

IMPLEMENTAÇÃO DE CURSO DE LABORATÓRIO DE PROCESSAMENTO DIGITAL DE SINAIS DE BAIXO CUSTO

Francisco Assis de Oliveira Nascimento – assis@unb.br

Adson Ferreira da Rocha – adson@unb.br

Universidade de Brasília – Faculdade de Tecnologia - Departamento de Engenharia Elétrica

Campus Universitário Darcy Ribeiro – Brasília

CEP:70.910-900 – Brasília - DF

***Resumo** - Este trabalho mostra como implementar um curso prático de processamento digital de sinais com uma infra-estrutura de baixo custo. A proposta envolve a utilização um computador pessoal hospedeiro e o cartão DSK para família TMS 320C6X da TEXAS Instruments de baixo custo. O curso experimental propicia ao aluno desenvolvimento de projetos simples utilizando tecnologia e arquitetura de processadores digitais de sinal. Os experimentos dão ênfase em aplicações em tempo real, características do uso de comunicação via interrupção do processador e via DMA (do inglês, Direct Memory Access), avaliação do esforço computacional na implementação de algoritmos, efeitos da amostragem e quantização de sinais analógicos, efeitos do ruído numérico e como minimizá-lo, verificação de instabilidades que podem ocorrer na implementação de um projeto, implementação de filtros digitais FIR e IIR, análise espectral via transformada rápida de Fourier e outras aplicações mais sofisticadas, como a implementação de equalizadores adaptativos e canceladores de ecos. Os experimentos podem ser realizados no laboratório específico ou, o aluno pode reter o DSK durante o curso, instalando o aplicativo e o cartão DSK co-processador de sinais em seu próprio computador pessoal. Os experimentos são codificados em linguagem de programação C o que permite rapidez na codificação e flexibilidade na implementação dos algoritmos. Este curso ainda protótipo já vem sendo oferecido há três anos (uma vez ao ano) com grande procura por parte dos alunos e os resultados foram positivos em termos da capacitação dos discentes e de trabalhos de fim de curso e dissertações de mestrado iniciadas durante o curso.*

***Palavras-chave:** processamento digital de sinais, implementação laboratorial, sistema co-processador de sinas.*

1. INTRODUÇÃO

A implementação de cursos experimentais com base tecnológica com uma infra-estrutura adequada de laboratório tem-se se consistido em grande barreira para as Universidades face aos custos envolvidos na aquisição de equipamentos diversos. Além de toda a instrumentação, geralmente é necessário um local físico específico com bancadas e com a instalação elétrica necessária.

No segmento de processamento digital de sinais cursos teóricos são factíveis com uma infra-estrutura mínima, uma sala de aula e professor podem ser suficientes. Quando envolvem atividades relacionadas com simulação computacional exigem ferramentas computacionais dedicadas, que geralmente são de custo elevado. Uma alternativa é o uso de software livre e/ou o desenvolvimento das ferramentas específicas que podem ser atualizadas com a participação do corpo discente (OPPENHEIN, 1999). Neste caso poder-se-ia construir uma biblioteca de algoritmos computacionais que seria atualizada à medida que o curso seja ministrado periodicamente. Microcomputadores pessoais e compiladores constituem os requisitos mínimos necessários de infra-estrutura. Muitas vezes os próprios alunos possuem um microcomputador pessoal onde podem desenvolver as atividades de simulação. Essa solução é interessante do ponto de vista da infra-estrutura mínima necessária, do custo dessa infra-estrutura e do ganho de conhecimento do aluno, contudo, não seria a melhor opção para o entendimento da arquitetura específica de processadores de sinais, bem como, fica restrita a implementação de diversos experimentos cuja finalidade é o desenvolvimento de sistemas de processamento digital de sinais em tempo real (MITRA, 2001).

Aplicações com o desenvolvimento de um *kernel* para tempo real, rotinas de comunicação, algoritmos com sensibilidade numérica à arquitetura do processador e técnicas de algoritmos rápidos podem ser mais bem entendidas quando associadas a experimentos específicos para o contexto do tempo real. Esse trabalho dá ênfase especificamente a tais aplicações, propondo uma solução de baixo custo de implementação de cursos experimentais em processamento digital de sinais.

2. ORGANIZAÇÃO DO CURSO

A idéia da criação de um curso de laboratório de processamento digital de sinais com aplicações em tempo real foi originado a partir de um convênio com a *TEXAS Instruments* no qual recebemos uma dúzia de kits DSK (TEXAS INSTRUMENTS, 1997). O kit consiste de um cartão com um processador de sinais da linha 6000, uma fonte de alimentação, um cabo para a conexão na porta paralela de um computador pessoal e a ferramenta integrada de desenvolvimento *Code Composer* (DAHNOUM, 2000). Além do kit para processamento digital de sinais é necessário um computador hospedeiro. Na figura 1 a seguir é ilustrada a configuração necessária para a realização dos experimentos.

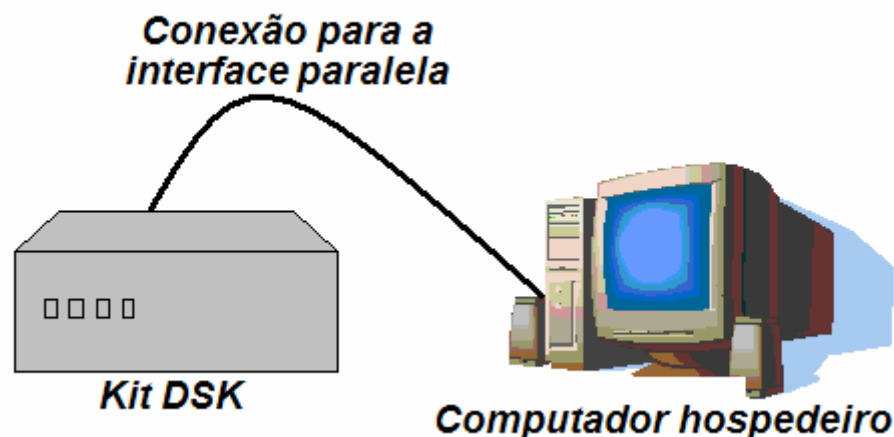


Figura 1 – Configuração com computador hospedeiro e Kit DSK utilizados na implementação dos experimentos do curso de laboratório de processamento digital de sinais

O curso laboratorial é constituído de um conjunto de experimentos subdivididos em duas etapas distintas. Na primeira etapa que envolve os três primeiros experimentos o aluno tem contato com a instrumentação que utilizará no curso, conhece a arquitetura do processador de sinais que está utilizando, as formas de comunicação entre periférico e hospedeiro e principalmente, a ferramenta integrada de desenvolvimento que possui interface interativa para plataforma Windows. Na segunda etapa do curso é desenvolvido um conjunto de experimentos específicos, que serão apresentados mais adiante, onde são as investigadas características de aplicações em tempo real.

O curso é ministrado em um total de 15 semanas compreendendo 60 horas/aula presenciais. Na proposta do curso existe a possibilidade dos alunos reterem os Kits DSK durante o período de realização do curso. Cada Kit é cedido por dupla de alunos que ficam responsáveis pelo mesmo até o final do semestre e tem a obrigação de devolvê-lo em bom estado. Desta forma, também se pode melhorar a disponibilidade de tempo para desenvolvimento das atividades alternando entre aulas expositivas, atividades realizadas em casa e seminários de apresentação dos resultados experimentais. Para alunos que não possuam microcomputadores pessoais para instalação do Kit DSK, ficam disponibilizados dois computadores no laboratório de processamento digital de sinais para esse fim. A utilização do laboratório é aberta a todo horário comercial, inclusive à noite e fim de semanas.

Os procedimentos experimentais são desenvolvidos em linguagem de programação C. Existe a possibilidade também de desenvolvimento de algoritmos em *Assembly*, entretanto, dada a complexidade da arquitetura do processador, não é recomendável (SIMSEK, 2000). São utilizadas ferramentas gráficas do próprio *Code Composer* para visualização de resultados, bem como outras instrumentações como por exemplo geradores de sinais e osciloscópios. Experimentos com sinal de voz podem ser implementados com a avaliação subjetiva utilizando a interface de áudio do kit DSK. Experimentos interessantes podem ser realizados na avaliação subjetiva de filtragem digital ou, por exemplo, os efeitos audíveis da superposição de espectros em processos de dizimação em sinais de voz.

A avaliação do curso é realizada com base nos relatórios experimentais produzidos pelos alunos, em um seminário descrevendo os aspectos tecnológicos e resultados de um experimento específico escolhido previamente no início do curso e, em uma prova final abordando os experimentos realizados e conceitos adquiridos durante o período de realização do curso.

2.1 Organização dos experimentos da primeira etapa do curso

Como foi comentado anteriormente, os experimentos são divididos em duas etapas distintas, a primeira aborda a arquitetura do processador e a instrumentação necessária para a realização dos experimentos. A seguir são apresentados a descrição desses experimentos com os tópicos abordados:

Experimento 1 – O sistema de desenvolvimento e depuração de algoritmos – *Code Composer Studio*

1 – Avaliação da funcionalidade do *Code Composer* e aplicação de facilidades como o *breakpoint* e do *watch window*. Aspectos da depuração de programas e no desenvolvimento de aplicativos dedicados para tempo real. Investigação das similaridades e as diferenças entre as ferramentas *break point* e *probe point*.

2 - Análise do subsistema GEL (do inglês, *General Extension Language*) do *Code Composer Studio* e que pode ser utilizado para o desenvolvimento e a depuração de um programa para execução no Kit DSK.

4 – Avaliação em diagramas de blocos, por meio de fluxograma ou por descrição de algoritmo do experimento realizado envolvendo sinais de entrada armazenado no microcomputador hospedeiro até sua a representação da saída seja ela do tipo gráfico ou pela interface D/A. Investigação da facilidade de se alterar a variáveis e parâmetros durante a execução de um algoritmo.

5 – Avaliação do conjunto de ferramentas do *Code Composer Studio* que podem ser utilizadas ser utilizadas na depuração de programas.

Experimento 2 – Desempenho em tempo real de uma arquitetura dedicada para processamento digital de sinais.

1 - Estudo de como as ferramentas de auxílio ao desenvolvimento de aplicações em tempo real podem ajudar na implementação de sistemas de processamento digitais.

2 – Análise estatística de *threads* como parâmetro de desempenho no desenvolvimento de uma aplicação para tempo real

3 – Avaliação do comportamento em tempo real de uma aplicação utilizando o RTDX (do inglês, *Real-Time Data Exchange*) do *Code Composer Studio*.

4 – Investigação do número significativo de interrupções que acarreta diminuição de desempenho do sistema em termos da capacidade média de processamento do processador de sinais. Técnicas que podem ser utilizadas para otimizar o desempenho de um algoritmo em termos da arquitetura de um DSP (do inglês, *Digital Signal Processor*).

Experimento 3 – Dispositivos de I/O em arquiteturas DSP.

1 – Necessidades e características de um sistema de processamento digital de sinais em se comunicar com dispositivos de I/O (do inglês, *Input/Output*).

2 - Tipos de dispositivos de I/O de um sistema de processamento digital de sinais. Exemplos e aplicações.

3 – Tratamento da entrada e da saída.

4 – Ferramenta integrada de desenvolvimento sistemas em tempo real para teste e refinamento de algoritmos sem ter que suspender o aplicativo que é executado no kit DSK.

5 – Visualização gráfica do sinal de entrada e do sinal de saída. Aplicações em DSP em que esta facilidade é importante na avaliação, teste e refinamento do sistema em desenvolvimento.

Na próxima seção é apresentado o conjunto de experimentos desenvolvidos na segunda etapa do curso.

2.2 Conjunto de experimentos da segunda etapa do curso

Na segunda etapa do curso laboratorial são implementados experimentos enfatizando aplicações específicas de processamento digital de sinais.

Experimento 4 – Interfaces A/D e D/A – comunicação via *polling* e DMA – implementação de linha de retardo e média móvel.

1 – Pesquisa e descrição das técnicas de *polling* e DMA. Vantagens e desvantagens de cada técnica.

2 – Aplicações em que se utilizaria uma técnica ou a outra (*polling* e DMA.).

3 – Implementação de linha de retardo via *polling* e via DMA. Avaliação em sinais de voz.

4 – Em termos da capacidade de computação do processador como o uso de *polling* ou DMA afeta o desempenho do mesmo

- 5 – Como o processo de *polling* pode garantir que uma determinada aplicação possa ser realizada em tempo real pelo processador.
- 6 – Implementação da média móvel de comprimento N (amostras) e cálculo do esforço computacional em MIPS necessários para a realização desse SLID (sistema linear e invariante ao deslocamento).

Experimento 5 – Estudo dos efeitos da quantização e da sub-amostragem sobre um sinal analógico digitalizado.

- 1 – Implementação de algoritmos em tempo real para emular os efeitos da diminuição do comprimento da palavra digital e da sub-amostragem em um sinal analógico gera uma distorção característica na informação digitalizada. Descrição de cada tipo de distorção. Verificação das características e dos componentes espectrais parece predominar em cada tipo de distorção.
- 2 – Estudo do ruído de quantização quando o comprimento de palavra digital é grande (>8 bits).
- 3 – Estudo do modelamento matemático descritivo do ruído de quantização. Comparação entre modelo teórico e os resultados obtidos no experimentalmente.
- 4 – Verificação do que acontece com o ruído de quantização quando o comprimento da palavra digital é pequeno (<7 bits). Avaliação do modelo estudado.
- 5 – Verificação do que acontece com o sinal digitalizado quando se utiliza o algoritmo de sub-amostragem. Interpretação dos resultados experimentais em confronto com a teoria.

Experimento 6 – Projeto e implementação de filtros digitais FIR com fase linear.

- 1 – Implementação no kit DSK de filtro digital FIR a partir de um vetor de coeficientes (fixos) que caracteriza o SLID (Sistema Linear Invariante ao Deslocamento) de interesse. Avaliação dos resultados em tempo real com a injeção de sinais (formas de ondas e voz) na interface A/D e audição dos resultados na interface D/A.
- 2 - Analise o vetor de coeficientes do filtro via FFT e caracterização do SLID com respeito ao seu comportamento no domínio das frequências.
- 3 – Projeto de um SLID FIR com fase linear que atenda as especificações da figura abaixo para comprimento $N \leq 100$ amostras. Utilizando uma ferramenta para análise espectral verificar se a resposta em frequência atende as especificações de projeto

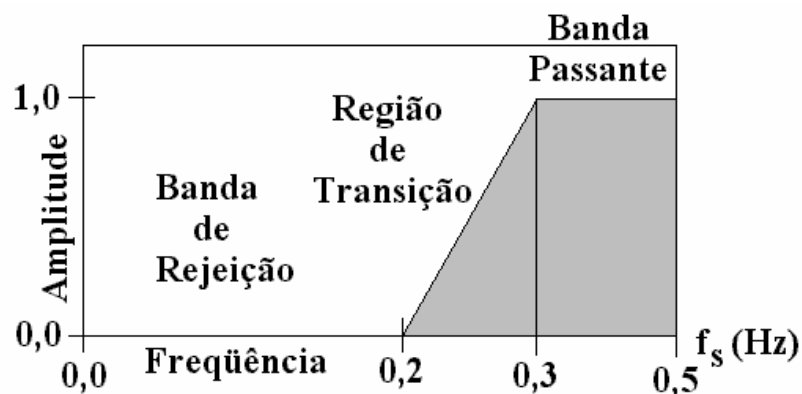


Figura 2 – Especificação de projeto de filtro passa-altas FIR.

4 – Projeto agora um SLID FIR com fase linear que atenda as especificações passa-banda mostrada na Figura 3 a seguir para $N \leq 100$. Verifique uma ferramenta para análise espectral se a resposta em frequência está de acordo com a especificação.

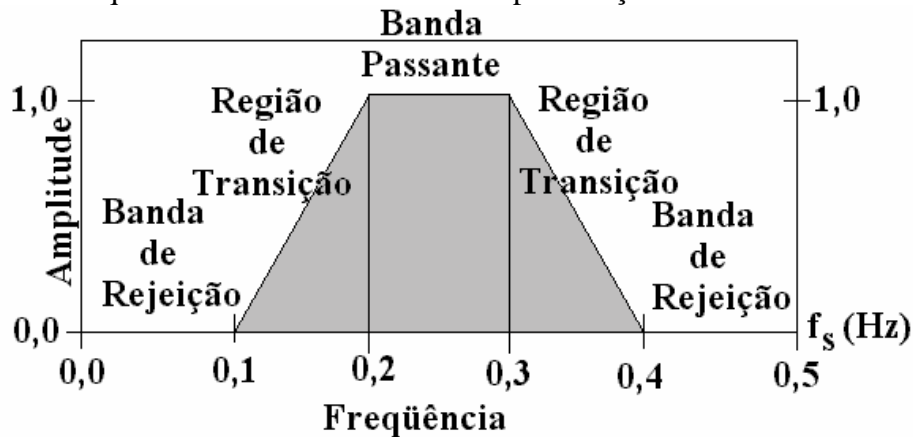


Figura 3 – Especificação de projeto de filtro passa-banda FIR.

5 – Implementação no kit DSK do filtro cujos coeficientes foram disponibilizados e dos dois SLID's projetados. A quantização dos coeficientes no formato Q.15 (multiplicar cada coeficiente por 2^{15} e depois truncar para a representação em ponto fixo). Avaliação dos filtros com formas de ondas e sinais de voz.

5 – Análise do ruído introduzido pela quantização coeficientes do filtro e pelo comprimento finito de palavra digital do processador.

6 – Cálculo do esforço computacional para a realização de cada SLID em MIPS para:

a) a realização na forma direta;

b) a realização na forma direta utilizando o *buffer* circular para a implementação da linha de retardo.

5 - Sabendo que a capacidade computacional do processador é de 1500 MIPS, avaliar quanto (%) se está utilizando de seus recursos computacionais ao implementarmos SLID's propostos no experimento.

6 – Avaliação do atraso oferecido por cada SLID em números de amostras e em milisegundos.

Experimento 7 – Projeto e implementação sistemas IIR

1 – Implementação se sistema para simular reverberação de som. Investigação do efeito da instabilidade do SLID modificando a posição dos pólos e avaliação de resultados por meio da resposta audível.

2 – Projeto um SLID IIR que atenda as especificações de amplitude mostradas na figura 4. Verificação se a resposta em frequência está de acordo com a especificação. Cálculo do erro médio quadrático das aproximações em função do comprimento N do filtro. Simule e implemente versões com $3 < N < 20$.

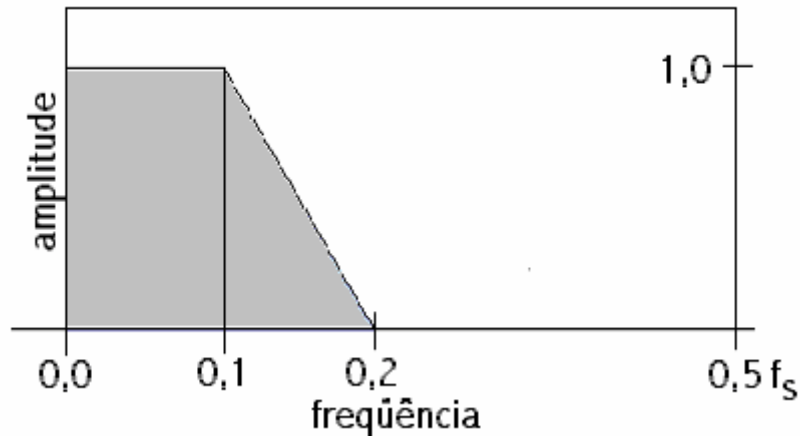


Figura 4 – Especificação de amplitude para filtro-passa-baixas.

3 – Projeto de um SLID IIR que atenda as especificações de amplitude passa-banda mostradas na figura 3. Verificação se a resposta em frequência está de acordo com a especificação para $4 < N < 20$.

4 – Implementação no Kit DSK dos dois SLID's projetados. A quantização dos coeficientes é feita no formato Q.15. Na realização é utilizada a cascata de funções biquadradas.

5 – Estudo e comparação das implementações de filtros FIR do experimento 06 com os filtros IIR implementados neste experimento com respeito as figuras de mérito:

- a) Complexidade/Esforço computacional;
- b) Estabilidade;
- c) Característica de resposta de fase;
- d) Característica de resposta de amplitude;
- e) Qualidade da aproximação (quantidade de coeficientes necessários para atender a aproximação).

Experimento 8 – Análise espectral via Transformada Discreta de Fourier

- 1 - Algoritmos de implementação rápida da Transformada Discreta de Fourier.
- 2 – Efeitos do comprimento e do tipo da janela utilizada.
- 3 - Avaliação do esforço computacional na implementação dos algoritmos em tempo real.

Na próxima seção será apresentado resumidamente o cartão co-processador de sinais utilizado, dando ênfase para as características fundamentais da arquitetura.

3. PLATAFORMA DE IMPLEMENTAÇÃO EM TEMPO REAL

A implementação dos experimentos é realizada utilizando-se o C6711™ DSK (do inglês, Development Starter Kit) que é uma placa de desenvolvimento de baixo custo baseada no processador de sinais TMS320C6711. O DSK se comunica com um microcomputador hospedeiro pela porta paralela encapsulando um protocolo de depuração conhecido como JTAG (TOSCANO, 2003).

O DSK é constituído de um processador de sinais de 150MHz, suporte para dois *clocks*, controlador de porta paralela (EPP ou SPP bi-direcional), memória *SDRAM* de 16Mb e memória *FLASH* de 128 kb, memória mapeada de I/O, emulação de *JTAG*, *codec* de áudio de 16 bits e *leds* indicadores.

Os algoritmos, em sua grande maioria, são implementados em linguagem de programação ANSI C. Somente códigos críticos e de iniciação foram escritos em ASSEMBLY. Para otimizar a compilação deste código foram obedecidas também algumas diretivas como o uso de ponteiros ao invés de matrizes, estruturas simples de *if-then-else* entre outros.

Para gerenciar as tarefas de aquisição de sinais, processamento de codificação e decodificação optou-se pelo uso do DSP/BIOS. O DSP/BIOS é um núcleo escalonável em tempo real. Foi desenvolvido para aplicações que necessitam de sincronismo execução de tarefas em tempo real, comunicação “*host-to-target*” e instrumentação em tempo real. DSP/BIOS fornece múltiplas tarefas preemptivas, abstração de *hardware*, análise em tempo real e ferramentas de configuração. A figura 5 ilustra a superfície superior do cartão DSK utilizado. Pode-se verificar que no lado direito na figura que mostra o DSK estão os conectores para a entrada do conversor A/D e para saída do D/A.

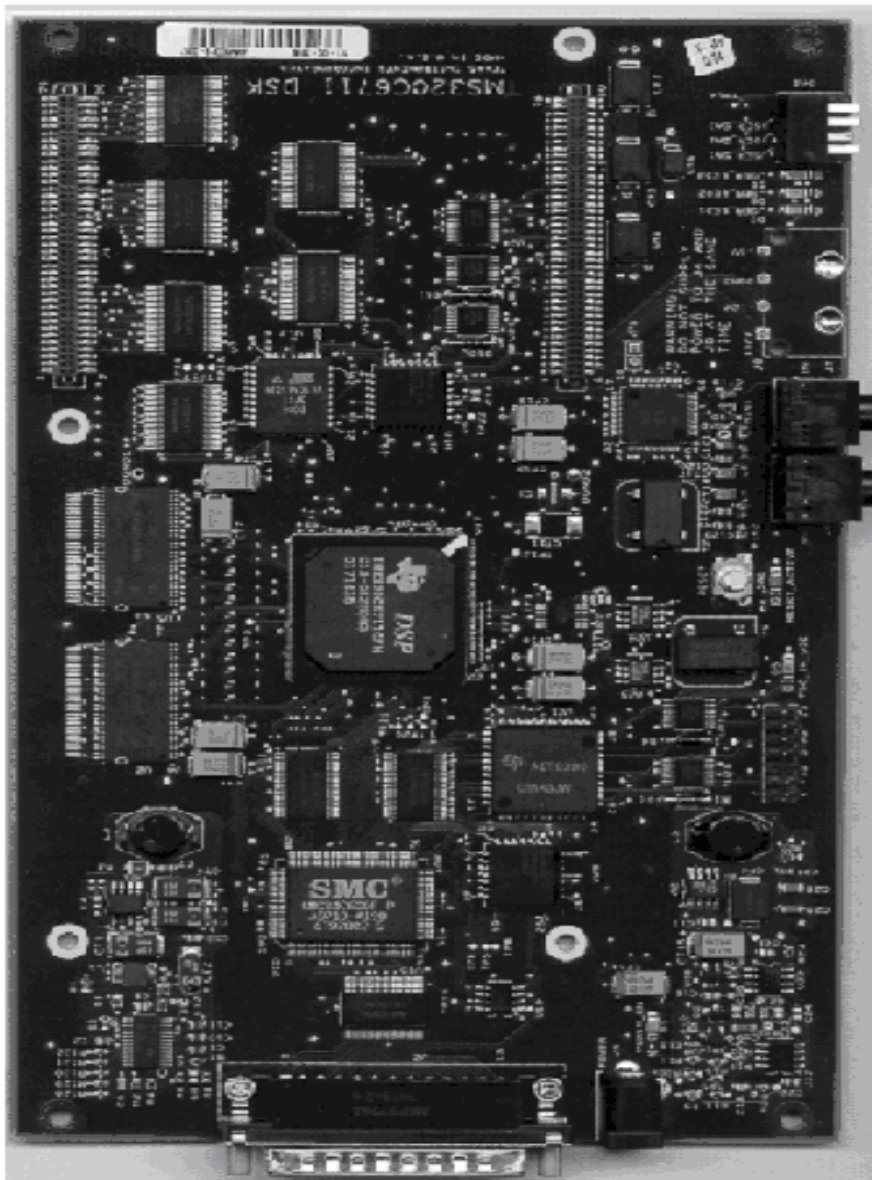


Figura 5 - Cartão DSK para família TMS 320C6X da TEXAS Instruments.

4. DISCUSSÃO

A proposta dos experimentos também abriu uma série de possibilidades para a solução dos problemas propostos. Um exemplo interessante e criativo foi a utilização da placa de som do microcomputador como um gerador de sinais controlado por um software aplicativo com a finalidade de excitação da porta A/D do cartão DSK.

Aplicações mais sofisticadas foram propostas mas ainda não fazem parte do curso. Aplicações em processamento adaptativo de sinais como equalizador adaptativo e cancelador de eco já existem em forma prototípica.

Um sistema de criptografia de voz para canal analógico baseado na mistura pseudo-aleatória e dinâmica dos coeficientes da transformada discreta de Hartley (NASCIMENTO, 1990), com codificação, decodificação e sincronismo também foi implementado (TOSCANO, 2003).

A procura pelo curso tem sido muito grande, entretanto, só existe a possibilidade de atender até 20 alunos por cada vez que ele é oferecido.

5. CONCLUSÕES

Os resultados obtidos com a realização do curso de laboratório de processamento digital de sinais têm sido promissores. Pôde-se observar um ganho de conhecimento tanto do ponto de vista de complementação da teoria como, principalmente, na capacitação no desenvolvimento de projetos em tempo real usando tecnologia DSP. A infra-estrutura física necessária para a implementação do curso é barata além de possibilitar que o aluno desenvolva o experimento em sua casa.

Espera-se a atualização dos experimentos com a participação de alunos de graduação em projetos de fim de curso e bolsas de iniciação científica e de alunos de pós-graduação em dissertações de mestrado. Propostas recentes para a implementação de novos experimentos envolvem o processamento adaptativo de sinais como equalizadores adaptativos de canal e, em criptografia como misturadores frequenciais baseados em transformadas ortogonais discretas.

REFERÊNCIAS BIBLIOGRÁFICAS

DAHNOUM, NAIN, **Digital Signal Processing Implementation Using the TMS320C6000 DSP Platform**, first edition, Prentice Hall, 2000.

MITRA, SANJIT K., **Digital Signal Processing – A Computer Based Approach**, McGraw-Hill, 2001.

NASCIMENTO, FRANCISCO ASSIS DE O. and MALVAR, HENRIQUE S., “*Computer Program for DHT*” - Appendice A.6 of **Discrete Cosine Transform - Algorithms And Applications**, R. K. Rao & P. Yip, Academic Press, 1990, pp. 417- 421.

SIMSEK, NASSER, KEHTARNAVAZ B., **C6x-Based Digital Signal Processing**, Prentice Hall, 2000.

OPPENHEIN, ALAN V. and SCHAFER, RONALD W., **Discrete-Time Signal Processing**, second edition, Prentice Hall, 1999.

TEXAS INSTRUMENTS Inc, **An Implementation of FFT, DCT, and Other Transforms on the TMS320C30**, Waltham, Massachusetts, 1997.

TOSCANO, RICARDO G., Misturador de Voz Baseado na Transformada Discreta de Hartley e Implementado com o Processador TMS320C6711, 2003. Dissertação de Mestrado, Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Universidade de Brasília – UnB.

LOW COST IMPLEMENTATION OF DIGITAL SIGNAL PROCESSING LABORATORY COURSE

***Abstract** - This work shows how to implement a digital signal processing experimental course based on low cost infra-structure. The proposal involves the use of a host personal computer personal and the low cost card DSK from TMS 320C6X family of TEXAS Instruments. The experimental course propitiates to undergraduate and graduate students the development of simple projects using technology and architecture of digital signal processors. The experiments emphasizes rea-time applications, the study of the communications characteristics using processor interruption and DMA access, the evaluation of the computational effort in the real-time algorithms implementation, the study effects of the sampling and the quantization of analogical signals, the effects of the numeric noise and how minimize then, verification of system instabilities that can occur in the project implementation, project of digital FIR and IIR filters, spectral analysis by real-time Fast Fourier Transform, and other more sophisticated applications, as the adaptive equalizers and echoes cancellers. The experiments can be accomplished at the specific laboratory or the student can keep the DSK kit during the course, installing the applicative software and the DSK co-processor signal card in his/her own personal computer. The experiments are codified in C programming language that allows fast code development and flexibility in the algorithms implementation. This course has already been offered for three years (once time a year) with great interest from the students and it has been shown positive results in terms of the students training in undergraduate and graduate activities.*

***Key-words:** digital signal processing, laboratorial implementation, signal co-processor system.*