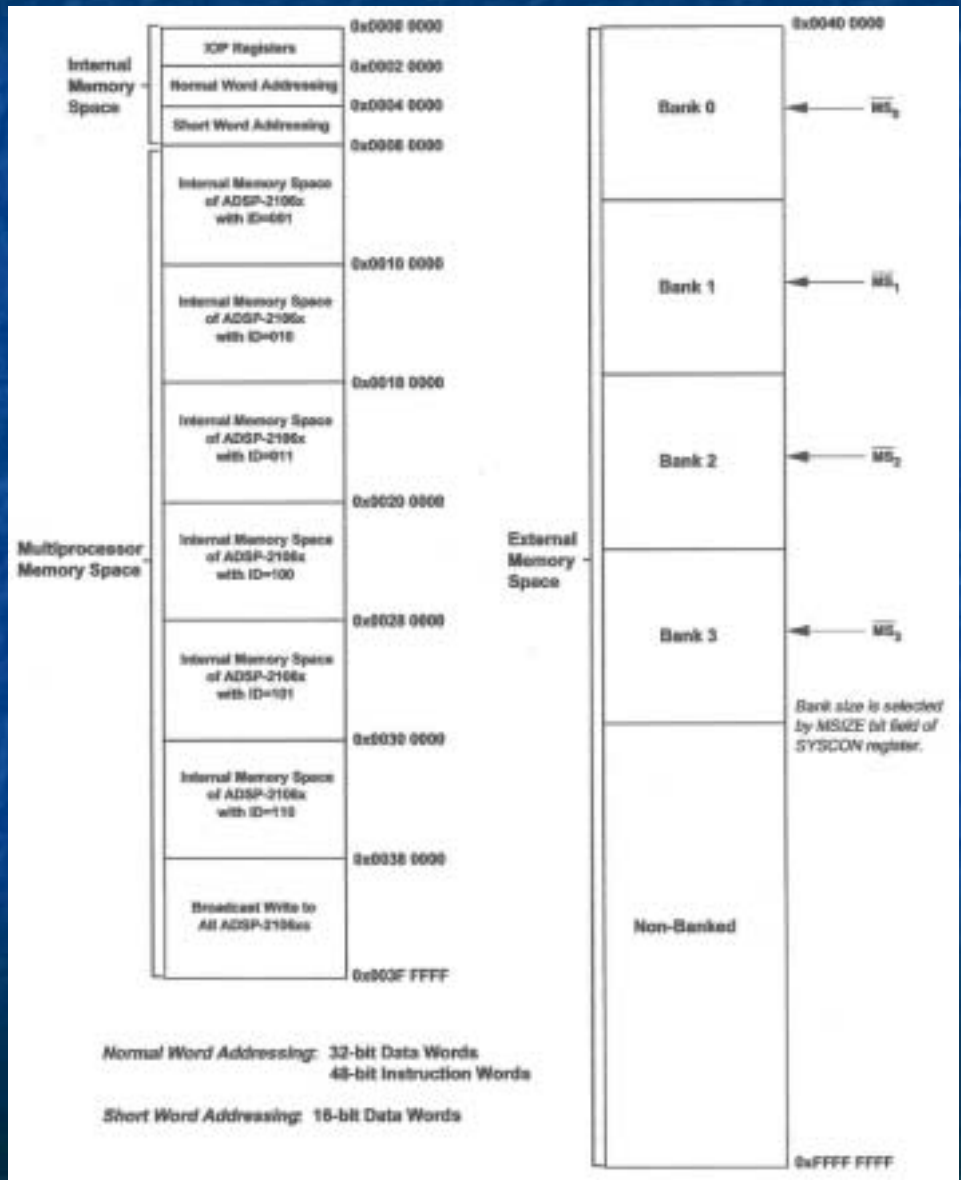


Multiprocessamento com DSPs
Com as Famílias Sharc e TigerSharc

- Divide-se em:
  - Acesso pela memória compartilhada.
  - Portas de ligação.
  - Conexão serial.

Acesso Pela Memória Compartilhada

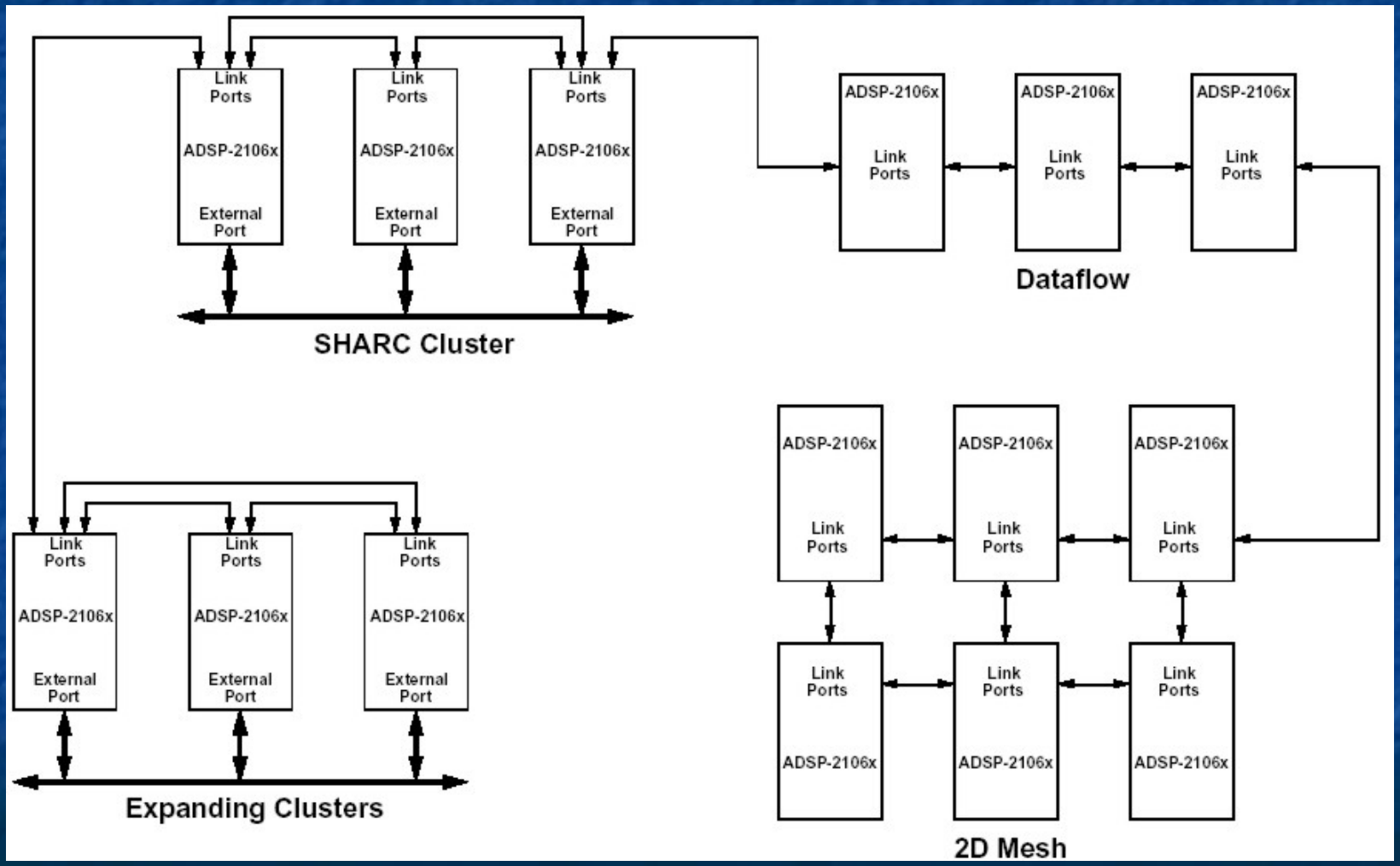
- Sistema baseado na filosofia mestre-escravo, onde um processador, ao ser definido como mestre, pode acessar a memória interna e os registradores de qualquer processador escravo.
- Possibilidade de escrita por irradiação (*broadcast*).
- Memória interna de todos os processadores estão mapeadas em um espaço de memória de processamento.
- A desvantagem são os problemas que sistemas baseados em memória compartilhada apresentam, como baixa configurabilidade e a disputa pelo barramento.



Portas de Ligação

- Fornecem capacidades adicionais de I/O.
- Conexão física entre processadores, podendo implementar topologias paralelas de uma, duas ou três dimensões.
- Possibilita a utilização de memória distribuída, facilitando possíveis *upgrades* no sistema.
- Podem operar simultaneamente e independentemente.
- Seus dados podem ser lidos diretamente pelo processador núcleo, ou pela DMA, para serem transmitidos para a memória interna do DSP.
- Podem ser acessadas pelo processador *host*, permitindo comunicação direta com o DSP.

Exemplo com Portas de Ligação



Conexão Serial

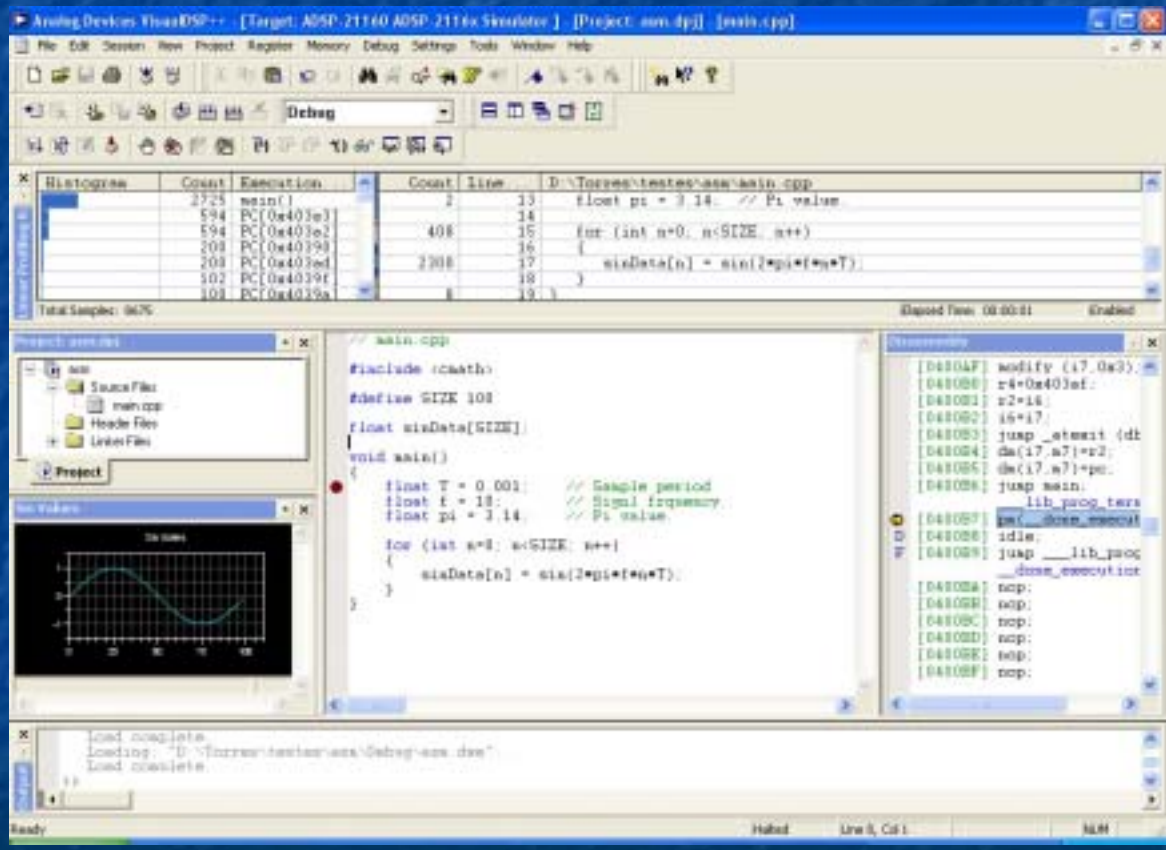
- Permitem ligar o DSP a dispositivos externos, como memória e outros periféricos.
- Pode ser uma maneira simples de expandir um sistema de multiprocessamento.
- Supondo que um determinado problema utiliza uma placa com 4 DSPS e que torna-se necessário aumentar este poder de processamento. Isto pode ser feito adquirindo-se uma nova placa com 4 DSPs e fazendo a interconexão de ambas pelas portas seriais, reduzindo drasticamente o custo e a complexidade deste *upgrade*.

Ferramentas de Desenvolvimento

- Permitem rápido desenvolvimento de projetos.
- Forte participação de empresas parceiras.
- Constituem-se de:
  - IDE
  - Emuladores
  - Placas de avaliação
  - Placas filhas

IDE para Elaboração de Projetos

- Permitem realizar tanto a elaboração do código, bem como o *debug* do mesmo.
- Possibilitam acessos à memória interna do DSP, de forma que é possível apresentar seus valores na forma de gráficos, ou na forma numérica convencional.
- Possibilidade de inserção de *breakpoints* para análise intermediária de execução.
- Possuem opções para *profile*, visando o calculo do tempo de execução das funções do sistema, e posterior identificação de gargalos de execução.
- Permitem também realizar simulações, de forma que se possa avaliar um determinado DSP, sem a necessidade inicial de compra do mesmo.
- Possuem diversas bibliotecas otimizadas em *assembly* como FFTs, filtros, etc.
- A codificação pode ser em C/C++ ou *assembly*.
- Interfaceamento com a controladora JTag do DSP.



Dispositivos de Emulação JTAG

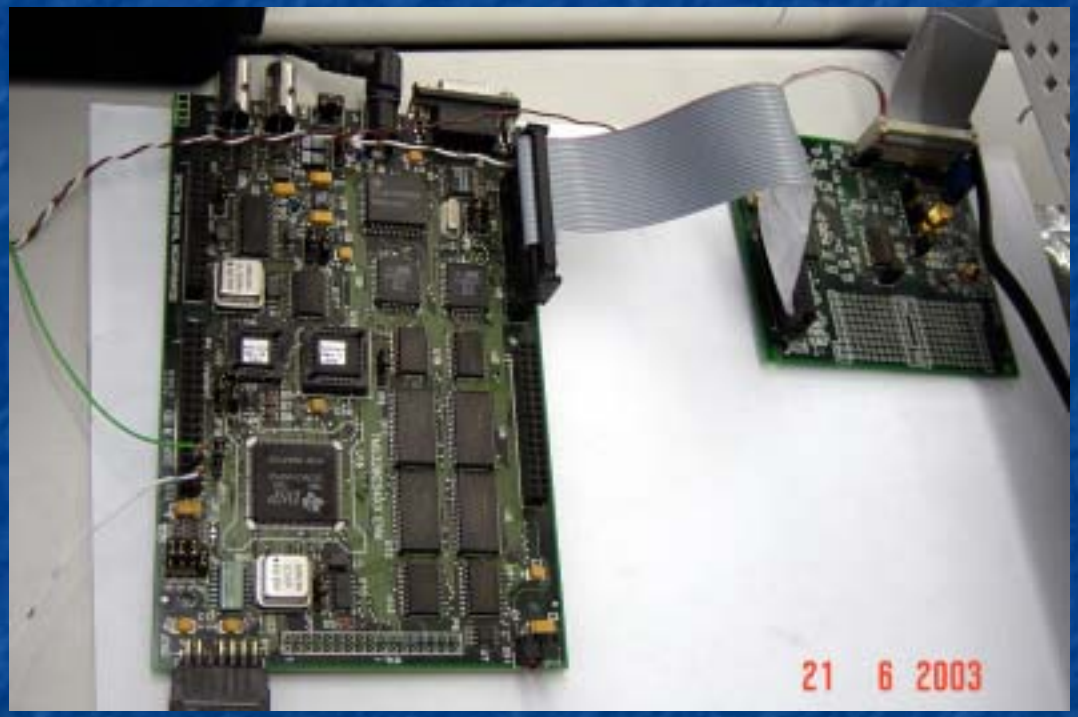
- Interface de comunicação direta com o DSP.
- Permite ler e escrever dados na memória ou registradores do DSP de maneira não-intrusiva, sem interrupção do processamento.
- OS dados lidos podem ser analisados na IDE usada.
- Solução mais eficiente do que a tradicional *In-Circuit Emulation.*

Placas de Avaliação

- Permitem realizar avaliação inicial do DSP escolhido após a simulação.
- Apresenta alguns recursos já conectados ao DSP (conversores AD/DA, memórias externas, etc).
- Apresentam, em geral, conector JTAG para emulação.
- Para permitir conexão com dispositivos externos à placa de avaliação (placas filhas, por exemplo), possuem conectores ligados aos pinos do DSP responsáveis por funções como:
  - Transmissão serial.
  - Acesso a bancos de memória externos.
  - Pinos de I/O genéricos.

Placas Filhas

- Estendem as funcionalidades das placas de avaliação.
- Placas destinadas a uma determinada família possuem compatibilidade pino a pino com a respectiva placa de avaliação.
- Implementam funcionalidades como:
  - Conversores AD/DA rápidos.
  - Bancos de memória externa.
  - Controladores variados (disco, rede, etc).



Fases de Um Projeto com DSPs

- Pesquisa
  - Busca por dispositivos através do conhecimento inicial do problema (ponto fixo / flutuante).
  - Normalmente realizada no site do fabricante.
- Simulação
  - Realizada na ferramenta de desenvolvimento.
  - Permite que se avalie um DSP, sem a necessidade de comprá-lo.
- Avaliação
  - Realizada em placas padronizadas de avaliação fornecidas pelo fabricante, que contém o DSP selecionado na simulação, e recursos que permitem a conexão do mesmo com outros dispositivos.
  - Evita-se assim, a necessidade de confecção de PCBs complexas.
- Emulação
  - Realizado quando o DSP já se encontra embutido no ambiente que operará (uma placa específica dentro de um submarino, por exemplo).
  - Necessita da utilização da *interface JTag.*
  - Permite futuras atualizações de código, visando correção de *bugs* só encontrados nesta etapa.

Projetos com DSPs: DSPs e Computadores

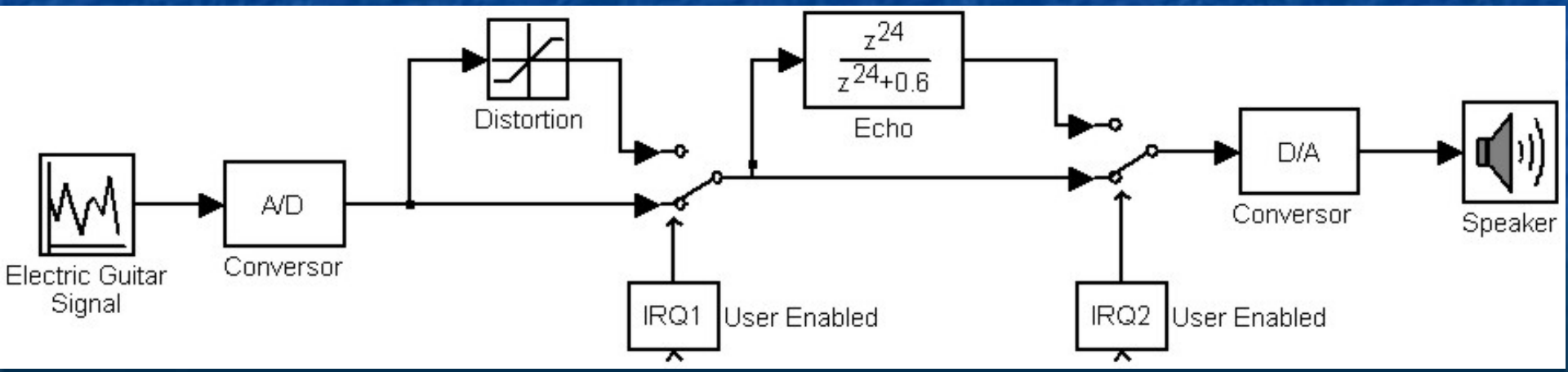
- DSP + hospedeiro: grande capacidade de processamento e poder de decisão
  - Hospedeiro (maneja dados)
    - ✓ Gerencia grande quantidade de dados.
    - ✓ Tem capacidade de tomar decisões complexas.
    - ✓ Facilidade em armazenar/resgatar dados.
    - ✓ Requer: linguagem de alto nível, sistema operacional, aplicativos.

DSPs e Computadores (2)

- DSP (maneja sinais)
  - Gerencia e processa sinais.
  - Realiza intensivamente operações matemáticas.
  - Processa dados conforme eles são produzidos (tempo real).
  - É independente do hospedeiro (*host).*
  - Requer: alta velocidade, *software* escrito eficientemente.

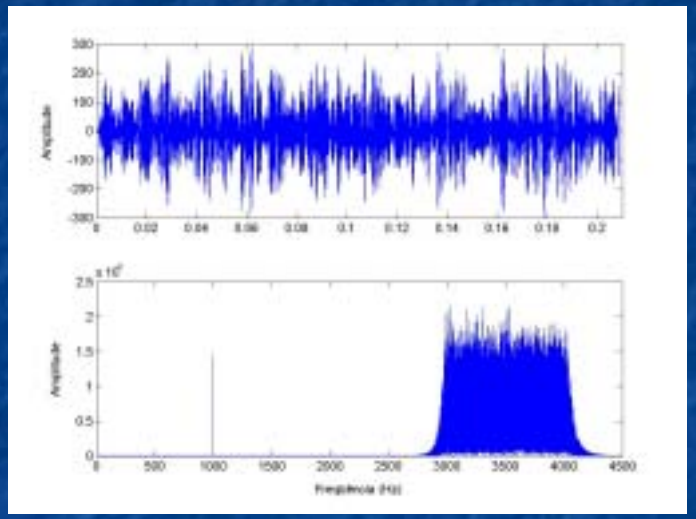
Aplicação de Exemplo: Sistema em Tempo Real de Efeitos para Guitarras

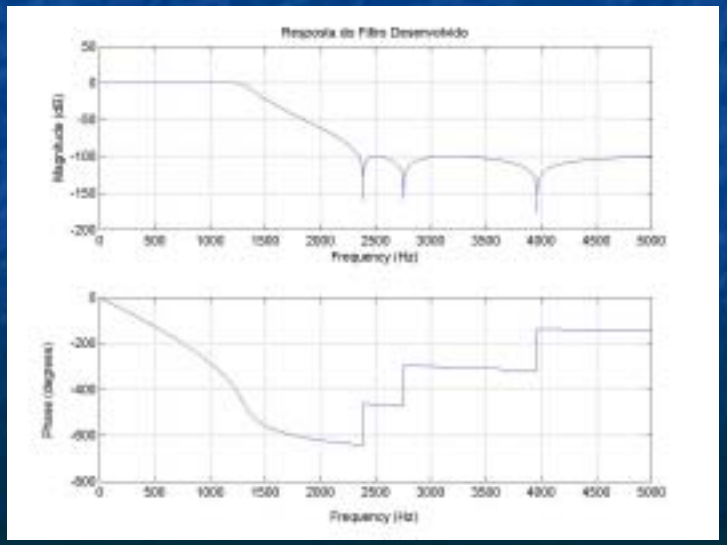
- Sistema operando em tempo real, que implementa dois efeitos conhecidos: distorção e eco.
- A ativação de um efeito é feita por interrupções, que trocam o estado (ligado / desligado) do efeito.
- *Leds* sinalizam ao usuário qual efeito está ativado.

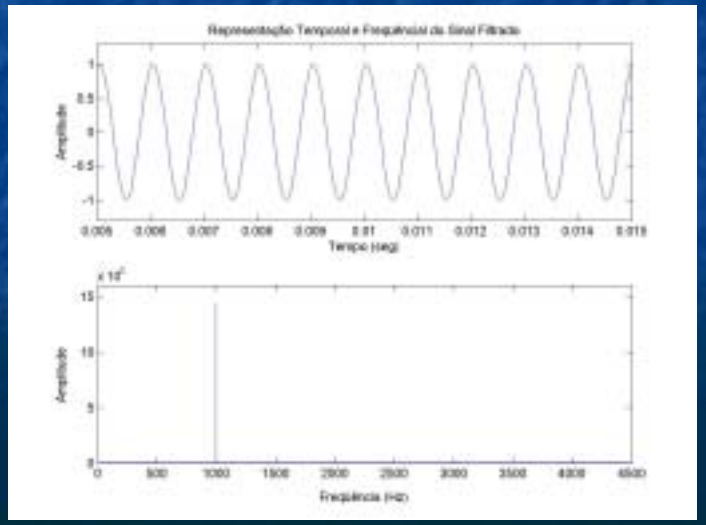


Aplicação de Exemplo: Filtragem em Tempo Real

- Supõe-se a recepção ruidosa de uma senoide de 1kHz.
- Observando-se o espectro, percebe-se que o ruído é limitado em banda, e que não compartilha a freqüência do sinal.
- Um filtro digital do tipo IIR de 8ª ordem pode, então, ser adotado para filtrar em tempo real o sinal recebido.







Aplicação de Exemplo: Detector *Offline* de Múons

- O DSP realiza, neste caso, o papel de emissor e receptor de sinais.
- Um filtro casado realiza a análise do sinal recebido, comparando com uma imagem conhecida, para determinar se um múon foi recebido.
- O controle do emissor é feito através de botões na placa de avaliação, e LEDs indicam a resposta dada pelo receptor.

Sintetizador *Online* de Voz

- Sistema baseado em três partes:
  1) *Interface* de usuário, operando em um PC.
  2) *Interface* de conversão de nível de tensão RS-232 / DSP.
  3) Sintetizador de voz em DSP.
- O DSP recebe o texto a ser sintetizado pela sua porta serial e realiza a síntese do mesmo, enviando as amostras resultantes para o DAC.

Conclusões

- DSPs permitem ao engenheiro eletrônico o desenvolvimento de sistemas em tempo real de maneira fácil e segura.
- Alguns DSPs já apresentam tamanha performance que já podem ser utilizados em projetos dominados até então por FPGAs.
- O crescimento do uso tem favorecido muito a contínua queda nos preços, bem como um crescente número de opções disponíveis.