



Universidade Federal do Paraná

Departamento de Engenharia Elétrica

Processamento Digital de Sinais

Professor: Marcelo Rosa

Aluno: Stefan Klaus Wolter

PROCESSADORES DE SINAIS DIGITAIS – DSP

Junho de 2007

Sumário

Parte I: O que é um DSP.....	3
1.1 A História dos DSP.....	3
1.2 O que é um DSP?.....	4
1.3 Aplicações para p DSP.....	5
1.3.1 Compressão e Descompressão de sinais.....	5
1.3.2 Filtros Digitais.....	6
1.3.3 Equipamentos Eletrônicos.....	7
Parte II: Desenvolvimento de Projetos.....	8
Parte III: Visão Geral de um DSP.....	9
Parte IV: Exemplo de Aplicação.....	14
4.1 Convolução.....	15
Referências.....	17

Parte I: O que é um DSP?

1.1. A história dos DSP

O primeiro DSP a ser lançado no mercado foi o MAC 4 que surgiu no em 1979, fabricado pela Bell Labs, mediante a necessidade crescente do mercado de processadores especializados em processamento digital de sinais.

Porém os DSP's ficaram populares quando a Texas Instruments (TI) lançou em 1983 o TMS32010, graças a inovadora ferramenta RTDX (Real Time Data Exchange) para processamento em tempo real dos sinais. Foi baseado na arquitetura Harvard, então tinha dados de instruções separados dos dados de memória. Também já tinha um conjunto especial de funções, com instruções como carregar e acumular ou multiplicar e acumular. Trabalhava com números de 16 bits e precisava de 390ns para calcular uma operação de multiplicação. A TI desde então é líder até hoje no mercado de DSP's. Mais tarde a Motorola também se tornou importante no desenvolvimento de DSP's com o lançamento do Motorola 56000.

Por volta de 5 anos mais tarde começou a surgir a segunda geração de DSP's. Eles tinham 3 memórias para armazenar dois operandos simultaneamente e tinham incluso hardware para acelerar loops fechados e alguns já operavam com 24 bits. A operação multiplicar-acumular durava por volta de 21ns.

O maior avanço na terceira geração foi o aumento na capacidade de processamento. Isso permitiu aceleração direta de hardware para problemas matemáticos específicos e complexos, como transformada de Fourier e operações de matrizes. Alguns chips até incluíam mais de um processador trabalhando em paralelo. Isso por volta de 1995.

A quarta geração é mais bem caracterizada pelas mudanças no conjunto de instruções e a instrução codificar/decodificar. Foram adicionadas as extensões SIMD (Single Instruction, Multiple Data) e MMX (MultiMedia eXtension or Multiple Math or Matrix Math eXtension), VLIW (Very Long Instruction Word) e a arquitetura superescalar apareceram. Como sempre a velocidade do clock também aumentou e uma operação de multiplicar e acumular passou a ser feita em 3ns.

Os processadores de sinais atuais possuem uma performance muito maior. Isso graças os avanços na tecnologia e arquitetura como regras de design, fast-access two-level cache, circuito (E)DMA e um sistema bus maior. Claro que nem todos os DSP's provêm a mesma velocidade e cada DSP existente é melhor usado em uma aplicação específica, custando de R\$5,00 até R\$800. Um C6000 da Texas Instruments

possui clock de 1GHz e é capaz de fazer até 8000 MIPS (milhões de instruções por segundo)! Os principais fabricantes de DSPs hoje são a Texas Instruments, a Motorola e a Analog Devices.

1.2. O que é um DSP?

Os DSPs são microprocessadores com características próprias que podem ser programados e operam em tempo real, com velocidades muito superiores aos microprocessadores para aplicações genéricas. A capacidade de processar grandes quantidades de números em pouco tempo é um dos principais benefícios que os Processadores Digitais de Sinais oferecem ao mundo da eletrônica.

DSP acima de tudo é um dispositivo programável, que detém seu próprio código de instruções. Cada empresa que cria o seu processador cria também o seu ambiente de desenvolvimento (IDE) próprio para aquele tipo de chip, tornando dessa forma a manipulação do microprocessador muito mais fácil e rápida.

Os DSP's foram projetados levando-se em consideração as operações mais habituais em um processamento digital são as de adição, multiplicação e transferência de memória, consecutivos. Para tal existem instruções de repetição que precedem tais operações tomando possível a execução destas usando-se muitas vezes apenas um ciclo de memória. A capacidade de processamento em tempo real dos DSP's torna este processador perfeito para aplicações onde o delay (tempo de resposta) não é tolerável, como por exemplo em um telefone celular.

A capacidade que os DSPs têm de repetir (com uma instrução prévia, por exemplo "RPT") em extrema velocidade uma instrução complexa (como por exemplo a "MPYA": "Multiply and Accumulate Previous Product") faz com que sejam rapidamente resolvidas, por exemplo, as Transformadas Rápidas de Fourier (FFT), permitindo a implementação de filtros digitais. Estes filtros são muito usados em telecomunicações, por exemplo para decodificar a sinalização (DTMF, MFC, etc).

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.

Os DSPs ganharam popularidade na eletrônica de consumo em aparelhos como os teclados, que sintetizam os sons de diversos instrumentos, como por exemplo os órgão de tubos, o piano e o violão.



Figura 1: Exemplo simples de um processamento de sinal

1.3. Aplicações para o DSP

A tecnologia DSP está presente atualmente em infinitos dispositivos como celulares, computadores multimídia, gravadores de vídeo, CD players, controladores de disco rígido, modems e televisores. Esta presente nas mais diversas áreas, como militar, médica, científica e automotiva.

1.3.1. Compressão e descompressão de sinais

Uma aplicação muito utilizada nos DSPs é a compressão e descompressão de sinais. A compressão de sinal em banda é usada larga escala em celulares, permitindo que um grande número de chamadas possa ser administrada simultaneamente por um único aparelho.

Outro exemplo bem característico está presente em câmeras multimídia dos PCs, permitido que pessoas possam ver e transmitir imagens em tempo real pela internet.

O uso dos DSPs associados a algoritmos de compressão permitiu a implementação de diversas tecnologias de CODEC's. CoDec é o acrônimo de Codificador/Decodificador, dispositivo de hardware ou software que codifica/decodifica sinais. Exemplos de chips DSP são o DSP542 e DSP549 fabricados pela Texas Instruments e usados pela CISCO Systems em seus *gateways* VoIP [CISCO-CODEC1].

Cada CODEC provê certa qualidade de voz. A medida de qualidade da voz transmitida é uma resposta subjetiva de um ouvinte. Uma medida comum usada para

determinar a qualidade do som produzido pelos CODECs específicos é o MOS (*Mean Opinion Score*). Com o uso do MOS, um amplo range de ouvintes julgam a qualidade de uma amostra de voz (correspondendo a um CODEC particular) numa escala de 1 a 5. A partir desses resultados, é calculada a média dos *scores* para atribuir o MOS para aquela amostra [CISCO-CODEC2].

Na tabela 1 são apresentados alguns *scores* MOS para os CODECs mais usados. Na Tabela 2 são descritos os *scores* MOS.

Método de Compressão	Bit Rate (kbit/s)	MOS Score	Delay (ms)
G.711 PCM	64	4.1	0.75
G.726 ADPCM	32	3.85	1
G.728 LD-CELP	16	3.61	3 to 5
G.729 CS-ACELP	8	3.92	10
G.729 x 2 Encodings	8	3.27	10
G.729 x 3 Encodings	8	2.68	10
G.729a CS-ACELP	8	3.7	10
G.723.1 MP-MLQ	6.3	3.9	30
G.723.1 ACELP	5.3	3.65	30

Tabela 1: Scores MOS de alguns CODEC's [CISCO-CODEC2]

Score	Definição	Descrição
5	Excelente	Um sinal de voz perfeito gravado em um local silencioso
4	Bom	Qualidade de uma chamada telefônica de longa distância (PSTN)
3	Razoável	Requer algum esforço na escuta
2	Pobre	Fala de baixa qualidade e difícil de entender
1	Ruim	Fala não clara, quebrada

Tabela 2: Scores MOS [UMSIS-MOS]

1.3.2. Filtros Digitais

No processamento de sinais o papel do filtro é remover partes indesejáveis do sinal, como ruídos ou simplesmente extrair partes importantes do mesmo, como componentes de certa banda de frequência.

O filtro analógico utiliza circuitos eletrônicos analógicos feitos basicamente de resistores, capacitores e amplificadores operacionais. Muitos destes filtros são largamente usados em aplicações como: redução de ruído, realce de sinal de vídeo, equalização gráfica em sistemas hi-fi, entre outros.

Hoje em dia, entretanto, temos a precisão dos filtros digitais. Estes tipos de filtros utilizam um processador digital para realizar cálculos numéricos em cima de amostras de sinal. Através da entrada de um sinal analógico, um conversor analógico-digital (ADC) realiza a conversão deste sinal para ser lido pelo DSP. Este então fica encarregado de realizar os cálculos necessários para fazer a filtragem, multiplicando os valores da entrada por constantes e somando os produtos resultantes, além de utilizar técnicas matemáticas, como Transformadas de Fourier e Hilbert. Caso seja necessário, estes valores que agora representam o sinal filtrado, podem ser convertidos através de um conversor digital-analógico (DAC).

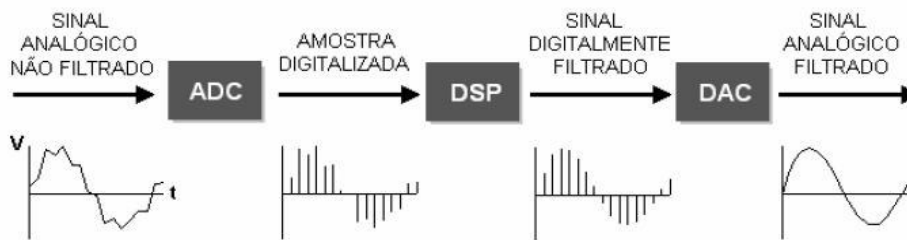


Figura 2: Esquema de configuração básica de um filtro digital

Utilizar um filtro digital em DSP tem inúmeras vantagens perante o filtro analógico. Entre elas está o fato dos filtros digitais serem programáveis, ou seja, pode-se durante uma operação armazenar os dados na memória do processador, por exemplo. Os filtros analógicos, geralmente compostos por componentes ativos, estão sujeitos a variações devido a alterações de temperatura ou devido a outros componentes. Variações cujos filtros digitais não estão sujeitos a sofrer, o que os tornam muito mais estáveis.

Em contraparte aos filtros analógicos, os filtros digitais suportam uma larga escala de trabalho em sinais de baixa frequência. Com o avanço da tecnologia dos DSPs a cada dia tem-se utilizado e aprimorado filtragens digitais também em alta frequência, principalmente com ondas de rádio, que anteriormente só era realizada através da filtragem analógica.

Outra característica importante dos filtros digitais em DSPs é a capacidade deste se adaptar automaticamente à característica de cada sinal, o que os torna muito mais versáteis no processo de filtragem.

1.3.3. Equipamentos Eletrônicos

O uso de DSPs na fabricação de equipamentos eletrônicos hoje em dia é muito grande. Vários itens como celulares, equipamentos de laboratório, automóveis e computadores fazem parte do mundo DSP.

Como anteriormente o DSP é utilizado em larga escala na fabricação de telefones celulares, permitindo fluxo contínuo das conversas, sem que haja interrupção do sinal enviado ou recebido em um determinado instante.

Alguns automóveis atualmente já dispõem da tecnologia de processamento digital de sinais. Neste caso usa-se o microprocessador para gerar os mapas e rotas automaticamente, através do recebimento do sinal via GPS, bem como analisar seu consumo médio de combustível, gerenciar as estações de rádio digital, analisar as situações onde o uso do air-bag se faz necessário, entre outras finalidades.

A fabricação de equipamentos eletrônicos utilizando a tecnologia DSP é muito extensa. Hoje em dia para praticamente qualquer produto analógico existe seu equivalente digital.

Parte II: DESENVOLVIMENTO DE PROJETOS

Por ser um dispositivos programável o DSP necessita de uma IDE própria para o desenvolvimento de projetos. É através das IDEs que são programados os DSPs. Em sua grande maioria estes trabalham com as principais linguagens de programação, como Assembly, C e C++. Cada fabricante fornece junto ao produto esta IDE de desenvolvimento que foi feita para trabalhar com as funções pré-definidas de cada DSP.

Um IDE fornece o ambiente de trabalho necessário para realização de um projeto. Estes softwares são na verdade compiladores que trabalham ligados diretamente no DSP, gerenciando toda a execução, simulação e depuração do código. Como forma de ilustração destes ambientes de desenvolvimento temos o IDE da *Texas Instruments*, o Code Composer Studio (CCP) e a IDE da *Analog Devices*, o visualDSP++.

A realização de um projeto envolve três principais etapas: pesquisa, simulação e emulação.

A pesquisa envolve basicamente o entendimento do problema, onde serão buscadas informações se o dispositivo a ser utilizado envolve necessidade de cálculos em ponto fixo ou em ponto flutuante.

Após a etapa de pesquisa concretizada, parte-se para a etapa de simulação. Neste momento ocorrerá a modelagem inicial do projeto, onde é realizada as compilações do código bem como as modificações e testes necessários.

Por fim temos a etapa de emulação, que envolve a realização testes do projeto já dentro do seu ambiente de uso. Nesta parte o dispositivo é submetido a análises detalhadas, visando-se obter um equipamento o mais preciso possível, que não apresente falha. Nesta fase que são corrigidos os bugs do programa.

O DSP tem uma vantagem perante outros processadores, que neste caso é a possibilidade de upgrade, sem que haja a necessidade da troca de equipamentos. Através da interface JTAG (Joint Test Action Group) pode-se a modificar qualquer momento o projeto.



Figura 3 – exemplo de emuladores JTAG. (A) XDS560 PCI-BUSS JTAG SCAN_BASED EMULATOR (DSP8032U). (B) BLACKHAWK XDS560 USB EMULATOR (DSP11990U)

Parte III: VISÃO GERAL DE UM DSP

Para entender melhor como ocorre o processamento digital de sinais é interessante analisar o que está por trás de um DSP, bem como as ferramentas e mecanismos que trabalham junto com o processador.

Um DSP não funciona sozinho. Por trás do processador estão componentes importantes como registradores, unidades aritméticas lógicas (ALU), geradores de endereço e memória. O termo DSP pode também referenciar a placa onde se encontra o chip DSP juntamente com este grupo de mecanismos.

No diagrama de blocos abaixo temos uma visão geral da formulação de uma placa de DSP. Composta por cinco blocos dominantes, os DSPs em sua maioria basicamente a mesma arquitetura: o processador principal (core processor), uma porta

SRAM, um emulador JTAG, uma porta para comunicação externa e um processador do tipo I/O, bem como seus componentes internos.

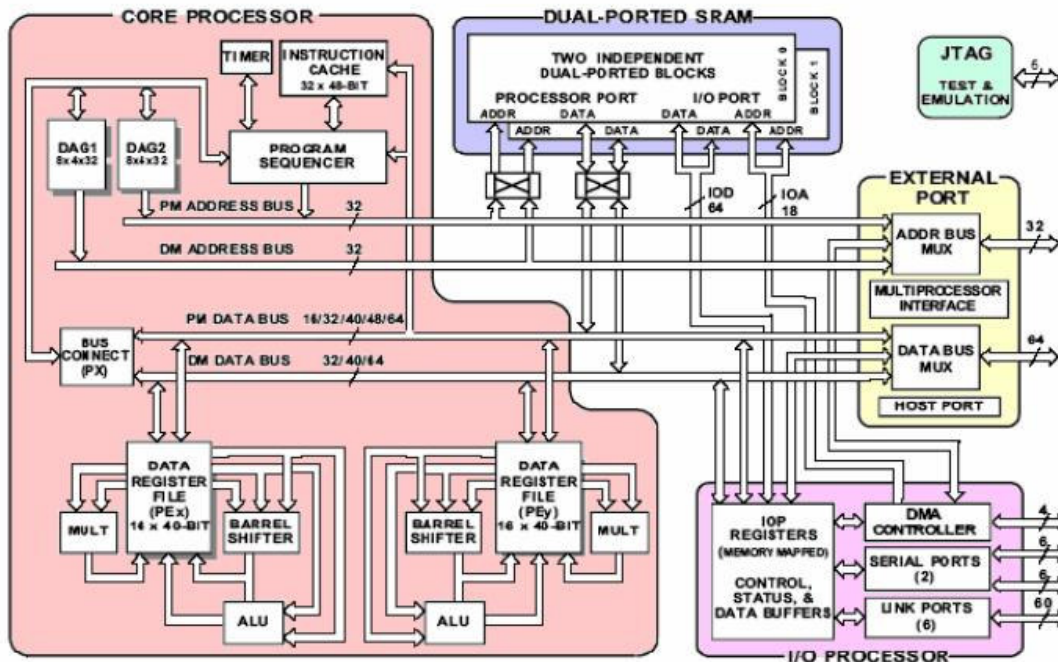


Figura 4- Diagrama de blocos de uma placa DSP

Dentro do core processor podemos encontrar subgrupos importantes. Os geradores de endereço (DAG) são responsáveis por fornecer um endereçamento imediato entre a memória e os registradores.

As unidades Aritméticas Lógicas (ALU) realizam operações aritméticas e lógicas, podendo estas serem em ponto fixo ou em ponto flutuante. A maioria das placas de DSP contam com um timer programável, que poder ser usado para realizar interrupções periódicas, bem como ativar funções ou rotinas a cada novo ciclo.

As unidades de DMA (Direct Memory Access) podem operar de forma independente e “invisível” ao processador principal. Além de ser importante na comunicação entre a memória interna e as portas seriais ou link ports, o DMA pode ser usado em programas onde é necessária a liberação do processador principal para a realização de outras rotinas.

Alguns DSPs ainda podem interagir diretamente com outros softwares, a fim de agilizar e dinamizar a execução dos programas. A *Texas Instruments* disponibiliza no seu DSP TMS320C64x+ uma interface com o programa *Matlab* onde tanto o software quanto o DSP funcionam de forma bidirecional no envio e recebimento de dados.

Dependendo do tipo e modelo do DSP desejado pode-se encontrar placas específicas para processamento de áudio ou de vídeo e imagem. Para tal são necessários dispositivos especiais de codificação e decodificação de sinais. Cada DSP pode vir acompanhado de um respectivo CODEC, que realiza as conversões necessárias para que ocorra seu processamento.

A figura 4 ilustra o diagrama do ADSP-21160M da Analog Devices. Esse DSP possui as seguintes especificações técnicas

<ul style="list-style-type: none">■ Clock Speed: 80 MHz■ MMACS: 160■ MFLOPS: 480■ On-Chip SRAM (Mbits): 4Mbit■ On-Chip ROM (Mbits): 0Mbit■ Serial Ports: 2■ Link Ports: 6■ Core Voltage (V): 2.5V■ DAI – 0■ SPDIF/DTCP-Package: PBGA	<ul style="list-style-type: none">■ Aplicações<ul style="list-style-type: none">■ Áudio■ Instrumentação médica■ Militar■ Gráfica■ Imagem■ Comunicações
---	--

Onde:

<ul style="list-style-type: none">■ MMACS<ul style="list-style-type: none">■ MACs: Hardware de Multiplicação e ACumulação.■ Base para criação de filtros, FFTs e convoluções (multiplicação e acumulação sucessivas).■ Realizam milhões de operações por segundo (daí o nome MMAC).■ Quanto mais MMACS mais rápido e eficiente é o DSP.■ MFLOPS<ul style="list-style-type: none">■ Parecidos com os MMACS.■ Realizam milhões de operações em ponto flutuante (FLOating Point).■ Quanto mais MFLOPS mais rápido e eficiente é o DSP.

- DAI
 - Digital Applications Interface.
 - Possui 20 pinos ligados ao processador que podem ser ligados à qualquer periférico ou a programas com input e output.
- SPDIF
 - Sony Phillips Digital Interface
 - Ligado ao codec de áudio do DSP (interface com CD player)
- DTCP
 - Encriptação e decriptação de informação de áudio

A placa é composta por 5 blocos dominantes:

- CORE PROCESSOR
 - DAG
 - INTERRUPTS
 - TIMER
 - ALU
 - BUS CAPACITIES
- DUAL-PORTED SRAM
- EXTERNAL PORT
- I/O PROCESSOR
 - DMA CONTROLER
- JTAG

Componentes do Core Processor:

1- DAG

- DAG (DATA ADDRESS GENERATORS)
 - Existem dois geradores de endereço.
 - Fornecem endereçamento imediato ou indireto quando informação é transferida entre memória e registradores.
 - Permitem o processador a fornecer saídas de endereço simultâneas para duas operações de leitura ou escrita.
 - DAG1: suporta endereços de 32-bits para armazenamento em memória.
 - DAG2: suporta 32-bits para programar a memória para acesso de dados

2 – Bus

■ BUS CAPACITIES

- Composto por 6 barramentos principais:
 - PM address, PM data, DM address, DM data, IO address, IO data.
- O PM address e o DM address bus transferem os endereços para instruções e dados.
- PM e IO address contem 32-bits.
- DM e IO data contem 64-bits.
- DM address fornece um trajeto para o conteúdo de qualquer registrador no processador para ser transferido para qualquer outro registrador ou locação de memória.
- O IO address e IO data acessam internamente a memória via DMA sem prejudicar o andamento do processador.

3 – Timer

■ TIMER

- Intervalo programável.
- Provem geração de interrupções periódicas.
- Quando ativo o timer decrementa um contador de 32-bits em cada ciclo.
- Quando seu contador chega a zero o ADSP-21160 gera uma interrupção. O contador é então automaticamente recarregado.

4- ALU

■ ALU (UNIDADE LOGICA E ARITMÉTICA)

- Realiza as operações aritméticas e lógicas
- Ponto fixo ou flutuante

5 – Interrupts

■ INTERRUPTS

- 4 interruptores
 - IRQ0-2
 - Reset
- Podem controlar:
 - Timer
 - Operações no DMA
 - Buffers circulares
 - Operações aritméticas
 - Multi-processamento de vetores
 - Definições do usuário
- Interface com LEDs
- Realização de tarefas pré-definidas em software
- Interação com o timer
 - SIG_TMZ0

Este é um DSP eficiente para processar áudio. Como dito antes, outros DSPs possuem outras especificações e são eficientes para outras áreas.

Parte IV: Exemplo de Aplicação

Neste exemplo foi utilizado o DSP da *Analog Devices*, modelo ADSP-21160M, cujo software de desenvolvimento foi o Visual DSP++ 3.5.

4.1. Convolução

```
void main()
{
    InitializeSineTable( Table, sizeof(Table) );
    GenerateInputPulse( Table, Input, sizeof(Table) );
    GenerateImpulseCoeffs( Table, Impulse, sizeof(Impulse) );
    CalculateOutputPulse( Input, sizeof(Input), Impulse,
                        sizeof(Impulse), Output );

    exit( 0 );
}

void InitializeSineTable( float Table[], size_t nSize )
{
    const float RADIANS = 0.017453292;

    for ( int i=0; i<nSize; i++ )
    {
        Table[i] = sin ( RADIANS * i );
    }

}

void GenerateInputPulse ( const float Table[], float Input[], size_t nSize )
{
    for (int i=1; i<=10; i++ )
    {
        for (int j=0; j<nSize/10; j++ )
        {
            Input [j*i] = Table [(j*i)/i];
        }
    }
}

void GenerateImpulseCoeffs ( const float Table[], float Impulse[], size_t nSize )
{
    for ( int i=0; i<nSize; i++ )
    {
        Impulse[i] = Table[(i*10)];
    }
}

void CalculateOutputPulse( const float Input[], size_t nInputSize,
                        const float Impulse[], size_t nImpulseSize, float Output[] )
{
    for( int i=0; i<nInputSize; i++ )
    {
        for( int j=0; j<nImpulseSize; j++ )
        {
            Output[i+j] = Output[i+j] + (Input[i] * Impulse[j]);
        }
    }
}
```

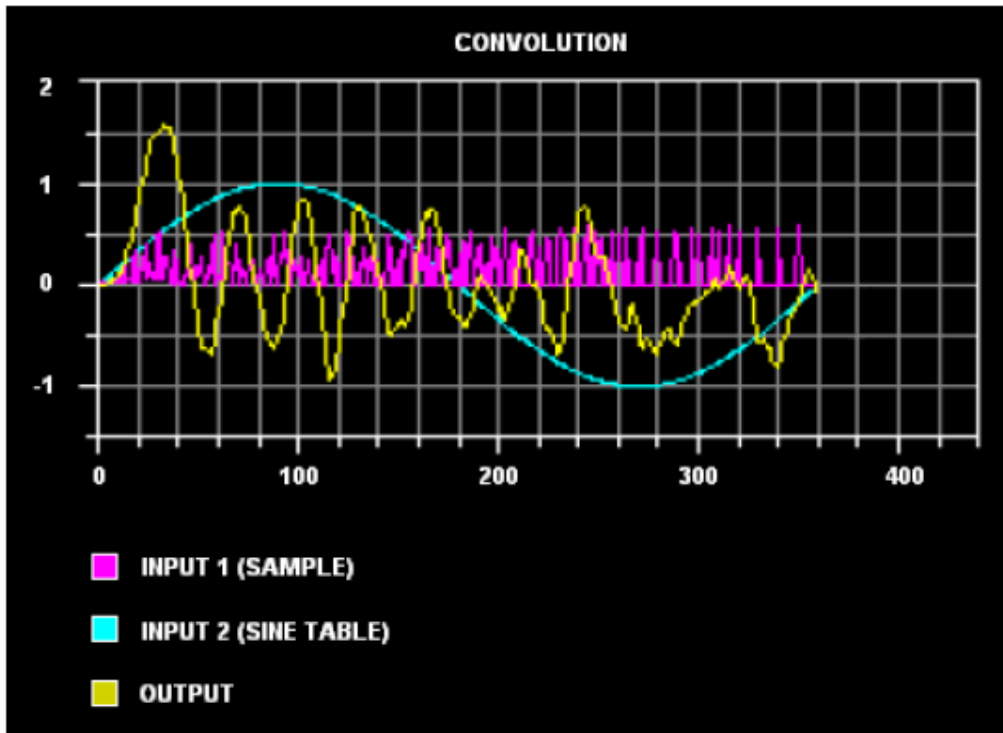


Figura 5- Resultado da convolução dentro do ambiente da IDE

Referências

[1] DSPTUTOR Website, www.dsptutor.freeuk.com

[2] ANALOG DEVICES , “Getting Started Guide”, www.analog.com

[3] TEXAS INSTRUMENTS website. www.ti.com

[4] Wikipedia ingles, <http://en.wikipedia.org/DSP>

[5] Wikipedia Português, <http://pt.wikipedia.org/wiki/DSP>

[6] <http://www.cbpf.br/~rastuto/main.html>