DSP TMS320C67x

Eduardo Putton
Universidade Estadual de Campinas
Unicamp – Instituto de Computação
Av. Albert Einstein, 1251 – Cidade
Universitária CEP 13083-970,
Campinas/SP- Brazil

+55 19 .3521 5838
edu.putton@gmail.com

RESUMO
Os processadores DSP (do inglês, *Digital Signal Processor*) foram projetados levando-se em consideração as operações mais habituais em um processamento digital de sinais. Eles conseguem reunir alto desempenho com baixo consumo de energia e custo através de arquitetura própria associada ao uso de instruções VLIW (do inglês, *Very Long Instruction Word*).

Categorias e Assunto
C.1.1 [***Processor Architectures***]*: Single Data Stream Architectures - RISC/CISC, VLIW architectures, Pipeline processors,*

Termos Gerais
Desempenho.

Palavras-chave
DSP, Pipeline, VLIW, Data Path.

1. INTRODUÇÃO
Os Processadores Digitais de Sinal (DSP) são dispositivos programáveis especialmente projetados para processamento digital de sinal de forma eficiente, em tempo real que causou uma grande evolução na área de processamento de sinais nas ultimas duas décadas.

Os DSPs diferem de microcomputadores e microcontroladores pelo tipo de aplicação. O alto desempenho, o baixo consumo de energia e o custo relativamente baixo conferem importantes vantagens aos DSPs frente aos demais processadores. Atualmente é aplicado em diversas áreas, desde bens de consumo até aplicações militares passando por aplicações automotivas, industriais e telecomunicações. É utilizado em sintetizadores de som profissional, radio e TV digital, sistemas de segurança com reconhecimento de face e impressão digital, sistemas avançado de *cruise control* com GSP que evitam colisões entre veículos, diagnóstico médico remoto, reconhecimento de voz e de som, análise de pontos finitos, detectores de radar, controle de motores, e de robôs, equipamentos de ultrasom, redes de computadores, etc.

Os microcomputadores são normalmente associados a uma CPU e foram projetados para atender a diversas aplicações e são capazes de executar programas nos mais diversos domínios. São normalmente utilizados em PCs e estações de trabalho.

Os microcontroladores, por sua vez, são projetados para uma aplicação especifica em sistemas embarcados (*embeded systems*) e são dedicados a computações bastante específicas. São utilizados, por exemplo, em sistemas de gerenciamento de motores de aviões e carros.

O TMS320C67x é produzido pela Texas Instruments (TI) e pertence a família TMS320C6000. Operando com ponto flutuante de 32 bits, ele pode atingir uma taxa de execução de até 1800 MIPS. O TMS320C67x é derivado do TMS320C62x, que opera com ponto-fixo de 16 bits. A TI produziu o primeiro DSP em 1983, denominado TMS32010, e foi um grande sucesso que ajudou a TI a liderar o mercado de DSPs de uso geral.

Em meados de 2003, a família TMS320C67x já era composta por 6 integrantes, operando de 100 a 1350 MHz e com desempenho de até 1350 MFLOPS (veja Tabela 1) [1].

Tabela 1. Comparação entre diferentes DSPs da família TMS320C67x.

Família TMS320C67x	C6701	C6711	C6711C	C6712	C6712C	C6713
freqüência (MHz)	167	150	200	100	150	225
desempenho (MFLOPS)	1000	900	1200	600	900	1350
memória L1+L2(KBytes)	128	72	72	72	72	264
encapsulamento (mm)	35	27	27	27	27	27
consumo interno (W)	1,4	1,1	0,8	0,7	0,5	1,2
preço estimado (USD)	104,80	28,50	18,65	16,73	13,50	26,85

Para a família TMS320C6000 existe um ambiente completo de ferramentas para desenvolvimento, incluindo um compilador C eficiente, otimizador para programação em linguagem assembly simplificada e debugger baseado em Windows.

2. ARQUITETURA
O DSP TMS320C67x possui dois d*ata paths* de 32 bits em ponto-flutuante, cada qual contendo 2 ALUs (do inglês, *Arithmetic Logic Unit*) para inteiros e ponto-flutuantes, 1 unidade de multiplicação (16x16 inteiros, 32x32 inteiros, 32 bits e 64 bits ponto-flutuante) e 1 unidade de adição e subtração para geração de endereçamento (Figura 1). Cada *data path* contém um *register file* com 16 registradores de propósito geral de 32 bits que podem ser utilizados tanto para armazenamento de endereços como

dados. Para cálculos aritméticos de 64 bits são utilizados pares de registradores de 16 bits [2].

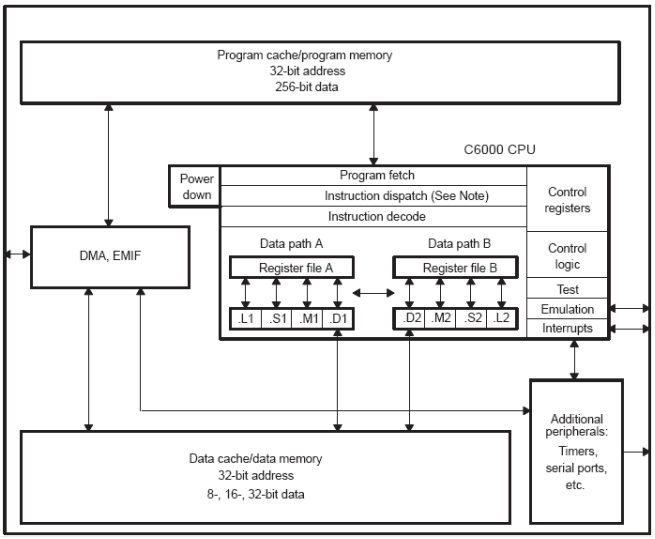
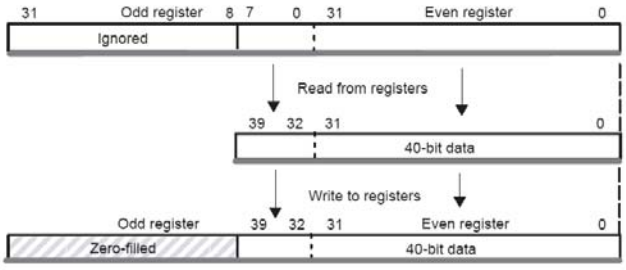


Figura 1. Diagrama de blocos do DSP TMS320C67x.

2.1 Registradores de Propósito Geral

Existem dois conjuntos de registradores (*register files*) de propósito geral, cada um contendo 16x registradores de uso geral de 32 bits (A0-A15 e B0-B15). Eles suportam dados com tamanhos variando de 16 bits a 40 bits em ponto-fixo e 64 bits em ponto-flutuante e podem ser armazenados em duplas de registradores concatenados. Nesse caso, os 32 LSBs (do inglês, *Lost Significativ Bit*) são definidos no registrador par enquanto os 32 ou 8 MSBs (do inglês, *More Significativ Bit*) irão para o registrador ímpar da dupla [2].



< ---------- 32 MSB ---------- > < ---------- 32 LSB ---------- >

Figura 2. Pares de registradores para armazenar dados de 40 bits.

2.2 Unidades Funcionais

As unidades de *program fetch*, *instruction dispatch* e *instruction decode* do TMS320C67x podem despachar até 8 instruções de 32 bits (algumas pode ser nops – *no operation*) a cada ciclo de *clock* da CPU. O processamento das instruções ocorre em cada um dos 2 *data path* (A e B), cada um deles contendo 4 unidades funcionais (.L, .S, .M e .D) e 16 registradores de propósito geral de 32 bits. Cada unidade funcional possui 2 multiplicadores e 6 ALUs [2].

Na tabela 2 está detalhado as operações que são realizadas em cada das 4 unidades funcionais [2].

Tabela 2. Unidades funcionais e suas operações.

Unidade Funcional	**Operações realizadas em ponto-fixo**	**Operações realizadas em ponto-flutuante**
Unidade .L (.L1 e .L2)	Operações aritméticas de 32/40 bits, operações de comparação, operações lógicas de 32 bits, deslocamentos de bits	Operações aritméticas e de conversão envolvendo inteiros, 32 bits e 16 bits
Unidade .S (.S1 e .S2)	Operações aritméticas de 32/40 bits, operações de comparação, operações lógicas de 32 bits, deslocamentos de 32/40 bits, desvios, geração de constantes, transferência do registrador de controle para registradores de propósito geral e vice-versa (apenas .S2)	Operações de comparação, operações de adição e subtração (16 bits, 32 bits), operações de valores absolutos
Unidade .M (.M1 e .M2)	Operações de multiplicação de 16 bits e de 32 bits	Operações de multiplicação em ponto-flutuante
Unidade .D (.D1, .D2)	Cálculo de endereços de 32 bits com adição, subtração, cálculos lineares e circulares; Cargas e armazenamento com offset de 5 bits no caso do .D2 de 15 bits	Carga de 64 bits com offset de 5 bits

Outras características do TMS320C67x:

- *Instruction packing*: propicia tamanho de código equivalente a oito instruções executadas serialmente ou em paralelo;
- Redução do tamanho do código e da quantidade de buscas de programas, assim como o consumo de energia;
- Execução condicional de qualquer instrução, possibilitando a execução condicional de todas as instruções para tentar reduzir as conseqüências drásticas de um desvio;
- Multiplicadores que podem de 32 bits que podem produzir o resultado em 32 ou 64 bits;
- Suporte em *hardware* para operações em ponto-flutuante em precisão de 32 ou 64 bits no formato IEEE;
- Lógica *power-dowm* que reduz os *clocks* reduzindo consumo de energia.

2.3 Data Path

O *data path* do TMS320C67x corresponde ao barramento de dados e as unidades de funcionais que compõem o DSP. Eles são altamente especializados na computação utilizada em processamento digital de sinal, objetivando alto desempenho do processador. Ele é composto de 2 *register files* (A e B), 8 unidades funcionais, 2 *data path* para carga da memória (LD1 e

LD2), 2 *data path* de endereçamento (DA1 e DA2) e 2 r*egister file* de dados para *cross path* (1X e 2X) (Figura 3).

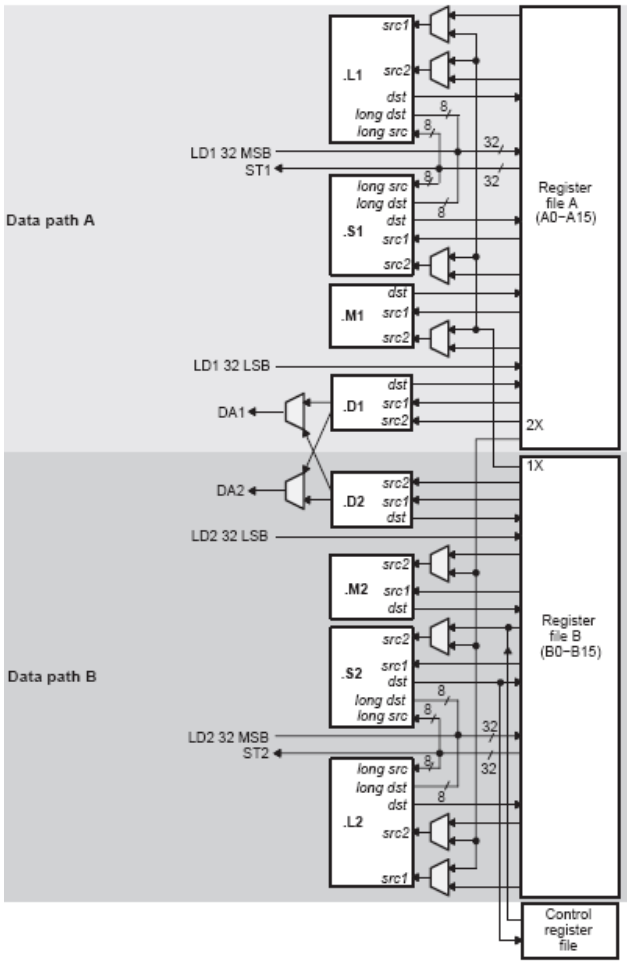


Figura 3. Data Path da CPU do TMS320C67x.

2.4 *Data Path* - Acesso a Memória para carga e armazenamento

O DSP TMS320C67x possui os *data path* A e B de 32 bits cada para carregar dados da memória para os *register file*: LD1 para *register file* A e LD2 para *register file* B. Além disso, ele possui uma segunda opção de carga de 32 bits para os *register files* A e B, o que permite instruções LDDW carregar simultaneamente dois valores de 32 bits para o *register file* A e dois para o register file B. Em LD1b estão os 32 bits menos significativos e em LD1a os 32 bits mais significativos. Existem também 2 *paths* de 32 bits cada, ST1 e ST2 para armazenamento de valores dos *register files* para a memória.

2.5 *Data Address Path* - Endereçamento de Dados

Os dois *data paths* de endereço, DA1 e DA2, estão ligados a cada unidade nos paths A e B, o que permite aos endereços de dados gerados para qualquer um dos *paths* o acesso a dados de qualquer registrador.

As linhas DA1 e DA2 e seus *data paths* associados são especificados respectivamente em T1 e T2. T1 consiste no *path* de endereço de DA1 e no *path* de dados LD1 e ST1.

2.6 *Register File* de dados para *cross path*

Cada unidade funcional pode ler e escrever nos seus próprios *register files*. .L1, .S1, .D1 e .M1 pode escrever diretamente no *register file* A e .L2, .S2, .D2 e .M2 no *register file* B. Os *register files* A e B são conectados entre si via o *cross path* 1X e 2X. Seis das oitos unidades funcionais são interligadas por meio do *cross path*.

3. MEMÓRIA

O TMS320C67x possui arquitetura com memória interna e externa.

A memória interna baseia-se no modelo de cachês de dois níveis.

O primeiro nível L1 é dividido em *cache* de programas L1P e *cache* de dados L1D, cada uma delas possuem 4 KBytes de capacidade, tempo de acesso a CPU de 1 ciclo de *clock* e sua arquitetura é baseada no modelo de Harvard.

A *cache* de programas L1P é diretamente mapeada, utiliza o protocolo *read allocate*, possui 64 bytes de tamanho de linha e a penalidade de erro é de 5 ciclo de *clock*.

A *cache* de dados L1D é associativa de 2 *ways*, utiliza o protocolo *read allocate*/*write back*, possui 32 bytes de tamanho de linha e a penalidade de erro é de 4 ciclo de *clock*.

O segundo nível de cachê L2 é associativa de 4 *ways*, utiliza o protocolo *read and write allocate*/*write back*, , tamanho de linha de 128 bytes e utiliza estratégia de substituição da menos recentemente utilizado e tem capacidade de 64 KBytes a 256 KBytes.

A memória externa é acessado por meio da interface EMIF (do inglês, *External Memory Interface*) de 32 bits e são suportados os tipos SBSRAM (*Synchronous Burst* RAM), SDRAM (*Synchronou*s DRAM) e *Asynchronous* SRAM de até 8 GBytes.

4. ENDEREÇAMENTO

O TMS320C67x possui os modos de endereçamento linear e circular. O modo é especificado pelo registrador de controle AMR (do inglês, *Addressing Mode Register*). Todos os registradores de propósito geral podem realizar endereçamento linear, mas apenas oito deles podem também realizar o modo de endereçamento circular (A4-A7 e B4-B7). Existem dois bits para identificar o modo de endereçamento: 00 para linear, 01 para circular usando o campo BK0 e 10 para circular usando o campo BK1.

O modo de endereçamento circular, conhecido também como de modulo, mantém um registrador de início e um registrador de fim com cada registrador de endereço, permitindo que os modos de endereçamento de auto-incremento e auto-decremento se reinicializem ao alcançarem o fim do buffer [3].

5. CONJUNTO DE INSTRUÇÕES

O TMS320C67x utiliza o formato de linguagem *opcode-operand assembly* no qual cada código tem um campo para a operação e campo de 1 a 4 operandos. Para adição existem 3 campos

adicionais para indicar execução paralela, condicional e qual unidade utilizar e caso esta esteja indisponível indicar outra unidade.

Todas as instruções podem ser executadas condicionalmente. 5 registradores de propósito geral são utilizados como registradores condicionais [4].

Loops em hardware não são suportados. Porem a arquitetura paralela do processador permite a implementação de *loops* virtuais sem *overhead* [4]

Todas as instruções têm o tamanho formato fixo de 32 bits e possuem o formato parecido com o RISC (do inglês, *Reduced Instruction Set Computer*). Todas instruções possuem os campos CREG (do inglês, *Conditional Register*) e Z, que servem para indicar a condição para a execução da sub-instrução de 32 bits, e o campo P que informa se a próxima sub-instrução pode ser executada de forma paralela.

6. PIPELINE

O Pipeline do TMS320C67x faz uso de instruções VLIW e possui 16 fases dividido em 3 etapas: *Fetch, Decode e Execute* [2].

Etapa de *Fetch*: possui 4 fases (PG, PS , PW e PR) e realiza a busca e carga das instruções. A cada *fetch packet*, uma instrução VLIW de 256 bits contendo até oito sub-instruções de 32 bits é carregada pelo barramento de instruções. Cada fase tem duração de um ciclo de *clock* independente do tipo e formato da instrução.

A geração do novo endereço de memória pelo processador é realizado na fase PG (*Program Address Generate*). Na fase PS (*Program Address Send*), o endereço gerado é enviado a memória. A preparação do processador para a instrução a ser lida é realizada na fase PW (*Program Address Ready Wait*) e na fase PR (*Program Address Fecht Packet Recieve*) o processador recebe a instrução para a execução (Figura 4).

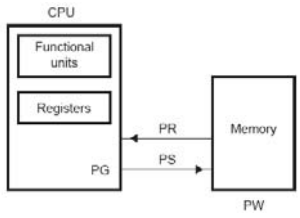


Figura 4. Fases da etapa *Fetch* do Pipeline.

Etapa de *Decode*: possui duas fases (DP e DC) e realiza a decodificação das instruções. Cada fase tem duração de um ciclo de *clock* independente do tipo e formato da instrução (Figura 5).

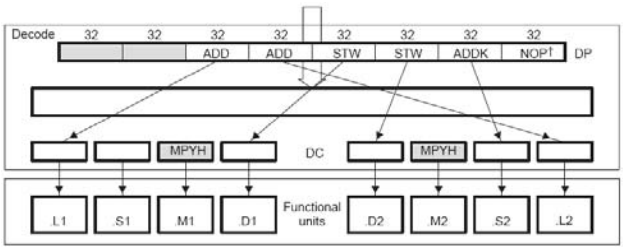


Figura 5. Fases da etapa *Decode* do Pipeline.

Na fase DP (*Instruction Dispach*) o *fetch packet* contendo até 8 sub-instruções em quebrado em um ou mais *execute packet*, determinando aquelas que podem ser executas em paralelo em um ciclo qualquer (bit P da sub-instrução) e em qual data path será utilizado (bit S da sub-instrução). Na fase DC (*Instruction Decode*), os registros fonte, os registros destino e os *paths* associados são decodificados para a execução das instruções nas unidades funcionais.

Etapa *Execute*: possui até 10 fases, nas quais é realizado a execução da instrução. A maioria das instruções de ponto-fixo tem a latência de 1 ciclo. As instruções de multiplicação em ponto-fixo, de carga e de desvio tem latência de até 5 ciclos de *clock* e as instruções de ponto-flutuante tem latência de máxima de 10 ciclos de *clock* (Figura 6).

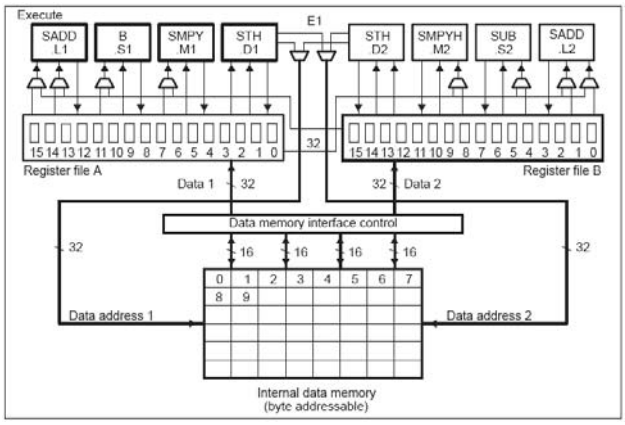


Figura 6. Etapa *Execute* do Pipeline.

7. PERIFÉRICOS

A família TMS320C67x possui uma grande variedade de periféricos para comunicação com dispositivos externos a CPU do DSP: DMA, HPI, EMIF, *ports* seriais bufferizados de vários canais, interfaces digitais de áudio e dois *timers* de 32 bits [4].

A controladora DMA (do inglês, *Direct Memory Access*) é composta de 4 canais independentes e programáveis permite a realização de quatro diferentes operações simultaneamente além de contar com um quinto canal, que pode servir a solicitações oriundas do HPI. Os TMS320C6711, TMS320C6712 e TMS320C6713 possuem um DMA melhorado de 16 canais.

A interface HPI (do inglês, *Host Port Interface*) de 16 bits permite que um dispositivo externo possa acessar diretamente a memória da CPU.

A interface EMIF (do inglês, *External Memory Interface*) é uma interface para memória externa para alguns tipos de dispositivos, que permite que um espaço de memória externa ao espaço de memória disponível internamente a CPU.

8. INTERRUPÇÕES

Existem 8 registradores de controle de interrupção:

- CSR (do inglês, *Control Status Register*) permite a habilitação global das interrupções que podem ser mascaradas;

- IER (do inglês, *Interrupt Enable Register*) permite a habilitar e desabilitar as interrupções individualmente;
- IFR (do inglês, *Interrupt Flag Register*) exibe o status das interrupções;
- ISR (do inglês, *Interrupt Set Register*) permite, manualmente, alterar o status de uma interrupção para pendente;
- ICR (do inglês, *Interrupt Clear Register*) permite, manualmente, alterar o status de uma interrupção retirando a condição de pendente;
- ISTP (do inglês, *Interrupt Service Table Pointer*) ponteiro para o início da tabela de interrupções;
- IRP (do inglês, *Interrupt Return Pointer*) contém o endereço de retorno utilizado pela CPU para retornar uma interrupção que pode ser mascarada;
- NRP (do inglês, *Nonmasklabe Interrupt Return Pointer*) contém o endereço de retorno utilizado pela CPU para retornar uma interrupção que não pode ser mascarada.

9. CONCLUSÃO

Este trabalho abordou o tema DSP sob a ótica da família TMS320C67x. A abordagem do texto foi oferecer os conceitos e estruturas básicas para estudos futuros e recomenda-se ao leitor que necessitar de maiores informações busque a literatura constante na bibliografia consultada para a confecção do texto. Os sites da Texas Instruments possuem um vasto e rico material.

Por fim vale a pena ressaltar que a família TMS320C67x é líder de mercado de processadores DSP e precursora no uso da tecnologia VLIW em DSP. Porém, existem outros fabricantes com outras tecnologias voltadas a execução de programa de uma forma mais eficiente.

10. REFERÊNCIAS

[1] TMS320C67x Floating-Point DSP Generation, literature number SPRT196C, 2003, Texas Instruments.

[2] DSP TMS320C67x CPU and Instruction Set Reference Guide, literature number SPRU733A, November 2006, Texas Instruments.

[3] Hennessy, John L. and Patterson, David A., Arquitetura de Computadores: Uma Abordagem Quantitativa, 3ª Edição, Editora Campus, Rio de Janeiro,2003, Capitulo 2, pagina 74.

[4] A BDTI Analysis of the Texas Instruments TMS320C67x, 2003, Berkeley Design Technology Inc., www.bdti.com (visitado em 15/08/2008).