



## INTRODUÇÃO A PROCESSADORES DE SINAIS DIGITAIS - DSP

AUTORES:

RAFAEL ASTUTO AROUCHE NUNES

MARCELOS PORTES DE ALBUQUERQUE

MARCIO PORTES DE ALBUQUERQUE

FEVEREIRO 2006

## SUMÁRIO

RESUMO.....	3
INTRODUÇÃO.....	4
PARTE I: O PROCESSAMENTO DIGITAL DE SINAIS.....	5
PARTE II: O QUE É UM DSP? .....	6
2.1. O COMEÇO.....	6
2.2. O QUE É UM DSP? .....	6
2.3. APLICAÇÕES PARA O DSP.....	7
2.3.1. COMPRESSÃO E DESCOMPRESSÃO DE SINAL.....	7
2.3.2. FILTROS DIGITAIS.....	7
2.3.3. EQUIPAMENTOS ELETRÔNICOS.....	9
PARTE III: DESENVOLVIMENTO DE PROJETOS.....	9
PARTE IV: VISÃO GERAL DE UM DSP.....	12
PARTE V: EXEMPLOS DE APLICAÇÕES.....	13
5.1. CONVOLUÇÃO.....	13
5.2. SINTETIZADOR.....	15
REFERÊNCIAS.....	17

## RESUMO

Esta Nota Técnica aborda de forma objetiva o conceito do DSP (Digital Signal Processor). Conceitos básicos de processamento de sinal e suas aplicações em DSP e os principais tipos de DSP e sua funcionalidade são apresentados.

## INTRODUÇÃO

Nos últimos anos tem crescido muito a demanda por equipamentos portáteis, de fácil manuseio e que tenham uma interface simples com o usuário. O DSP procura unir todos esses preceitos de forma a se tornar uma ferramenta de trabalho das mais completas atualmente. Com suas mais diversas possibilidades de uso, nas mais abrangentes áreas, os Processadores de sinal digital são muito versáteis, podendo se enquadrar em quaisquer tipos de projetos.

## PARTE I: O PROCESSAMENTO DE SINAIS DIGITAIS

O processamento de sinais digitais consiste no método de analisar sinais do mundo real (representado por uma seqüência de números) usando ferramentas matemáticas, podendo assim realizar transformações ou extrair informações desses sinais.

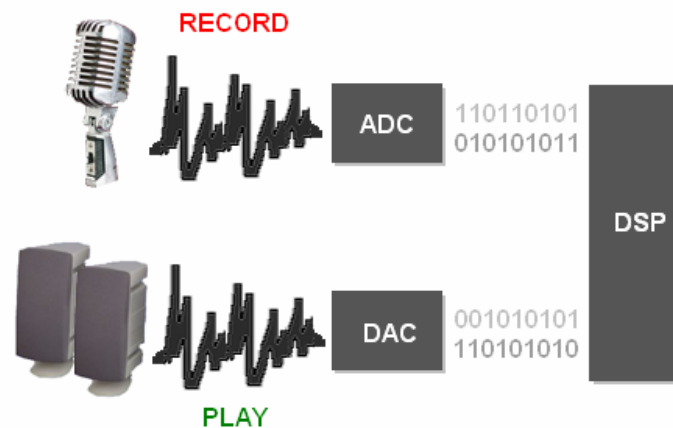
Nem a natureza nem os seres humanos utilizam sinais digitais para se comunicar no dia a dia. O mundo real consiste numa infinidade de sinais analógicos, que não são ou não podem ser entendidos em linguagem de máquina. Para que nossos computadores, celulares e quaisquer dispositivos eletrônicos possam trabalhar, é necessário que haja um processamento desses sinais analógicos na forma digital.

Para entender mais sobre o processamento digital de sinais é necessário primeiro que entendamos o que são, de onde vem e do que consistem os sinais analógicos. Estes sinais abrangem o som, a luz, pressão, temperatura, entre outros. Todas essas informações nos são enviadas o tempo todo e analisadas por nosso cérebro através de impulsos elétricos. Impulsos estes que também são uma forma de sinal analógico.

O sinal digital é justamente uma representação numérica de sinais analógicos. Para que haja essa representação, são utilizados conversores analógico-digitais (ADC), que processam os sinais analógicos e os transformam numa seqüência de 0s e 1s, podem assim ser analisados por computadores e programas.

Após a análise computacional, pode-se ainda realizar o reenvio destas informações para o mundo real. Para isso são utilizados conversores digital-analógicos (DAC), que tem a função de analisar uma determinada seqüência de 0s e 1s e transformá-la num sinal analógico.

Abaixo segue um pequeno exemplo de como ocorre a conversão dos sinais tanto na forma analógico-digital quanto na forma digital-analógica, em um processo de gravação e reprodução de voz.



**FIGURA 1: EXEMPLO DE CONVERSÃO E PROCESSAMENTO**

O ADC recebe o sinal de voz vinda do microfone e realiza a conversão para transformá-la em um sinal digital, podendo assim ser analisada pelo DSP. Após a análise feita pelo processador, ocorre a análise inversa, onde o DAC recebe o sinal digital e o converte para a forma analógica, enviando o sinal para os auto-falantes.

## PARTE II: O QUE É UM DSP?

### 2.1. O COMEÇO

Criado no começo da década de 80 pelas principais empresas de eletrônicos, como a *Texas Instruments*, *Analog Devices* e *Motorola*, o DSP se tornaria em pouco mais de uma década o centro das atenções no mundo da eletrônica, sendo hoje o coração de muitos equipamentos de diversas áreas da indústria.

Este surgiu com o propósito de se criar um microprocessador com uma arquitetura desenvolvida especificamente para operações que requeressem um processamento digital de sinais. Hoje em dia tem-se um produto que engloba, em um único chip, tecnologia suficiente para realizar praticamente qualquer tipo de processamento e análise de dados e sinais.

### 2.2. O QUE É UM DSP?

DSP na verdade representa dois significados, podendo ser utilizado como: Processamento Digital de Sinais (Digital Signal Processing) ou Processador de sinais digitais (Digital Signal Processor). Em ambos os casos seu uso esta correto, porem é mais comumente usado ao tratar-se do processador de sinais em si e não do ato de processar um sinal.

O DSP acima de tudo é um dispositivo programável, que detêm 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 desta forma a manipulação do microprocessador muito mais fácil e rápida.

Os DSP's foram projetados levando-se em consideração que 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, tornando 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. Celulares estes que hoje em dia, por utilizar a tecnologia dos processadores de sinal digital, permitem que ambos os lados possam falar simultaneamente, sem que o sinal seja cortado.

## 2.3. APLICAÇÕES PARA O DSP

A tecnologia DSP está presente atualmente em infinitos dispositivos como celulares, computadores multimídia, gravadores de vídeo, CD players, controlares de disco rígido e modems. Inclusive sendo desenvolvidas, atualmente, formas de se substituir os circuitos analógicos em televisores e em telefones. Desta forma o DSP se faz presente nas mais diversas áreas, como militar, médica, científica e automotiva.

### 2.3.1. COMPRESSÃO E DESCOMPRESSÃO DE SINAL

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

Outro exemplo bem característico está presente em câmeras multimídia, em PC's de todo o mundo, permitindo com que pessoas possam ver e transmitir imagens em tempo real, sem que haja interrupção do sinal.

Em sistemas de áudio que utilizam CD players, esta tecnologia DSP é utilizada para realizar uma complexa performance de detecção e correção de "raw data" (registro sem informação), diretamente a partir da leitura do disco.

### 2.3.2. FILTROS DIGITAIS

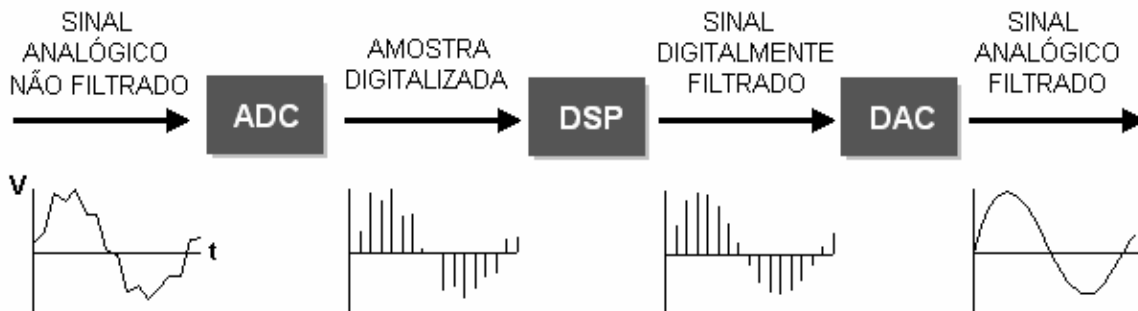
No processamento de sinais o papel do filtro é de remover partes indesejadas do sinal, bem como ruídos, ou simplesmente extrair partes importantes do mesmo, como componentes de uma certa banda de frequência.



**FIGURA 2:** ESQUEMA BÁSICO DE REPRESENTAÇÃO DA FILTRAGEM DE UM SINAL

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 as Transformadas de Fourier e Hilbert. Caso seja necessário, estes valores, que agora representam numericamente o sinal filtrado, podem ser convertidos através de um conversor digital-analógico (DAC).



**FIGURA 3:** ESQUEMA DE CONFIGURAÇÃO BÁSICA DE UM FILTRO DIGITAL [1]

Utilizar um filtro digital em um 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 torna 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 DSP's, 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 DSP's é a capacidade deste se adaptar automaticamente à característica de cada sinal, o que os torna muito mais versáteis no processo de filtragem.



### 2.3.3. EQUIPAMENTOS ELETRÔNICOS

O uso de DSP's 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 dito anteriormente o DSP é utilizado em larga escala na fabricação de telefones celulares, permitindo o 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 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 para qualquer produto analógico existe seu equivalente digital, seja com áudio, imagem, vídeo.

## PARTE III: DESENVOLVIMENTO DE PROJETOS

Por ser um dispositivo programável o DSP necessita de uma IDE própria para o desenvolvimento de projetos. É através das IDE's que são programados os DSP's. 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.

Uma 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++.

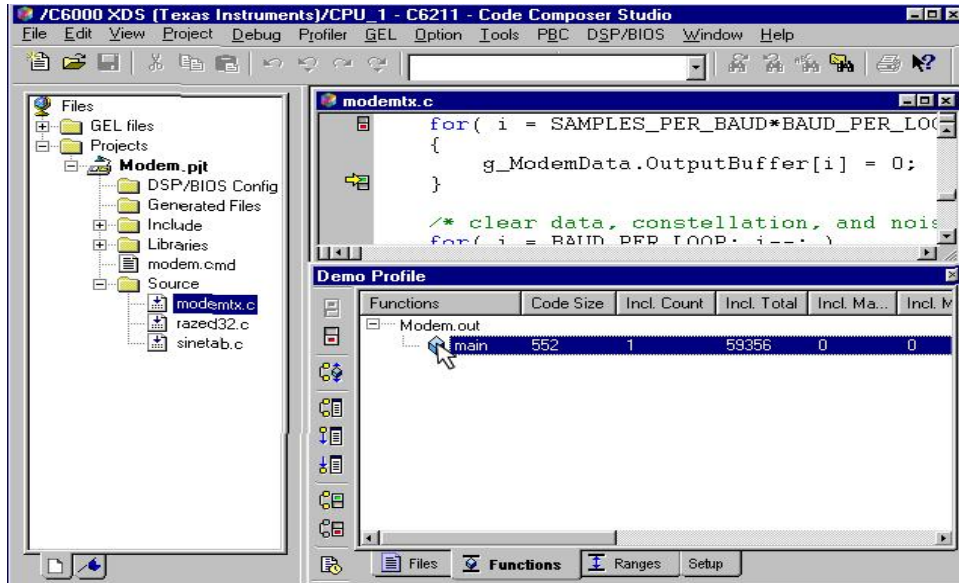


FIGURA 3: IDE CODE COMPOSER STUDIO (CCP) DA TEXAS INSTRUMENTS

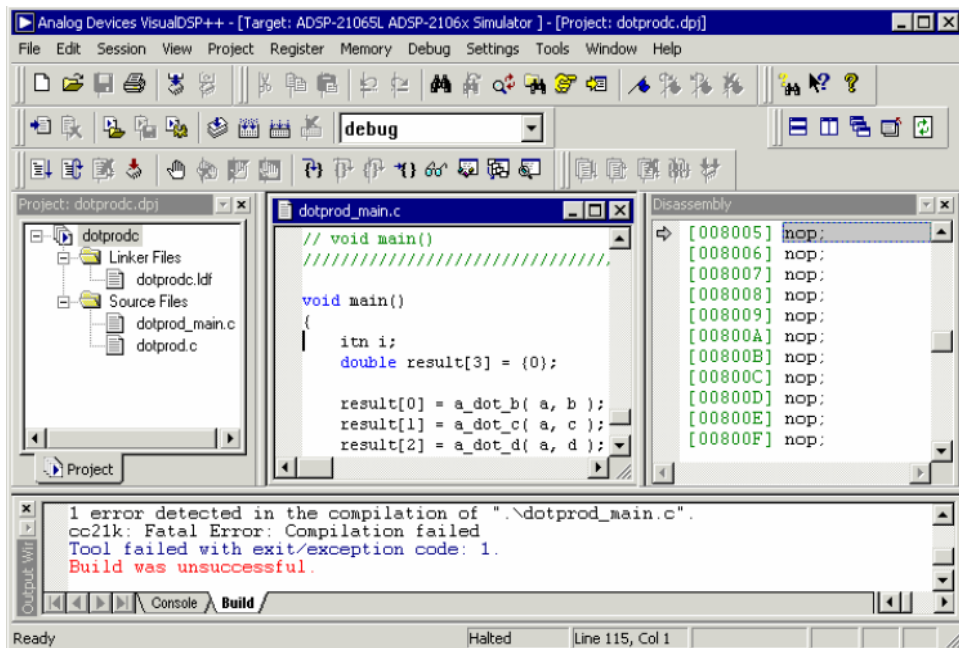


FIGURA 4: IDE VISUALDSP++ DA ANALOG DEVICES

A realização de um projeto utilizando-se a tecnologia DSP 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, através dos ambientes de desenvolvimento de cada DSP, as compilações do código, bem como todas as modificações e testes necessários.

Por fim temos a etapa de emulação, que envolve a realização de testes do projeto já dentro de 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 costuma-se encontrar muitos bugs (erros indesejados).

O DSP tem uma vantagem perante outros processadores, que neste caso é a possibilidade de upgrade, sem que haja necessidade da troca dos equipamentos. Através de interface JTAG pode-se modificar a qualquer momento o projeto, fazendo assim o DSP um aparato ideal que sempre estará atualizado.



**FIGURA 5:** EXEMPLO DE EMULADORES JTAG [2].

(A) XDS560 PCI-BUSSS JTAG SCAN-BASED EMULATOR (DSP8032U).

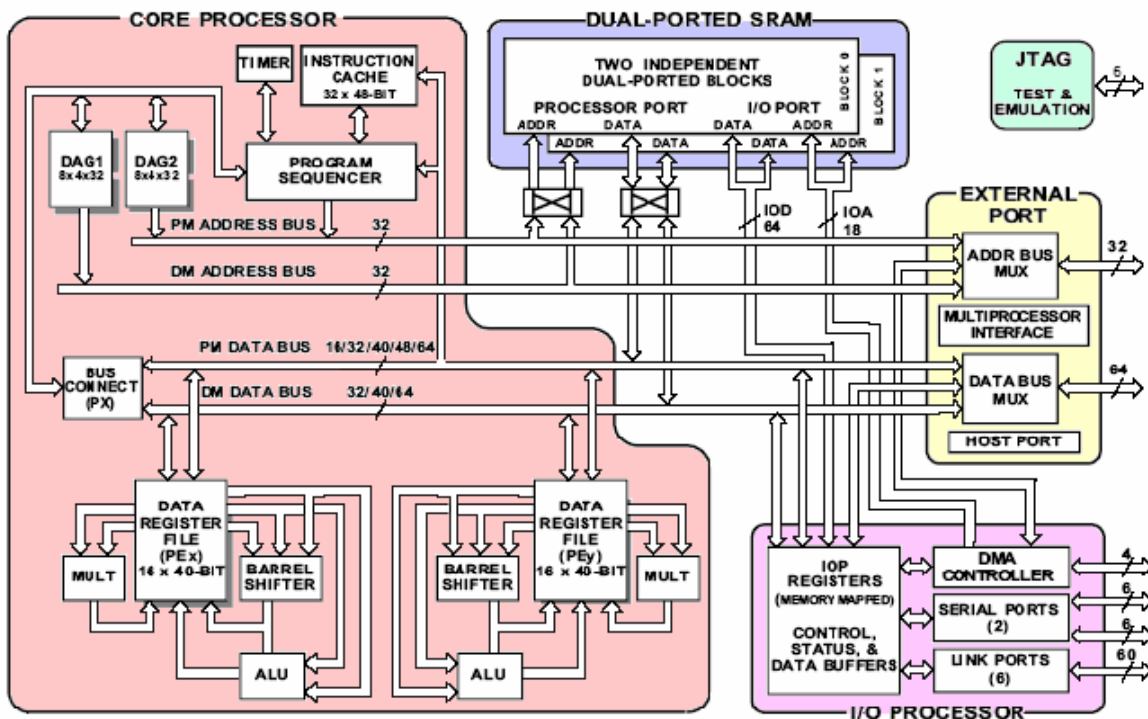
(B) BLACKHAWK XDS560 USB EMULATOR (DSP11990U)

## PARTE IV: 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 DSP's em sua maioria possuem 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 6:** DIAGRAMA DE BLOCOS DE UMA PLACA DSP

Dentro do *core processor* podemos encontrar sub-grupos 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 pode 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 realização de outras rotinas.

Alguns DSP's 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 dos sinais. Cada DSP vem acompanhado de um respectivo codec, que realiza as conversões necessárias para que ocorra seu processamento.

## PARTE V: EXEMPLOS DE APLICAÇÕES

Nesta seção serão mostrados alguns exemplos práticos de programas, funções e projetos que podem ser realizados em DSP's. Neste caso foi utilizado o DSP da Analog Devices, modelo ADSP-21160M, cujo software de desenvolvimento foi o VisualDSP++ 3.5.

### 5.1. 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 );
    }
}
```

```

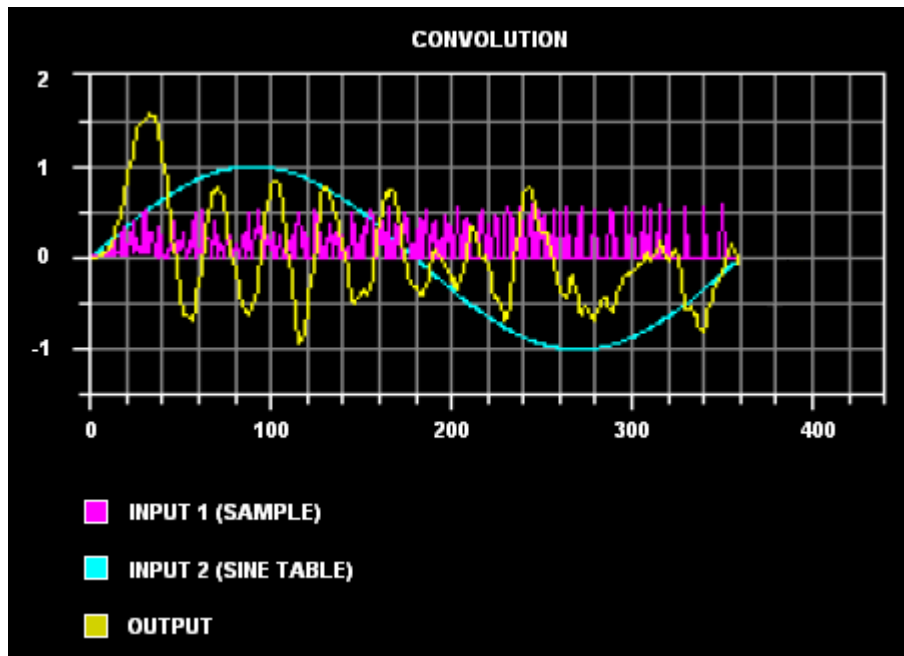
}

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]);
        }
    }
}

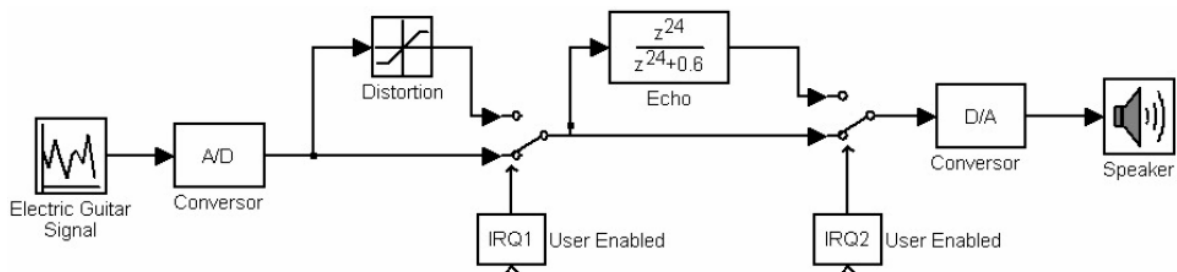
```



**FIGURA 6:** RESULTADO OBTIDO EM SIMULAÇÃO DENTRO DO AMBIENTE DA IDE. [3]

## 5.2. SINTETIZADOR

Rodrigo C. Torres [4] apresentou um esquema de um sintetizador simples que fornece dois efeitos de guitarra: distorção e eco. As ações são controladas por botões dentro da placa do DSP, cujos LED's correspondentes são acesos, indicando qual efeito está sendo utilizado.



**FIGURA 7: ESQUEMA DE FUNCIONAMENTO DO SINTETIZADOR**

```
void distortion()
{
    static float top = 15000000.0;
    left *= (left < 0) ? log(-1*left) : log(left);
    if (left > top) left = top;
    else if (left < -top) left = -top;
    left *= 2;
}

void echo()
{
    left = *pos = left + GAIN*(*pos);
    pos = (float *) circptr(pos, 1, buffer, DELAY);
}

void flanger()
{
    left *= ( (1 + sinf(2*3.14*0.5*t)) / 2);
    t += 2.0833e-5;
}

void intDistortion(int x)
{
    distEnabled = !distEnabled;
    set_flag(SET_FLAG0, TGL_FLAG);
}

void intEcho(int x)
{
    echoEnabled = !echoEnabled;
    set_flag(SET_FLAG1, TGL_FLAG);
    if (echoEnabled)
    {
        pos = buffer;
        #pragma SIMD_for
        for (int i=0; i<DELAY; i++) buffer[i] = 0.0;
    }
}
```

```

    }
}

void intFlanger(int x)
{
    flangerEnabled = !flangerEnabled;
    set_flag(SET_FLAG2, TGL_FLAG);
    t = 0.0;
}

void main()
{
    CSound codec;
    distEnabled = echoEnabled = flangerEnabled = false;
    set_flag (SET_FLAG0, CLR_FLAG);
    set_flag (SET_FLAG1, CLR_FLAG);
    set_flag (SET_FLAG2, CLR_FLAG);
    asm("bit set MODE2 IRQ0E | IRQ1E | IRQ2E;")
    interrupt (SIG_IRQ0, intDistortion);
    interrupt (SIG_IRQ1, intEcho);
    interrupt (SIG_IRQ2, intFlanger);
    while (true)
    {
        asm ("idle;");
        codec.getSamples (left, right);           // input do ADC
        if (distEnabled) distortion();
        if (echoEnabled) echo();
        if (flangerEnabled) flanger();
        codec.setSamples (left, left);           // output do DAC
    }
}

```



## REFERÊNCIAS

[1] DSPTUTOR website, <http://www.dsptutor.freeuk.com/>

[2] TEXAS INSTRUMENTS website, <http://www.ti.com/>

[3] ANALOG DEVICES Inc., "Getting Started Guide", Revision 4.0, Janeiro 2003,  
Part Number 82-001993-01, <http://www.analog.com/>

[4] TORRES, Rodrigo Coura, "DSP e Suas Aplicações", p.50, p.27, Outubro 2004,  
[http://www.cbpf.br/cat/pdsi/downloads/dsps\\_e\\_suas\\_aplicacoes\\_em\\_dsp.pdf/](http://www.cbpf.br/cat/pdsi/downloads/dsps_e_suas_aplicacoes_em_dsp.pdf/)