

## **IMPLEMENTAÇÃO DE FILTROS DIGITAIS EM LINGUAGEM DE DESCRIÇÃO DE HARDWARE**

**RELATÓRIO FINAL DE PROJETO DE INICIAÇÃO CIENTÍFICA  
(PIBIC/CNPq/INPE)**

**Emanoel Raimundo Queiroz Chaves Junior (UFRN, Bolsista PIBIC/CNPq)**  
E-mail: emchavesjr@gmail.com

**Manoel Joseane Mafra de Carvalho (CRN/INPE, Orientador)**  
E-mail: manoel@crn.inpe.br

### **COLABORADORES**

**Dr. José Alberto Nicolau de Oliveira (DEE/UFRN)**  
**Sr. João Carlos Pécala Rae (CRN/INPE)**

**Julho de 2011**



## **AGRADECIMENTOS**

O bolsista Emanuel Raimundo Queiroz Chaves Junior é particularmente grato pela orientação recebida por Manoel Jozeane, João Carlos Pécala Rae e José Alberto Nicolau de Oliveira ao longo do projeto Transponder ITASAT.

O bolsista agradece também aos alunos de graduação Carlyle Câmara Santos Junior e Carlos Antônio Mendes da Costa Junior e aos alunos de mestrado Tiago Costa de Araújo, Hélio Sousa Peres e Jefferson Dantas Marinho pelas sugestões, após a leitura deste artigo.

E, finalmente, a Vinicius Araújo Cavalcanti e a Filipe Emanuel Vieira Taveiros pela ajuda na elaboração do código VHDL, à orientação de manuseamento do FPGA e com os testes em laboratório.



## **RESUMO**

Este documento tem como objetivo descrever amplamente o projeto de filtros FIR em linguagem de descrição de hardware que será utilizado no bloco de processamento digital do transponder que está sendo desenvolvido pelo INPE – CRN e que será embarcado no satélite universitário ITASAT, um projeto que envolve os órgãos INPE, ITA e AEB. Esta descrição parte desde a base teórica de sinais discretos até a estratégia de projeto e descrição em VHDL.



## **ABSTRACT**

This document has the goal to describe the project of FIR filters in hardware description language that will be used on the digital processing block of the transponder that is being developed by INPE –CRN and will be embedded on the university satellite ITASAT, an INPE, ITA and AEB project. This description ranges from the theory of discrete signals till the strategy of project and description in VHDL.



## LISTA DE FIGURAS

	Pág.
1. Resposta ideal do filtro passa-baixa	4
2.a. Resposta do filtro FIR à entrada impulso	5
2.b. Resposta com o atraso $M = \frac{N-1}{2}$	5
3. Resposta ideal do filtro passa-alta	6
4. Resposta ideal do filtro passa-faixa	7
5. Resposta ideal do filtro rejeita-baixa	7
6. Estrutura do filtro FIR	10
7. Fluxograma da descrição em HDL do filtro FIR	12



## LISTA DE TABELAS

	Pág.
1. Expressões matemáticas dos coeficientes para cada tipo de filtro.	8
2. Expressões matemáticas dos multiplicadores de cada tipo de método de janela.	9
3a. Numeração binária não sinalizada (unsigned).	10
3b. Numeração binária sinalizada (signed).	11

## **LISTA DE ABREVIACOES**

HDL – HARDWARE DESCRIPTION LANGUAGE

VHDL – VERY HIGH SPEED INTEGRATED CIRCUIT HARDWARE DESCRIPTION LANGUAGE

FPGA – FIELD PROGRAMABLE GATE ARRAY

FIR – FINITE IMPULSE RESPONSE

FFT – FAST FOURIER TRANSFORM

LPF – LOW PASS FILTER

HPF – HIGH PASS FILTER

BPF – BAND PASS FILTER

RBF – REJECT BAND FILTER



## SUMÁRIO

	Pág.
1. INTRODUÇÃO	1
2. FERRAMENTAS MATEMÁTICAS	3
Equações diferença	
Convolução discreta	
3. OBTENÇÃO DOS COEFICIENTES DO FILTRO FIR	4
Introdução	
Filtro passa-baixa	
Filtro passa-alta	
Filtro passa-faixa	
Filtro rejeita-faixa	
Métodos de janela	
4. IMPLEMENTAÇÃO DO FILTRO FIR	10
Estrutura	
Conversão dos coeficientes o sistema binário	
Fluxograma	
5. CONCLUSÃO	13
6. REFERÊNCIAS BIBLIOGRÁFICAS	14
APÊNDICE – JAVA FIR Filter	15



## 1. INTRODUÇÃO

A tecnologia digital surgiu e vem sendo largamente utilizada pela sociedade desde meados do século XX. Durante esse período, vários eventos serviram de motivação para se investir neste advento. Como por exemplo, a guerra fria, a corrida do homem ao espaço, o surgimento da informática, a necessidade de novas tecnologias de comunicação, automação e monitoramento, etc.

Neste cenário, foram desenvolvidos os filtros digitais FIR (*Finite Impulse Response*) e IIR (*Inifinite Impulse Response*) que são largamente utilizados em processamento digital de sinais ou tratamento digital.

O CRN-INPE propôs a construção do Transponder PCD para o satélite universitário ITASAT que abrange determinados blocos de projeto, sendo um deles, o de processamento digital que contém dois filtros FIR. O software tem a finalidade de obter os coeficientes do filtro, adequá-los a um método de janela que melhor convenha à escolha do projetista; e converter os resultados à representação binária para serem utilizados no projeto em FPGA da construção do bloco.

O grupo responsável pelo projeto do bloco de processamento digital atualmente se encontra na fase de realização de testes com o FPGA em laboratório.



## 2. FERRAMENTAS MATEMÁTICAS

### Equações diferença

A equação diferença é uma relação entre dois ou mais termos em sucessão. Como exemplo no exemplo abaixo:

$$y[k] = 2y[k - 1] + y[k - 2] + u[k] \quad (O)$$

Se considerarmos o sinal  $u[k]$  como sendo o sinal de entrada de um sistema qualquer e  $y[k]$  como sendo a sua respectiva saída, podemos interpretar a equação acima da seguinte forma: a saída do sistema no instante atual é igual a duas vezes a saída do instante anterior mais a saída de dois instantes anteriores ao atual mais o sinal de entrada atual.

### Convolução discreta

A convolução discreta entre dois sinais pode ser enunciada abaixo (Eq. L):

$$a[k] * b[k] = \sum_{n=0}^{\infty} a[n]b[k - n] \quad (L)$$

É possível obter a saída de um sistema através da convolução da respectiva entrada com a saída desse sistema à entrada impulso.

De acordo como a definição da convolução discreta, considerando  $u[n]$  como sendo o sinal de entrada e  $h[n]$  como sendo a saída ao impulso, a equação da saída à qualquer entrada do filtro FIR é:

$$y[k] = \sum_{n=0}^N h[n]u[k - n] \quad (K)$$

### 3. OBTENÇÃO DOS COEFICIENTES DO FILTRO FIR

#### 3.1 Introdução

O projeto de filtros FIR inicia-se pelo cálculo da expressão dos coeficientes do filtro observando a resposta que se deseja no domínio frequência. Essa obtenção é realizada através da Transformada Inversa de Fourier no tempo discreto que tem como definição a expressão abaixo (Eq. 1):

$$y[n] = \mathcal{F}^{-1}\{Y[\Omega]\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} Y[\Omega] e^{j\Omega n} d\Omega \quad (1)$$

Na expressão acima,  $\Omega$  é a frequência angular discreta é a razão entre a frequência angular do sinal e a taxa de amostragem (Eq. 2):

$$\Omega = \frac{\omega}{f_s} \quad (2)$$

#### 3.2 Filtro passa-baixa

Para um filtro passa-baixa, temos como região de banda passante, a faixa anterior à frequência de corte do filtro e com ganho é unitário. E a região além da frequência de corte, a banda de rejeição, com ganho nulo, como mostra a figura a seguir (Figura 1):

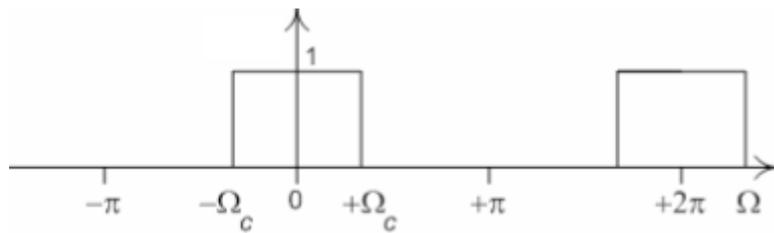


Figura 1. Resposta ideal do filtro passa-baixa

Calculando a Transformada Inversa de Fourier do sinal da Figura 1 obteremos a expressão abaixo (Eq. 3):

$$h[k] = \mathcal{F}^{-1}\{H[\Omega]\} = \frac{1}{2\pi} \int_{-\Omega_c}^{\Omega_c} e^{j\Omega k} d\Omega = \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{-\Omega_c}^{\Omega_c} = \frac{1}{j2\pi k} (e^{j\Omega_c k} - e^{-j\Omega_c k}) \quad (3)$$

Segundo a Identidade de Euler (Eq. 4), podemos representar a expressão acima como mostrada na equação 5:

$$\frac{1}{j2} (e^{j\theta} - e^{-j\theta}) = \text{sen}(\theta) \quad (4)$$

$$h[k] = \frac{1}{\pi k} \text{sen}(\Omega_c k) \quad (5)$$

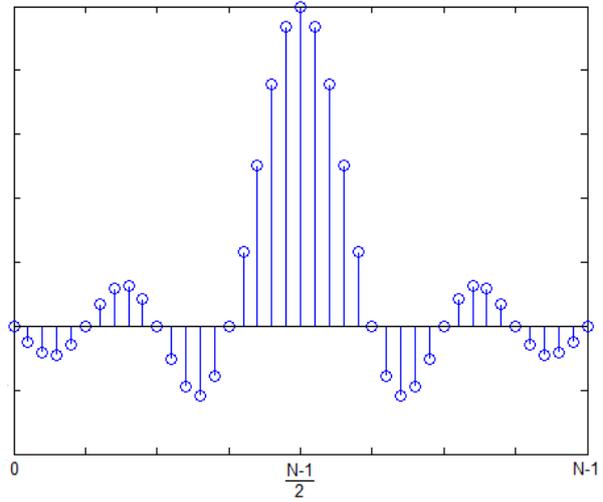
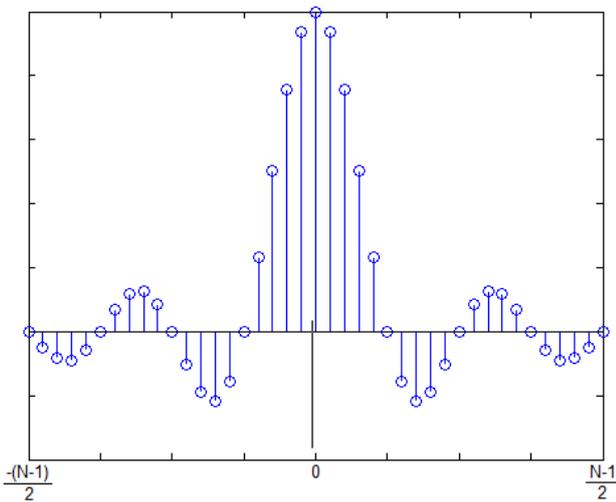
Fazendo um artifício matemático, multiplicando o numerador e o denominador da expressão anterior por  $\Omega_c$  teremos (eq. 6):

$$h[k] = \frac{\Omega_c}{\Omega_c \pi k} \text{sen}(\Omega_c k) = \frac{\Omega_c}{\pi} \text{sinc}(\Omega_c k) \quad (6)$$

Para se conseguir a resposta ideal mostrada na Figura 1, a quantidade de coeficientes do filtro ou termos  $h[k]$  deverá ser infinita o que é impossível.

O projetista precisa ter noção de que quão maior é a quantidade de coeficientes utilizada, a resposta será mais próxima da ideal, porém o esforço computacional será grande.

Como o intervalo de integração da Transformada Inversa de Fourier utilizada para a dedução da equação 6 é simétrico à origem, então a expressão é válida para instantes simétricos a origem, que vão de  $-\frac{N-1}{2}$  a  $\frac{N-1}{2}$  como mostrada na Figura 2a, sendo  $N$  a quantidade de coeficientes do filtro.



**Figura 2a. Resposta do filtro FIR à entrada impulso**      **Figura 2b. Resposta com o atraso  $M = \frac{N-1}{2}$**

Torna-se necessário realizar um deslocamento no tempo  $M$  (Eq. 7) por que o sistema na condição atual depende de instantes anteriores a zero o que algo impossível em sistemas não físicos.

$$M = \frac{N - 1}{2} \quad (7)$$

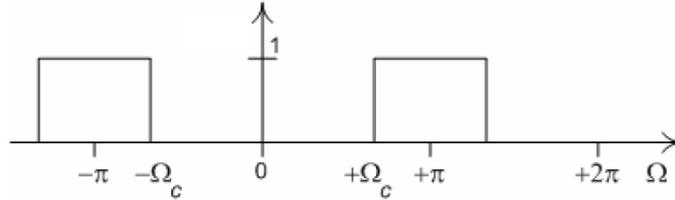
O sistema com o deslocamento no tempo passa a ser um sistema causal e está mostrado na Figura 2b.

Reescrevendo a equação 6 com o deslocamento M, obtemos a expressão dos coeficientes do filtro passa baixa causal (eq. 8).

$$h[k] = \frac{\Omega_c}{\pi} \text{sinc}[\Omega_c(M - k)] \quad 0 \leq k \leq N - 1 \quad (8)$$

### 3.3 Filtro passa-alta

Para um filtro passa-alta, temos como região de banda passante, a faixa após a frequência de corte do filtro como mostra a figura a seguir (Figura 3):



**Figura 3 Resposta ideal do filtro passa-alta**

Repetindo o mesmo processo realizado para calcular os coeficientes do filtro passa-baixa.

A Transformada de Fourier Inversa da resposta em frequência na Figura 3 (eq. 9):

$$\begin{aligned} h[k] &= \frac{1}{2\pi} \left[ \int_{-\pi}^{-\Omega_c} e^{j\Omega k} d\Omega + \int_{\Omega_c}^{\pi} e^{j\Omega k} d\Omega \right] = \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{-\pi}^{-\Omega_c} + \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{\Omega_c}^{\pi} = \\ &= \frac{1}{j2\pi k} (e^{-j\Omega_c k} - e^{-j\pi k} + e^{j\pi k} - e^{j\Omega_c k}) = \\ &= \frac{1}{\pi k} (\text{sen}(\pi k) - \text{sen}(\Omega_c k)) = \text{sinc}(\pi k) - \frac{\Omega_c}{\pi} \text{sinc}(\Omega_c k) \end{aligned} \quad (9)$$

Considerando o atraso M para que o sistema seja causal, a equação Y passa a ser a expressão abaixo (Eq. 10):

$$h[k] = \text{sinc}[\pi(M - k)] - \frac{\Omega_c}{\pi} \text{sinc}[\Omega_c(M - k)] \quad 0 \leq k \leq N - 1 \quad (10)$$

### 3.4 Filtro passa-banda

Para o filtro passa-banda, a região de banda passante compreende a faixa entre a frequência de corte inferior e a frequência de corte superior.

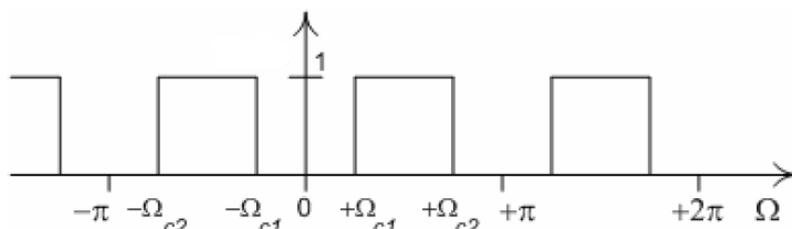


Figura 4. Resposta ideal do filtro passa-faixa

Realiza-se a Transformada Inversa de Fourier do sinal mostrado na Figura 4 para obter os coeficientes do filtro (eq. 11).

$$\begin{aligned}
 [k] &= \frac{1}{2\pi} \left[ \int_{-\Omega_{c2}}^{-\Omega_{c1}} e^{j\Omega k} d\Omega + \int_{\Omega_{c1}}^{\Omega_{c2}} e^{j\Omega k} d\Omega \right] = \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{-\Omega_{c2}}^{-\Omega_{c1}} + \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{\Omega_{c1}}^{\Omega_{c2}} = \\
 &= \frac{1}{j2\pi k} (e^{-j\Omega_{c1}k} - e^{-j\Omega_{c2}k} + e^{j\Omega_{c2}k} - e^{j\Omega_{c1}k}) = \\
 &= \frac{1}{\pi k} [\text{sen}(\Omega_{c2}k) - \text{sen}(\Omega_{c1}k)] = \frac{1}{\pi} [\Omega_{c2} \text{sinc}(\Omega_{c2}k) - \Omega_{c1} \text{sinc}(\Omega_{c1}k)] \quad (11)
 \end{aligned}$$

Com o atraso M a equação acima passa a ser (eq. 12):

$$h[k] = \frac{1}{\pi} \{ \Omega_{c2} \text{sinc}[\Omega_{c2}(M - k)] - \Omega_{c1} \text{sinc}[\Omega_{c1}(M - k)] \} \quad 0 \leq k \leq N - 1 \quad (12)$$

### 3.5 Filtro rejeita-banda

Para o filtro rejeita-banda, é a região de banda de rejeição que compreende a faixa entre a frequência de corte inferior e a frequência de corte superior, a região de banda passante encontra-se fora desta faixa.

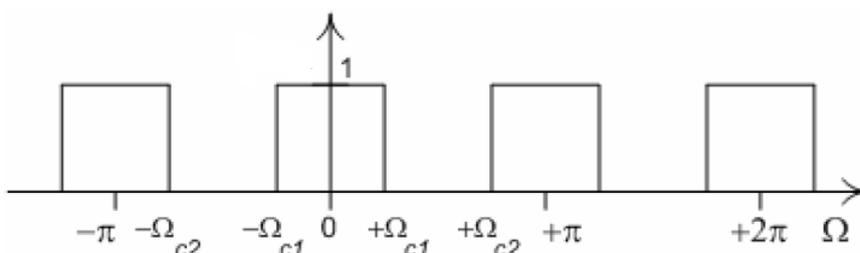


Figura 5. Resposta ideal do filtro rejeita-faixa

Transformada Inversa de Fourier do sinal mostrado na Figura 5:

$$\begin{aligned}
[k] &= \frac{1}{2\pi} \left[ \int_{-\pi}^{-\Omega_{c2}} e^{j\Omega k} d\Omega + \int_{-\Omega_{c1}}^{\Omega_{c1}} e^{j\Omega k} d\Omega + \int_{\Omega_{c2}}^{\pi} e^{j\Omega k} d\Omega \right] = \\
&= \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{-\pi}^{-\Omega_{c2}} + \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{-\Omega_{c1}}^{\Omega_{c1}} + \frac{1}{j2\pi k} e^{j\Omega k} \Big|_{\Omega_{c2}}^{\pi} = \\
&= \frac{1}{j2\pi k} (e^{-j\Omega_{c2}k} - e^{-j\pi k} + e^{j\Omega_{c1}k} - e^{-j\Omega_{c1}k} + e^{j\pi k} - e^{j\Omega_{c2}k}) = \\
&= \frac{1}{\pi k} (\text{sen}(\pi k) - \text{sen}(\Omega_{c2}k) + \text{sen}(\Omega_{c1}k)) = \\
&= \text{sinc}(\pi k) - \frac{1}{\pi} [\Omega_{c2} \text{sinc}(\Omega_{c2}k) - \Omega_{c1} \text{sinc}(\Omega_{c1}k)] \tag{13}
\end{aligned}$$

Com o atraso M a equação 13 acima passa a ser (Eq. 14):

$$\begin{aligned}
h[k] &= \text{sinc}[\pi(M - k)] - \frac{1}{\pi} \{ \Omega_{c2} \text{sinc}[\Omega_{c2}(M - k)] - \Omega_{c1} \text{sinc}[\Omega_{c1}(M - k)] \} \\
&0 \leq k \leq N - 1 \tag{14}
\end{aligned}$$

Organizando as expressões dos coeficientes para cada uma dos quatro tipos de filtro, temos a tabela abaixo (Tabela 1):

Tipos de filtros	Expressão matemática dos coeficientes
Passa-baixa	$h[k] = \frac{\Omega_c}{\pi} \text{sinc}[\Omega_c(M - k)]$
Passa-alta	$h[k] = \text{sinc}[\pi(M - k)] - \frac{\Omega_c}{\pi} \text{sinc}[\Omega_c(M - k)]$
Passa-faixa	$h[k] = \frac{1}{\pi} \{ \Omega_{c2} \text{sinc}[\Omega_{c2}(M - k)] - \Omega_{c1} \text{sinc}[\Omega_{c1}(M - k)] \}$
Rejeita-Faixa	$h[k] = \text{sinc}[\pi(M - k)] - \frac{1}{\pi} \{ \Omega_{c2} \text{sinc}[\Omega_{c2}(M - k)] - \Omega_{c1} \text{sinc}[\Omega_{c1}(M - k)] \}$

**Tabela 4.** Expressões matemáticas dos coeficientes para cada tipo de filtro

### 3.6 Métodos De Janela

Como já foi falado anteriormente, no contexto do projeto, para que um filtro FIR seja ideal, a quantidade de coeficientes do filtro deverá ser infinita, o que é uma coisa impossível de se conseguir.

Os métodos ou técnicas de janela são multiplicadores (Eq. 15) que manipulam os coeficientes do filtro a fim de proporcionar uma melhor distribuição de energia e assim obter um melhor desempenho.

Os métodos de janela mais comuns são: retangular, triangular ou Bartlett, Hanning, Hamming e Blackman.

$$h_{janela}[k] = h[k] \times w[k] \quad (15)$$

<b>Método de janela</b>	<b>Expressão matemática</b>
Retangular	$w[k + M] = 1 \quad -M \leq k \leq M$
Triangular (Bartlett)	$w[k + M] = 1 - \frac{ k }{M} \quad -M \leq k \leq M$
Hanning	$w[k + M] = 0.5 + 0.5 \cos\left(\frac{k\pi}{M}\right) \quad -M \leq k \leq M$
Hamming	$w[k + M] = 0.54 + 0.46 \cos\left(\frac{k\pi}{M}\right) \quad -M \leq k \leq M$
Blackman	$w[k + M] = 0.42 + 0.5 \cos\left(\frac{k\pi}{M}\right) + 0.08 \cos\left(\frac{2k\pi}{M}\right) \quad -M \leq k \leq M$

**Tabela 5. Expressões matemáticas dos multiplicadores de cada tipo de método de janela**

## 4. IMPLEMENTAÇÃO DO FILTRO FIR

### 4.1 Estrutura

Uma vez obtido os coeficientes da resposta ao impulso do filtro FIR desejado, é possível obter a saída deste sistema a qualquer entrada utilizando a convolução discreta entre os coeficientes do filtro e o sinal a ser filtrado como está definido na equação K. Se expandirmos esta equação, obteremos a seguinte equação diferença (Eq. 16):

$$y[k] = \sum_{n=0}^N h[n]u[k-n] = h[0]u[k] + h[1]u[k-1] + \dots + h[N]u[k-N] \quad (16)$$

Podemos representar a equação anterior como um sistema de blocos (Figura 6):

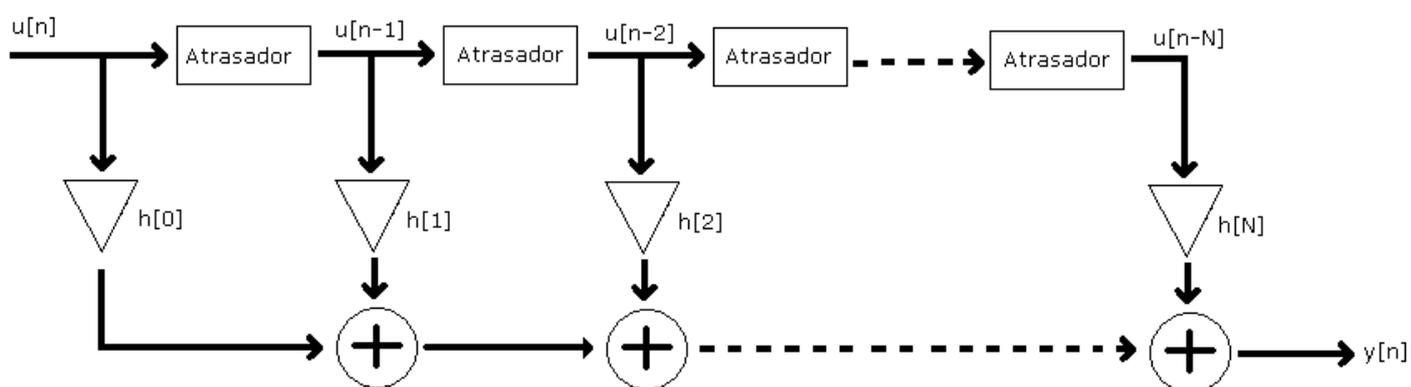


Figura 6. Estrutura do filtro FIR

### 4.2 Conversão dos coeficientes para o sistema binário

Analisando as tabelas 3a e 3b abaixo referentes aos sistemas binários não-sinalizados (unsigned) e sinalizados (signed) de 3 bits respectivamente como exemplo, observa-se que a quantidade de números em ambas as representação binárias é limitada pela quantidade de bits do sistema. Na numeração não-sinalizada, o conjunto é formado de  $2^N$  números entre 0 a  $2^N - 1$ , sendo N a quantidade de bits. Na numeração sinalizada, o conjunto também é composto de  $2^N$  números, porém entre  $-2^{N-1}$  a  $2^{N-1} - 1$ .

Nº decimal	Nº binário	Nº decimal	Nº binário
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

Tabela 6a. Numeração binária não-sinalizada (unsigned)

Nº decimal	Nº binário	Nº decimal	Nº binário
-4	100	0	000
-3	101	1	001
-2	110	2	010
-1	111	3	011

**Tabela 3b. Numeração binária sinalizada (signed)**

Para implementação do filtro FIR em VHDL, é necessário a conversão dos coeficientes para a notação binária por que esta é a notação numérica reconhecida pelo hardware. Os coeficientes do filtro são número do sistema real, em sua maioria, fracionários. Esta característica tornaria a conversão binária dos coeficientes, um processo bastante complicado de ser realizado. Para conseguir reverter esta situação e tornar a conversão dos coeficientes para o sistema binário mais fácil, realiza-se uma normalização dos coeficientes usando o maior valor entre eles através do seguinte raciocínio:

Visto que os coeficientes do filtro podem ser números fracionários, positivos ou negativos, usaremos a representação binária sinalizada. Sabendo que o maior número dessa representação é  $2^{N-1} - 1$ , realiza-se a normalização utilizando a seguinte expressão (eq.) na qual a função “round()” significa substituir o valor pelo próximo número inteiro na sequência (Eq. 17):

$$\text{round}\left(\frac{h[n](2^{N-1} - 1)}{\text{Max}(h[n])}\right) \quad (17)$$

Sabendo que os coeficientes normalizados pela expressão acima serão todos números inteiros, basta utilizar qualquer método para converter um número inteiro em número complexo. Por exemplo, na linguagem JAVA existe um método chamado *toBinaryString()* que recebe um valor inteiro e retorna a respectiva representação binária sinalizada como String. No apêndice deste artigo há uma apresentação sobre um software que desenvolvemos usando a linguagem JAVA que é muito útil em projeto de filtros FIR. Este software fornece os coeficientes já na notação binária tendo o usuário informado apenas os dados fundamentais do filtro e o número de bits para a representação.

### 4.3 Fluxograma

A implementação do filtro é feita através da sua descrição em HDL seguido pela programação da mesma em um FPGA. Na figura abaixo (Figura 8) está o fluxograma da descrição do filtro FIR em que N é o tamanho do filtro, h[n] são os N coeficientes, u[n] é o sinal de entrada e y[n] é o sinal de saída.

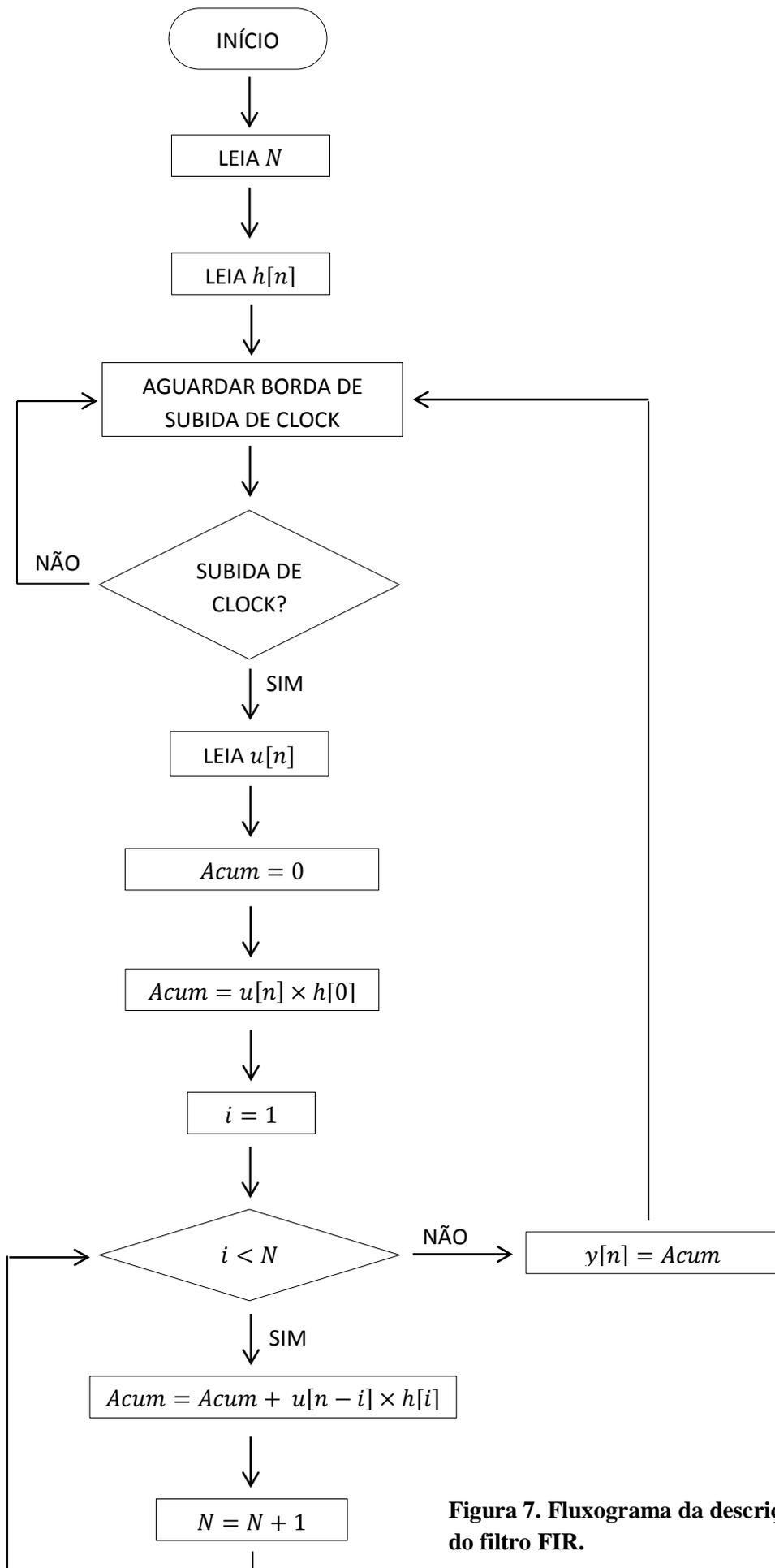


Figura 7. Fluxograma da descrição em HDL do filtro FIR.

## 5. CONCLUSÃO

O objetivo do grupo responsável pelo bloco de processamento digital é conseguir programar o VHDL do filtro FIR no FPGA da Actel, por este ser mais apropriado para utilização em espaço. Porém este modelo não dispõe de um conversor Analógico-to-Digital nem de um conversor Digital-to-Analógico integrados à placa. Para realização dos primeiros testes, utilizamos outro FPGA, o modelo Cyclone II DSP da Altera que já dispõe dos dispositivos citados integrados.

Já conseguimos bons resultados realizando testes em laboratório. O sinal na saída do FPGA encontrou-se dentro da banda de passagem, apresentou pouca atenuação e desvio de fase, como já era esperado. Necessitamos realizar mais testes, principalmente para as situações que se espera que o satélite encontre no espaço.

Finalizados os testes com o FPGA Altera, daremos início à implementação de uma interface que possibilite conectar os conversores Analógico-to-Digital e Digital-to-Analógico à placa Actel e, em seguida, repetir os testes realizados com o FPGA da Altera no FPGA da Actel.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

THEDE, LES. **Practical Analog and Digital Filter Design**. Arthec House, Inc. 2004.

TAN, LI. **Digital Signal Processing: Fundamentals and Applications**. Elsevier. 2008.

HAYKIN, SIMON. VEEN, BARRY VAN. **Sinais e Sistemas**. Bookman. 2006.

MEYER-BAESE, UWE. **Digital Signal Processing with Field Programmable Gate Arrays**. Springer. 2004.

SHENOI, B. A. **Introduction to Digital Signal Processing and Filter Design**. John Wiley & Sons. 2006.

IAN GROUT . **Digital Systems Design with FPGAs and CPLDs**. Elsevier. 2008.

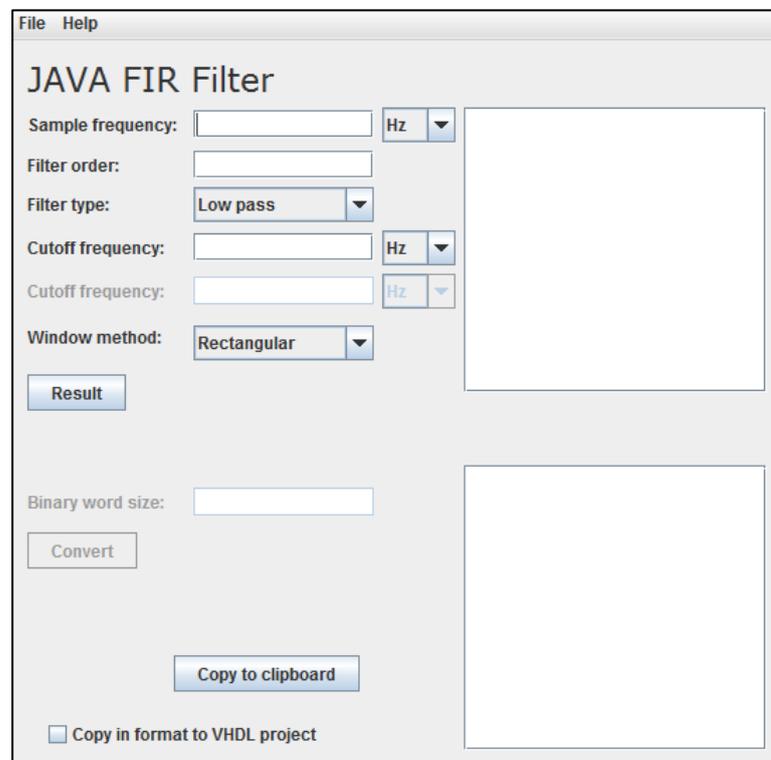
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. **Manual para elaboração, formatação e submissão de teses, dissertações e outras publicações do INPE**. 2011.

## APÊNDICE – JAVA FIR Filter

O projeto de filtros FIR demanda uma grande quantidade de cálculos a serem feitos. Esta quantidade é proporcional ao tamanho do filtro e conseqüentemente, como visto, à qualidade da resposta do mesmo.

A fim de tornar o projeto mais prático, desenvolvemos um software na linguagem JAVA que é capaz de realizar os cálculos necessários para obtenção dos coeficientes do filtro, converter esses coeficientes para a notação binária desejada e podendo copiar o resultado para a memória do computador do usuário para já ser inserido no código em HDL.

Abaixo está uma figura da janela inicial do JAVA FIR Filter (Figura A1):



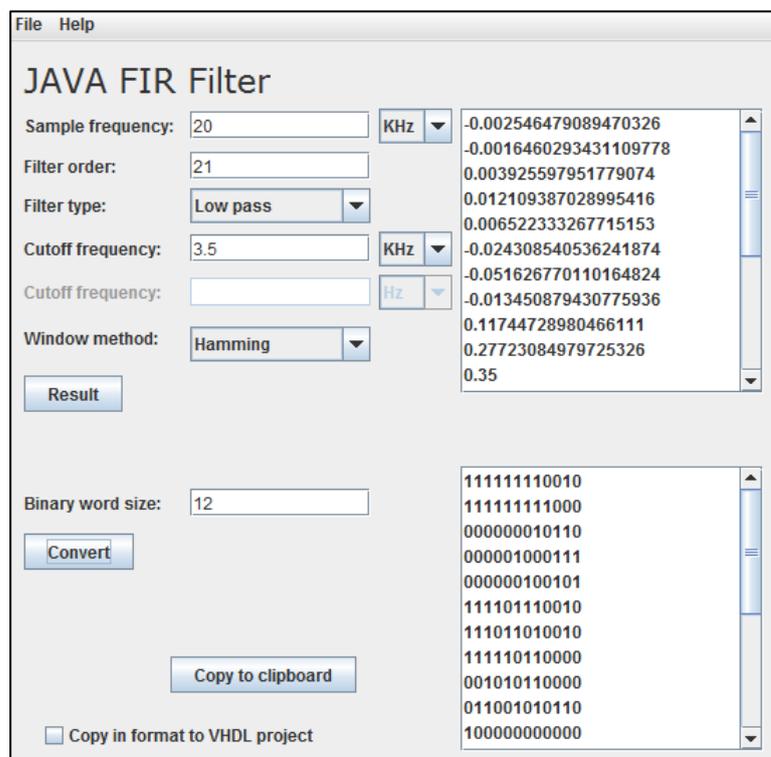
**Figura A1. Janela inicial do JAVA FIR Filter**

Demonstraremos o funcionamento do software através do exemplo:

Deseja-se projetar um filtro FIR passa-baixa de tamanho 21 e frequência de corte 3.5 kHz . A frequência de amostragem deverá ser 20 kHz na janela de Hamming e os coeficientes devem ser representados com 12 bits.

Clicando no botão “Result” após ter preenchido os campos acima do mesmo, os coeficientes do filtro desejado aparecerão no quadro superior direito. Em seguida o campo abaixo “Binary word size” o botão abaixo “Convert” ficaram disponíveis para uso. Preenchendo o campo com 12 e clicando no botão citado, os coeficientes representados na notação binária aparecerão no quadro inferior direito.

Abaixo está a figura com o resultado do exemplo (Figura A2):



**Figura A2. Resultado do exemplo**

Após a obtenção dos coeficientes na representação binária, é possível exportar os resultados para a memória clipboard clicando no botão “Copy to clipboard”, podendo ser exportada já na estrutura do projeto em VHDL. Veja a tabela a seguir que mostra o resultado em ambas as possibilidades

Opção “Copy in format to VHDL Project” desmarcada	Opção “Copy in format to VHDL Project” marcada
<pre> 111111110010 111111111000 000000010110 000001000111 000000100101 111101110010 111011010010 111110110000 001010110000 011001010110 100000000000 011001010110 001010110000 111110110000 111011010010 111101110010 000000100101 000001000111 000000010110 111111111000 111111110010                     </pre>	<pre> "111111110010", "111111111000", "000000010110", "000001000111", "000000100101", "111101110010", "111011010010", "111110110000", "001010110000", "011001010110", "100000000000", "011001010110", "001010110000", "111110110000", "111011010010", "111101110010", "000000100101", "000001000111", "000000010110", "111111111000", "111111110010";                     </pre>

**Tabela 7. Formas de exportar os coeficientes**