

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE ENERGIA**

**BRENO MENDES FERREIRA**

**MODELAGEM E IMPLEMENTAÇÃO DE UM SISTEMA DE  
PROCESSAMENTO DIGITAL DE SINAIS BASEADO EM FPGA  
PARA GERAÇÃO DE IMAGENS POR ULTRASSOM USANDO O  
SIMULINK**

**DISSERTAÇÃO DE MESTRADO**

**CURITIBA**

**2017**



**BRENO MENDES FERREIRA**

**MODELAGEM E IMPLEMENTAÇÃO DE UM SISTEMA DE  
PROCESSAMENTO DIGITAL DE SINAIS PARA GERAÇÃO DE  
IMAGENS BASEADO EM FPGA POR ULTRASSOM USANDO O  
SIMULINK**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Sistemas de Energia da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de “Mestre em Engenharia Elétrica” – Área de Concentração: Automação e Sistemas de Energia.

Orientador: Prof. Dr. Amauri Amorin Assef

**CURITIBA**

**2017**

---

**Dados Internacionais de Catalogação na Publicação**

---

F383m  
2017

Ferreira, Breno Mendes  
Modelagem e implementação de um sistema de processamento digital de sinais baseado em FPGA para geração de imagens por ultrassom usando o Simulink / Breno Mendes Ferreira.- 2017.  
115 p.: il.; 30 cm.

Disponível também via World Wide Web.  
Texto em português, com resumo em inglês.  
Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Sistemas de Energia. Área de Concentração: Automação e Sistemas de Energia, Curitiba, 2017.  
Bibliografia: p. 102-115.

1. Ultrassom. 2. Processamento de sinais - Técnicas digitais. 3. Arranjos de lógica programável em campo. 4. Reconstrução de imagens. 5. Beamforming. 6. Programação em tempo real. 7. SIMULINK (Programa de computador). 8. Métodos de simulação. 9. Sistemas de energia elétrica - Dissertações. I. Assef, Amauri Amorin, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Sistemas de Energia. III. Título.

CDD: Ed. 22 -- 621.31

---

**Biblioteca Central do Câmpus Curitiba - UTFPR**

Título da Dissertação Nº. 004

# Modelagem e Implementação de um Sistema de Processamento Digital de Sinais Baseado em FPGA para Geração de Imagens por Ultrassom Usando o Simulink

por

**Breno Mendes Ferreira**

Orientador: Prof. Dr. Amauri Amorin Assef (UTFPR)

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM ENGENHARIA ELÉTRICA – Área de Concentração: **Automação e Sistemas de Energia** do Programa de Pós-Graduação em Sistemas de Energia – PPGSE – da Universidade Tecnológica Federal do Paraná – UTFPR, às 9h30 do dia 04 de abril 2017. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores doutores:

---

Prof. Dr. Amauri Amorin Assef  
(Presidente – UTFPR)

---

Profa. Dra. Elisangela Ferretti Manffra  
(PUCPR)

---

Prof. Dr. Joaquim Miguel Maia  
(UTFPR)

Visto da coordenação:

---

Prof. Dr. Eduardo Félix Ribeiro Romaneli  
(Coordenador do PPGSE)

A Folha de Aprovação assinada encontra-se na Coordenação do Programa



*Dedico este trabalho à minha noiva Isadora  
e aos meus pais Carlos e Marileide.*





## AGRADECIMENTOS

Ao Prof. Dr. Amauri Amorin Assef, pela orientação esplêndida, amizade, sua dedicação e confiança em mim depositada para a realização deste trabalho.

À minha noiva Isadora, aos meus pais Marileide e Carlos, e aos meus irmãos Julia e Iago que sempre acreditaram no meu potencial e estiveram ao meu lado, ajudando-me em todos os momentos de dificuldades, emocional e financeira, e proporcionando-me chegar até a obtenção do título de Mestre. Mas principalmente por aceitarem minha ausência durante esta caminhada.

À instituição Universidade Tecnológica Federal do Paraná (UTFPR) pela permissão para utilização dos equipamentos do Laboratório de Ultrassom e ao Programa de Pós-Graduação em Sistemas de Energia (PPGSE) por ter me concedido a oportunidade de realizar este mestrado.

Todos os colegas do Laboratório de Ultrassom da UTFPR, em destaque a equipe de Iniciação Científica e suas realizações de tarefas de implementação e testes dos blocos de processamento.

À Prof<sup>a</sup> Marcia Becker do núcleo de línguas da UTFPR pela correção do *abstract*.

Aos amigos Cássio, Guilherme, João Paulo, Ramon e Rodolfo que sempre me receberam como hospede quando não morava mais em Curitiba.

A empresa Intel Corp. pelos *kits* de FPGA concedidos, assim como as licenças de *software* utilizadas para o desenvolvimento do modelo proposto.



## RESUMO

FERREIRA, Breno Mendes. Modelagem e implementação de um sistema de processamento digital de sinais baseado em FPGA para geração de imagens por ultrassom usando o Simulink. 2017. 115 f. Dissertação de Mestrado – Programa de Pós-Graduação em Sistemas de Energia, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

O ultrassom (US) é uma técnica bem consolidada que vem sendo amplamente utilizada para teste, caracterização e visualização de estruturas internas de materiais biológicos e não biológicos. Na Universidade Tecnológica Federal do Paraná, o grupo de pesquisa do US desenvolveu o sistema ULTRA-ORS que, apesar de adequado para pesquisa relacionada à excitação e recepção multicanal, possui tempo de computação muito elevado, devido a processamento em computador pessoal. Este trabalho apresenta a modelagem, implementação e validação de um sistema de processamento digital de sinais baseado em dispositivo FPGA (*Field-Programmable Gate Array*) de alto desempenho para reconstrução de imagens por US através da técnica *beamforming*. O *software* Simulink e a ferramenta DSP Builder foram empregados para simulação e transformação dos seguintes modelos em linguagem de descrição de *hardware*: filtro digital FIR (*Finite Impulse Response*), filtro de interpolação CIC (*Cascaded Integrator-Comb*), atraso variável, apodização, somatório coerente, decimação, demodulação com detecção de envoltória e compressão logarítmica. Após validação no Simulink, o projeto foi sintetizado para uma FPGA Stratix IV e implementado na placa Terasic DE4-230. A ferramenta SignalTap II do *software* Quartus II foi utilizada para aquisição dos sinais processados pela FPGA. Para avaliação gráfica e quantitativa da acurácia deste método, foram empregados dados brutos reais de US, adquiridos do ULTRA-ORS com frequência de amostragem de 40 MHz e resolução de 12 *bits*, e a função de custo da raiz quadrada do erro quadrático médio normalizado (NRMSE) em comparação com as mesmas funções implementadas através de *scripts* no Matlab. Como resultado principal do modelamento, além das respostas individuais de cada bloco implementado, são apresentadas as comparações entre as imagens reconstruídas pelo ULTRA-ORS e pelo processamento em FPGA para quatro janelas de apodização. A excelente concordância entre os resultados simulados e experimentais com valores de NRMSE inferiores à 6,2% e latência total de processamento de 0,83  $\mu$ s corroboram a simplicidade, modularidade e efetividade do modelamento proposto para utilização em pesquisas sobre o processamento de sinais de US para reconstrução de imagens em tempo real.

**Palavras-chave:** Reconstrução de Imagem por Ultrassom. FPGA. *Beamforming*. Sistema de Pesquisa.



## ABSTRACT

FERREIRA, Breno Mendes. Modeling and implementation of a FPGA-based digital signal processing for ultrasound imaging using Simulink. 2017. 115 f. Dissertação de Mestrado – Programa de Pós-Graduação em Sistemas de Energia, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

Ultrasound (US) is a well-established technique that has been widely used for testing, characterizing and visualizing internal structures of biological and non-biological material. The US research group of the Federal University of Technology - Paraná developed the ULTRA-ORS system, which, although suitable for research related to multichannel excitation and reception, uses a large computing time, due to the personal computer processing. This research presents the modeling, implementation and validation of a digital processing system of signals based on a FPGA (Field-Programmable Gate Array) device of high performance for the reconstruction of images through US, using the beamforming technique. The software Simulink and the tool DSP Builder were used for simulation and transformation of the following models in hardware description language: digital filter FIR (Finite Impulse Response), CIC (Cascaded Integrator-Comb) Interpolation filter, variable delay, apodization, coherent summation, decimation, demodulation with envelope detection and logarithmic compression. After the Simulink validation, the design was synthesized for a Stratix IV FPGA and implemented on the Terasic DE4-230 board. The tool SignalTap II in the software Quartus II was used to acquire the processed signals from the FPGA. For the graphic and quantitative evaluation of the accuracy of this method, we used real raw US data, acquired from the ULTRA-ORS with sampling frequency of 40 MHz and 12-bit resolution, and the normalized root mean squared error (NRMSE) in comparison with the same functions implemented through scripts in Matlab. As a main result of the modeling, in addition to the individual responses of each implemented block, comparisons between the reconstructed images by ULTRA-ORS and FPGA processing for four apodization windows are presented. The excellent agreement between the simulated and experimental results with NRMSE values lower than 6.2% and total processing latency of 0.83  $\mu$ s corroborates the simplicity, modularity and effectiveness of the proposed modeling for use in US signal processing research for real-time image reconstruction.

**Keywords:** Ultrasound Imaging Reconstruction. FPGA. Beamforming. Research System.



## LISTA DE FIGURAS

Figura 1 – Representação da geração de ondas de US pelo método pulso-eco para a determinação de parâmetros e caracterização de diferentes meios. ....	23
Figura 2 – Sequência de aberturas com oito elementos ativos de um transdutor matricial linear de 128 elementos como ilustração da técnica <i>beamforming</i> ....	25
Figura 3 – Perfis temporais de atraso e dos feixes acústicos com focalização simétrica em relação à abertura de oito elementos ativos. ....	25
Figura 4 – Exemplo de formação de imagem em Modo A e Modo B em diferentes meios. ....	26
Figura 5 – Diagrama em blocos das principais funções de processamento digital do <i>beamforming</i> de recepção na unidade <i>back-end</i> para geração de imagem por US Modo B. ....	28
Figura 6 – Estrutura convencional generalizada de um Filtro FIR.....	31
Figura 7 – Exemplo de resposta ao impulso para um filtro FIR com simetria ímpar de 11 coeficientes. ....	31
Figura 8 – Estrutura simplificada de um filtro FIR explorando a simetria ímpar de coeficientes. ....	32
Figura 9 – Estrutura simplificada de um filtro FIR explorando a simetria par de coeficientes. ....	32
Figura 10 – Componentes de um filtro CIC. (a) Estágio integrador. (b) Estágio <i>comb</i> . ....	34
Figura 11 – Estrutura simplificada e generalizada de um filtro de interpolação CIC. ....	34
Figura 12 – Focalização de recepção dinâmica de US através da abertura sintética pelo método <i>beamforming</i> .....	36
Figura 13 – Exemplos de esquemas de demodulação. (a) Demodulação analógica convencional. (b) Método de demodulação em quadratura por Transformada de Hilbert.....	39
Figura 14 – Ciclo metodológico do MDE com a aplicação de ferramentas específicas para transformação de modelos em nível de registradores sintetizáveis.....	42
Figura 15 – Fotografia da placa de desenvolvimento DE4-230 com a FPGA Stratix IV.....	46
Figura 16 – Fotografias da plataforma de bancada ULTRA-ORS. (a) Foto frontal. (b) Foto traseira com transdutor matricial convexo conectado.....	46
Figura 17 – <i>Phantom</i> mimetizador de tecidos biológicos utilizado para avaliação do trabalho. ....	48
Figura 18 – Ampliação da ROI no <i>phantom</i> de US com seus respectivos pontos alvos avaliados neste trabalho. ....	48
Figura 19 – Diagrama em blocos do sistema de modelagem, implementação e validação das etapas DSP para reconstrução de imagem por US.....	49
Figura 20 – Foto do PC com exemplos de telas de aquisição e processamento, e <i>kit</i> DE4-230.....	49





Figura 21 – Janela do ambiente Simulink com a biblioteca Altera DSP Builder Standard Blockset. ....	50
Figura 22 – Implementação dos dados de entrada com oito canais armazenados em memórias individuais. ....	53
Figura 23 – Detalhe do subsistema “Dados de Entrada”.....	54
Figura 24 – Sinais brutos de RF com abertura de oito elementos para formação da <i>scanline</i> 61. ....	55
Figura 25 – Resposta em frequência da magnitude do filtro FIR obtida com a ferramenta FDATool no Matlab. ....	57
Figura 26 – Resposta de fase do filtro FIR obtida com a ferramenta FDATool no Matlab. ....	57
Figura 27 – Resposta ao impulso do filtro FIR obtida com a ferramenta FDATool no Matlab. ....	57
Figura 28 – Diagrama em blocos com o filtro FIR de oito canais. ....	58
Figura 29 – Modelo do subsistema “Filtro FIR” com aproveitamento de simetria dos coeficientes para redução de multiplicadores.....	59
Figura 30 – Diagrama em blocos com a Interpolação de oito canais.....	60
Figura 31 – Modelo do subsistema “Interpolação” com fator $R = 4$ , implementado com quatro integradores e quatro diferenciadores.....	61
Figura 32 – Diagrama em blocos do Atraso Variável de oito canais com entrada dos ciclos de atraso.....	63
Figura 33 – Perfil temporal de focalização em 25 mm para abertura com 8 elementos do transdutor AT3C52B. ....	63
Figura 34 – Perfil de focalização com ciclos de <i>clock</i> de 40 MHz. ....	64
Figura 35 – Perfil de focalização com ciclos de <i>clock</i> de 160 MHz. ....	64
Figura 36 – Detalhe do subsistema “Delay_Block” em ciclos de <i>clock</i> .....	65
Figura 37 – Modelo do Subsistema “Atraso_Variável” implementado em memória com contador para gravação e atraso no contador para leitura da memória. ....	65
Figura 38 – Fragmento do subsistema “Atraso_Variavel_mux” para determina o atraso de <i>clock</i> . ....	66
Figura 39 – Implementação da Apodização em oito canais com respectivos coeficientes. ....	67
Figura 40 – Detalhe das chaves seletoras para escolha dos coeficientes de apodização.....	68
Figura 41 – Modelo do subsistema de seleção de apodização por multiplexadores ....	68
Figura 42 – Resultados simulados para as técnicas de apodização Retangular, Gaussiana, Hamming e Hanning para abertura de oito elementos ativos.....	69
Figura 43 – Modelo do subsistema “Apodizacao” implementado com multiplicador. ....	70
Figura 44 – Implementação do bloco “Somatório Coerente” de oito canais.....	71
Figura 45 – Detalhe interno do modelo do subsistema “Somatorio_Coerente” implementado.....	71



Figura 46 – Implementação do Bloco subsistema “Decimação”.....	72
Figura 47 – Modelo do bloco de “Decimacao” implementado com <i>flip-flop</i> .....	72
Figura 48 – Bloco subsistema “Demodulação”.....	73
Figura 49 – Modelo do subsistema “Demodulação” com os blocos “Transformada_Hilbert” e “Detecção_de_Envelope”.....	73
Figura 50 – Modelo do subsistema “Transformada_Hilbert” implementado utilizando simetria de coeficientes, bem como os coeficientes de valores nulos. .....	74
Figura 51 – Detalhe interno do modelo do subsistema “Detecção de Envelope” implementado.....	75
Figura 52 – Implementação do Bloco subsistema “Compressão Logarítmica”. 76	
Figura 53 – Tabela logarítmica com 2048 endereços (11 <i>bits</i> ).....	77
Figura 54 – Tabela logarítmica normalizada com 2048 endereços e faixa dinâmica de -30 dB. ....	77
Figura 55 – Modelo do subsistema “Compressão Logarítmica” implementado utilizando LUT e chave de seleção.....	78
Figura 56 – Validação do filtro FIR através da resposta ao impulso. ....	79
Figura 57 – Comparação entre o sinal filtrado pelo modelo “Filtro Digital FIR” e pelo bloco gerado pelo FDATool.....	80
Figura 58 – Comparação de filtragem do sinal senoidal com ruído. (a) Sinal de entrada com ruído. . (b) Sinal processado no Simulink. (c) Sinal processado na FPGA .....	81
Figura 59 – Comparação de filtragem do sinal de US real com ruído. (a) Sinal de entrada com ruído. (b) Sinal processado no Simulink. (c) Sinal processado na FPGA.....	81
Figura 60 – Espectros de frequência do sinal senoidal de teste adquiridos da placa DE4-230. (a) FFT do sinal com ruído. (b) FFT do sinal filtrado.....	82
Figura 61 – Espectros de frequência do sinal de US real adquiridos da placa DE4-230. (a) FFT do sinal com ruído. (b) FFT do sinal filtrado. ....	82
Figura 62 – Exemplo do resultado da interpolação. (a) Sinal entrada do bloco. (b) Sinal de saída gerado. ....	83
Figura 63 – Comparação entre o sinal original (40 MHz) e o sinal interpolado (160 MHz) para visualização do tempo de atraso de processamento.....	83
Figura 64 – Comparação dos espectros de frequência do sinal de US <i>scanline</i> 61. (a) FFT do sinal original ( $F_s = 40$ MHz). (b) FFT do sinal interpolado ( $F_s =$ 160 MHz).....	84
Figura 65 – Comparação dos sinais antes e após o bloco “Atraso Variável”. (a) Sinais de US antes e após o bloco “Atraso Variável”. (b) Contador de referência (sem atraso) e contador após atraso de 12 ciclos.....	85
Figura 66 – Comparação entre os desvios calculados para $F_s$ de 40 MHz e 160 MHz.....	85
Figura 67 – Exemplo de apodização Gaussiana referentes à abertura de oito elementos.....	86



Figura 68 – Exemplo de sinal obtido pelo processamento do bloco “Somatório Coerente”. (a) Resposta de oito linhas de RF (61 a 68) com apodização Gaussiana. (b) Linha de varredura resultante do somatório coerente.....	87
Figura 69 – Comparação de duas linhas de varredura geradas com diferentes apodizações. (a) Linhas de varredura geradas a partir de sinais processados com apodização Retangular e Gaussiana. (b) Ampliação do ítem (a) para melhor visualização. ....	88
Figura 70 – Comparativo entre os sinais de entrada e saída do bloco “Decimação”.....	89
Figura 71 – Comparação da resposta ao impulso da Transformada de Hilbert baseada em filtro FIR entre o FDATool e modelo implementado.....	90
Figura 72 – Exemplo de sinais envolvidos na detecção de envoltória gerados pelo modelo proposto.....	90
Figura 73 – Resultado da detecção de envoltória. (a) Linha de varredura e respectivo envelope do sinal. (b) Ampliação dos primeiros picos do ítem (a). .	91
Figura 74 – Validação do processamento de detecção de envoltória através da comparação dos resultados do modelo e do Matlab.....	91
Figura 75 – Validação do modelo de compressão logarítmica. (a) Envelope do sinal de entrada do bloco. (b) Sinais de saída limitados em -30 dB (vermelho) e -50 dB (azul).....	92
Figura 76 – Comparativo dos sinais processados pelo modelo e no Matlab sobrepostos com NRMSE igual a 0,45%. ....	93
Figura 77 – Comparação entre as imagens reconstruídas em Modo B com as apodizações (a) Retangular, (b) Gaussiana, (c) Hamming e (d) Hanning no Simulink, e (e) Retangular, (f) Gaussiana, (g) Hamming e (h) Hanning com dados computados pela FPGA.....	94
Figura 78 – Comparação entre as imagens reconstruídas em Modo B com as apodizações (a) Retangular, (b) Gaussiana, (c) Hamming e (d) Hanning na ULTRA_ORs, e (e) Retangular, (f) Gaussiana, (g) Hamming e (h) Hanning com dados computados pela FPGA.....	95



## LISTA DE TABELAS

Tabela 1 – Vantagens e desvantagens do Filtro FIR. ....	30
Tabela 2 – Especificação de requisitos do modelo de processamento. ....	41
Tabela 3 – Parâmetros de configuração do bloco “Device”.....	51
Tabela 4 – Parâmetros de configuração do bloco “Signal Compiler”. ....	51
Tabela 5 – Parâmetros de configuração do bloco “Clock”.....	52
Tabela 6 – Parâmetros de configuração do bloco “PLL” .....	52
Tabela 7 – Especificações do subsistema “Dados de Entrada”. ....	54
Tabela 8 – Especificação do filtro digital FIR.....	56
Tabela 9 – Coeficientes simétricos de filtragens calculados no FDATool. ....	56
Tabela 10 – Especificações de configuração dos componentes que compõem o filtro FIR.....	60
Tabela 11 – Especificações de configuração dos componentes que compõem a interpolação.....	62
Tabela 12 – Especificações de configuração dos componentes do subsistema “Delay_Block”.....	65
Tabela 13 – Especificações de configuração dos componentes da etapa “Atraso_Variável”.....	66
Tabela 14 – Especificações de configuração dos componentes do subsistema “Atraso_Variavel_mux”.....	67
Tabela 15 – Posição das chaves seletoras para seleção dos coeficientes para as técnicas de apodização Retangular, Gaussiana, Hamming e Hanning com abertura de oito elementos.....	69
Tabela 16 – Especificações de configuração dos componentes do subsistema “Apodizacao”.....	70
Tabela 17 – Especificações de configuração dos componentes da etapa de “Decimacao”.....	72
Tabela 18 – Especificações de configuração dos componentes da etapa de “Detecção de Envelope”.....	75
Tabela 19 – Especificações de configuração dos componentes da etapa de “Compressão Logarítmica”.....	78
Tabela 20 – Recursos de <i>hardware</i> da FPGA Stratix IV consumidos. ....	96





## LISTA DE ABREVIATURAS E SIGLAS

2-D	Bidimensional
A	Amplitude
ADC	<i>Analog-to-Digital Converter</i>
ASIC	<i>Application-Specific Integrated Circuit</i>
B	Brilho
CI	Circuito Integrado
CPGEI	Pós-Graduação em Engenharia Elétrica e Informática Industrial
DAS	<i>Delay and Sum</i>
dB	Decibel
DSP	<i>Digital Signal Processor</i>
E/S	Entrada/Saída
FFA	<i>Fast FIR Algorithm</i>
FFT	<i>Fast Fourier Transform</i>
FIR	<i>Finite Impulse Response</i>
FPB	Filtro Passa-Baixa
FPGA	<i>Field-Programmable Gate Array</i>
HDL	<i>Hardware Description Language</i>
IIR	<i>Infinite Impulse Response</i>
LCD	<i>Liquid Crystal Display</i>
LE	<i>Logical Element</i>
LED	<i>Light-Emitting Diode</i>
LUS	Laboratório de Ultrassom
LUT	<i>Look-Up Table</i>
M	Movimento
Mbps	Mbit por segundo
MCU	Microcontrolador
MDE	<i>Model Driven Engineering</i>
NDE	<i>Non-Destructive Evaluation</i>
NRMSE	<i>Normalized Root-Mean-Square Error</i>
PC	<i>Personal Computer</i>
PLL	<i>Phase-Locked Loop</i>
PPGSE	Programa de Pós-Graduação em Sistemas de Energia



RAM	<i>Random Access Memory</i>
RASMUS	Remotely Accessible Software Configurable Multi-channel Ultrasound Sampling
RF	Radiofrequência
ROI	<i>Region of Interest</i>
RP	<i>Research Package</i>
RTL	<i>Register-Transfer Level</i>
S/H	<i>Sample and Hold</i>
SD	<i>Secure Device</i>
SDRAM	<i>Synchronous Dynamic Random Access Memory</i>
SNR	<i>Signal-to-Noise Ratio</i>
SRAM	<i>Static Random Access Memory</i>
SRC	<i>Sampling Rate Conversion</i>
TGC	<i>Time Gain Compensation</i>
ULA-OP	ULtrasound Advanced Open Platform
ULTRA-ORS	Ultrasound Open Research System
US	Ultrassom
UTFPR	Universidade Tecnológica Federal do Paraná
VHDL	<i>VHSIC Hardware Description Language</i>
VHSIC	<i>Very-High-Speed Integrated Circuit</i>
Vpp	Tensão pico-a-pico



## LISTA DE SÍMBOLOS

$\bar{M}$	Média das amostras do modelo
$A_r$	Onda refletida
$b_k$	Coefficientes de filtragem
$c$	Velocidade de propagação da onda
$E$	Envelope
$F_s$	Frequência de amostragem
$H_c$	Função da seção <i>comb</i>
$H_i$	Função da seção integradora
$i$	Índice da amostra
$l$	<i>In-phase</i>
$k$	Índice do coeficiente de filtragem
$M$	Ordem do filtro FIR
$M_i$	Amostra do modelo
$N$	Número de amostra
$n_a$	Número de elementos da abertura
$n_e$	Número total de elementos do transdutor
$n_{sc}$	Número de linhas de varredura do sistema
$Q$	<i>Quadrature</i>
$R$	<i>Ratio</i>
$r_d$	Sinal recebido
$r_f$	Sinal após <i>beamforming</i>
$S_i$	Amostra de referência
$t$	Tempo
$w_m$	Peso da janela de ponderação
$x(n)$	Sinal de entrada do filtro
$x_{dB}$	Resultado do processamento bloco de compressão logarítmica
$x_m$	Sinal de entrada
$x_{max}$	Valor máximo da tabela logarítmica implementada
$y(n)$	Sinal de saída processado pelo filtro
$z$	Distância
$z[m]$	Saída do formador de feixe
$z^{-1}$	Fator de atraso



$\Delta_m$

Retardo (*delay*) apropiado

$\tau$

Atraso temporal





## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>18</b>
1.1	MOTIVAÇÃO	20
1.2	OBJETIVOS	21
1.2.1	Objetivo geral	21
1.2.2	Objetivos específicos	21
1.3	ESTRUTURA DO TRABALHO	22
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>23</b>
2.1	MÉTODO PULSO-ECO	23
2.2	<i>BEAMFORMING</i>	24
2.3	FORMAÇÃO DE IMAGEM EM MODO B	26
2.4	FPGA	27
2.5	PROCESSAMENTO DE SINAL PELA TÉCNICA <i>BEAMFORMING</i>	28
2.5.1	Filtros digitais	29
2.5.1.1	Filtros de resposta ao impulso finita FIR	30
2.5.2	Conversão de taxa de amostragem	33
2.5.2.1	Filtro de interpolação	33
2.5.2.2	Decimação	35
2.5.3	Atraso variável	35
2.5.4	Apodização	36
2.5.5	Somatório coerente	37
2.5.6	Demodulação	38
2.5.6.1	Transformada de Hilbert e detecção de envoltória	38
2.5.7	Compressão logarítmica	39
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>41</b>
3.1	ESPECIFICAÇÃO DE REQUISITOS	41
3.2	ENGENHARIA DIRIGIDA A MODELOS	42
3.2.1	Simulink	43
3.2.2	DSP Builder	43
3.3	FUNÇÃO DE CUSTO NRMSE	44
3.4	FERRAMENTAS DE PROGRAMAÇÃO E VALIDAÇÃO	44
3.4.1	<i>Software</i> Quartus II	44
3.4.1.1	Analisador Lógico SignalTap II	45
3.5	<i>KIT</i> COMERCIAL DE DESENVOLVIMENTO DE4-230	45
3.6	PLATAFORMA DE PESQUISA ULTRA-ORS E <i>PHANTOM</i> DE US PARA TESTES	46
3.7	DIAGRAMA EM BLOCOS DO SISTEMA PROPOSTO	48
3.8	IMPLEMENTAÇÃO DO MODELO PROPOSTO	49



3.8.1	Configurações iniciais.....	50
3.8.1.1	Bloco Device.....	51
3.8.1.2	Bloco Signal Compiler.....	51
3.8.1.3	Bloco Clock.....	52
3.8.1.4	Bloco PLL.....	52
3.8.2	Bloco Sinal RF.....	53
3.8.3	Bloco Filtro Digital FIR.....	55
3.8.4	Bloco Interpolação <i>Cascaded Integrator-Comb</i> .....	60
3.8.5	Bloco Atraso Variável.....	62
3.8.6	Bloco Apodização.....	67
3.8.6.1	Subsistema Apodização referente a cada canal.....	69
3.8.7	Bloco Somatório Coerente.....	70
3.8.8	Bloco Decimação.....	71
3.8.9	Bloco Demodulação.....	72
3.8.10	Bloco Compressão Logarítmica.....	75
<b>4</b>	<b>RESULTADOS.....</b>	<b>79</b>
4.1	RESULTADO INDIVIDUAL DAS ETAPAS DE PROCESSAMENTO.....	79
4.1.1	Implementação do filtro digital FIR.....	79
4.1.2	Interpolação CIC.....	82
4.1.3	Atraso variável.....	84
4.1.4	Apodização.....	86
4.1.5	Somatório.....	86
4.1.6	Decimação.....	88
4.1.7	Demodulação.....	89
4.1.8	Compressão logarítmica.....	92
4.2	RESULTADO FINAL DO PROCESSAMENTO PARA RECONSTRUÇÃO DE IMAGEM.....	93
4.2.1	Comparativo entre os processamentos simulado e experimental ..	93
4.2.2	Comparativo entre os processamentos da plataforma de pesquisa ULTRA-ORS e do sistema implementado em FPGA.....	93
4.2.3	Utilização de recursos de <i>hardware</i> da FPGA.....	96
4.2.4	Latência total de processamento.....	96
<b>5</b>	<b>DISCUSSÃO E CONCLUSÕES.....</b>	<b>97</b>
5.1	ANÁLISE DOS RESULTADOS.....	97
5.2	CONCLUSÕES.....	99
5.3	TRABALHOS FUTUROS.....	100
	<b>REFERÊNCIAS.....</b>	<b>102</b>
	<b>APÊNDICE A – PROJETO COMPLETO DO SISTEMA DSP BASEADO EM FPGA PARA A GERAÇÃO DE IMAGEM POR US NO SIMULINK.....</b>	<b>113</b>
	<b>PUBLICAÇÕES REFERENTES AO TRABALHO.....</b>	<b>115</b>



## 1 INTRODUÇÃO

O ultrassom (US) é uma técnica bem consolidada que vem sendo amplamente empregada desde a década de 50 para testes, caracterização e visualização de estruturas internas de materiais biológicos e não biológicos (WILD, 1950; HEDRICK; HYKES; STARCHMAN, 1995). Com técnicas de formação de imagens ultrassônicas semelhantes ao sonar e radar<sup>1</sup>, o método é baseado na emissão de ondas acústicas com frequência acima de 20 kHz que interagem com o meio investigado e recepção dos ecos refletidos e espalhados, devido à não homogeneidade e presença de microestruturas com diferentes propriedades acústicas (FISH, 1990; SHUNG, 2006).

Sabendo-se que cada material apresenta propriedades acústicas singulares como impedância, velocidade de propagação do som e atenuação da onda, é possível interpretar quantitativamente os ecos de radiofrequência (RF) refletidos que transportam informações sobre a posição e densidade dos objetos refletidos. Considerando aplicações biomédicas e dependendo da informação requerida é possível, por exemplo, prover imagens e informações das estruturas anatômicas internas ou atenuação das ondas, além do estado funcional de determinados órgãos (CHRISTENSEN, 1988; HASSAN; YOUSSEF; KADAH, 2011).

Diversas características do US contribuem para tornar a técnica uma importante ferramenta de auxílio ao diagnóstico médico por imagem, destacando-se a facilidade e segurança associados à sua utilização (CHRISTENSEN, 1988), características não invasiva e não ionizante, portabilidade, versatilidade, e ter custo relativamente baixo comparado às outras modalidades de geração de imagens para diagnóstico médico (ACHIM; ANASTASIOS; TSAKALIDES, 2001; KIM et al., 2012). Além disso, pode-se destacar a capacidade de geração de imagens em tempo real e em qualquer orientação espacial do alvo analisado, características fundamentais em análises dinâmicas (CÂMARA, 2015).

---

<sup>1</sup> A imagem ultrassônica obtida por sonar e radar são da forma pulso-eco. São transmitidos impulsos ultrassônicos, no meio a ser investigado, e calcula-se o tempo de retorno dos ecos para formar uma imagem representativa das distâncias dos obstáculos (SHUNG, 2006).

Por motivos semelhantes, as indústrias automotiva, marítima, aeroespacial, nuclear, e de petróleo e gás, dentre outras, também utilizam o US pulso-eco em aplicações de manutenção preventiva, através da técnica de avaliação não destrutiva NDE (*Non-Destructive Evaluation*) (KELLY; FARLOW; HAYWARD, 1996). Nesses casos, o ensaio por US atende, de forma adequada, às exigências industriais, sendo considerado mais rápido, preciso e seguro na inspeção, tanto na superfície (JHANG, 2009) quanto na busca por descontinuidades internas em estruturas ou componentes, em comparação com outros métodos convencionais, os quais utilizam pressão de ar ou água para detecção de vazamento de líquidos ou gases (KRAUTKRÄMER; KRAUTKRÄMER, 1990; KELLY; FARLOW; HAYWARD, 1996; MEO; POLIMENO; ZUMPANO, 2008; JHANG, 2009).

Também na inspeção de sistema elétricos, o US é utilizado para identificação de falhas operacionais através do nível de ruído gerado, principalmente, por três problemas básicos: (1) arco elétrico, (2) efeito corona<sup>2</sup> e (3) descargas elétricas embrionárias (antes de se tornarem um arco propriamente dito). Como vantagem, a inspeção por US não necessita de desligamento e abertura de painéis energizados, sendo aplicada para detecção de correntes de fuga oriundas de falhas em isoladores, transformadores, barramentos, cabos, relés, disjuntores, motores, geradores, dentre outros (MACHADO et al., 2011).

Atualmente, os modernos equipamentos para geração de imagem por US em tempo real, também nomeados como *scanners* digitais, exigem alto poder de processamento de sinais e capacidade de transferência massiva de dados (TANTER; FINK, 2014). Durante o processo de geração do feixe acústico de recepção, técnica conhecida como *beamforming* digital de recepção, devido à alta demanda computacional, são tradicionalmente empregadas arquiteturas de *hardware* "fechadas", baseadas em dispositivos ASICs (*Application-Specific Integrated Circuits*) e, mais recentemente, arquiteturas reconfiguráveis com dispositivos FPGAs (*Field-Programmable Gate Array*), além de processadores digitais de sinais (DSPs – *Digital Signal Processors*) com vários núcleos

---

<sup>2</sup> Ocorre quando a tensão em um condutor elétrico excede o gradiente de potencial do ar que circunda este condutor e começa a ionizá-lo e formar uma nuvem azul ou púrpura ao redor.

(AGARWAL et al., 2007; SCHNEIDER et al., 2010; BONI et al., 2012; ASSEF, 2013).

Apesar da disponibilidade de equipamentos comerciais de US com acesso aos sinais de RF, como o Ultrasonix 500RP (*Research Package*) (WILSON et al., 2006) e SonixMDP (ULTRASONIX MEDICAL CORP., CANADÁ), além do Verasonics Vantage Research US System (VERASONICS INC., USA), diversos centros de pesquisa têm desenvolvido novos sistemas e métodos que permitem maior flexibilidade e controle sobre os módulos de transmissão, recepção e processamento (MURPHY; WAKEFIELD; FRIEDMAN, 2008). Como exemplo, pode-se citar os sistemas de pesquisa ULA-OP (Ultrasound Advanced Open Platform) (BONI et al., 2012) e o RASMUS (Remotely Accessible Software configurable Multi-channel Ultrasound Sampling) (JENSEN et al., 2005). Tal capacidade é essencial para promover melhorias na qualidade das informações obtidas para as diversas modalidades de US existentes, bem como contribuir, significativamente, para o desenvolvimento de técnicas inovadoras de formação de imagem por US.

## 1.1 MOTIVAÇÃO

A dinâmica da evolução tecnológica dos materiais e componentes utilizados na fabricação dos produtos para diagnóstico médico e ensaios NDE, aliada à defasagem tecnológica verificada no País, e conseqüentemente, a enorme dependência de importações, configura uma situação de vulnerabilidade para a sociedade brasileira. Apesar de existirem diversos fabricantes de equipamentos comerciais de geração de imagem por US, todos eles são multinacionais, o que demanda um elevado custo, afetando diretamente o déficit na balança comercial (MINISTÉRIO DA SAÚDE BRASIL, 2008).

Nesse sentido, o grupo de pesquisa do US da Universidade Tecnológica Federal do Paraná (UTFPR) vem desempenhando um importante papel para o desenvolvimento científico-tecnológico e inovação de equipamentos e materiais no Brasil. Uma das recentes contribuições do grupo foi o desenvolvimento da plataforma de pesquisa ULTRA-ORS (do inglês Ultrasound Open Research System) (ASSEF, 2013). Constituída de uma arquitetura modular de *hardware*

(*front-end*), baseada em dispositivos FPGA, e *software (back-end)*, baseada em computador pessoal (PC - *Personal Computer*), o ULTRA-ORS é um sistema totalmente programável, aberto e flexível com 128 canais de transmissão e recepção, com resolução de 12 *bits* e frequência de amostragem de até 50 MHz, para pesquisa e investigação de novas técnicas de geração de imagens por US. Entretanto, o tempo para reconstrução de imagens é muito elevado, podendo chegar a dez minutos, uma vez que o processamento é realizado inteiramente em PC, a partir de algoritmos de rotinas (*scripts*) com instruções sequenciais elaboradas com o *software* MATLAB, o que inviabiliza a análise das informações obtidas em tempo real.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

O objetivo deste trabalho foi modelar um sistema de processamento digital de sinais embarcado em dispositivo FPGA de alto desempenho visando a reconstrução de imagens por US em tempo real, através da técnica *beamforming*, para processamento de sinais brutos de RF gerados pela plataforma de *hardware* ULTRA-ORS, de forma a equipar o Laboratório de Ultrassom (LUS) dos Programas de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI) e de Sistemas de Energia (PPGSE) da UTFPR, com instrumentação adequada para o estudo da técnica.

### 1.2.2 Objetivos específicos

Além do objetivo principal, este trabalho teve como objetivos específicos:

- 1) Investigar as principais técnicas de processamento de sinais para geração de imagens por US, a partir do método *beamforming*;
- 2) Utilizar a técnica de desenvolvimento de *software* Engenharia Dirigida a Modelos para modelagem e integração das etapas DSP;



- 3) Modelar as etapas DSP para reconstrução de imagem por US utilizando o *software* Matlab/Simulink (MATHWORKS, EUA) e a ferramenta DSP Builder (INTEL CORP, EUA);
- 4) Empregar a ferramenta Quartus II (INTEL CORP, EUA) para compilação dos códigos gerados em linguagem de descrição de *hardware*, gravação da FPGA e aquisição do resultado do processamento;
- 5) Adotar um *kit* comercial de FPGA para o desenvolvimento do trabalho;
- 6) Utilizar dados brutos reais adquiridos pela plataforma ULTRA-ORS com frequência de amostragem de 40 MHz e resolução de 12 *bits*;
- 7) Validar os blocos de processamento através de simulações no Simulink;
- 8) Utilizar a função de custo da raiz quadrada do erro quadrático médio normalizado NRMSE (*Normalized Root Mean Squared Error*) para avaliação quantitativa dos principais blocos do modelo em comparação com as funções implementadas no Matlab;
- 9) Comparar o resultado entre a simulação do modelo e o obtido pelo processamento embarcado em FPGA;
- 10) Validar o resultado experimental do modelo embarcado em FPGA através da comparação com o resultado obtido pelo sistema ULTRA-ORS.

### 1.3 ESTRUTURA DO TRABALHO

Essa dissertação está organizada em cinco capítulos. No capítulo 2, faz-se uma revisão da literatura no qual são apresentados os principais conceitos sobre os princípios de geração de imagem por US pulso-eco necessários para o desenvolvimento da pesquisa. O capítulo 3 descreve os materiais e métodos para o desenvolvimento dos modelos DSP para reconstrução de imagens por US em Modo B. No capítulo 4 são apresentados os resultados individuais de cada etapa de processamento implementada e o resultado final da reconstrução de imagem. O capítulo 5 apresenta as discussões, conclusão e relação de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão descritos os principais conceitos para melhor compreensão sobre a instrumentação ultrassônica necessária para a geração de imagens por US Modo B utilizando a técnica *beamforming* com atraso de soma.

### 2.1 MÉTODO PULSO-ECO

No método pulso-eco, o mesmo transdutor é utilizado durante a transmissão e recepção de ondas de US (CHRISTENSEN, 1988). Após a aplicação de um pulso estreito de alta tensão o transdutor converte o sinal elétrico em ondas mecânicas que se propagam no meio e trazem informações sobre a profundidade e densidade de objetos refletoras com diferentes impedâncias acústicas (HEDRICK; HYKES; STARCHMAN, 1995). Na Figura 1 é ilustrado esse princípio para um elemento transdutor, resultando em uma imagem Modo A (Amplitude).

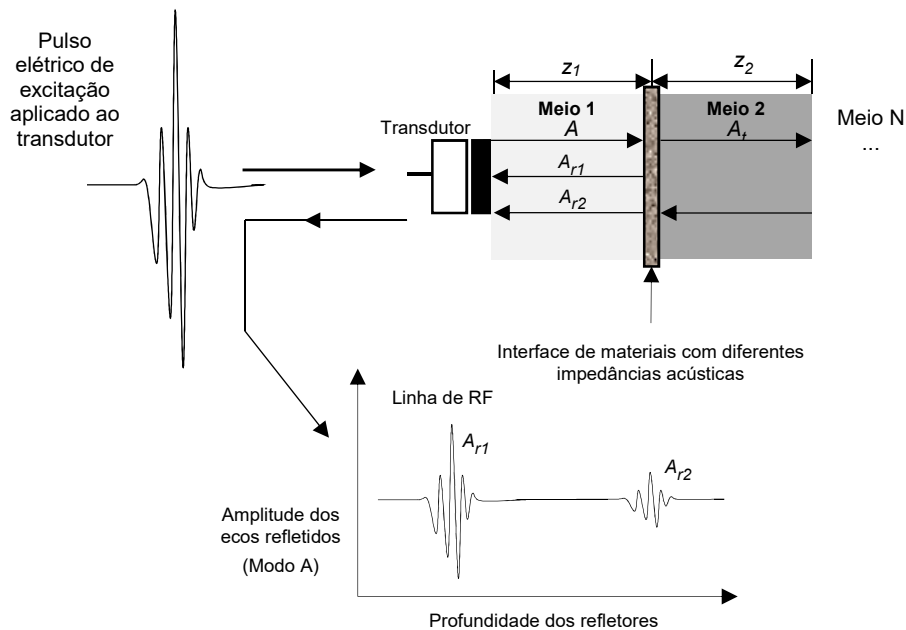


Figura 1 – Representação da geração de ondas de US pelo método pulso-eco para a determinação de parâmetros e caracterização de diferentes meios.

Fonte: Adaptado de Christensen (1988).

No exemplo, são considerados os Meios 1 e 2 adjacentes, com diferentes propriedades acústicas e com espessuras  $z_1$  e  $z_2$ , respectivamente. A onda de US se propaga no Meio 1 e, assim que incide na interface entre os dois meios, parte da onda é refletida ( $Ar_1$ ) e outra propaga-se no Meio 2 ( $At$ ). De forma similar, o sinal  $At$  apresenta o mesmo comportamento, originando o sinal  $Ar_2$ . As ondas refletidas  $Ar_1$  e  $Ar_2$  trazem informações sobre as características do meio. Através da Equação 1, pode-se calcular a distância ( $z$ ) ou a velocidade de propagação da onda ( $c$ ) no meio, a partir de um tempo ( $t$ ) decorrido entre a aplicação do pulso de excitação e a recepção do eco (CHRISTENSEN, 1988).

$$c = \frac{2 \cdot z}{t} \quad (1)$$

## 2.2 BEAMFORMING

Para geração de imagens por US são utilizados modernos transdutores matriciais multielementos com características piezoelétricas, ou seja, convertem energia elétrica em vibrações mecânicas que se propagam no meio investigado (AGARWAL; BANERJEE; DE, 2016). No método *beamforming*, exemplificado na Figura 2, um conjunto de elementos é excitado com tempos de atraso programáveis para focalizar as ondas ultrassônicas em uma determinada região de interesse (ROI - *Region of Interest*). Após a recepção dos ecos gerados, devido à reflexão e espalhamento nas interfaces com diferentes impedâncias acústicas, pelos mesmos elementos transdutores, é realizado o processo de correção temporal dos atrasos e posterior somatório coerente, formando uma linha de varredura (*scanline*) (HASSAN; YOUSSEF; KADAH, 2011). No exemplo, foram utilizados oito elementos transdutores para compor a abertura de excitação e recepção, formando uma abertura sintética.

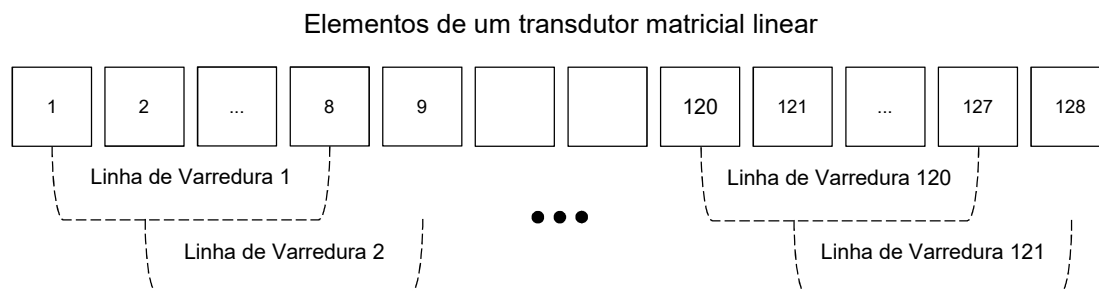


Figura 2 – Sequência de aberturas com oito elementos ativos de um transdutor matricial linear de 128 elementos como ilustração da técnica *beamforming*.

Assim, essa abertura sintética é deslocada em um elemento e a sequência é repetida, até o último elemento da matriz para uma abertura ativa. O mesmo perfil de atraso de focalização das ondas utilizado na transmissão ultrassônica também deve ser aplicado na recepção, antes do somatório coerente para composição de uma linha de varredura (DUBE, 2004; HASSAN; KADAH, 2013). Tal procedimento é ilustrado na Figura 3 para a mesma abertura de oito elementos.

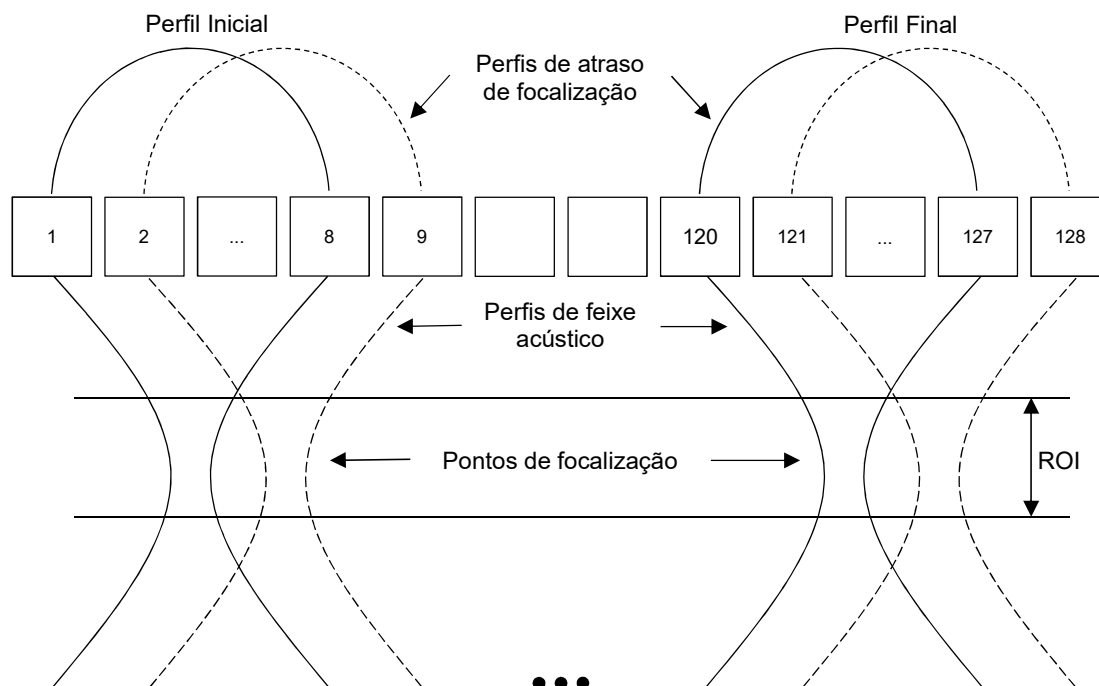


Figura 3 – Perfis temporais de atraso e dos feixes acústicos com focalização simétrica em relação à abertura de oito elementos ativos.

## 2.3 FORMAÇÃO DE IMAGEM EM MODO B

No Modo B, o brilho e cor atribuídos a cada elemento de imagem (*pixel*) estão relacionados com a amplitude do sinal recebido pelo transdutor – tipicamente é utilizada a escala de cinza de 8 *bits* (SHUNG, 2006). A posição espacial do ponto é determinada pela posição dos elementos e pelo tempo necessário para a onda ultrassônica percorrer duas vezes a distância entre o transdutor e a ROI (SHATTUCK et al., 1984; HEDRICK; HYKES; STARCHMAN, 1995).

Como exemplo, é apresentada na Figura 4 a envoltória dos ecos recebidos das interfaces entre ambos os Modos A e B, a partir da excitação de um transdutor monoelemento posicionado a uma distância ( $d$ ) da pele de um corpo biológico.

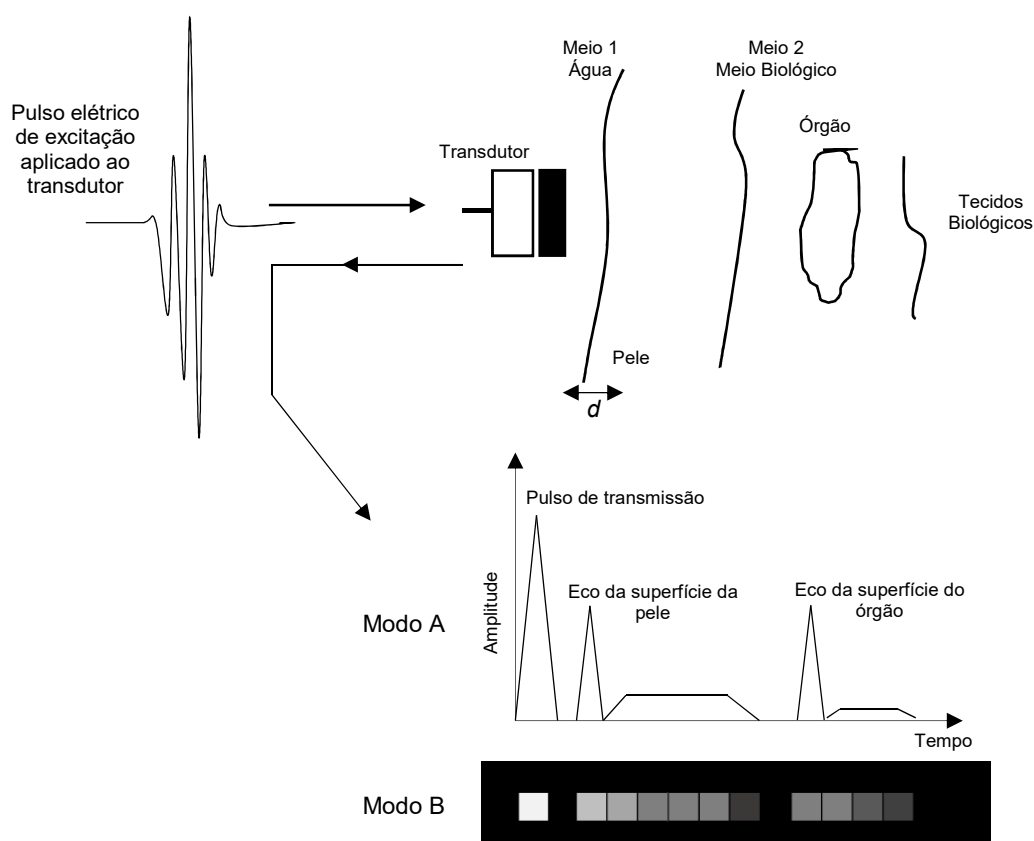


Figura 4 – Exemplo de formação de imagem em Modo A e Modo B em diferentes meios.

Fonte: Adaptado de Shung (2006).

A diferença entre a amplitude dos sinais ocorre devido a fatores geométricos (dimensões da fonte sonora, comprimento de onda, presença de superfícies refletoras, etc.), mecanismos de absorção, onde parte da energia é convertida em calor, e profundidade da ROI, sendo que a última pode ser compensada por uma curva de amplificação característica da atenuação do meio (HEDRICK; HYKES; STARCHMAN, 1995; WELLS, 1999; OTAKE et al., 2003; SHUNG, 2006).

## 2.4 FPGA

A FPGA é um dispositivo semicondutor constituído de unidades lógicas digitais (na ordem de milhares), encapsuladas em um único circuito integrado (CI), cujas interconexões podem ser programadas/reprogramadas pelo usuário de forma modular, ou seja, cada bloco de *hardware* é responsável por uma função (RODRIGUEZ-ANDINA; MOURE; VALDES, 2007; MONMASSON et al., 2011). Através do processo denominado roteamento, pode-se reproduzir desde lógicas simples até sistemas combinacionais mais complexos com registradores e multiplicadores para funções DSP, o que acrescenta flexibilidade ao projeto (SANTOS, 2008).

O funcionamento de uma FPGA depende de como é organizada a sua arquitetura. Estes dispositivos podem operar em ciclos de *clock*, onde cada operação matemática é realizada em um ciclo (HUDA; MALLICK; ANDERSON, 2009). Também possuem, como característica principal, capacidade de processamento em paralelo. No interior de cada bloco lógico da FPGA existem vários modos possíveis para a implementação de funções lógicas e matemáticas (KANG et al., 2016). O mais utilizado pelos fabricantes é o bloco de memória LUT (*Look-Up Table*). Esse tipo de bloco lógico contém células de armazenamento de memória voláteis RAM (*Random Access Memory*) que são utilizadas para implementar funções lógicas de forma otimizada (DA COSTA, 2014).

As linguagens utilizadas para a programação dos dispositivos FPGA são chamadas de linguagem de descrição de *hardware* (HDL - *Hardware Description Language*). Tais linguagens são utilizadas para descrever o funcionamento de

circuitos digitais, incluindo, por exemplo, multiplexadores, demultiplexadores, contadores, comparadores, registradores, multiplicadores, memórias, etc. Dentre as principais HDLs, destacam-se VHDL (*Very-High-Speed Integrated Circuit* HDL) e Verilog (BERGÉ et al., 1992).

## 2.5 PROCESSAMENTO DE SINAL PELA TÉCNICA *BEAMFORMING*

Todas as etapas DSP envolvidas na reconstrução de imagem por US Modo B, implementadas neste trabalho, são apresentadas de forma simplificada no diagrama em blocos da Figura 5. No diagrama, os blocos circunscritos pela linha tracejada correspondem ao módulo *back-end* (OTAKE et al., 2003; ASSEF, 2013), sendo descritos a seguir.

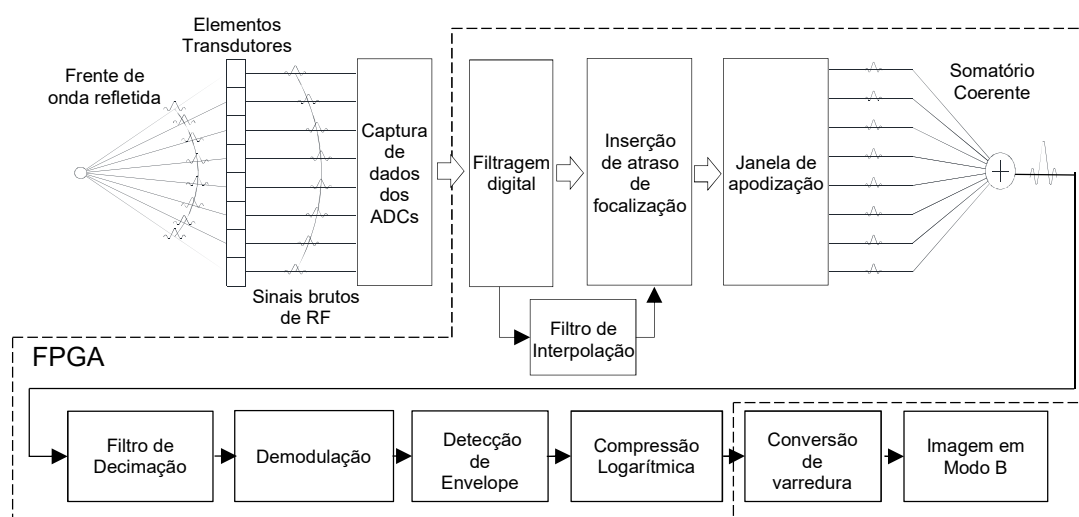


Figura 5 – Diagrama em blocos das principais funções de processamento digital do *beamforming* de recepção na unidade *back-end* para geração de imagem por US Modo B.

Fonte: Adaptado de Thomenius (2005) e Assef (2013).

Conforme descrito anteriormente, durante o processo para composição de imagem por US, gera-se um conjunto de pulsos elétricos de alta tensão, com atrasos temporais programados, para excitação dos elementos transdutores piezoelétricos (matriz piezoelétrica) que convertem os pulsos em ondas de pressão ultrassônica. Os ecos brutos de RF são recebidos e transformados em sinais elétricos pela mesma matriz piezoelétrica (BROWN; LOCKWOOD, 2005;

ASSEF, 2013). Após a aquisição e pré-amplificação dos ecos, circuitos conversores Analógico/Digital (ADCs - *Analog-to-Digital Converters*) realizam a digitalização para posterior processamento de sinais em função do modo de imagem a ser apresentada (ASSEF, 2013).

Considerando o Modo B, são realizadas, inicialmente, a filtragem dos sinais para atenuação dos ruídos fora da banda desejada, interpolação, controle apropriado dos tempos de atraso para correção de focalização na recepção, apodização para redução de lóbulos de espaçamento que geram falsos artefatos na imagem e, finalmente, o somatório coerente dos sinais para compor uma linha de varredura para melhorar a relação sinal-ruído (SNR - *Signal-to-Noise Ratio*). Esse fluxo de processamento é denominado de *Delay and Sum* (DAS) (MATRONE et al., 2015). Na sequência, são realizadas as etapas decimação para diminuir o número de amostras, filtragem digital de demodulação e detecção de envoltória, seguida pela compressão logarítmica, utilizada para adequar a faixa dinâmica dos dados aos níveis adequados para visualização bidimensional (2-D) em escala de cinza (SHUNG, 2006; HASSAN; KADAH, 2013). A última etapa se refere à conversão de varredura (*scan conversion*) para transformação de coordenadas polares para retangulares, antes da apresentação no monitor de imagem (OTAKE et al., 2003).

### 2.5.1 Filtros digitais

Filtros digitais são sistemas amplamente utilizados em DSP para modificar ou retirar alguma característica do sinal – mudar a amplitude relativa dos componentes em frequência de um sinal ou eliminar alguns componentes em frequência (OPPENHEIM; WILLSKY; NAWAB, 1997). A classificação dos filtros em relação à sua resposta ao impulso, divide-se em filtros FIR (*Finite Impulse Response*), ou de resposta ao impulso finita, e IIR (*Infinite Impulse Response*), ou de resposta ao impulso infinita (RABINER; SCHAFER, 1978; MADISSETTI; WILLIAMS, 1998).

A implementação do filtro FIR é geralmente mais aplicada em DSP, em relação ao IIR, por não apresentar deslocamento de fase através da banda de



frequência, possuir inerente estabilidade e não necessitar realimentação (RABINER et al., 1974; BRANDENSTEIN; UNBEHAUEN, 1998).

### 2.5.1.1 Filtros de resposta ao impulso finita FIR

Na década de 70, Crochiere e Rabiner (1975) apresentaram a teoria geral para os filtros decimadores e interpoladores de vários estágios para redução e aumento da taxa de amostragem, respectivamente. Dentre as contribuições principais do método, pode-se destacar a minimização do número de multiplicações por segundo. Após a padronização do método, diversas pesquisas, fundamentadas em decimação e interpolação, mostraram as propriedades atrativas da implementação em DSP dos filtros digitais de fase linear FIR de banda estreita que podem ser obtidas por cascata. Destaca-se ausência de problemas de estabilidade e a disponibilidade de projetos de métodos iterativos eficientes (MCCLELLAN; PARKS, 1973; CROCHIERE; RABINER, 1975, 1976; HOGENAUER, 1981), entre outras vantagens indicadas na Tabela 1.

Tabela 1 – Vantagens e desvantagens do Filtro FIR.

<b>Vantagens</b>	<b>Desvantagens</b>
São intrinsecamente estáveis Baixa complexidade computacional Possibilidade de terem fase linear	Em geral precisam de ordem alta para satisfazer às exigências do sistema

Fonte: Adaptado de Rabiner e Schafer (1978).

A estrutura de um filtro FIR é expressa matematicamente em termos de uma soma de convolução, como indicado na Equação 2:

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

onde  $M$  define a ordem do filtro FIR,  $n$  é o índice da amostra,  $k$  é o índice do coeficiente e  $b_k$  são os coeficientes do filtro (RABINER; SCHAFER, 1978).

A representação da estrutura convencional de um filtro FIR, apresentada na Figura 6, é implementada a partir da interpretação da Equação 2 do filtro discreto, onde  $x(n)$  representa o sinal de entrada a ser filtrado,  $z^{-1}$  o fator de atraso e  $y(n)$  o sinal de saída processado (JI-FENG et al., 2015).

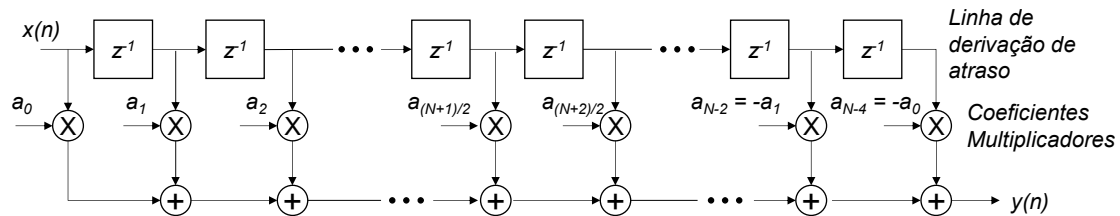


Figura 6 – Estrutura convencional generalizada de um Filtro FIR.

Fonte: Adaptado de Altera Corp. (2011) e Xilinx (2011).

Uma importante característica dos filtros FIR é a simetria da resposta ao impulso que pode ser explorada para diminuir significativamente os requisitos de *hardware* da função – multiplicadores, registradores de deslocamento e somadores. Como exemplo, a Figura 7 mostra a resposta ao impulso para um filtro FIR com simetria ímpar de 11 coeficientes.

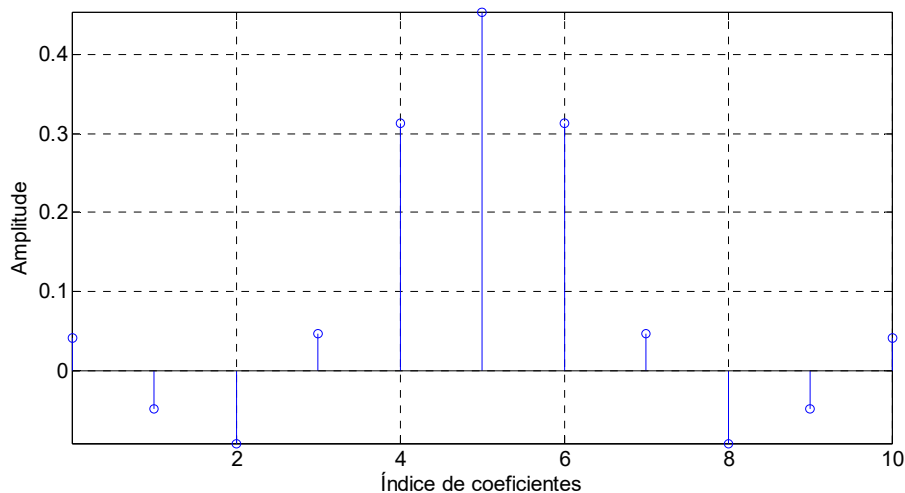


Figura 7 – Exemplo de resposta ao impulso para um filtro FIR com simetria ímpar de 11 coeficientes.

Em vez de implementar este filtro utilizando a arquitetura ilustrada na Figura 6, uma estrutura mais eficiente e otimizada pode ser explorada para gerar implementações de *hardware* de filtros eficientes, conforme a Figura 8. Em geral,

a primeira abordagem requer  $M$  multiplicações. Em contraste, a arquitetura simplificada utiliza  $(M+1)/2$  multiplicadores. A simetria de coeficientes para um número par de termos também pode ser explorada, reduzindo para  $M/2$  multiplicadores, como ilustrado na Figura 9.

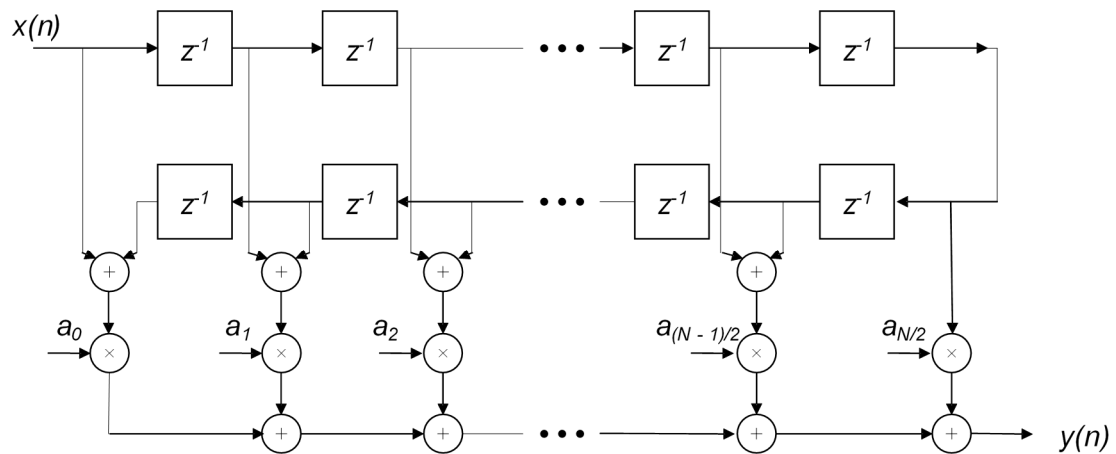


Figura 8 – Estrutura simplificada de um filtro FIR explorando a simetria ímpar de coeficientes.

Fonte: Adaptado de Xilinx (2011).

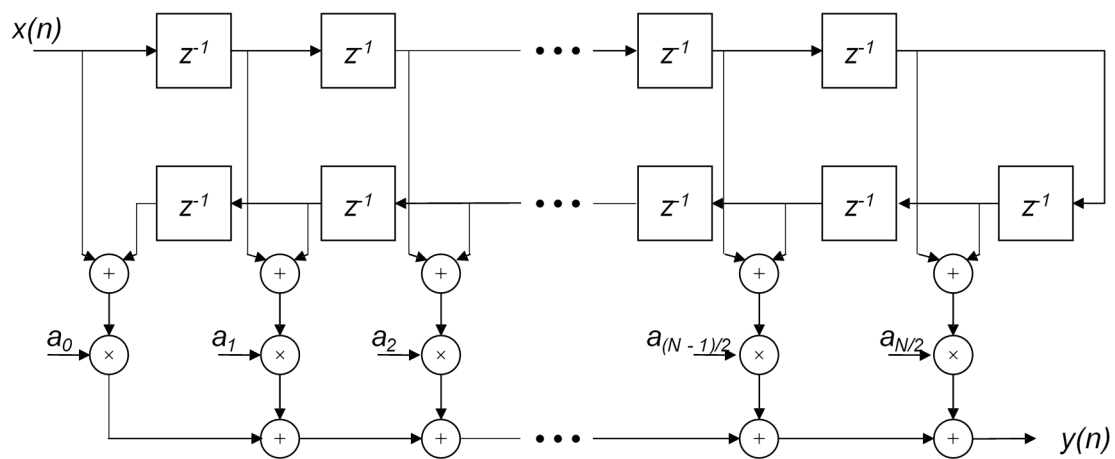


Figura 9 – Estrutura simplificada de um filtro FIR explorando a simetria par de coeficientes.

Fonte: Adaptado de Xilinx (2011).

## 2.5.2 Conversão de taxa de amostragem

A técnica que realiza a alteração da taxa de amostragem de dados é conhecida como conversão de taxa de amostragem (SRC - *Sampling Rate Conversion*) (PAU, 2000).

A SRC realiza dois tipos de conversões: interpolação e decimação. Ambos são métodos para DSP fundamentados em aplicações de sistemas discretos que requerem, respectivamente, aumento e decrescimento na taxa de amostragem. Assim, utiliza-se a SRC com finalidade de manter os erros dentro de limites prescritos, como *aliasing*<sup>3</sup> de banda de passagem ou erro de imagem, visando atingir um baixo consumo de processamento (HENTSCHEL; FETTWEIS, 2000; NGUYEN; SHIN; YEN, 2011; SRIVASTAVA; MEHRA, 2015).

Embora o intuito de minimizar o número de multiplicadores em aplicações de interpolação/decimação seja um desafio, o trabalho de Hogenauer (1981) mostra algumas realizações satisfatórias. Em trabalhos mais recentes, a técnica foi aprimorada para ser aplicada em algoritmos *Fast FIR* (FFAs) em filtros de interpolação e decimação, a fim de reduzir ainda mais a complexidade computacional da aplicação (MAYILAVELANE; BERSCHIED, 2016).

### 2.5.2.1 Filtro de interpolação

A interpolação é uma técnica bem consolidada em SRC que consiste em aumentar a frequência de amostragem por um fator inteiro  $R$ , com o objetivo de aproximar-se ao sinal original, ou seja, analógico (PAU, 2000; JINBO, 2007).

Uma alternativa de estruturas menos complexas que pode ser utilizada é o filtro CIC (*Cascaded Integrator-Comb*) ou filtro de Hogenauer. Os filtros CIC são implementados utilizando uma cascata de integradores (Figura 10(a)), denominada seção integradora, que opera a uma taxa de amostragem  $F_S$ , e um

---

<sup>3</sup> *Aliasing*: Comumente referida às distorções provocadas por violações do teorema de amostragem – frequência no sinal original acima da metade da frequência de amostragem (OPPENHEIM; WILLSKY; NAWAB, 1997).

número igual de estágios de diferenciadores (Figura 10(b)), denominado seção *comb*, operando a uma taxa de amostragem  $F_s/R$ .

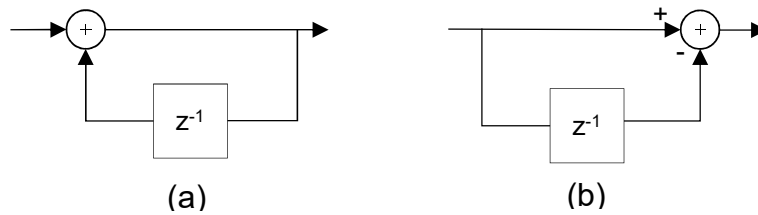


Figura 10 – Componentes de um filtro CIC. (a) Estágio integrador. (b) Estágio *comb*.

Fonte: Babic e Renfors (2005).

Esses filtros não exigem multiplicadores (HOGENAUER, 1981; BABIC; RENFORS, 2005; XUAN et al., 2014). A Figura 11 indica a forma básica de um filtro de interpolação CIC. Esta seção consiste de  $N$  estágios de *comb* com um atraso diferencial de  $M$  amostras por fase. Conforme Hogenauer (1981), na prática o valor de  $M$  varia entre 1 e 2.

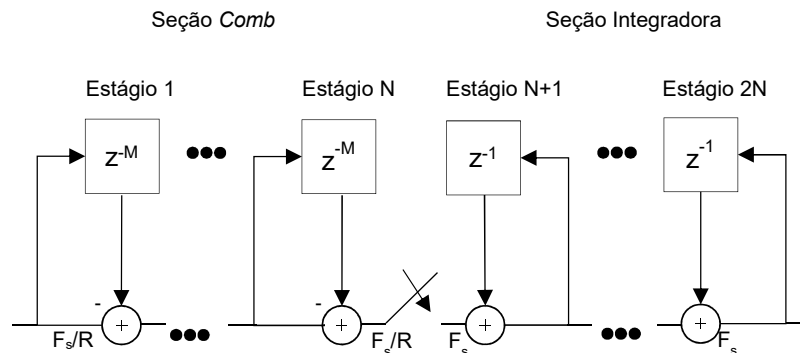


Figura 11 – Estrutura simplificada e generalizada de um filtro de interpolação CIC.

Fonte: Hogenauer (1981).

As funções do sistema para um integrador simples e para um filtro *comb* são representadas pelas Equações 3 e 4, respectivamente.

$$H_i[z] = \frac{1}{1 - z^{-1}} \quad (3)$$

$$H_c[z] = 1 - z^{-R.M} \quad (4)$$

### 2.5.2.2 Decimação

A decimação é uma técnica que consiste na diminuição da frequência de amostragem por um fator inteiro  $R$  de dados amostrados acima da sua taxa de Nyquist (CROCHIERE; RABINER, 1981). Essa técnica é importante para reduzir a exigência de taxa de dados no DSP (MA et al., 2014) e, no caso da imagem ultrassônica, compactar dados para transferência à etapa de conversão de varredura, onde o custo dos componentes de compressão e descompressão é mais baixo do que os circuitos necessários para a rápida transferência de dados (BOONLEELAKUL et al., 2013).

### 2.5.3 Atraso variável

Para a realização da abertura sintética com  $na$  elementos, na técnica de focalização *beamforming*, os elementos da matriz piezoelétrica são excitados com tempos de atraso programáveis distintos para focalizar as ondas ultrassônicas. Sendo assim, os sinais recebidos também herdam esses atrasos (SHEN et al., 1995; FELDKAMPER et al., 2000; AGARWAL et al., 2007). O tempo de atraso aplicado aos sinais de um elemento transdutor varia durante a recepção de uma linha de varredura e a precisão dos cálculos desses valores influencia diretamente na qualidade da imagem de US (FELDKAMPER et al., 2000).

Na Figura 12 é ilustrada a focalização da recepção dinâmica de US pela técnica DAS, onde frentes de ondas, originadas de uma fonte pontual, são captadas pelos elementos da matriz e convertidas nos sinais elétricos de RF. Em seguida, aplica-se coeficientes de atraso temporal (*delay*) para o alinhamento, possibilitando assim o somatório coerente após a etapa de apodização.

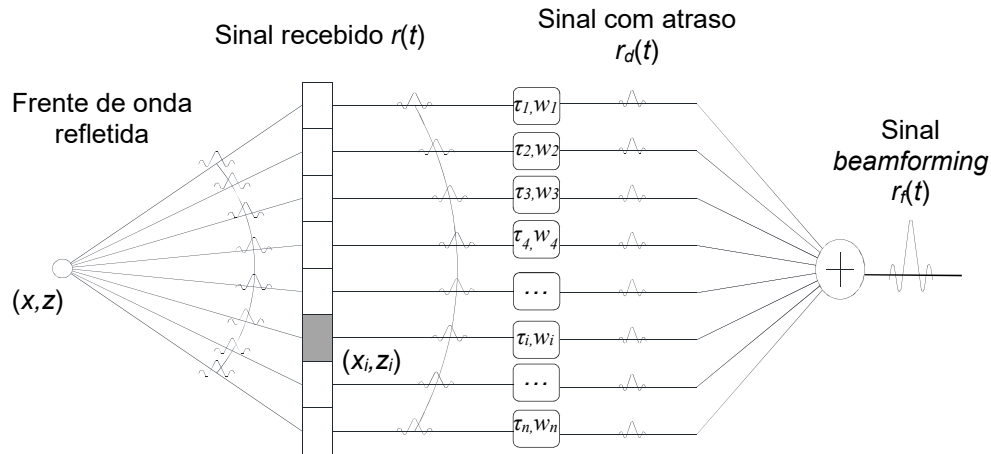


Figura 12 – Focalização de recepção dinâmica de US através da abertura sintética pelo método *beamforming*.

Fonte: Adaptado de Sohn et al. (2011).

O número de linhas de varredura do sistema ( $n_{sc}$ ) é calculado a partir da definição do tamanho de abertura aplicando-se a Equação 5:

$$n_{sc} = ne - na + 1, \quad (5)$$

sendo  $ne$  o número total de elementos do transdutor e  $na$  o número de elementos da abertura (ASSEF, 2013). Por exemplo, considerando um transdutor de 128 elementos e abertura ativa de oito elementos, o número total de linhas de varredura é 121.

#### 2.5.4 Apodização

A apodização realiza o controle da amplitude dos sinais recebidos, através de janelas de ponderação, para a redução da intensidade no feixe irradiado ao redor do ponto de focalização, gerando-se assim, imagens de US superiores em resolução de contraste (NIKOLOV; JENSEN; TOMOV, 2008; SYNNEVÅG; AUSTENG; HOLM, 2011; ASSEF, 2013).

As constantes de apodização são calculadas e aplicadas individualmente nos sinais de US que irão compor cada linha de varredura, a fim

de reduzir os lóbulos laterais do padrão de feixe (CINCOTTI et al., 1999). Porém, há uma relação de compromisso na escolha da técnica, pois estes coeficientes também têm o efeito de aumentar a largura do lóbulo principal, degradando a resolução lateral da imagem (FRAZIER; O'BRIEN, 1998).

Para calcular os coeficientes de apodização é necessária uma relação direta entre as características do transdutor e o padrão de campo (JENSEN; MUNK, 1998). No trabalho de Assef et al. (2016), foi avaliada uma variedade de janelas de ponderação, na transmissão e recepção, tais como Hamming, Hanning, Blackman e Kaiser, dentre outras.

Tendo em vista que para realizar a apodização na transmissão é necessário um sistema eletrônico de custo adicional, uma alternativa aplicada na maioria dos sistemas de imagem pulso-eco é realizar a apodização apenas no modo de recepção (CINCOTTI et al., 1999).

#### 2.5.5 Somatório coerente

No processo de reconstrução de imagem pela técnica *beamforming*, após o alinhamento dos sinais da abertura, um único sinal é gerado a partir do somatório coerente de todos os feixes de recepção, criando assim uma linha de varredura ou *scanline* (NIKOLOV; JENSEN; TOMOV, 2008). Esse processo é chamado *delay and sum* (DAS) e é representado pela Equação 6:

$$r_f[n] = \sum_{m=1}^M w_m \cdot x_m \cdot (n - \Delta_m[n]), \quad (6)$$

onde  $w_m$  é o peso (valor entre 0 e 1) da janela de ponderação,  $x_m[n]$  é o sinal de entrada registrado no número de elemento  $m \in [1, M(\text{canais})]$  com o tempo  $n$  atrasado com um retardo (*delay*) apropriado  $\Delta_m[n]$  e como resultado ( $z[n]$ ) a saída do formador do feixe (ASEN et al., 2012).



### 2.5.6 Demodulação

No caso do DSP para geração de imagens por US em Modo B, a intensidade do brilho, na imagem, é resultado da modulação da amplitude do sinal extraído através da demodulação, seguida pela detecção de envoltória (JINBO, 2007). Tipicamente três métodos convencionais são aplicados para extrair o envelope de baixa frequência das linhas de varredura: demodulação baseada na Transformada de Hilbert, demodulação de quadratura, e quadratura e filtragem (ZHOU; ZHENG, 2015). Conforme discutido por Levesque e Sawan (2009), embora a Transformada de Hilbert seja mais eficiente e tenha mais acurácia que os demais métodos, os algoritmos de filtragem e quadratura são geralmente preferidos, devido ao seu baixo custo computacional e complexidade.

No trabalho de Yang e Fox (2004), ressalta-se a importância da demodulação associada à qualidade de imagem de US, uma vez que no sinal processado há presença de *speckles*<sup>4</sup> que afetam tanto a interpretação humana das imagens quanto as técnicas automatizadas de detecção e extração de características.

Neste trabalho será utilizado um método baseado na aproximação da Transformada de Hilbert, evitando-se assim a necessidade da complexa computação da transformada rápida de Fourier (FFT - *Fast Fourier Transform*).

#### 2.5.6.1 Transformada de Hilbert e detecção de envoltória

A demodulação baseada na Transformada de Hilbert envolve a extração do sinal analítico da informação de RF. Uma vez que o sinal analítico é complexo, onde o componente real (*I* - em fase) é o sinal original e o componente imaginário (*Q* - quadratura) é a Transformada de Hilbert do sinal original, com deslocamento

---

<sup>4</sup> *Speckle*: Em imagens coerentes, tais como em US, o valor de sinal observado de uma célula de resolução é gerado pela soma de sinais coerentes de vários dispersores dentro da célula. Estes sinais interferem para causar o padrão de imagem granular conhecido como *speckle* (HOKLAND; TAXT, 1994).

de fase de  $90^\circ$  na banda de operação, a sua magnitude  $E$  é calculada conforme a Equação 7 (GOLD; OPPENHEIM; RADER, 1969; QIU et al., 2014b):

$$E(t) = \sqrt{I^2(t) + Q^2(t)}, \quad (7)$$

onde  $I(t)$  e  $Q(t)$  são, respectivamente, os valores absolutos de fase  $I$  e de quadratura  $Q$  oriundos da Transformada de Hilbert (OPPENHEIM; SCHAFER, 1975).

Como exemplo, na Figura 13 são ilustrados os esquemas simplificados de demodulação para o método de imagem por US, em termos do método de demodulação analógico tradicional (Figura 13(a)) e o método de demodulação digital (Figura 13(b)), onde o sinal de eco é adquirido diretamente pelo ADC (QIU et al., 2014a). Os blocos FPB e S/H representam as funções de filtro passa-baixa e amostragem e retenção (*sample and hold*), respectivamente.

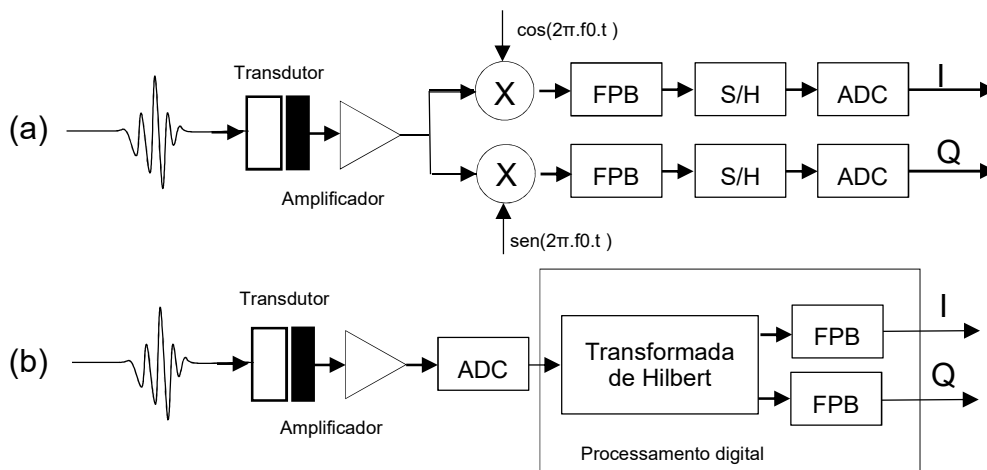


Figura 13 – Exemplos de esquemas de demodulação. (a) Demodulação analógica convencional. (b) Método de demodulação em quadratura por Transformada de Hilbert.

Fonte: Adaptado de Qiu et al. (2014a).

### 2.5.7 Compressão logarítmica

Como a variação na amplitude dos dados de RF é relativamente alta, caso a mesma imagem fosse mapeada linearmente para uma imagem de escala de cinza, por exemplo com oito *bits*, muitas estruturas importantes do tecido

poderiam apresentar valores de imagem muito baixos. Sendo assim, a presença de pontos de amplitude muito alta ofusca os sinais menores. Dessa maneira, a fim de obter um equilíbrio e adequar a faixa dinâmica à visualização, os valores de amplitude são mapeados não linearmente por uma função de aparência logarítmica que ajusta o intervalo dinâmico e, conseqüentemente, melhora o contraste da imagem (WIKISONIX, 2013). Geralmente, as imagens de US são apresentadas com faixa dinâmica entre -30 a -50 dB (JENSEN et al., 2005; BONI et al., 2012).

### 3 MATERIAIS E MÉTODOS

Neste capítulo são apresentadas as especificações de requisitos e as ferramentas empregadas durante a modelagem, implementação e validação dos módulos que compõem a cadeia de processamento de US embarcada em FPGA para reconstrução de imagem.

#### 3.1 ESPECIFICAÇÃO DE REQUISITOS

Para a modelagem e implementação do sistema de processamento de sinais de US foi definida inicialmente a especificação de requisitos. Esta especificação é apresentada na Tabela 2 com as definições dos principais parâmetros para processamento dos sinais de RF para reconstrução de imagem.

Tabela 2 – Especificação de requisitos do modelo de processamento.

Item	Especificação
1	Utilização dos sinais de RF adquiridos pelo sistema ULTRA-ORS
2	Emprego do <i>kit</i> comercial de desenvolvimento DE4-230 com a FPGA Intel Stratix IV
3	Modelagem através da ferramenta DSP Builder no Simulink
4	Reconstrução baseada no transdutor matricial convexo AT3C52B (BROADSOUND CORP., TAIWAN) de 128 elementos
5	Abertura de transmissão e recepção de oito canais
6	Processamento de 121 <i>scanlines</i> com comprimento de 2046 <i>words</i>
7	Frequência de amostragem de 40 MHz e resolução de 12 <i>bits</i>
8	Filtro passa-baixa de entrada com frequência de corte de 8 MHz
9	Fator de interpolação $R = 4$ (resolução temporal de 6,25 ns)
10	Aplicação da técnica <i>beamforming</i> com o método DAS
11	Ajuste de atraso de focalização com profundidade de 25 mm
12	Aplicação das apodizações Retangular, Gaussiana, Hamming e Hanning
13	Fator de decimação de $R = 4$ após o DAS
14	Demodulação utilizando uma aproximação da Transformada de Hilbert
15	Detecção de envoltória a partir dos componentes <i>I</i> e <i>Q</i>
16	Compressão logarítmica de -30 e -50 dB
17	Utilização do analisador lógico SignalTap II do <i>software</i> Quartus II para aquisição de sinais processados pela FPGA
18	Correção de varredura para geração de imagem final de US através do <i>software</i> Matlab

### 3.2 ENGENHARIA DIRIGIDA A MODELOS

A metodologia adotada neste trabalho de pesquisa foi exploratória experimental utilizando a técnica de Engenharia Dirigida a Modelos (MDE - *Model Driven Engineering*). Nesta abordagem, são utilizados modelos com diferentes níveis de abstração que representam sistemas como elementos principais do desenvolvimento. Com isso, é possível melhorar a compreensão do problema, bem como facilitar a manutenção e evolução do sistema, melhorando a depuração e possibilitando a identificação de erros durante a fase do desenvolvimento. Também viabiliza a reutilização de códigos, diminuindo o tempo gasto no desenvolvimento e, conseqüentemente, o custo final da aplicação (BRANCO et al., 2011; GIRON, 2015). A utilização do MDE permite a transformação de modelos (ESTEVEZ; MARCOS, 2012; GIRON, 2015), realizando um ciclo metodológico conforme ilustrado na Figura 14.

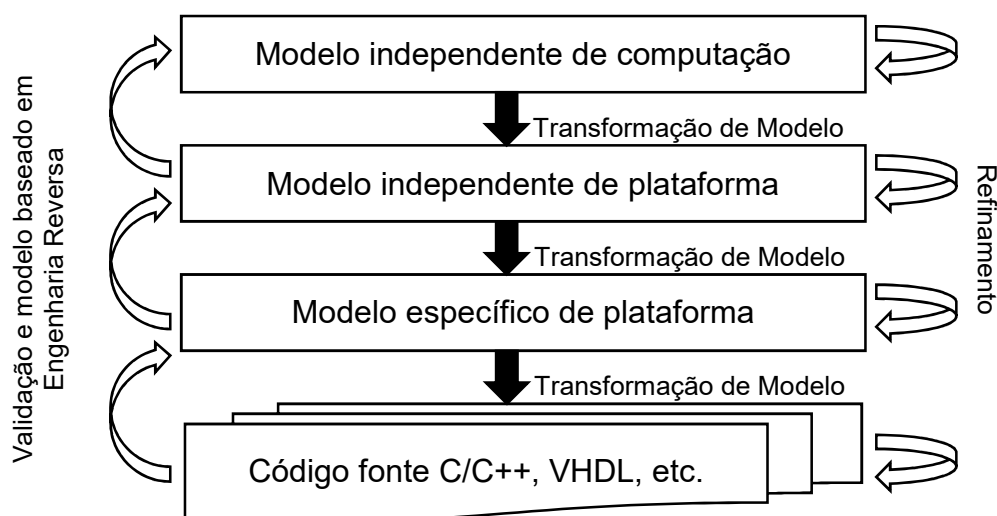


Figura 14 – Ciclo metodológico do MDE com a aplicação de ferramentas específicas para transformação de modelos em nível de registradores sintetizáveis.

Fonte: Adaptado de Giron (2015).

### 3.2.1 Simulink

O Simulink (MATHWORKS, EUA) é uma ferramenta para modelagem, simulação e análise de sistemas dinâmicos integrada ao *software* de cálculo numérico Matlab. Tal programa possibilita uma abordagem MDE para desenvolvimento de dispositivos embarcados, apresentando como principais vantagens sua linguagem de modelagem de fluxo de dados que permite a simulação, geração automática de código e verificação de sistemas (MATHWORKS, 2013; STEPHAN; ALALFI; CORDY, 2014). Dessa forma, o Simulink foi empregado como a ferramenta de desenvolvimento com maior nível de abstração, responsável pela transformação dos modelos necessários para este trabalho. De forma complementar, foi utilizada a ferramenta DSP Builder (INTEL CORP., EUA) para geração dos códigos HDL correspondentes à modelagem do sistema de processamento.

### 3.2.2 DSP Builder

O DSP Builder realiza o próximo passo do MDE que é a transformação para o modelo específico de plataforma. A ferramenta consiste de um conjunto de blocos padrões e avançados que permite a geração automática e rápida de códigos HDL para algoritmos DSP de elevado desempenho, através da abstração do projeto do sistema digital no nível RTL (*Register-Transfer Level*), a partir da modelagem realizada no Simulink.

A ferramenta converte os modelos projetados em códigos VHDL de baixo nível, otimizados e direcionados para as FPGA da Intel, através de conexão direta entre a combinação do Matlab e blocos do Simulink com o *software* de desenvolvimento Quartus II (INTEL CORP., EUA). Com isso, é possível simular e validar a aplicação desenvolvida. Na sequência, o sistema é convertido para o domínio de descrição de *hardware*, o qual modela o circuito digital síncrono, em termos de fluxo dos sinais digitais entre os registradores de *hardware* e operações lógicas realizadas nesses sinais (INTEL CORP., 2017).

### 3.3 FUNÇÃO DE CUSTO NRMSE

Para avaliação quantitativa dos principais blocos do modelo proposto neste trabalho em comparação com as funções implementadas no Matlab, foi adotada a função de custo da raiz quadrada do erro quadrático médio normalizado NRMSE (*Normalized Root Mean Squared Error*).

Segundo a literatura científica, o modelo é considerado excelente com NRMSE menor que 10%, bom para NRMSE maior que 10% e menor que 20%, justo para o NRMSE maior que 20% e menor que 30%, e pobre caso o NRMSE seja maior que 30% (JAMIESON; PORTER; WILSON, 1991; DETTORI et al., 2011).

O cálculo da NRMSE é apresentado na Equação 8:

$$NRMSE = \sqrt{\left(\frac{\sum_{i=1}^N ((M_i - S_i)^2)}{\sum_{i=1}^N ((M_i - \bar{M})^2)}\right)} * 100, \quad (8)$$

onde  $i$  é o índice da amostra,  $M_i$  é a amostra do modelo,  $S_i$  é a amostra de referência e  $\bar{M}$  é a média das amostras do modelo.

### 3.4 FERRAMENTAS DE PROGRAMAÇÃO E VALIDAÇÃO

#### 3.4.1 *Software* Quartus II

De forma geral, cada fabricante de dispositivo de lógica reconfigurável também disponibiliza as respectivas ferramentas de desenvolvimento e programação. O Quartus II é a ferramenta oficial da Intel para o desenvolvimento de projetos utilizando os componentes FPGA, SoC (*System-on-Chip*), CPLDs (*Complex Programming Logic Devices*) e PowerSoCs (SoC de alto desempenho). A versão atual do *software* é o Quartus II 16 (Prime), sendo disponibilizado nas versões de avaliação – com número limitado de funcionalidades – e completa, sendo esta paga. Entretanto, foi utilizada uma

licença do Programa Universitário da Intel, disponível no LUS da UTFPR para utilização da versão completa do *software* no meio acadêmico.

#### 3.4.1.1 Analisador Lógico SignalTap II

O *software* Quartus II possui a ferramenta integrada analisador lógico SignalTap II que permite capturar, depurar e exportar os sinais gerados por qualquer FPGA, em tempo real, sem utilizar pinos extras de entrada e saída e sem o uso de equipamentos externos. A desvantagem é que esta depuração exige recursos do dispositivo, uma vez que todos os dados de sinal capturados são convenientemente armazenados na memória interna do dispositivo (INTEL CORP., 2016).

### 3.5 KIT COMERCIAL DE DESENVOLVIMENTO DE4-230

Devido à complexidade das etapas de processamento digital do *beamforming* de recepção para geração de imagem por US Modo B, foi imprescindível a adoção de um *kit* de FPGA para o desenvolvimento deste trabalho. Dessa forma, foi utilizado o sistema DE4-230 *Development and Education Board*, produzido pela empresa Terasic Inc. (Taiwan), em parceria com a Intel, e disponível no LUS da UTFPR.

A placa DE4-230 (Figura 15) representa uma plataforma de *hardware* projetada para aplicações de pesquisa que exigem maior desempenho, conectividade serial, e interface avançada para memórias. O *kit*, comercializado ao preço de U\$ 2.995<sup>5</sup> possui uma FPGA Stratix IV GX EP4SGX230KF40C2 com 228.000 elementos lógicos, 744 pinos livres de entrada/saída, até 17,2 *Mbits* de memória RAM e 1024 multiplicadores de 18x18 *bits*.

---

<sup>5</sup> Valor pesquisado em 20 de março de 2017 (TERASIC TECH., 2017).



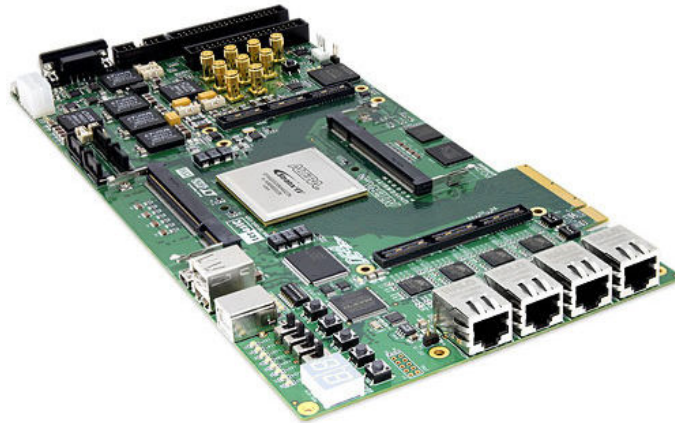


Figura 15 – Fotografia da placa de desenvolvimento DE4-230 com a FPGA Stratix IV.

Fonte: Terasic Tech. (2017).

### 3.6 PLATAFORMA DE PESQUISA ULTRA-ORS E *PHANTOM* DE US PARA TESTES

O protótipo completo do sistema ULTRA-ORS desenvolvido na UTFPR é mostrado na Figura 16(a) e Figura 16(b), nas disposições frontal e traseira, respectivamente.

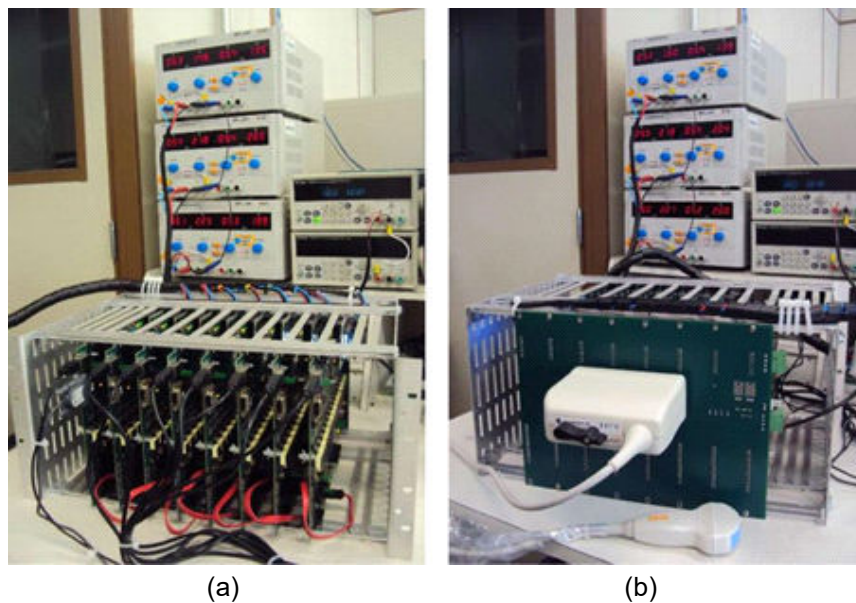


Figura 16 – Fotografias da plataforma de bancada ULTRA-ORS. (a) Foto frontal. (b) Foto traseira com transdutor matricial convexo conectado.

Fonte: Assef (2013).

O módulo eletrônico lógico para formação do feixe possibilita excitação simultaneamente de 128 canais com formas de ondas arbitrárias, abertura programável, e tensão de excitação de até 200 Vpp (Tensão pico-a-pico), permitindo controle individual de habilitação, amplitude de apodização com até 256 níveis, ângulo de fase e atraso temporal de disparo adequado para focalização na transmissão (ASSEF, 2013).

O módulo de recepção realiza a aquisição simultânea de 128 canais com acesso e transferência dos dados de RF digitalizados para um PC através de interfaces seriais para processamento das imagens. Entretanto, a latência na transferência de dados via a interface serial emulada, com taxa máxima de 1 Mbps (*Mbit* por segundo), somando ao tempo de processamento das rotinas do MATLAB, inviabilizam a pesquisa de geração de imagens em tempo real.

Neste trabalho, os dados brutos de RF foram capturados pelo sistema ULTRA-ORS utilizando um *phantom* mimetizador de tecidos biológicos de uso geral modelo 84-317. Os principais parâmetros do *phantom* são o coeficiente de atenuação e a velocidade do som, especificados em 0,5 dB/cm/MHz e 1540 m/s  $\pm$  6 m/s, respectivamente (Victoreem Inc., 1993). Na Figura 17 é apresentado o conjunto formado pelo *phantom*, transdutor multielemento AT3C52B, acrílico para suporte do transdutor e a ROI tracejada. Na Figura 18 é apresentada a ampliação dos pontos alvos da ROI avaliada neste trabalho.

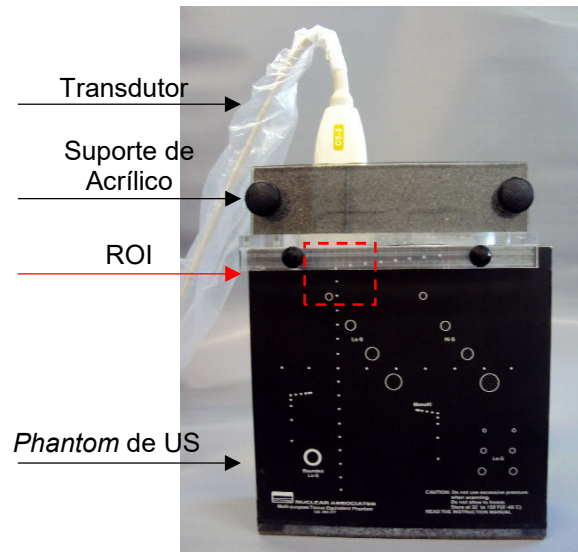


Figura 17 – *Phantom* mimetizador de tecidos biológicos utilizado para avaliação do trabalho.

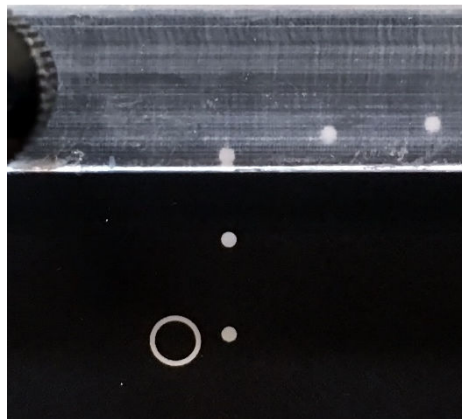


Figura 18 – Ampliação da ROI no *phantom* de US com seus respectivos pontos alvos avaliados neste trabalho.

### 3.7 DIAGRAMA EM BLOCOS DO SISTEMA PROPOSTO

Para melhor entendimento deste trabalho, a Figura 19 ilustra o diagrama em blocos do sistema de modelagem, implementação e validação das etapas de processamento de sinais para reconstrução de imagens por US, composto por: plataforma de pesquisa ULTRA-ORS, transdutor matricial convexo AT3C52B, *phantom* de US, PC com interfaces USB e *kit* de FPGA DE4-230.

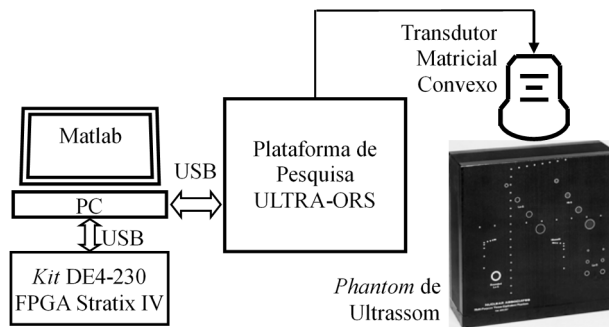


Figura 19 – Diagrama em blocos do sistema de modelagem, implementação e validação das etapas DSP para reconstrução de imagem por US.

Na Figura 20 é apresentada a foto do microcomputador Dell Inspiron 3647 com processador Intel Inside CORE i5, memória RAM de 4 GB, HD de 1 TB, além da placa de desenvolvimento DE4-230. Na tela do PC são exemplificadas a aquisição dos sinais processados através do SignalTap II e a uma imagem em Modo B resultante do processamento proposto.

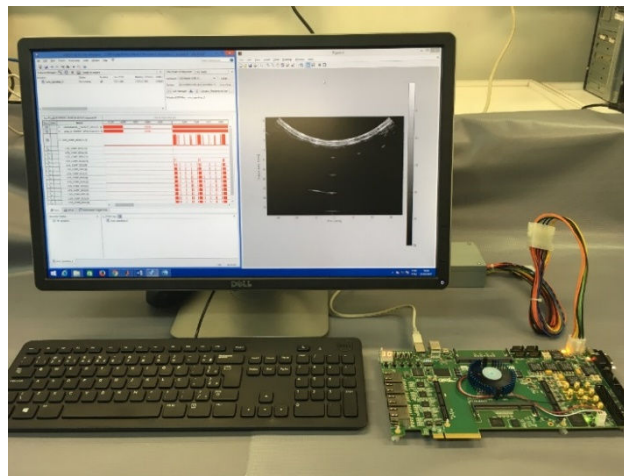


Figura 20 – Foto do PC com exemplos de telas de aquisição e processamento, e *kit* DE4-230.

### 3.8 IMPLEMENTAÇÃO DO MODELO PROPOSTO

Este tópico descreve os detalhes de implementação das etapas necessárias para desenvolver o modelo proposto e definir os parâmetros necessários. O projeto foi, primeiramente, modelado e simulado no ambiente Simulink, e então implementado e testado no *kit* FPGA integrado. O projeto completo do sistema DSP baseado em FPGA para a geração de imagem por US

é apresentado no Apêndice A. Para a conversão de varredura e apresentação final dos resultados da reconstrução da imagem, utilizaram-se rotinas desenvolvidas no *software* Matlab.

O modelo de processamento digital de reconstrução de imagem foi desenvolvido com o auxílio da biblioteca Altera DSP Builder Standard Blockset (Figura 21) desenvolvida pela Intel em seu pacote DSP Builder integrado ao Simulink.

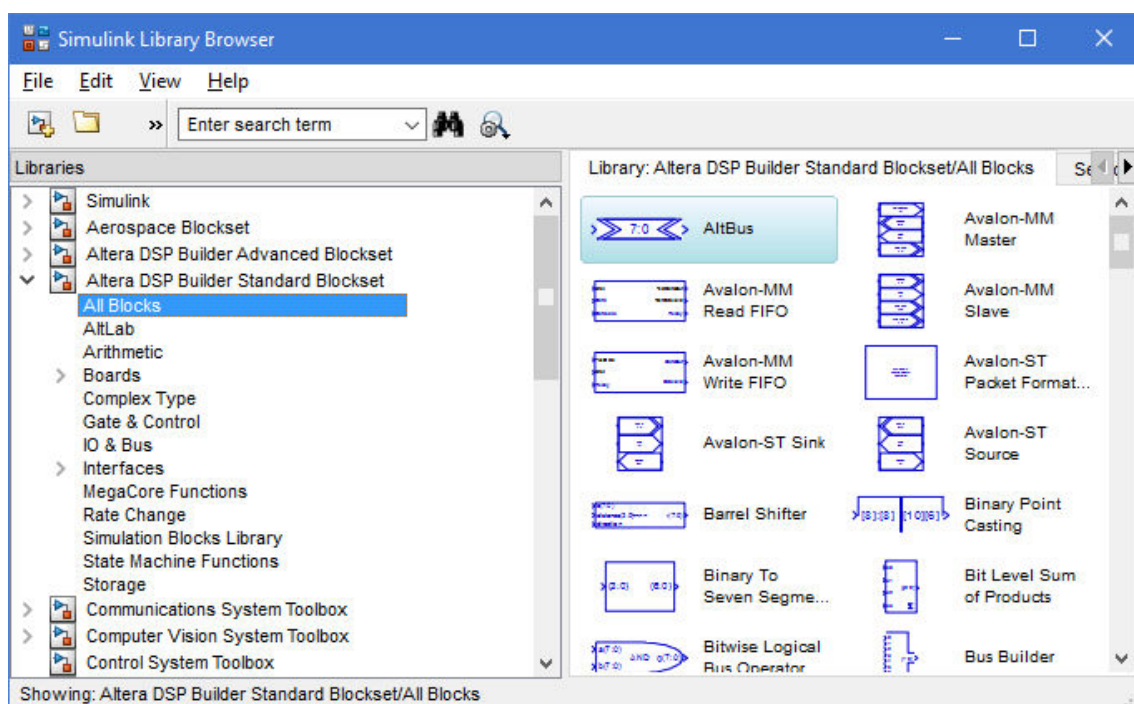


Figura 21 – Janela do ambiente Simulink com a biblioteca Altera DSP Builder Standard Blockset.

### 3.8.1 Configurações iniciais

Anterior ao desenvolvimento do diagrama de blocos do processamento de sinal *beamforming*, são adicionados os blocos de configuração que relaciona o projeto com o modelo de FPGA para geração de síntese. Nos tópicos seguintes estão descritas as configurações necessárias.

### 3.8.1.1 Bloco Device

O bloco “Device” marca um subsistema Simulink específico com o tipo do dispositivo FPGA para síntese do projeto. Com a inserção do bloco, o DSP Builder gera os arquivos de projeto e rotinas de algoritmos que se relacionam com um nível de hierarquia. Todos os blocos em subsistemas abaixo deste nível tornam-se parte do projeto RTL e todos os blocos acima deste nível de hierarquia tornam-se parte do *testbench*<sup>6</sup> (INTEL CORP., 2016). A Tabela 3 mostra os controles e parâmetros para o bloco do dispositivo para a realização do projeto.

Tabela 3 – Parâmetros de configuração do bloco “Device”.

Parâmetro	Especificação
Família do Dispositivo	Stratix IV
Dispositivo	Stratix IV GX
Membro da família	EP4SGX230KF40C2
Grau de velocidade	2

### 3.8.1.2 Bloco Signal Compiler

O bloco “Signal Compiler” tem como função compilar o modelo e gerar os arquivos HDL. Após a geração dos códigos em VHDL, o *software* Quartus II realiza a síntese para posterior gravação no *kit* com FPGA Intel. A Tabela 4 relaciona os parâmetros para o bloco.

Tabela 4 – Parâmetros de configuração do bloco “Signal Compiler”.

Parâmetro	Especificação
Família	Stratix IV
Dispositivo	EP4SGX230KF40C2

<sup>6</sup> *Testbench*: Ambiente (geralmente virtual) usado para verificar a correção ou integridade de um projeto ou modelo de um produto de *software*.

### 3.8.1.3 Bloco Clock

O bloco de “Clock” no nível superior de um projeto serve para definir o domínio de *clock* (ciclo) de *hardware*. A Tabela 5 lista os parâmetros para a configuração do bloco “Clock”.

Tabela 5 – Parâmetros de configuração do bloco “Clock”.

Parâmetro	Especificação
Período de <i>Clock</i>	20
Unidade do Período	ns
Tempo de amostragem Simulink	20e-9
Nome do <i>Reset</i>	aclr
Tipo de <i>Reset</i>	<i>Atcive Low</i>

### 3.8.1.4 Bloco PLL

O bloco “PLL” (*Phase-Locked Loop*) gera um ou mais sinais de *clock* baseado em uma fonte de referência, frequentemente o oscilador de cristal (WANG et al., 2015).

Realiza-se a configuração inicial do “PLL” com quatro saídas de *clock*. Na Tabela 6 é mostrada a configuração para gerar os *clocks* de 40, 50, 80 e 160 MHz, respectivamente, e a nomenclatura utilizada no projeto.

Tabela 6 – Parâmetros de configuração do bloco “PLL”

Configuração			Resultado	
Nome	Período Multiplicador	Período Multiplicador	Período (ns)	Frequência (MHz)
PLL_clk0	1	1	20	50
PLL_clk1	5	4	25	40
PLL_clk2	5	8	12,5	80
PLL_clk3	5	16	6,25	160

### 3.8.2 Bloco Sinal RF

Os dados brutos de entrada contêm 121 *scanlines*, sendo cada uma com oito canais – devido ao número de abertura selecionado – e 2046 amostras. Os mesmos foram divididos em 4 matrizes (com [30 30 30 31] *scanlines* x 8 canais) armazenadas em oito memórias (tipo M9k) de 65536 x 14 *bits* para serem processadas individualmente devido à capacidade da FPGA. A entrada de dados dos oito canais ocorre paralelamente. A Figura 22 exemplifica os subsistemas “Dados\_Entrada”, contendo os blocos de memórias com os sinais brutos de US que são enviados simultaneamente para a etapa de processamento “Filtro FIR”.

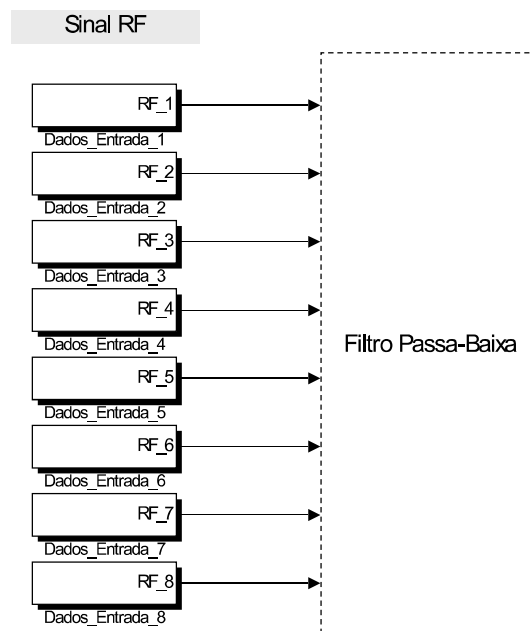


Figura 22 – Implementação dos dados de entrada com oito canais armazenados em memórias individuais.

Cada canal tem uma memória dedicada para armazenamento dos dados brutos e um contador na frequência de 40 MHz para acesso aos dados da memória, além de um pino de sincronismo. Sendo assim, foram criados oito subsistemas contendo os blocos indicados na Figura 23 e respectivas configurações na Tabela 7.



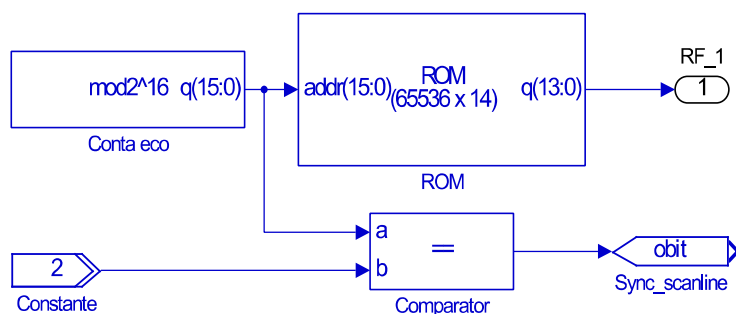


Figura 23 – Detalhe do subsistema “Dados de Entrada”.

Tabela 7 – Especificações do subsistema “Dados de Entrada”.

Bloco	Parâmetro	Especificação
Counter	Tipo do barramento	Inteiro sinalizado
	Número de <i>bits</i>	[16]:[ ]
	<i>Clock</i>	PLL_clk1 (25 ns)
ROM	Número de <i>words</i>	2 <sup>16</sup>
	Tipo de dados	Inteiros sinalizados
	Número de <i>bits</i>	[14]:[ ]
	Tipo do Bloco de Memória	M9K
	Matriz Matlab	“dados_de_entrada”
Comparator	Operador	a == b
Constant	Valor	2
	Tipo de barramento	Inteiro não sinalizado
	Número de <i>bits</i>	[2]:[ ]
Output	Tipo de barramento	Único <i>bit</i>
	Tipo externo	Inferido

A Figura 24 ilustra os dados brutos de entradas de oito canais referentes à *scanline* central 61 (canais de 61 a 68) armazenados em memória.

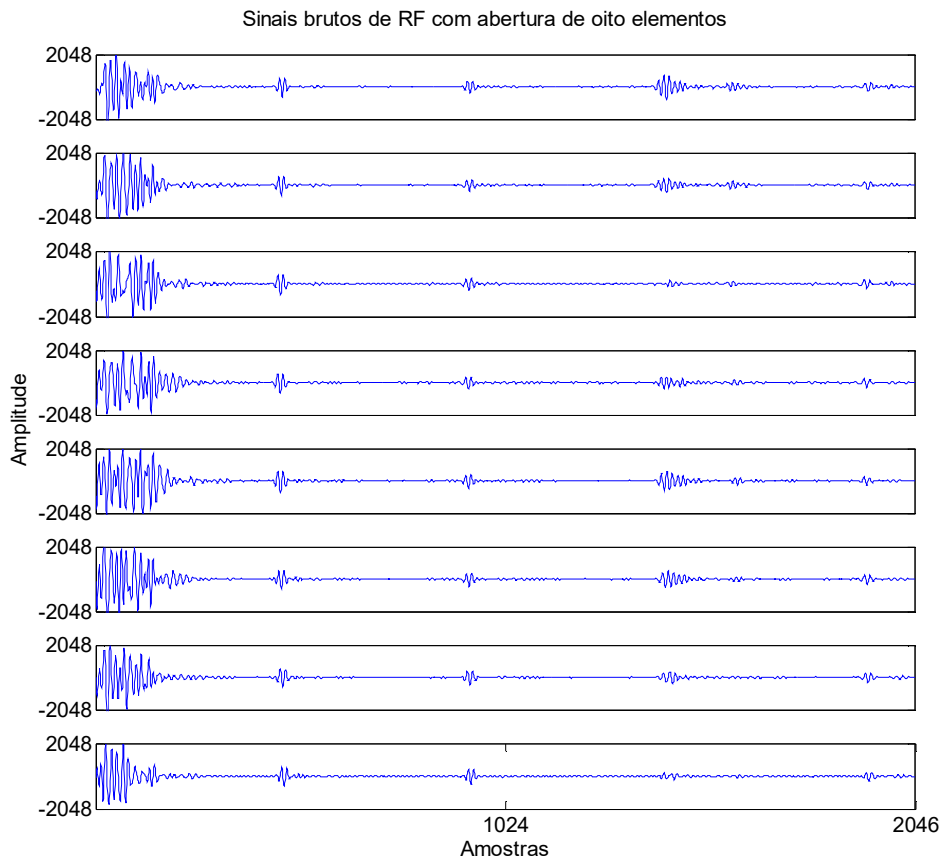


Figura 24 – Sinais brutos de RF com abertura de oito elementos para formação da *scanline* 61.

### 3.8.3 Bloco Filtro Digital FIR

Nesta seção é apresentado o projeto do filtro digital FIR passa-baixa implementado em FPGA, como a primeira etapa de processamento de sinais brutos de US. O projeto e avaliação do filtro digital foi realizado com o auxílio da ferramenta de projeto e avaliação de filtros digitais FDATool, disponibilizada no Matlab. Os parâmetros especificados para o projeto do filtro constam na Tabela 8.

Tabela 8 – Especificação do filtro digital FIR.

Parâmetro	Especificação
Frequência de amostragem - $F_s$	40 MHz
Tipo de resposta	Passa-baixa
Método do projeto	FIR - <i>Generalized Equiripple</i>
Ordem do Filtro	15
Fator de Densidade	20
Frequência da banda passante – $F_{pass}$	3,2 MHz
Frequência da banda de parada – $F_{stop}$	8,0 MHz
Atenuação na banda passante – $A_{pass}$	-1 dB
Atenuação na banda de parada – $A_{stop}$	-50 dB

Fonte: Assef et al. (2016).

A partir da definição dos parâmetros, utilizou-se a ferramenta FDATool para cálculo dos coeficientes de filtragem, resultando em um filtro com resposta ao impulso simétrica de 16 coeficientes indicados na Tabela 9 com precisão do tipo *double* (64 bits).

Tabela 9 – Coeficientes simétricos de filtragens calculados no FDATool.

Nº do Coeficiente	Coeficiente obtido	Nº do Coeficiente	Coeficiente obtido
1	-0,00805246535535765	9	0,24892007978955700
2	-0,02031883006153170	10	0,19336131713264000
3	-0,02880053432836590	11	0,10853961742278900
4	-0,01643518329923500	12	0,03036968142921930
5	0,03036968142921930	13	-0,01643518329923500
6	0,10853961742278900	14	-0,02880053432836590
7	0,19336131713264000	15	-0,02031883006153170
8	0,24892007978955700	16	-0,00805246535535765

As respostas em frequência da magnitude, fase e resposta ao impulso são apresentadas, respectivamente, na Figura 25, Figura 26 e Figura 27.

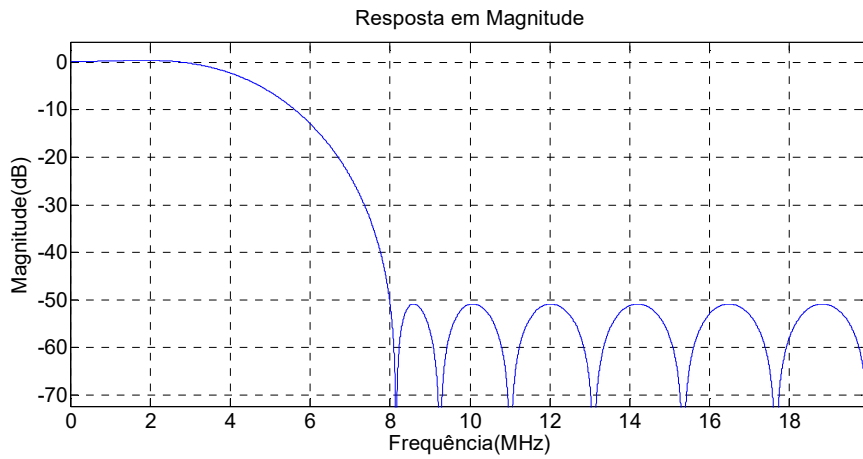


Figura 25 – Resposta em frequência da magnitude do filtro FIR obtida com a ferramenta FDATool no Matlab.

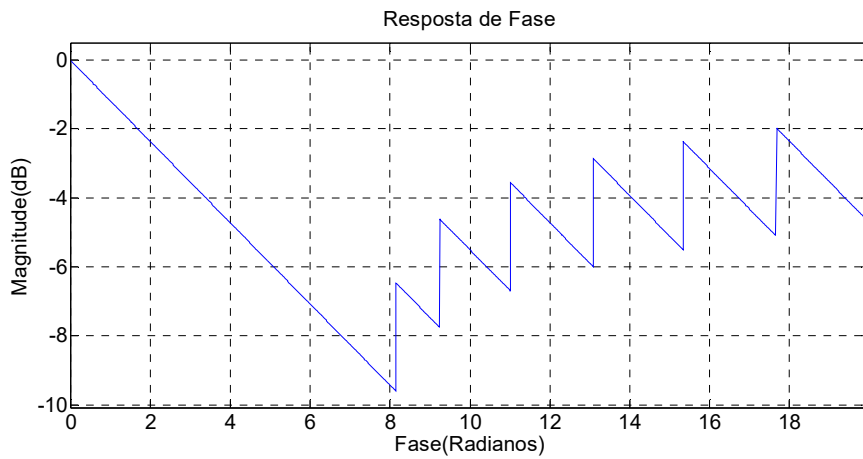


Figura 26 – Resposta de fase do filtro FIR obtida com a ferramenta FDATool no Matlab.

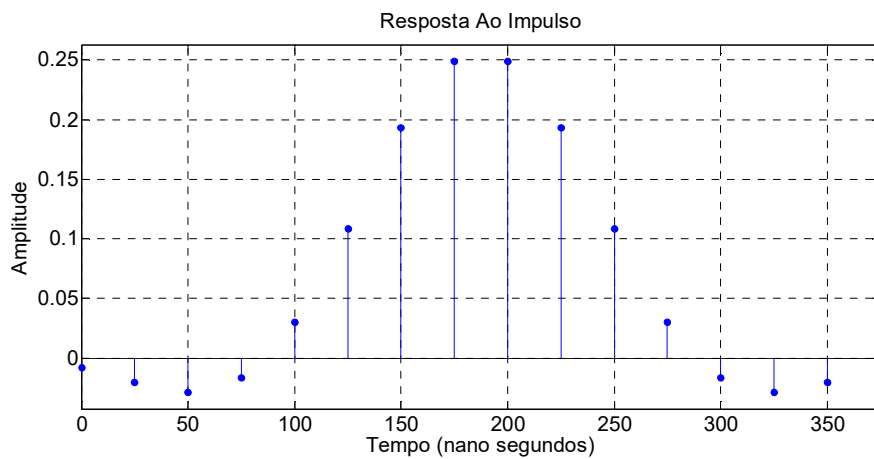


Figura 27 – Resposta ao impulso do filtro FIR obtida com a ferramenta FDATool no Matlab.

O filtro FIR passa-baixa foi implementado em oito canais paralelos (Figura 28) relacionados aos oito elementos da abertura sintética. Cada canal recebe os dados do bloco “Sinal RF” e tem entrada em um subsistema (FPB\_FIR\_[1 a 8]), onde é processado. Os sinais de saída são enviados para o bloco “Interpolação”.

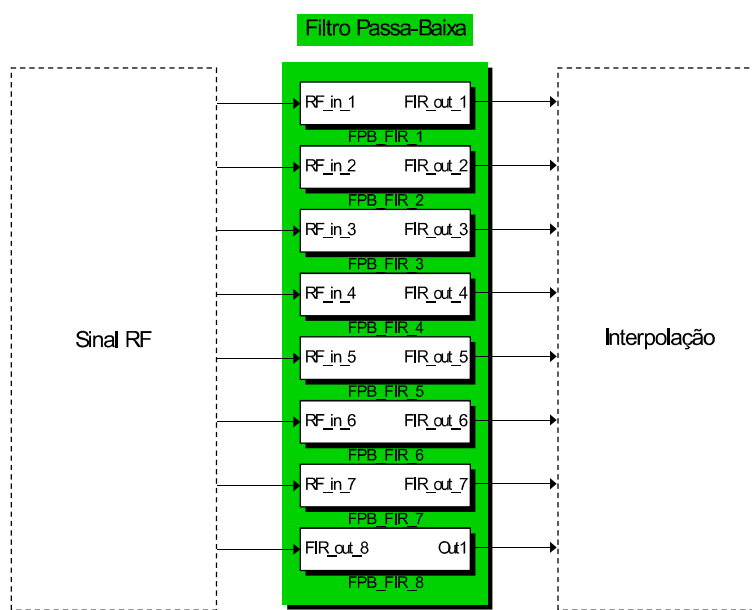


Figura 28 – Diagrama em blocos com o filtro FIR de oito canais.

O filtro proposto possui ordem par e uma resposta de impulso simétrica (Figura 27). Este fato foi explorado a fim de reduzir ainda mais a complexidade computacional do sistema. O modelo do subsistema que contém o filtro FIR discreto é apresentado na Figura 29. Em sequência, apresentam-se os parâmetros de configurações dos blocos utilizados na Tabela 10. Ressalta-se a otimização na quantidade de multiplicadores, aproveitando a simetria dos coeficientes de entrada, utilizando apenas oito multiplicadores para 16 coeficientes importados da área de trabalho do Matlab.

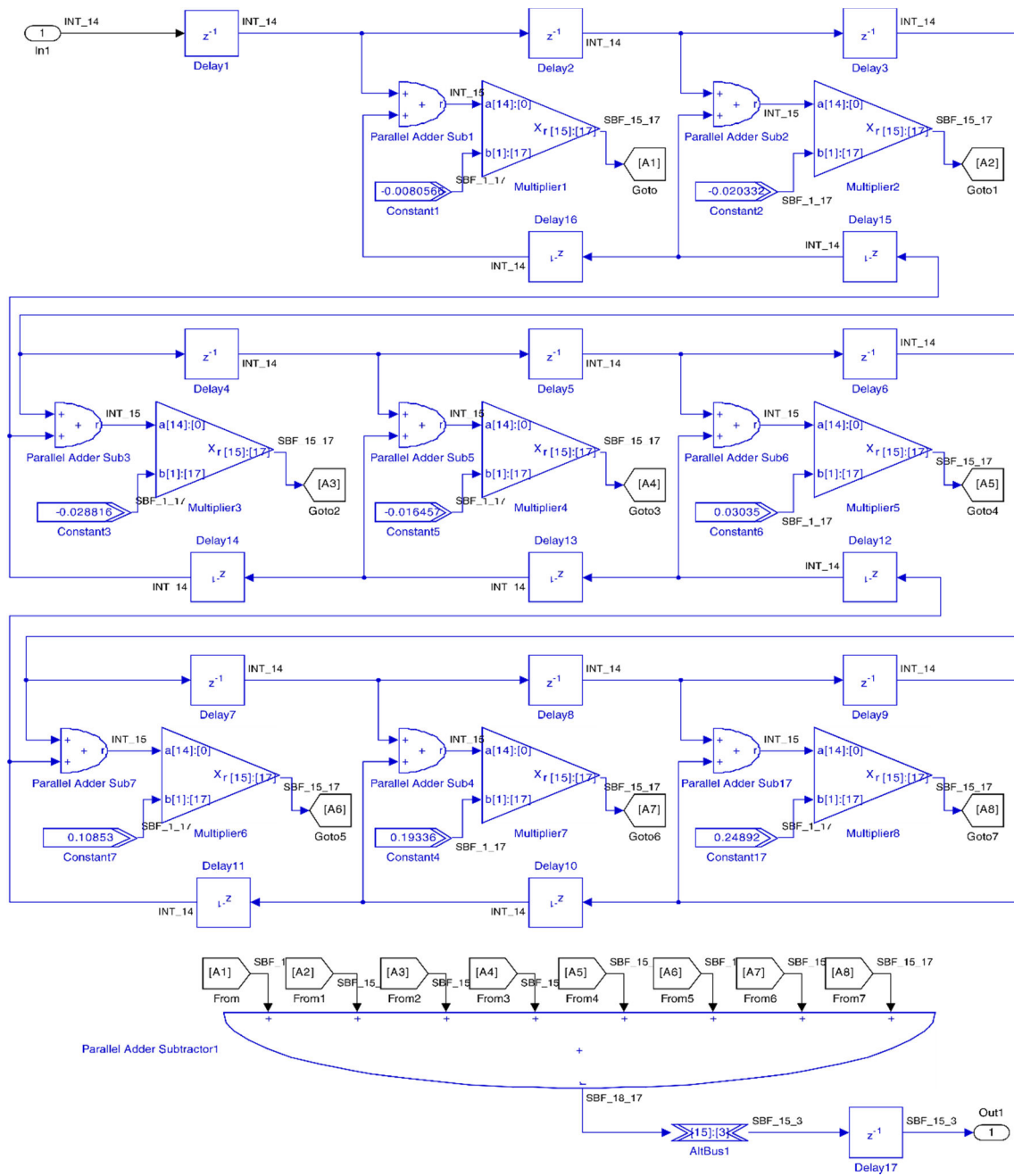


Figura 29 – Modelo do subsistema “Filtro FIR” com aproveitamento de simetria dos coeficientes para redução de multiplicadores.

Tabela 10 – Especificações de configuração dos componentes que compõem o filtro FIR.

Componente	Parâmetro	Especificação
Multiplier	Tipo do Barramento	Inteiro sinalizado
	Número de <i>bits</i> de entrada	[14]:[0]
	Número de estágios <i>pipeline</i>	1
Constant	Valor da constante	Simple_LPF_coefs(1 a 8)
	Tipo do Barramento	Fracionário sinalizado
	AltBus	Número de <i>bits</i>
	Saturar saída	Ativado

### 3.8.4 Bloco Interpolação *Cascaded Integrator-Comb*

A interpolação CIC foi implementada em oito canais paralelos (Figura 30) relacionados aos oito elementos da abertura sintética. Cada canal recebe os dados do “Filtro Passa-baixa” pela entrada do subsistema “Interpolacao\_[1 a 8]”, onde é processada a interpolação projetada e tem saída para o bloco “Atraso Variável”.

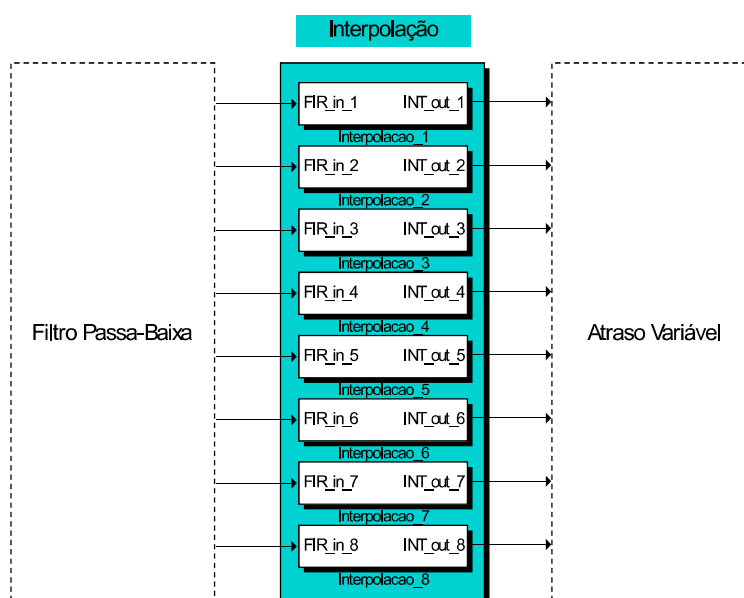


Figura 30 – Diagrama em blocos com a Interpolação de oito canais.

Esta seção apresenta considerações de projeto para filtros de interpolação CIC com a utilização da técnica SRC para DSP na conversão dos sinais amostrados em 40 MHz para 160 MHz.

O filtro de interpolação CIC é implementado com quatro diferenciadores (seção *comb*), conseqüentemente quatro integradores, e entre eles um bloco *flip-flop* tipo D para alteração da taxa de amostragem, como é apresentado na Figura 31. A especificação interna de cada item que compõe a interpolação encontra-se na Tabela 11.

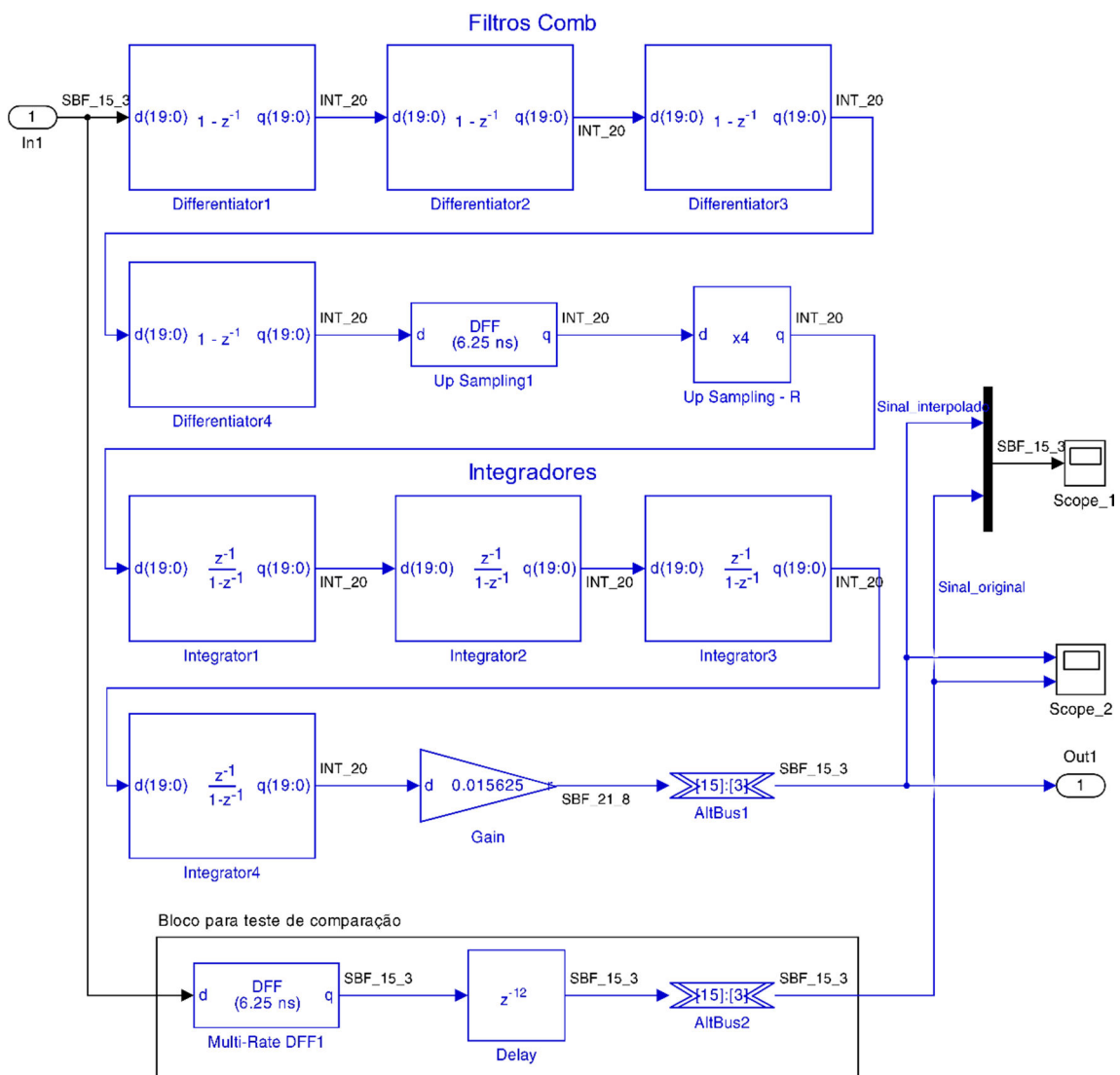


Figura 31 – Modelo do subsistema “Interpolação” com fator  $R = 4$ , implementado com quatro integradores e quatro diferenciadores.



Tabela 11 – Especificações de configuração dos componentes que compõem a interpolação.

<b>Componente</b>	<b>Parâmetro</b>	<b>Especificação</b>
Differentiator	Número de <i>bits</i>	20
	Profundidade	1
Integrator	Número de <i>bits</i>	20
	Profundidade	1
UP Sampling	Taxa de aumento	4
Multi-Rate DFF	Número de estágios <i>Pipeline</i>	1
	Nome do <i>Clock</i>	PLL_clk3 (6.25 ns)
AltBus	Número de <i>bits</i>	[15]:[3]
	Saturar saída	Ativado

### 3.8.5 Bloco Atraso Variável

O “Atraso Variável” foi implementado em oito canais paralelos (Figura 32) relacionados aos oito elementos da abertura sintética. Cada canal recebe os dados da Interpolação e tem entrada em um subsistema “Atraso\_Variavel\_[1 a 8]”, que também recebe os coeficientes de atraso, onde é processado o atraso variável no subsistema “Delay\_Block”. O sinal processado segue para a etapa de “Apodização”.

Com base no trabalho de Assef (2013), considerando a geometria do transdutor AT3C52B e focalização simétrica em relação ao eixo z com profundidade de 25 mm, foi traçado um perfil temporal de atraso utilizando a biblioteca de simulação de campo acústico Field II (JENSEN, 1996) no Matlab, resultando na Figura 33. Também são destacados os valores temporais para a abertura sintética de oito elementos.

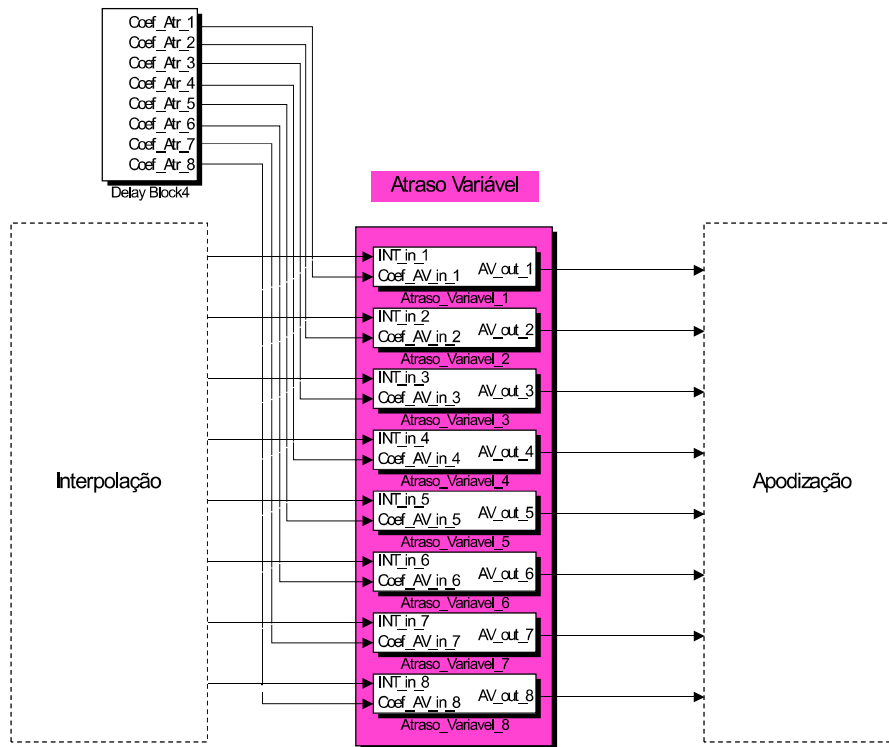


Figura 32 – Diagrama em blocos do Atraso Variável de oito canais com entrada dos ciclos de atraso.

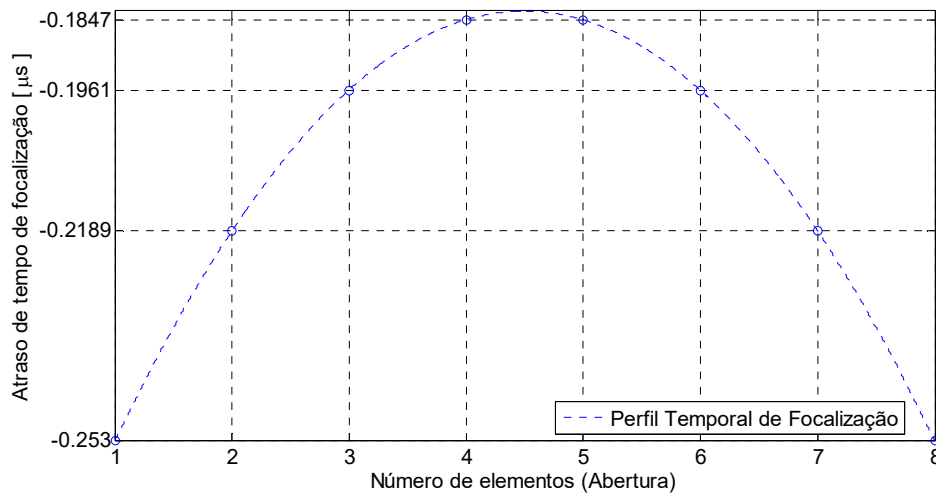


Figura 33 – Perfil temporal de focalização em 25 mm para abertura com 8 elementos do transdutor AT3C52B.

A partir dos tempos de atraso individuais de focalização, calculou-se uma aproximação em ciclos de *clock* da FPGA com resolução de 25 ns (Figura 34) e 6,25 ns (Figura 35).

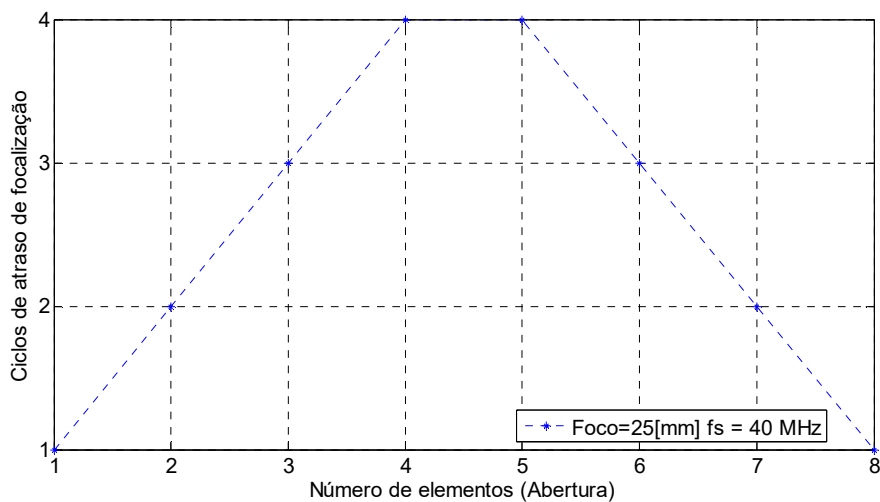


Figura 34 – Perfil de focalização com ciclos de *clock* de 40 MHz.

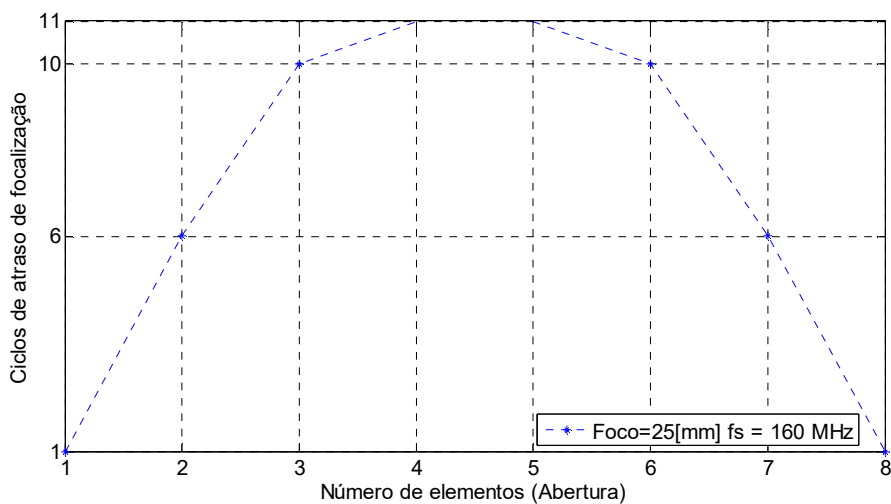


Figura 35 – Perfil de focalização com ciclos de *clock* de 160 MHz.

Na Figura 36 são apresentados os valores de ciclos de *clock* da FPGA, com resolução de 6,25 ns, atribuídos ao subsistema “Delay\_Block” que contém os blocos com os valores das constantes importadas da área de trabalho do Matlab. A Tabela 12 indica os parâmetros de configuração dos blocos do subsistema “Delay”.

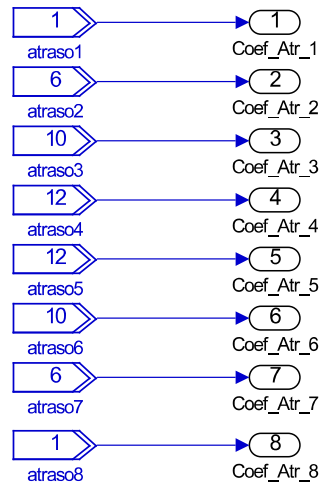


Figura 36 – Detalhe do subsistema “Delay\_Block” em ciclos de *clock*.

Tabela 12 – Especificações de configuração dos componentes do subsistema “Delay\_Block”.

Bloco	Especificação	Parâmetro
Constant	Valor	ciclos_atraso_interp_8e([1 a 8],1)

O modelo final é formado por oito blocos de atraso, sendo um para cada canal. Em cada bloco foi incluído um subsistema “Variavel\_Atraso”, e um componente “Dual-Port RAM”. Esta etapa utiliza memórias embarcadas para implementar registradores de deslocamento. Na Figura 37 é ilustrado o diagrama de blocos do subsistema “Atraso\_Variavel” e na Tabela 13 as especificações individuais de cada componente.

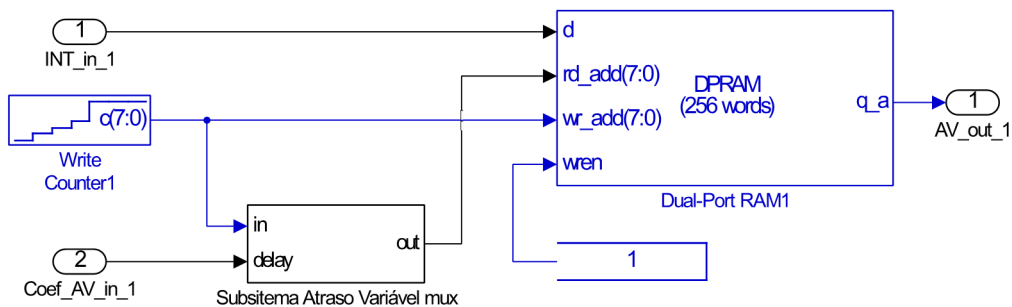


Figura 37 – Modelo do Subsistema “Atraso\_Variavel” implementado em memória com contador para gravação e atraso no contador para leitura da memória.

Tabela 13 – Especificações de configuração dos componentes da etapa “Atraso\_Variável”.

Bloco	Parâmetro	Especificação
Increment Decrement	Tipo do Barramento	Inteiro não sinalizado
	<i>Clock</i>	PLL_clk3 (6.25 ns)
	Sentido	Incremento
	Número de <i>Bits</i>	[8].[ ]
	Valor inicial	0
	Seleção de fase do <i>Clock</i>	1
Dual-Port RAM	Número de palavras	256
	Tipo do bloco de memória	M9K
	Seleção de fase do <i>Clock</i>	1
	Tipo de dados	Inferido
	Inicialização	Em Branco
VCC	Fornece um valor constante de um <i>bit</i> de	1
Atraso Variável	-	Subsistema

O subsistema “Atraso\_Variavel\_mux”, apresentado na Figura 38 de forma resumida com apenas 4 atrasos para melhor visualização, é responsável por selecionar o atraso aplicado ao sinal a partir dos sinais de seleção do multiplexador que recebe os valores das constantes de atraso em ciclos de *clock* (o sistema desenvolvido suporta 64 ciclos). Na Tabela 14 estão relacionados os parâmetros de configuração dos blocos que compõem o subsistema.

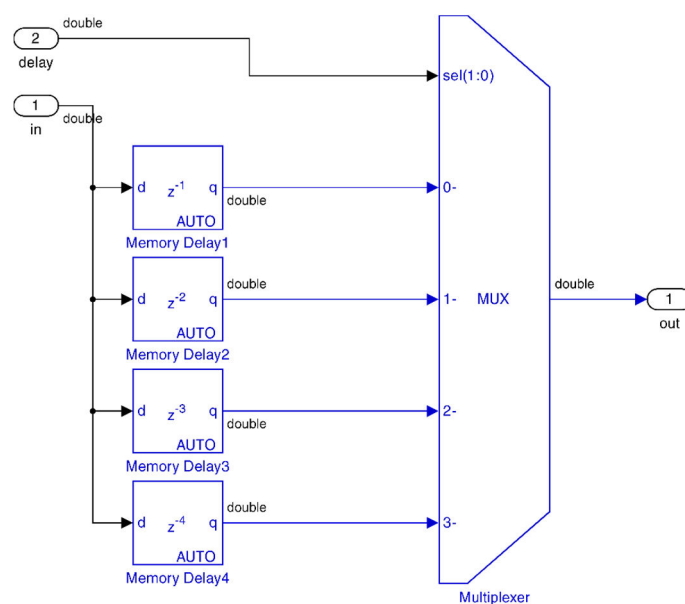
Figura 38 – Fragmento do subsistema “Atraso\_Variavel\_mux” para determina o atraso de *clock*.

Tabela 14 – Especificações de configuração dos componentes do subsistema “Atraso\_Variavel\_mux”.

Bloco	Parâmetro	Especificação
Atraso	Tipo dos Dados	Inferido
	Números de estágios do <i>pipeline</i>	1 a 63
	Tipo do bloco de memória	Automático
Multiplexador	Número de linha de entrada de dados	64
	Números de estágios do <i>pipeline</i>	0
	Permitir sobreposição de ponto flutuante para este bloco	(Ativado)

### 3.8.6 Bloco Apodização

O bloco “Apodização” foi implementado em oito canais paralelos (Figura 39), relacionados aos oito elementos da abertura sintética. Cada canal recebe os dados do “Atraso Variável” e tem entrada em um subsistema “Apodizacao\_[1 a 8]”, que também recebe os coeficientes de apodização (Figura 41). O sinal processado segue para a etapa de “Somatório Coerente”.

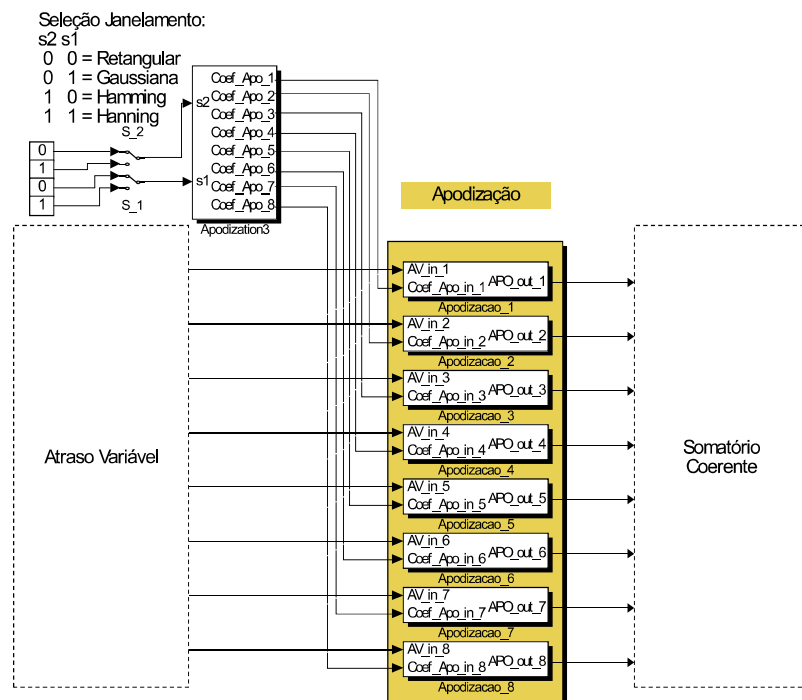


Figura 39 – Implementação da Apodização em oito canais com respectivos coeficientes.

O subsistema “Apodização” relaciona os coeficientes de apodização adequados aos janelamentos Retangular, Gaussiana, Hamming e Hanning, que são selecionados a partir de duas chaves seletoras (Figura 40). A primeira opção é com o janelamento retangular (chaves  $S_1 = 0$  e  $S_2 = 0$ ). Para a utilização de outros coeficientes do janelamento, como por exemplo, Gaussiana, altera-se a posição das chaves seletoras para  $S_1 = 1$  e  $S_2 = 0$ , atualizando-se os coeficientes do subsistema “Apodização” (Figura 41). Para isso, foram incluídos oito blocos multiplexadores. No caso do *kit* de FPGA, foram utilizadas duas chaves do tipo *slide*.

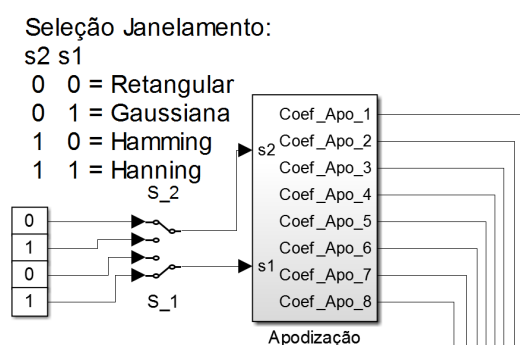


Figura 40 – Detalhe das chaves seletoras para escolha dos coeficientes de apodização.

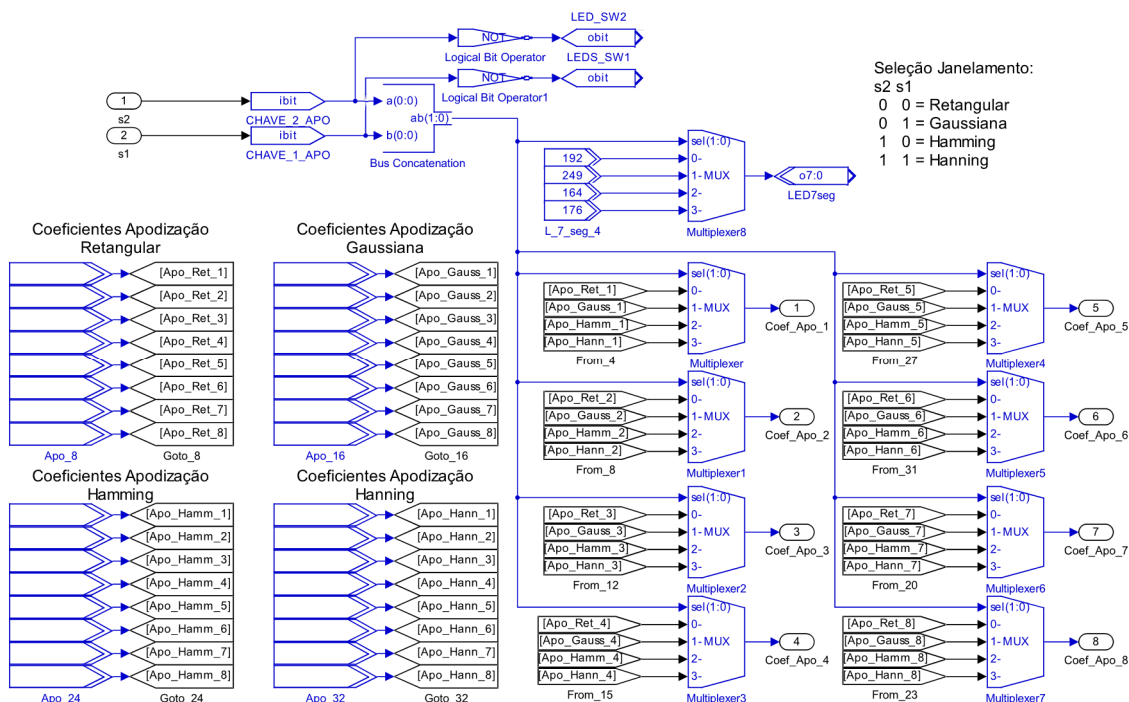


Figura 41 – Modelo do subsistema de seleção de apodização por multiplexadores

Os coeficientes das janelas de apodização para abertura de oito elementos estão relacionados na Tabela 15 com suas respectivas funções em Matlab, assim como a posição das chaves seletoras.

Tabela 15 – Posição das chaves seletoras para seleção dos coeficientes para as técnicas de apodização Retangular, Gaussiana, Hamming e Hanning com abertura de oito elementos.

Chave Seletora		Janela	Função no Matlab
S2	S1		
0	0	Retangular	$w = \mathit{ones}(1, 8);$
0	1	Gaussiana	$w = \mathit{gausswin}(8);$
1	0	Hamming	$w = \mathit{hamming}(8);$
1	1	Hanning	$w = \mathit{hann}(8, 'Symmetric');$

A partir dos dados calculados pelas funções no Matlab, é apresentado na Figura 42 o comparativo entre os perfis dos coeficientes normalizados das janelas de ponderação para abertura de oito elementos ativos.

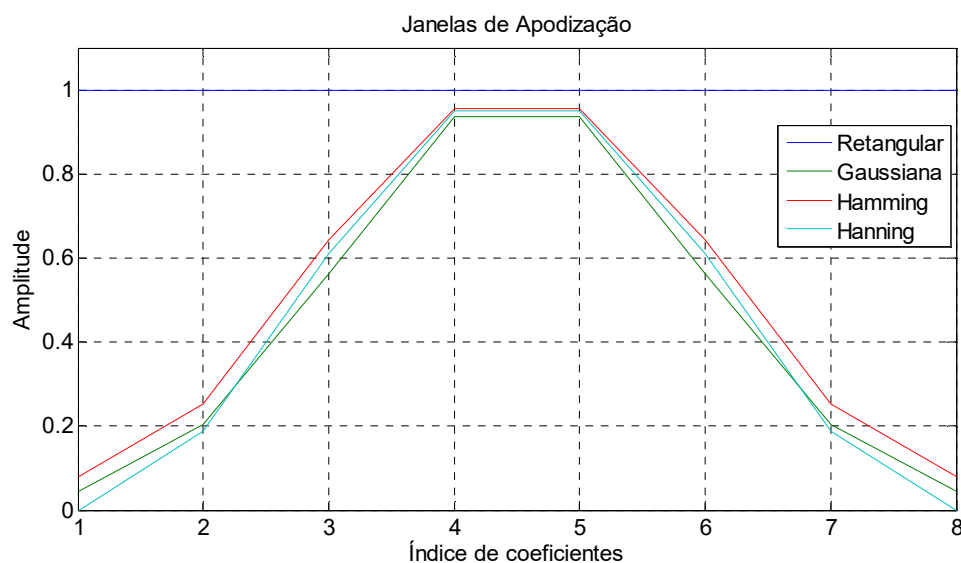


Figura 42 – Resultados simulados para as técnicas de apodização Retangular, Gaussiana, Hamming e Hanning para abertura de oito elementos ativos.

### 3.8.6.1 Subsistema Apodização referente a cada canal

O subsistema “Apodizacao” realiza a multiplicação do sinal de entrada pelo coeficiente da janela de apodização selecionada, referente a cada canal.



Na Figura 43 está representado o modelo do subsistema com entrada do sinal (AV\_in\_[1 a 8]), entrada do coeficiente de apodização (Coef\_Apo\_in\_[1 a 8]) e saída (APO\_out\_[1 a 8]) com o resultado da multiplicação. A especificação interna de cada item que compõe a “Apodizacao” encontra-se na Tabela 16.

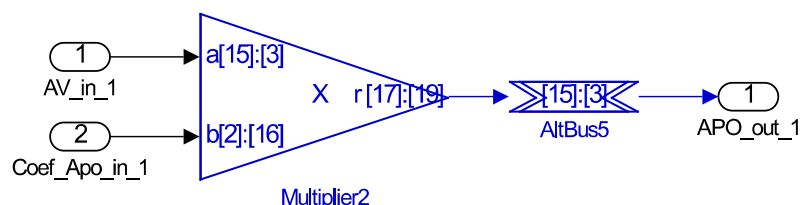


Figura 43 – Modelo do subsistema “Apodizacao” implementado com multiplicador.

Tabela 16 – Especificações de configuração dos componentes do subsistema “Apodizacao”.

Bloco	Parâmetro	Especificação
	Tipo de barramento	Fracionário Sinalizado
Multiplier	Entrada Número de <i>bits</i>	[15] : [3]
	Número de estágio <i>Pipeline</i>	1

### 3.8.7 Bloco Somatório Coerente

O bloco “Somatório Coerente” foi implementado em oito canais paralelos (Figura 44), relacionados aos oito elementos da abertura sintética. Todos os canais recebem dados do bloco “Apodização”, têm entradas no subsistema “Somatorio\_Coerente” e uma saída que segue para o bloco “Decimação”. O detalhe interno do subsistema está indicado na Figura 45. Foi incluído um bloco de ganho (“Gain”) de fator 0,125 (1/8) para condicionar a saída do bloco em 15 *bits* na parte inteira e 3 *bits* fracionários.

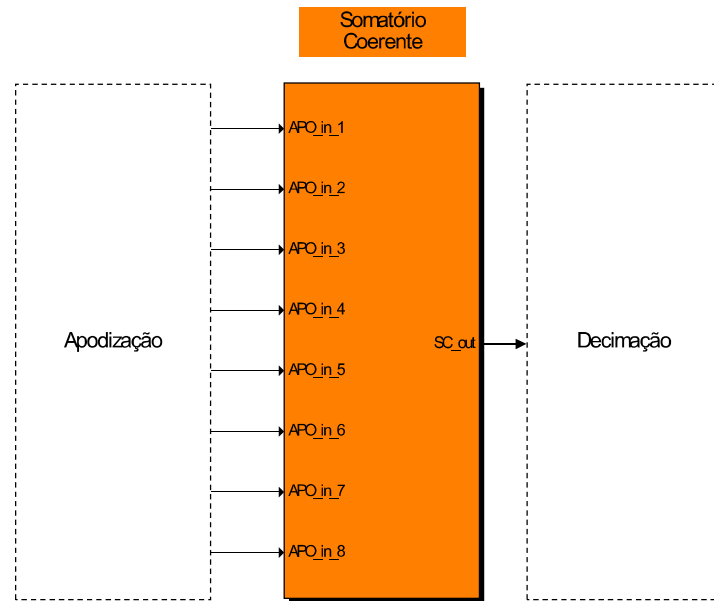


Figura 44 – Implementação do bloco “Somatório Coerente” de oito canais.

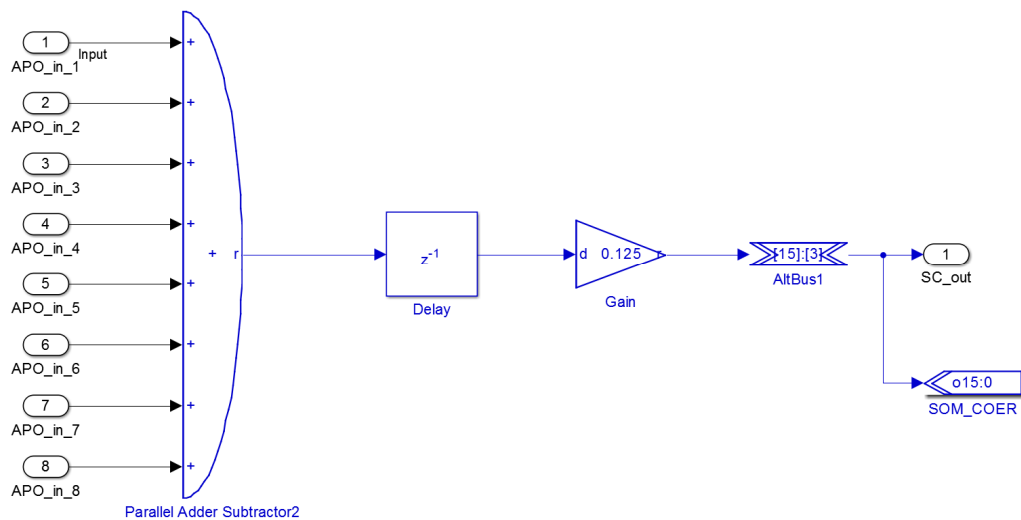


Figura 45 – Detalhe interno do modelo do subsistema “Somatorio\_Coerente” implementado.

### 3.8.8 Bloco Decimação

O bloco “Decimação” foi implementado em um subsistema (Figura 46) que recebe o sinal do bloco “Somatório Coerente”, com uma saída para o bloco “Demodulação”. O detalhe interno do subsistema está indicado na Figura 47 e os parâmetros de configuração são apresentados na Tabela 17.

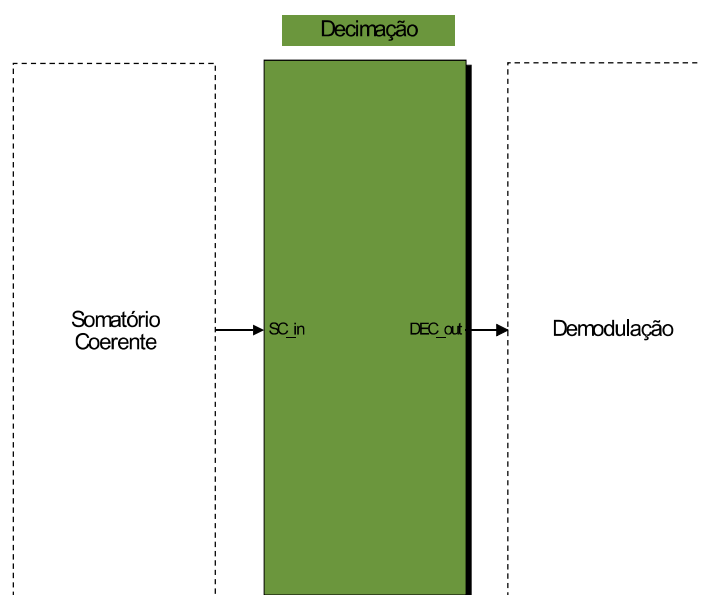


Figura 46 – Implementação do Bloco subsistema “Decimação”.

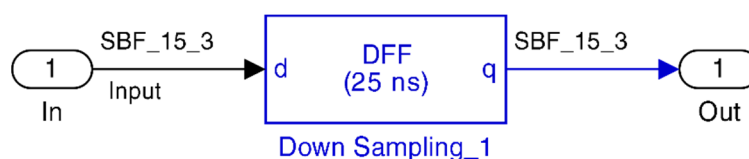


Figura 47 – Modelo do bloco de “Decimação” implementado com *flip-flop*.

Tabela 17 – Especificações de configuração dos componentes da etapa de “Decimação”.

Bloco	Parâmetro	Especificação
	Número de estágios <i>Pipeline</i>	1
Multi-taxa DFF	Nome do <i>Clock</i>	PLL_clk1 (25 ns)
	Tipo do Bloco de Memória	Automático

### 3.8.9 Bloco Demodulação

O bloco “Demodulação” foi implementada em um subsistema (Figura 48) que recebe o sinal do bloco “Decimação”, com uma saída para “Compressão Logarítmica”. O detalhe deste bloco é apresentado na Figura 49 indicando a separação em dois blocos: “Transformada\_Hilbert” e “Detecção\_de\_Envelope”.

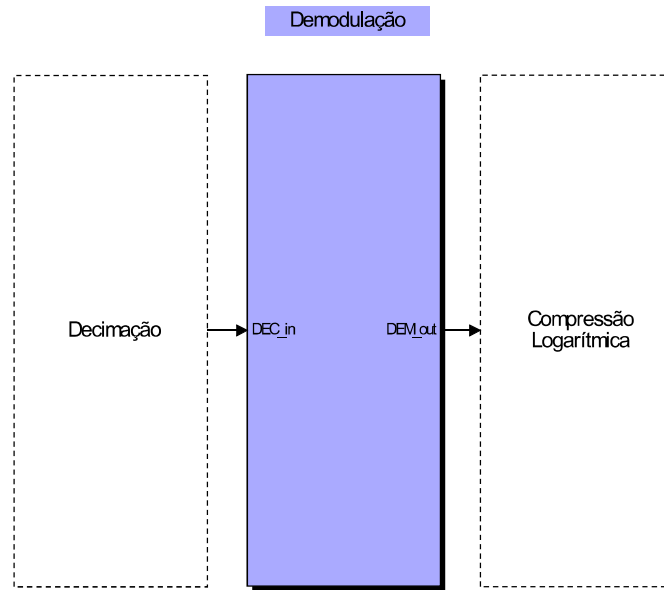


Figura 48 – Bloco subsistema “Demodulação”.

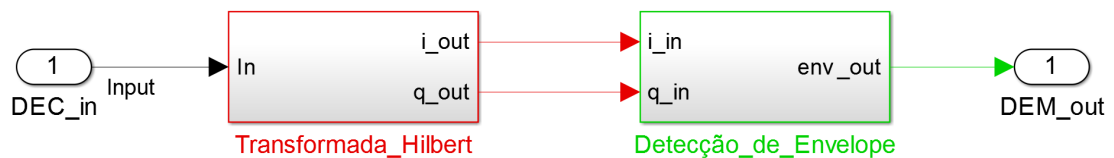


Figura 49 – Modelo do subsistema “Demodulação” com os blocos “Transformada\_Hilbert” e “Detecção\_de\_Envelope”.

O método proposto para o subsistema “Transformada\_Hilbert” (Figura 50) explora tanto a simetria ímpar negativa como os coeficientes de valor zero alternados na resposta ao impulso do filtro FIR com 33 coeficientes. Desta forma, são gerados os componentes  $I$  e  $Q$ .

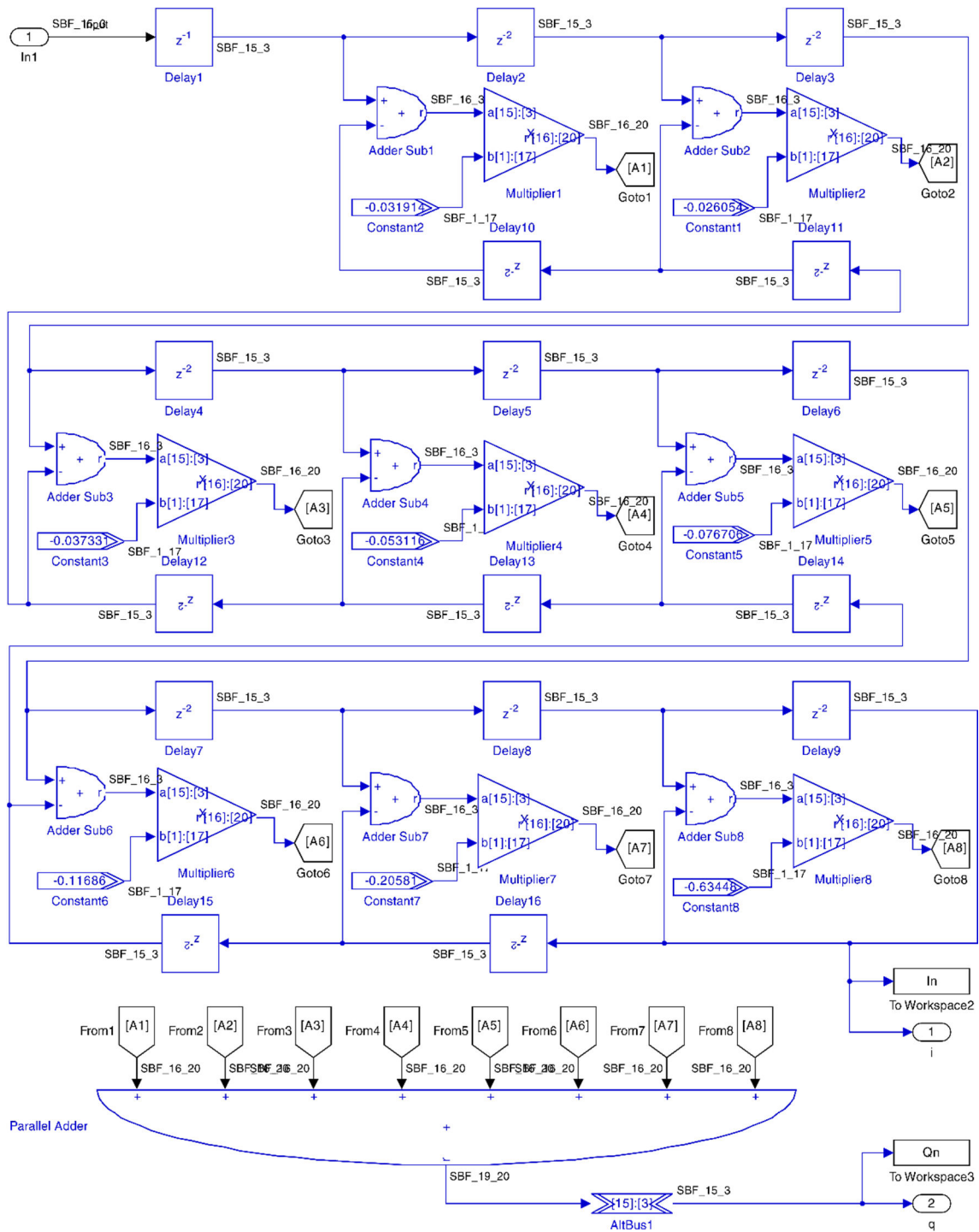


Figura 50 – Modelo do subsistema “Transformada\_Hilbert” implementado utilizando simetria de coeficientes, bem como os coeficientes de valores nulos.

Após a Transformada de Hilbert, os sinais  $I$  e  $Q$  foram elevados ao quadrado e somados utilizando o bloco “Multiply\_Add”. O resultado da operação foi otimizado em 32 *bits* e aplicado ao bloco “Square Root” para computação do

envelope do sinal (Figura 51). Na Tabela 18 estão relacionados os parâmetros de configuração do subsistema “Detecção\_de\_Envelope”.

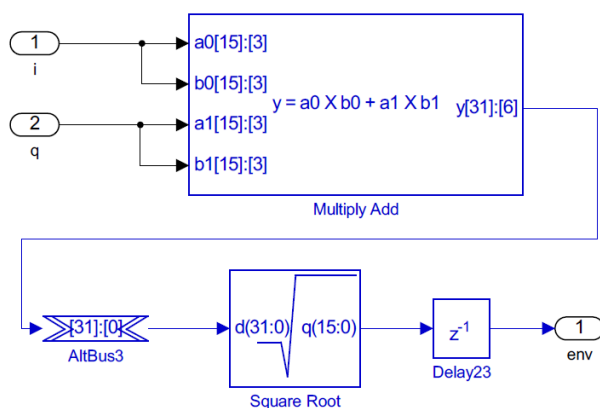


Figura 51 – Detalhe interno do modelo do subsistema “Detecção de Envelope” implementado.

Tabela 18 – Especificações de configuração dos componentes da etapa de “Detecção de Envelope”.

Bloco	Parâmetro	Especificação
MultiplyAdd	Número de multiplicadores (2 - 4)	2
	Tipo de Barramento	Fracionário sinalizado
	Entrada Número de <i>bits</i>	[15] : [3]
	Modo adicionador	add add
	Registro <i>pipeline</i>	Entradas multiplicador e adicionador
	Usar circuitos dedicados	Selecionado
Square Root	Número de entrada ( <i>bits</i> )	32
	Número de estágio <i>pipeline</i>	0
Delay	Número de atrasos <i>pipeline</i>	1
	Seleção fase de <i>clock</i>	1

### 3.8.10 Bloco Compressão Logarítmica

O bloco “Compressão Logarítmica” foi implementado em um subsistema que recebe o sinal do bloco “Demodulação” e fornece uma saída para o processamento de conversão de varredura que é realizado através de algoritmo no Matlab, conforme mostrado na Figura 52.

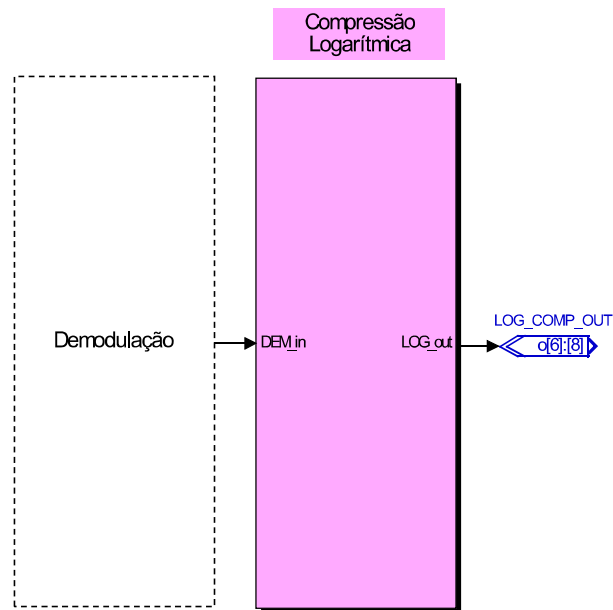


Figura 52 – Implementação do Bloco subsistema “Compressão Logarítmica”.

A resolução da tabela de conversão logarítmica foi definida em 11 *bits* (0 a 2047 pontos) devido à análise de amplitude do sinal de entrada. A curva de normalização está indicada na Figura 53. Após o cálculo, o valor é convertido em Decibéis (dB), utilizando a Equação 9:

$$x_{dB}(n) = 20 \cdot \log_{10} \left( \frac{x(n)}{x_{max}} \right), \quad (9)$$

onde  $x(n)$  é o sinal de entrada e  $x_{max}$  foi considerado como o valor máximo da tabela logarítmica calculada anteriormente (2047). O projeto apresenta uma flexibilidade em escolher a faixa dinâmica entre dois valores: -30 dB ou -50 dB. A Figura 54 apresenta a curva de normalização em dB, limitada em -30, utilizada para validação do projeto.

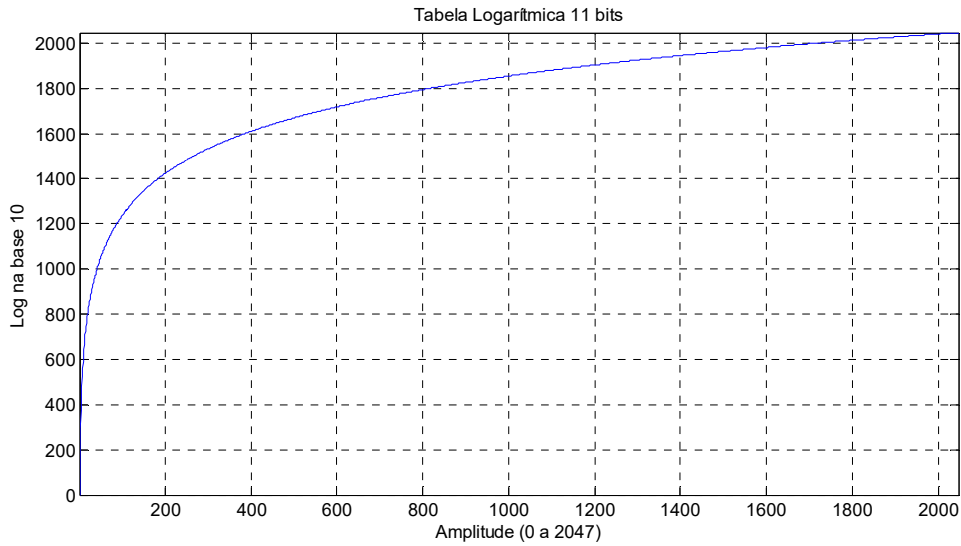


Figura 53 – Tabela logarítmica com 2048 endereços (11 bits).

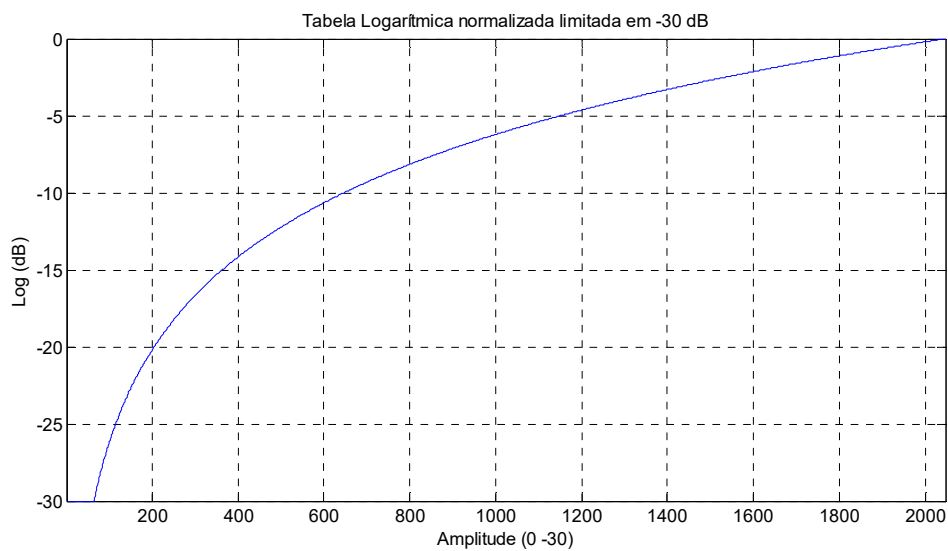


Figura 54 – Tabela logarítmica normalizada com 2048 endereços e faixa dinâmica de -30 dB.

Na Figura 55 está indicado o diagrama em blocos do subsistema “Compressão Logarítmica” e na Tabela 19 os parâmetros de configuração dos blocos utilizados.



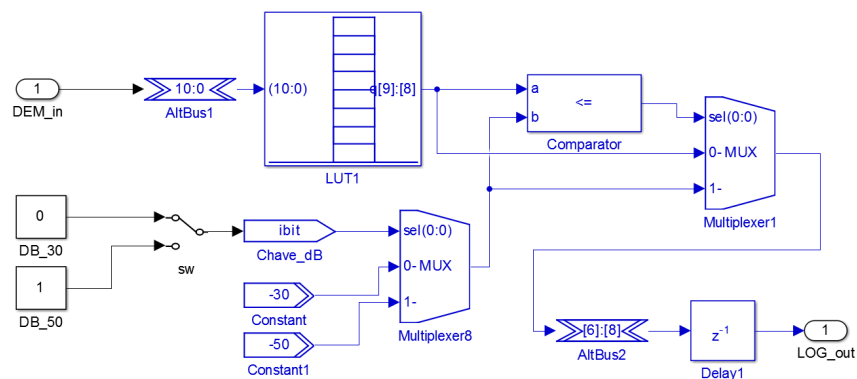


Figura 55 – Modelo do subsistema “Compressão Logarítmica” implementado utilizando LUT e chave de seleção.

Tabela 19 – Especificações de configuração dos componentes da etapa de “Compressão Logarítmica”.

Bloco	Parâmetro	Especificação
LUT	Largura do endereço	11
	Número de <i>bits</i>	[9] : [8]
	Matriz Matlab	Tabel_log (área de trabalho)
	Tipo de Bloco de Memória	AUTO
Comparador	Operador	$a \leq b$
AltBus 1	Número de <i>bits</i>	[11] : [0] (Saturado)
AltBus 1	Número de <i>bits</i>	[6] : [8] (Saturado)

## 4 RESULTADOS

Neste capítulo são apresentados os resultados quantitativos e qualitativos obtidos através de simulações e de forma experimental para verificação e validação de cada módulo DSP que compõe o sistema proposto, além do resultado final do processamento para reconstrução de imagem por US.

### 4.1 RESULTADO INDIVIDUAL DAS ETAPAS DE PROCESSAMENTO

#### 4.1.1 Implementação do filtro digital FIR

A validação do projeto do filtro digital FIR resultou em um artigo publicado no Congresso Brasileiro de Engenharia Biomédica, CBEB 2016 (ASSEF et al., 2016).

Na Figura 56, apresentam-se as respostas sobrepostas ao impulso do filtro FIR gerado pelo FDATool e pelo modelo implementando neste trabalho. A verificação gráfica foi realizada no Matlab com dados importados do ambiente Simulink. Para isso, a entrada de dados dos multiplicadores foi alterada para base 1s17 (um *bit* na parte inteira e 17 *bits* fracionários).

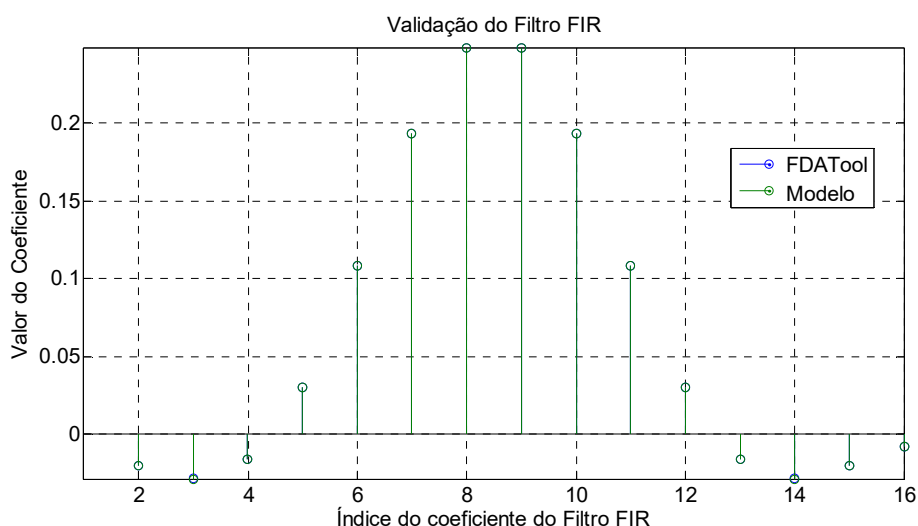


Figura 56 – Validação do filtro FIR através da resposta ao impulso.

A avaliação quantitativa do bloco “Filtro Digital FIR” foi realizada através da aplicação da função de custo NRMSE, comparando-se os sinais processados pelo modelo e pelo bloco gerado automaticamente pelo FDATool e exportado para o Simulink. Após o processamento, ambas as respostas foram exportadas para a área de trabalho do Matlab, sendo computado a NRMSE. A Figura 57 apresenta o comparativo dos sinais gerados e sobrepostos com NRMSE igual a 0,03%.

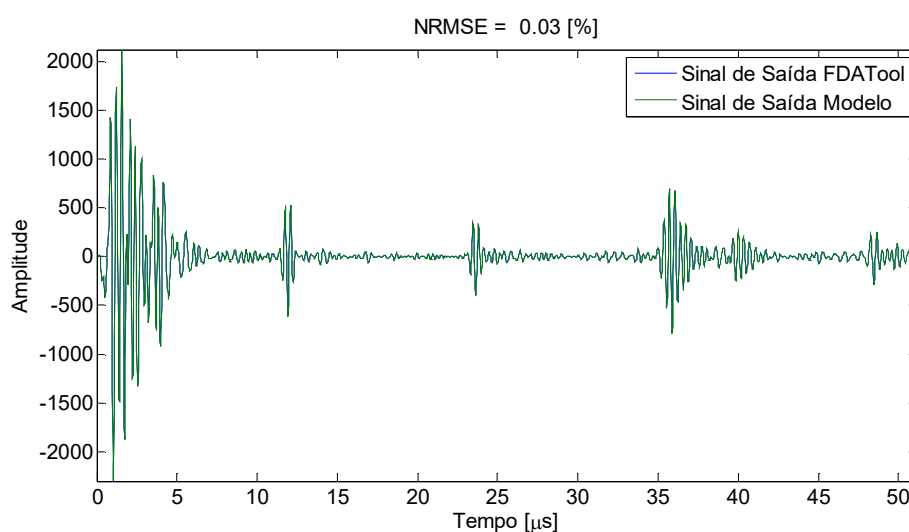


Figura 57 – Comparação entre o sinal filtrado pelo modelo “Filtro Digital FIR” e pelo bloco gerado pelo FDATool.

Apresentam-se também os resultados das simulações de dois sinais com frequência de amostragem de 40 MHz no Simulink, além da implementação prática para demonstração da viabilidade do filtro proposto para aplicações de imageamento por US. O primeiro sinal (Figura 58) representa uma senoide com frequência de 3,2 MHz somada a uma senoide com frequência de 8 MHz e amplitude de 50% do sinal original, para simulação de ruído. Na Figura 58(a) é apresentado o sinal resultante com ruído de alta frequência. Na Figura 58(b) e Figura 58(c) são mostradas, respectivamente, as respostas geradas pelo processamento no Simulink e adquirido da FPGA. O segundo sinal (Figura 59) é formada pela resposta do primeiro elemento da linha de varredura 61 somada ao mesmo sinal de ruído. Na Figura 59(a) é apresentado o sinal resultante da soma com ruído de alta frequência. Na Figura 59(b) e Figura 59(c) são mostradas as mesmas respostas da Figura 58, porém para o sinal real de US.

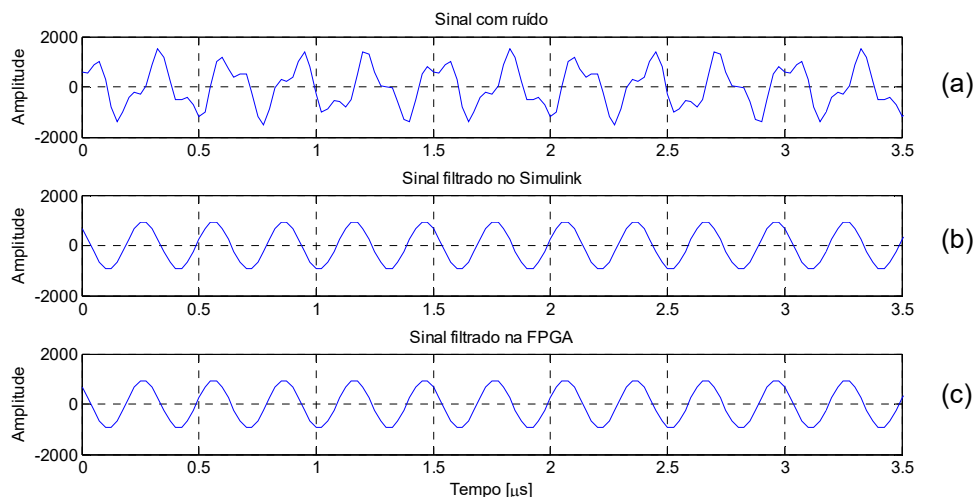


Figura 58 – Comparação de filtragem do sinal senoidal com ruído. (a) Sinal de entrada com ruído. (b) Sinal processado no Simulink. (c) Sinal processado na FPGA

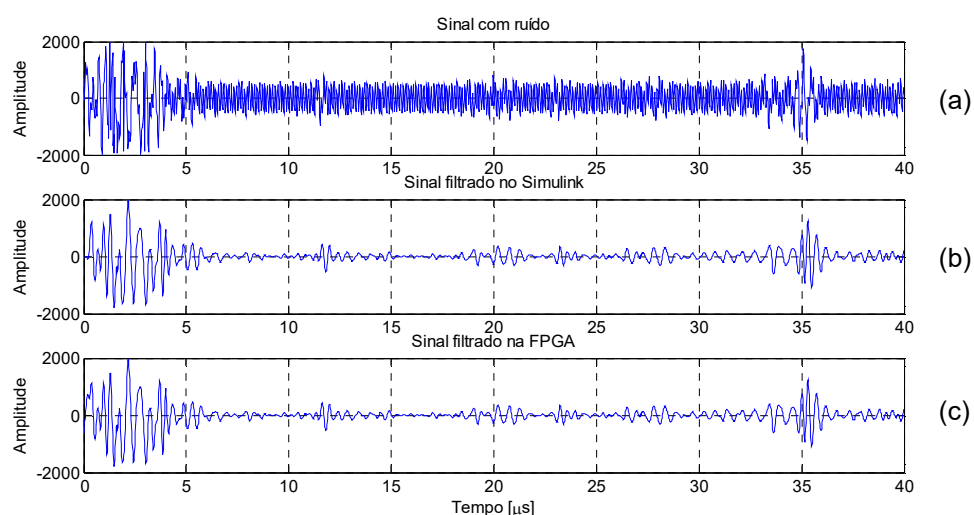


Figura 59 – Comparação de filtragem do sinal de US real com ruído. (a) Sinal de entrada com ruído. (b) Sinal processado no Simulink. (c) Sinal processado na FPGA.

Através dos resultados experimentais foi comprovada de forma qualitativa e quantitativa o desempenho coerente do sistema, conforme as FFTs apresentadas na Figura 60 e Figura 61. Na Figura 60 são apresentadas as FFTs do sinal senoidal com ruído antes (Figura 60(a)) e após (Figura 60(b)) o processamento do filtro na FPGA. Também são apresentadas na Figura 61(a) e Figura 61(b) as mesmas análises para o sinal real de US. Pode ser observado que na Figura 60 o componente de 8 MHz foi atenuado em mais de 55 dB após a filtragem, enquanto que, na Figura 61, em aproximadamente 40 dB.

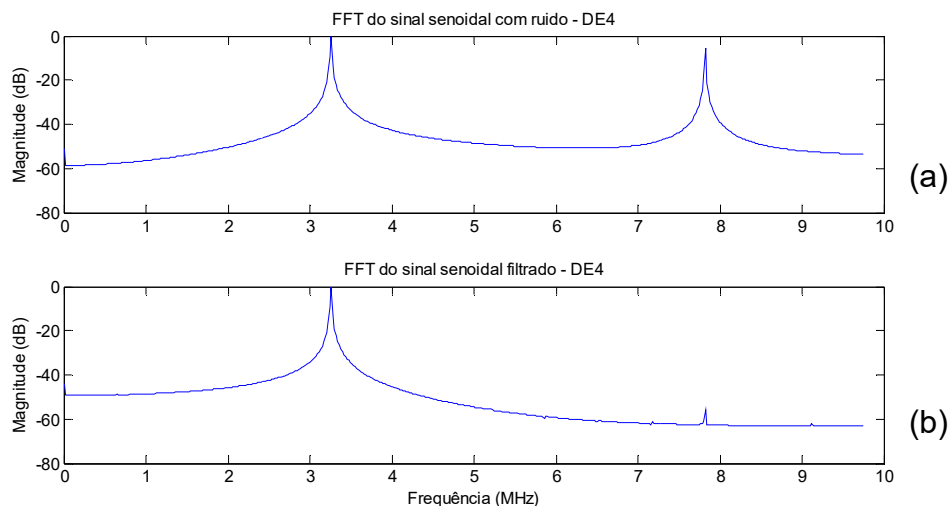


Figura 60 – Espectros de frequência do sinal senoidal de teste adquiridos da placa DE4-230. (a) FFT do sinal com ruído. (b) FFT do sinal filtrado.

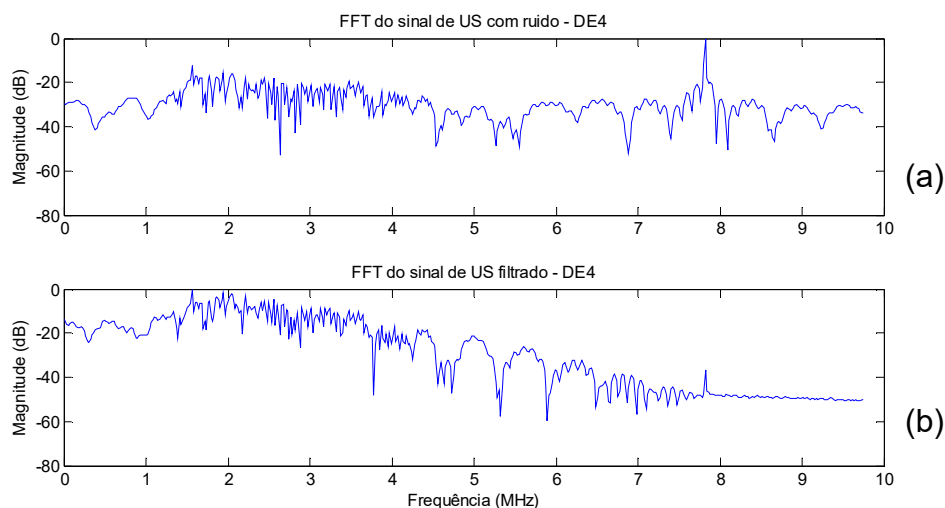


Figura 61 – Espectros de frequência do sinal de US real adquiridos da placa DE4-230. (a) FFT do sinal com ruído. (b) FFT do sinal filtrado.

#### 4.1.2 Interpolação CIC

Na Figura 62 pode-se observar a semelhança dos sinais antes (Figura 62(a)) e após (Figura 62(b)) a interpolação com fator 4, sendo que o primeiro possui 2046 amostras ( $F_s = 40$  MHz) e o segundo 8184 amostras ( $F_s = 160$  MHz).

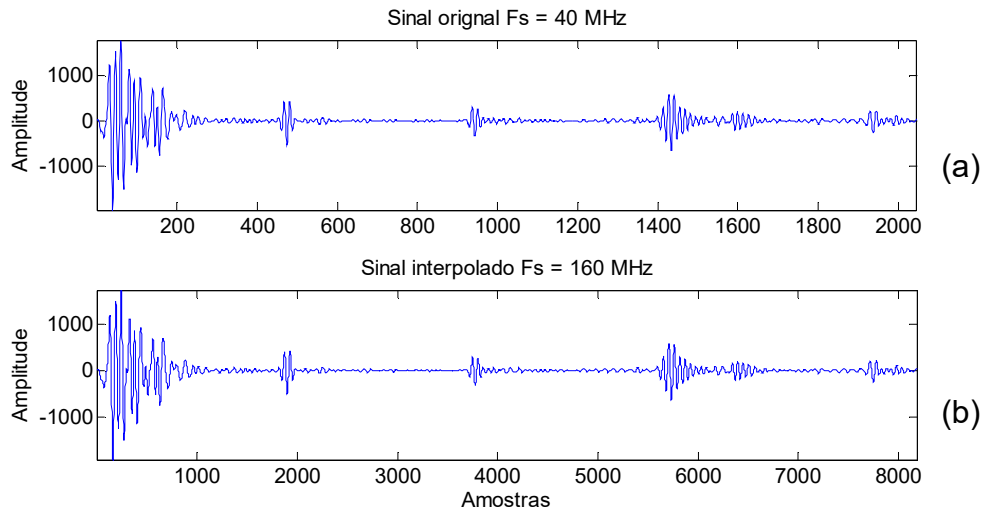


Figura 62 – Exemplo do resultado da interpolação. (a) Sinal entrada do bloco. (b) Sinal de saída gerado.

Na Figura 63 apresenta-se a comparação dos sinais de entrada e saída do bloco de interpolação processado no Simulink. Pode-se observar os degraus causados pela amostragem do sinal original de 40 MHz para 160 MHz. Destaca-se também, o atraso de 4 ciclos de *clock* (40 MHz) devido ao tempo de processamento do bloco de interpolação.

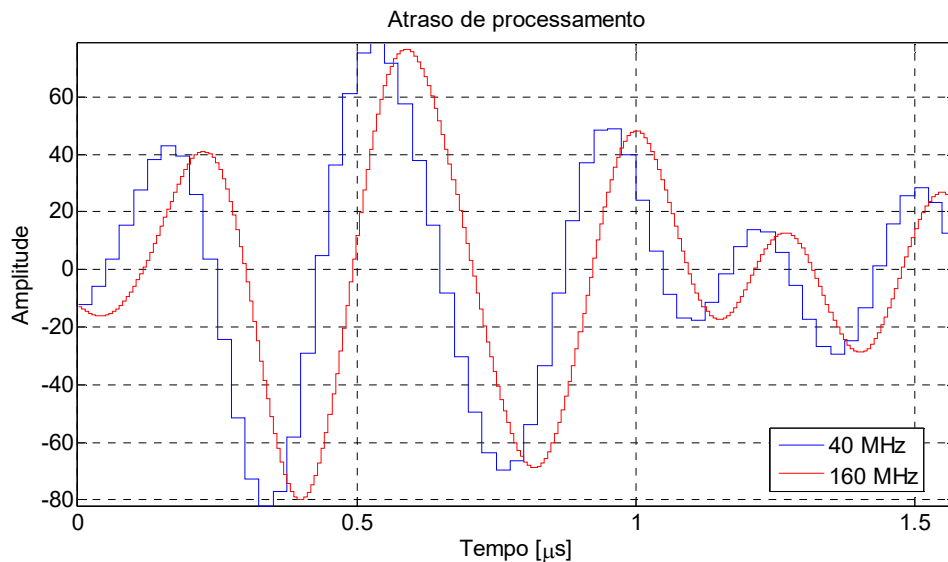


Figura 63 – Comparação entre o sinal original (40 MHz) e o sinal interpolado (160 MHz) para visualização do tempo de atraso de processamento.

Através dos resultados experimentais foi comprovada de forma gráfica o desempenho coerente do processamento do bloco de interpolação, conforme as FFTs apresentadas na Figura 64. Na Figura 64(a) é apresentada a FFT do sinal de US (*scanline* 61) original amostrado em 40 MHz e na Figura 64(b) o sinal após o processamento do bloco de interpolação no Simulink.

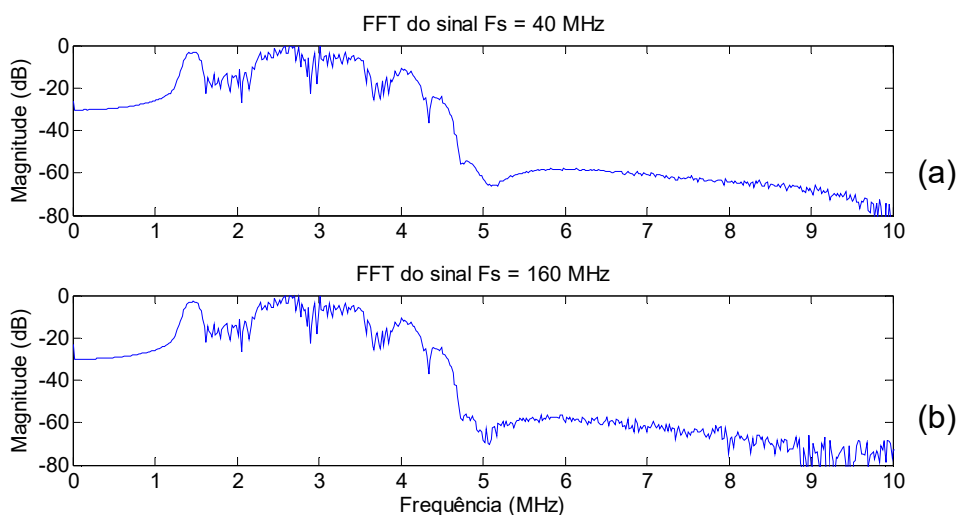


Figura 64 – Comparação dos espectros de frequência do sinal de US *scanline* 61. (a) FFT do sinal original ( $F_s = 40$  MHz). (b) FFT do sinal interpolado ( $F_s = 160$  MHz).

#### 4.1.3 Atraso variável

A Figura 65 mostra o efeito do atraso variável em um sinal com 512 amostras (Figura 65(a)) e os contadores (Figura 65(b)) utilizados como seletor da memória RAM, responsável pelo atraso. Foi escolhido, para comparação, o atraso variável de 12 ciclos de *clock* ( $F_s = 160$  MHz), realizado pela etapa de “Atraso Variável” no ambiente Simulink.

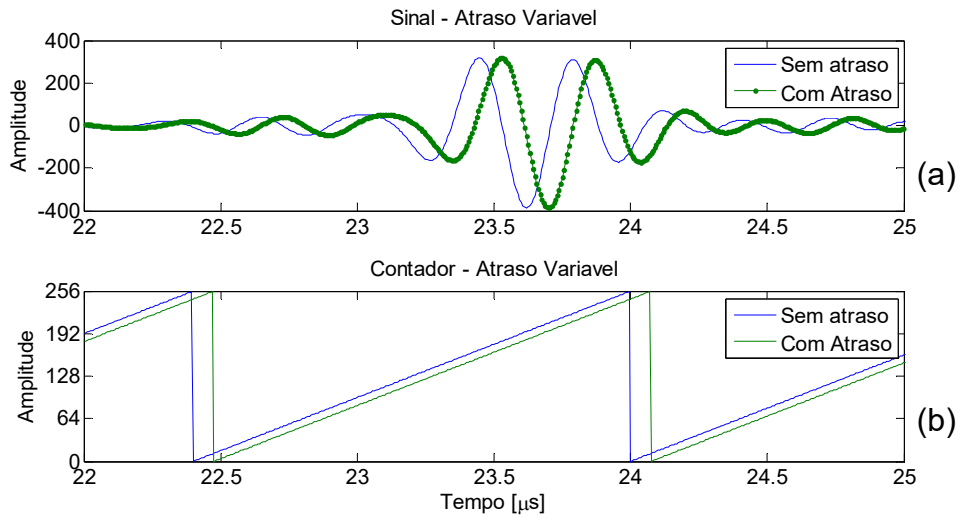


Figura 65 – Comparação dos sinais antes e após o bloco “Atraso Variável”. (a) Sinais de US antes e após o bloco “Atraso Variável”. (b) Contador de referência (sem atraso) e contador após atraso de 12 ciclos.

Para justificar o uso da técnica de interpolação para melhorar a focalização na recepção, o resultado da comparação da porcentagem do desvio do perfil de atraso em relação à aproximação em ciclos de *clock* de 40 MHz e 160 MHz está indicado na Figura 66.

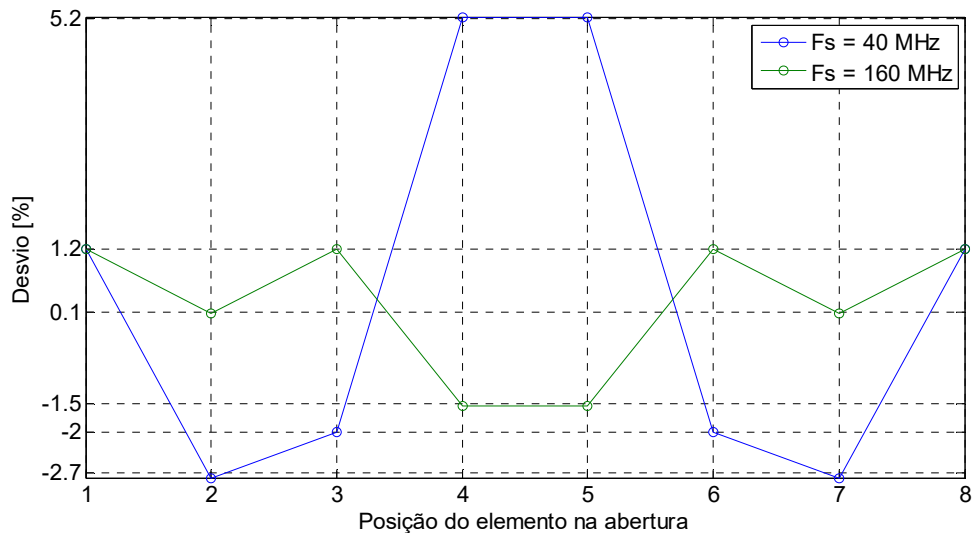


Figura 66 – Comparação entre os desvios calculados para  $F_s$  de 40 MHz e 160 MHz.



#### 4.1.4 Apodização

Na Figura 67 é apresentado o comparativo entre os sinais de entrada (lado esquerdo) e dos sinais apodizados e atenuados (lado direito) referente à abertura de oito elementos. Foi elaborado um padrão de cores para destacar a simetria da janela de apodização Gaussiana.

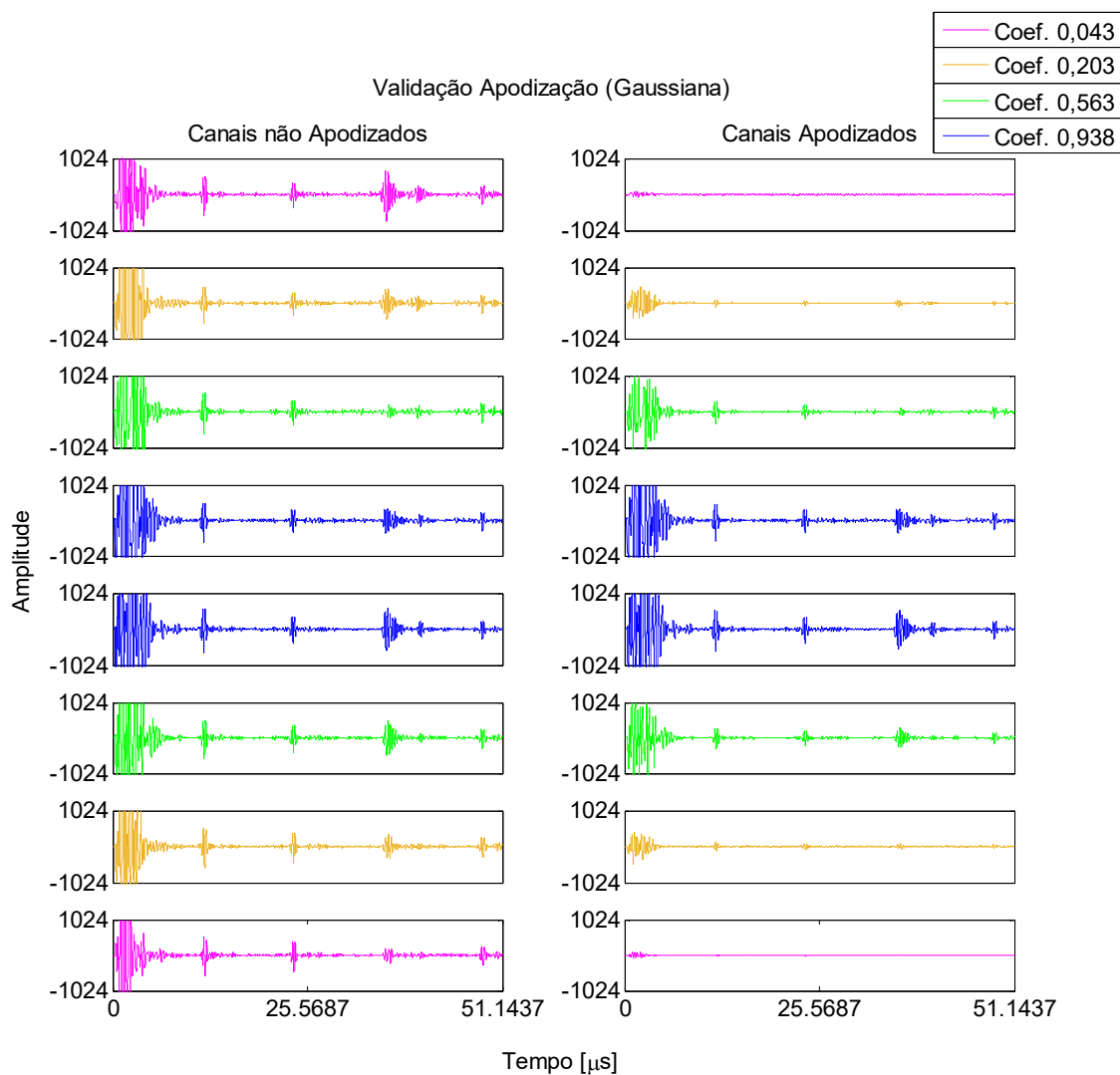


Figura 67 – Exemplo de apodização Gaussiana referentes à abertura de oito elementos.

#### 4.1.5 Somatório

Na Figura 68 são apresentadas as respostas de oito linhas de RF (61 a 68) com apodização Gaussiana (Figura 68 (a)) e o resultado do processamento

do bloco “Somatório Coerente” (Figura 68 (b)). Adicionalmente, na Figura 69 compara-se dois sinais resultantes do bloco “Somatório Coerente” utilizando os janelamentos Retangular e Gaussiana. Na Figura 69(a) são mostrados os sinais computados pelo bloco e na Figura 69(b) é mostrada a ampliação destes sinais para melhor visualização.

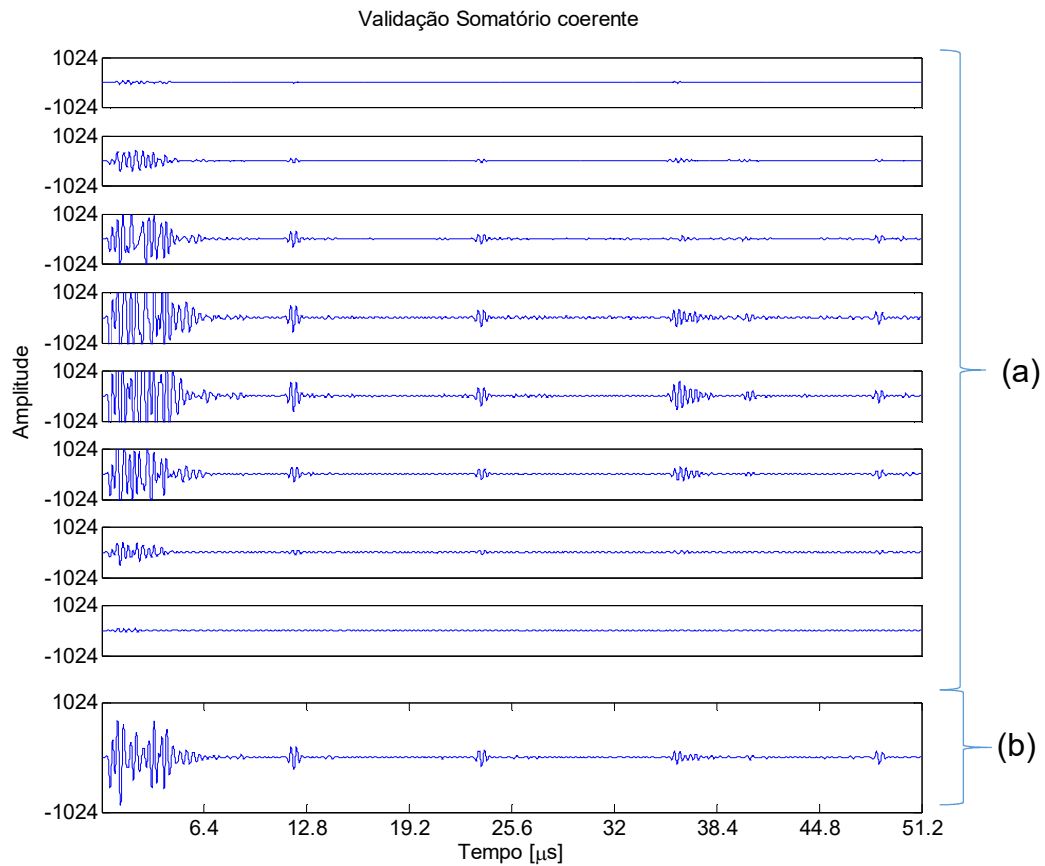


Figura 68 – Exemplo de sinal obtido pelo processamento do bloco “Somatório Coerente”.

(a) Resposta de oito linhas de RF (61 a 68) com apodização Gaussiana. (b) Linha de varredura resultante do somatório coerente.

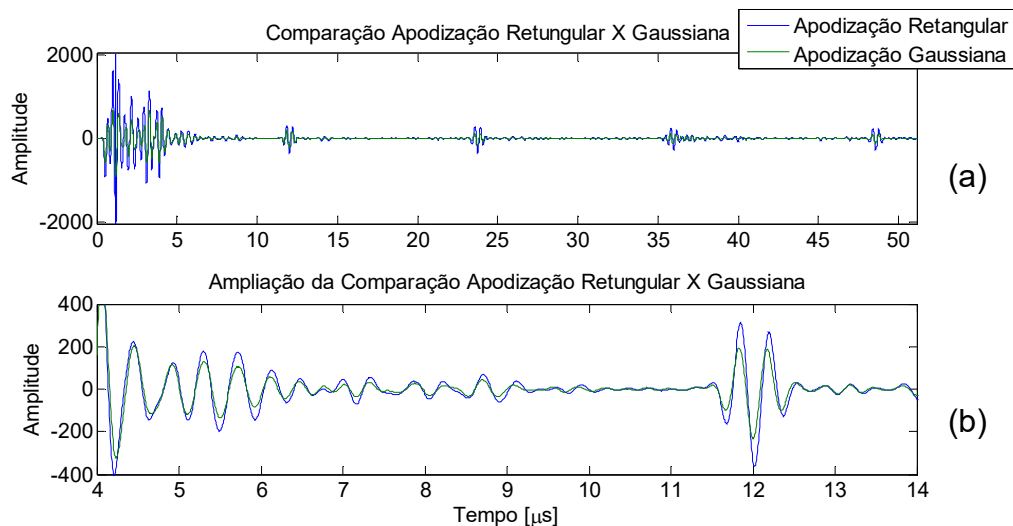


Figura 69 – Comparação de duas linhas de varredura geradas com diferentes apodizações. (a) Linhas de varredura geradas a partir de sinais processados com apodização Retangular e Gaussiana. (b) Ampliação do ítem (a) para melhor visualização.

#### 4.1.6 Decimação

De forma similar à interpolação, na Figura 70 são apresentados os sinais de entrada do bloco de decimação ( $F_s = 160$  MHz) e de saída decimado com fator 4 ( $F_s = 40$  MHz). Percebe-se, através do comparativo, os degraus causados pela amostragem do sinal original de 160 MHz para 40 MHz. Destaca-se também o atraso de dois ciclos de *clock* (40 MHz) devido ao tempo de processamento do bloco de decimação.

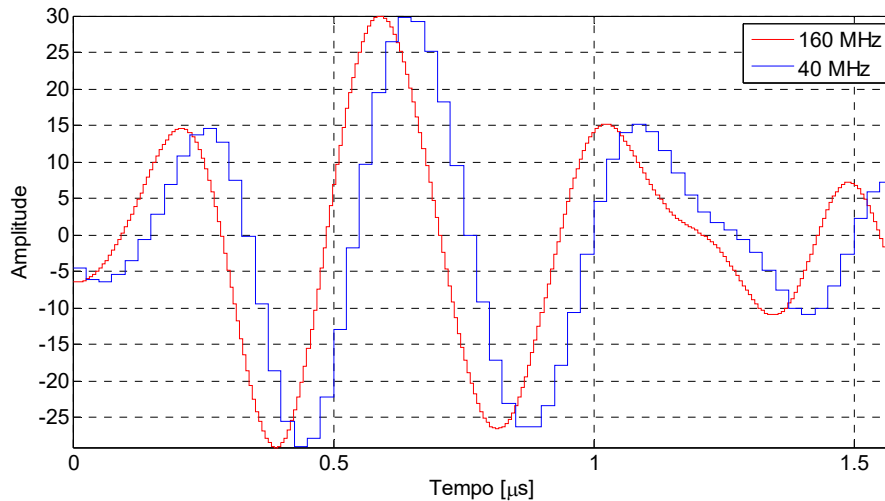


Figura 70 – Comparativo entre os sinais de entrada e saída do bloco “Decimação”.

#### 4.1.7 Demodulação

Na Figura 71 são apresentadas as repostas sobrepostas ao impulso do filtro da Transformada de Hilbert gerado pelo FDATool, assim como do modelo implementando no trabalho. A análise por semelhança foi realizada em ambiente Matlab com dados importados do Simulink.

Na Figura 72 estão indicados todos os sinais envolvidos na detecção de envoltória do sinal de US entre 10 e 14  $\mu\text{s}$ , para melhor visualização: Sinal de entrada,  $I$ ,  $Q$  e envelope processado.

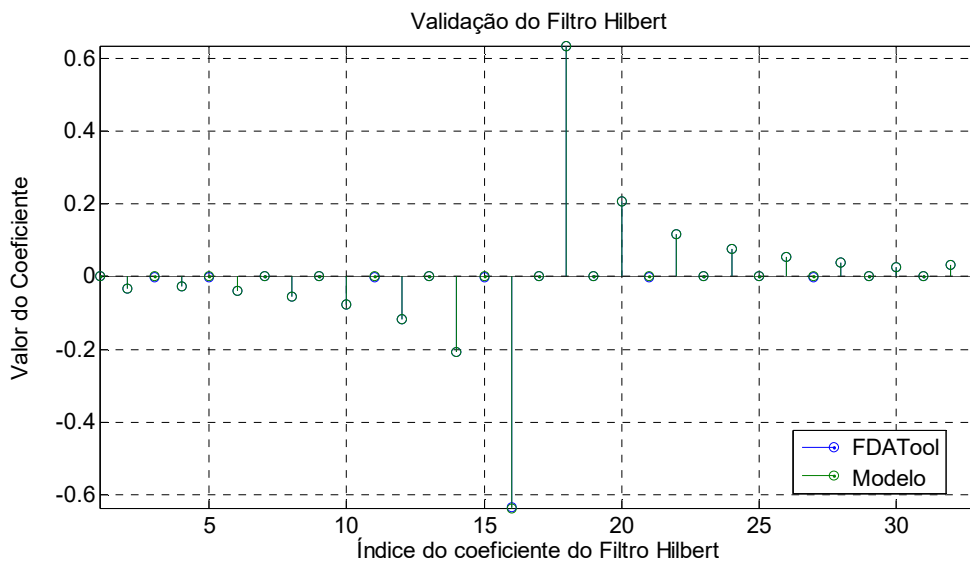


Figura 71 – Comparação da resposta ao impulso da Transformada de Hilbert baseada em filtro FIR entre o FDATool e modelo implementado.

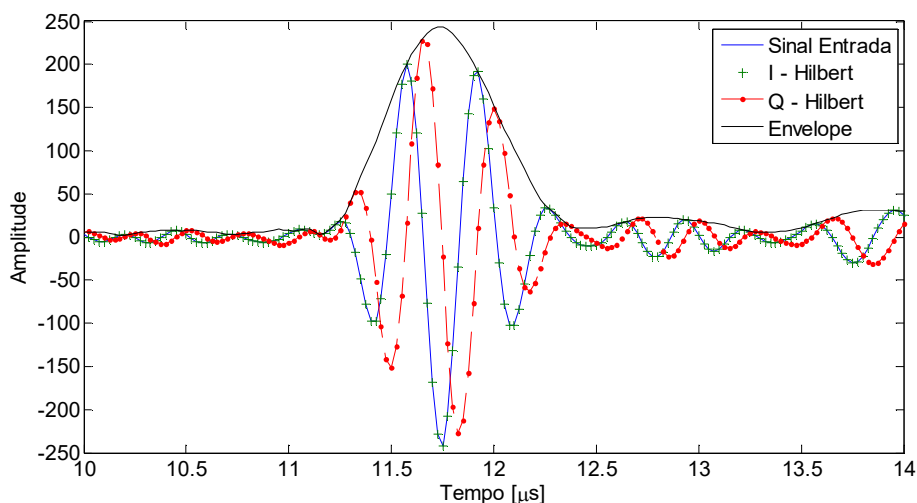


Figura 72 – Exemplo de sinais envolvidos na detecção de envoltória gerados pelo modelo proposto.

O resultado da detecção de envoltória é mostrado na Figura 73. A Figura 73(a) mostra a linha de varredura do sinal ultrassônico e o envelope do mesmo, calculado através do valor absoluto da Transformada de Hilbert, com os primeiros picos ampliados na Figura 73(b).

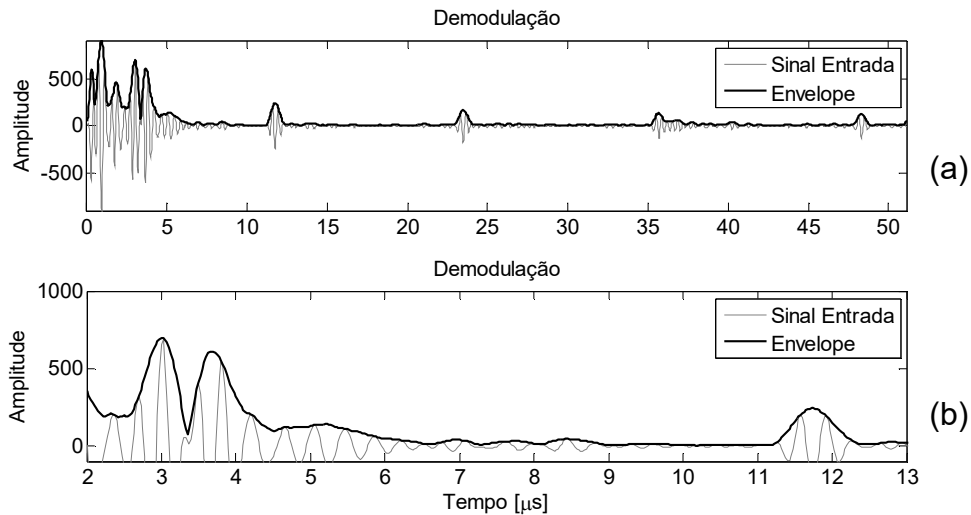


Figura 73 – Resultado da detecção de envoltória. (a) Linha de varredura e respectivo envelope do sinal. (b) Ampliação dos primeiros picos do item (a).

A validação do bloco “Demodulação” foi realizada através da aplicação da NRMSE, comparando o sinal gerado pelo modelo e o valor absoluto da Transformada de Hilbert gerado no Matlab, através da sintaxe:  $abs(hilbert(x))$ . Na Figura 74 são apresentados os sinais computados sobrepostos, resultando em uma NRMSE igual a 6,19%.

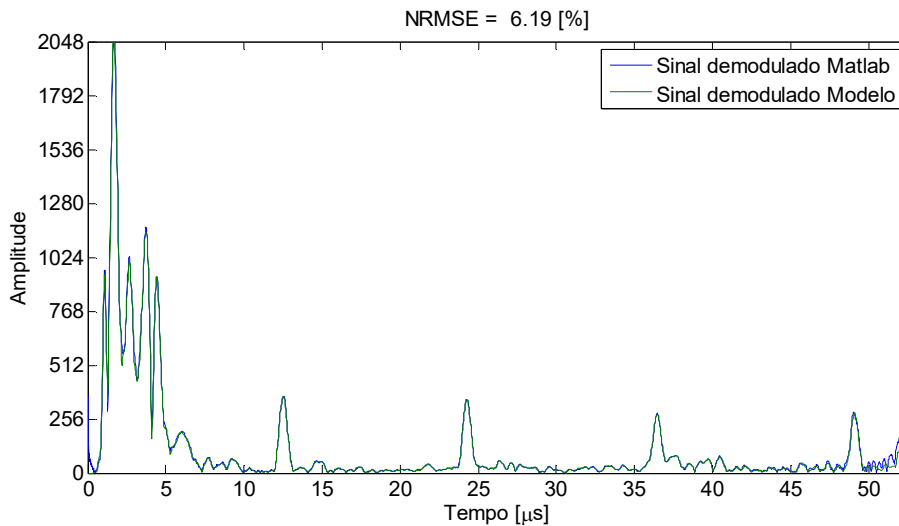


Figura 74 – Validação do processamento de detecção de envoltória através da comparação dos resultados do modelo e do Matlab.

#### 4.1.8 Compressão logarítmica

Na Figura 75 são apresentados os resultados do modelo de compressão logarítmica. Na Figura 75(a) apresenta-se a envoltória do sinal de US após a demodulação que é aplicada ao modelo de compressão logarítmica. A Figura 75(b) mostra o sinal computado pelo modelo, assim como o sinal limitado na faixa dinâmica de -30 dB para melhoria na resolução de contraste.

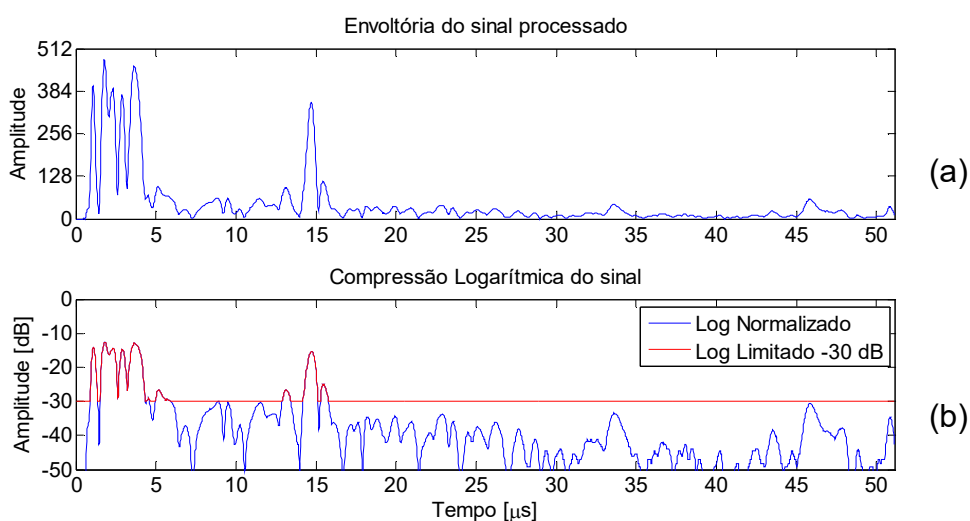


Figura 75 – Validação do modelo de compressão logarítmica. (a) Envelope do sinal de entrada do bloco. (b) Sinais de saída limitados em -30 dB (vermelho) e -50 dB (azul).

A validação do bloco “Compressão Logarítmica” foi realizada através da aplicação da função de custo NRMSE comparando os sinais resultantes do modelo e do Matlab, considerando o mesmo dado de entrada. Na Figura 76 é apresentado o comparativo dos sinais de saída processados sobrepostos com NRMSE igual a 0,45%.

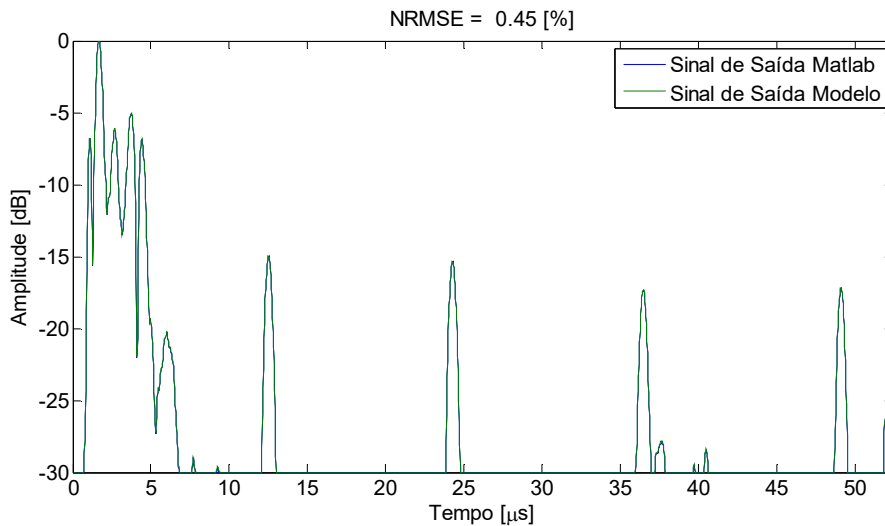


Figura 76 – Comparativo dos sinais processados pelo modelo e no Matlab sobrepostos com NRMSE igual a 0,45%.

## 4.2 RESULTADO FINAL DO PROCESSAMENTO PARA RECONSTRUÇÃO DE IMAGEM

### 4.2.1 Comparativo entre os processamentos simulado e experimental

As imagens de US Modo B geradas por rotinas no Matlab, a partir de dados adquiridos por simulação no ambiente Simulink, e pelo processamento em FPGA, são apresentadas na Figura 77. A coluna da esquerda representa o processamento em Simulink com os seguintes janelamentos: (a) Retangular, (b) Gaussiana, (c) Hamming e (d) Hanning. A coluna da direita representa o resultado do processamento da FPGA com os mesmos janelamentos.

### 4.2.2 Comparativo entre os processamentos da plataforma de pesquisa ULTRA-ORS e do sistema implementado em FPGA

De forma similar, as imagens de US Modo B, geradas conforme o trabalho de Assef (2013), e pelo processamento em FPGA, são apresentadas na Figura 78. A coluna da esquerda representa o processamento do sistema ULTRA-ORS com os seguintes janelamentos: (a) Retangular, (b) Gaussiana, (c) Hamming e (d) Hanning. A coluna da direita representa o resultado do processamento da FPGA com os mesmos janelamentos.



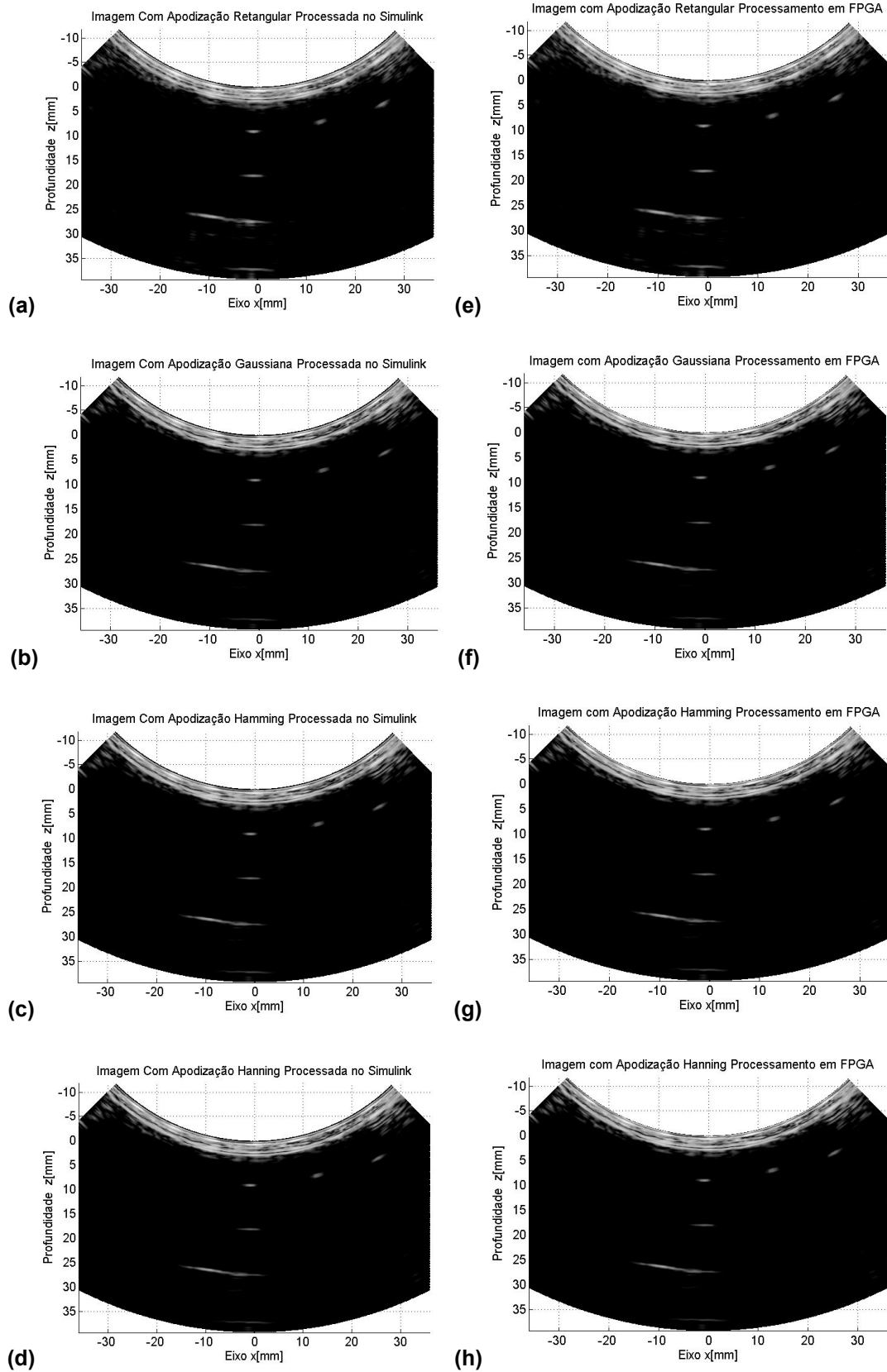


Figura 77 – Comparação entre as imagens reconstruídas em Modo B com as apodizações (a) Retangular, (b) Gaussiana, (c) Hamming e (d) Hanning no Simulink, e (e) Retangular, (f) Gaussiana, (g) Hamming e (h) Hanning com dados computados pela FPGA.

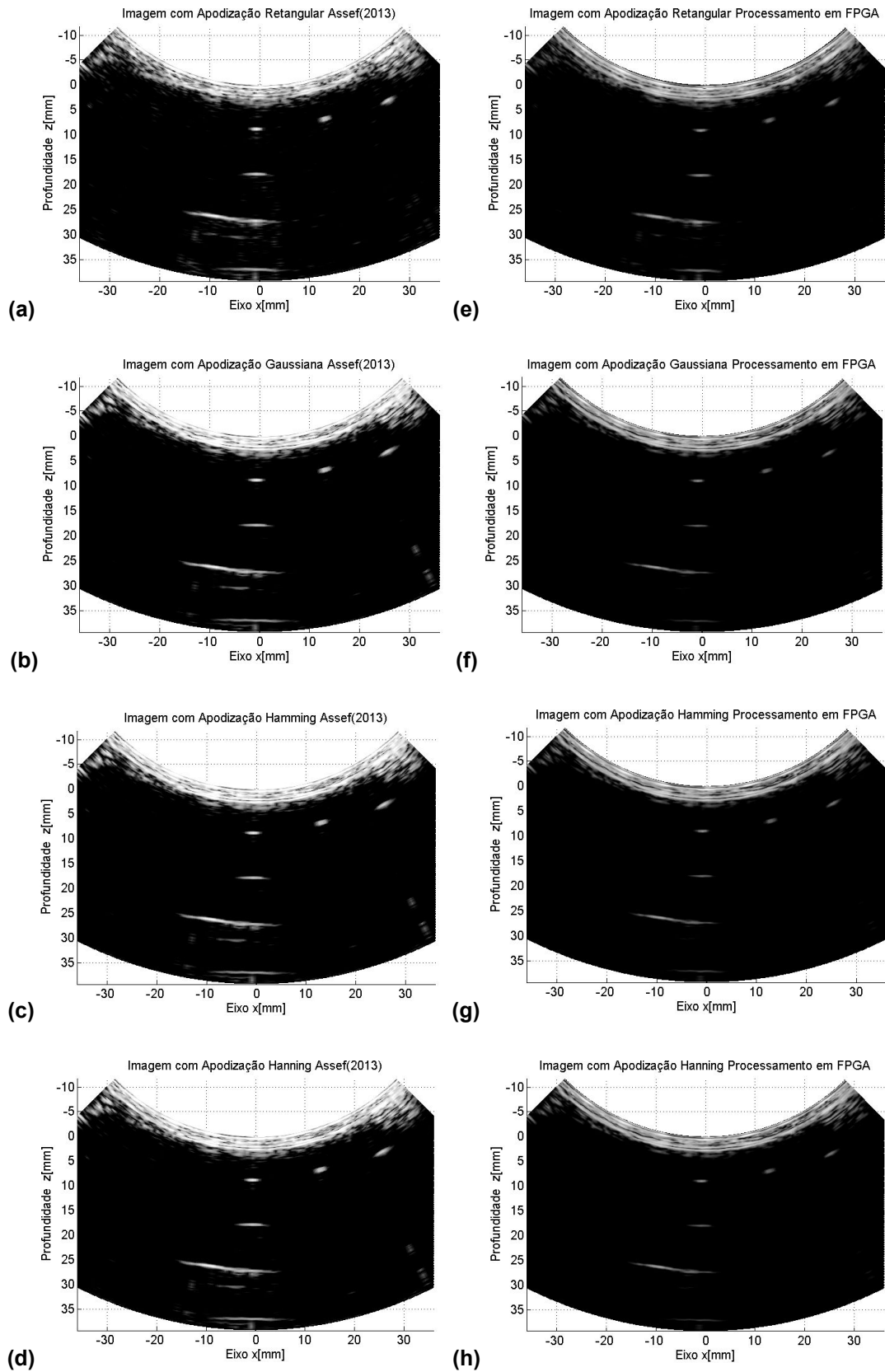


Figura 78 – Comparação entre as imagens reconstruídas em Modo B com as apodizações (a) Retangular, (b) Gaussiana, (c) Hamming e (d) Hanning na ULTRA\_ORS, e (e) Retangular, (f) Gaussiana, (g) Hamming e (h) Hanning com dados computados pela FPGA.

#### 4.2.3 Utilização de recursos de *hardware* da FPGA

Os recursos de *hardware* da FPGA Stratix IV utilizados na implementação proposta de cada um dos quatro projetos, já com a inclusão dos dados brutos de US, são sumarizados na Tabela 20. Estes resultados mostram que a maior parte dos recursos foi destinada à memória para a gravação dos dados. Além disso, apesar da máxima frequência do projeto ser de 160 MHz, a máxima frequência de operação da FPGA utilizada é de 550 MHz (ALTERA CORP., 2014).

Tabela 20 – Recursos de *hardware* da FPGA Stratix IV consumidos.

<b>Recurso</b>	<b>Especificação</b>
ALUTs Combinacional	4.402/182.400 (2%)
Registradores lógicos dedicados	4.334/182.400 (2%)
PLL	1/8 (13%)
Quantidade de memória em <i>bits</i> sem SignalTap	7.412.064/14.625.792 (51%)
Quantidade de memória em <i>bits</i> com SignalTap	10.164,576/14.625.792 (69%)
Blocos multiplicadores DSP de 18- <i>bits</i>	162/1.288 (13%)

#### 4.2.4 Latência total de processamento

Após a implementação e validação do sistema de processamento digital de sinais através de simulações e de forma experimental com a placa DE4-230, foi possível mensurar a latência da implementação do modelo em *hardware*, resultando em 33 ciclos de *clock* da FPGA. Como a frequência do sistema é igual a 40 MHz, a latência total é de aproximadamente 0,83  $\mu$ s, sendo adequada para geração de imagens em tempo real (JENSEN et al., 2005; CHANG; YEN; SHUNG, 2007).

## 5 DISCUSSÃO E CONCLUSÕES

Este capítulo apresenta as discussões dos resultados obtidos neste trabalho, inclusive comparando o resultado final com o trabalho relacionado (ASSEF, 2013), assim como sugestões de trabalhos futuros que poderão dar sequência a esta pesquisa.

### 5.1 ANÁLISE DOS RESULTADOS

O método de desenvolvimento adotado neste trabalho apresentou uma alternativa adequada para modelagem e implementação dos blocos de processamento para reconstrução de imagem por US. Soma-se a isso, a capacidade de simulação e validação de cada etapa do processamento de forma individual através de comparações qualitativas e quantitativas utilizando como referência algoritmos implementados em rotinas do Matlab.

A utilização do dispositivo FPGA neste trabalho se justifica devido à necessidade de computação paralela de todas as etapas da cadeia de processamento até o somatório coerente. Neste trabalho, foi utilizada a abertura de oito canais, porém outras aberturas sintéticas podem ser exploradas, sendo o fator limitante a quantidade de recursos de *hardware* da FPGA.

O modelo do filtro FIR proposto apresentou excelente concordância com o resultado de referência, apresentando uma NRMSE igual a 0,03%. Como desvantagem do modelo, o tempo de atraso de processamento foi de dois ciclos comparado com o modelo de referência. Entretanto, essa diferença pode ser anulada através do aprimoramento das funções utilizadas.

O processamento de interpolação CIC mostrou-se confiável e viável para aplicação neste trabalho. Considerando a máxima frequência de operação da FPGA como 550 MHz, o fator máximo de interpolação possível nesse módulo é 13 (550 MHz/40 MHz). Entretanto, conforme a Figura 66, na qual o desvio máximo do número de ciclos de atraso para frequência de 160 MHz é aproximadamente 1,5%, não foi necessário utilizar a máxima capacidade em frequência do dispositivo. Além disso, esse desvio é três vezes menor que na

frequência de amostragem em 40 MHz, justificando a escolha do fator de interpolação, e posterior decimação. Apesar do módulo de atraso variável possuir a capacidade de correção de atraso de até 64 ciclos, esse número pode ser aumentado para possibilitar a focalização da abertura em outras profundidades e frequências.

Para a avaliação da apodização foram escolhidos quatro algoritmos de janelamento, selecionáveis através de duas chaves presentes no *kit* de FPGA. Conforme a Figura 68 e a Figura 69, este método facilitou a computação das linhas de varredura, após o somatório coerente, e avaliação dos resultados do trabalho para diferentes apodizações evitando a recompilação do projeto. Entretanto, outras técnicas de apodização podem ser empregadas, conforme os trabalhos de Assef (2013), Cincotti et al. (1999) e Frazier e O'Brien (1998).

A técnica de demodulação apresentada neste trabalho é vantajosa quando comparada a outras técnicas de demodulação que requerem a utilização de algoritmos complexos baseados em FFT ou funções seno e cosseno para computar o sinal analítico das linhas de varredura (CHANG; YEN; SHUNG, 2007; LEVESQUE; SAWAN, 2009; QIU et al., 2012; HASSAN; KADAH, 2013). Na avaliação quantitativa através da função de custo NRMSE o valor obtido foi igual a 6,19%, apresentando concordância com os resultados encontrados na literatura científica (LEVESQUE; SAWAN, 2009; ZHOU; ZHENG, 2015). Apesar de ainda ser considerado um excelente resultado (JAMIESON; PORTER; WILSON, 1991; DETTORI et al., 2011) esse valor pode ser melhorado aumentando-se o número de coeficientes da Transformada de Hilbert baseada no filtro FIR.

Para compressão logarítmica foi adotada a normalização de 11 *bits*, resultando em um valor máximo de 2047 endereços, onde cada um armazena o valor logarítmico correspondente ao endereço da LUT utilizada. Esse procedimento foi diferente do implementado no sistema ULTRA-ORS e mais simples que o apresentado por Chang; Yen e Shung (2007). No caso do ULTRA-ORS, foi empregada uma função do Matlab para encontrar o valor máximo de amplitude entre todas as 121 linhas de varredura com abertura de oito elementos. Esse processo não foi implementado no modelo proposto, visto que acrescentaria um atraso de computação igual a 6,19 ms ( $33+(121 \text{ linhas de$

varredura x 2046 amostras)), sendo inadequado para processamento de imagens em tempo real (JENSEN et al., 2005).

Na Figura 77 e Figura 78 foram mostrados os efeitos da aplicação da apodização, utilizando os quatro janelamentos especificados. Em ambos os casos, observam-se resultados adequados. Na Figura 77 foi apresentada a comparação entre as imagens reconstruídas com os sinais processados no Simulink e na FPGA. Na Figura 78, pode-se observar que as imagens reconstruídas com os dados do processamento em FPGA são mais escuras. Isso se explica, pois, a computação logarítmica implementada não utiliza o valor máximo da envoltória dos valores processado e, sim, o valor máximo da escala de 11 *bits*. Cabe salientar que o tempo total para compilar o projeto no Simulink foi de aproximadamente 45 minutos, enquanto que na FPGA o tempo total de processamento é de 6,19 ms, possibilitando, teoricamente, 161 quadros por segundo, e, conseqüentemente, a geração de imagem em tempo real (HEDRICK; HYKES; STARCHMAN, 1995). Dessa forma, considerando uma futura interface de transferência de dados de alta velocidade entre o sistema ULTRA-ORS e a FPGA, esse é um resultado promissor para reconstrução de imagem em tempo real por US.

## 5.2 CONCLUSÕES

Pode-se concluir que o objetivo deste trabalho, que foi a modelagem, implementação e validação de um sistema de processamento digital de sinais baseado em dispositivo FPGA de alto desempenho para geração de imagens por US Modo B usando o Matlab/Simulink, foi atingido com sucesso.

Todas as características definidas na especificação de requisitos foram implementadas e apresentaram funcionalidade correta.

A maior contribuição deste trabalho de pesquisa é a disponibilização dos modelos validados dos principais blocos de processamento do *beamforming* de recepção utilizando o Simulink, em conjunto com a ferramenta DSP Builder. Tais estruturas poderão ser usadas para o desenvolvimento de pesquisas na área de US envolvendo instrumentação eletrônica, estudo de detecção de ondas

ultrassônicas, testes NDE e a sua interação com meios biológicos e não biológicos.

Os resultados apresentados neste trabalho são compatíveis com os resultados publicados na literatura científica(CHANG; YEN; SHUNG, 2007; LEVESQUE; SAWAN, 2009; HASSAN; YOUSSEF; KADAH, 2011; ASSEF, 2013; HASSAN; KADAH, 2013; ZHOU; ZHENG, 2015). Isto indica que o modelamento proposto é válido para dar continuidade aos trabalhos de pesquisa do US realizados no LUS da UTFPR.

### 5.3 TRABALHOS FUTUROS

Como sugestão para dar continuidade ao trabalho, podem-se citar os seguintes temas:

- Realização de testes de contraste comparativos entre a resposta do processamento da plataforma ULTRA-ORS e os dados processados pelo modelo proposto;
- Avaliação das resoluções lateral e axial para quantificar a qualidade do sistema de processamento digital de sinais embarcado em FPGA;
- Estudo de uma interface de comunicação serial de alta velocidade para transferência massiva de dados entre o sistema ULTRA-ORS e o *kit* de processamento em FPGA;
- Desenvolvimento de uma placa interface entre a plataforma ULTRA-ORS e *kits* de FPGA para transferência dos sinais brutos de RF;
- Estudo aprofundado de cada bloco implementado, a fim de minimizar os recursos de *hardware* empregados;
- Implementação do modelo de conversão de varredura na FPGA para transdutores matriciais convexo e linear;
- Avaliação da aplicação dos modelos implementados para geração de imagens por US utilizando outras modalidades, como por exemplo, estimação de fluxo sanguíneo através da técnica *Doppler*;

- Inclusão de um processador embarcado para flexibilizar a alteração de parâmetros relativos ao processamento de imagem, incluindo, por exemplo: ajuste da profundidade de focalização, ajuste do tamanho da abertura de elementos, controle automático digital de ganho de compensação de atenuação com o tempo/profundidade (TGC - *Time Gain Compensation*), outras janelas de apodização, dentre outros;
- Avaliação de outras técnicas de processamento para reconstrução de imagem por US, como por exemplo: *Ultrafast Imaging* (TANTER; FINK, 2014) e *Compounding Imaging* (ENTREKIN et al., 2001).



## REFERÊNCIAS

ACHIM, Alin; ANASTASIOS, Bezerianos; e TSAKALIDES, Panagiotis. Novel Bayesian multiscale method for speckle removal in medical ultrasound images. **IEEE Transactions on Medical Imaging**, v. 20, n. 8, p. 772–783, 2001.

AGARWAL, A.; FUKUOKA, T.; SCHNEIDER, F. K.; YOO, Y. M.; BALUYOT, F.; e KIM, Y. P2B-17 Single-Chip Solution for Ultrasound Imaging Systems: Initial Results. In: 2007 IEEE Ultrasonics Symposium Proceedings, **Anais...IEEE**, out. 2007.

AGARWAL, Mayur; BANERJEE, Swapna; e DE, Arijit. Architecture of a real-time delay calculator for digital beamforming in ultrasound system. **IET Circuits, Devices & Systems**, v. 10, n. 4, p. 322–329, jul. 2016.

ALTERA CORP. **Stratix IV Device Handbook - Volume 4: Device Datasheet and Addendum**. Disponível em: <[https://www.altera.com/en\\_US/pdfs/literature/hb/stratix-iv/stx4\\_5v4.pdf](https://www.altera.com/en_US/pdfs/literature/hb/stratix-iv/stx4_5v4.pdf)>. Acesso em: 24 mar. 2017.

ASEN, Jon P.; BUSKENES, Jo I.; NILSEN, Carl-Inge C.; AUSTENG, Andreas; e HOLM, Sverre. Implementing Capon beamforming on the GPU for real time cardiac ultrasound imaging. In: 2012 IEEE International Ultrasonics Symposium, **Anais...IEEE**, out. 2012.

ASSEF, Amauri A. **Arquitetura de hardware multicanal reconfigurável com excitação multinível para desenvolvimento e testes de novos métodos de geração de imagens por ultrassom**. 2013. Universidade Tecnológica Federal do Paraná, 2013.

ASSEF, Amauri A.; MAIA, Joaquim M.; FERREIRA, Breno M.; DE SOUZA, Higor S.; BASSAN, Guilherme M.; DE MORAES FILHO, Alexandre F.; e COSTA, Eduardo T. Projeto de um filtro digital FIR passa-baixa em FPGA para aplicações de processamento de sinais de ultrassom. In: Congresso Brasileiro de Engenharia Biomédica, CBEB 2016, Foz do Iguaçu. **Anais... Foz do Iguaçu: Congresso Brasileiro de Engenharia Biomédica**, 2016.

BABIC, Djordje; e RENFORS, Markku. Power efficient structure for conversion between arbitrary sampling rates. **IEEE Signal Processing Letters**, v. 12, n. 1, p. 1–4, jan. 2005.

BERGÉ, Jean-Michel.; FONKOUA, Alain.; MAGINOT, Serge.; e ROUILLARD, Jacques. **Vhdl designer's reference**. [s.l.] Springer US, 1992.

BONI, E.; BASSI, L.; DALLAI, A.; GUIDI, F.; RAMALLI, A.; RICCI, S.; HOUSDEN, J.; e TORTOLI, P. A reconfigurable and programmable FPGA-based system for nonstandard ultrasound methods. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 59, n. 7, p. 1378–1385, jul. 2012.

BOONLEELAKUL, W.; TECHAVIPOO, U.; WORASAWATE, D.; KEINPRASIT, R.; PINUNSOTTIKUL, P.; SUGINO, N.; e THAJCHAYAPONG, P. Compression of ultrasound RF data using quantization and decimation. In: The 6th 2013 Biomedical Engineering International Conference, **Anais...IEEE**, out. 2013.

BRANCO, Kalinka R. L. J. C.; PELIZZONI, Jorge M.; NERIS, Luciano O.; TRINDADE, Onofre; OSORIO, Fernando S.; e WOLF, Denis F. Tiriba - a new approach of UAV based on model driven development and multiprocessors. In: 2011 IEEE International Conference on Robotics and Automation, Shanghai. **Anais... Shanghai: IEEE**, maio 2011.

BRANDENSTEIN, Hartmut; e UNBEHAUEN, Rolf. Least-squares approximation of FIR by IIR digital filters. **IEEE Transactions on Signal Processing**, v. 46, n. 1, p. 21–30, 1998.

BROWN, J. A.; e LOCKWOOD, G. R. A Digital Beamformer for High-Frequency Annular Arrays. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 52, n. 8, p. 1262–1269, ago. 2005.

CÂMARA, Danilo Januário. **Plataforma de ultrassom programável dedicada à pesquisa**. 2015. Biblioteca Digital de Teses e Dissertações da Universidade de São Paulo, Ribeirão Preto, 2015.

CHANG, Jin; YEN, Jesse; e SHUNG, K. A Novel Envelope Detector for High-Frame Rate, High-Frequency Ultrasound Imaging. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 54, n. 9, p. 1792–1801, set. 2007.

CHRISTENSEN, Douglas A. **Ultrasonic bioinstrumentation**. New York: John Wiley & Sons, 1988.

CINCOTTI, G.; CARDONE, G.; GORI, P.; e PAPPALARDO, M. Efficient transmit

beamforming in pulse-echo ultrasonic imaging. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 46, n. 6, p. 1450–1458, 1999.

CROCHIERE, Ronald E.; e RABINER, Lawrence R. Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 23, n. 5, p. 444–456, 1975.

CROCHIERE, Ronald E.; e RABINER, Lawrence R. Further considerations in the design of decimators and interpolators. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. ASSP-24, n. 4, p. 296–311, 1976.

CROCHIERE, Ronald E.; e RABINER, Lawrence R. Interpolation and decimation of digital signals—A tutorial review. **Proceedings of the IEEE**, v. 69, n. 3, p. 300–331, 1981.

DA COSTA, César. **Projetos de circuitos digitais com fpga**. 3. ed. ed. São Paulo: Érica, 2014.

DETTORI, Marco; CESARACCIO, Carla; MOTRONI, Andrea; SPANO, Donatella; e DUCE, Pierpaolo. Using CERES-Wheat to simulate durum wheat production and phenology in Southern Sardinia, Italy. **Field Crops Research**, v. 120, n. 1, p. 179–188, jan. 2011.

DUBE, Noël. Introduction to phased array ultrasonic technology applications : R/D Tech guideline. . 2004.

ENTREKIN, Robert R.; PORTER, Bruce A.; SILLESEN, Henrik H.; WONG, Anthony D.; COOPERBERG, Peter L.; e FIX, Cathy H. Real-time spatial compound imaging: Application to breast, vascular, and musculoskeletal ultrasound. **Seminars in Ultrasound, CT and MRI**, v. 22, n. 1, p. 50–64, fev. 2001.

ESTEVEZ, E.; e MARCOS, M. Model-Based Validation of Industrial Control Systems. **IEEE Transactions on Industrial Informatics**, v. 8, n. 2, p. 302–310, maio 2012.

FELDKAMPER, H. T.; SCHWANN, R.; GIERENZ, V.; e NOLL, T. G. Low power delay calculation for digital beamforming in handheld ultrasound systems. In:

2000 IEEE Ultrasonics Symposium. Proceedings. An International Symposium (Cat. No.00CH37121), San Juan. **Anais...** San Juan: IEEE, 2000.

FISH, Peter. **Physics and instrumentation of diagnostic medical ultrasound**. [s.l.] John Wiley & Sons Incorporated, 1990.

FRAZIER, C. H.; e O'BRIEN, W. D. Synthetic aperture techniques with a virtual source element. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 45, n. 1, p. 196–207, jan. 1998.

GIRON, Alexandre A. **Validação de Transformações de Modelos de Engenharia Dirigida a Modelos (MDE)**. Disponível em: <<https://pt.slideshare.net/AlexandreAugustoGiron/mde-model-driven-engineering-e-validao-de-transformaes>>. Acesso em: 2 fev. 2017.

GOLD, B.; OPPENHEIM, A. V.; e RADER, C. M. Theory and implementation of the discrete Hilbert transform. **Presented at the Symposium on Computer Processing in Communications**, v. 235, 1969.

HASSAN, Mawia A.; e KADAH, Yasser M. Digital signal processing methodologies for conventional digital medical ultrasound imaging system. **American Journal of Biomedical Engineering**, v. 3, n. 1, p. 14–30, 2013.

HASSAN, Mawia A.; YOUSSEF, Abou-Bakr M.; e KADAH, Yasser M. Modular FPGA-based digital ultrasound beamforming. In: 2011 1st Middle East Conference on Biomedical Engineering, United States. **Anais...** United States: IEEE, fev. 2011.

HEDRICK, Wayne R.; HYKES, David L.; e STARCHMAN, Dale E. **Ultrasound physics and instrumentation**. 3rd ed / W ed. St. Louis ; London: CV Mosby, 1995.

HENTSCHEL, Tim; e FETTWEIS, Gerhard. Sample rate conversion for software radio. **IEEE Communications Magazine**, v. 38, n. 8, p. 142–150, 2000.

HOGENAUER, E. An economical class of digital filters for decimation and interpolation. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 29, n. 2, p. 155–162, abr. 1981.

HOKLAND, J. H.; e TAXT, T. Ultrasound speckle reduction using harmonic oscillator models. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 41, n. 2, p. 215–224, mar. 1994.

HUDA, Safeen; MALLICK, Muntasir; e ANDERSON, Jason H. Clock gating architectures for FPGA power reduction. In: 2009 International Conference on Field Programmable Logic and Applications, Czech Republic. **Anais...** Czech Republic: IEEE, ago. 2009.

INTEL CORP. **FIR Compiler User Guide**. Disponível em: <[https://www.altera.com/en\\_US/pdfs/literature/ug/fircompiler\\_ug.pdf](https://www.altera.com/en_US/pdfs/literature/ug/fircompiler_ug.pdf)>. Acesso em: 12 mar. 2017.

**DSP Builder for Intel FPGAs**. Disponível em: <<https://www.altera.com/products/design-software/model---simulation/dspbuilder/%0Amathworks.html>>. Acesso em: 16 mar. 2017.

**FPGA CPLD and ASIC from Altera**. Disponível em: <<https://www.altera.com/>>. Acesso em: 16 mar. 2017.

JAMIESON, P. D.; PORTER, J. R.; e WILSON, D. R. A test of the computer simulation model ARCWHEAT1 on wheat crops grown in New Zealand. **Field Crops Research**, v. 27, p. 337–350, 1991.

JENSEN, J. A.; e MUNK, P. A new method for estimation of velocity vectors. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 45, n. 3, p. 837–851, maio 1998.

JENSEN, Jørgen A. FIELD: A Program for Simulating Ultrasound Systems. **10TH NORDICBALTIC CONFERENCE ON BIOMEDICAL IMAGING, VOL. 4, SUPPLEMENT 1, PART 1:351--353**, v. 34, p. 351--353, 1996.

JENSEN, Jørgen A.; HOLM, Ole; JERISEN, LJ; BENDSEN, Henrik; NIKOLOV, Svetoslav I.; TOMOV, Borislav G.; MUNK, Peter; HANSEN, Martin; SALOMONSEN, Kent; e HANSEN, Johnn. Ultrasound research scanner for real-time synthetic aperture data acquisition. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 52, n. 5, p. 881–891, maio 2005.

JHANG, Kyung-Young. Nonlinear ultrasonic techniques for nondestructive assessment of micro damage in material: A review. **International Journal of**

**Precision Engineering and Manufacturing**, v. 10, n. 1, p. 123–135, jan. 2009.

JI-FENG, Ding; SHUANG, Xu; ZHANG, Jun-Xing; e YA-NING, Yang. Filter Design Based on DSP Builder. **Open Electrical & Electronic Engineering Journal**, v. 9, p. 15–21, 2015.

JINBO, Maro. **Imagens ultra-sônicas em modo-b com técnicas de abertura focal sintética - saft**. 2007. Biblioteca Digital de Teses e Dissertações da Universidade de São Paulo, São Paulo, 2007.

KANG, Jeeun; YOON, Changhan; LEE, Jaejin; KYE, Sang-Bum; LEE, Yongbae; CHANG, Jin Ho; KIM, Gi-Duck; YOO, Yangmo; e SONG, Tai-kyong. A System-on-Chip Solution for Point-of-Care Ultrasound Imaging Systems: Architecture and ASIC Implementation. **IEEE Transactions on Biomedical Circuits and Systems**, v. 10, n. 2, p. 412–423, abr. 2016.

KELLY, Stlephen P.; FARLOW, Roger; e HAYWARD, G. Applications of through-air ultrasound for rapid NDE scanning in the aerospace industry. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 43, n. 4, p. 581–591, jul. 1996.

KIM, Gi-Duck; YOON, Changhan; KYE, Sang-Bum; LEE, Youngbae; KANG, Jeeun; YOO, Yangmo; e SONG, Tai-Kyong. A single FPGA-based portable ultrasound imaging system for point-of-care applications. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 59, n. 7, p. 1386–1394, jul. 2012.

KRAUTKRÄMER, Josef; e KRAUTKRÄMER, Herbert. **Ultrasonic testing of materials**. 4th fully ed. New York: Springer-Verlag, 1990.

LEVESQUE, Philippe; e SAWAN, Mohamad. Real-Time Hand-Held Ultrasound Medical-Imaging Device Based on a New Digital Quadrature Demodulation Processor. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 56, n. 8, p. 1654–1665, ago. 2009.

MA, Jieming; KARADAYI, Kerem; ALI, Murtaza; e KIM, Yongmin. Ultrasound phase rotation beamforming on multi-core DSP. **Ultrasonics**, v. 54, p. 99–105, 2014.

MACHADO, Rafael P.; KOWALSKI, Edemir L.; LAZZARETTI, André E.;

FRAGOSO, Jackson R.; SWINKA FILHO, V.; JUNIOR, José A. T.; RAVAGLIO, Marcelo A.; DE OLIVEIRA, Nilson; BATHKE, Rodolfo C.; e DE HOLANDA CAVALCANTI, Daniel. Metodologia para a inspeção instrumentalizada de isoladores de pino de redes de distribuição por RFI e ultrassom. In: VI Congresso de Inovação Tecnológica em Energia Elétrica, Fortaleza/CE. **Anais...** Fortaleza/CE: VI CITENEL, 2011.

MADISETTI, Vijay K.; e WILLIAMS, Douglas. **The digital signal processing handbook**. Boca Raton, Fla.: CRC Press, 1998.

MATHWORKS. **MATLAB and Simulink for Technical Computing**. Disponível em: <<http://www.mathworks.com/>>. Acesso em: 16 mar. 2017.

MATRONE, Giulia; SAVOIA, Alessandro Stuart; CALIANO, Giosue; e MAGENES, Giovanni. The Delay Multiply and Sum Beamforming Algorithm in Ultrasound B-Mode Medical Imaging. **IEEE Transactions on Medical Imaging**, v. 34, n. 4, p. 940–949, abr. 2015.

MAYILAVELANE, Aroutchelvame; e BERSCHIED, Brian. A Fast FIR filtering technique for multirate filters. **Integration, the VLSI Journal**, v. 52, p. 62–70, 2016.

MCCLELLAN, J.; e PARKS, T. A unified approach to the design of optimum FIR linear-phase digital filters. **IEEE Transactions on Circuit Theory**, v. 20, n. 6, p. 697–701, nov. 1973.

MEO, Michele; POLIMENO, Umberto; e ZUMPANO, Giuseppe. Detecting Damage in Composite Material Using Nonlinear Elastic Wave Spectroscopy Methods. **Applied Composite Materials**, v. 15, n. 3, p. 115–126, maio 2008.

MINISTÉRIO DA SAÚDE BRASIL. **PORTARIA Nº 375, DE 28 DE FEVEREIRO DE 2008 - Programa Nacional para Qualificação, Produção e Inovação em Equipamentos e Materiais de Uso em Saúde no Complexo Industrial da Saúde**.

MONMASSON, Eric; IDKHAJINE, Lahoucine; CIRSTEIA, Marcian N.; BAHRI, Imene; TISAN, Alin; e NAOUAR, Mohamed Wissem. FPGAs in Industrial Control Applications. **IEEE Transactions on Industrial Informatics**, v. 7, n. 2, p. 224–243, maio 2011.

MURPHY, Brett; WAKEFIELD, Amory; e FRIEDMAN, Jon. **Best Practices for Verification, Validation, and Test in Model-Based Design** SAE International, , 2008. .

NGUYEN, M.; SHIN, J.; e YEN, J. Fresnel beamforming and dual apodization with cross-correlation for curvilinear arrays in low-cost portable ultrasound system. In: 2011 IEEE International Ultrasonics Symposium, Orlando. **Anais...** Orlando: IEEE, 2011.

NIKOLOV, Svetoslav I.; JENSEN, Jorgen A.; e TOMOV, Borislav G. Fast parametric beamformer for synthetic aperture imaging. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 55, n. 8, p. 1755–1767, ago. 2008.

OPPENHEIM, Alan V; e SCHAFER, Ronald W. **Digital signal processing**. Englewood Cliffs ; London: Prentice-Hall, 1975.

OPPENHEIM, Alan V; WILLSKY, Alan S.; e NAWAB, S.Hamid. **Signals & systems**. 2nd ed. ed. Upper Saddle River: Prentice Hall ; London : Prentice-Hall International, 1997.

OTAKE, Tsuyoshi; KAWANO, Toshihiko; SUGIYAMA, Takashi; MITAKE, Tsuyoshi; e UMEMURA, Shinichiro. High-quality/High-resolution Digital Ultrasound Diagnostic Scanner. **Hitachi Review**, v. 52, n. 4, 2003.

PAU, C. J. A stereo audio chip using approximate processing for decimation and interpolation filters. **IEEE Journal of Solid-State Circuits**, v. 35, n. 1, p. 45–55, jan. 2000.

QIU, Weibao; YE, Zongying; YU, Yanyan; CHEN, Yan; CHI, Liyang; MU, Peitian; LI, Guofeng; WANG, Congzhi; XIAO, Yang; DAI, Jiyan; SUN, Lei; e ZHENG, Hairong. A Digital Multigate Doppler Method for High Frequency Ultrasound. **Sensors**, v. 14, n. 8, p. 13348–13360, jul. 2014a.

QIU, Weibao; YU, Yanyan; TSANG, Fu Keung; e SUN, Lei. An FPGA-based open platform for ultrasound biomicroscopy. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 59, n. 7, p. 1432–1442, jul. 2012.

QIU, Weibao; ZONGYING YE; LIYANG CHI; PEITIAN MU; GUOFENG LI; CONGZHI WANG; YANG XIAO; MING QIAN; SUN, Lei; e ZHENG, Hairong. A



programmable ultrasound platform for multi-gate Doppler measurement. In: 2014 IEEE International Ultrasonics Symposium, Chicago. **Anais...** Chicago: IEEE, set. 2014b.

RABINER, L. R.; KAISER, J. F.; HERRMANN, O.; e DOLAN, M. T. Some Comparisons Between FIR and IIR Digital Filters. **Bell System Technical Journal**, v. 53, n. 2, p. 305–331, fev. 1974.

RABINER, Lawrence R.; e SCHAFER, Ronald. **Digital processing of speech signals**. Englewood Cliffs ; London: Prentice-Hall, 1978.

RODRIGUEZ-ANDINA, J. J.; MOURE, M. J.; e VALDES, M. D. Features, Design Tools, and Application Domains of FPGAs. **IEEE Transactions on Industrial Electronics**, v. 54, n. 4, p. 1810–1823, ago. 2007.

SANTOS, Fábio Dissan dos. **Utilização de dsps e fpgas em unidades eletrônicas automotivas**. 2008. Universidade de São Paulo, 2008.

SCHNEIDER, Fabio K.; AGARWAL, Anup; YOO, Yang M.; FUKUOKA, Tetsuya; e KIM, Yongmin. A Fully Programmable Computing Architecture for Medical Ultrasound Machines. **IEEE Transactions on Information Technology in Biomedicine**, v. 14, n. 2, p. 538–540, mar. 2010.

SHATTUCK, David P.; WEINSHENKER, Marc D.; SMITH, Stephen W.; e VON RAMM, Olaf T. Explososcan: A parallel processing technique for high speed ultrasound imaging with linear phased arrays. **The Journal of the Acoustical Society of America**, v. 75, n. 4, p. 1273–1282, abr. 1984.

SHEN, J.; WANG, H.; CAIN, C.; e EBBINI, E. S. A post-beamforming processing technique for enhancing conventional pulse-ultrasound imaging contrast resolution. In: 1995 IEEE Ultrasonics Symposium. Proceedings. An International Symposium, Washington. **Anais...** Washington: IEEE, 1995.

SHUNG, Kirk. **Diagnostic ultrasound: imaging and blood flow measurements**. New York: CRC press, 2006.

SOHN, H. Y.; KANG, J.; CHO, J.; SONG, T. K.; e YOO, Y. Time-sharing bilinear delay interpolation for ultrasound dynamic receive beamformer. **Electronics Letters**, v. 47, n. 2, p. 89, 2011.

SRIVASTAVA, Deepa; e MEHRA, Rajesh. Performance estimation of distinctive configuration based CIC filter for interpolation. In: 2015 Annual IEEE India Conference (INDICON), New Delhi. **Anais...** New Delhi: IEEE, dez. 2015.

STEPHAN, Matthew; ALALFI, Manar H.; e CORDY, James R. Towards a Taxonomy for Simulink Model Mutations. In: 2014 IEEE Seventh International Conference on Software Testing, Verification and Validation Workshops, Cleveland. **Anais...** Cleveland: IEEE, mar. 2014.

SYNNEVÅG, J. F.; AUSTENG, A.; e HOLM, S. A low-complexity data-dependent beamformer. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 58, n. 2, p. 281–289, fev. 2011.

TANTER, Mickael; e FINK, Mathias. Ultrafast imaging in biomedical ultrasound. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 61, n. 1, p. 102–119, jan. 2014.

TERASIC TECH. **DE4-230 Development Board**. Disponível em: <<http://www.terasic.com.tw/cgi-bin/%0Apage/archive.pl?Language=English&No=501>>. Acesso em: 15 mar. 2016.

VICTOREEM INC. Multi-purpose ultrasound phantom instruction manual. In: [s.l.: s.n.]p. 84–317.

WANG, Ziyu; JIAN, Chen; WANG, Benchao; e YANG, Wenli. The Application of Clock Synchronization in the TDOA Location System. **2nd International Conference on Electrical, Computer Engineering and Electronics**, 2015.

WELLS, Peter NT. Ultrasonic imaging of the human body. **Reports on Progress in Physics**, v. 62, n. 5, p. 671–722, maio 1999.

WIKISONIX. **Ultrasound Image Computation**. Disponível em: <[http://www.ultrasonix.com/wikisonix/index.php/Ultrasound\\_Image\\_Computatio%0An#Logarithmic\\_Compression\\_and\\_Adjustment\\_in\\_the\\_Dynamic\\_Range](http://www.ultrasonix.com/wikisonix/index.php/Ultrasound_Image_Computatio%0An#Logarithmic_Compression_and_Adjustment_in_the_Dynamic_Range)>. Acesso em: 20 fev. 2017.

WILD, John J. The use of ultrasonic pulses for the measurement of biologic tissues and the detection of tissue density changes. **Surgery**, 1950.

WILSON, Thaddeus; ZAGZEBSKI, James; VARGHESE, Tomy; CHEN, Quan; e RAO, Min. The ultrasonix 500RP: A commercial ultrasound research interface. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 53, n. 10, p. 1772–1782, out. 2006.

XILINX. **LogiCORE IP FIR Compiler v6.3**.

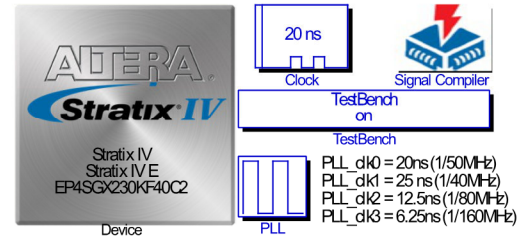
XUAN, Guixin; HAN, Yawei; REN, Min; ZHANG, Zhenyu; e ZENG, Fanxin. Efficient digital IF architecture for wideband OFDM systems based on CIC filter. In: 2014 12th International Conference on Signal Processing (ICSP), Hangzhou. **Anais...** Hangzhou: IEEE, out. 2014.

YANG, Zhi; e FOX, Martin D. Speckle Reduction and Structure Enhancement by Multichannel Median Boosted Anisotropic Diffusion. **EURASIP Journal on Advances in Signal Processing**, v. 2004, n. 16, p. 2492–2502, 2004.

ZHOU, Hao; e ZHENG, Yin-fei. An efficient quadrature demodulator for medical ultrasound imaging. **Frontiers of Information Technology & Electronic Engineering**, v. 16, n. 4, p. 2–2, abr. 2015.

**APÊNDICE A – PROJETO COMPLETO DO SISTEMA DSP BASEADO EM  
FPGA PARA A GERAÇÃO DE IMAGEM POR US NO SIMULINK**

O modelo esquemático do projeto completo do sistema DSP baseado em FPGA para a geração de imagem por US no Simulink é apresentado na próxima folha em formato A3 para melhor visualização dos componentes.

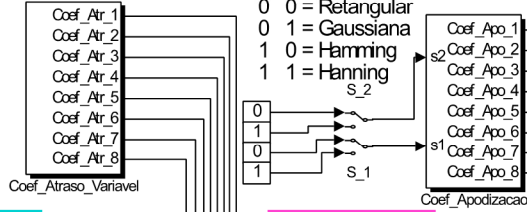


Seleção Janelamento:

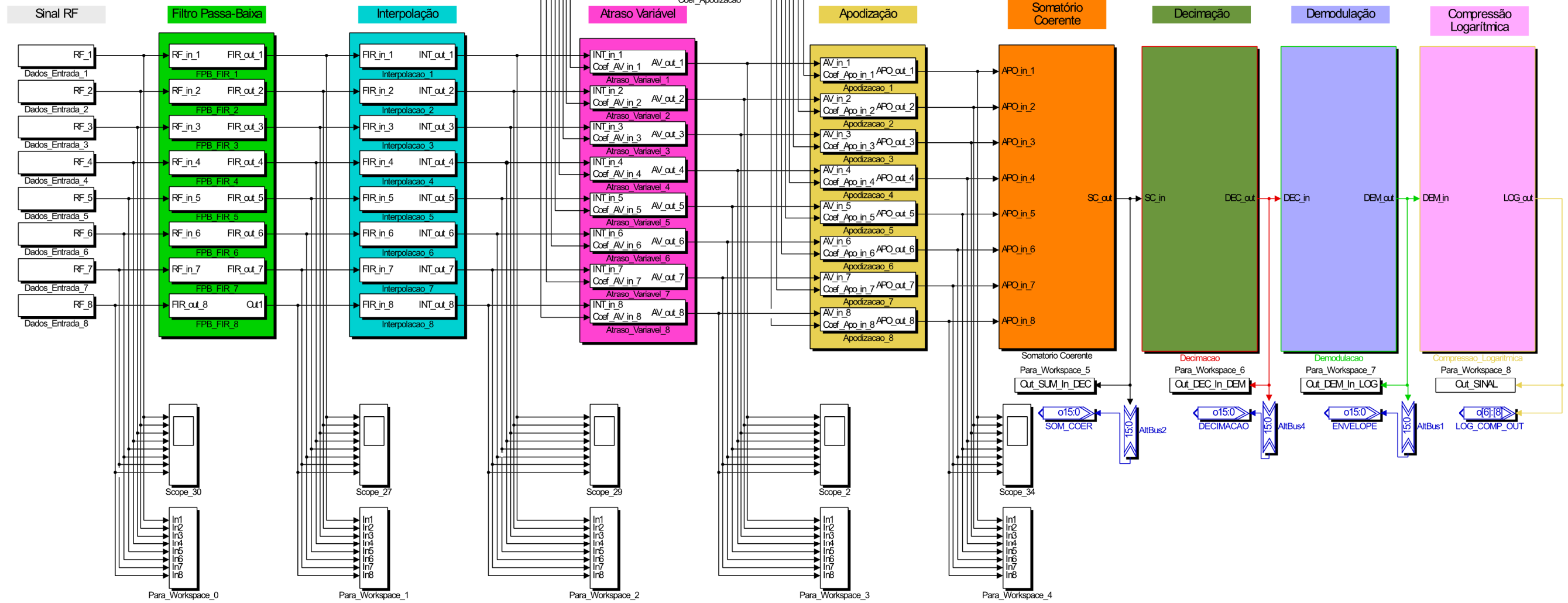
s2 s1

- 0 0 = Retangular
- 0 1 = Gaussiana
- 1 0 = Hamming
- 1 1 = Hanning

S<sub>2</sub>  
S<sub>1</sub>



### SISTEMA DE PROCESSAMENTO DIGITAL DE SINAIS BASEADO EM FPGA PARA GERAÇÃO DE IMAGENS POR ULTRASSOM USANDO O SIMULINK



**PUBLICAÇÕES REFERENTES AO TRABALHO**

1. ASSEF, A. A.; MAIA, J. M. ; FERREIRA, B. M. ; SERAFIN, H. S. ; BASSAN, G. M. ; DE MORAES FILHO, A. F. ; COSTA, E. T. .  
MODELAGEM E AVALIAÇÃO DE UM BEAMFORMING DIGITAL DE ULTRASSOM BASEADO EM FPGA USANDO O MATLAB/SIMULINK. In: XXV Congresso Brasileiro de Engenharia Biomédica, 2016, Foz do Iguaçu. CBEB 2016, 2016.
2. ASSEF, A. A.; MAIA, J. M. ; FERREIRA, B. M. ; SERAFIN, H. S. ; BASSAN, G. M. ; DE MORAES FILHO, A. F. ; COSTA, E. T. .  
PROJETO DE UM FILTRO DIGITAL FIR PASSA-BAIXA EM FPGA PARA APLICAÇÕES DE PROCESSAMENTO DE SINAIS DE ULTRASSOM. In: XXV Congresso Brasileiro de Engenharia Biomédica, 2016, Foz do Iguaçu. CBEB 2016, 2016.