



## **Desenvolvimento de um filtro de resposta ao impulso finito em VHDL** **Erik Farias da Silva<sup>1</sup>, Kesia Cristiane dos Santos Farias<sup>2</sup>, Evandson Claude Seabra Dantas<sup>3</sup>, Johannes Warwick da Costa Farias<sup>4</sup>**

<sup>1</sup>Mestre em Eng. Elétrica - IFPB. e-mail: erik@ifpb.edu.br

<sup>2</sup>Mestre em Eng. Elétrica - IFPB. e-mail: kesia.farias@ifpb.edu.br

<sup>3</sup>Graduante do curso Sup. e Tecn. em Sistemas de Telecomunicações - IFPB. Bolsista do PIBIC. e-mail: evandson@live.com

<sup>4</sup>Graduante do curso Sup. e Tecn. em Sistemas de Telecomunicações - IFPB. e-mail: johanneswarwick@live.com

**Resumo:** A utilização de FPGAs (*Field Programmable Gate Arrays*) na área de processamento digital de sinais é uma solução distinta da clássica abordagem utilizando DSPs (*Digital Signal Processors*). O uso de FPGA no processamento digital de sinais é um tema que deve ser abordado desde a graduação em diversas disciplinas dos cursos de engenharia elétrica e tecnologia em sistemas de telecomunicações. Este trabalho aborda o desenvolvimento de um filtro digital do tipo FIR (*Finite Impulse Response*) na linguagem de descrição de *hardware* VHDL. Este trabalho é uma etapa do desenvolvimento de uma biblioteca pública de experimentos na área de processamento digital de sinais utilizando FPGA.

**Palavras-Chave:** Filtro digital, FPGA, VHDL

### **1. INTRODUÇÃO**

Sistemas de comunicações modernos são baseados no processamento de sinais digitais. O domínio do conhecimento dos processos relacionados à digitalização e tratamento de sinais é fundamental à qualificação profissional de alunos dos cursos de engenharia e tecnologia. Experimentos práticos relacionados ao tema podem enriquecer o ensino de disciplinas como processamento digital de sinais, comunicações digitais, entre outras. O desenvolvimento de uma biblioteca pública de experimentos na área de processamento digital de sinais possibilita uma padronização do ensino, permitindo uma maior interatividade nas pesquisas realizadas em diferentes Institutos Federais.

A utilização de filtros digitais é fundamental para um bom funcionamento dos sistemas de recepção de dados digitais. Além de tratar ruídos indesejáveis ao sistema, os filtros também possibilitam a divisão do canal de dados em várias partes distintas, otimizando o sistema para trabalhar com vários fluxos de dados no mesmo canal (DINIZ, SILVA e NETTO, 2004).

Os filtros são divididos em duas categorias principais, FIR e IIR (*Infinity Impulse Response*). A diferença fundamental entre esses filtros é que o primeiro não possui uma realimentação de dados, o que reduz o número de operações de soma e multiplicação necessárias, tornando sua realização mais simples quando comparado com um IIR de mesma ordem. Outra característica dos filtros FIR é sua estabilidade de fase, neste ela sempre é estável, enquanto nos IIR pode ocorrer instabilidade na fase.

Para filtros de mesma ordem, os IIR apresentam uma maior seletividade em frequência. A ausência de realimentação ocasiona uma menor seletividade nos FIR, sendo necessário elevar a ordem do filtro para compensar a redução da seletividade. Com uma ordem maior, o número de coeficientes e iterações aumenta, elevando assim o custo de *hardware* (HAYKIN e VEEN, 2001).

A utilização de dispositivos reconfiguráveis para a realização do processamento digital de sinais possibilita uma maior integração de funcionalidades em um mesmo dispositivo, o que é desejável em sistemas embarcados. Tipicamente o processamento digital de sinais é realizado por microcontroladores conhecidos como DSP. Os DSPs não são dispositivos reconfiguráveis, não sendo possível reformular sua arquitetura para adequar seu funcionamento a uma aplicação específica.

Há uma tendência de que diversas soluções, atualmente implementadas em DSP, migrem para dispositivos reconfiguráveis, a exemplo do FPGA. O fato de ser uma solução de *hardware*



dedicado torna o uso do FPGA mais atrativo devido ao seu maior desempenho, quando comparado ao DSP. Quando descrita, a arquitetura do FPGA deixa de ser genérica, e somente os recursos de *hardware* necessários são utilizados (MEYER-BAESE, 2001).

O FPGA é um dispositivo do tipo VHSIC (*Very High Speed Integrated Circuit*), e para esse tipo de dispositivo foi desenvolvida a linguagem VHDL (*VHSIC Hardware Description Language*). A linguagem define o circuito que será criado em nível de *hardware* (D'AMORE, 2010).

Para fins educacionais, os FPGAs também são vendidos em *kits* completos (nesse ponto ele torna-se um equipamento didático), em que é possível realizar um controle por meio de suas chaves e visualizar o que está acontecendo pelos seus indicadores. No ambiente dos Institutos Federais, é interessante ressaltar que um laboratório com diversos *kits* de FPGA pode servir a diversas disciplinas, como processamento digital de sinais, circuitos lógicos, sistemas digitais, comunicações digitais, codificação, entre outras.

A criação de uma biblioteca pública de funções pode enriquecer o ensino de disciplinas como PDS. Neste trabalho foi desenvolvido um filtro FIR que será parte integrante da biblioteca de experimentos. O filtro desenvolvido permite que seus parâmetros como frequência de corte e ordem do filtro sejam atribuídos pelo usuário.

## 2. FILTRO DIGITAL

Os filtros digitais são importantes elementos presentes nos sistemas de comunicações, possuindo a função de modificar ou alterar atributos de um sinal de tempo discreto, no domínio da frequência ou do tempo. (MEYER-BAESE, 2001).

As diversas vantagens do processamento digital de sinais (PDS) aliadas à acessibilidade dos circuitos integrados, tornaram o uso de filtros digitais mais comuns que os analógicos. Algumas características de destaque são:

- Alteração de parâmetros realizados por *software*.
- Versatilidade, processando sinais digitais de qualquer tipo.
- Replicabilidade, pode ser facilmente duplicado em outro sistema (HAYKIN e VEEN, 2001).

As duas principais categorias de filtros digitais são os filtros FIR (*Finite Impulse Response*) e filtros IIR (*Infinite Impulse Response*). O filtro FIR apresenta uma resposta ao impulso de duração finita, simplificando a soma de convolução. Por sua vez, o filtro IIR apresenta uma resposta ao impulso de duração infinita, devido à sua natureza recursiva (MEYER-BAESE, 2001).

Neste artigo optou-se pela criação de um filtro FIR. Uma de suas propriedades inerentes é apresentar uma resposta em frequência de fase linear, que corresponde a um retardo constante. Desse modo, o projeto de filtros FIR é simplificado, buscando uma aproximação da resposta em amplitude desejada (HAYKIN e VEEN, 2001).

Com relação à sua seletividade em frequência, os filtros podem ser de 6 tipos: Passa-baixa, passa-alta, passa-faixa, rejeita-faixa, passa tudo e rejeita tudo. Em geral, as equações do filtro passa-baixa são utilizadas para criar os demais tipos de filtros como é mostrado na . Sendo essas operações realizadas no domínio da frequência, é necessário fazer uma transformada inversa de Fourier para obter a resposta no domínio no tempo.

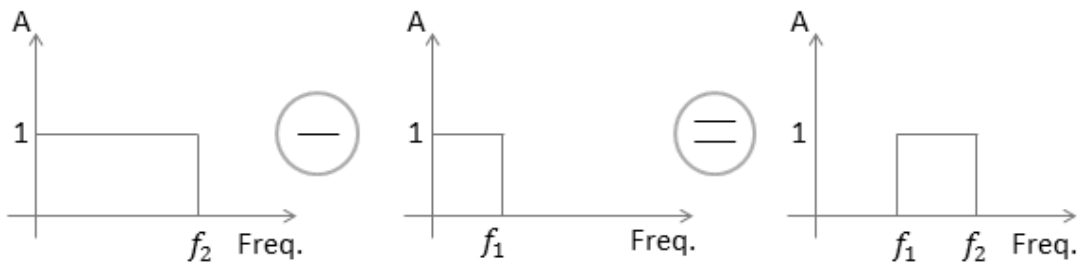


Figura 1 - Confeção de filtro passa faixa usando filtros passa baixa.

O janelamento é uma técnica que se utiliza de uma forma de onda quadrada para definir a faixa de operação ou largura de banda de um sistema. A multiplicação da janela por outro sinal no domínio da frequência apresenta-se no domínio do tempo como uma convolução.

Existem várias equações de janelamento, *softwares* como o LabView® e o MATLAB®, possuem funções pré-definidas que implementam diversas dessas equações, a citar: Retangular, Hanning, Hamming, Blackman, Kaiser-Bessel, entre outras. (ANDRADE e SOARES)

A janela de Kaiser-Bessel pode ter sua forma modificada pelo ajuste de um parâmetro  $\beta$ . Dessa maneira, modifica-se a forma da janela para controlar a sua perda espectral. A Fórmula 1 define uma janela Kaiser-Bessel com N amostras,

$$\omega_n = \begin{cases} \frac{I_0 \left( \beta \sqrt{1 - \left( \frac{2n}{M} - 1 \right)^2} \right)}{I_0 \beta}, & 0 \leq n \leq M \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

$$y_i = x_i \times \omega_n$$

onde  $I_0$  representa a primeira ordem da janela,  $\beta$  é um número arbitrário real que determina a forma da janela (relação entre número de lóbulos e seletividade de frequência) e M o número de amostras do sinal (HAYKIN e VEEN, 2001).

Após o cálculo da janela, são obtidos os coeficientes do filtro que são representados por  $h[n]$ . O próximo passo é a realização da soma de convolução da janela com o sinal de entrada, o que representa sua multiplicação no domínio da frequência. O sinal resultante da convolução é descrito pela Fórmula 2,

$$y[n] = \sum_{k=0}^{M-1} x[k] \times h[n - k], \quad (2)$$

em que M representa a ordem da janela calculada e n o número de elementos.

A Figura 2 ilustra o processo de convolução realizado no FPGA. Na etapa de inicialização, são carregados os coeficientes do filtro. Após o carregamento, para cada dado de entrada  $x[n]$ , é realizada uma multiplicação dos coeficientes pelos dados de entrada armazenados. O resultado dessas multiplicações é somado e em seguida o sinal de entrada é deslocado.

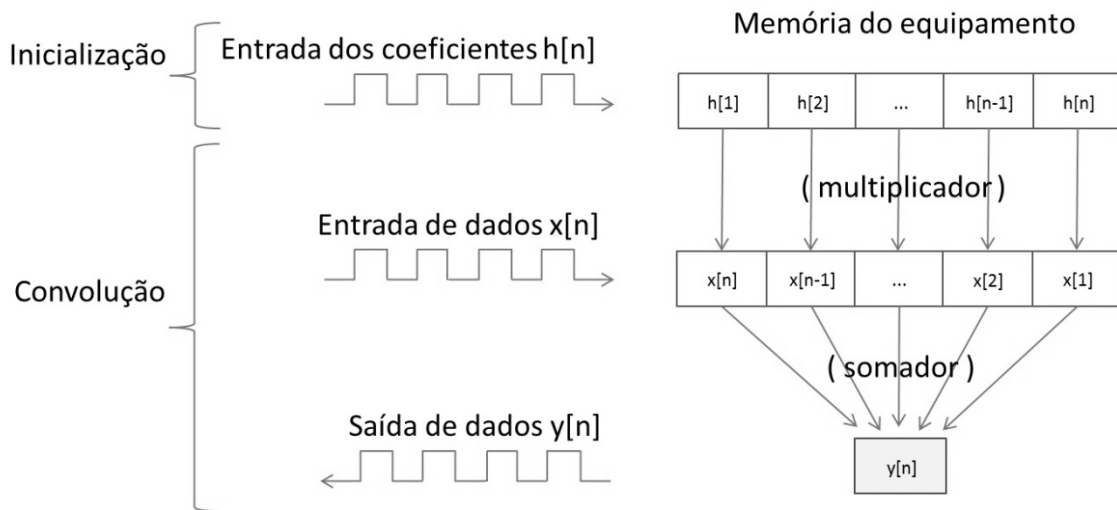


Figura 2 - Fluxograma da convolução em VHDL

### 3. IMPLEMENTAÇÃO EM VHDL

A descrição de *hardware* difere em diversos aspectos da programação em linguagens convencionais, a programação em VHDL apresenta diversas particularidades relativas à arquitetura de *hardware* desejada o que vai além do alcance do desenvolvimento de um *software*.

O desenvolvimento em VHDL é concorrente, ou seja, realiza instruções de forma paralela como ilustrado na Figura 3. Não atentar para essa particularidade pode gerar erros que não permitem a compilação do código, o mais comum é a atribuição de dois valores a uma mesma variável em pontos diferentes do código. Essa prática comum em linguagens de programação leva a um erro em VHDL, o erro ocorre devido ao paralelismo das instruções do código, o que leva à definição de dois valores concorrentes à mesma variável.

Para realizar uma sequência lógica não concorrente são criados processos, nos quais as instruções são executadas de maneira sequencial. Para evitar conflitos e respeitar as dependências existentes entre blocos concorrentes, foram utilizadas *flags* de controle para realizar transferência de dados entre processos.

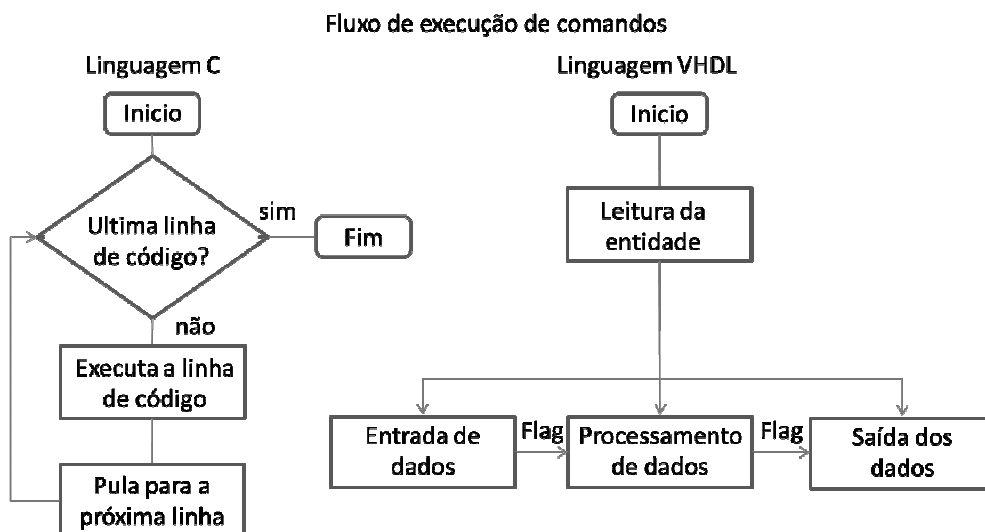


Figura 3 - Fluxograma comparativo entre a linguagem C e VHDL

A estrutura básica da linguagem pode ser dividida em duas partes, conforme ilustrado na Figura 4. A entidade é a responsável pela interface do equipamento com o ambiente exterior controlando as entradas e saídas de dados. A arquitetura é o circuito propriamente dito, que faz operações com as portas de entrada gerando sinais nas portas de saída (D'AMORE, 2010).

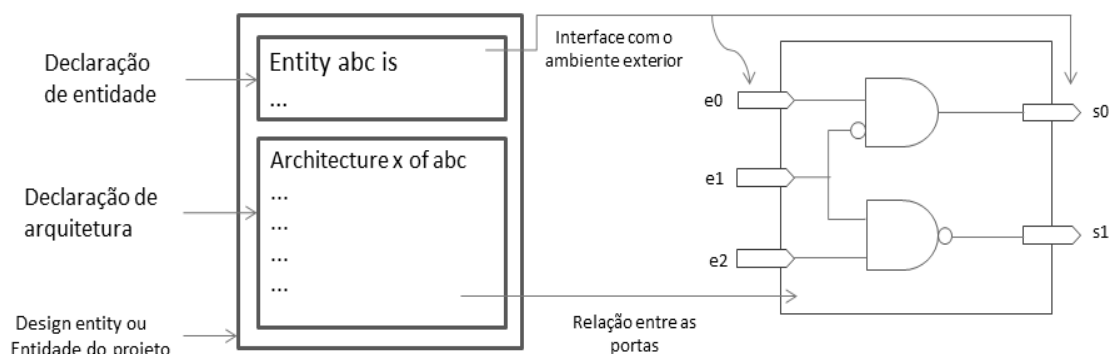


Figura 4 - Exemplo de entidade de projeto (D'AMORE, 2010)

Para realizar o filtro, inicialmente foi definida a entidade, nela são criadas as portas do sistema, bem como as constantes globais. Após a criação da entidade, a arquitetura foi definida por três processos: adequação da entrada de dados, convolução e saída de dados.

Na prática, um sinal analógico é digitalizado por um conversor A/D e posteriormente entregue ao processador de sinais.

O processo de adequação do sinal de entrada de dados realiza a formatação do sinal e o preenchimento de um *buffer* de entrada com número de *bits* coerente com a característica do filtro realizado.

O processo de convolução realiza a convolução entre o sinal do *buffer* de entrada e os coeficientes do filtro.

No processo de saída o resultado da convolução é armazenado e entregue à porta de saída. Esse processo ainda realiza um deslocamento de índices no *buffer* preparando assim a próxima iteração.

Os coeficientes do filtro podem ser calculados utilizando o MatLab<sup>®</sup> que dispõe de uma ferramenta chamada de “fdatool” para projetar filtros digitais.

#### 4. RESULTADOS E DISCUSSÃO

A simulação foi realizada no *software* ModelSim<sup>®</sup> e seus resultados exportados para o MatLab<sup>®</sup> onde foram gerados os gráficos apresentados neste artigo.

Foi realizado um filtro FIR passa baixas de 20<sup>ª</sup> ordem com frequência de corte em 3 kHz, frequência de amostragem de 8 kHz usando a janela Kaiser com  $\beta$  igual a 0,5. Os coeficientes desse filtro foram calculados utilizando a ferramenta “fdatool” do MatLab<sup>®</sup>.

Como entrada foi utilizado um sinal de voz com frequência de amostragem de 8 kHz. A Figura 5 ilustra o sinal de voz de entrada no domínio do tempo e da frequência. Ao analisar o espectro da voz verifica-se que a maior concentração de informação deve-se às componentes de baixa frequência, que representam a parte grave da voz. Ao remover a componente de alta frequência, cria-se o efeito de abafamento do sinal.

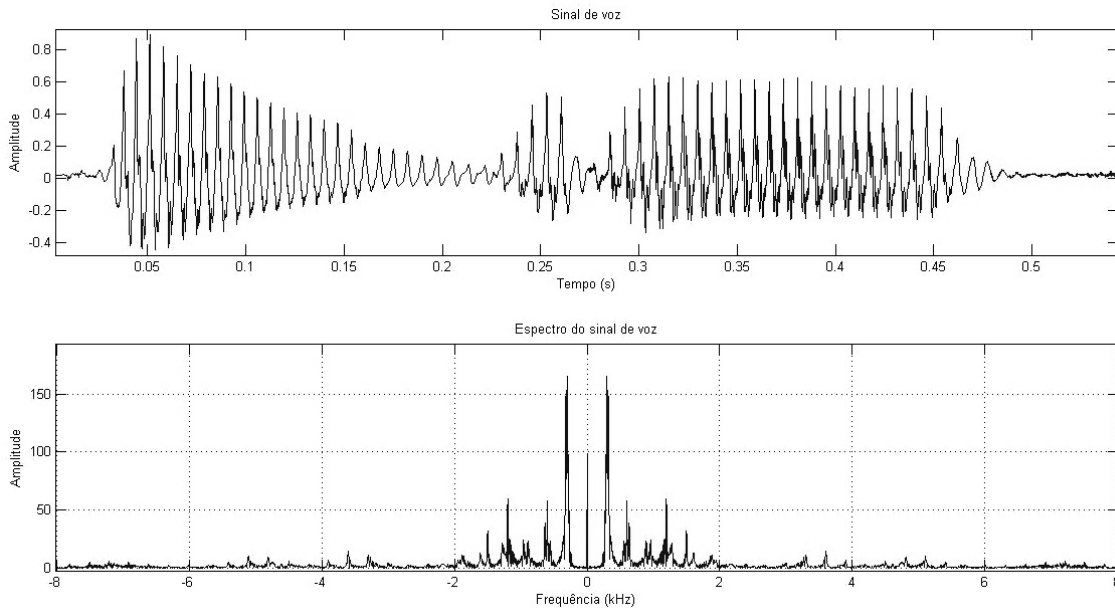


Figura 5 - Sinal de voz no domínio do tempo e da frequência.

A simulação foi realizada utilizando o ModelSim<sup>®</sup> e a linguagem utilizada foi a VHDL. Inicialmente dois arquivos binários contendo, respectivamente, os dados relativos ao sinal de áudio de entrada e os coeficientes do filtro são carregados na memória.

O *buffer* de entrada, inicialmente zerado, é preenchido gradativamente com os dados provenientes do arquivo binário já armazenado, enquanto a variável coeficientes é preenchida pelos dados do arquivo de coeficientes. Esse processo de carregamento é ilustrado pela Figura 6.

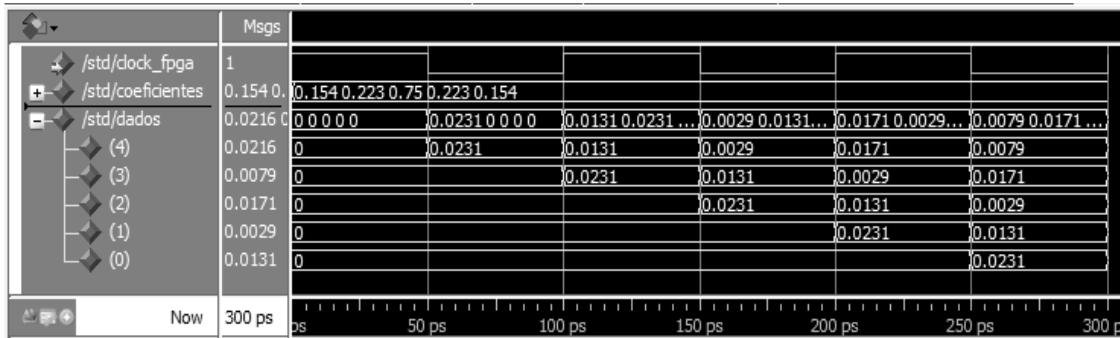


Figura 6 – Exemplo de processamento e análise do sinal de voz no ModelSim<sup>®</sup>.

O sinal resultante da filtragem foi salvo em um arquivo binário, este por sua vez foi utilizado no MatLab para a geração dos gráficos apresentados na Figura 7. Os gráficos representam o sinal filtrado nos domínios do tempo e frequência. No domínio da frequência pode-se verificar a redução na amplitude dos sinais com frequências acima de 3 kHz, o que sinaliza o correto funcionamento do filtro.

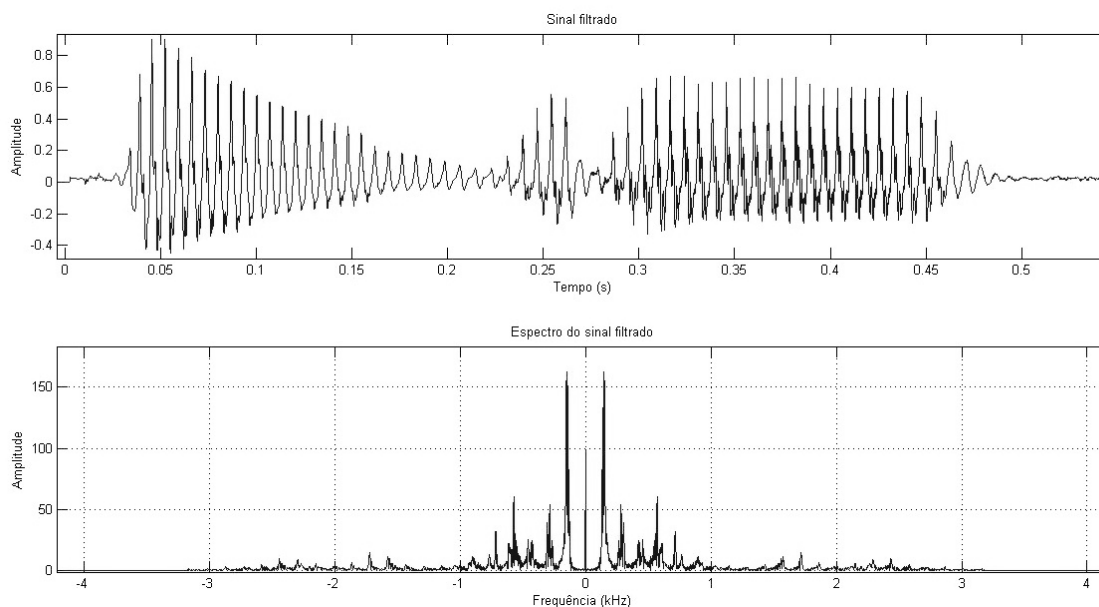


Figura 7 - Sinal de voz filtrado no ModelSim®.

Para obter a execução desejada do arquivo de áudio é importante conhecer a frequência de amostragem do sinal. Ao executar o arquivo antes e após a filtragem do sinal, foi possível perceber a redução no nível do ruído de altas frequências no arquivo filtrado. Entretanto a filtragem gera certa perda de qualidade tornando o som mais abafado.

## CONCLUSÕES

A liberdade de criação de *hardware* proporcionado pelo FPGA somado a uma ferramenta de simulação gera um ambiente interessante de pesquisa, estimulando o desenvolvimento de estudos na área de processamento digital de sinais.

A possibilidade de adequar o sinal antes de enviá-lo à etapa de processamento é uma vantagem associada ao uso do FPGA para processamento digital de sinais. Em um sistema que utiliza DSP, é necessário o uso de um dispositivo de interface capaz de adequar o número de *bits* do A/D ao número de *bits* dos registradores do DSP, ou escolher o conversor A/D compatível com o DSP. Como no FPGA é possível definir o tamanho dos registradores, também é possível integrar no mesmo dispositivo a etapa de adequação, permitindo que o processador se adapte a diferentes sistemas.

O estudo da linguagem VHDL e o uso de FPGA nos institutos federais é uma forma de estimular os estudantes integrando diversas áreas do conhecimento na produção de soluções para problemas abordados no curso. Por exemplo, um estudante da área de telecomunicações pode integrar, em um só dispositivo, projetos desenvolvidos em disciplinas de comunicações digitais, codificação, processamento digital de sinais e circuitos lógicos. O resultado é uma visão mais ampla e ao mesmo tempo detalhada do funcionamento de sistemas de telecomunicações.

O próximo passo da pesquisa é a integração do sistema simulado em um *kit* de desenvolvimento de FPGA. Para tal será desenvolvida a interface de comunicação entre o FPGA e um computador.

A criação de uma biblioteca pública de funções em VHDL, voltadas ao processamento digital de sinais, é uma etapa importante para a integração de diferentes institutos federais, sendo esse um dos desdobramentos futuros deste trabalho.



## REFERÊNCIAS

D'AMORE, R. **VHDL - Descrição e síntese de circuitos digitais**. 2ª. ed. [S.l.]: LTC, v. I, 2010.

DINIZ, P. S. R.; SILVA, E. A. B. D.; NETTO, S. L. **Processamento digital de sinais: projeto e análise de sistemas**. única. ed. [S.l.]: Bookman, 2004.

ANDRADE, A. O.; SOARES A. B. **Técnicas de Janelamento de Sinais**. *III Seminário dos Estudantes de Engenharia Elétrica da UFU*, Uberlândia-MG, 2000.

HAYKIN, S. S.; VEEN, B. V. **Sinais e Sistemas**. 1ª. ed. [S.l.]: Bookman, 2001. 688 p.

MELLO, C. A. **Processamento Digital de Sinais**. Recife: UFPE, 2011.

MEYER-BAESE, U. **Digital Signal Processing**. 3ª. ed. Tallahassee: Springer, 2001.

OPPENHEIM, A. V.; SCHAFER, R. W. **Discrete-Time Signal Processing**. 3ª. ed. [S.l.]: Prentice Hall, 2009. 1120 p.