

# Arquiteturas DSP e Aplicações

Raquel da Silva Cabral e Vilar Fiuza da Camara Neto  
Universidade Federal de Minas Gerais  
Departamento de Ciência da Computação  
{raquelc,neto}@dcc.ufmg.br

## Abstract

*O Processamento Digital de Sinais é uma tecnologia que permite o armazenamento, processamento e reprodução de sinais em representação digital. Geralmente os sinais de interesse são representações de áudio, voz e seqüências de imagens (vídeos). Os Processadores Digitais de Sinais (DSPs) são processadores especializados que encapsulam a funcionalidade necessária para realizar processamento digital de sinais. Em geral são desenhados para trabalhar em tempo real, consumir pouca energia e ocupar um espaço pequeno: assim se tornam adequados no uso de sistemas embarcados.*

*Este trabalho apresenta um estudo sobre as características de projeto da maioria dos DSPs modernos, comparando-as com processadores de outras classes e ressaltando suas peculiaridades. Também são abordados os cenários de aplicações correntes dos DSPs, com ênfase na arquitetura necessária para cada tipo de aplicação.*

## 1 Introdução

Hoje em dia sistemas multimídia e comunicação demandam, mais e mais computações que não são satisfeitas por microprocessadores de propósito geral, quando são considerados, custo e consumo de energia aceitáveis [6].

*Digital Signal Processors* (DSP) são microprocessadores especializados em processamento digital de sinal usados para processar sinais de áudio, vídeo, etc., quer em tempo real quer em off-line. Um dos usos do DSP que chamaram a atenção da mídia foi a proposta do cancelamento de ruídos: através do sistema proposto um dispositivo captaria o ruído ambiente e geraria um “anti-ruído”, com as ondas simétricas: a cada vale corresponderia um pico e vice-versa. Assim poderia se cancelar o ruído de um ambiente, por exemplo, dentro de um automóvel. Em comparação com outros microprocessadores, os DSPs frequentemente tem vantagens em termos de velocidade, custo e eficiência de energia [5].

Este artigo está organizado nas seguintes seções: a Seção 2 descreve as características específicas das arquiteturas dos DSPs em comparação com outras arquiteturas; a Seção 3 apresenta um breve histórico dos DSPs; a Seção 4 discute as aplicações correntes de DSPs e apresenta alguns processadores atuais; e a Seção 5 lança algumas notas conclusivas sobre o trabalho.

## 2 Características dos DSPs

Os processadores podem ser projetados para suprir as necessidades em três grandes áreas distintas [7]:

1. as *aplicações para desktop* exigem performance de programas dedicados a operações aritméticas com números inteiros ou reais. Nesse mercado, a demanda de desempenho é muito maior do que a preocupação com o tamanho do processador ou com o consumo de energia;
2. as *aplicações para servidor* necessitam de performance na execução de muitas tarefas em paralelo e na transferência e tratamento de grandes massas de dados. Operações com aritmética real não são tão importantes, a não ser que o servidor seja específico para o processamento matemático. O consumo de energia também não é uma preocupação;
3. as *aplicações embarcadas* exigem uma abordagem oposta: a economia de fatores como memória, custo de produção, tamanho e energia é o fator mais importante.

Entretanto, é difícil projetar um processador otimizado para cobrir tarefas tão distintas [9]. O desenvolvimento de processadores para aplicações embarcadas requer cuidado especial, pois cada funcionalidade planejada pelo projetista torna o circuito mais complexo, gerando maior necessidade de energia e aumento na área do núcleo. Uma maneira de buscar esse equilíbrio é construir processadores especializados em uma tarefa específica e projetar uma arquitetura

(ISA, largura de endereçamento, *clock rate*, etc.) capaz de suprir as necessidades dessa tarefa.

Nesse sentido, os DSPs são processadores projetados para um cenário bem específico: em geral, o objetivo é tratar matematicamente um fluxo contínuo e infinito de dados de entrada, gerando um fluxo também contínuo de resultados em tempo real com o menor atraso possível.

Embora as necessidades para cada caso sejam bem distintas (por exemplo, as decisões de projeto tomadas no desenvolvimento de um DSP para descompressão de vídeo podem ser muito diferentes das tomadas no caso de um controlador de servo-motor), existem várias características comuns aos DSPs e que os distinguem das demais classes de processadores. Essas características são discutidas nas subseções seguintes.

## 2.1 Instruções aritméticas

Como os DSPs realizam operações aritméticas intensivas sobre um fluxo de dados contínuo, é natural que as operações mais comuns sejam implementadas de maneira bastante otimizada. A multiplicação em ponto flutuante, por exemplo, é executada em um único ciclo em boa parte dos DSPs modernos (o que não ocorre em processadores para *desktops*). Mais importante do que a multiplicação é a operação de *Multiply and Accumulate*, ou MAC: uma única instrução multiplica dois valores e acumula o resultado em um registrador. A operação de MAC é a base para várias operações mais complexas: produto interno, convolução, Transformada Rápida de Fourier, avaliação de polinômios, resposta finita ao impulso, etc. Também é comum encontrar DSPs capazes de sustentar uma operação de MAC por ciclo de *clock*.

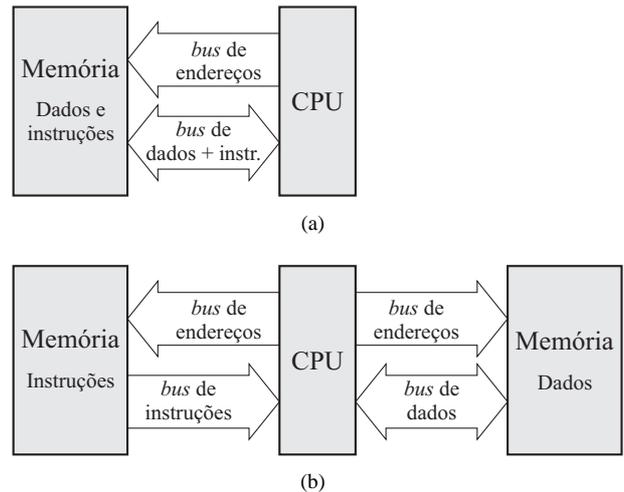
De fato, a operação de MAC é tão utilizada em DSPs que a medida de *benchmark* mais comum é a contagem de MMAC/s (milhões de MACs por segundo). Embora nenhum *benchmark* represente fielmente a performance de um processador, a medida de MMAC/s é bastante utilizada como primeiro parâmetro de comparação entre DSPs;

## 2.2 Representação de números reais

Ao contrário dos processadores *paradesktops*, boa parte dos DSPs adota o modelo de ponto fixo para a representação de valores reais. Esta não é uma regra geral: há DSPs que adotam ponto flutuante ou mesmo os que permitem a operação em qualquer modo; porém, a implementação de aritmética em ponto fixo reduz a complexidade do processador e conseqüentemente o consumo de energia;

## 2.3 Modelo de arquitetura de memória

Os dois grandes modelos de arquitetura de memória são vistos na Figura 1. A arquitetura Von Neumann permite



**Figura 1. Modelos tradicionais de arquiteturas de memória: (a) Arquitetura Von Neumann (memória unificada de dados e código); (b) Arquitetura Harvard (memórias separadas para código e dados).**

uma flexibilidade maior no uso da memória, pois qualquer porção desta pode ser alocada tanto para dados quanto para código. No entanto, essa arquitetura leva a um gargalo estrutural: o canal (*bus*) de comunicação é o mesmo para a carga de instruções e para a transferência de dados entre a memória e o computador, o que o torna um recurso bastante demandado. Como os DSPs são voltados para o processamento contínuo de grandes volumes de dados, a adoção da arquitetura Von Neumann poderia levar a *stalls* causados por ocupação do canal. Já a arquitetura Harvard, que separa a memória de dados da de código (e exige canais distintos para cada memória), evita esses problemas;

## 2.4 Tamanho das palavras

Uma vantagem da adoção da arquitetura Harvard é a liberdade de adotar palavras de instrução e de dados de tamanhos diferentes. De fato, essa flexibilidade é aproveitada em muitos DSPs atuais: o projetista pode estabelecer uma largura adequada para o tipo de dados que serão tratados sem se preocupar com o impacto na densidade de código.

Uma conseqüência interessante dessa liberdade é a adoção de palavras de tamanhos não convencionais — seja, o tamanho da palavra não é uma potência de 2. É comum encontrar DSPs que trabalham com dados de 12 ou 24 bits de largura;

## 2.5 Custo de energia

Já foi dito que o consumo de energia é uma preocupação primordial no projeto de DSPs. Para reduzir a necessidade de energia, os DSPs contam com pelo menos um modo de baixo consumo: o STOP, sinalizado por um dispositivo externo, que faz com que o processador se desative quase completamente e passe a drenar uma energia quase irrisória. Alguns modelos também disponibilizam outros modos, como o WAIT, onde o processador entra em dormência aguardando a chegada de novos dados. Também é possível desativar partes do processador que não estejam em uso: por exemplo, se o programa chavear para a aritmética de ponto fixo, o DSP pode optar por desativar os segmentos das ALUs que tratam com ponto flutuante.

Há uma abordagem paralela aos modos de baixo consumo: é o chamado *underclock*, onde o usuário do DSP configura o gerador de pulsos de *clock* para uma frequência bem menor do que a máxima especificada. Dessa maneira, é possível injetar uma tensão menor de alimentação para o DSP, o que causa uma redução sensível na potência requerida. Essa abordagem é tão comum que os próprios fabricantes apresentam algumas configurações de *clock* reduzido com a tensão e o consumo respectivos.

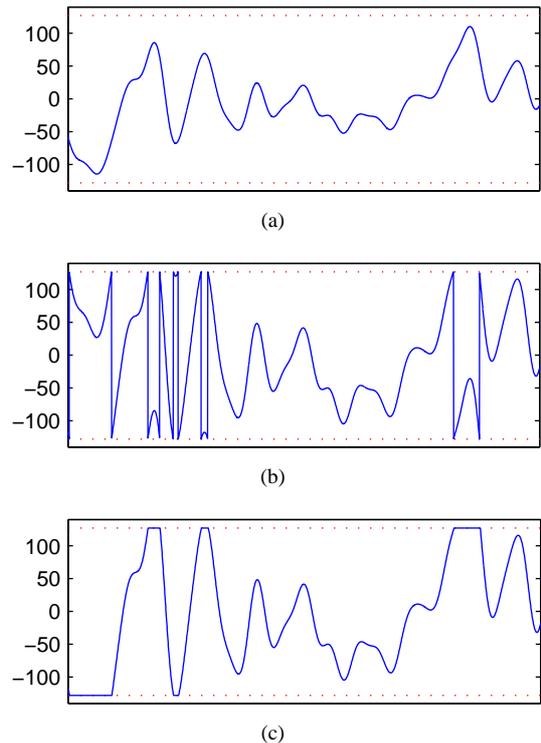
## 2.6 Previsibilidade

Como os DSPs são voltados para aplicações em tempo real, a ocorrência de fatores inesperados (interrupções, *stalls*, *cache misses*, colisões estruturais, etc.) deve ser bem planejada e especificada, pois esses fatores não podem comprometer o fluxo de processamento de dados. Assim, o programador é capaz de trabalhar com alguns parâmetros críticos do sistema, como a taxa máxima de dados de entrada ou o limite de tempo gasto para calcular os dados de saída. É interessante observar que a maioria dos fabricantes especifica o tempo de execução de certas operações “no pior caso”.

## 2.7 Tratamento de estouros

A grande maioria dos processadores para *desktops* implementa aritmética por *complemento de dois*. Um artefato dessa implementação é que operações que estouram a capacidade de representação geram resultados estranhos (por exemplo, a soma de dois valores positivos pode gerar um valor negativo). Um exemplo desse artefato pode ser visto na Figura 2(b): o sinal original (visto na Figura 2(a)) é multiplicado por 2, gerando estouro em alguns intervalos. A aritmética por complemento de 2 faz com que os valores estourados “circulem” dentro dos limites.

No caso dos DSPs, é mais comum a implementação da aritmética com *saturação*: ou seja, quando uma operação



**Figura 2. Exemplos de estratégias de tratamento de estouros aritméticos, com valores limitados ao intervalo  $[-128, 127]$ : (a) sinal original; (b) sinal amplificado em  $2\times$  com estouros tratados por complemento de 2; (c) sinal amplificado em  $2\times$  com estouros tratados por saturação.**

gera um resultado impossível de ser representado, a unidade aritmética retorna o limite de representação mais próximo do resultado desejado. O efeito desse tratamento é visto na Figura 2(c), onde os valores estourados resultantes da amplificação do sinal original são simplesmente limitados à capacidade de representação numérica.

## 2.8 Acumuladores de alta precisão

Em alguns procedimentos matemáticos, como a aplicação do MAC a um conjunto de dados, realiza-se uma série de operações aritméticas cujos resultados são acumulados em um registrador. Naturalmente, cada operação pode gerar pequenos erros de arredondamento — que, embora desprezíveis em cada passo, podem se acumular e comprometer a qualidade do resultado calculado. Para evitar esse problema, muitos DSPs adotam um acumulador de alta precisão, de modo que esses erros acumulados tendem a ser de magnitude muito inferior à capacidade de representação de

uma palavra de dados. Essa solução também pode evitar os problemas de saturação descritos anteriormente;

## 2.9 Tratamento de exceções

No caso dos DSPs, o projetista pode decidir que a manutenção do processamento em tempo real é, em certa medida, mais importante do que o tratamento de exceções. As exceções aritméticas geradas pela ALU são um exemplo interessante: muitos DSPs simplesmente ativam um *flag* que sinaliza a ocorrência de uma exceção; o aplicativo em execução deve checar periodicamente esse *flag* (por exemplo, após uma seqüência de operações) para tratar a exceção, que pode ter ocorrido há muitos ciclos;

## 2.10 Formas de endereçamento

Em geral, os processadores apresentam vários modos de endereçamento (imediato, direto, indexado, etc.) cuja essência é o cálculo do endereço desejado por meio de somas (e eventualmente deslocamentos à esquerda) dos valores fornecidos. As peculiaridades dos DSPs levaram os projetistas a incorporar novas formas de endereçamento para lidar com situações específicas.

Uma dessas peculiaridades deriva do fato de que os DSPs lidam com fluxos contínuos de dados de entrada, o que leva naturalmente à adoção de *buffers* circulares. Por conveniência, o próprio *hardware* já dispõe de implementações desses *buffers*: o programador configura registradores específicos que estabelecem os endereços inicial e final de um *buffer* circular. Uma unidade aritmética específica — a *Address Generator Unit*, ou AGU — se encarrega de tratar os endereços e as aritméticas de ponteiro para que o acesso ao *buffer* seja sempre restringido aos seus limites. Por exemplo: se for configurado um *buffer* circular de 8 palavras de comprimento, uma tentativa de acessar a 21ª palavra retorna a 5ª palavra ( $21 \bmod 8$ ) do *buffer*, de maneira transparente ao programador.

Outro caso particular de endereço refere-se a uma operação específica: a transformada rápida de Fourier (FFT). No cálculo da FFT os dados são acessados em uma ordem específica e bem determinada. Dessa maneira, alguns DSPs dispõem de um modo de endereçamento próprio para endereçar elementos segundo essa ordem.

É interessante notar que todos os modos alternativos de endereçamento podem ser conseguidos sem que sejam implementados em *hardware*, bastando utilizar a ALU para calcular o endereço desejado antes de ser usado. No entanto, o objetivo dos DSPs é otimizar ao máximo essas operações, pois em certos cenários elas são usadas constantemente. A presença da AGU, que não utiliza a ALU principal, livra o processador e o programador do peso de realizar esses

cálculos. Em muitos DSPs, esse endereçamento é transparente no tempo e não causa *stall* no processamento.

## 2.11 Laços de custo zero

Um dos problemas sérios enfrentados pelos projetistas de processadores com *pipeline* é a ocorrência de quebras no fluxo de instruções, causadas por desvios condicionais e incondicionais. Várias técnicas sofisticadas foram desenvolvidas para diminuir o impacto desse problema, mas é inevitável que essas soluções sejam de implementação muito cara em processadores de *desktops* dotados de execução fora de ordem, arquiteturas superescalares, paralelismo de instruções, *pipelines* profundos, etc.

Em algumas famílias de DSPs, o problema não é tão grave porque os projetistas tentam manter a simplicidade do processador: o *pipeline* é curto e os potenciais conflitos de recursos são resolvidos pela multiplicidade de *data paths* e ALUs, permitindo que operações sejam executadas em paralelo sem a ocorrência de *stalls*.

Para resolver parte dos problemas causados por desvios condicionais, uma solução interessante foi criada: os *laços de custo zero*. Há instruções específicas para configurar a execução repetida de uma instrução ou de trecho de código, desde que o número de iterações seja conhecido *a priori* (o que é verdade em alguns trechos de vários algoritmos). O processador é dotado de registradores específicos para manter o controle do número de iterações realizadas, de modo que o fluxo de execução do laço é completamente conhecido: após buscar a última instrução do laço, o processador enfileira novamente a primeira instrução (caso haja iterações pendentes) ou a instrução que segue o laço (caso todas as iterações tenham sido executadas). Como as comparações não são feitas na ALU principal, o impacto do gerenciamento do laço é nulo. Alguns DSPs suportam até mesmo um certo número de laços aninhados.

## 2.12 Acesso direto à memória (DMA)

O mecanismo de acesso direto à memória (*Direct Memory Access*, ou DMA) permite que dispositivos externos ao processador realizem leituras e escritas na memória RAM do DSP sem a intervenção do processador. As vantagens desse mecanismo são grandes, pois o sistema executado no DSP não precisa lidar diretamente com a recepção e o envio de dados para dispositivos externos: a carga de processamento e os recursos (registradores, memória, código, etc.) necessários para a comunicação caem para quase zero e os *stalls* causados por falha na comunicação não impedem que o processador execute suas tarefas.

### 3 Histórico

Em meados de 1980 os primeiros microprocessadores de propósito específico foram introduzidos. Os chips foram projetados baseados na arquitetura *Havard* com barramentos separados para memória de dados e memória de programas. As partes funcionais dos microprocessadores eram um multiplicador, somador e um acumulador, mas estes processadores não eram capazes de realizar operações com valores reais. Estes processadores foram os TMS320C10 da Texas Instruments [1] e o ADSP-2101 da Analog Devices [4].

Uma segunda geração de processadores DSP, que se estende até hoje, surgiu por volta de 1990. Eles estenderam os microprocessadores originais com algumas funcionalidades, tais como *pipelining*, uniades de lógica e aritmética (ALUs) especializadas em operações de multiplicação, a adoção de valores reais de ponto fixo e de mais acumuladores. A vantagem é que os novos processadores são compatíveis com os seus predecessores e também aumentam a velocidade nas operações [10]. Exemplos de processadores dessa geração foram: TMS320C20 da Texas Instruments, o DSP56002 da Motorola e a família DSP16xxx da Lucent Technologies.

Ao longo do tempo, o uso cada vez mais disseminado de DSPs exigiu que as funcionalidades fossem incrementadas, dando ao processador a capacidade de executar tarefas mais rapidamente e com menos consumo de energia. A evolução dos processadores se deu em várias frentes: as tecnologias de miniaturização permitiram a construção de chips mais compactos e de lógicas mais complexas, ao mesmo tempo que reduziram a tensão de operação; a capacidade de representação numérica dos registradores aumentou, sendo possível hoje encontrar DSPs que trabalham com valores de 64 bits e números em ponto flutuante; os *data paths* foram multiplicados para permitir que várias operações de cálculo e transferência de dados sejam executados em paralelo; finalmente, a demanda de processamento exigiu a implementação de mais de uma ALU no mesmo processador e a criação de unidades aritméticas específicas para o cálculo de endereços de memória.

Atualmente, algumas famílias especializadas de processadores combinam em uma única unidade funcionalidades de DSPs e de microcontroladores. Esses processadores são chamados de *Controladores Digitais de Sinais (Digital Signal Controllers* ou DSCs) e ampliam muito a capacidade original dos DSPs, permitindo a realização de tarefas complexas como compressão e descompressão de dados, criptografia e descryptografia, codificação e decodificação de fluxos (*streams*) de dados digitais (MPEG-2/4, MP3, AAC, etc.) e várias outras.

### 4 Aplicações e tecnologias atuais

Esta seção apresenta as grandes áreas de aplicações onde o emprego de DSPs é bastante disseminado. Cada subseção descreve uma grande área, suas peculiaridades tecnológicas e exemplos de aplicações práticas. Também são apresentadas algumas famílias de processadores atuais adequadas a cada área e um breve perfil de suas arquiteturas. Este trabalho não tem a intenção de apresentar um levantamento completo das tecnologias disponíveis para cada ramo de aplicação, mas dar ao leitor uma idéia do *hardware* adequado a cada cenário.

Os preços apresentados referem-se ao valor da unidade em lotes de 10.000 (valores vigentes em abril [3] e outubro de 2006 [2]).

#### 4.1 Controle digital

O cenário mais simples para o uso de DSPs é o chamado *controle digital*. Nesse cenário, tipicamente o DSP recebe como entrada um sinal de controle ou de sensoriamento, realiza um processamento matemático simples e gera um sinal de controle para um dispositivo externo.

O controle digital encontra utilidade em aplicações diversas. Algumas são apresentadas a seguir:

- *Sistemas industriais*: controle de atuadores e manipuladores robóticos; controle de linhas de produção;
- *Sistemas automotivos*: controle de freios ABS, piloto automático, ignição, injeção eletrônica, transmissão automática e continuamente variável (CVT); evitação de colisão (radar de curta distância); comandos *drive-by-wire* (substituição de cabos por sistemas eletrônicos capazes de corrigir erros do motorista).
- *Controle de motores*: DSPs podem ser usados para controlar digitalmente vários tipos de motores: monofásicos, trifásicos, sem escova, motores de passo, etc.;
- *Usos domésticos e comerciais*: DSPs de baixo custo são usados corriqueiramente em qualquer cenário que necessite do tratamento ou análise simples de sinais analógicos. São encontrados em leitores de barras e de cartões magnéticos, em detectores de dispositivos de RFIDs, no controle de monitores sensíveis ao toque, fontes ininterruptas de força (UPS), máquinas lavadoras e secadoras de roupa e lavadoras e louça, fornos microondas e em muitas outras aplicações.

Em comparação com outros cenários de utilização de DSPs, o controle digital de sinais exige baixa capacidade de processamento. Algumas aplicações são tão simples que a única necessidade de memória regravável é suprida pelos

próprios registradores: assim, os custos de produção da unidade são ainda mais baixos.

Alguns DSPs em produção adequados a aplicações de controle digital de sinais são apresentados a seguir:

- Família Texas Instruments TMS320C24x: trabalha com aritmética de ponto fixo de 16 bits e um acumulador de 32 bits, *clock* de 40MHz e chega ao limite teórico de 40MMAC/s (1MAC/ciclo). Suportam laços de custo zero de uma única instrução, sem aninhamento. Uma unidade custa no máximo US\$ 8,00 e o pacote ocupa apenas 49mm<sup>2</sup>;
- Família Texas Instruments TMS320C28x: em comparação com a família TMS320C24x, esta trabalha com aritmética de ponto fixo de 32 bits com acumulador de mesma largura e chega a 150MHz, também sustentando 1 MAC/ciclo. Consome 625mW. O preço da unidade chega a US\$ 14,00;
- Família Freescale 56F8xx: trabalha com aritmética de 16 bits, com dois acumuladores independentes de 36 bits, *clock* de até 60MHz e 60MMAC/s (1MAC/ciclo). Laços de custo zero podem ser aplicados a blocos de instruções, sem aninhamento. Uma unidade custa entre US\$ 6,00 e US\$ 17,00.

## 4.2 Decodificação de áudio

A miniaturização dos dispositivos portáteis permitiu o surgimento de equipamentos de reprodução digital de áudio muito compactos e leves (poucas dezenas de gramas) e capazes de reproduzir dezenas de horas sem a necessidade de energia externa. Em consequência, sistemas de reprodução de áudio digital tornaram-se atualmente muito populares, a tal ponto que os DSPs responsáveis por essa tarefa passaram a ser incorporados em outros dispositivos eletrônicos, como aparelhos celulares e PDAs.

A decodificação de áudio não exige muito poder de processamento, mas a quantidade de memória RAM deve ser suficiente para comportar os pacotes codificados e a correspondente representação do sinal decodificado. Se o dispositivo lidar com criptografia (em especial por questões de DRM, *Digital Rights Management*), o processador deve ser dimensionado para suportar também a carga de trabalho necessária para realizar a descriptografia. Normalmente também são incorporados alguns controles a cargo do usuário, como volume, balanço e equalização.

Os seguintes DSPs atuais são adequados para as tarefas de decodificação de áudio:

- Família Texas Instruments TMS320C55x: trabalha com aritmética de 16 bits de ponto fixo, com duas ALUs (uma de 40 bits e outra de 16 bits), duas unidades independentes de MAC e quatro acumuladores

de 40 bits. Os modelos mais rápidos da família rodam a 300MHz e chegam a 600MMAC/s (2MAC/ciclo). Consome 180mW. Os processadores dessa família são capazes de executar decodificação de áudio nos formatos MP3/AAC em tempo real. O custo da unidade fica entre US\$ 4,00 e US\$ 18,00;

- Processador ARM AudioDE: esse processador foi desenhado especificamente para o processamento de áudio e voz. Opera com aritmética de 24 bits e possui um acumulador de 48 bits. Opera com *clock* máximo de 75MHz e consome 7,5mW. Segundo o fabricante, a decodificação de uma seqüência de áudio estéreo de 48kHz codificada a 320kHz em formato MP3 requer apenas 8MHz de dedicação do processador;
- Processador Philips CoolFlux DSP: opera com aritmética de 24 bits, possui 2 acumuladores de 56 bits e duas unidades independentes de MAC. Opera com *clock* de 185MHz e chega a 370MMAC/s (2MAC/ciclo). Consome 10,5mW. Segundo o fabricante, a decodificação de uma seqüência de áudio estéreo de 44,1kHz codificada a 128kHz em formato MP3 requer apenas 12MHz de dedicação do processador.

## 4.3 Controle por voz

*Sistemas controlados por voz* são os que respondem a instruções verbais dadas pelo usuário. Nas aplicações construídas para o uso em processadores poderosos, o processo pode envolver a detecção e interpretação da seqüência fonética emitida, mas no contexto de DSPs, cujo poder de processamento e memória é limitado, a interpretação da instrução usualmente consiste em armazenar em um *buffer* uma representação digital do som falado pelo usuário e em seguida compará-lo com um banco de dados armazenado em ROM ou memória não-volátil.

O controle por voz foi bastante popularizado por sua adoção em vários modelos de telefones celulares, permitindo que o usuário dite o número a ser chamado. A natureza portátil desses dispositivos exige um *hardware* pequeno e de baixo consumo de energia. Outros exemplos de aplicações de controle por voz incluem eletrodomésticos (lavadora de roupas, microondas, aparelhos de som, controles remotos universais), comunicadores sem fio para ambientes hospitalares e militares, secretárias eletrônicas, centrais telefônicas e brinquedos.

Nesses casos, e comum que o sinal de voz seja tratado no domínio da frequência: portanto, é importante que o DSP seja capaz de executar a operação de FFT na velocidade exigida pela aplicação.

Um dos processadores adequados a esse cenário é descrito a seguir:

- Família Freescale DSP568xx: essa família de DSCs opera com aritmética de 16 bits de ponto fixo e possui 4 acumuladores de 36 bits. Trabalha a 120MHz e sustenta 120MMAC/s (1MAC/ciclo). Permite laços de custo zero aninhados e aplicados a blocos de instruções. A unidade custa entre US\$ 3,00 e US\$ 20,00.

#### 4.4 Voz sobre pacotes

Os sistemas de voz sobre pacotes são os especializados em codificar, transmitir, receber e decodificar áudio de voz por meio de pacotes de dados digitais. Os sistemas mais conhecidos nesse nicho são os de voz sobre IP, baseados em pacotes sobre o protocolo IP (*Internet Protocol*), outras tecnologias também foram desenvolvidas para o transporte sobre outras infra-estruturas, como no caso das redes de telefonia digital.

O som voz possui algumas peculiaridades quando comparado a áudio em geral: além da banda de frequência de interesse ser mais restrita (tipicamente até 8kHz, contra 20MHz para música), o espectro de densidade de potência da voz possui certas características matemáticas que permitem compressão mais eficiente em formato digital. Por outro lado, tanto os procedimentos de envio (compressão, codificação e eventual criptografia) quanto os correspondentes procedimentos de recepção devem ser executados com a menor latência possível, pois o atraso de poucos décimos de segundo é o suficiente para ser percebido. Outro problema é a ocorrência de eco, um efeito desagradável comum em transmissões telefônicas e mais grave quando a transferência é feita em meio digital [8]. Para isso, os DSPs devem contar com recursos de cancelamento de eco, que geralmente são disponibilizados pelos fabricantes como rotinas de *software*.

Alguns processadores adequados a esse cenário são vistos abaixo:

- Família Analog Devices ADSP-21xx: trabalha com aritmética de ponto fixo de 16 bits com acumulador de 40 bits, frequências de 75 a 160MHz. Os modelos mais avançados da família podem chegar a \$79;
- Família Texas Instruments TMS320C55x (descrita na Seção 4.2).

#### 4.5 Vídeo digital

As aplicações de vídeo digital exigem alto poder de processamento, espaço em memória RAM e a capacidade de sustentar um grande fluxo de dados, tanto de entrada quanto de saída. Isso é necessário porque o *stream* de dados comprimidos pode chegar a alguns Mb/s, enquanto o fluxo de dados descomprimidos com facilidade ultrapassa os 10MB/s.

Nesse cenário, é adequado o uso de DSPs com alto poder de paralelismo de instruções e de dados, DMAs de alta velocidade e ISA com extensões próprias para tratamento de vídeo. Além disso, deve-se levar em conta que em geral os fluxos de vídeo são combinados com alguns canais de áudio e podem vir criptografados por questões de DRM (*Digital Rights Management*).

Alguns processadores adequados a esse cenário são vistos abaixo:

- Processador Analog Devices ADSP-21535 Blackfin: Esse processador combina unidades de processamento de sinais com um microcontrolador RISC e possui arquitetura especialmente voltada para o tratamento de vídeo, como instruções específicas para o tratamento de pixels e 4 ALUs de 8 bits dedicadas a vídeo. Opera a 300MHz e possui duas unidades aritméticas para MAC de 16 bits associadas a dois acumuladores independentes de 40 bits, podendo chegar ao limite teórico de 600MMAC/s (2MAC/ciclo);
- Processador Texas Instruments TMS320DM6446: Combina um núcleo DSP C64x+ (que trabalha a 594MHz) com um processador ARM9. Possui 8 unidades aritméticas, sendo 6 de uso geral (32 ou 40 bits) e 2 específicas para multiplicação de  $16 \times 16$  bits, com os quais pode-se alcançar o limite de 4.752MMAC/s. Possui também 64 registradores de 32 bits. Permite codificação e decodificação em tempo real dos seguintes padrões de vídeo: MPEG-2, MPEG-4, VC1/WMV e H.264.

### 5 Conclusões

Este trabalho apresentou as principais características de processadores DSP, como também as principais aplicações que necessitam deste tipo de microprocessador, desde uma simples aplicação de controle digital até processamento de áudio e vídeo. Existe um grande esforço para cada vez mais se conseguir melhoras de performance, consumo de energia, aumento de velocidade e principalmente em relação ao custo, o que sempre torna os DSP a melhor solução para muitas aplicações. Os requisitos das próximas gerações de DSPs estão cada vez mais aumentando com o uso de dispositivos móveis e o crescimento da necessidade de processamento de áudio, vídeo nesses dispositivos. Existe também uma tendência crescente de se desenvolver arquiteturas que facilitem o desenvolvimento de compiladores mais eficientes, permitindo que as aplicações DSP sejam escritas em linguagens de alto nível. Como as aplicações para DSPs estão em contínua mudança espera-se uma evolução contínua dos processadores DSP.

## Referências

- [1] Texas instruments [Online] Disponível:<http://www.ti.com/>, Acesso: Outubro 2006.
- [2] Berkeley Design Technology, Inc. Pocket guide to processors for DSP. Disponível em <http://www.bdti.com/pocket/pocket.htm>, October 2006.
- [3] R. Cravotta. EDN Magazine's 2006 digital signal processing directory. Disponível em <http://www.edn.com/index.asp?layout=DSP&industryid=44259>, 2006.
- [4] A. Devices. Adsp-21xx processors [Online] Disponível:<http://www.analog.com/processors/adsp/index.html>, Acesso: Outubro 2006.
- [5] J. Eyre and J. Bier. The evolution of dsp processors. *IEEE Signal Processing Magazine*, 17(2):43 – 51, 2000.
- [6] A. Gatherer, T. Stetzler, M. McMahan, and E. Auslander. Dsp-based architectures for mobile communications: past, present and future. *IEEE Communications Magazine*, 38(1):84–90, 2000.
- [7] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, 2002.
- [8] Y. Lu, R. Fowler, W. Tian, and L. Thompson. Enhancing echo cancellation via estimation of delay. *IEEE Transactions on Signal Processing*, 53(11):4159–4168, November 2005.
- [9] S. W. Smith. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Newnes, 2002.
- [10] E. J. Tan and W. B. Heinzelman. Dsp architectures: Past, present and future. *Computer Architecture News*, 31(3):6 – 19, 2003.