

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

JONATHAN DE OLIVEIRA

**IMPLEMENTAÇÃO DE UM SISTEMA DE TRANSFERÊNCIA DE
DADOS DE ULTRASSOM VIA ETHERNET E PROCESSAMENTO
EMBARCADO EM DISPOSITIVO FPGA**

DISSERTAÇÃO

CURITIBA

2020

JONATHAN DE OLIVEIRA

**IMPLEMENTAÇÃO DE UM SISTEMA DE TRANSFERÊNCIA DE
DADOS DE ULTRASSOM VIA ETHERNET E PROCESSAMENTO
EMBARCADO EM DISPOSITIVO FPGA**

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

Orientador: Prof. Dr. Amauri Amorin Assef

CURITIBA

2020

Dados Internacionais de Catalogação na Publicação

Oliveira, Jonathan de

Implementação de um sistema de transferência de dados de ultrassom via Ethernet e processamento embarcado em dispositivo FPGA [recurso eletrônico] / Jonathan de Oliveira. -- 2020.

1 arquivo eletrônico (109 f.): PDF; 4,17 MB.

Modo de acesso: World Wide Web.

Texto em inglês com resumo em português.

Dissertação (Mestrado) - Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Sistemas de Energia. Área de Concentração: Automação e Sistemas de Energia. Linha de Pesquisa: Sistemas de Automação e Instrumentação, Curitiba, 2020.

Bibliografia: f. 100-107.

1. Sistemas de energia elétrica - Dissertações. 2. Ultrassom - Qualidade da imagem. 3. Diagnóstico por ultrassom. 4. Sistemas de transmissão de dados. 5. Ethernet (Sistema de rede local de computação). 6. Processamento de sinais - Técnicas digitais. 7. Reconstrução de imagens. 8. Beamforming. 9. Arranjos de lógica programável em campo. 10. Processamento eletrônico de dados em tempo real. 11. SIMULINK (Programa de computador). 12. Métodos de simulação. I. Assef, Amauri Amorin, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Sistemas de Energia. III. Título.

Biblioteca Central do Câmpus Curitiba - UTFPR
Bibliotecária: Luiza Aquemi Matsumoto CRB-9/794

TERMO DE APROVAÇÃO DE DISSERTAÇÃO

A Dissertação de Mestrado intitulada “**IMPLEMENTAÇÃO DE UM SISTEMA DE TRANSFERÊNCIA DE DADOS DE ULTRASSOM VIA ETHERNET E PROCESSAMENTO EMBARCADO EM DISPOSITIVO FPGA**”, defendida em sessão pública pelo candidato **Jonathan De Oliveira**, no dia **14 de maio de 2020**, foi julgada para a obtenção do título de Mestre em Engenharia Elétrica, Área de concentração **Automação E Sistemas De Energia**, Linha de pesquisa **Sistemas De Automação E Instrumentação**, e aprovada em sua forma final, pelo Programa de Pós-Graduação em Sistemas de Energia - PPGSE.

BANCA EXAMINADORA:

Prof. Dr. Amauri Amorin Assef - Presidente - UTFPR

Prof. Dr. Joaquim Miguel Maia - UTFPR

Prof. Dr. Marcos Santos Hara - IFPR

A via original deste documento encontra-se arquivada na Secretaria do Programa, contendo a assinatura da Coordenação após a entrega da versão corrigida do trabalho.

Curitiba, 14 de maio de 2020.

Carimbo e Assinatura do(a) Coordenador(a) do Programa

*Dedico este trabalho aos meus pais, aos meus familiares e aos
meus amigos.*

AGRADECIMENTOS

Inicialmente agradeço aos meus pais, Claudete e Hermogenes, por me apoiarem nas decisões inesperadas e aos meus familiares pela compreensão durante esses anos.

Em especial, agradeço ao meu amigo Alexandre Biembengut, pela amizade, apoio e suporte motivacional e incontáveis discussões ao longo desses anos.

Agradeço também à minha amiga Luiza Fugita, pelo carinho e pelas conversas filosóficas mesmo em momentos difíceis.

Agradeço ao meu amigo Luiz Carlos Giacomossi Jr, pela parceria e ajuda nas correções.

Ao professor e orientador Dr. Amauri Amorin Assef, pela dedicação, paciência, amizade e orientação singular, além de proporcionar essa oportunidade ímpar de realizar esse trabalho.

A todos os colegas de laboratório e amigos que estiveram presentes nessa jornada.

À UTFPR, CNPq, CAPES, FINEP, Fundação Araucária e Ministério da Saúde pelo apoio financeiro.

RESUMO

OLIVEIRA, Jonathan de. **Implementação de um sistema de transferência de dados de ultrassom via Ethernet e processamento embarcado em dispositivo FPGA.** 2020. 108 f. Dissertação (Mestrado em Sistemas de Energia) – Programa de Pós-Graduação em Sistemas de Energia (PPGSE), Universidade Tecnológica Federal do Paraná (UTFPR). Curitiba, 2020.

A geração de imagens por ultrassom para auxílio ao diagnóstico, caracterização e avaliação não destrutiva de materiais exige alto poder de processamento digital de sinais e capacidade de transferência massiva de dados. Na Universidade Tecnológica Federal do Paraná, o Grupo de Pesquisa do Ultrassom vem atuando no desenvolvimento da plataforma de pesquisa ULTRA-ORS (do inglês Ultrasound Open Research System) que, apesar de adequada às exigências de flexibilidade e acesso ao fluxo de dados de ultrassom, excede o tempo de transferência e processamento para geração de imagens em tempo real. Neste trabalho, propôs-se a implementação de um sistema de transferência de sinais brutos de ultrassom em alta velocidade com processamento para reconstrução de imagem Modo B embarcado em arquitetura de hardware reconfigurável, objetivando reduzir ambos os tempos de transferência e computação. O sistema proposto é formado por duas placas de desenvolvimento com FPGA: DE2-115 e DE4-230 (Terasic Inc., Taiwan). A primeira placa tem a função de montar e transmitir os pacotes de dados via rede Ethernet. A segunda placa recebe os pacotes, extrai os dados de ultrassom e realiza o processamento para reconstrução da imagem usando a técnica beamforming Delay and Sum (DAS) para as aberturas de 8 e 32 elementos ativos. Além do software Quartus Prime e da ferramenta Qsys, utilizou-se a biblioteca DSP Builder no ambiente Simulink para modelagem, simulação e transformação dos seguintes modelos em linguagem de descrição de hardware: filtro FIR, atraso variável, apodização, somatório coerente, demodulação com detecção de envoltória baseada em aproximação da Transformada de Hilbert e compressão logarítmica. Para a implementação e avaliação dos algoritmos das duas placas foram utilizados dados brutos de ultrassom, adquiridos pelo sistema ULTRA-ORS com frequência de amostragem de 40 MHz e resolução de 12 bits, usando um transdutor convexo de 128 elementos em um phantom de ultrassom. O resultado experimental do processamento foi transferido para um computador para posterior conversão de varredura e apresentação em monitor. Para as avaliações quantitativa e qualitativa da acurácia dos métodos apresentados foi empregada a função de custo da raiz quadrada do erro quadrático médio normalizado (NRMSE) em comparação com as mesmas funções implementadas através de script no Matlab, simulação no Simulink e imagens reconstruídas. Por fim, aplicou-se a razão de contraste (CR) e razão de contraste-ruído (CNR) para avaliação das imagens geradas. Todos os resultados apresentaram excelente concordância com NRMSE inferior a 10%. Os erros do CR e CNR foram inferiores a 5,7% e 6,8%, respectivamente. Houve um avanço significativo no tempo de processamento máximo, passando de 30 minutos para 9 segundos. No entanto, novos estudos são necessários para possibilitar a geração de imagens por ultrassom em tempo real.

Palavras-chave: Sistema de Pesquisa. Ultrassom. Ethernet. Processamento Digital de Sinais. FPGA.

ABSTRACT

OLIVEIRA, Jonathan de. **Implementation of a digital signal transfer and processing system for generating ultrasound images based on FPGA.** 2020. 108 f. Dissertação (Mestrado em Sistemas de Energia) – Programa de Pós-Graduação em Sistemas de Energia (PPGSE), Universidade Tecnológica Federal do Paraná (UTFPR). Curitiba, 2020.

The generation of ultrasound images to aid in the diagnosis, characterization and non-destructive evaluation of materials requires high power of digital signal processing and capacity for massive data transfer. In order to circumvent the typically closed architecture of most commercial ultrasound equipment, as well as optimize resources and improve the quality of the images generated by the technique, several research groups have presented new approaches to ultrasound imaging. At the Federal Technological University of Paraná, the Ultrasound Research Group has been working on development of the open research platform ULTRA-ORS (Ultrasound Open Research System) which, although is suitable for the requirements of flexibility and ultrasound dataflow, it exceeds the data transfer and processing time to generate images in real time. In this work, we propose the implementation of a system for transferring raw ultrasound signals with high speed processing for reconstruction of B Mode image in reconfigurable hardware architecture, aiming at a reduction at the time required for transfer and computation. The proposed system consists of two FPGA development boards: Terasic DE2-115 and DE4-230 (Terasic Inc., Taiwan). The first board has the function of assembling and transmitting data packets via an Ethernet network using the TCP/IP communication protocol. The second board receives the packets and performs the processing for image reconstruction using the beamforming Delay and Sum (DAS) technique for 8 and 32 active aperture elements. In addition to the Quartus Prime software and the Qsys tool, the DSP Builder library was used in the Simulink environment for modeling, simulation and transformation of the following models in hardware description language (HDL): Finite Impulse Response (FIR) filter, variable delay, apodization, coherent summation, demodulation with envelope detection based on the Hilbert Transform approximation and logarithmic compression. For the algorithm implementation and evaluation, within two FPGA devices, raw ultrasound data were used, acquired by the ULTRA-ORS system with a sampling frequency of 40 MHz and 12-bits resolution, using a convex transducer of 128 elements in an ultrasound phantom. The experimental result of the processing was transferred to a computer for later scan conversion and display on monitor. For the quantitative and qualitative assessments of the accuracy of the presented method, the cost function of the square root of the normalized mean square error (NRMSE) was used in comparison with the same functions implemented through a script in Matlab, simulation in Simulink and reconstructed images. Finally, the contrast ratio (CR) and contrast-noise ratio (CNR) were applied to evaluate the generated images. All results showed excellent agreement with NRMSE below 10%. The CR and CNR errors were less than 5.7% and 6.8%, respectively. There has been a significant improvement in the maximum processing time, going from 30 minutes to 9 seconds. However, further studies are needed to enable the generation of ultrasound images in real-time.

Keywords: Research System. Ultrasound. Ethernet. Digital Signal Processing. FPGA.

LISTA DE FIGURAS

Figura 1 – Plataforma de pesquisa ULTRA-ORS.	23
Figura 2 – Definição de dimensões e parâmetros de um transdutor multielemento.....	26
Figura 3 – Transdutor convexo AT3C52B.	27
Figura 4 – Representação da geração de ondas ultrassônicas pelo método pulso-eco para determinação de parâmetros e caracterização de diferentes meios.	28
Figura 5 – Exemplo de excitação de um grupo de elementos piezoelétricos de um transdutor matricial convexo para geração de abertura focal sintética.	29
Figura 6 – Exemplo de perfil de campo acústico focalizado gerado a partir da excitação de um grupo de 32 elementos piezoelétricos de um transdutor matricial linear. A primeira <i>scanline</i> é formada pelos elementos 1 a 32 e a última <i>scanline</i> é formada pelos elementos 97 a 128.....	30
Figura 7 – 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.	31
Figura 8 – Modelo de estrutura de filtro digital FIR.	33
Figura 9 – Exemplo de comparação das técnicas de janela de ponderação Retangular (contínuo) e Gaussiana (tracejado). (a) Respostas normalizadas para abertura de 8 e (b) 32 elementos ativos e respectivas respostas em frequência normalizada (c-d).....	34
Figura 10 – Diagrama em blocos do detector de envelope baseado em uma aproximação da Transformada de Hilbert com filtro FIR.	36
Figura 11 – Exemplo de detecção de envoltória com a aplicação da Transformada de Hilbert.	37
Figura 12 – Arquitetura da Transformada Hilbert com filtro FIR explorando a simetria negativa e os coeficientes nulos intercalados.....	37
Figura 13 – Transformação de coordenada polar para coordenada cartesiana.....	38
Figura 14 – Exemplo de formação de imagem em Modo A e Modo B.	39
Figura 15 – Composição do <i>frame</i> Ethernet.....	43
Figura 16 – Exemplo de diagrama da execução de uma estrutura do tipo FIFO.	45
Figura 17 – Diagrama em blocos do sistema de transferência e processamento de sinais US proposto.	47
Figura 18 – Conjunto formado pelo transdutor multielemento convexo AT3C52B, suporte de acrílico de fixação das partes e <i>phantom</i> de US para captura dos sinais de eco.	47
Figura 19 – Fluxograma de desenvolvimento e integração do projeto no Quartus.....	49

Figura 20 – Placa de desenvolvimento DE2-115.	50
Figura 21 – Fluxograma do fluxo de dados entre as FPGAs Tx e Rx.	52
Figura 22 – Placa de desenvolvimento DE4-230.	53
Figura 23 – Resposta ao impulso do filtro FIR passa-baixa com ordem 15.	55
Figura 24 – Resposta em magnitude do filtro FIR passa-baixa com ordem 15.	55
Figura 25 – Resposta em fase do filtro FIR passa-baixa com ordem 15.	55
Figura 26 – Projeto do Filtro FIR no Simulink utilizando a biblioteca DSP Builder.	56
Figura 27 – Conexão dos blocos da estrutura “Filtro Passa-Baixa” no Simulink.	57
Figura 28 – Perfil temporal de atraso de focalização para a abertura de 8 elementos e profundidade de 25 mm.	58
Figura 29 – Perfil de focalização com ciclos de <i>clock</i> de 40 MHz para a abertura de 8 elementos.	58
Figura 30 – Perfil temporal de atraso de focalização a abertura de 32 elementos e profundidade de 25 mm.	58
Figura 31 – Perfil de focalização com ciclos de <i>clock</i> de 40 MHz para a abertura de 32 elementos.	59
Figura 32 – Blocos de “Memory_Delay” em ciclos de <i>clock</i> de 40 MHz.	59
Figura 33 – Conexão do bloco da estrutura “Atraso Variável” no Simulink.	60
Figura 34 – Modelo de apodização para as janelas Retangular, Gaussiana, Hamming e Hanning.	60
Figura 35 – Modelo de escolha de janelamento através de multiplexadores para o projeto com 8 elementos de abertura.	61
Figura 36 – Conexão do bloco da estrutura “Apodização” no Simulink.	62
Figura 37 – Modelo de multiplicador para etapa de apodização.	62
Figura 38 – Subsistema do bloco “Somatório Coerente”.	63
Figura 39 – Conexão do bloco da estrutura “Somatório Coerente” no Simulink.	63
Figura 40 – Subsistema do bloco “Demodulação”.	64
Figura 41 – Bloco da estrutura “Hilbert Transform”.	65
Figura 42 – Bloco de detecção envelope.	65
Figura 43 – Conexão do bloco da estrutura “Demodulação” no Simulink.	66
Figura 44 – Tabela de compressão logarítmica limitada em -30 dB.	67
Figura 45 – Subsistema de compressão logarítmica.	67
Figura 46 – Valores normalizados correspondentes à compressão logarítmica limitada em -30 dB.	68

Figura 47 – Bloco de compressão logarítmica.	68
Figura 48 – Fluxograma de coordenação de dados.	69
Figura 49 – Blocos de fluxo de dados pré-processamento.	70
Figura 50 – Subsistema composto por 8 memórias FIFO.	71
Figura 51 – Blocos de leitura de <i>words</i> das FIFOs.....	71
Figura 52 – Bloco de habilitação do processamento de dados.	72
Figura 53 – Bloco de leitura de dados pós-processamento.	72
Figura 54 – Bancada de desenvolvimento e teste formada por um notebook, placas DE2-115 e DE4-230, e um monitor LCD.	74
Figura 55 – Comparação entre as respostas ao impulso do filtro digital FIR passa-baixa implementado via <i>script</i> e modelado no Simulink utilizando os métodos: (a) EQ com 8 coeficientes; (b) LS com 8 coeficientes.....	75
Figura 56 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via <i>script</i> e modelado no Simulink utilizando os métodos: (a) EQ com 16 coeficientes; (b) LS com 16 coeficientes.	76
Figura 57 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via <i>script</i> e modelado no Simulink utilizando os métodos: (a) EQ com 24 coeficientes; (b) LS com 24 coeficientes.	76
Figura 58 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via <i>script</i> e modelado no Simulink utilizando os métodos: (a) EQ com 32 coeficientes; (b) LS com 32 coeficientes.	76
Figura 59 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via <i>script</i> e modelado no Simulink utilizando os métodos: (a) EQ com 64 coeficientes; (b) LS com 64 coeficientes.	77
Figura 60 – Comparação quantitativa da implementação do filtro digital FIR com variação do número de <i>bits</i> (<i>b</i>) da parte fracionária dos coeficientes (<i>N</i>) para os métodos: (a) EQ e (b) LS.	77
Figura 61 – Avaliação da resposta do filtro FIR com aplicação de um sinal senoidal de 1MHz adicionado ao sinal senoidal de 8 MHz (25 % de amplitude). (a) Sinal de entrada do filtro. (b) Espectro em frequência do sinal de entrada. (c) Sinal de saída filtrado. (d) Espectro em frequência do sinal filtrado.	78
Figura 62 – Avaliação da resposta do filtro FIR com aplicação de um sinal Chirp de 1 a 10 MHz. (a) Sinal de entrada do filtro. (b) Espectro em frequência do sinal de entrada. (c) Sinal de saída filtrado. (d) Espectro em frequência do sinal filtrado.	79

Figura 63 – Avaliação da resposta do filtro FIR com aplicação de um sinal real de RF adicionado a uma senoide de 8 MHz. (a) Sinal de entrada do filtro. (b) Espectro em frequência do sinal de entrada. (c) Sinal de saída filtrado. (d) Espectro em frequência do sinal filtrado.	80
Figura 64 – Comparação entre as respostas ao impulso do detector de envoltória implementado via <i>script</i> e modelado no Simulink utilizando os métodos: (a) EQ com 33 coeficientes; (b) LS com 33 coeficientes.	81
Figura 65 – Exemplo de comparação de resultados entre os métodos de detecção de envoltória utilizando uma aproximação da Transformada de Hilbert.	81
Figura 66 – Ampliação da marcação indicada na Figura 65 para melhor visualização da comparação dos resultados de detecção de envoltória.	82
Figura 67 – Comparação quantitativa da implementação do detector de envoltória baseado na aproximação da Transformada de Hilbert com filtro FIR para os métodos EQ e LS. (a) Avaliação de b variando de 4 a 17 com $N=32$. (b) Avaliação de N variando de 8 a 64 e $b=17$	82
Figura 68 – Comparação de custo computacional simulado e experimental para N variando de 8 a 64. (a) Tempo de computação. (b) Latência em ciclos de <i>clock</i> da FPGA.....	83
Figura 69 – Sinal de RF para comparação da aplicação do atraso variável entre o <i>script</i> do Matlab e o modelo do Simulink. (a) Sinais referentes à primeira linha da <i>scanline</i> 58 sem processamento. (b) Comparação dos sinais após a etapa de atraso variável.	84
Figura 70 – Comparação entre os espectros em frequência dos sinais gerados pelo <i>script</i> do Matlab e o modelo do Simulink após a etapa de atraso variável.....	85
Figura 71 – Resultados da apodização com janela Gaussiana. (a) Sinais antes e (b) sinais após a apodização.	85
Figura 72 – Exemplo de resultados de somatório coerente. (a) Sinal de 8 canais pré-somatório com apodização Retangular. (b) Sinal normalizado após a etapa de somatório coerente.	86
Figura 73 – Comparação entre os espectros em frequência dos sinais gerados pelo <i>script</i> do Matlab e o modelo do Simulink após a etapa de somatório coerente.....	87
Figura 74 – Sinal de envelope gerado pelo modelo do Simulink.....	88
Figura 75 – Comparação entre os sinais de envelope gerados pelo <i>script</i> do Matlab e o modelo do Simulink.	88
Figura 76 – Comparação entre os sinais após a etapa de compressão logarítmica gerados pelo <i>script</i> do Matlab e modelo do Simulink.	89
Figura 77 – Imagem final Modo B gerada pelo sistema de processamento digital de sinais embarcado em <i>hardware</i> pelo método DAS com abertura ativa de 8 elementos.	90

Figura 78 – Imagem final Modo B gerada pelo sistema de processamento digital de sinais embarcado em <i>hardware</i> pelo método DAS com abertura ativa de 32 elementos.	90
Figura 79 – Imagem exemplo de indicação dos alvos para as análises de contraste CR e CNR.	91
Figura 80 – Imagens de US Modo B para análise de contraste geradas por simulações a partir do <i>script</i> e modelo Simulink e de forma experimental por FPGA com abertura ativa de 8 elementos (a,c,e) e 32 elementos (b,d,f).	92
Figura 81 – Resultados do contraste para os alvos A, B, C e D e aberturas ativas de 8 e 32 elementos. (a) Dados do CR. (b) Erro percentual do CR. (c) Dados do CNR. (d) Erro percentual do CNR.	94

LISTA DE TABELAS

Tabela 1 – Principais parâmetros da geometria de um transdutor de US.....	26
Tabela 2 – Especificação de requisitos do sistema proposto.....	46
Tabela 3 – Abertura ativa <i>versus</i> quantidade de memória para um transdutor de 128 elementos.	50
Tabela 4 – Tamanho da variável “ <i>data</i> ” para armazenamento dos dados processados.	51
Tabela 5 – Especificações do Filtro FIR.	54
Tabela 6 – Coeficientes do filtro digital FIR passa-baixa pelo método Equiripple.	54
Tabela 7 – Coeficientes do filtro FIR digital usando o método de projeto com a Transformada de Hilbert.	64
Tabela 8 – Comparação dos parâmetros de contraste CR e CNR para os alvos A, B, C e D. .	93
Tabela 9 – Comparação de custo computacional dos recursos de <i>hardware</i> entre as placas DE2-115 e DE-230.....	96

LISTA DE ABREVIATURAS E SIGLAS

A	Amplitude
ADC	<i>Analog-to-Digital Converters</i>
ASIC	<i>Application-Specific Integrated Circuits</i>
B	Brilho
bps	<i>Bits por segundo</i>
CI	Circuito Integrado
CNR	<i>Constrast-to-Noise Ratio</i>
CP	Compressão Logarítmica
CR	<i>Constrast Ratio</i>
CRC	<i>Cyclic Redundancy Check</i>
DAS	<i>Delay-and-Sum</i>
DDR2	<i>Double Data Type 2</i>
DMA	<i>Direct Memory Access</i>
DSP	<i>Digital Signal Processor</i>
E/S	Entrada/Saída
EMIF	<i>External Memory Interface</i>
EN	<i>Enable</i>
EQ	Equiripple
FCS	<i>Frame Check Sequence</i>
FFT	<i>Fast Fourier Transfom</i>
FIFO	<i>First In, First Out</i>
FIR	<i>Finite Impulse Response</i>
Fpass	Banda passante
FPGA	<i>Field Programmable Gate Array</i>
FS	Frequência de Amostragem
Fstop	Banda de parada
GE	General Electric
GPU	<i>Graphical Processing Unit</i>
HDL	<i>Hardware Description Language</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>

IIR	<i>Infinite Impulse Response</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
LS	Least-Squares
LUS	Laboratório de Ultrassom
LUT	<i>Look-Up Table</i>
MAC	<i>Medium Access Control</i>
NDE	<i>Non-Destructive Evasion</i>
NRMSE	<i>Normalized Root Mean Square Error</i>
OSI	<i>Open System Interconnection</i>
OUI	<i>Organizationally Unique Identifier</i>
PC	Computador Pessoal
PLD	<i>Programmable Logic Device</i>
PLL	<i>Phase-Locked Loop</i>
PVDF	Difluoreto de Polivinilideno
PZT	Titanato Zirconato de Chumbo
RAM	<i>Random Access Memory</i>
RASMUS	<i>Remotely Accessible Software Configurable Multichannel Ultrasound Sampling</i>
RF	Radiofrequência
ROI	<i>Region of Interest</i>
Rx	Recepção
SDRAM	<i>Synchronous Dynamic RAM</i>
SO-DIMM	<i>Dual In-line Memory Module</i>
SoC	<i>System on Chip</i>
SUS	Sistema Único de Saúde
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TH	Transformada de Hilbert
Tx	Transmissão
ULA-OP	<i>ULtrasound Advanced Open Platform</i>
US	Ultrassom
UTFPR	Universidade Tecnológica Federal do Paraná
VHDL	<i>Very-High Integrated Circuit HDL</i>

LISTA DE SÍMBOLOS

a_i	Coefficiente do filtro digital
A_r	Onda refletida
A_t	Onda transmitida
b	Número de <i>bits</i> da parte fracionária de um número
c	Velocidade de propagação da onda acústica no meio
d	Distância percorrida pela onda ultrassônica
E	Envelope
$hy(n)$	Valor de referência obtido através do <i>script</i> do Matlab
I	Sinal em fase
m	Índice do número de canais
M	Número de canais
M	Número de amostras
n	Índice das amostras
N	Número de coeficientes
n	Índice de amostras
na	Número de elementos ativos da abertura aplicada
ne	Número total de elementos do transdutor
N_{sc}	Número de <i>scanlines</i>
Q	Sinal em quadratura
r_f	Resultado na saída do formador do feixe
t	Tempo decorrido entre o pulso ultrassônico e recepção do eco
w_m	Peso de ponderação com valor entre 0 e 1
$x(n)$	Sinal de entrada
x_c	Posição do centro de um elemento transdutor no eixo lateral
X_{db}	Sinal após a detecção de envoltória em escala logarítmica
x_f	Posição do ponto focal no eixo lateral
x_m	Sinal de entrada registrado no número de elementos m
x_{max}	Amplitude máximo do sinal de entrada
$y(n)$	Sinal de saída processado
$Y(n)$	Amostra do modelo

y_c	Posição do centro de um elemento transdutor no eixo elevacional
Δ_m	Índice de deslocamento da amostra referente ao número de canais
\bar{h}_y	Média das amostras do modelo implementado no Simulink ou na FPGA
μ_b	Média da intensidade da região fundo da imagem
μ_t	Intensidade do alvo em dB
y_f	Posição do ponto focal no eixo elevacional
z	Espaço equivalente a 2 vezes a distância entre o transdutor e o alvo
z_c	Posição do centro de um elemento transdutor no eixo axial
z_f	Posição do ponto focal no eixo axial
σ_b	Desvio padrão da intensidade da região de fundo da imagem

SUMÁRIO

1. INTRODUÇÃO	20
1.1 MOTIVAÇÃO.....	22
1.2 OBJETIVOS.....	24
1.3 ESTRUTURA DO TRABALHO	24
2 FUNDAMENTAÇÃO TEÓRICA.....	25
2.1 TRANSDUTOR DE ULTRASSOM.....	25
2.2 MÉTODO PULSO-ECO	27
2.3 ABERTURA FOCAL SINTÉTICA.....	28
2.4 PROCESSAMENTO DE SINAL PELA TÉCNICA BEAMFORMING	30
2.4.1 Filtragem Digital	32
2.4.2 Atraso Variável de Recepção	33
2.4.3 Apodização	33
2.4.4 Somatório Coerente	35
2.4.5 Demodulação	35
2.4.5.1 Transformada de Hilbert.....	36
2.4.6 Compressão Logarítmica.....	37
2.4.7 Conversão de Varredura	38
2.4.7.1 Imagem em Modo B	39
2.4.8 Imagem em Tempo Real.....	40
2.5 FPGA	40
2.6 MÉTODOS DE TRANSFERÊNCIA DE DADOS.....	41
2.6.1 Ethernet.....	42
2.6.1.1 <i>Frame</i> e Transmissão.....	43
2.6.1.2 Preâmbulo	43
2.6.1.3 Delimitador de <i>Frame</i> Inicial.....	43
2.6.1.4 Endereço de Destino	43
2.6.1.5 Endereço de Origem	44
2.6.1.6 Tipo ou Tamanho	44
2.6.1.7 Dados	44
2.6.1.8 Sequência de Checagem	45
2.7 ESTRUTURA DE DADOS	45

3	MATERIAIS E MÉTODOS.....	46
3.1	ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA PROPOSTO	46
3.2	SOFTWARE DE MODELAGEM – MATLAB/SIMULINK.....	48
3.3	<i>SOFTWARE</i> QUARTUS PRIME.....	48
3.4	PLACA DE2-115 – FPGA TX.....	49
3.4.1	Protocolo de Comunicação via Rede Ethernet	51
3.5	PLACA DE4-230 – FPGA RX E PROCESSAMENTO.....	52
3.6	ETAPAS DE PROCESSAMENTO IMPLEMENTADAS.....	53
3.6.1	Modelagem do Filtro Digital	53
3.6.2	Modelagem do Bloco de Atraso Variável	57
3.6.3	Modelagem do Bloco de Apodização.....	60
3.6.4	Modelagem do Bloco de Somatório Coerente.....	63
3.6.5	Modelagem do Bloco de Demodulação.....	64
3.6.6	Modelagem do Bloco de Compressão Logarítmica	66
3.6.7	Integração do Projeto do DSP Builder com a Ferramenta Qsys.....	68
3.7	FERRAMENTAS PARA AVALIAÇÃO DE SINAIS E IMAGENS	73
4	RESULTADOS E DISCUSSÕES	75
4.1	AVALIAÇÃO DA FILTRAGEM DIGITAL PASSA-BAIXA	75
4.2	AVALIAÇÃO DA DEMODULAÇÃO E DETECÇÃO DE ENVOLTÓRIA	80
4.3	AVALIAÇÃO DAS ETAPAS DE PROCESSAMENTO	83
4.3.1	Etapa de Atraso Variável em Sinal Bruto de RF.....	84
4.3.2	Etapa de Apodização	85
4.3.3	Etapa de Somatório Coerente	86
4.3.4	Etapa de Demodulação e Detecção de Envoltória.....	87
4.3.5	Etapa de Compressão Logarítmica	88
4.3.6	Etapa de Conversão de Varredura	89
4.4	ANÁLISE DE CONTRASTE CNR E CR	91
4.5	TEMPO DE PROCESSAMENTO E CUSTO COMPUTACIONAL.....	95
5	CONCLUSÃO.....	98
5.1	TRABALHOS FUTUROS	98
	REFERÊNCIAS	100
	APÊNDICE A – PROJETO DO SISTEMA DE PROCESSAMENTO PARA GERAÇÃO	
	DE IMAGEM POR ULTRASSOM NO SIMULINK	108

1. INTRODUÇÃO

O ultrassom (US) baseia-se em ondas sonoras que excedem a capacidade de audição humana, possuindo frequência superior a 20 kHz. As técnicas de geração de ondas ultrassônicas têm sido investigadas e desenvolvidas desde o início do século 20, a partir das pesquisas realizadas com o sonar marítimo (SZABO, 2004; BUSHBERG; BOONE, 2011), e, atualmente, apresenta-se como uma das principais e mais seguras modalidades de geração de imagem para auxílio ao diagnóstico, caracterização e avaliação não destrutiva de materiais (*Non-Destructive Evaluation* - NDE).

Na transmissão, o método é caracterizado pela emissão de ondas acústicas que se propagam e interagem com o meio investigado. Tipicamente, é usado um transdutor com propriedades piezoelétricas que é excitado por um pulso elétrico de alta tensão (até centenas de volts) para gerar uma deformação mecânica que, conseqüentemente, produz a onda acústica que se propaga no meio. Na recepção, o mesmo ou outro elemento transdutor é usado para captar os sinais de ecos refletidos e espalhados, conforme os modos pulso-eco ou transmissão-recepção (Tx-Rx), respectivamente. Esses ecos, também denominados de sinais de radiofrequência (RF), são gerados devido a não homogeneidade e presença de estruturas com diferentes propriedades acústicas do meio, como por exemplo, impedância acústica, velocidade de propagação e atenuação, dentre outros (FISH, 1990; SHUNG, 2006; ASSEF, 2007).

A aplicação do US na medicina, passa a ter um uso para detecção de hematomas intracerebrais, tendo aplicação para detecção de tumores cerebrais algumas décadas mais tarde. A partir de 1960, impulsionado pelas características não-ionizante, não-invasiva e relativa facilidade e segurança associadas à sua utilização, houve um grande progresso na técnica de auxílio ao diagnóstico médico por US, passando de um simples escaneamento para aplicações e imagens compostas através da utilização de circuitos eletrônicos complexos. Nesse período, o projeto dos equipamentos, técnicas de aquisição e capacidade de processamento de dados também evoluíram significativamente, surgindo os transdutores eletrônicos vetoriais e a geração de imagens em tempo real (BUSHBERG; BOONE, 2011).

Pelos mesmos motivos, o US ganhou abrangência em diversas áreas industriais, nas quais o método pulso-eco tem sido usado para caracterização de materiais e meios, além de aplicações de manutenção preventiva, através da técnica de NDE. Dessa forma, é possível localizar defeitos e trincas, avaliar a porosidade, medir tamanhos, formatos, espessuras e orientação de falhas. Como exemplo, pode-se citar as indústrias automotiva, aeroespacial,

nuclear, aeronáutica, biomédica, elétrica, dentre outras (PAYNE,1985; MACHADO *et al.*, 2011).

Um dos maiores avanços na modalidade de imageamento por US foi a introdução do feixe eletrônico direcionado utilizando tecnologia com vetores de fase, sendo apenas em meados de 1970 que os equipamentos de imageamento por US passaram a ser fabricados por diversas companhias (SUETENS, 2009). Atualmente, os sofisticados sistemas de US para auxílio ao diagnóstico, caracterização e avaliação não destrutiva de materiais demandam alto poder de processamento digital de sinais e capacidade de transferência massiva de dados. Tais características são essenciais para possibilitar a geração de imagens, em tempo real, de estruturas anatômicas internas e fluxo sanguíneo em diferentes modos, sendo os principais: Modo A (Amplitude), Modo B (Brilho), Modo M, Modo Doppler Contínuo, Modo Doppler Pulsátil e Modo Elastográfico (ASSEF, 2013).

Devido à complexidade eletrônica, tipicamente a arquitetura desses equipamentos é baseada em Circuitos Integrados de Aplicação Específica (*Application Specific Integrated Circuits* - ASICs), Arranjo de Portas Programáveis em Campo (*Field Programmable Gate Array* - FPGAs), Processadores Digitais de Sinais (*Digital Signal Processors* - DSPs) com diversos núcleos (SIKDAR *et al.*, 2003; SCHNEIDER *et al.*, 2010; BONI *et al.*, 2012; ASSEF, 2013) e, mais recentemente, Unidades Gráficas de Processamento (*Graphical Processing Unit* - GPU) (BONI *et al.*, 2018).

Apesar da alta tecnologia envolvida e da capacidade de geração de imagem em diferentes modos, a maioria dos sistemas comerciais de imagem por US possui arquitetura típica “fechada”, não atendendo às exigências de flexibilidade e acesso aos dados de RF necessárias ao desenvolvimento e teste de novas modalidades e técnicas. Por outro lado, plataformas “abertas” de pesquisa do US, desenvolvidas tipicamente por empresas internacionais, como o OPEN SYSTEM *Phased array device* (Lecoeur Electronique, França), o Ultrasonix 500RP (WILSON *et al.*, 2006), o Sonix MDP (Ultrasonix Medical Corp., Canadá) e o Vantage Research System (Verasonics Inc., Estados Unidos da América) (ASSEF *et al.*, 2016) ainda são muito caras e dependentes de taxas de importação, tornando-as, muitas vezes, financeiramente inacessíveis à maioria dos estudantes e pesquisadores brasileiros.

Assim, diversos centros de pesquisa têm apresentado soluções de sistema “abertos” para suportar mais funções de processamento paralelo (CHANG-HONG HU *et al.*, 2008; HASSAN *et al.*, 2011; SANIIE & ORUKLU, 2012; BONI *et al.*, 2016; BONI *et al.*, 2018), possibilitando também uma melhor adequação do *hardware* às aplicações de *software* e permitindo a exploração de novas estratégias em função da aplicação a ser executada

(HAZARD & LOCKWOOD, 2000; PEI-JIE CAO *et al.*, 2002; SIKDAR *et al.*, 2003). Como exemplo, pode-se citar as plataformas ULA-OP 256 (ULtrasound Advanced Open Platform) (BONI *et al.*, 2016) e o Remotely Accessible Software Configurable Multichannel Ultrasound Sampling (RASMUS) (JENSEN; MUNK, 1998), sendo o último extremamente complexo em relação ao número de placas e processadores.

Tais ações têm sido apoiadas pela sociedade que espera que os laboratórios de pesquisa de US, especialmente em universidades, proponham novas abordagens para, por exemplo, transmissão/recepção ultrassônica, métodos de processamento de sinal, transferência de dados, formação de imagem e investigações teóricas ou procedimentos experimentais, contribuindo para um diagnóstico mais simples e confiável.

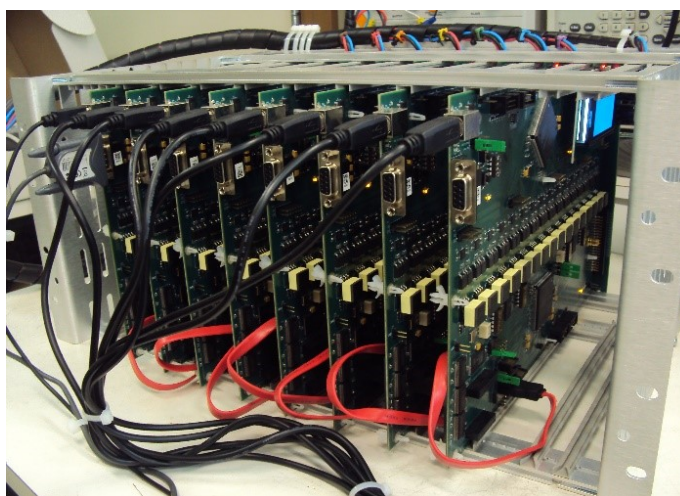
1.1 MOTIVAÇÃO

Apesar de existirem diversos fabricantes de sistemas comerciais de imageamento para auxílio ao diagnóstico médico por US, todos eles são importados, o que demanda um elevado custo para o Sistema Único de Saúde (SUS) para importação de equipamentos, transdutores, *softwares* aplicativos e peças de manutenção, elevando o déficit na balança comercial. Soma-se a isso que o setor de equipamentos de ultrassom, bem como o de outras técnicas de diagnóstico por imagem no Brasil, não deveria ficar sujeito às oscilações do mercado financeiro internacional e refém de estratégias competitivas descoladas do interesse nacional. Por outro lado, existe um grande potencial para desenvolver o setor de equipamentos de imagem médica no âmbito da indústria, incluindo empresas de *software* e partes mecânicas, integrando inúmeras tecnologias ao mesmo tempo, o que incrementaria a inovação e capacitação de profissionais.

Um importante passo foi dado pelo Grupo de Pesquisa do Ultrassom da Universidade Tecnológica do Paraná (UTFPR) que vem desenvolvendo a plataforma de pesquisa aberta ULTRASound Open Research System (ULTRA-ORS), apresentada na Figura 1. A construção da plataforma com tecnologia nacional visa apoiar o desenvolvimento de pesquisas e técnicas inovadoras que possam no futuro atender uma maior parcela da população brasileira, promovendo ao mesmo tempo capacitação e formação de recursos humanos qualificados. A plataforma é composta por uma arquitetura modular de *hardware front-end*, baseada em dispositivos FPGA para controle dos segmentos Tx e Rx, e de *software* para processamento *back-end* baseado em computador pessoal (PC). O sistema é formado por um rack 5U que aloja 8 placas Tx/Rx com 16 canais cada, placa *backplane*, na qual é posicionado um soquete ZIF para inserção de um transdutor matricial de até 128 elementos, e um PC. Cada placa Tx/Rx é

controlada por duas FPGA Intel da família Cyclone III. No segmento Tx, o ULTRA-ORS possui 128 canais com capacidade de geração de formas de onda arbitrária de US até 200 V_{pp} com abertura programável (4, 8, 16 e 32 elementos ativos). O segmento Rx é formado por 128 canais com taxa de amostragem de até 50 MHz e resolução de 12 *bits* (ASSEF, 2013; FERREIRA, 2017). Após a etapa de digitalização no segmento Rx, os sinais são enviados para o PC, através de uma porta serial emulada via USB, que realiza o processamento *back-end* para geração de imagens em Modo B através de algoritmos desenvolvidos no Matlab.

Figura 1 – Plataforma de pesquisa ULTRA-ORS.



Fonte: Assef (2013).

Apesar do ULTRA-ORS ser adequado às exigências de pesquisa das etapas de transmissão multicanal e recepção ultrassônica, o tempo de transferência de dados entre plataforma e o PC adicionado ao tempo de computação pode chegar a 30 minutos (ASSEF, 2013), inviabilizando a pesquisa de técnicas para geração de imagens em tempo real.

Dessa forma, para melhorar o desempenho e mitigar essa limitação, foi implementado neste trabalho de Mestrado um sistema baseado em duas placas de FPGA com comunicação via Ethernet para a avaliação da transferência de dados e processamento embarcado em *hardware* reconfigurável. A primeira placa simula o sistema ULTRA-ORS, atuando na montagem e transferência dos pacotes de dados via rede Ethernet. A segunda placa recebe os pacotes do protocolo de comunicação TCP/IP, extrai os dados de US e realiza o processamento para reconstrução da imagem em Modo B para as aberturas de 8 ou 32 elementos ativos.

1.2 OBJETIVOS

O objetivo principal deste trabalho foi implementar um sistema de transferência de dados brutos de US via rede Ethernet para realização e avaliação das etapas de processamento digital de sinais usando a técnica *beamforming Delay and Sum* (DAS) na geração de imagens de US em Modo B embarcado em FPGA de alto desempenho, com intuito de reduzir o tempo de transferência e computação.

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

- Adotar as placas de FPGA DE2-115 e DE4-230 (Terasic Inc., Taiwan) com interface Ethernet para transmissão e processamento dos sinais de US para reconstrução de imagens, respectivamente;
- Implementar a arquitetura de *hardware* do processador Nios II de 32 *bits* embarcado nas duas placas de FPGA para controle da interface Ethernet e armazenamento de dados brutos e processados;
- Desenvolver os códigos em linguagem C para leitura, transferência e escrita de dados no pacote do protocolo TCP/IP;
- Utilizar a biblioteca DSP Builder no ambiente Simulink para modelagem, simulação e transformação das etapas de processamento em linguagem de descrição de *hardware* (*Hardware Description Language* - HDL);
- Avaliar a acurácia entre os resultados simulados e experimentais;
- Validar os resultados experimentais do processamento embarcado em FPGA.

1.3 ESTRUTURA DO TRABALHO

Esta dissertação está organizada em 5 capítulos: Introdução, Fundamentação Teórica, Materiais e Métodos, Resultados e Discussões e, por fim, Conclusão. O capítulo 1 apresenta uma breve introdução e os objetivos deste trabalho. O capítulo 2 trata da fundamentação teórica referente às etapas de processamento para geração de imagem em tempo real pela técnica pulso-eco, bem como o estudo da interface Ethernet para transferência de dados. O capítulo 3 descreve os materiais e métodos utilizados para o desenvolvimento deste projeto de pesquisa, incluindo a especificação de requisitos, *softwares* utilizados, placas de desenvolvimento e ferramentas de avaliação dos resultados da pesquisa. No capítulo 4 são apresentados e discutidos os resultados da implementação do projeto através de resultados simulados e obtidos de forma experimental. Finalmente, no capítulo 5 são apresentadas as conclusões e sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos utilizados para a realização desta dissertação, sendo eles: transdutor de ultrassom, método pulso-eco, abertura sintética, processamento digital de sinal e imagem Modo B pela técnica *beamforming* DAS, dispositivo FPGA, métodos de transferência e estrutura de dados.

2.1 TRANSDUTOR DE ULTRASSOM

Os transdutores são um dos principais componentes de um sistema de US, sendo responsáveis pela transmissão e recepção dos sinais ultrassônicos. Esses dispositivos são compostos por uma ou mais lâminas de cristal piezoelétrico, responsável por reagir a uma tensão elétrica, atribuindo algumas deformações físicas para o próprio cristal. Esse efeito é denominado piezoeletricidade, o qual foi descoberto pelos irmãos Pierre e Jacques Curie, em 1880. Atualmente, entende-se que essa propriedade é advinda de determinados cristais e substâncias cristalizadas que apresentam anisotropia, sendo essa a capacidade de reagir diferentemente de acordo com a direção de propagação de um fenômeno físico sobre si, como luz, calor, etc. (TER HAAR, 1999).

As ondas ultrassônicas são geradas pela excitação de um sinal elétrico de elevada amplitude nos transdutores piezoelétricos de modo a alterar o tamanho do cristal/cerâmica momentaneamente. Esse processo gera um realinhamento das moléculas polarizadas que aumenta ou diminui a pressão na parede frontal do transdutor produzindo as ondas de pressão. Para a etapa da recepção, basicamente ocorre o processo inverso: o transdutor receptor converte a onda sonora propagada no meio em sinal elétrico (GOLIO, 2008).

Além de cristais, outros materiais piezoelétricos são utilizados na conversão de um sinal elétrico em sinal mecânico e vice-versa, podendo ser produzidos a partir de polímeros e cerâmicas sintetizadas, em especial o PZT (Titanato Zirconato de Chumbo) e o PVDF (Difluoreto de Polivinilideno) (YOUNG, 1990). Esses elementos transdutores podem ser alinhados em conjunto de forma plana ou convexa, sendo que a sua geometria tem um grande impacto sobre a técnica de reconstrução da imagem.

Os principais parâmetros e dimensões físicas de um transdutor multielemento são apresentados na Tabela 1 em conjunto com a Figura 2. Estes parâmetros são utilizados em programas de simulação do campo acústico, como por exemplo, o *software* de simulação de

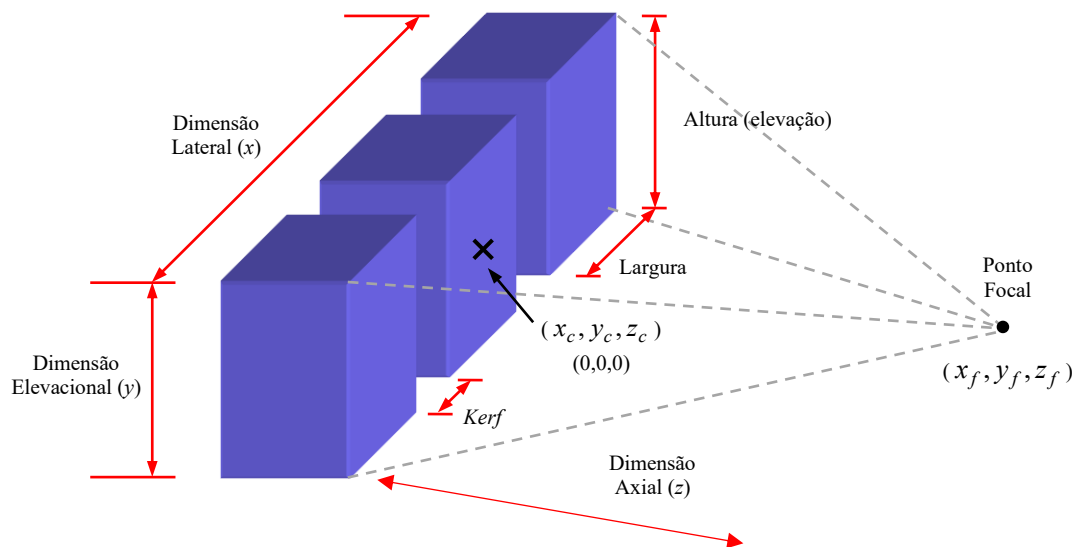
campo acústico Field II (JENSEN, 1996) e a biblioteca K-Wave, ambos para utilização em conjunto com o Matlab.

Tabela 1 – Principais parâmetros da geometria de um transdutor de US.

Parâmetro do transdutor	Unidade	Descrição
<i>Kerf</i>	μm	Espaçamento entre elementos na dimensão lateral
Largura	mm	Largura de um elemento individual
Altura	mm	Altura (elevação) de um elemento individual
Ponto focal	[mm, mm, mm]	É o ponto no espaço na frente da face do transdutor para o qual as ondas produzidas por cada elemento convergem construtivamente. Os eixos são definidos em (lateral, elevacional e axial). O centro do transdutor é definido como ponto (0,0,0)
Número de elementos	-	Número de cristais/cerâmicas empregados na construção do transdutor, tipicamente variando de 64 a 256 elementos

Fonte: Adaptado de Jensen (1996).

Figura 2 – Definição de dimensões e parâmetros de um transdutor multielemento.



Fonte: Adaptado de Jensen (1996).

Além disso, outros parâmetros como o número de elementos, frequência central e largura de banda do transdutor estão diretamente relacionados com a resolução e qualidade da imagem gerada. Por exemplo, conforme Maia (2004), quanto maior a largura de banda, menor será o comprimento espacial do pulso e, conseqüentemente, a resolução axial será melhor.

A maioria dos equipamentos comerciais de US utilizados para auxílio ao diagnóstico médico utilizam transdutores matriciais com 64 a 256 elementos piezoelétricos e frequência central entre 5 e 30 MHz (ASSEF, 2013). Porém, existem pesquisas que utilizam até 1000

elementos (MOROS, 2012). Basicamente, são usados três tipos de transdutores matriciais de US: linear, convexo e *phased array*.

Entre os principais fabricantes de transdutores comerciais utilizados para diagnóstico médico na faixa de frequência entre 5 a 30 MHz destacam-se as empresas Siemens, Philips, General Electric (GE) e Toshiba, dentre outros. Como exemplo, na Figura 3 é apresentado o transdutor convexo de 128 elementos AT3C52B com frequência central de 3,2 MHz, fabricado pela empresa Broadsound Corporation, utilizado para a aquisição de dados deste trabalho.

Figura 3 – Transdutor convexo AT3C52B.

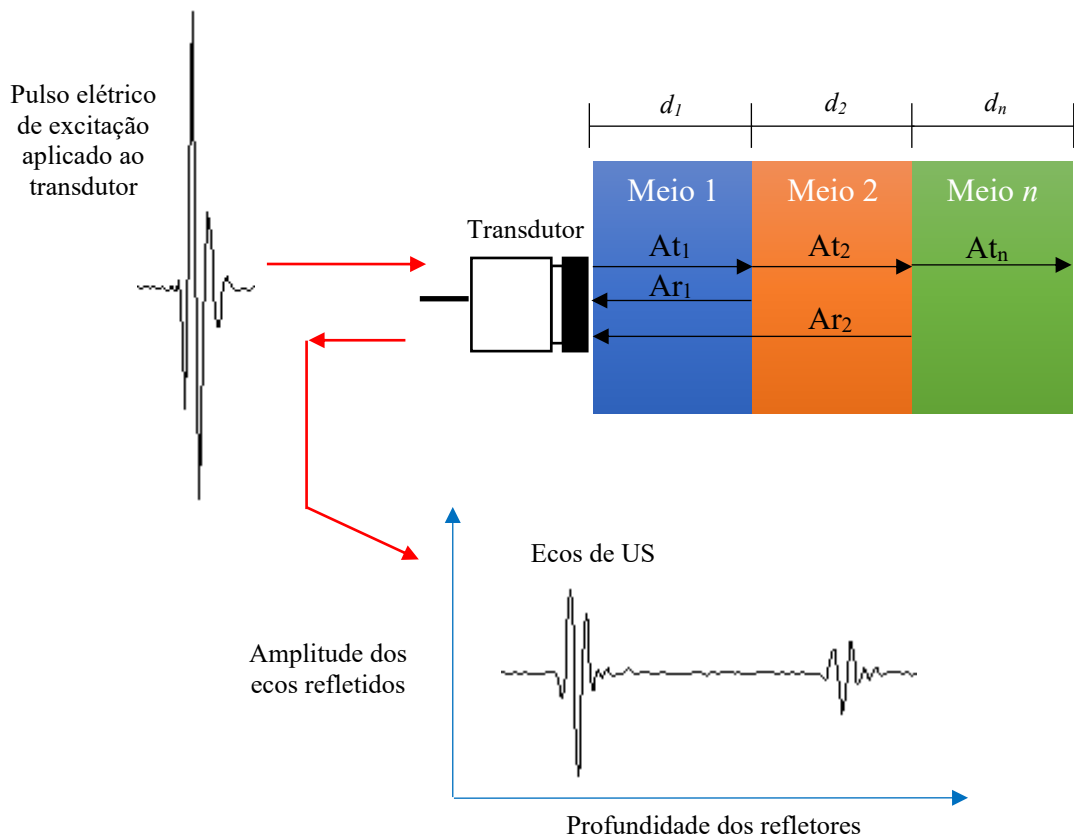


Fonte: Broadsound Corporation (2014).

2.2 MÉTODO PULSO-ECO

No método pulso-eco o mesmo transdutor é usado na transmissão e recepção dos sinais ultrassônicos. Conforme ilustrado na Figura 4, um pulso elétrico de alta tensão, que pode variar de 5 Vpp em aplicações Doppler a 400 Vpp para aplicações industriais, é aplicado ao transdutor, gerando uma onda ultrassônica. Essa onda se propaga através do meio 1 até encontrar a interface entre o meio 1 e o meio 2 com diferentes impedâncias acústicas, onde parte da onda é refletida (sinal Ar_1) e parte se propaga pelo meio 2 (sinal At_2). A onda que se propaga através do meio 2, ao encontrar uma outra interface, terá uma parte refletida (Ar_2) e outra parte transmitida (At_n). Esse processo continua até a onda não ser mais capaz de se propagar no próximo meio, devido principalmente à atenuação e absorção do meio. As ondas refletidas Ar_1 e Ar_2 são captadas pelo transdutor trazendo informações sobre a distância, atenuação e velocidade de propagação (ASSEF, 2006).

Figura 4 – Representação da geração de ondas ultrassônicas pelo método pulso-eco para determinação de parâmetros e caracterização de diferentes meios.



Fonte: Adaptado de Assef (2006).

Conhecendo-se a velocidade de propagação da onda no meio c e tempo t decorrido entre a aplicação do pulso e recepção do eco é possível calcular a profundidade de cada interface refletora, ou a distância percorrida pelo sinal ultrassônico, conforme a Equação (1). Nesse caso, o espaço percorrido z equivalente a 2 vezes a distância entre o transdutor e a interface alvo (HEDRICK *et al.*, 1995).

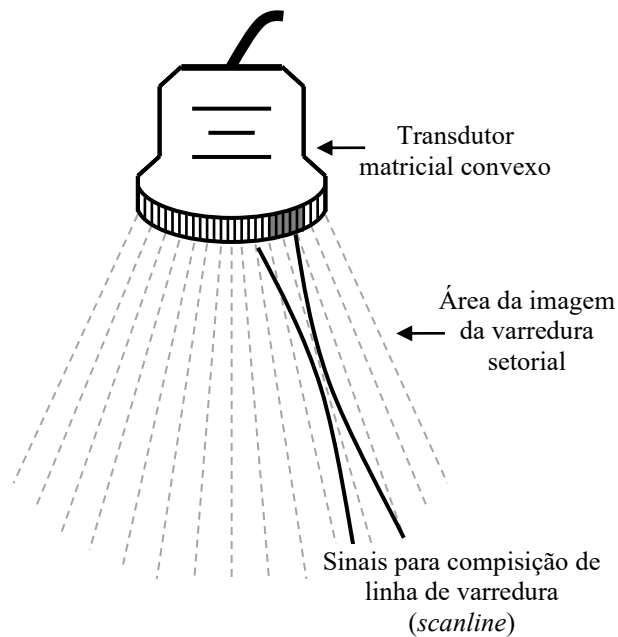
$$c = z/t \quad (1)$$

2.3 ABERTURA FOCAL SINTÉTICA

Considerando os transdutores multielementos, as ondas de US podem ser orientadas, ou mesmo focalizadas em um ponto específico da região de interesse (*Region of Interest - ROI*), uma vez que o meio seja suficientemente homogêneo (ASSEF, 2013). Na Figura 5 é ilustrado

como um transdutor convexo usado para imagem setorial pode focalizar a propagação das ondas ao longo de linhas de varredura (*scanlines*) na região de interesse. Neste caso, um conjunto de elementos piezoelétricos adjacentes são excitados de forma individual ou em grupos com tempos de atraso de disparo programados, sendo utilizado para geração de abertura focal sintética (JENSEN *et al.*, 2006).

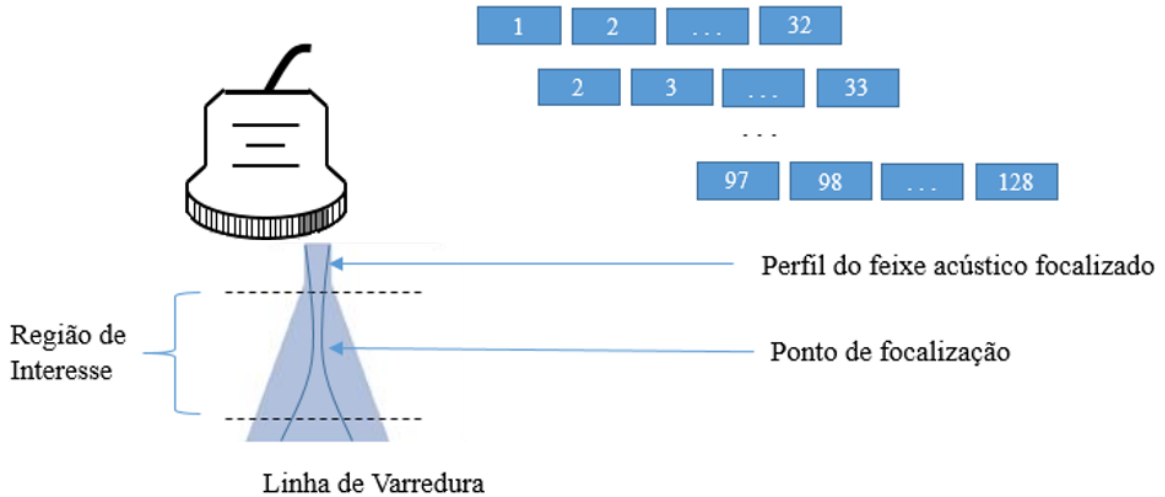
Figura 5 – Exemplo de excitação de um grupo de elementos piezoelétricos de um transdutor matricial convexo para geração de abertura focal sintética.



Fonte: Adaptado de Hedrick *et al.* (1995) e Jensen (1999).

Como exemplo de um dos projetos desenvolvidos neste trabalho, na Figura 6 é mostrada um perfil de excitação e campo acústico focalizado com uma abertura ativa de 32 elementos, para compor uma *scanline* do transdutor linear de 128 elementos. Completando a primeira transmissão e aquisição de dados com os elementos de 1 a 32 selecionados, a próxima *scanline* é gerada após o deslocamento da varredura eletrônica em um elemento, sendo composta pelos elementos 2 a 33, e assim sucessivamente. Esse processo é realizado até que todas as linhas de varredura sejam geradas para formar a imagem por US, isto é, até o final do último elemento ativo do transdutor.

Figura 6 – Exemplo de perfil de campo acústico focalizado gerado a partir da excitação de um grupo de 32 elementos piezoelétricos de um transdutor matricial linear. A primeira *scanline* é formada pelos elementos 1 a 32 e a última *scanline* é formada pelos elementos 97 a 128.



Fonte: Autoria própria.

O processo de geração de múltiplas ondas acústicas para orientar e focalizar o campo acústico é chamado de *beamforming*. De acordo com Hedrick e colaboradores (1995), a quantidade de *scanlines* N_{sc} do sistema é calculada pela Equação (2):

$$N_{sc} = ne - na + 1, \quad (2)$$

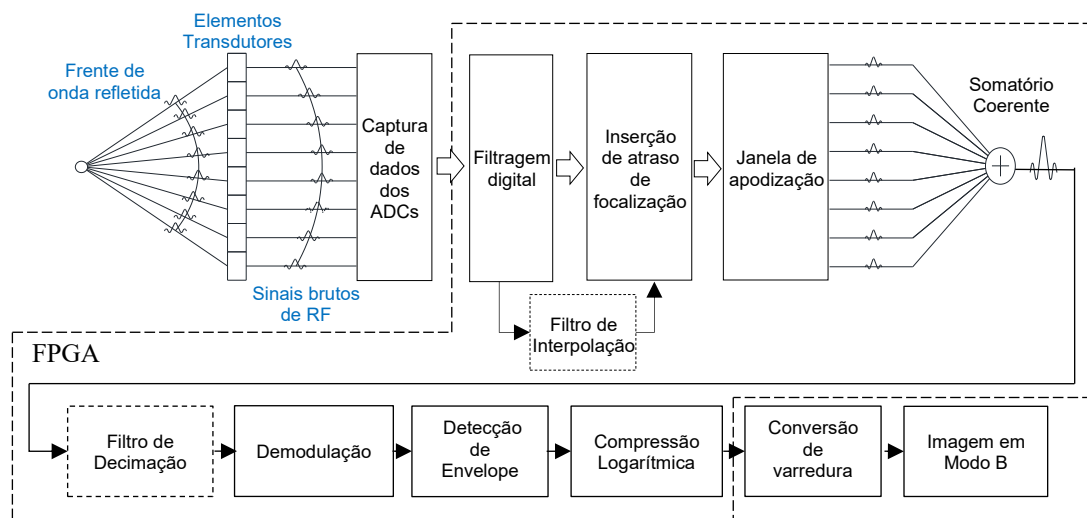
na qual ne é o número total de elementos do transdutor e na é o número de elementos ativos da abertura. Por exemplo, considerando um transdutor de 128 elementos, o número de *scanlines* geradas pelas aberturas ativas de 8 e 32 elementos é 121 e 97, respectivamente.

2.4 PROCESSAMENTO DE SINAL PELA TÉCNICA BEAMFORMING

O processamento dos sinais de US em Modo B utilizando a técnica DAS, foco da pesquisa deste trabalho, é realizado em várias etapas na unidade *back-end*, conforme a Figura 7. O método DAS é baseado no somatório coerente das linhas de RF após as etapas de filtragem digital, inserção de atrasos de focalização e apodização para redução dos lóbulos secundários. Em alguns sistemas é utilizada a técnica de interpolação para melhorar a resolução e desempenho de focalização. Após o somatório coerente, os sinais podem passar pela etapa de

decimação para otimizar a quantidade de informação, demodulação seguida pela detecção de envoltória, para extrair o envelope em baixa frequência do sinal ultrassônico, e compressão logarítmica para adequação das amplitudes dos sinais processados em uma escala de cinza, visando melhor visualização e análise dos resultados. Por fim, é necessária a conversão de varredura (*scan conversion*) para correção da geometria e ajuste de resolução do monitor de imagem.

Figura 7 – 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 Ferreira (2017).

Tradicionalmente, são usados ASICs e FPGAs nos segmentos de Tx e Rx para controle dos módulos e cálculos dos tempos de atraso de focalização em tempo real. Essas informações também podem estar armazenadas em blocos lógicos de memória *Look-up Table* (LUT). Vários trabalhos descrevem a utilização de DSPs *multicore* nas etapas de gerenciamento, transferência e processamento de dados, utilizando interfaces de alta velocidade como Ethernet, PCI Express, EMIF (*External Memory Interface*), USB 2.0 e, mais recentemente, USB 3.0 (LEWANDOWSKI *et al.*, 2013). Outros trabalhos abordam o uso de GPUs para reconstrução de imagens através da técnica de composição e Ultrafast (COUADE, 2016).

2.4.1 Filtragem Digital

A filtragem digital é uma técnica de processamento digitais de sinais amplamente utilizada para separar a informação de eco de RF de um sinal de ruído indesejado ou eliminar alguns componentes em frequência fora da faixa de interesse. Os filtros digitais também servem para cancelar a componente contínua indesejada das amostras de sinal bruto de RF (ASSEF, 2013). Essa etapa é realizada tipicamente por filtros passa-faixa com frequência central igual à frequência fundamental do transdutor e passa-baixa com frequência de corte relativa à faixa de operação a largura de banda do transdutor (ALI *et al.*, 2008).

Para realização da filtragem digital neste projeto, foi utilizado um filtro de resposta ao impulso finita FIR do tipo passa-baixa, uma vez que esses filtros apresentam características preferíveis em comparação aos filtros de resposta ao impulso infinita (*Infinite Impulse Response* - IIR), devido às suas propriedades de estabilidade inerente, linearidade de fase e simples implementação (DEBRUNNER; WANG, 2006).

Cada valor de saída da sequência de um filtro FIR em tempo discreto de ordem N pode ser expresso como uma soma ponderada dos valores de entrada de acordo com a soma discreta da convolução expressa pela Equação (3):

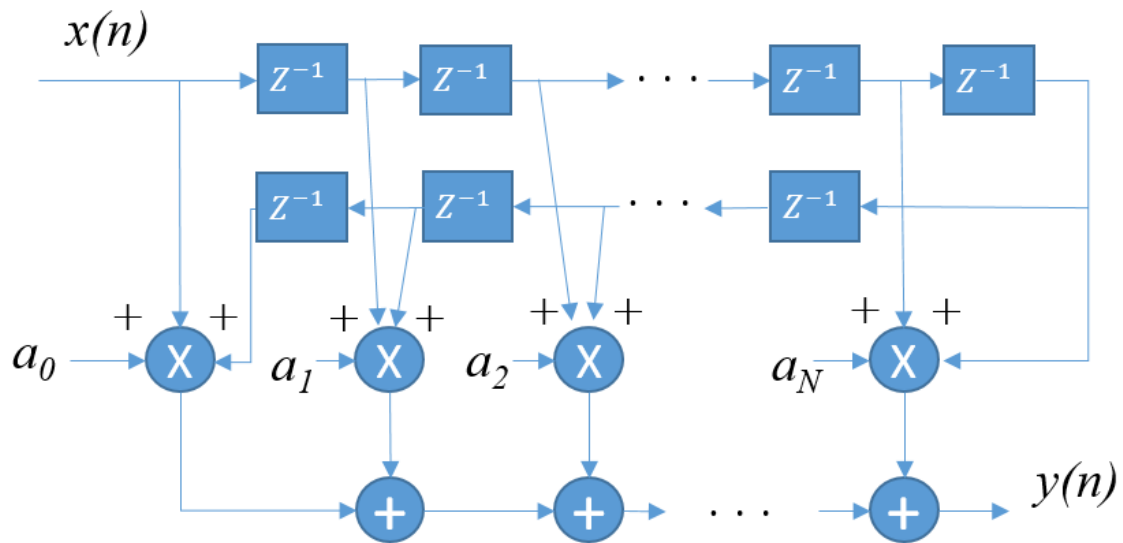
$$y(n) = a_0x(n) + a_1x(n-1) + \dots + a_Nx(n-N) = \sum_{i=0}^N a_i(n-i), \quad (3)$$

na qual n é o índice das amostras, $x(n)$ é o sinal de entrada, $y(n)$ é o sinal de saída e a_i é o coeficiente do filtro (INGLE; MANOLAKIS, 2011).

Assumindo uma ordem par e de acordo com a simetria da resposta ao impulso encontrada na Equação (3), tomando como exemplo, $a_0 = a_N$ e $a_1 = a_{N-1}$, o filtro pode ser descrito utilizando apenas metade dos coeficientes, de forma a economizar custo computacional e recursos de um dispositivo eletrônico. Essa técnica foi utilizada na arquitetura do projeto, ilustrada na Figura 8.

Para um projeto de filtros de fase linear FIR, recomenda-se utilizar o algoritmo de McClellan-Parks, denominado como Equiripple (EQ) (KUMAR *et al.*, 2019). A vantagem desses filtros é de se obter um projeto ótimo utilizando as menores especificações possíveis e, consequentemente, economizando custo computacional. No entanto, dadas circunstâncias específicas, a abordagem Least-Squares (LS) para o projeto de filtros FIR pode ser muito vantajosa (NGUYEN; VAIDYANATHAN, 1987). Neste trabalho foram avaliados os dois métodos de projeto de filtro: Equiripple (EQ) e Least-Squares (LS).

Figura 8 – Modelo de estrutura de filtro digital FIR.



Fonte: Adaptado de Oliveira *et al.* (2019).

2.4.2 Atraso Variável de Recepção

O atraso variável é uma técnica utilizada para focalizar as ondas ultrassônicas com tempo de atraso diferente para cada elemento da abertura de recepção da matriz piezoelétrica (FELDKAMPER *et al.*, 2000; AGARWAL *et al.*, 2007). A qualidade da imagem é seriamente influenciada pela precisão dos cálculos dos valores recebidos de uma *scanline*. Dessa maneira, o tempo de atraso aplicado aos sinais de um elemento transdutor varia durante a recepção de uma *scanline* (FELDKAMPER *et al.*, 2000). Basicamente, o tempo de atraso necessário para correção do atraso é convertido em ciclos de *clock* e aplicado para possibilitar o somatório coerente nas etapas seguintes.

2.4.3 Apodização

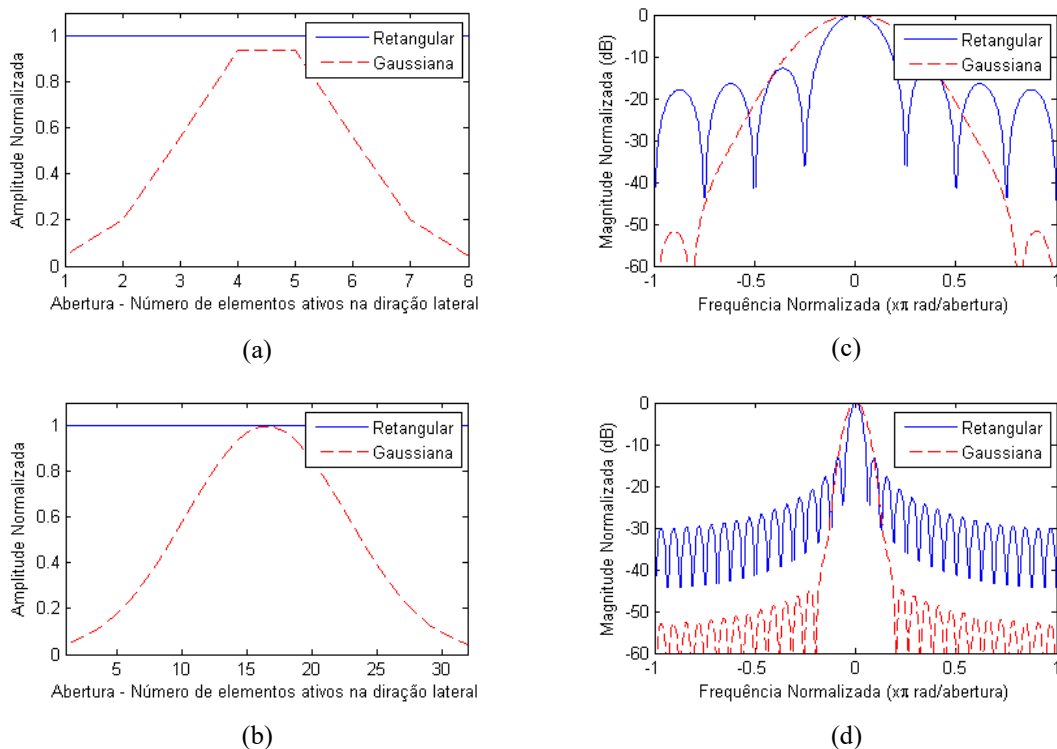
A apodização é uma técnica que pode ser aplicada tanto na transmissão como na recepção de US. Esta técnica utiliza janelas de ponderação para controle das propriedades do feixe acústico e melhoria na qualidade de imagem. Essas janelas têm como objetivo limitar os efeitos devidos aos elementos mais distantes do centro da matriz, resultando em um feixe acústico com lóbulos laterais reduzidos e uma imagem de maior qualidade (CINCOTTI *et al.*,

1999). Por consequência, essa janela reduz a intensidade do feixe irradiado ao redor do ponto de focalização, melhorando o contraste da imagem (ASSEF, 2013). Para calcular os coeficientes de apodização é essencial uma relação direta de características do transdutor e o padrão de campo (JENSEN; MUNK, 1998).

Na maioria dos sistemas comerciais, o processo de apodização é realizado apenas no modo de recepção, pois para ser implementado na transmissão são necessários circuitos eletrônicos complexos (CINCOTTI *et al.*, 1999).

Para ilustrar o efeito da aplicação da técnica de apodização, na Figura 9 são apresentados os resultados das simulações das técnicas de janela de ponderação Retangular e Gaussiana. As amplitudes normalizadas da Figura 9(a) e (b) e respectivas respostas normalizadas em frequência da Figura 9(c) e (d) são apresentadas para as aberturas de 8 e 32 elementos ativos. Em ambas as respostas simuladas, observa-se a redução dos lóbulos laterais em relação ao lóbulo principal em mais de 45 dB.

Figura 9 – Exemplo de comparação das técnicas de janela de ponderação Retangular (contínuo) e Gaussiana (tracejado). (a) Respostas normalizadas para abertura de 8 e (b) 32 elementos ativos e respectivas respostas em frequência normalizada (c-d).



Fonte: Adaptado de Assef (2013).

2.4.4 Somatório Coerente

Posteriormente, concluído o alinhamento dos sinais de abertura, um único sinal é obtido a partir do somatório coerente de todos os feixes de recepção, criando uma *scanline* (NIKOLOV *et al.*, 2008). O processo é conhecido como *Delay and Sum* e representado por:

$$r_f[n] = \sum_{m=1}^M w_m x_m(n - \Delta_m), \quad (4)$$

na qual 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 pertencentes a $[1, M(\text{canais})]$ com tempo n atrasado apropriadamente em $\Delta_m[n]$ e, por fim, resultando na saída do formador do feixe $r_f[n]$ (ASEN *et al.*, 2012).

2.4.5 Demodulação

Em conjunto com a detecção de envoltória, a demodulação é essencial para extrair a *scanline* final da faixa de frequência da portadora do eco. Para a formação da imagem em Modo B, é utilizada a detecção de envoltória do sinal resultante do somatório coerente dos multicanais de recepção (JINBO, 2007). A intensidade do brilho é resultado da modulação da amplitude do sinal extraída através da demodulação, seguida pela envoltória (JINBO, 2007; FERREIRA, 2017).

Existem alguns métodos convencionais para extrair a envoltória do sinal em baixa frequência. Um dos métodos é a Transformada de Hilbert, que apresenta respostas precisas para geração da envoltória, no entanto possui um custo computacional elevado, devido utilização da Transformada Rápida de Fourier (FFT). Outro método bastante utilizado é a quadratura e filtragem, que apresenta implementação simples e custo computacional inferior (ZHENG; ZHOU, 2015).

Neste projeto foi utilizado um método aproximado da Transformada de Hilbert baseada em filtro FIR (ASSEF *et al.*, 2019), ou seja, foi realizada a computação do envelope dos sinais a partir dos componentes de sinais em fase e quadratura.

2.4.5.1 Transformada de Hilbert

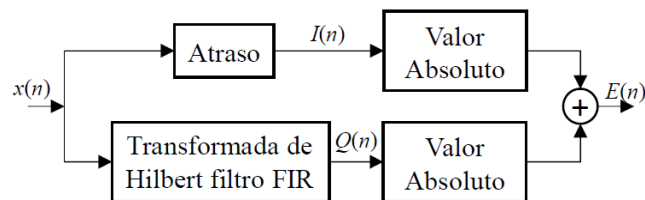
A Transformada de Hilbert (TH) envolve a extração de características do sinal de RF, separando em componente real, para o sinal em fase (I), e em componente imaginária (Q), para o sinal deslocado em 90° em relação à banda de operação. A magnitude $E(n)$ do sinal é regida pela Equação (5):

$$E(n) = \sqrt{I^2(n) + Q^2(n)}, \quad (5)$$

na qual $I(t)$ é o valor do sinal em fase e $Q(t)$ é o valor de quadratura característico da Transformada de Hilbert (OPPENHEIM, 1975).

O diagrama de blocos funcionais do detector de envelope avaliado neste projeto é mostrado na Figura 10. O sinal analítico de eco de US recebido é um sinal complexo, no qual a parte real (componente em fase) é o sinal original e a parte imaginária (componente em quadratura) é a Transformada de Hilbert do sinal original (CHANG *et al.*, 2007). O componente de quadratura $Q(n)$ é produzido por um filtro FIR com coeficientes da Transformada de Hilbert. O componente em fase $I(n)$ é o sinal de entrada atrasado por um montante para compensar o atraso de fase do processo FIR, empregado para gerar a saída $Q(n)$. Depois que os componentes complexos são processados, as informações do envelope $E(n)$ é obtido conforme a Equação (5). Os resultados dessa implementação foram publicados pelos autores Assef *et al.* (2019).

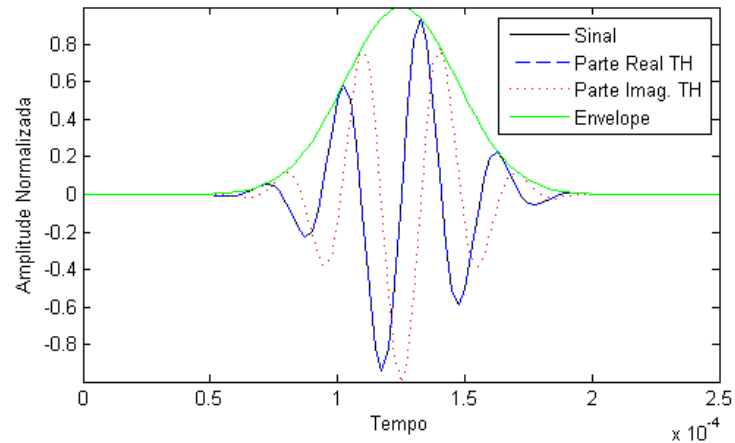
Figura 10 – Diagrama em blocos do detector de envelope baseado em uma aproximação da Transformada de Hilbert com filtro FIR.



Fonte: Adaptado de Assef *et al.* (2019).

Como exemplo para melhor ilustração, na Figura 11 são apresentados um sinal com perfil similar ao eco característico de US, a componente em fase (parte real da TH), a componente em quadratura (parte imaginária da TH) e o envelope do sinal.

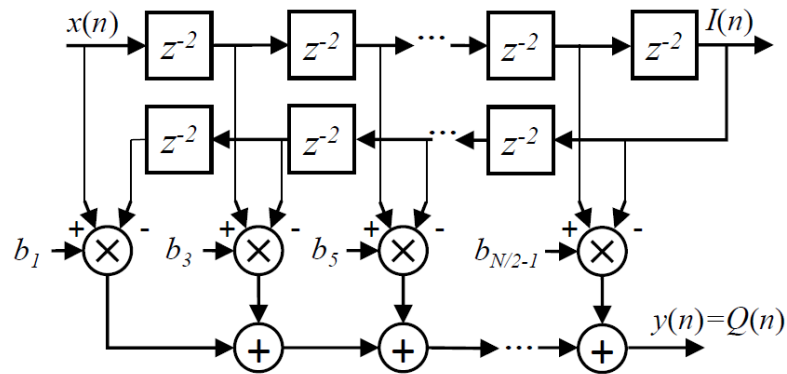
Figura 11 – Exemplo de detecção de envoltória com a aplicação da Transformada de Hilbert.



Fonte: Autoria própria.

Considerando ambas a simetria negativa e os coeficientes intercalados de valores nulo, característicos do projeto do filtro FIR com coeficientes gerados pela Transformada de Hilbert, a estrutura do detector de envoltória pode ser simplificada, conforme a Figura 12. Esta estrutura foi implementada no modelo do detector de envoltória deste trabalho.

Figura 12 – Arquitetura da Transformada Hilbert com filtro FIR explorando a simetria negativa e os coeficientes nulos intercalados.



Fonte: Adaptado de Assef *et al.* (2019).

2.4.6 Compressão Logarítmica

A compressão logarítmica é usada para adequar a faixa dinâmica das *scanlines* à visualização que, por consequência melhora o contraste da imagem, pois essas linhas muitas vezes excedem a faixa de escala de cinza (ASSEF, 2013). A variação na amplitude dos dados de RF é relativamente alta. Esse fator acaba escondendo alguns sinais menores. Em casos

práticos, algumas estruturas importantes do tecido poderiam apresentar valores de imagem muito baixos (WIKISONIX, 2013; FERREIRA 2017). Dessa forma, tipicamente as imagens de US são apresentadas com faixa dinâmica de -30 a -50 dB (JENSEN *et al.*, 2005; BONI *et al.*, 2012).

Os valores obtidos após a detecção de envoltória são comprimidos na forma logarítmica em decibéis (dB), utilizando a Equação (6):

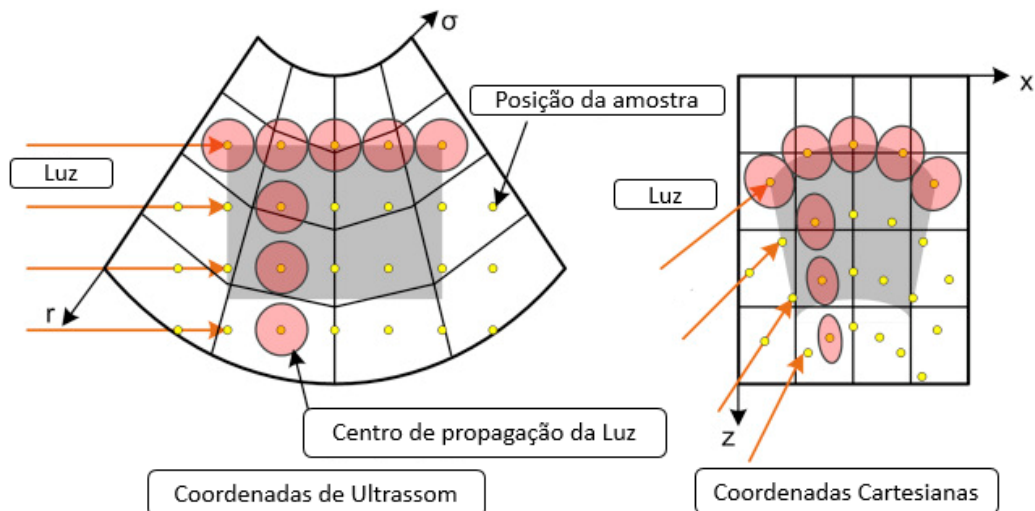
$$X_{dB}(n) = 20\log_{10}\left(\frac{x(n)}{x_{max}}\right), \quad (6)$$

na qual $x(n)$ é o sinal de entrada e x_{max} é a amplitude máximo do sinal de entrada. Dessa forma, o sinal de entrada com a máxima amplitude possui valor correspondente X_{dB} de 0 dB e os demais possuem valor negativo. Tipicamente, os equipamentos comerciais de US possuem um ajuste de limiar que pode ser ajustado pelo operador (HEDRICK *et al.*, 1995).

2.4.7 Conversão de Varredura

A Conversão de Varredura é a última etapa do processamento, implementada para apresentar, de maneira coerente, a imagem num *display* ou monitor. No entanto, como esses dispositivos possuem orientação cartesiana, é necessário ajustar as informações processadas (SZABO, 2004). Esse procedimento realiza a transformação dos dados em coordenadas polares para coordenadas cartesianas, conforme apresentado na Figura 13.

Figura 13 – Transformação de coordenada polar para coordenada cartesiana.



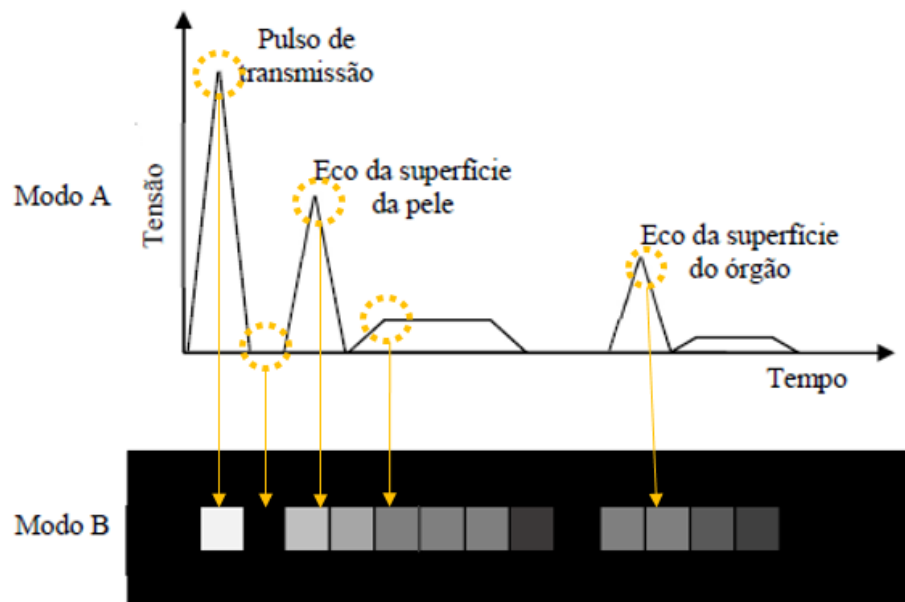
Fonte: Adaptado de Kwon *et al.* (2020).

A partir dessa etapa de processamento, pode-se gerar a imagem em Modo B como resultado final.

2.4.7.1 Imagem em Modo B

Cada ponto da imagem (*pixel*) determina basicamente o brilho e a cor atribuída a cada elemento da imagem, sendo relacionado com a amplitude do sinal recebido pelo transdutor, que por convenção, é usado em 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 de US percorrer 2 vezes a distância entre o transdutor e a ROI (SHATTUCK *et al.*, 1984; HEDRICK *et al.*, 1995). Na Figura 4 é ilustrada a envoltória dos ecos recebidos das interfaces entre ambos Modos A e B, a partir da excitação de um transdutor monoelemento posicionado a uma distância da pele de um corpo biológico. Pode ser observado que o sinal com maior amplitude é convencionalmente com a cor branca e o de menor intensidade com a cor preta na escala de cinza.

Figura 14 – Exemplo de formação de imagem em Modo A e Modo B.



Fonte: Adaptado de Shung (2006) e Ferreira (2017).

A diferença de amplitude entre os sinais se deve a fatores geométricos, mecanismos de absorção e a profundidade ROI. Fatores geométricos são determinados por dimensões da fonte sonora, comprimento da onda, presença de superfícies refletoras, etc. Em mecanismos de

absorção, parte da energia é convertida em calor. Já em relação à ROI, pode ser compensada por uma curva de amplificação característica de atenuação do meio (HEDRICK *et al.*, 1995; WELLS, 1999; OTAKE *et al.*, 2003; SHUNG, 2006).

2.4.8 Imagem em Tempo Real

As imagens são geradas rápida e repetitivamente para apresentar uma quantidade adequada de quadros por segundos de seções através do tecido, de modo que o movimento das interfaces possa ser observado. Vetores simples ou varredura linear são normalmente feitos por feixes ultrassônicos com instrumento em tempo real. A produção de imagens é possível devido à alta velocidade das ondas ultrassônicas no tecido. As informações referentes aos ecos por cada direção de feixe podem ser coletadas em uma fração de milissegundos, resultando em uma grande quantidade de dados por segundo. Mesmo que uma imagem de 150 linhas de dados de eco possa estar comprometida, 25 a 50 imagens são geradas por segundo, implicando que as imagens em tempo real são livres de cintilações e permitem um estudo mais detalhado do movimento do tecido (MCDICKEN, 1976; CHRISTENSEN, 1988; HEDRICK *et al.*, 1995).

2.5 FPGA

Atualmente inúmeros equipamentos comerciais são compostos por circuitos integrados (CI) dedicados para tarefas específicas no processamento, como microprocessadores, dispositivos de programação lógica (*Programmable Logic Device* - PLD), FPGA ou ASIC. Os PLDs possuem uma arquitetura definida, com número reduzido de elementos lógicos em comparação com um FPGA. Sendo assim, os circuitos sintetizados em seu interior tendem a ser menos robustos. Os ASICs são utilizados para aplicações específicas, diferentemente dos PLDs e FPGA que possuem ampla aplicação. Eles são extremamente rápidos e necessitam de menor espaço para síntese do circuito equivalente. Pelo fato deste dispositivo ser intransigente quanto a modificações internas e possuir um custo elevado no mercado, apenas produções específicas em larga escalas utilizam do mesmo (CÂMARA, 2015).

O dispositivo FPGA foi desenvolvido pela Xilinx em 1985 e utilizado pela Marinha Americana (MAXFIELD, 2004). Trata-se de um semicondutor composto por milhares de lógicas digitais, encapsuladas em um CI, no qual cada bloco de *hardware* é designado para uma função específica, podendo ser programado ou reprogramado pelo usuário de forma modular (RODRIGUEZ-ANDINA *et al.*, 2007; MONMASSON *et al.*, 2011; FERREIRA, 2017). A

partir do roteamento, processo capaz de reproduzir lógicas simples até sistemas combinacionais mais complexos com registradores e multiplicadores para função DSP, acrescenta-se flexibilidade ao projeto. Por fim, devido a essas vantagens o dispositivo possui um baixo custo em relação aos concorrentes e alta confiabilidade, sendo possível operar com velocidade temporizada por *hardware* e com número elevado de circuitos através da configuração do próprio dispositivo (SANTOS, 2008).

As tarefas de processamento do FPGA são enviadas para uma seção dedicada do *chip* que funciona de modo autônomo sem interferência de outros blocos lógicos. Desse modo, o desempenho da aplicação não é influenciado com a adição de mais tarefas (NATIONAL INSTRUMENTS, 2013). O funcionamento da FPGA depende exclusivamente da sua arquitetura. Ela pode operar em ciclos de *clock*, onde cada operação matemática é executada a cada ciclo (HUDA *et al.*, 2009). O processamento é realizado em paralelo. Há disponibilidade de diversos modos para implementações de funções lógicas e matemáticas em cada bloco lógico da FPGA (KANG *et al.*, 2016). Em geral, o mais utilizado pelos fabricantes é o LUT, sendo responsável pelas células de armazenamento de memória volátil RAM (*Random Access Memory*), na qual são implementadas de forma otimizada as funções lógicas (DA COSTA, 2014; FERREIRA, 2017).

A linguagem de descrição do *hardware* (HDL) é popularmente a mais utilizada para programação de FPGA. Essa linguagem é responsável pela configuração e programação dos circuitos digitais. Entre as principais linguagens HDL, pode-se citar: VHDL (*Very-High Integrated Circuit HDL*) e Verilog (BERGÉ *et al.*, 1992).

2.6 MÉTODOS DE TRANSFERÊNCIA DE DADOS

Uma transferência de dados é caracterizada pela troca de dados entre meios digitais e/ou físicos. Esse procedimento depende do equipamento e dos métodos utilizados para encapsular os dados. Existem dois tipos de transferência de dados: a transferência com interface serial e a transferência com interface paralela. A transferência serial é realizada *bit a bit*, possuindo uma limitação na taxa de transmissão. Já na transferência paralela, vários *bits* são transmitidos simultaneamente (CLARK, 1983; MULLER, 2003).

Existem diversos métodos de transferência de dados de US explorados pela comunidade científica, assim como os métodos via Bluetooth, USB, Ethernet, etc. O método escolhido para este projeto foi o de Ethernet, devido à sua alta taxa de transferência e implementação simples.

2.6.1 Ethernet

Até a seção anterior foram apresentados os conceitos para fundamentar as características necessárias para o entendimento da geração de imagem por US em Modo B. A partir desta seção serão descritos os conceitos referentes à transferência de dados via Ethernet.

O modelo OSI (*Open System Interconnection*), definido pela ISO (*International Organization Standardization*), detalha a organização ótima de *hardware* e *software* de forma a operar na formação de redes de dados (BERTSEKAS; GALLAGER, 1992; NAUGLE, 1994; MENOTTI, 2005). Assim, o sistema Ethernet é formado por 4 componentes:

- *Frame*: é o quadro de dados que consiste em um conjunto de *bytes* organizados por um padrão usado para o tráfego de informações na rede;
- Protocolo de controle de acesso à mídia (*Media Access Control Protocol - MAC*): conjunto de regras estabelecidas em cada interface Ethernet, tornando possível o acesso a um meio compartilhado adequadamente;
- Componentes de sinalização (*Signalling Components*): dispositivos eletrônicos que enviam e recebem sinais sobre cada interface Ethernet;
- Meio físico (*Physical Medium*): formado por cabos e outros circuitos de *hardware* usados para transmitir o sinal digital entre os computadores ligados à rede.

O elemento principal do sistema Ethernet é o *frame*, sendo que os demais garantem apenas o carregamento e o transporte dos *frames* de um dispositivo para outro (SPURGEON, 2000; MENOTTI, 2005). O protocolo MAC arbitra o acesso à mídia compartilhada a partir de um conjunto de regras, sendo utilizado pelo modo de operação *half-duplex*. O *frame* é enviado por um canal de sinal compartilhado e todas as interfaces ligadas ao canal obtêm os *bits* do sinal e entendem o endereço de destino. A interface compara o seu endereço *unicast* de 6 *bytes*, ou mesmo de algum endereço *multicast* que esteja habilitado a ser reconhecido. Conforme o reconhecimento do endereço do *frame*, as interfaces de rede continuam a leitura ou descartam o *frame*.

Dada a fabricação, cada interface possui endereço *unicast* único. Os endereços *multicast* são responsáveis pelo recebimento de um *frame* Ethernet por um grupo de estações. Uma exceção ao endereço *multicast* é dado como endereço *broadcast*, que possui 6 *bits* em nível lógico 1. Todas interfaces de rede obtêm os pacotes enviados com esse destino e transmite ao *software* de rede do computador ou outro equipamento de comunicação.

2.6.1.1 *Frame* e Transmissão

A composição, organização e a transmissão dos *frames* são regidas pelas especificações do Ethernet. O *frame* é definido pelo padrão IEEE (*Institute of Electrical and Electronics Engineers*) 802.3. A Figura 15 apresenta a composição de cada *frame* Ethernet.

Figura 15 – Composição do *frame* Ethernet.

Tamanho	7 bytes	1 byte	6 bytes	6 bytes	2 bytes	46 a 1500 bytes	4 bytes
Descrição	Preâmbulo	SFD	Endereço de Destino	Endereço de Origem	Tipo/Tamanho	Dados	Sequência de Checagem

Fonte: Adaptado de Menotti (2005).

2.6.1.2 Preâmbulo

Inicialmente, tem-se o preâmbulo composto por 64 *bits*, de forma a permitir que as interfaces que operam a 10 Mbps sincronizem o recebimento antes que os campos de dados importantes cheguem até as mesmas. A partir disso, é possível descartar alguns *bits* mesmo sem o sincronismo ter sido estabelecido. Sistemas como Fast Ethernet (100 Mbps) e Gigabit (1000 Mbps) têm uma codificação complexa, de forma a não necessitar de um preâmbulo, já que os dados não são perdidos no início do recebimento. Contudo, esse campo ainda é utilizado para que estes sistemas mais robustos não percam a compatibilidade com sistemas Ethernet mais antigos.

2.6.1.3 Delimitador de *Frame* Inicial

O campo delimitador de *frame* inicial (SFD) é uma sequência binária *10101011*. Logo após esse limitador inicia-se o *frame* MAC (IEEE, 2012).

2.6.1.4 Endereço de Destino

Cada *frame* Ethernet tem um único endereço de 6 *bytes*. Esse campo contém o endereço de uma interface, um endereço *multicast*, responsável por identificar uma ou mais interfaces que estejam habilitadas a receber o endereço de *broadcast* padrão. Caso o endereço não seja um *frame* apto a ser recebido, então o restante do *frame* é ignorado. O padrão IEEE 802.3 adiciona uma importância especial ao segundo *bit* desse campo, sendo utilizado para distinção das interfaces administradas globalmente e localmente. Se o *bit* for 0, a interface é

global e o endereço foi atribuído pelo seu fabricante. Para sistemas locais, o segundo *bit* é 1 (3COM, 1998; IBM, 2001; SPURGEON, 2000).

Os endereços físicos são escritos em 12 dígitos hexadecimais agrupados em pares e transmitidos da esquerda para a direita. Contudo, cada octeto de *bits* é transmitido do sentido do *bit* menos significado ao mais significativo.

2.6.1.5 Endereço de Origem

Representa o endereço físico da interface de origem do *frame*. Esse campo é enviado para os protocolos de mais alto nível, no entanto não é interpretado pelo protocolo MAC. É passível de mudança tanto pelo administrador de rede ou pelo *software* de rede, uma vez que não existe restrição pelo padrão IEEE.

De forma a obter um endereço físico, é preciso que o fabricante solicite um identificador único denominado OUI (*Organizationally Unique Identifier*) junto ao IEEE. Esse identificador possui 4 *bytes* e forma a primeira parte do endereço físico da interface fabricada sendo útil para identificação do fabricante ou um possível problema na rede.

2.6.1.6 Tipo ou Tamanho

De acordo com o padrão IEEE esse campo é dado como tipo/tamanho e depende apenas de seu valor. Se o mesmo for numericamente menor ou igual a 1500, este indica o tamanho do campo de dados automaticamente preenchido, caso não alcance o mínimo de 2 *bytes* (MENOTTI, 2005).

2.6.1.7 Dados

Deve conter dados de 46 *bytes* a 1500 *bytes*. Por se tratar de um campo com total de transparência de dados, qualquer sequência arbitrária de *bytes* irá aparecer nesse campo até o máximo determinado (IEEE, 2012).

2.6.1.8 Sequência de Checagem

Este é o último campo do *frame* Ethernet de ambos padrões, conhecido como *Frame Check Sequence* (FCS) ou *Cyclic Redundancy Check* (CRC) e tem a função de checar a integridade dos demais campos, exceto o preâmbulo, e possui um tamanho de 4 *bytes*. Esse campo é utilizado por ser capaz de identificar facilmente erros, gerar um pequeno *overhead* e ser de fácil instalação (COMER, 2001).

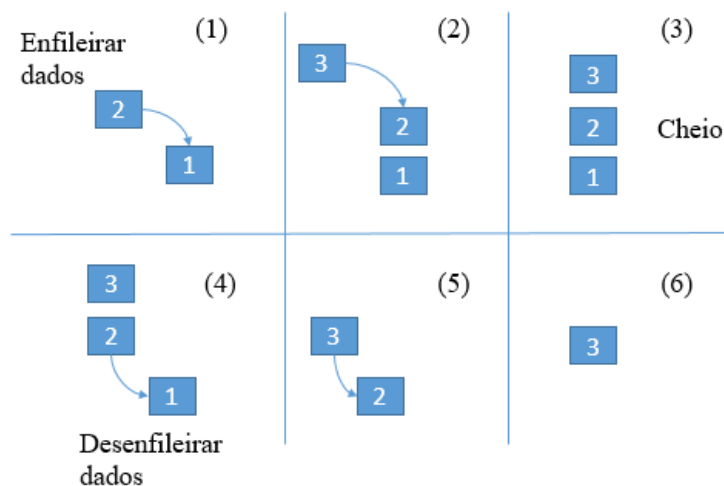
2.7 ESTRUTURA DE DADOS

Quando se trata de memória, custo computacional e gerenciamento de espaço, a organização de informação tende a melhorar a eficiência do algoritmo. Na Ciência da Computação são utilizadas algumas Estruturas de Dados para esse fim. Essas estruturas podem ser vetores, listas, filas, pilhas, árvores, etc. (BLACK, 2004).

Neste projeto, foi implementado em arquitetura digital e em linhas de programação, um dos conceitos de fila, conhecido como *First In First Out* (FIFO). Esse conceito baseia-se na ideia de “primeiro a entrar, primeiro a sair”, conforme ilustrado no exemplo da Figura 16.

Inicialmente, os dados de entrada são enfileirados de acordo com a ordem de chegada, mostrado na primeira etapa do diagrama. Em seguida, a FIFO recebe mais dados até estar completamente cheia. Por fim, os dados são desenfileirados, de acordo com a ordem de chegada, até não sobrar mais nenhum elemento na fila.

Figura 16 – Exemplo de diagrama da execução de uma estrutura do tipo FIFO.



Fonte: Autoria própria.

3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais e métodos utilizados na implementação do sistema de transferência de dados de US via Ethernet para processamento embarcado em *hardware* reconfigurável. A partir da especificação de requisitos, são mostradas as ferramentas de desenvolvimento, as placas de FPGA utilizadas, as etapas de processamento digital de sinais e as métricas adotadas para avaliação dos sinais e imagens geradas.

3.1 ESPECIFICAÇÃO DE REQUISITOS DO SISTEMA PROPOSTO

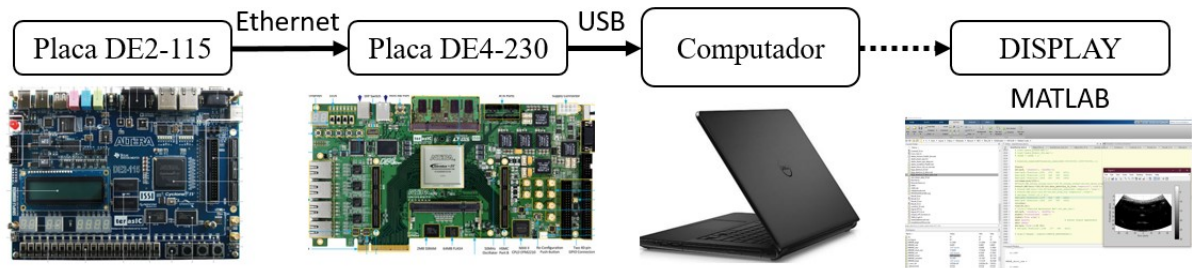
Para a implementação do sistema de transferência de dados via Ethernet e processamento embarcado em FPGA foi inicialmente definida a especificação de requisitos. Essa especificação é mostrada na Tabela 2 com as principais características do sistema proposto. De forma complementar, para ilustrar o fluxo de dados entre os dispositivos deste projeto de pesquisa, na Figura 17 é apresentado o diagrama em blocos completo do sistema proposto, mostrando também as interconexões entre as etapas.

Tabela 2 – Especificação de requisitos do sistema proposto.

Nº	Especificação
1	Utilização dos sinais de RF adquiridos pelo sistema ULTRA-ORS
2	Emprego da placa de FPGA DE2-115 como sistema de transferência de dados via Ethernet
3	Emprego da placa de FPGA DE4-230 como sistema de recepção e processamento de dados de US
4	Transmissão de dados via rede Ethernet no modo Gigabit (1000 Mbps)
5	Reconstrução de imagem baseada no transdutor matricial convexo AT3C52B de 128 elementos
6	Implementação do sistema de processamento para as aberturas de 8 e 32 elementos ativos
7	Processamento de 121 e 97 <i>scanlines</i> com comprimento de 2046 <i>words</i> (duas <i>words</i> de cabeçalho)
8	Frequência de amostragem de 40 MHz e resolução de 12 <i>bits</i>
9	Modelagem das etapas de processamento através da biblioteca Intel DSP Builder no Simulink
10	Implementação das seguintes etapas de processamento em <i>hardware</i> para o <i>beamforming</i> DAS: filtragem digital, atraso variável, apodização, somatório coerente, demodulação com detecção de envoltória e compressão logarítmica
11	Filtro digital FIR com frequência de corte de 8 MHz
12	Projeto do filtro digital FIR pelos métodos de projeto de filtro Equiripple e Least-Squares
13	Avaliação da ordem do filtro FIR de 8, 16, 24, 32 e 64 coeficientes
14	Avaliação do número de <i>bits</i> da parte fracionária dos coeficientes do filtro FIR de 4 a 17 <i>bits</i> (limite dos blocos DSP da FPGA)
15	Projeto da etapa de demodulação com detecção de envoltória baseada em uma aproximação da Transformada de Hilbert com filtro digital FIR
16	Avaliação da ordem do detector de envelope com filtro FIR de 8 a 64 coeficientes (passo de 8)
17	Avaliação do número de <i>bits</i> da parte fracionária dos coeficientes do detector de envelope com filtro FIR de 4 a 17 <i>bits</i>
18	Implementação do modelo de compressão logarítmica baseado em LUT
19	Etapas de conversão de varredura realizada em <i>software</i> através do Matlab
20	Avaliação da acurácia entre os resultados simulados e experimentais através da função NRMSE
21	Aplicação das métricas CR e CNR para validação os resultados experimentais do processamento

Fonte: Autoria própria.

Figura 17 – Diagrama em blocos do sistema de transferência e processamento de sinais US proposto.

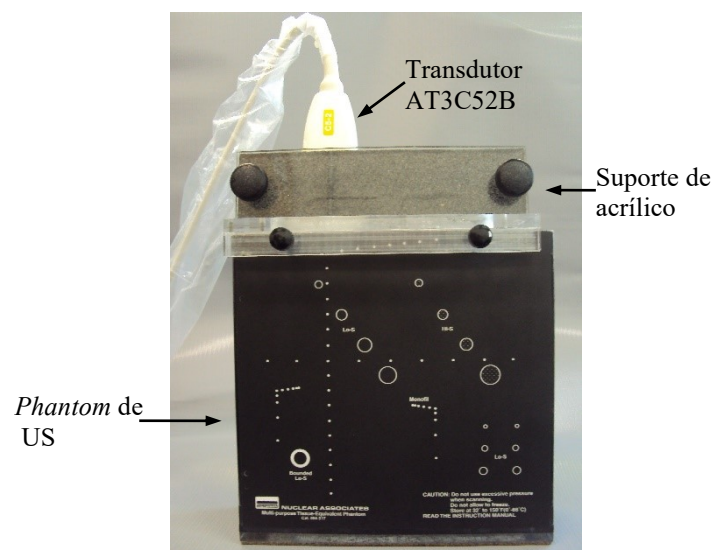


Fonte: Autoria própria.

O sistema proposto é formado por duas placas de desenvolvimento de FPGA: Terasic DE2-115 e DE4-230. Ambas as placas possuem interface de rede Ethernet e foram disponibilizadas pelo Laboratório de US (LUS) da UTFPR. A primeira placa tem a função de montar e transmitir os pacotes de dados via rede Ethernet usando o protocolo de comunicação TCP/IP. A segunda placa recebe os pacotes, extrai os dados de RF e realiza o processamento para reconstrução da imagem usando a técnica *beamforming* DAS.

Os projetos das duas placas de FPGA foram realizados e avaliados com dados brutos de US, provenientes do sistema ULTRA-ORS, usando o transdutor convexo AT3C52B (Broadsound Corp., Taiwan) de 128 elementos em um *phantom* mimetizador de tecidos biológicos de ultrassom, modelo 84-317 (FLUKE BIOMEDICAL, 2020), conforme a Figura 18.

Figura 18 – Conjunto formado pelo transdutor multielemento convexo AT3C52B, suporte de acrílico de fixação das partes e *phantom* de US para captura dos sinais de eco.



Fonte: Autoria própria.

Os dados com frequência de amostragem de 40 MHz e resolução de 12 *bits* foram adquiridos pelo sistema ULTRA-ORS e pré-gravados na placa de transmissão DE2-115, considerando o método DAS para a avaliação das aberturas de 8 e 32 elementos. Dessa forma, foi possível desenvolver o sistema de transferência e processamento sem a necessidade de acrescentar uma interface adicional para o ULTRA-ORS ou fazer novas aquisições a cada novo projeto.

O *software* Quartus Prime e a ferramenta Qsys foram empregados para a criação dos respectivos processadores Nios II de 32 *bits* para controle das aplicações. Para a programação da placa de processamento DE4-230 com a FPGA de alto desempenho Stratix IV, foi empregada a ferramenta DSP Builder em conjunto com o *software* Matlab/Simulink, possibilitando a modelagem, simulação e transformação dos modelos proposto em linguagem HDL.

3.2 SOFTWARE DE MODELAGEM – MATLAB/SIMULINK

O Simulink é a ferramenta para modelagem, simulação e análise de sistemas dinâmicos integrada ao *software* de cálculo Matlab. O *software* possui um ambiente de diagrama de bloco para simulação em múltiplos domínios. Também apresenta um editor gráfico com uma biblioteca de blocos customizáveis para modelagem e simulação de sistemas dinâmicos. Como uma das vantagens de abstração de projeto, o Simulink é capaz de gerar códigos em linguagem Verilog e VHDL para programação de dispositivos FPGA, através da biblioteca HDL Coder (MATHWORKS, 2018).

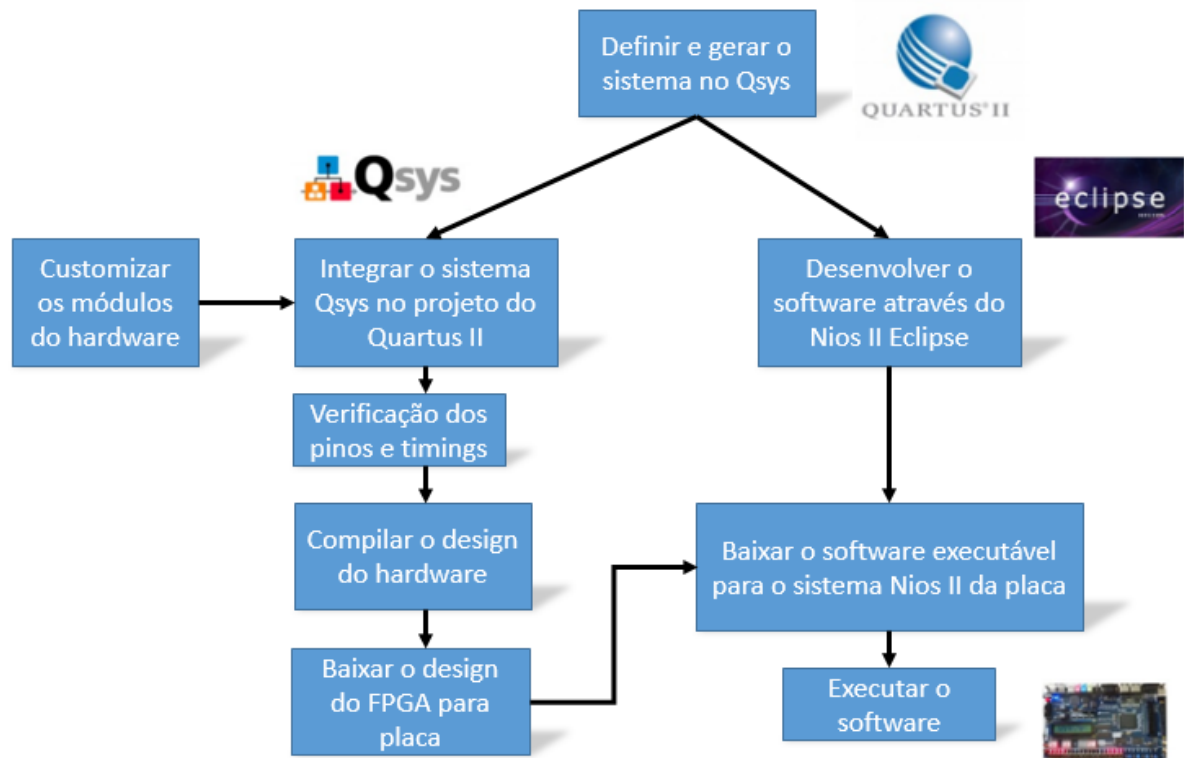
Outro suporte para esta ferramenta com capacidade para desenvolvimento de sistemas de processamento digital de sinais, é a biblioteca DSP Builder, responsável pela integração do Simulink com o *software* Quartus Prime para configuração de FPGAs da empresa Intel.

3.3 SOFTWARE QUARTUS PRIME

O *software* Quartus Prime é a ferramenta para desenvolvimento, programação e configuração de FPGAs da empresa Intel. O *software* agrega uma ferramenta de projetos de microprocessadores Nios II de até 32 *bits* para FPGA, denominado Qsys. Essa ferramenta é empregada para tornar mais simples o desenvolvimento e reutilização referente ao *hardware* de processadores em FPGAs Intel. Através do Qsys é possível gerar automaticamente interconexões lógicas para conectar blocos de componentes ao subsistemas, além de ser

baseado em um *framework* hierárquico para interconectar sistemas mais complexos, conforme é demonstrado no fluxograma da Figura 19. Para programação dos processadores Nios II implementados nas FPGAs para controle e comunicação via rede Ethernet foi utilizado o *software* Nios II Eclipse, outra ferramenta incorporada ao Quartus Prime.

Figura 19 – Fluxograma de desenvolvimento e integração do projeto no Quartus.



Fonte: Autoria própria.

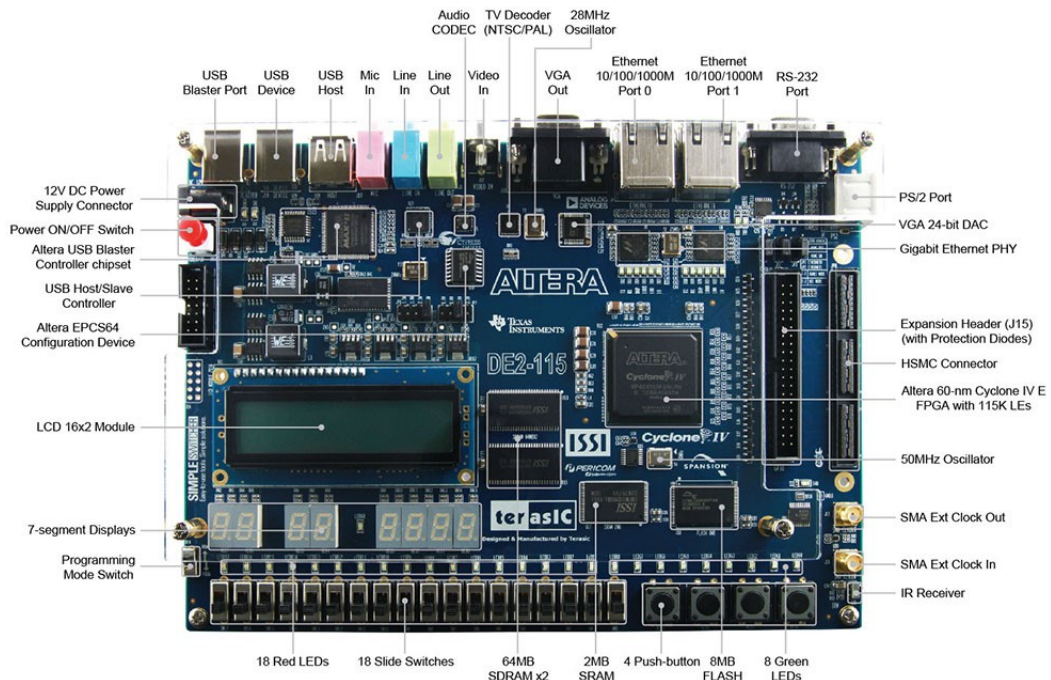
3.4 PLACA DE2-115 – FPGA TX

O primeiro *kit* de desenvolvimento que foi utilizado para este projeto é o DE2-115 (Terasic Inc., Taiwan), apresentado na Figura 20. O componente central da placa é a FPGA Cyclone IV EP4CE115F29C7N que possui 114.480 elementos lógicos, 3,8 MB de memória interna, 4 blocos *Phase-Locked Loop* (PLL) e 266 multiplicadores 18x18 (INTEL, 2020a). Entre os diversos periféricos e funcionalidades da placa destacam-se as duas portas Ethernet e duas memórias *Synchronous Dynamic Random Access Memory* (SDRAM) IS42S16320B que somadas possuem capacidade de 128 MB para armazenamento de códigos do processador Nios II. Cada porta Ethernet é controlada por um *chip* Marvel 88E1111 Gigabit Ethernet PHY com

conector RJ45, possibilitando configuração da velocidade de operação de 10/100/1000 Mbps e suporte para núcleos IP de Ethernet industrial.

Neste trabalho de pesquisa, a placa DE2-115 tem a função de simular o sistema ULTRA-ORS e transmitir os dados de aquisição para a placa de processamento via Ethernet. Dessa forma, foram gravados os sinais digitalizados das instigações do ULTRA-ORS em um *phantom* de US usando as aberturas de 8 e 32 elementos ativos, gerando 121 e 97 *scanlines*, respectivamente. Na Tabela 3 é apresentado o resumo comparativo do tamanho da abertura ativa *versus* a quantidade de memória para o armazenamento dos dados.

Figura 20 – Placa de desenvolvimento DE2-115.



Fonte: Terasic (2018).

Tabela 3 – Abertura ativa *versus* quantidade de memória para um transdutor de 128 elementos.

Abertura ativa	Número de <i>scanlines</i>	Número de amostras por canal (<i>words</i> de 12 <i>bits</i>)	Quantidade de memória (<i>words</i> de 12 <i>bits</i>)
8	121	2046	1.980.528
32	97	2046	6.350.785

Fonte: Autoria própria.

3.4.1 Protocolo de Comunicação via Rede Ethernet

Para a implementação da etapa transferência (Tx), os dados brutos de US gerados nas investigações do ULTRA-ORS foram exportados para um PC e gravados em dois blocos de notas do tipo *txt*, um para o projeto com 8 elementos de abertura e outro para o projeto com 32 elementos de abertura. Esses dados foram arranjados como um vetor em linguagem C para serem descarregados na FPGA DE2-115, placa Tx, durante a gravação dos códigos fonte, de forma a simular a transferência dos dados adquiridos através da plataforma ULTRA-ORS.

O código implementado para a placa Tx tem a função de organizar os dados em 749 *words* para transferência, conforme mostrado no fluxograma da Figura 21. Levando em conta que cada canal tem o tamanho de 2046 *words*, para completar um bloco é realizada uma terceira seção de 548 *words*. Isto é, para cada canal são enviados 3 pacotes de dados.

Após uma transferência ser concluída, a placa de Tx aguarda o *flag* de permissão da FPGA DE4-230, placa de Rx, para efetuar a próxima transmissão. Assim que os 8 ou 32 canais são preenchidos, dependendo da abertura avaliada, as memórias FIFOs são descarregadas em paralelo e é iniciado o processamento dos dados referente àquela *scanline*. Dessa forma, é garantido o sincronismo das etapas de processamento. O processo se sucede até o preenchimento das 121 ou 97 *scanlines*.

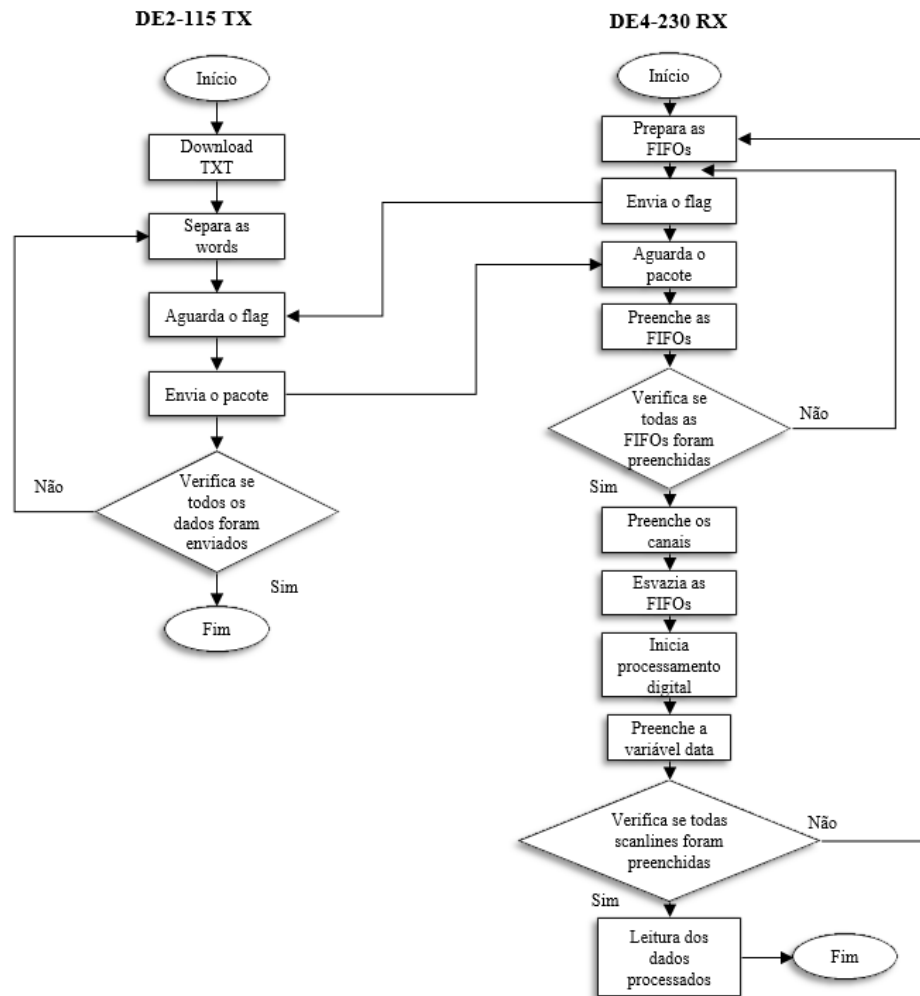
Quando finalizado todo procedimento em *hardware*, os dados processados e armazenados na variável “*data*” são enviados, através de uma porta serial emulada via USB, para o PC com taxa de transmissão de 115200 bps. A análise do tamanho da variável “*data*” é apresentada na Tabela 4. Finalmente, esses dados são exportados para o Matlab para finalizar a conversão de varredura e geração da imagem em Modo B.

Tabela 4 – Tamanho da variável “*data*” para armazenamento dos dados processados.

Abertura ativa	Número de <i>scanlines</i>	Número de amostras por canal (<i>words</i> de 12 <i>bits</i>)	Tamanho da variável <i>data</i> (<i>words</i> de 16 <i>bits</i>)
8	121	2046	247.566
32	97	2046	198.462

Fonte: Autoria própria.

Figura 21 – Fluxograma do fluxo de dados entre as FPGAs Tx e Rx.

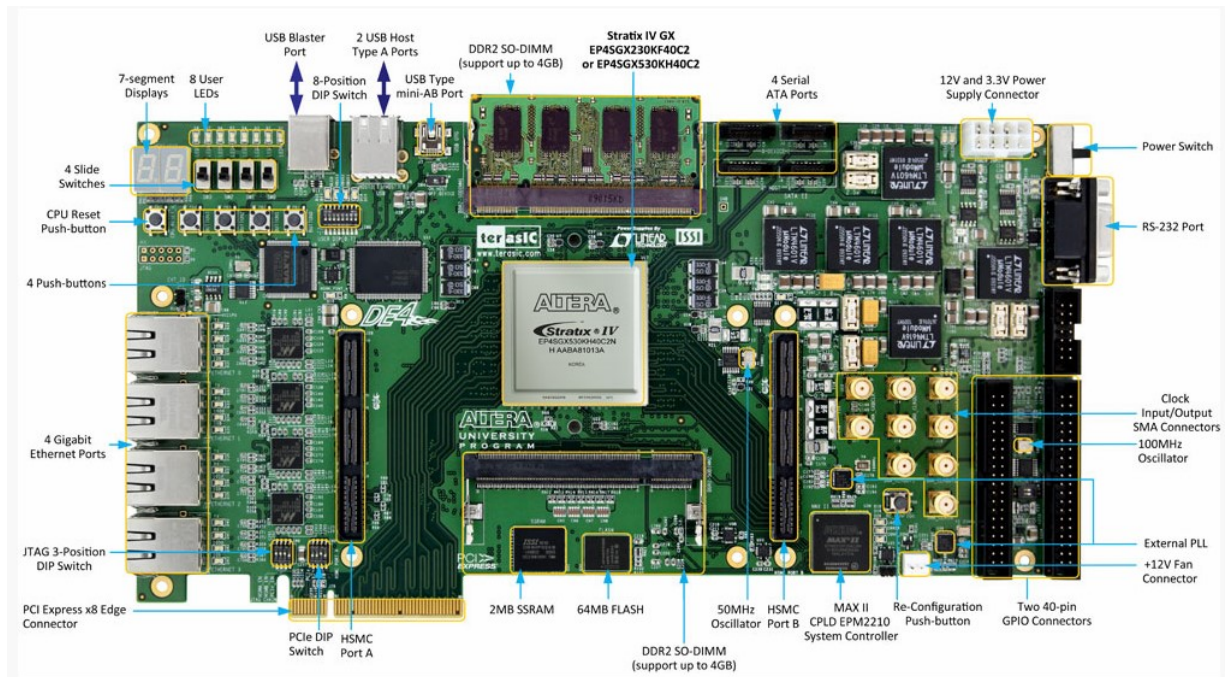


Fonte: Autoria própria.

3.5 PLACA DE4-230 – FPGA RX E PROCESSAMENTO

A segunda placa de desenvolvimento é a DE4-230 (Terasic Inc., Taiwan). Na Figura 22 é apresentado o *kit* utilizado no projeto para recepção e processamento dos sinais de US recebidos através da interface Ethernet. A placa foi escolhida para este trabalho pois possui uma FPGA Stratix IV EP4SGX230KF40C2 de alto desempenho, capaz de realizar todo processamento embarcado. Essa FPGA possui 228.000 elementos lógicos, 17,13 MB de memória interna, 8 blocos PLL, 1.288 multiplicadores 18x18 e dois blocos IP de barramento PCI Express (INTEL, 2020b). No projeto, as duas funcionalidades principais exploradas foram as 4 portas para Gigabit Ethernet integradas e os 2 soquetes SO-DIMM (*Dual In-line Memory Module*) para memórias DDR2 (*Double Data Type 2*) com taxa de *clock* de 400 MHz.

Figura 22 – Placa de desenvolvimento DE4-230.



Fonte: Terasic (2018).

3.6 ETAPAS DE PROCESSAMENTO IMPLEMENTADAS

Nesta seção são apresentadas as etapas de processamento digital de sinais implementadas em *hardware* descritas na seção 2.4. Cada etapa de processamento foi avaliada em simulações no Simulink e de forma experimental na placa Rx.

Para efeitos de visualização e praticidade, todas as figuras mostradas nesta seção consideram apenas o projeto com abertura de 8 elementos ativos, pois o modelo com abertura de 32 elementos ativos apresenta os mesmos blocos, porém em quantidade 4 vezes maior até a etapa de somatório coerente.

3.6.1 Modelagem do Filtro Digital

Na primeira etapa do processamento foi utilizado um filtro digital FIR passa-baixa, para a atenuação de frequências acima da banda do transdutor utilizado. A estrutura do filtro FIR é preferida em relação ao filtro IIR devido à sua inerente estabilidade, linearidade de fase e simplicidade de implementação (DEBRUNNER; WANG, 2006). Para o projeto do filtro foi utilizado a ferramenta FDA Tool do Matlab, conforme os parâmetros elencados na Tabela 5:

Tabela 5 – Especificações do Filtro FIR.

Parâmetros	Especificações
Ordem do Filtro (N)	15
Frequência de Amostragem (F_s)	40 MHz
Banda passante (F_{pass})	3,2 MHz
Banda de parada (F_{stop})	8 MHz
Atenuação da banda passante	1 dB
Atenuação da banda de parada	50 dB
Método de projeto do filtro	Equiripple e Least-Squares

Fonte: Autoria própria.

Conforme apresentado no trabalho de Oliveira *et al.* (2019), tipicamente são utilizados dois métodos de projeto de filtro: Equiripple (EQ) e Least-Squares (LS). O método de projeto mais eficiente é o EQ de ordem 15, adotado para este projeto (OLIVEIRA *et al.*, 2019). Na Tabela 6 são apresentados os 16 coeficientes gerados com precisão do tipo *double* para o método EQ. Entretanto, neste trabalho foram avaliados os dois métodos de projeto de filtros. Para cada método foram avaliadas as respostas considerando o projeto do filtro com diferente número de coeficientes: 8 a 32.

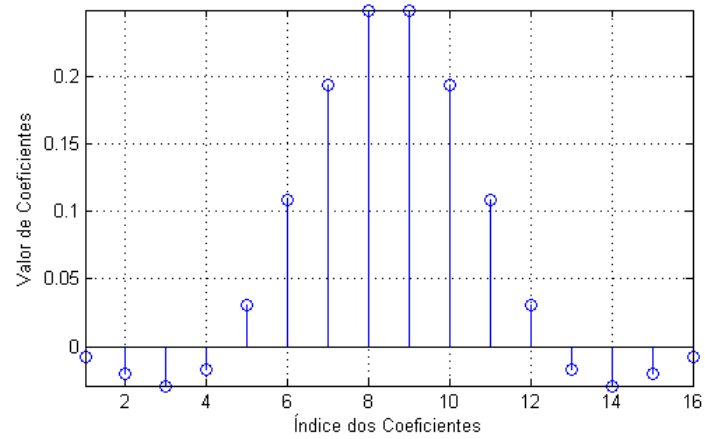
Tabela 6 – Coeficientes do filtro digital FIR passa-baixa pelo método Equiripple.

Índice do Coeficiente	Coeficiente	Índice do Coeficiente	Coeficiente
1	-0,0080566	9	0,24892
2	-0,020332	10	0,19336
3	-0,028816	11	0,10853
4	-0,016457	12	0,03035
5	0,03035	13	-0,016457
6	0,10853	14	-0,028816
7	0,19336	15	-0,020332
8	0,24892	16	-0,0080566

Fonte: Autoria própria.

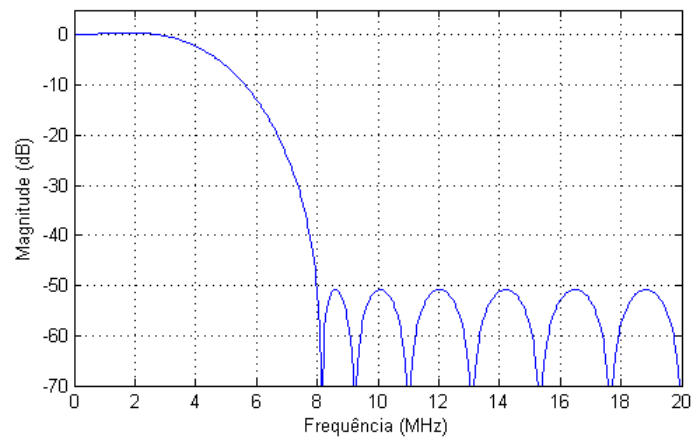
Adicionalmente, como exemplo para validação e estudo das características desse filtro, foram geradas a resposta ao impulso, resposta em magnitude e resposta em fase, mostradas na Figura 23, Figura 24 e Figura 25, respectivamente. Observa-se nesse exemplo a simetria da resposta ao impulso, a frequência de parada em aproximadamente 8 MHz com atenuação de - 50 dB e a resposta em fase linear na faixa de interesse.

Figura 23 – Resposta ao impulso do filtro FIR passa-baixa com ordem 15.



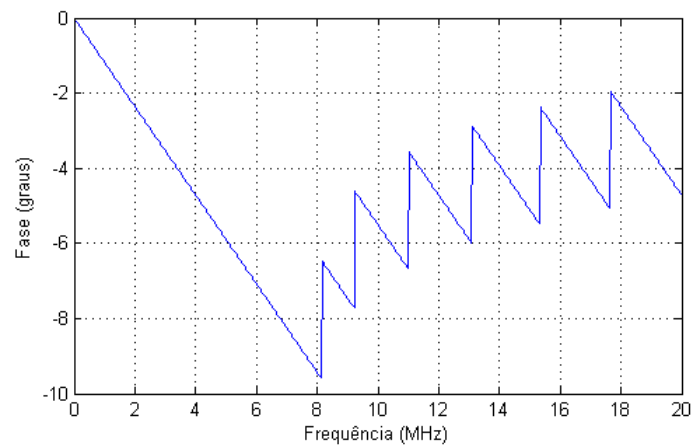
Fonte: Autoria própria.

Figura 24 – Resposta em magnitude do filtro FIR passa-baixa com ordem 15.



Fonte: Autoria própria.

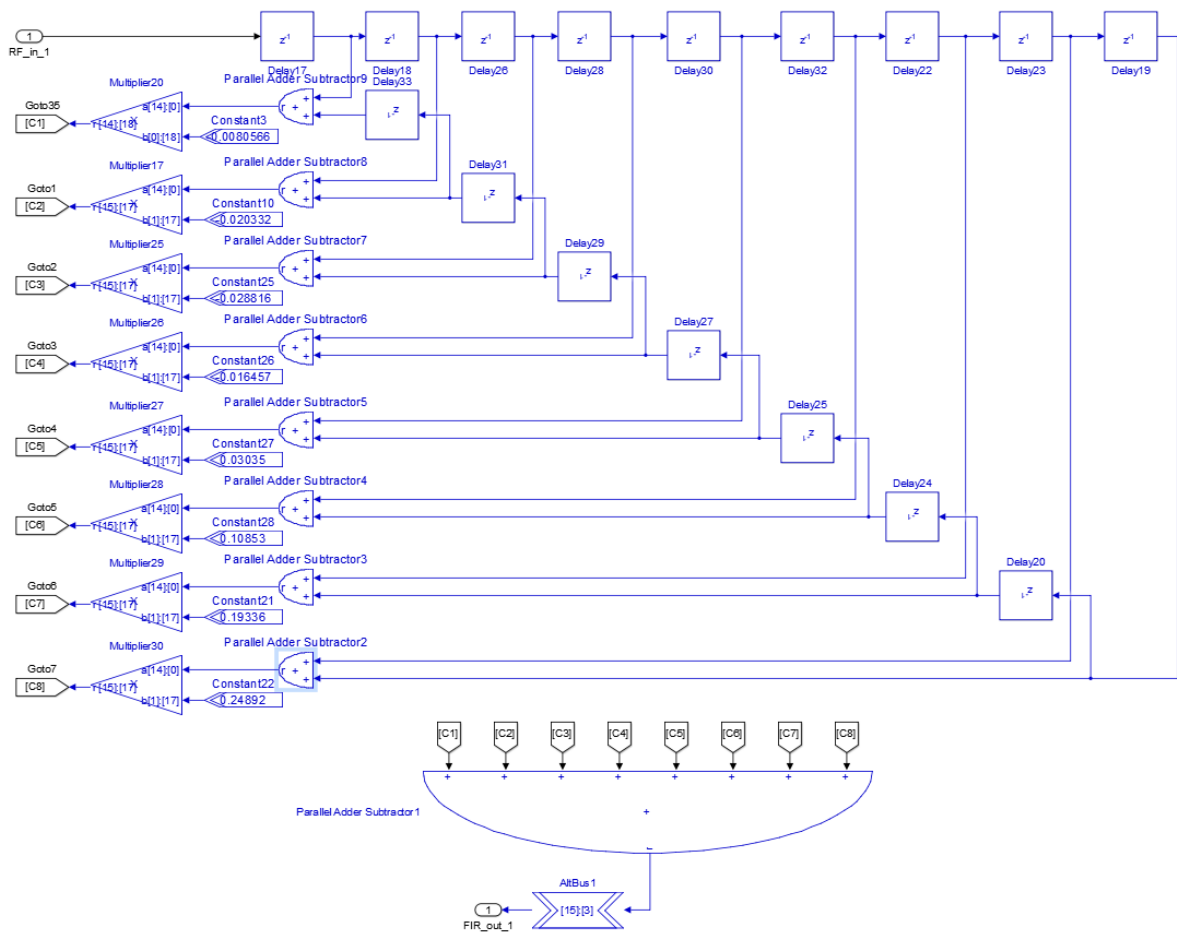
Figura 25 – Resposta em fase do filtro FIR passa-baixa com ordem 15.



Fonte: Autoria própria.

O filtro projetado foi implementado utilizando componentes multiplicadores, somadores e blocos de atraso da biblioteca DSP Builder. A partir da característica simétrica encontrada na resposta ao impulso, é possível espelhar os coeficientes, com intuito de reduzir o custo computacional do sistema. Essa característica foi explorada no projeto do filtro, que pode ser observado na Figura 26. Os blocos “*constant*” que são utilizados na entrada dos multiplicadores para carga dos coeficientes previamente calculados possuem precisão de 17 *bits* na parte fracionária, quantidade máxima de cada multiplicador 18x18 da FPGA. Dessa maneira, evitou-se o desperdício de recursos de *hardware*. O bloco “*AltBus*” é utilizado para ajustar o comprimento do barramento dos sinais processados.

Figura 26 – Projeto do Filtro FIR no Simulink utilizando a biblioteca DSP Builder.

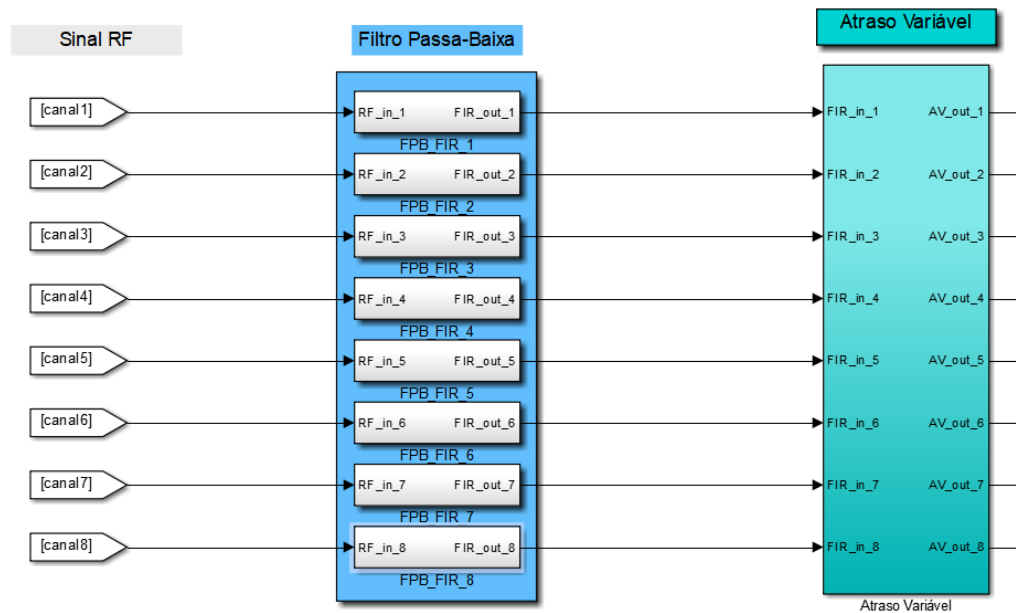


Fonte: Autoria própria.

O projeto da Figura 26 é um subsistema da estrutura “Filtro Passa-Baixa”, mostrado na Figura 27. Os sinais de entrada dessa estrutura são os sinais de RF de uma abertura de 8 elementos. Esse bloco é composto por 8 canais, sendo que cada canal representa um subsistema

do filtro. Após a etapa de filtragem ser concluída, os dados são passados para o bloco de “Atraso Variável”, também composto por 8 canais.

Figura 27 – Conexão dos blocos da estrutura “Filtro Passa-Baixa” no Simulink.



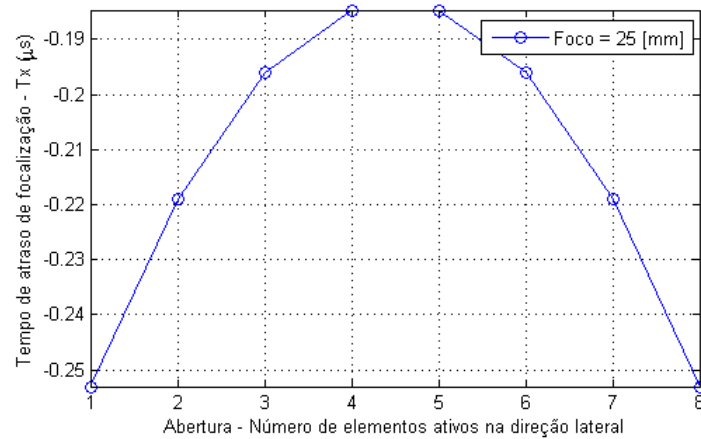
Fonte: Autoria própria.

3.6.2 Modelagem do Bloco de Atraso Variável

O bloco de “Atraso Variável” foi implementado na segunda etapa do processamento do sinal para ajustar os atrasos dos canais relacionados aos elementos da abertura sintética do transdutor AT3C52B.

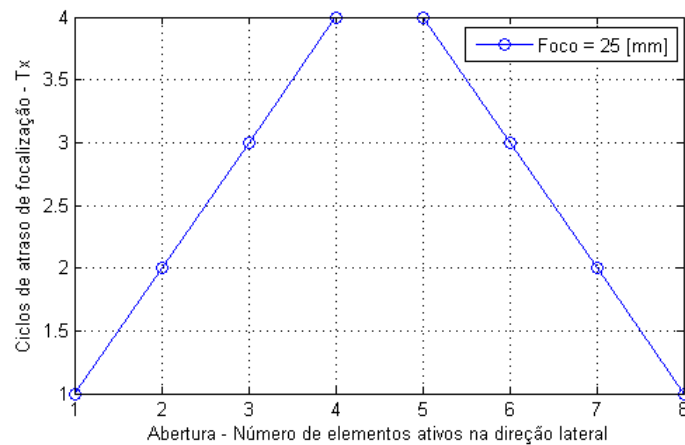
Para calcular esses atrasos foi traçado o perfil temporal de focalização utilizando a ferramenta de simulação de campo acústico Field II (JENSEN, 1996). De acordo com a geometria do transdutor e com a parametrização do ULTRA-ORS, a profundidade do foco foi de 25 mm, utilizando focalização simétrica no eixo z com 8 e 32 elementos de abertura, conforme ilustrados na Figura 28 e Figura 30. Convertendo esse cálculo para ciclos de *clock* de 40 MHz foram traçados os perfis da Figura 29 e Figura 31, respectivamente.

Figura 28 – Perfil temporal de atraso de focalização para a abertura de 8 elementos e profundidade de 25 mm.



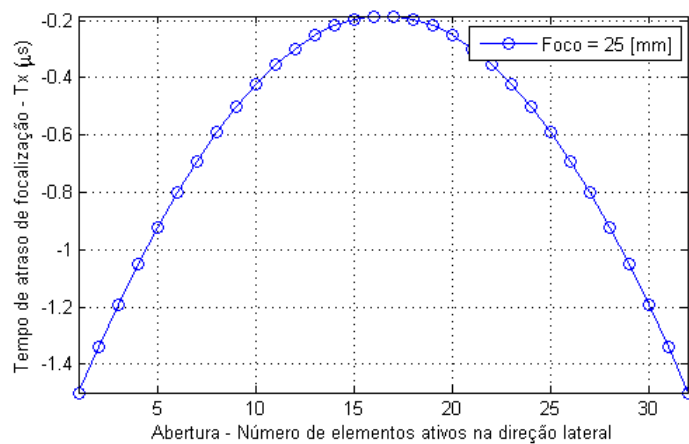
Fonte: Autoria própria.

Figura 29 – Perfil de focalização com ciclos de *clock* de 40 MHz para a abertura de 8 elementos.



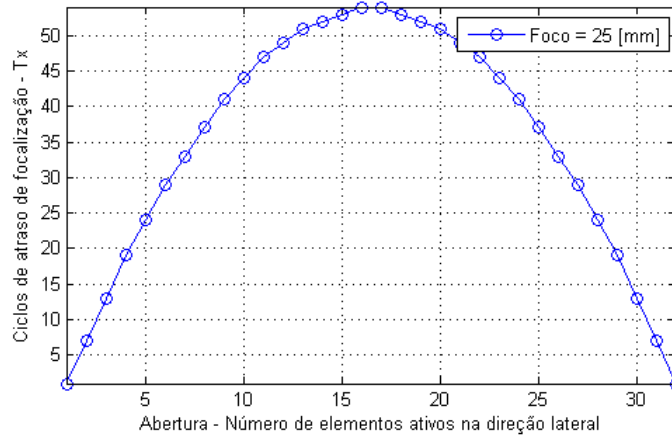
Fonte: Autoria própria.

Figura 30 – Perfil temporal de atraso de focalização a abertura de 32 elementos e profundidade de 25 mm.



Fonte: Autoria própria.

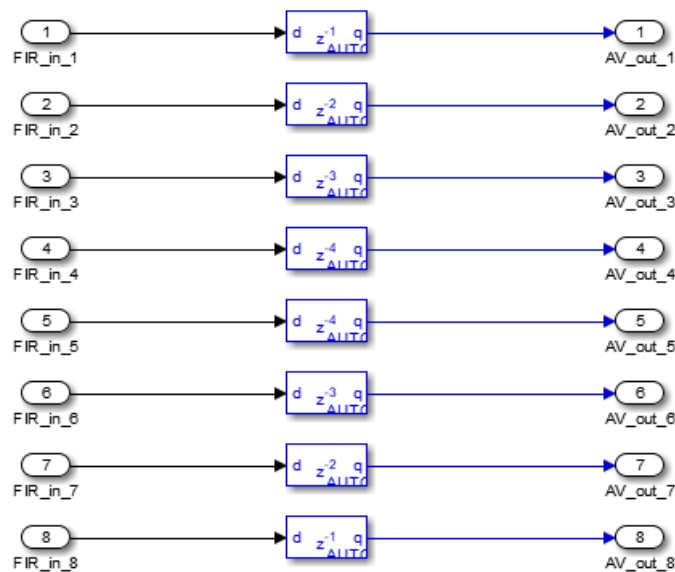
Figura 31 – Perfil de focalização com ciclos de *clock* de 40 MHz para a abertura de 32 elementos.



Fonte: Autoria própria.

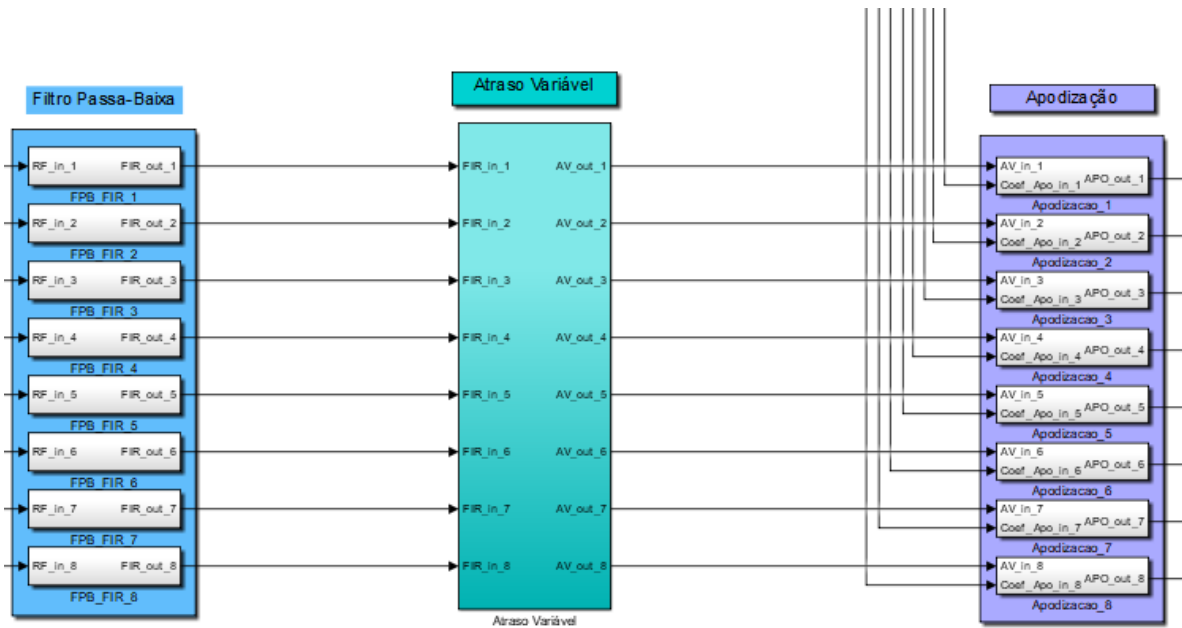
A implementação dessa etapa no Simulink foi realizada utilizando o bloco “*Memory Delay*” que gera atrasos em ciclos de *clock* específicos para cada canal designado no projeto. Para cada bloco foram utilizados os valores pré-calculados dos perfis de focalização com profundidade de 25 mm em ciclos de *clock* de 40 MHz. O projeto da Figura 32 representa o subsistema do bloco de “Atraso Variável”, apresentado na Figura 33. Os 8 canais resultantes do bloco do “Filtro Passa-Baixa” são passados para o bloco de “Atraso Variável” e posteriormente enviados ao bloco de “Apodização”.

Figura 32 – Blocos de “*Memory_Delay*” em ciclos de *clock* de 40 MHz.



Fonte: Autoria própria.

Figura 33 – Conexão do bloco da estrutura “Atraso Variável” no Simulink.

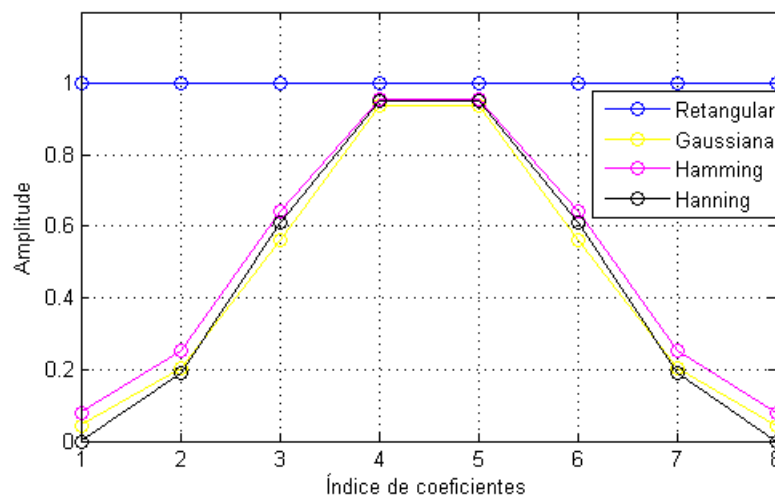


Fonte: Autoria própria.

3.6.3 Modelagem do Bloco de Apodização

O bloco de “Apodização” ajusta o peso de cada canal de acordo com a janela de ponderação utilizada. Na avaliação experimental do projeto, foi utilizada apenas a janela retangular, devido à quantidade de resultados. No entanto, para validação do bloco foi simulado o perfil de apodização para diferentes métodos, conforme a Figura 34.

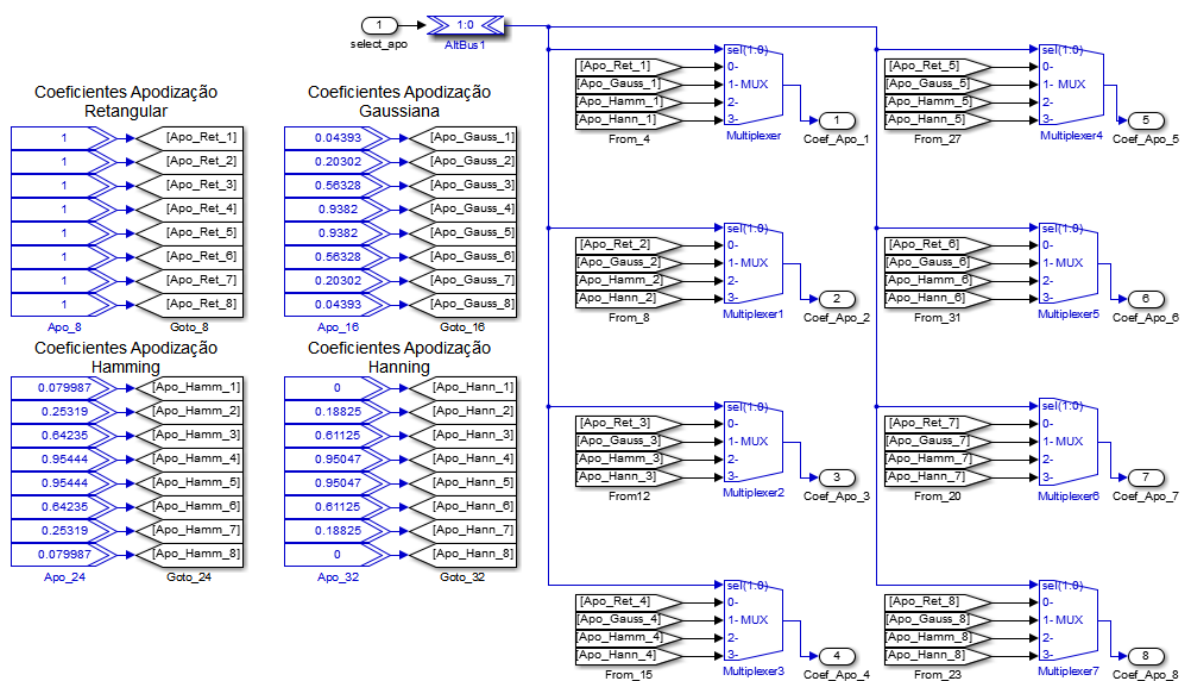
Figura 34 – Modelo de apodização para as janelas Retangular, Gaussiana, Hamming e Hanning.



Fonte: Autoria própria.

Os coeficientes traçados no perfil foram utilizados no subsistema de “Apodização”. Esse subsistema utiliza multiplexadores para comutar entre as 4 janelas de ponderação disponíveis, sendo elas: Retangular, Gaussiana, Hamming e Hanning, conforme apresentadas no subsistema da Figura 35. A seleção da técnica de apodização é realizada através de 2 bits que podem ser configurados pelas chaves disponíveis na placa DE4-230. Como nos casos anteriores, os coeficientes de apodização são pré-calculados e carregados na inicialização do modelo.

Figura 35 – Modelo de escolha de janelamento através de multiplexadores para o projeto com 8 elementos de abertura.

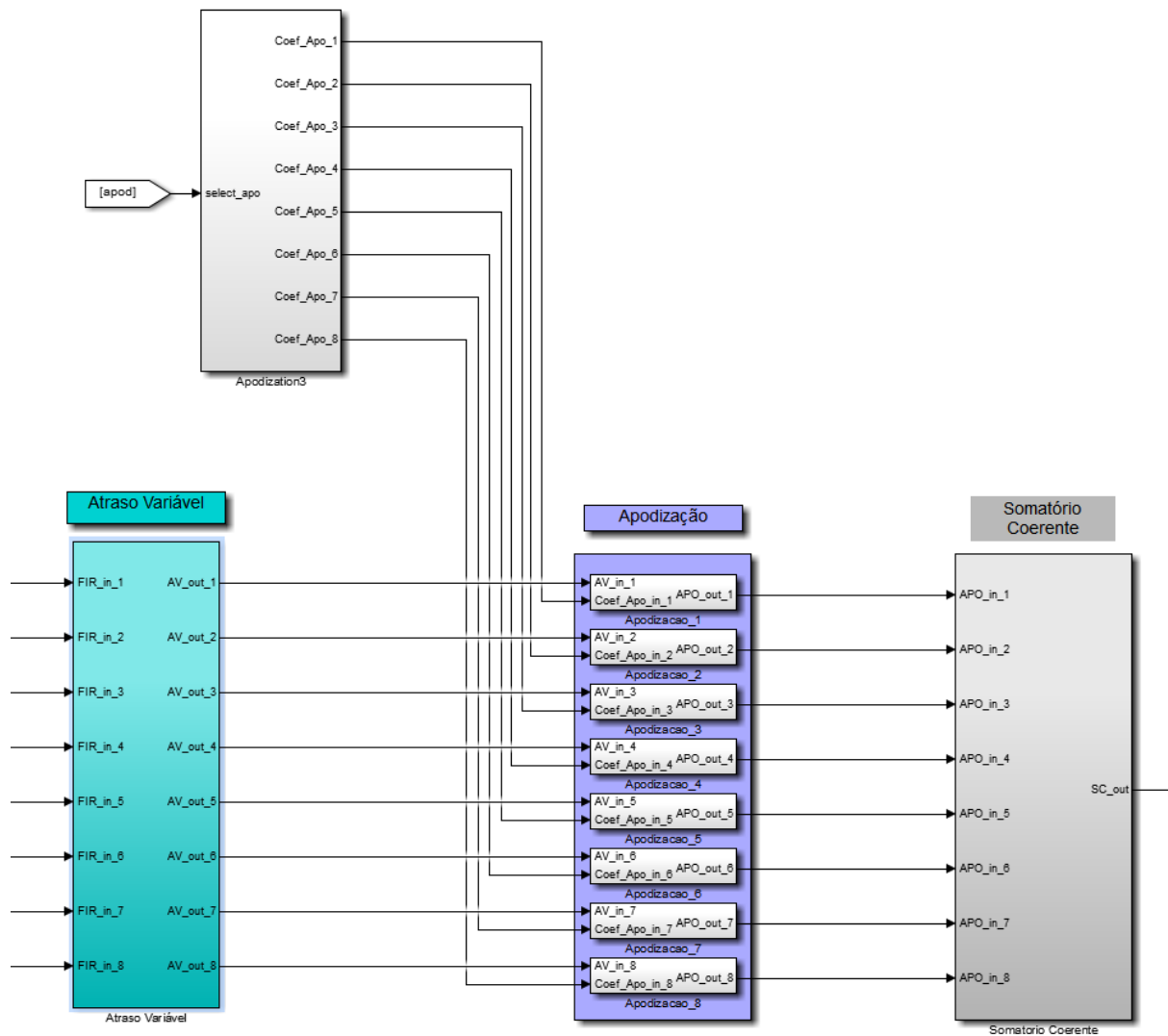


Fonte: Autoria própria.

Para a seleção da janela foi utilizado uma *tag* “apod”, situada na entrada do bloco de coeficientes de apodização. Consecutivamente, os coeficientes resultantes são encaminhados às entradas do bloco de “Apodização”, conforme apresentado na Figura 36.

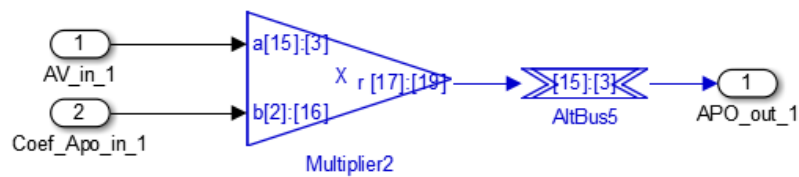
No bloco “Apodização” é realizada a multiplicação entre o coeficiente de apodização selecionado e o canal resultante do bloco “Atraso Variável”, representado no subsistema da Figura 37. O resultado da multiplicação é encaminhado ao bloco “Somatório Coerente”.

Figura 36 – Conexão do bloco da estrutura “Apodização” no Simulink.



Fonte: Autoria própria.

Figura 37 – Modelo de multiplicador para etapa de apodização.

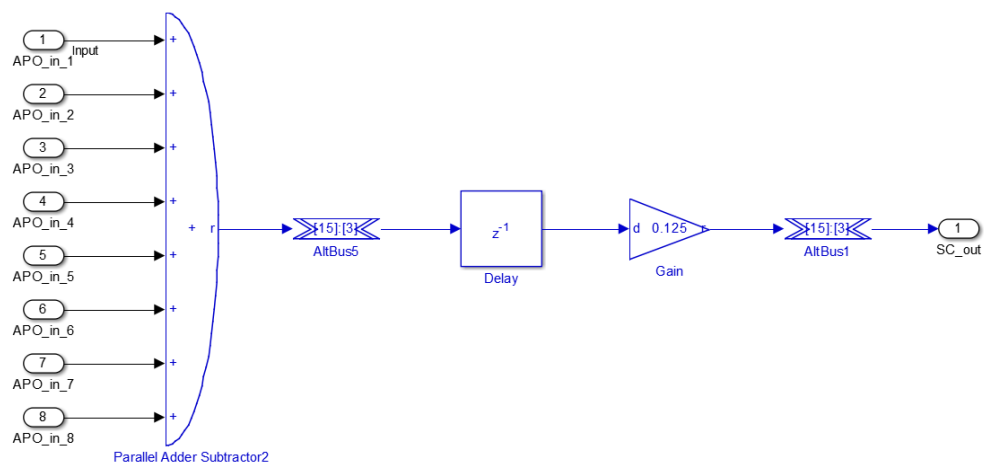


Fonte: Autoria própria.

3.6.4 Modelagem do Bloco de Somatório Coerente

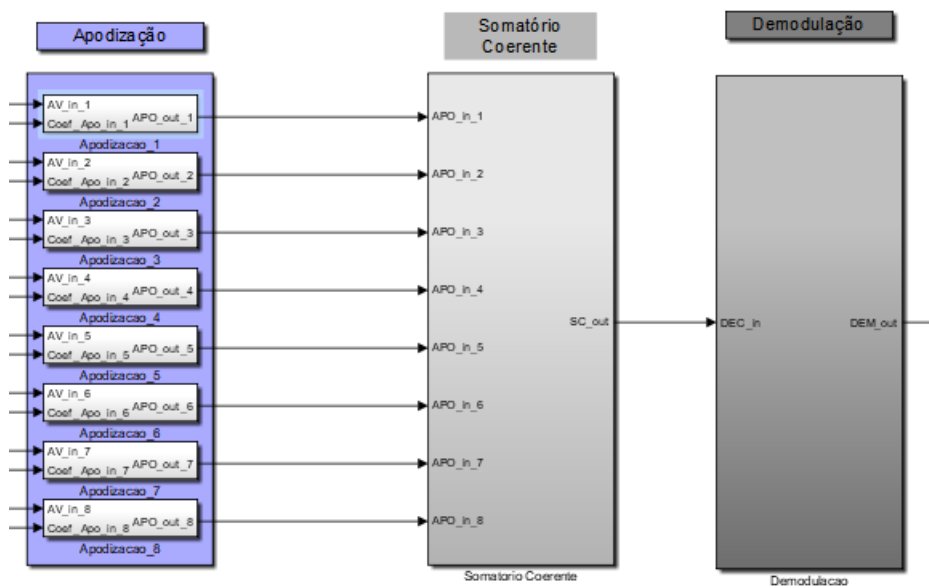
O bloco “Somatório Coerente” foi utilizado para efetuar a soma entre os 8 canais previamente destacados. Nesse subsistema, foi implementado uma função de ganho de 1/8, de forma a nivelar o valor obtido para apenas 1 canal, conforme a Figura 38. O sinal resultante SC_out é enviado ao bloco “Demodulação”, ilustrado na Figura 39.

Figura 38 – Subsistema do bloco “Somatório Coerente”.



Fonte: Autoria própria.

Figura 39 – Conexão do bloco da estrutura “Somatório Coerente” no Simulink.

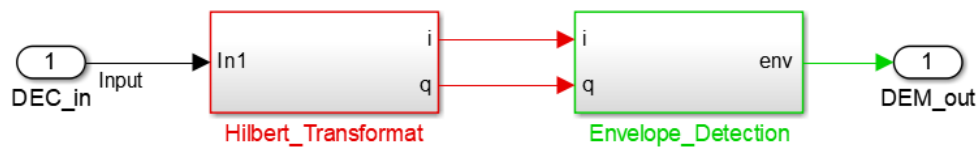


Fonte: Autoria própria.

3.6.5 Modelagem do Bloco de Demodulação

O bloco “Demodulação” é responsável pela execução da demodulação com detecção de envoltória baseada em uma aproximação da Transformada de Hilbert usando estrutura de filtro FIR, ilustrado no subsistema da Figura 40.

Figura 40 – Subsistema do bloco “Demodulação”.



Fonte: Autoria própria.

A Transformada de Hilbert apresenta uma característica de simetria ímpar negativa, alternando entre valores nulos, conforme a sua resposta ao impulso, exemplificada na Tabela 7, na qual são apresentados 33 coeficientes gerados através da ferramenta FDATool.

Tabela 7 – Coeficientes do filtro FIR digital usando o método de projeto com a Transformada de Hilbert.

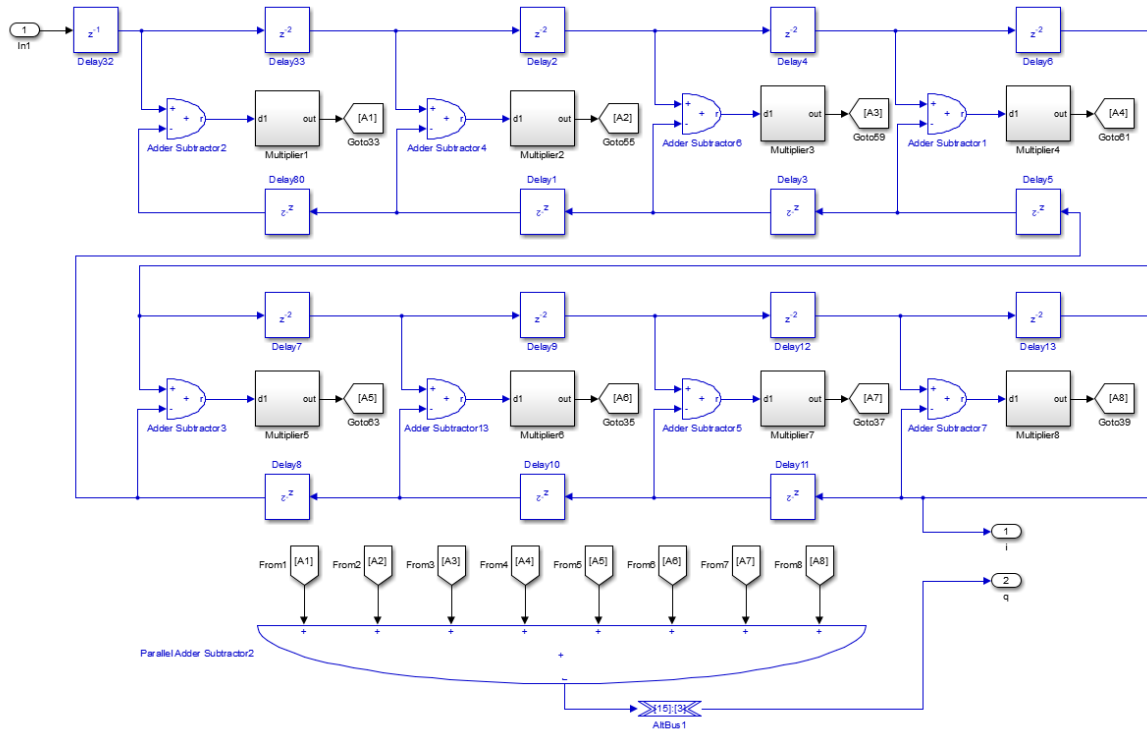
Índice do Coeficiente	Coeficiente	Índice do Coeficiente	Coeficiente
1	0	18	0,6345
2	-0,0319	19	0
3	0	20	0,2058
4	-0,0260	21	0
5	0	22	0,1169
6	-0,0373	23	0
7	0	24	0,0767
8	-0,0531	25	0
9	0	26	0,0531
10	-0,0767	27	0
11	0	28	0,0373
12	-0,1169	29	0
13	0	30	0,0260
14	-0,2058	31	0
15	0	32	0,0319
16	-0,6345	33	0
17	0	-	-

Fonte: Autoria própria.

Dos coeficientes da Tabela 7, são utilizados apenas os valores não nulos para implementação do subsistema “Hilbert Transform” da Figura 41. Essa característica é explorada e adicionalmente incorporado o espelhamento de 8 coeficientes não-nulos, necessitando apenas de $\frac{1}{4}$ do número total de multiplicadores. Na Figura 41 o sinal I é apenas

atrasado pela metade do número total das etapas de atraso e o sinal Q é o resultado do processamento.

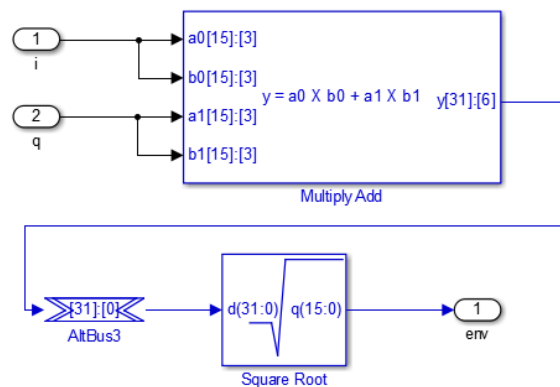
Figura 41 – Bloco da estrutura “Hilbert Transform”.



Fonte: Autoria própria.

Posteriormente, no subsistema da Figura 42 é finalizada a detecção de envoltória. As componentes I e Q são elevadas ao quadrado e calculada a raiz quadrada das somas dos valores resultantes, de forma a gerar a envoltória.

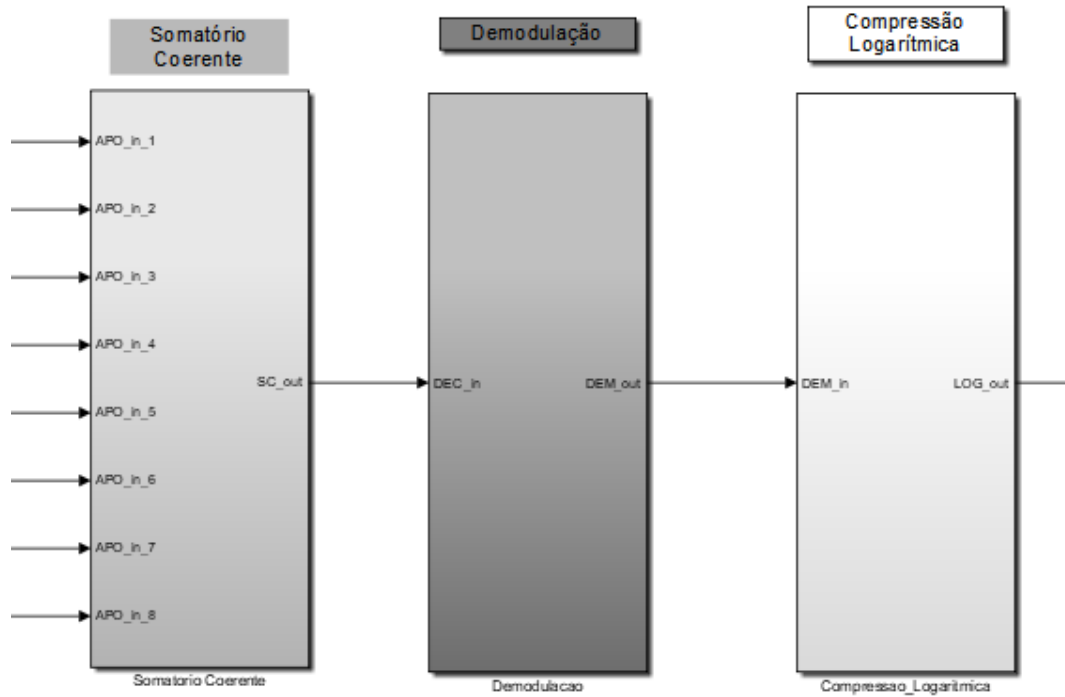
Figura 42 – Bloco de detecção envelope.



Fonte: Autoria própria.

Concluindo a detecção de envoltória do sinal, o resultado é encaminhado ao bloco “Compressão Logarítmica”, mostrado na Figura 43.

Figura 43 – Conexão do bloco da estrutura “Demodulação” no Simulink.



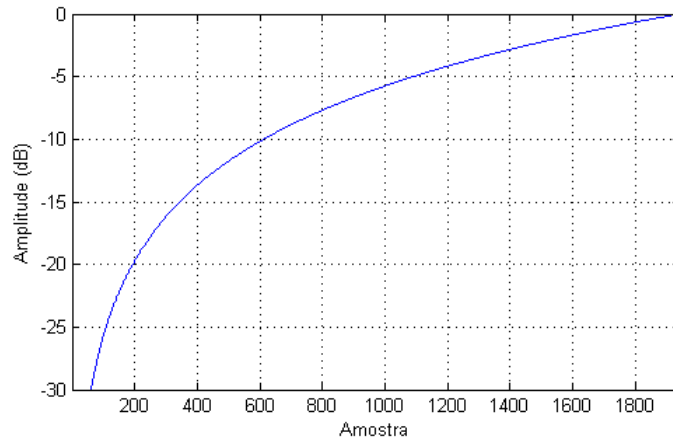
Fonte: Autoria própria.

De forma similar ao filtro FIR modelado, foram avaliados os métodos de projeto de filtros EQ e LS. Para cada método, verificaram-se as respostas do projeto do filtro considerando a ordem de 8 a 64.

3.6.6 Modelagem do Bloco de Compressão Logarítmica

O bloco “Compressão Logarítmica” é a última etapa do processamento embarcado em *hardware*. Essa etapa é responsável por relacionar os valores obtidos pela etapa de detecção de envoltória com um bloco “LUT” de resultados pré-computados da compressão logarítmica. Na Figura 44 é apresentada a curva dos resultados para os valores de entrada no eixo das abcissas (Amostras) e do valor correspondente na escala logarítmica no eixo das ordenadas (Amplitude (dB)), calculados a partir da Equação (6), considerando o valor máximo de 2047 (valor máximo de 11 *bits*).

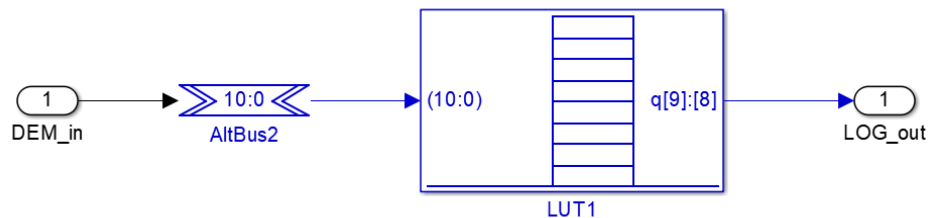
Figura 44 – Tabela de compressão logarítmica limitada em -30 dB.



Fonte: Autoria própria.

O bloco “LUT” utilizado no subsistema da Figura 45 possui endereçamento de 11 *bits*, alcançando os índices de 0 a 2047. A partir da área de trabalho do Matlab, foi importada a tabela de compressão logarítmica, condicionada em 9 *bits* na parte inteira e 8 *bits* na parte fracionária.

Figura 45 – Subsistema de compressão logarítmica.



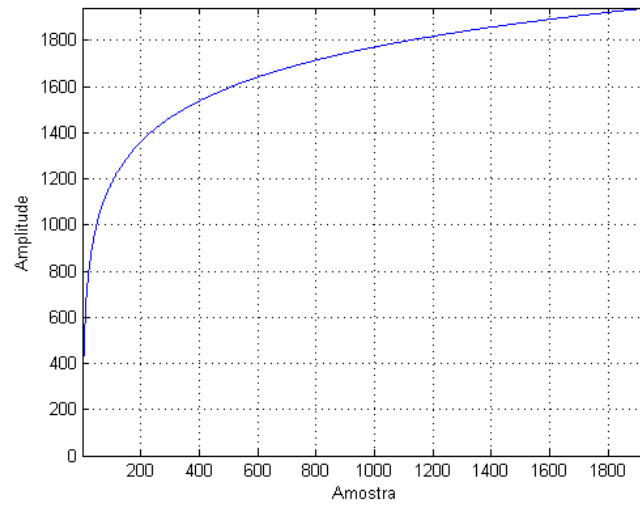
Fonte: Autoria própria.

Para melhorar a resolução dos sinais, os resultados da compressão logarítmica foram normalizados da seguinte forma:

1. Os dados foram multiplicados por 2^6 , através do deslocamento de 6 *bits* para esquerda com o bloco “Binary Point Casting”;
2. Como o valor mínimo da compressão logarítmica é -30 dB, o resultado foi somado a 1920 para gerar somente números não sinalizados para carga da memória FIFO de leitura de dados processados.

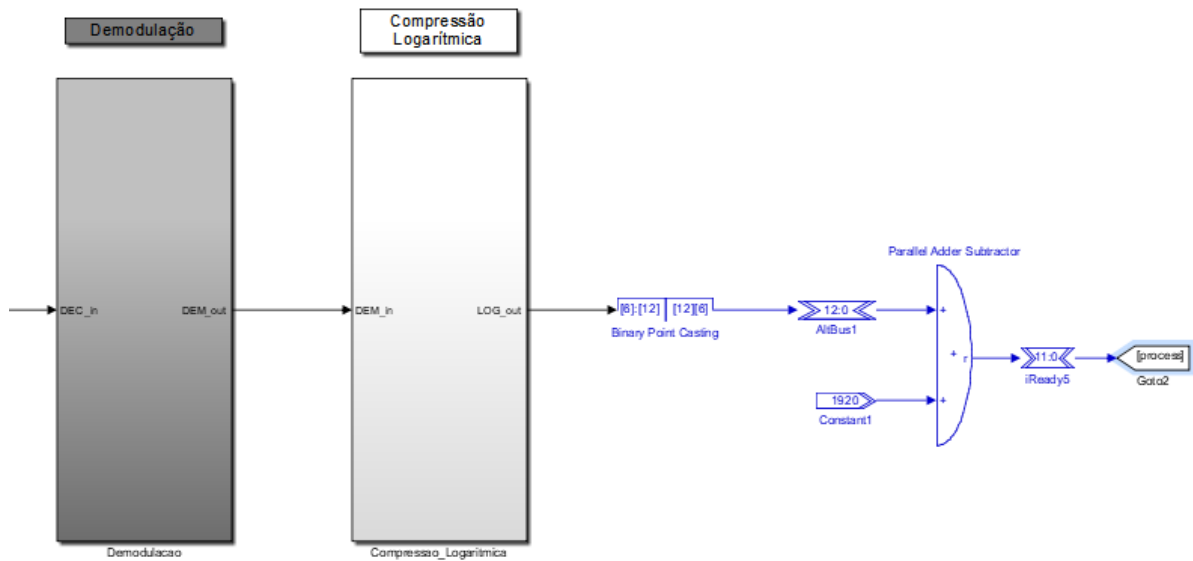
O resultado dessa normalização é apresentado na Figura 46 e a implementação no Simulink é mostrada na Figura 47.

Figura 46 – Valores normalizados correspondentes à compressão logarítmica limitada em -30 dB.



Fonte: Autoria própria.

Figura 47 – Bloco de compressão logarítmica.



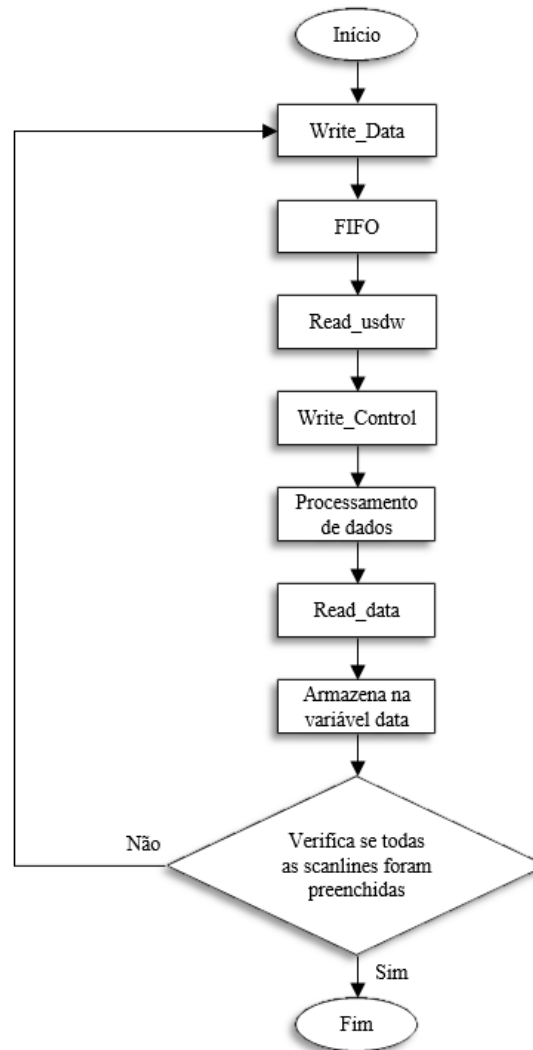
Fonte: Autoria própria.

3.6.7 Integração do Projeto do DSP Builder com a Ferramenta Qsys

Para integração dos blocos do processamento digital do sinal com o processador Nios II que tem a função de gerenciar a leitura, transmissão de dados de alta velocidade, armazenamento, processamento do fluxo de dados controle de dispositivos externos, é

necessário utilizar blocos da interface Avalon. O barramento Avalon é empregado para simplificar o projeto de sistemas digitais, permitindo conectar facilmente componentes da arquitetura de processadores nas FPGA Intel. A sequência de processamento do Nios II para essas funções é apresentada na Figura 48.

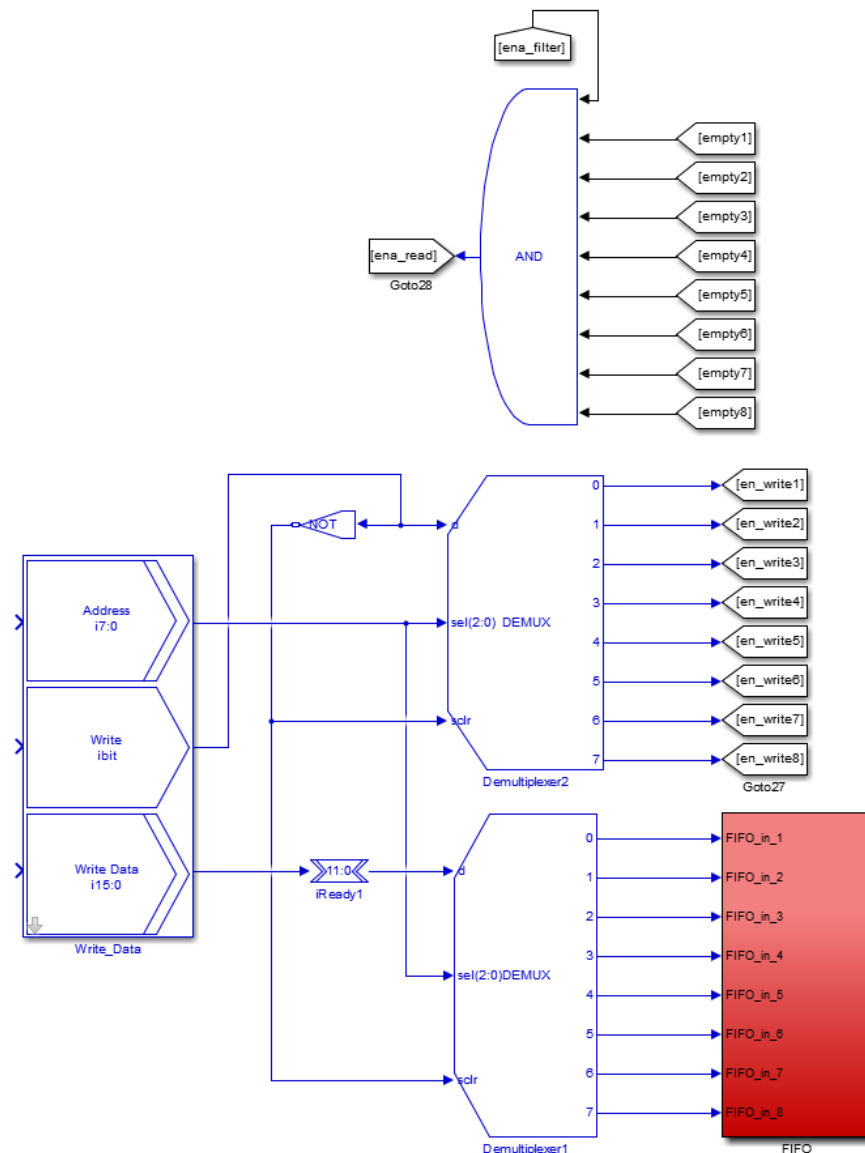
Figura 48 – Fluxograma de coordenação de dados.



Fonte: Autoria própria.

O bloco “Write_Data” apresentado no esquema da Figura 49 é responsável por endereçar e encaminhar os dados recebidos à FIFO correspondente, de acordo com o *tag* “en_write”.

Figura 49 – Blocos de fluxo de dados pré-processamento.

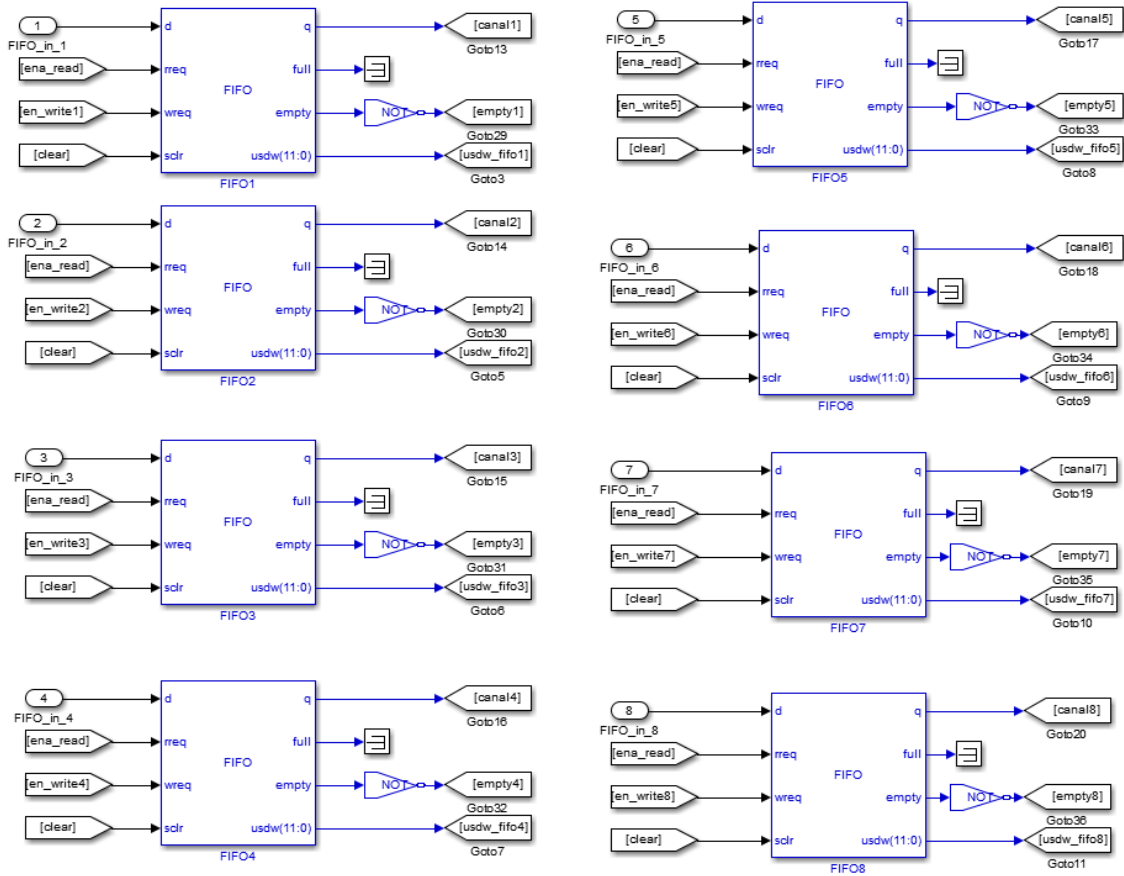


Fonte: Autoria própria.

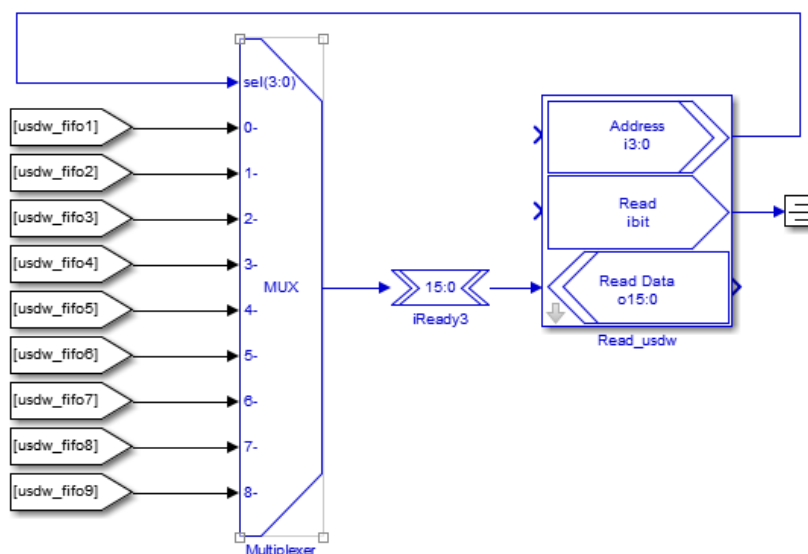
O bloco vermelho em destaque na Figura 49 é composto por 8 memórias FIFO, conforme apresentado no subsistema da Figura 50.

A variável “usdw_fifo”, mostrada no diagrama da Figura 51, é responsável pela leitura da quantidade de palavras presente na FIFO correspondente ao seu índice. Essa leitura é realizada quando a variável completa 2046 *words*, implicando que a FIFO correspondente está cheia. Consecutivamente, os dados são direcionados para a FIFO seguinte.

Figura 50 – Subsistema composto por 8 memórias FIFO.



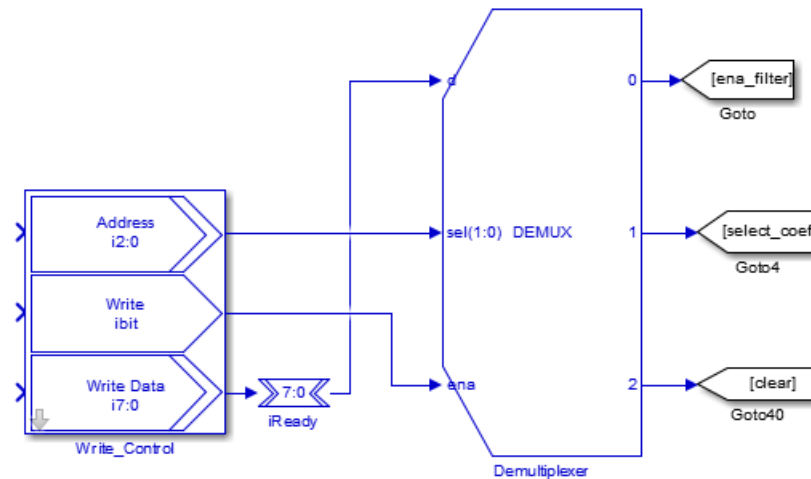
Fonte: Autoria própria.

Figura 51 – Blocos de leitura de *words* das FIFOs.

Fonte: Autoria própria.

Depois de preencher todas as 8 FIFOs, o bloco “Write_Control”, situado no diagrama da Figura 52, habilita o processamento digital do sinal através da *tag* “ena_filter”, no qual cada FIFO encaminha os dados ao “canal” referente ao seu índice.

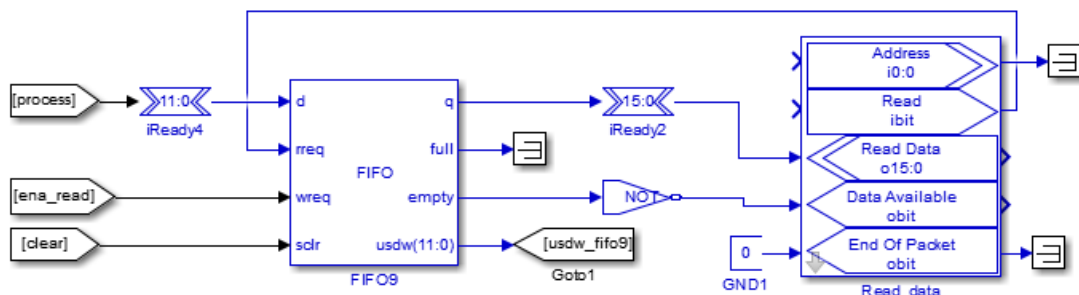
Figura 52 – Bloco de habilitação do processamento de dados.



Fonte: Autoria própria.

Ao final do processamento, os dados do sinal processado são encaminhados à *tag* “process”, a qual é lida através da FIFO referente ao bloco “Read_data” de leitura de dados pós-processamento da Figura 53. Nos modelos implementados, o *clock* da FPGA é realizado em uma frequência de 40 MHz.

Figura 53 – Bloco de leitura de dados pós-processamento.



Fonte: Autoria própria.

3.7 FERRAMENTAS PARA AVALIAÇÃO DE SINAIS E IMAGENS

Os modelos das etapas de processamento implementadas neste projeto foram avaliados a partir da função de custo raiz do erro quadrático normalizado (NRMSE) e do perfil do erro absoluto. A função NRMSE é obtida pela Equação (7):

$$NRMSE = \sqrt{\frac{\sum_{n=0}^{M-1} |Y(n) - hy(n)|^2}{\sum_{n=0}^{M-1} |Y(n) - \bar{hy}|^2}} * 100, \quad (7)$$

na qual n é o índice da amostra, M é número de amostras, $Y(n)$ é a amostra do modelo, $hy(n)$ é a referência obtida através do *script* do Matlab e \bar{hy} é a média das amostras do modelo implementado no Simulink ou na FPGA.

O NRMSE implica na diferença relativa entre o resultado conhecido como modelo padrão e o dado mensurado. Os valores menores de 10% indicam excelente desempenho do modelo; valores entre 10 a 20% indicam bom desempenho; os modelos com resultados acima de 30% configuram desempenho pobre (CHANG *et al.*, 2007).

Adotou-se como modelo de referência os algoritmos validados em pesquisas anteriores desenvolvidos via *scripts* do Matlab com as mesmas etapas de processamento implementadas neste trabalho (ASSEF, 2013; FERREIRA, 2017). Para a avaliação entre as imagens em Modo B geradas pelo *script* do Matlab, modelo projetado no Simulink e pelos dados obtidos através da placa de desenvolvimento DE4-230 pós-processamento, foi utilizado a comparação quantitativa de resolução de contraste (CR) e taxa de contraste-ruído (CNR), apresentadas nas Equações (8) e (9), respectivamente:

$$CR = 1 - \frac{\mu_t}{\mu_b} \quad (8)$$

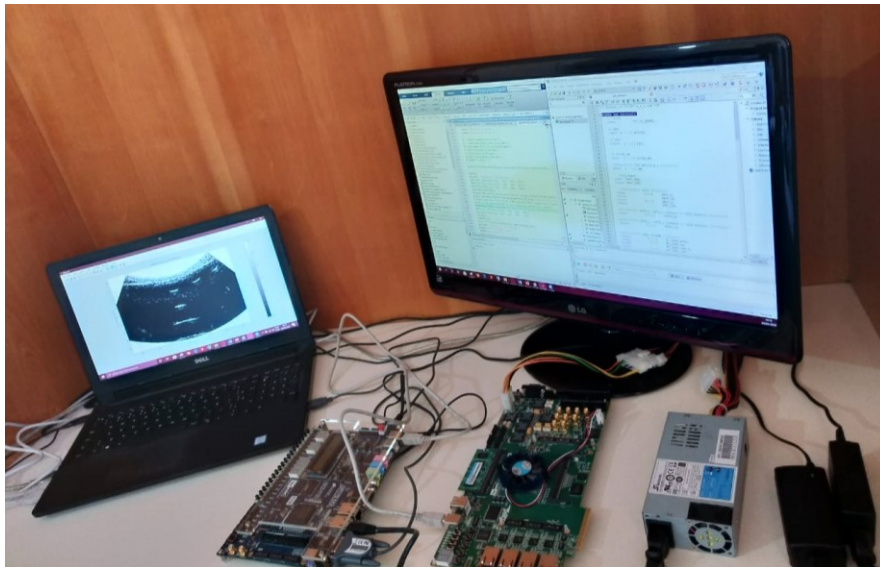
$$CNR = 1 - \frac{|\mu_t - \mu_b|}{\sigma_b} \quad (9)$$

nas quais μ_t é a intensidade em dB do alvo, em escala de cinza, μ_b é a média da intensidade da região fundo e σ_b é o desvio padrão da intensidade da região de fundo (ZHENG; ZHOU, 2016).

O trabalho de pesquisa foi realizado com um notebook Dell Inspiron 15-3567 com processador Intel CORE i5 7th Gen, memória RAM do tipo DDR4 de 8 GB e comportando um

HD de 1 TB. A bancada mostrada na Figura 54 apresenta o notebook, monitor LCD, placa DE2-115 com a FPGA Cyclone IV E e a placa DE4-230 com a FPGA Stratix IV GX, além das fontes de alimentação. No monitor são intercalados os *softwares* do Quartus Prime e o Matlab. Na tela do notebook é apresentada uma imagem em Modo B, como exemplo referente ao processamento decorrido.

Figura 54 – Bancada de desenvolvimento e teste formada por um notebook, placas DE2-115 e DE4-230, e um monitor LCD.



Fonte: Autoria própria.

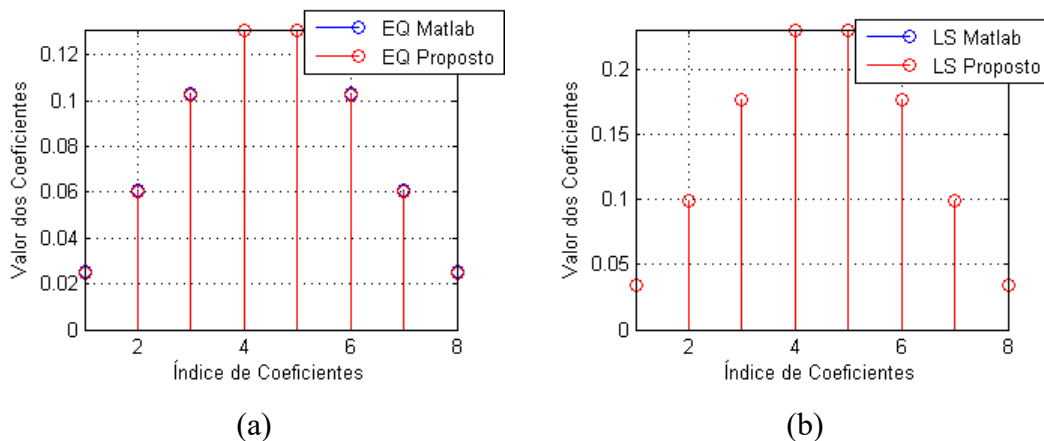
4 RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados qualitativos e quantitativos das simulações de algoritmos escritos em *scripts* do Matlab e modelos desenvolvidos no Simulink. Essas análises foram realizadas em comparação com a implementação experimental das etapas de processamento digital de sinais embarcado em *hardware* reconfigurável para geração de imagens em Modo B. Inicialmente, foram validadas as etapas cujas estruturas são mais complexas, sendo elas a filtragem digital e a demodulação e detecção de envoltória. Posteriormente, avaliou-se individualmente cada etapa proposta de processamento até a caracterização das imagens.

4.1 AVALIAÇÃO DA FILTRAGEM DIGITAL PASSA-BAIXA

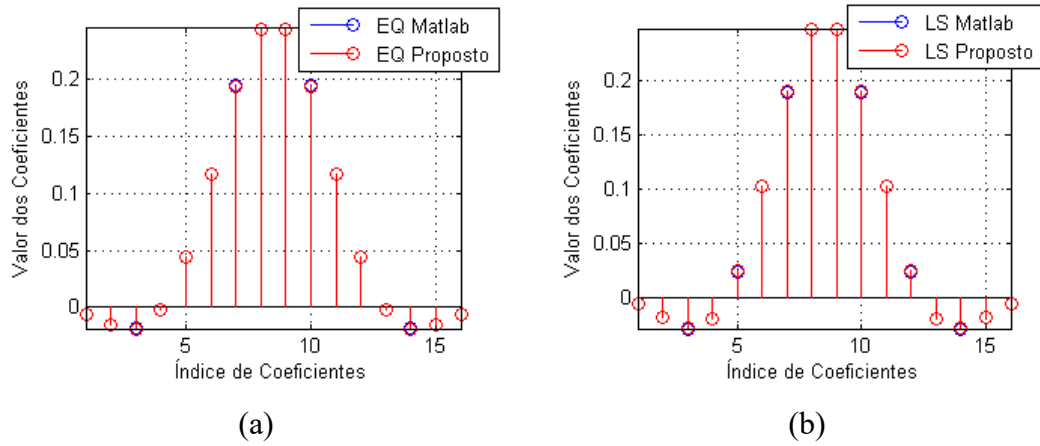
Para a avaliação dos projetos de filtro digital FIR passa-baixa foram realizadas comparações entre os resultados gerados por algoritmos de *script* do Matlab e os modelos implementados no Simulink das respostas ao impulso do filtro projetados com os métodos EQ e LS. Os dados gerados pelo *script* estão no formato *double* e nos modelos considerou-se 1 *bit* na parte inteira e 17 *bits* na parte fracionária para os seguintes números de coeficientes: 8 (Figura 55), 16 (Figura 56), 24 (Figura 57), 32 (Figura 58) e 64 (Figura 59). Todos esses resultados indicam que o modelo do filtro proposto foi realizado com sucesso, sendo que a diferença é resultante da resolução imposta pelo tipo das variáveis.

Figura 55 – Comparação entre as respostas ao impulso do filtro digital FIR passa-baixa implementado via *script* e modelado no Simulink utilizando os métodos: (a) EQ com 8 coeficientes; (b) LS com 8 coeficientes.



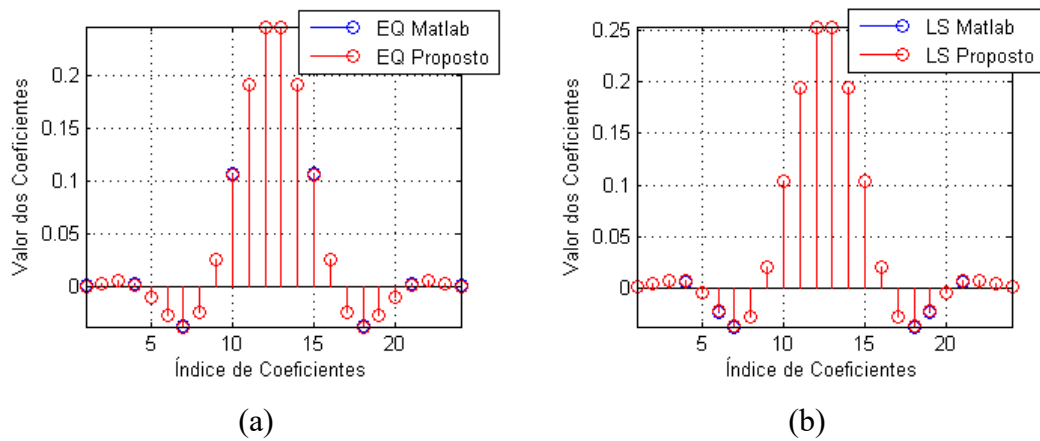
Fonte: Autoria própria.

Figura 56 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via *script* e modelado no Simulink utilizando os métodos: (a) EQ com 16 coeficientes; (b) LS com 16 coeficientes.



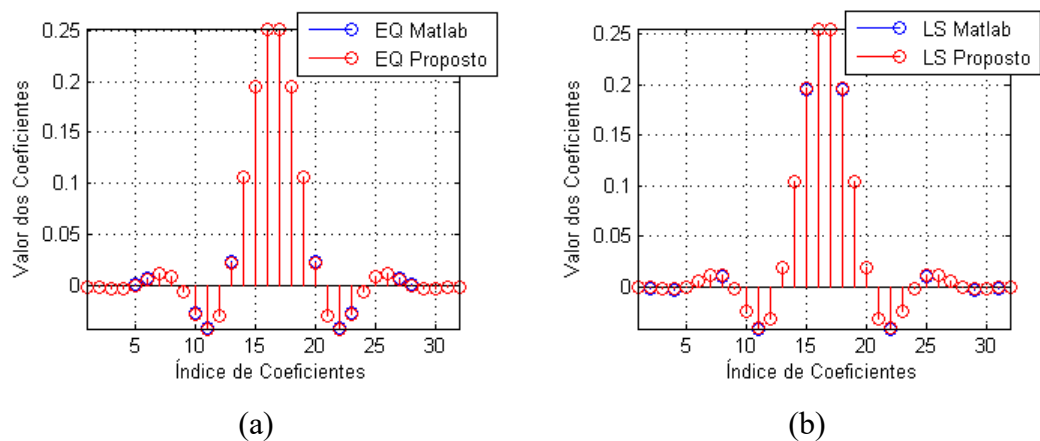
Fonte: Autoria própria.

Figura 57 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via *script* e modelado no Simulink utilizando os métodos: (a) EQ com 24 coeficientes; (b) LS com 24 coeficientes.



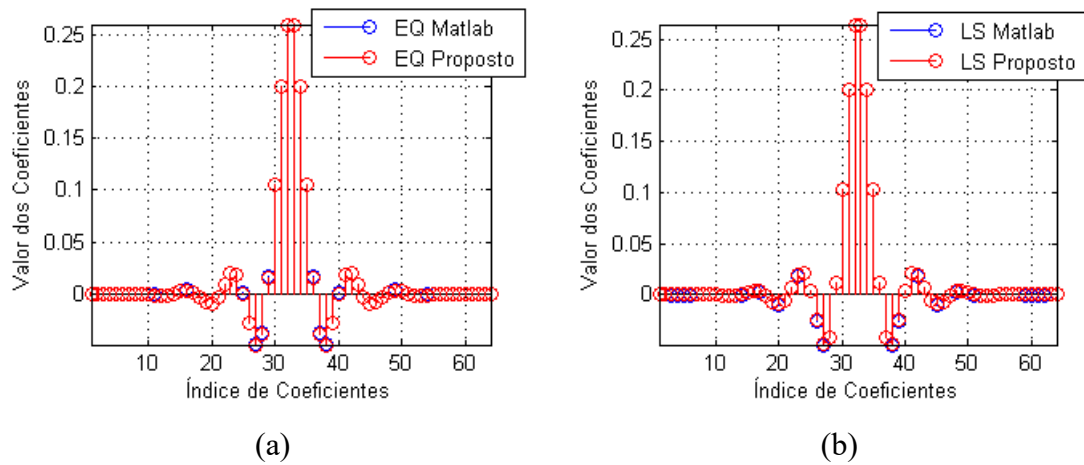
Fonte: Autoria própria.

Figura 58 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via *script* e modelado no Simulink utilizando os métodos: (a) EQ com 32 coeficientes; (b) LS com 32 coeficientes.



Fonte: Autoria própria.

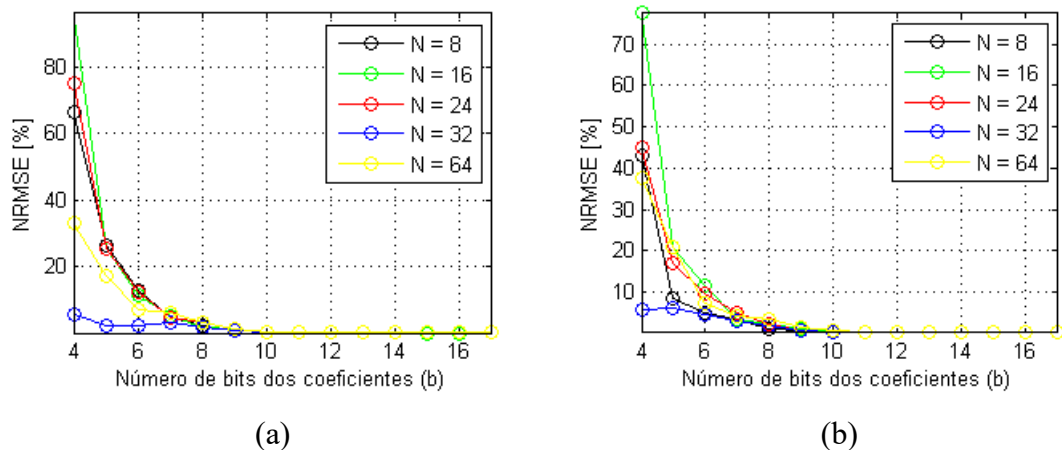
Figura 59 – Comparação entre as respostas ao impulso do filtro FIR passa-baixa implementado via *script* e modelado no Simulink utilizando os métodos: (a) EQ com 64 coeficientes; (b) LS com 64 coeficientes.



Fonte: Autoria própria

Na Figura 60 é mostrada a comparação do número de *bits* (b) da parte fracionário dos coeficientes em relação ao número de coeficientes (N) do filtro. A avaliação do projeto do filtro digital FIR é realizada através da função de custo NRMSE para o método EQ na Figura 60(a) e LS na Figura 60(b).

Figura 60 – Comparação quantitativa da implementação do filtro digital FIR com variação do número de *bits* (b) da parte fracionária dos coeficientes (N) para os métodos: (a) EQ e (b) LS.



Fonte: Autoria própria

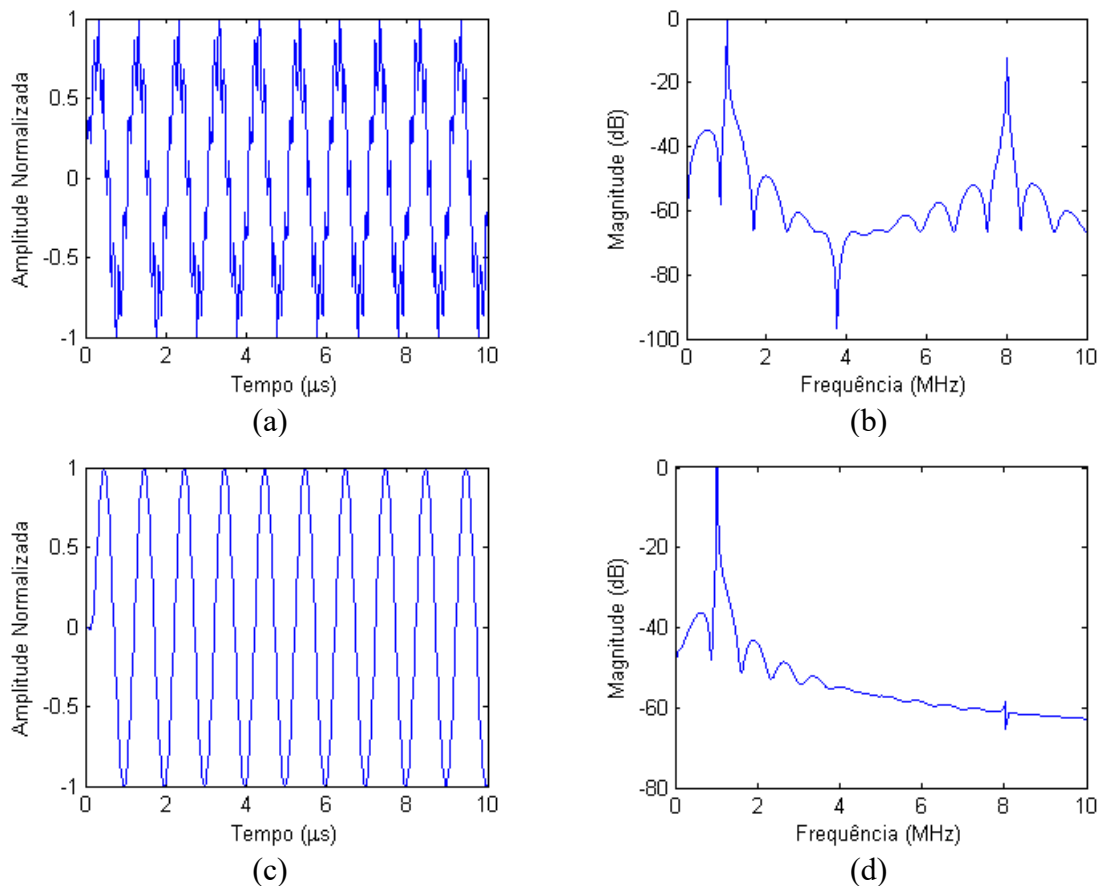
Pode-se verificar na Figura 60 que os resultados convergem em ambos os métodos de projeto para a estrutura com 8 bits dos coeficientes ($b=8$). Através da comparação quantitativa do NRMSE bem como a análise das curvas, avaliou-se que o método EQ é mais eficiente para o projeto (OLIVEIRA *et al.*, 2019). Visando melhorar a resolução dos resultados, para as

demais etapas do projeto o número de *bits* (b) da parte fracionária foi arbitrado como 17, uma vez que essa é a quantidade máxima de *bits* para um bloco multiplicador 18x18 da FPGA.

Além da avaliação da resposta ao impulso, realizaram-se três avaliações com diferentes sinais de entrada com a estrutura do filtro FIR de 16 coeficientes: sinal senoidal de baixa e alta frequência somados (Figura 61), sinal Chirp (Figura 62) e sinal real US correspondendo a uma linha de RF (Figura 63).

Na Figura 61(a) é mostrado um sinal senoidal de entrada com frequência de 1 MHz adicionado a uma senoide de 8 MHz (25% da amplitude). O espectro em frequência do sinal de entrada obtido por FFT é mostrado na Figura 61(b), evidenciando as componentes em frequência de 1 MHz e 8 MHz. Na Figura 61(c) é apresentado o sinal de saída após a etapa de filtragem digital, na qual é possível notar claramente que o sinal de alta frequência foi eliminado. Esse resultado é confirmado no espectro em frequência da Figura 61 (d), no qual a componente de 8 MHz foi atenuada em mais de 50 dB, conforme a especificação do filtro.

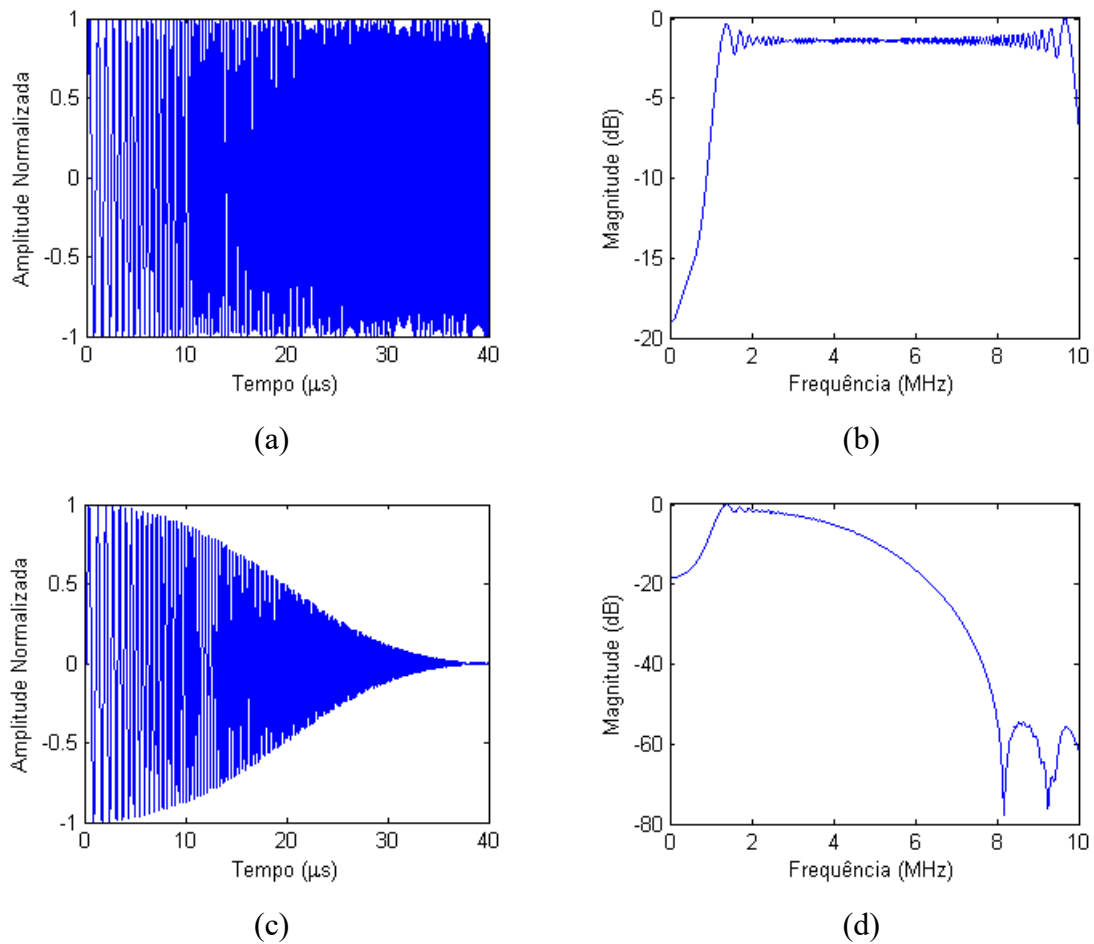
Figura 61 – Avaliação da resposta do filtro FIR com aplicação de um sinal senoidal de 1MHz adicionado ao sinal senoidal de 8 MHz (25 % de amplitude). (a) Sinal de entrada do filtro. (b) Espectro em frequência do sinal de entrada. (c) Sinal de saída filtrado. (d) Espectro em frequência do sinal filtrado.



Fonte: Autoria própria

Na Figura 62(a) é apresentado um sinal Chirp com variação de frequência de 1 a 10 MHz e na Figura 62 (b) o seu espectro em frequência. Após a filtragem é obtido o sinal da Figura 62 (c) e sua respectiva resposta espectral na Figura 62(d). Nessa avaliação, verifica-se que as componentes próximas à banda de parada foram atenuadas em aproximadamente 50 dB. Entretanto, parte do sinal de interesse também é atenuado, indicando que o filtro pode ser melhorado. Por exemplo, na frequência de 6 MHz o sinal é atenuado em 20 dB.

Figura 62 – Avaliação da resposta do filtro FIR com aplicação de um sinal Chirp de 1 a 10 MHz. (a) Sinal de entrada do filtro. (b) Espectro em frequência do sinal de entrada. (c) Sinal de saída filtrado. (d) Espectro em frequência do sinal filtrado.

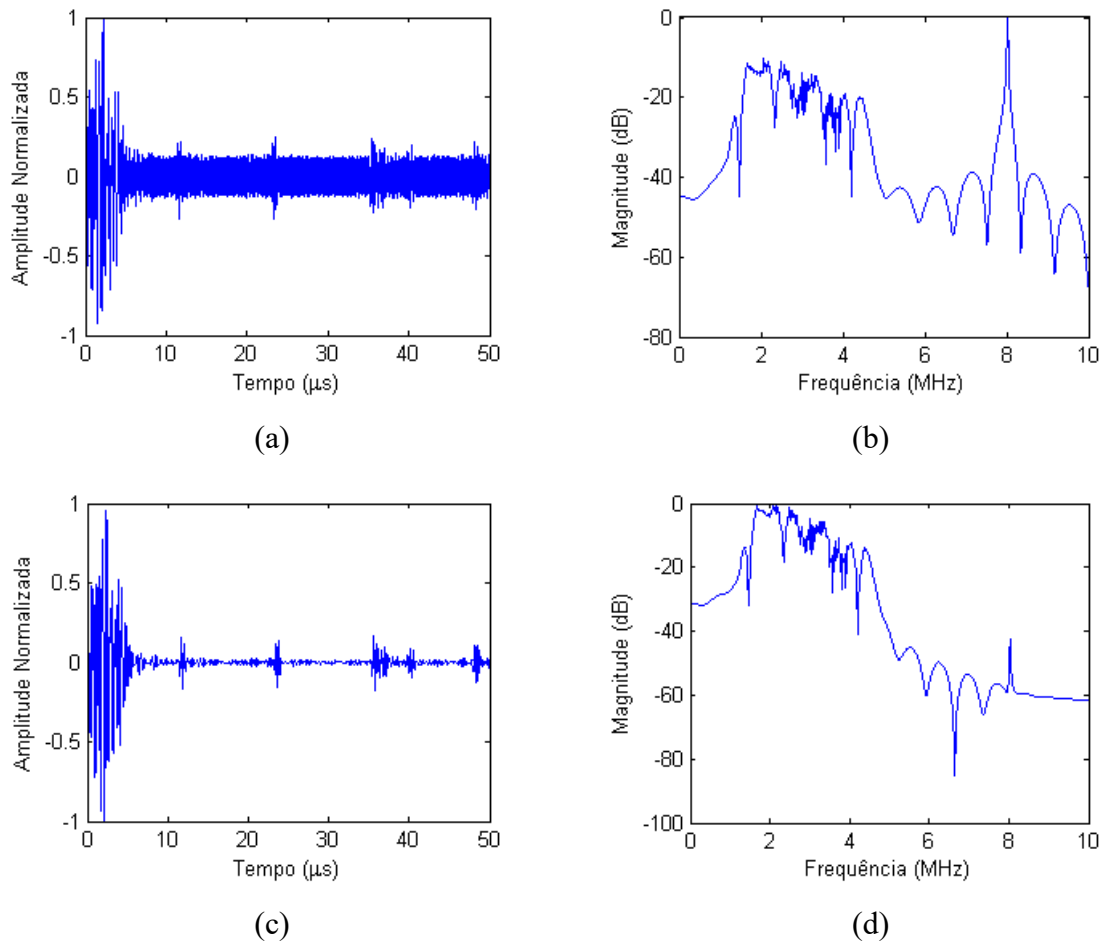


Fonte: Autoria própria

Na Figura 63(a) é apresentado um sinal real de RF de amplitude normalizada adicionado a uma senoide de 8 MHz (15% da amplitude). Na excitação foi utilizado um pulso com frequência central de 3,2 MHz com perfil Gaussiano de 3 ciclos. Na Figura 63(b) é mostrado o seu comportamento espectral evidenciando a componente de 8 MHz. Após o

processo de filtragem, o sinal senoidal de 8 MHz é atenuado em aproximadamente 42 dB, conforme ilustra a Figura 63(c) e o seu espectro na Figura 63(d).

Figura 63 – Avaliação da resposta do filtro FIR com aplicação de um sinal real de RF adicionado a uma senoide de 8 MHz. (a) Sinal de entrada do filtro. (b) Espectro em frequência do sinal de entrada. (c) Sinal de saída filtrado. (d) Espectro em frequência do sinal filtrado.



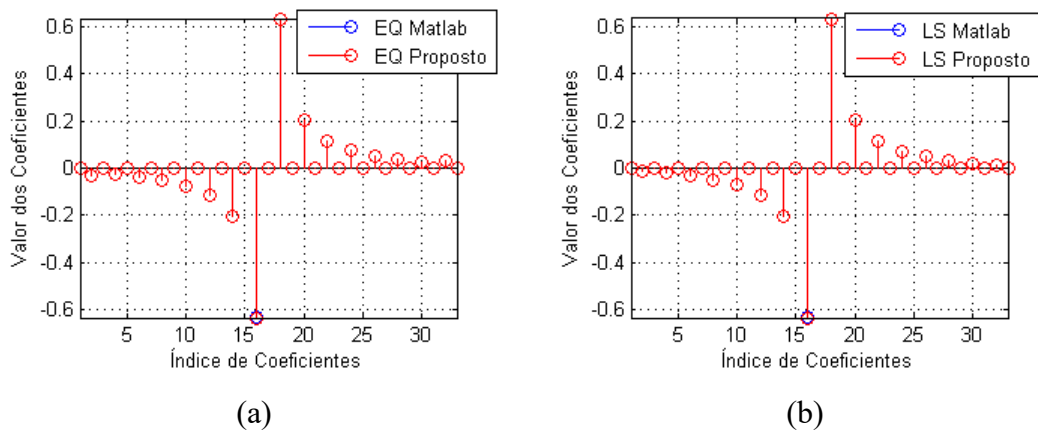
Fonte: Autoria própria

4.2 AVALIAÇÃO DA DEMODULAÇÃO E DETECÇÃO DE ENVOLTÓRIA

Nessa etapa, avaliou-se o modelo do detector de envoltória de forma similar à etapa de filtragem. Realizaram-se comparações entre os resultados gerados por algoritmos de *script* do Matlab e os modelos implementados no Simulink de uma aproximação da Transformada de Hilbert baseada em filtro FIR projetado com os métodos EQ e LS. Os dados gerados pelo *script* estão no formato *double* e nos modelos considerou-se 1 *bit* na parte inteira e 17 *bits* na parte fracionária para uma estrutura de filtro com ordem 32.

Na Figura 64(a) e na Figura 64(b) são apresentadas as respostas ao impulso do filtro FIR projetado com os coeficientes da Transformada de Hilbert para os métodos LS e EQ, respectivamente. Esses resultados indicam uma ótima concordância entre os algoritmos e os modelos implementados.

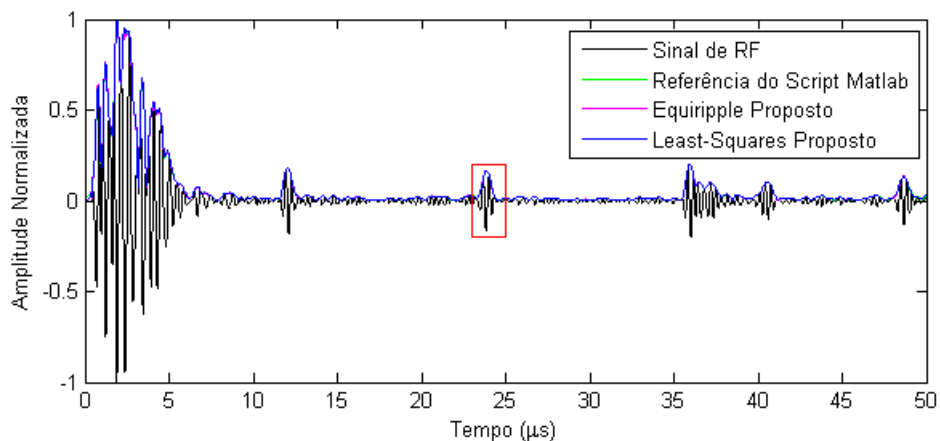
Figura 64 – Comparação entre as respostas ao impulso do detector de envoltória implementado via *script* e modelado no Simulink utilizando os métodos: (a) EQ com 33 coeficientes; (b) LS com 33 coeficientes.



Fonte: Autoria própria

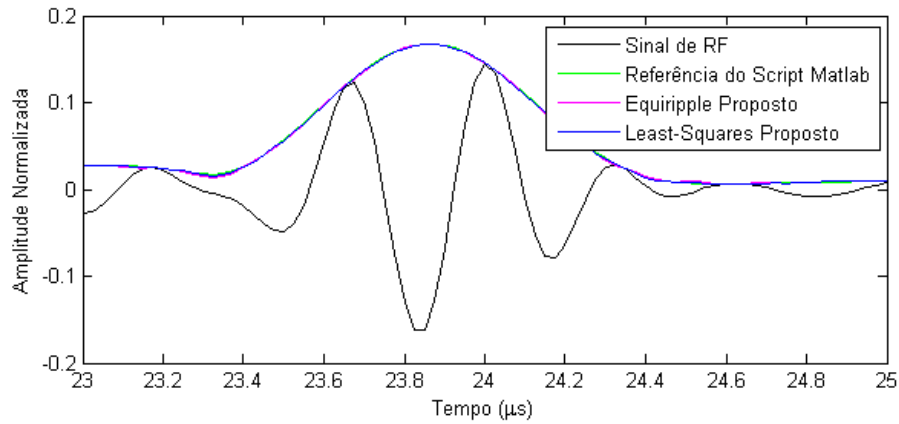
Para melhor avaliação desta etapa, na Figura 65 é apresentado um sinal de RF com amplitude normalizada e a comparação entre os resultados da detecção de envoltória através do *script* de referência do Matlab e dos métodos EQ e LS propostos. O retângulo marcado na cor vermelha indica a área que é ampliada na Figura 66, entre os tempos de 23 e 25 μs , para melhor visualização.

Figura 65 – Exemplo de comparação de resultados entre os métodos de detecção de envoltória utilizando uma aproximação da Transformada de Hilbert.



Fonte: Autoria própria.

Figura 66 – Ampliação da marcação indicada na Figura 65 para melhor visualização da comparação dos resultados de detecção de envoltória.

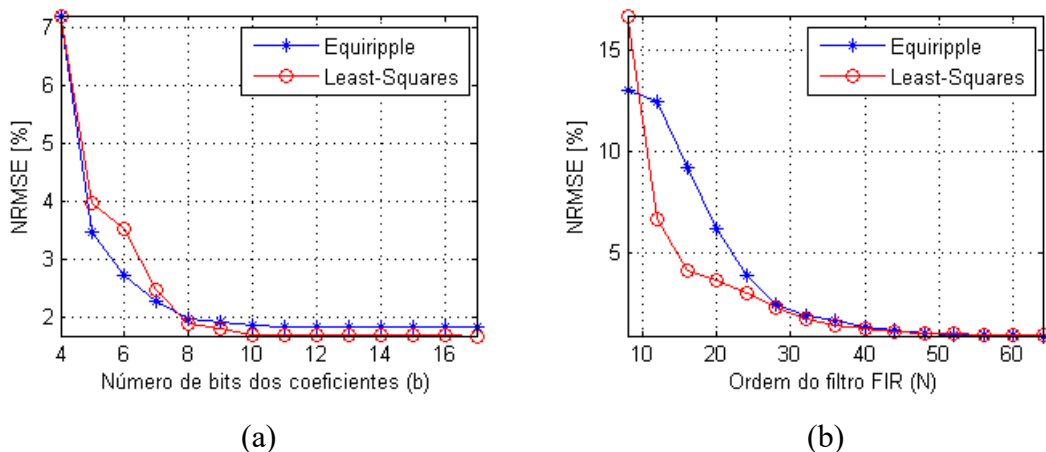


Fonte: Autoria própria.

Adicionalmente, a Figura 67 apresenta a avaliação do NRME entre b e da ordem (N) do filtro FIR utilizado na detecção de envoltória com os métodos EQ e LS. Na Figura 67(a) é feito um estudo quantitativo entre b variando de 4 a 17 e N igual a 32. Pode-se notar que os resultados convergem quando b é igual ou maior do que 8. Isso implica que podem ser utilizados menos *bits* no projeto do filtro, economizando custo computacional.

Considerando b igual a 17, na Figura 67(b) é analisado o comportamento do projeto dos filtros com N variando de 8 a 64. Percebe-se que os métodos convergem para N igual ou maior do que 32. Em ambos os casos, o NRMSE foi menor que 10% indicando excelente concordância entre os algoritmos de referência e os modelos implementados do Simulink.

Figura 67 – Comparação quantitativa da implementação do detector de envoltória baseado na aproximação da Transforma de Hilbert com filtro FIR para os métodos EQ e LS. (a) Avaliação de b variando de 4 a 17 com $N=32$. (b) Avaliação de N variando de 8 a 64 e $b=17$.

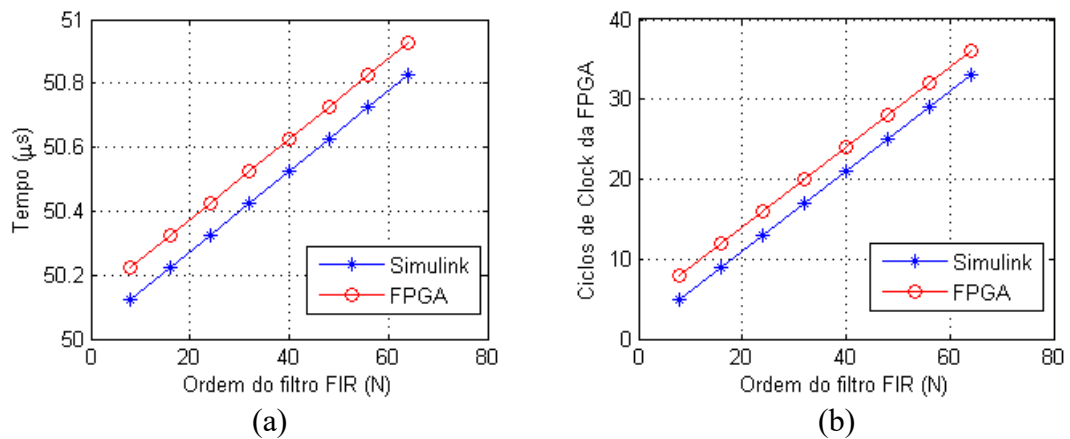


Fonte: Autoria própria.

Assim, arbitrou-se b igual a 17 e N igual a 32 para a utilização no modelo do detector de envoltória baseado na aproximação da Transformada de Hilbert com filtro FIR. Para esse caso, o NRMSE obtido foi de 1,68% e 1,83% para os métodos EQ e LS, respectivamente. Além disso, como no projeto da etapa de filtragem, escolheu-se o método de projeto de filtro EQ para a implementação final do detector de envoltória.

Complementarmente, a Figura 68 mostra a comparação entre o custo computacional da simulação no Simulink e experimental na placa Rx DE4-230. A Figura 68(a) apresenta a comparação gráfica entre o tempo de computação para execução da detecção de envoltória com N variando 8 a 64. Na Figura 68(b) essa comparação é feita entre os ciclos de *clock* da FPGA, considerando o *clock* do sistema igual a 40 MHz. Como pode ser visto, a latência da FPGA para gerar dados de envelope a cada ciclo de *clock* aumenta linearmente com N e pode ser calculada como $N/2+4$. Esse resultado apresenta apenas 3 ciclos a mais de *clock* do que é a implementação em *script* do Matlab e do trabalho de Chang *et al.* (2007), indicando uma economia de tempo para o processamento.

Figura 68 – Comparação de custo computacional simulado e experimental para N variando de 8 a 64. (a) Tempo de computação. (b) Latência em ciclos de *clock* da FPGA.



Fonte: Autoria própria

4.3 AVALIAÇÃO DAS ETAPAS DE PROCESSAMENTO

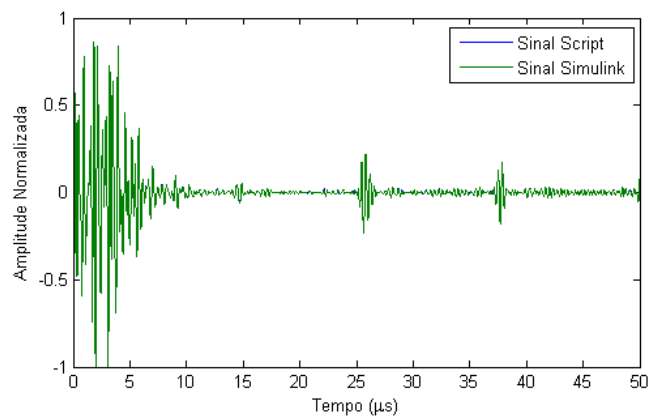
Nesta seção é apresentada a sequência dos resultados do processamento digital de sinais e imagens. Para facilitar o entendimento, arbitrou-se a *scanline* 58 com abertura ativa de 8 elementos para comparação dos resultados dos algoritmos de referência do Matlab com os

resultados dos sinais dos modelos no Simulink. Essa *scanline* foi escolhida pois apresenta claramente três alvos distintos nos tempos de 15 μs , 26 μs e 37 μs .

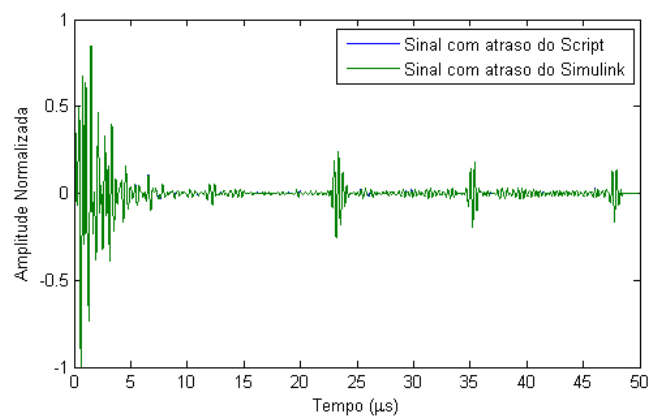
4.3.1 Etapa de Atraso Variável em Sinal Bruto de RF

Na Figura 69(a) é apresentado a primeira linha do sinal RF normalizado da *scanline* 58 com abertura de 8 elementos ativos sem pré-processamento. Na Figura 69(b) é mostrada a comparação gráfica do sinal após a aplicação do atraso variável na linha de RF entre o algoritmo do Matlab e o modelo do Simulink. Adicionalmente, na Figura 70 são apresentados os espectros em frequência dos sinais da Figura 69(b) através da FFT para validar a etapa de atraso variável. Percebe-se uma ótima concordância entre as respostas. Aqui, como nas outras análises, acredita-se que a diferença entre os sinais é causada pelo tipo das variáveis adotadas.

Figura 69 – Sinal de RF para comparação da aplicação do atraso variável entre o *script* do Matlab e o modelo do Simulink. (a) Sinais referentes à primeira linha da *scanline* 58 sem processamento. (b) Comparação dos sinais após a etapa de atraso variável.



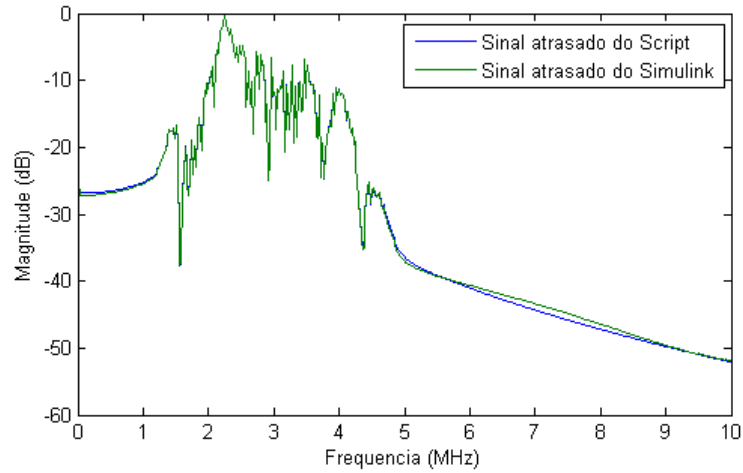
(a)



(b)

Fonte: Autoria própria.

Figura 70 – Comparação entre os espectros em frequência dos sinais gerados pelo *script* do Matlab e o modelo do Simulink após a etapa de atraso variável.

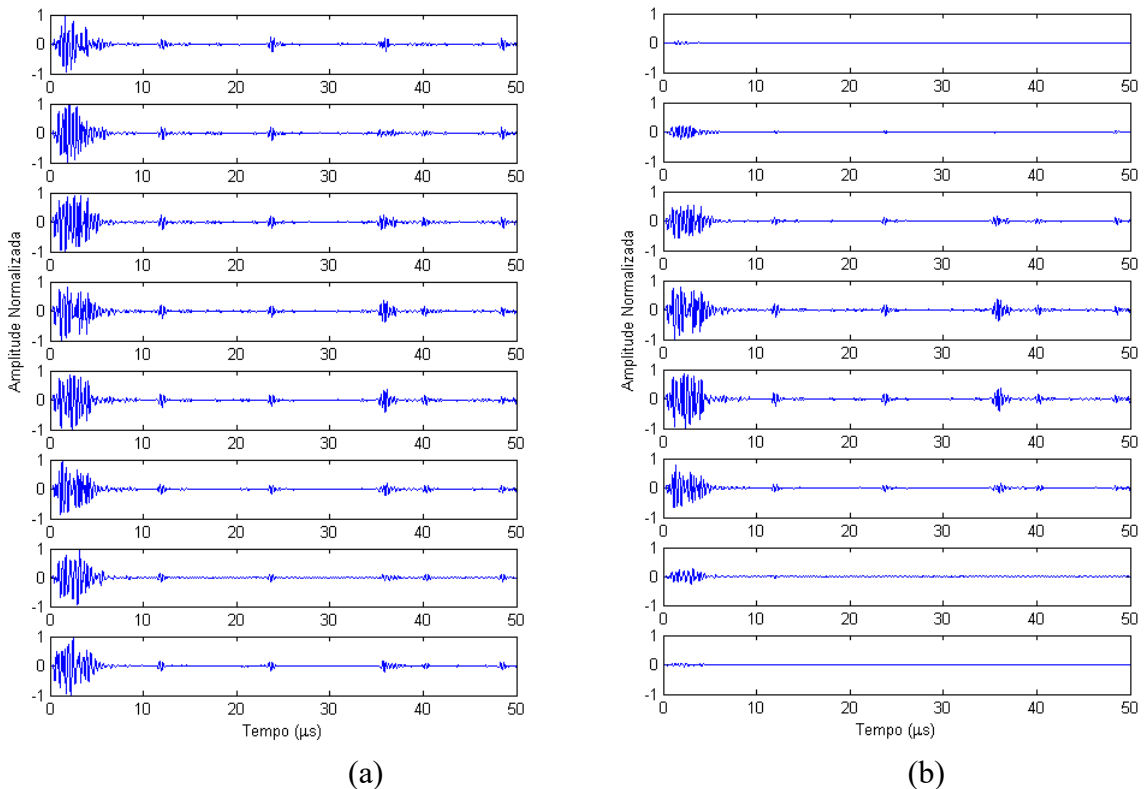


Fonte: Autoria própria.

4.3.2 Etapa de Apodização

Na Figura 71(a) e (b) são apresentados os sinais dos 8 canais da *scanline* 58 antes e após a etapa de apodização com janelamento Gaussiano, respectivamente.

Figura 71 – Resultados da apodização com janela Gaussiana. (a) Sinais antes e (b) sinais após a apodização.



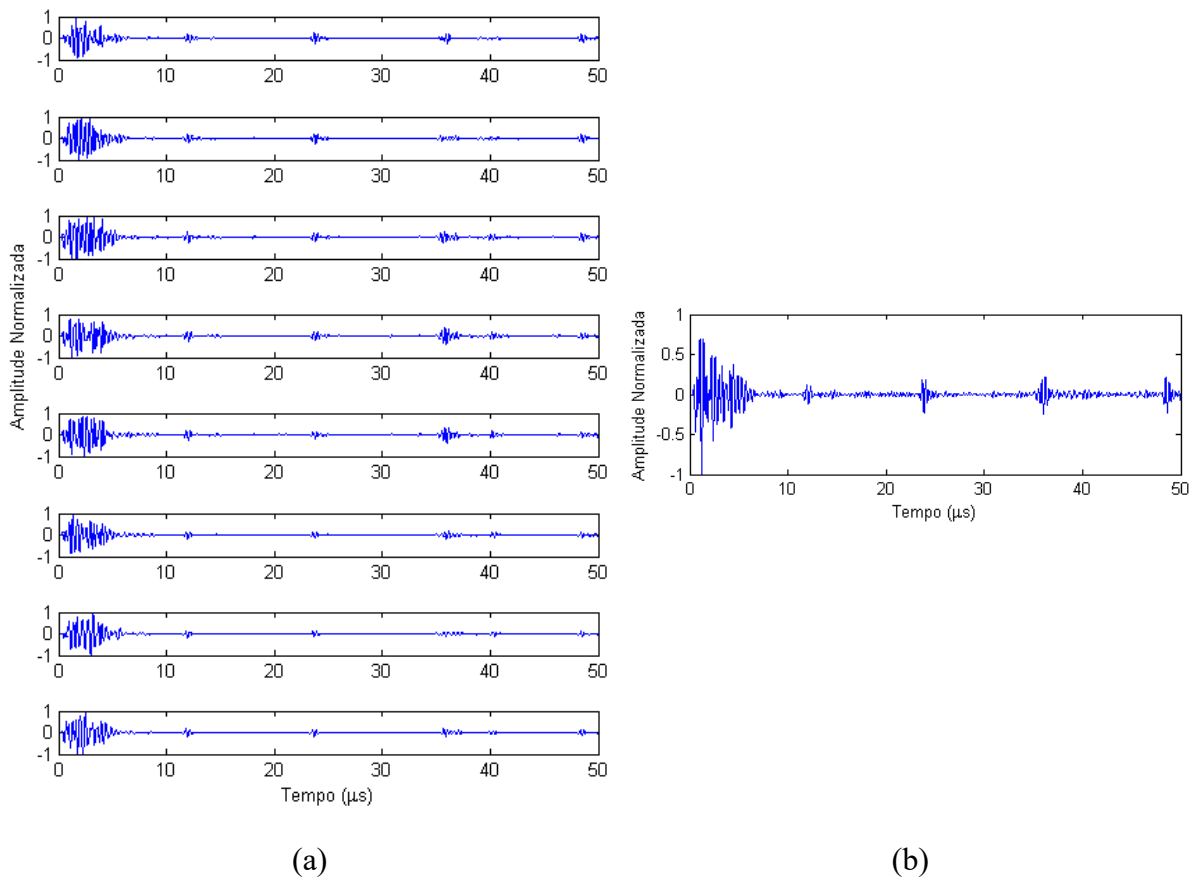
Fonte: Autoria própria.

A Figura 71 foi apresentada como um exemplo, pois o sistema possibilita a avaliação de diversas técnicas de janelamento. Percebe-se nitidamente que os canais dos extremos da abertura sofreram maior atenuação, que é uma característica da aplicação da técnica, conforme descrito na seção 2.4.3.

4.3.3 Etapa de Somatório Coerente

Na Figura 72 é apresentado um exemplo de resultados da etapa de somatório coerente da *scanline* 58. Na Figura 72(a) são mostrados os 8 canais após a etapa da apodização utilizando a janela Retangular. Nesse caso, os sinais são multiplicados por ganho unitário, isto é, não sofrem alteração. O resultado normalizado após a etapa de somatório coerente é ilustrado na Figura 72(b).

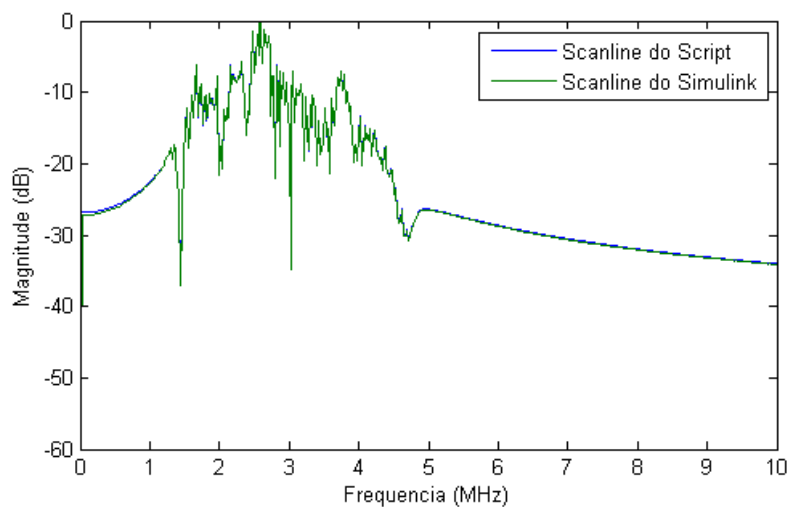
Figura 72 – Exemplo de resultados de somatório coerente. (a) Sinal de 8 canais pré-somatório com apodização Retangular. (b) Sinal normalizado após a etapa de somatório coerente.



Fonte: Autoria própria.

Na sequência é apresentado o espectro em frequência do sinal resultante da etapa de somatório coerente a fim de analisar a coerência entre o modelo do Simulink e o algoritmo de referência, representado na Figura 73. Percebe-se uma ótima concordância entre as respostas, em específico na banda passante do transdutor de US. A média do erro absoluto dos sinais é de 0,87% e o NRMSE é igual a 3,84%, implicando em um ótimo desempenho do modelo em análise.

Figura 73 – Comparação entre os espectros em frequência dos sinais gerados pelo *script* do Matlab e o modelo do Simulink após a etapa de somatório coerente.



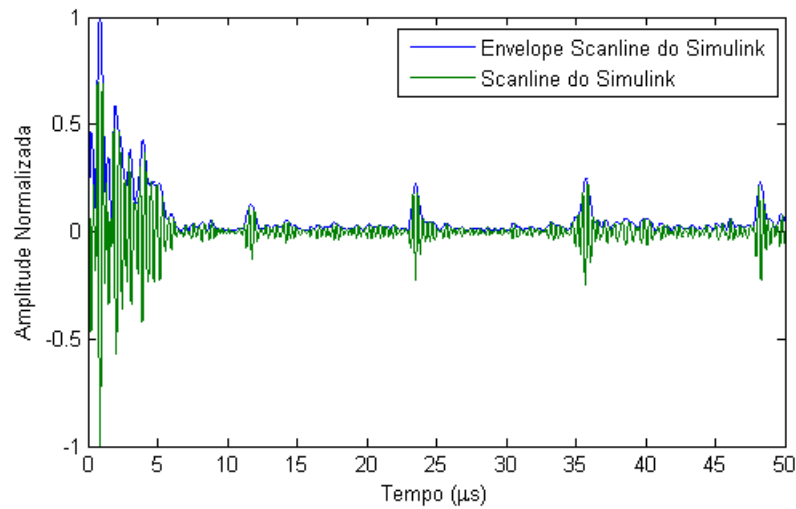
Fonte: Autoria própria.

4.3.4 Etapa de Demodulação e Detecção de Envoltória

Na Figura 74 são apresentadas a *scanline* 58 após o somatório coerente e a sua envoltória sobrepostas, ambas geradas através do modelo no Simulink. Pode-se notar que o processamento de detecção de envoltória foi realizado com sucesso.

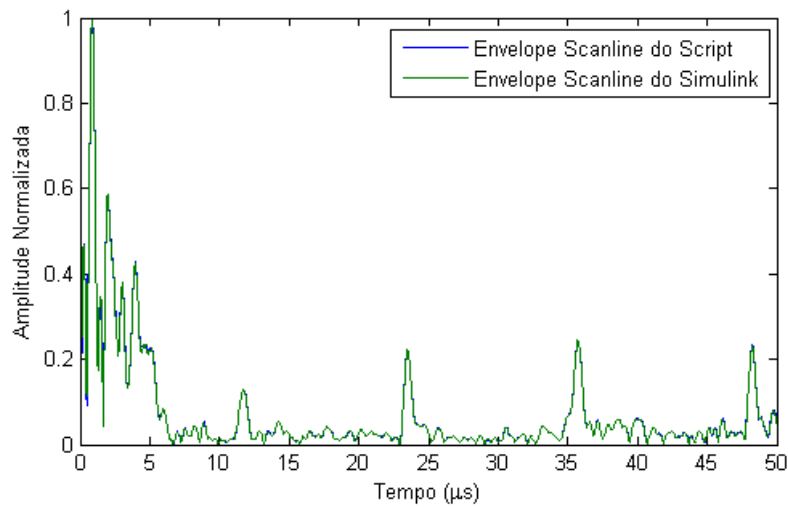
Para melhor visualização, na Figura 75 é representada a comparação entre a envoltória da *scanline* realizada pelo *script* do Matlab, através do comando de valor absoluto da Transformada de Hilbert do sinal, “*abs(hilbert())*”, e pelo modelo implementado no Simulink. Foi calculada a média do erro absoluto de 1,89% entre o sinal de referência e o resultado do Simulink. Também, obteve-se NRMSE de 3,04%, demonstrando um ótimo desempenho.

Figura 74 – Sinal de envelope gerado pelo modelo do Simulink.



Fonte: Autoria própria.

Figura 75 – Comparação entre os sinais de envelope gerados pelo *script* do Matlab e o modelo do Simulink.

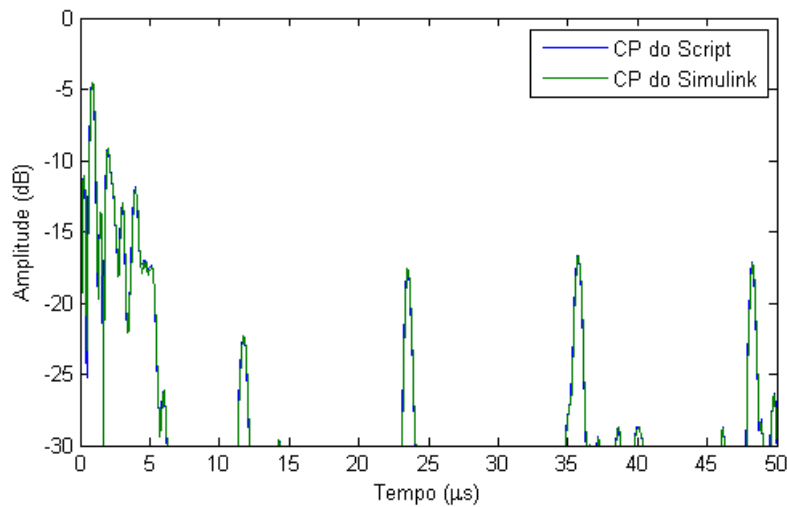


Fonte: Autoria própria.

4.3.5 Etapa de Compressão Logarítmica

Após a etapa de detecção de envoltória o sinal é processado pela etapa de compressão logarítmica (CP). Na Figura 76 é apresentada a comparação entre o resultado após a compressão logarítmica do *script* de referência e o modelo do Simulink.

Figura 76 – Comparação entre os sinais após a etapa de compressão logarítmica gerados pelo *script* do Matlab e modelo do Simulink.



Fonte: Autoria própria.

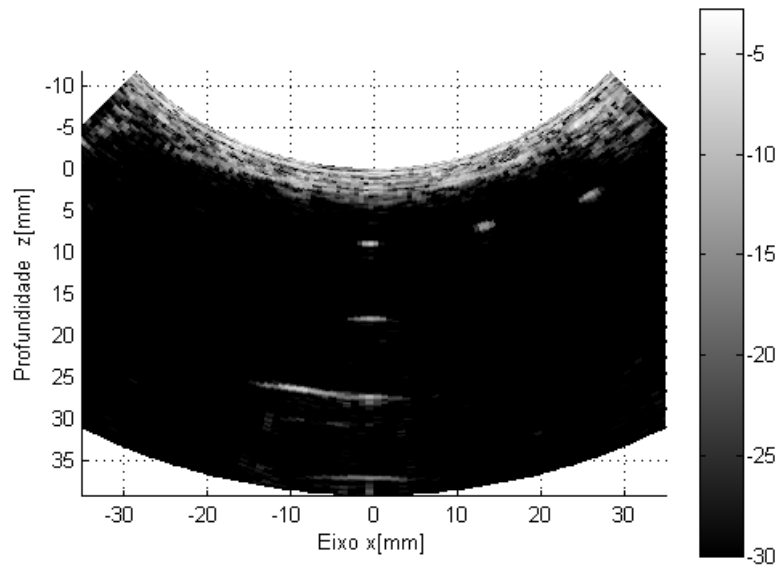
O NRMSE calculado para essa etapa foi de 8,99%, resultando em bom desempenho. A média do erro absoluto não foi calculada nesta etapa, pois apresenta uma quantidade significativa de elementos nulos, influenciando no resultado.

4.3.6 Etapa de Conversão de Varredura

Todas as etapas anteriores foram avaliadas no Simulink e implementadas na FPGA. A etapa de conversão de varredura foi a única implementada em *software*, através do Matlab, utilizando os dados pós-processados exportados da placa de desenvolvimento DE4-230 para o computador através da interface serial RS 232 emulada via USB. Na Figura 77 é ilustrada a imagem gerada em Modo B utilizando a técnica *beamforming* DAS com abertura ativa de 8 elementos, após a aplicação da conversão de varredura no Matlab.

Utilizando o *script* do Matlab como referência, foram comparadas todas as *scanlines* geradas por simulação pelo modelo do Simulink com os dados experimentais adquiridos a partir do processamento embarcado na FPGA. O NRMSE a partir do Simulink foi de 8,40%, enquanto para FPGA foi de 20,31%. No entanto, fazendo a mesma comparação, porém eliminando as primeiros 100 amostras das *scanlines*, o resultado do NRMSE para a FPGA diminui para 8,31%. Isso implica que a comparação é relevante apenas para os pontos de interesse, isto é, valores mais distantes da face do transdutor em aproximadamente 1,9 mm.

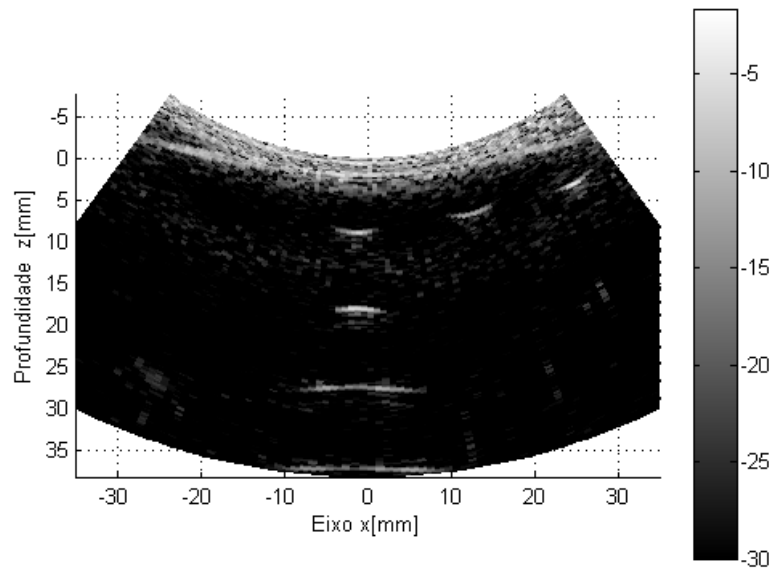
Figura 77 – Imagem final Modo B gerada pelo sistema de processamento digital de sinais embarcado em *hardware* pelo método DAS com abertura ativa de 8 elementos.



Fonte: Autoria própria.

A imagem Modo B utilizando a técnica *beamforming* DAS com abertura de 32 elementos ativos, após a aplicação da conversão de varredura no Matlab, é mostrada na Figura 78.

Figura 78 – Imagem final Modo B gerada pelo sistema de processamento digital de sinais embarcado em *hardware* pelo método DAS com abertura ativa de 32 elementos.



Fonte: Autoria própria.

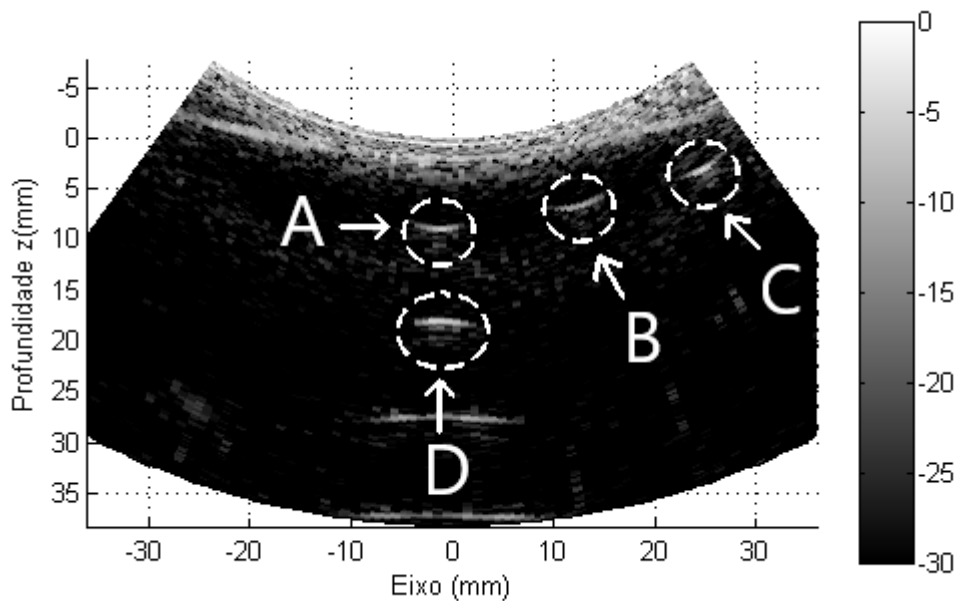
Pode-se observar que com uma abertura maior, a imagem em Modo B torna-se mais nítida e precisa em relação a identificação de alvos. Para este projeto também foram verificadas todas as *scanlines* e calculado o NRMSE dos dados obtidos através do Simulink e dos dados adquiridos da FPGA, em relação ao modelo utilizado no *script* do Matlab.

O NRMSE calculado para os dados simulados no Simulink foi de 10,16% e para os dados experimentais da FPGA foi obtido 21,25%. Utilizando a mesma comparação realizada no projeto com 8 elementos de abertura, ou seja, eliminando as 100 primeiras amostras das *scanlines*, obteve-se o resultado de 9,61%, indicando uma excelente concordância com o modelo de referência.

4.4 ANÁLISE DE CONTRASTE CNR E CR

Nesta seção são apresentadas as análises dos parâmetros CR e CNR das imagens geradas pelos modelos com abertura ativa de 8 e 32 elementos em comparação com os resultados do processamento de referência no Matlab, simulação do Simulink e dados experimentais do processamento realizado a partir da FPGA Rx. Foram selecionados 4 alvos para a análise, sendo divididos em A, B, C e D, conforme ilustrado na Figura 79.

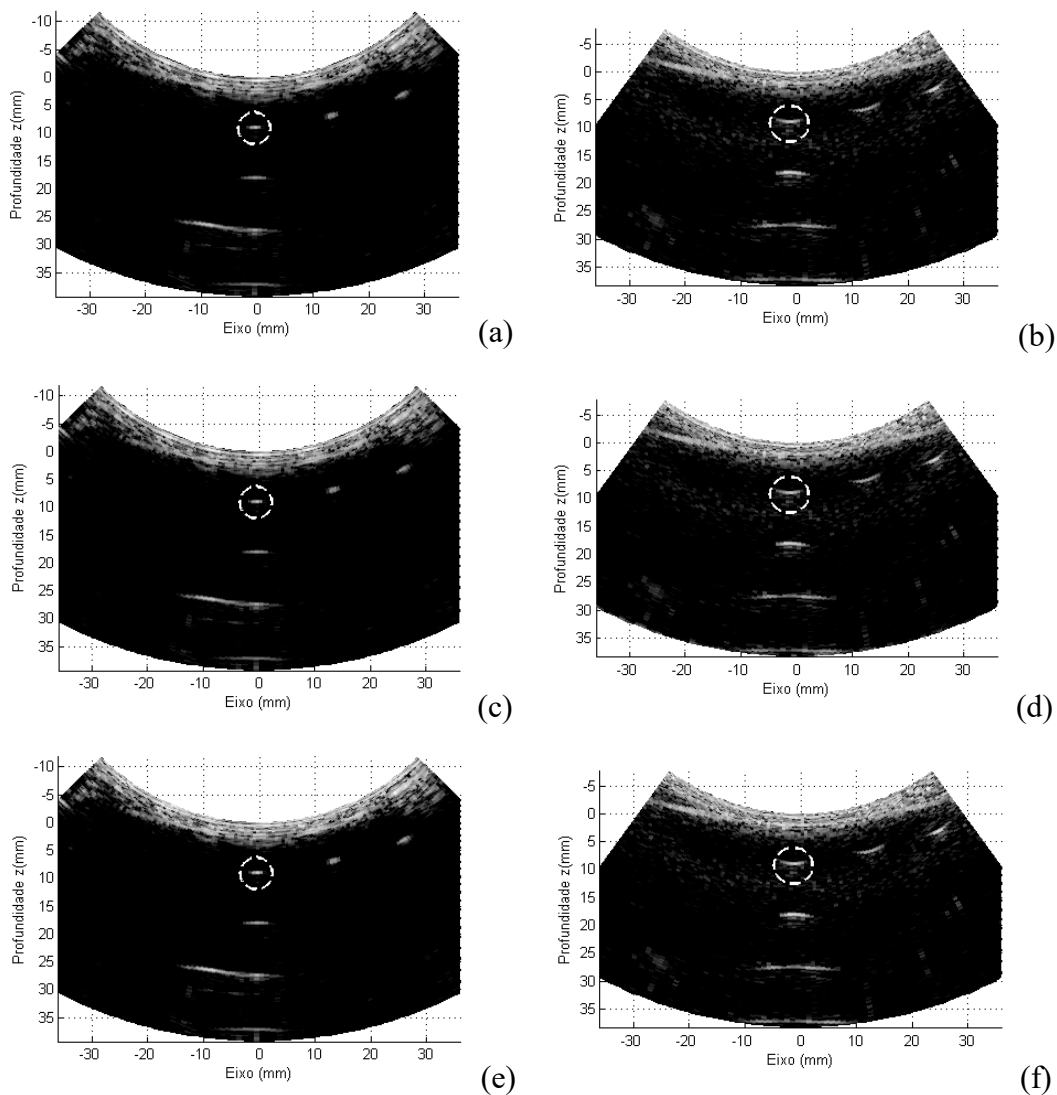
Figura 79 – Imagem exemplo de indicação dos alvos para as análises de contraste CR e CNR.



Fonte: Autoria própria.

Para realizar as análises de contraste através dos parâmetros CR e CNR foram comparadas as imagens de US Modo B geradas pelo método *beamforming* DAS com aberturas de 8 e 32 elementos ativos nas seguintes investigações: geração de imagem a partir dos algoritmos do *script* do Matlab, geração de imagem através dos dados do modelo do Simulink e geração de imagem resultante do processamento embarcado em FPGA. Essas análises foram baseadas nas imagens da Figura 80, nas quais o alvo A é indicado.

Figura 80 – Imagens de US Modo B para análise de contraste geradas por simulações a partir do *script* e modelo Simulink e de forma experimental por FPGA com abertura ativa de 8 elementos (a,c,e) e 32 elementos (b,d,f).



Fonte: Autoria própria.

Na Tabela 8 são apresentados os valores calculados do CR e CNR para os alvos A, B, C e D, bem como seus respectivos erros percentuais para as aberturas ativas de 8 e 32 elementos.

Tabela 8 – Comparação dos parâmetros de contraste CR e CNR para os alvos A, B, C e D.

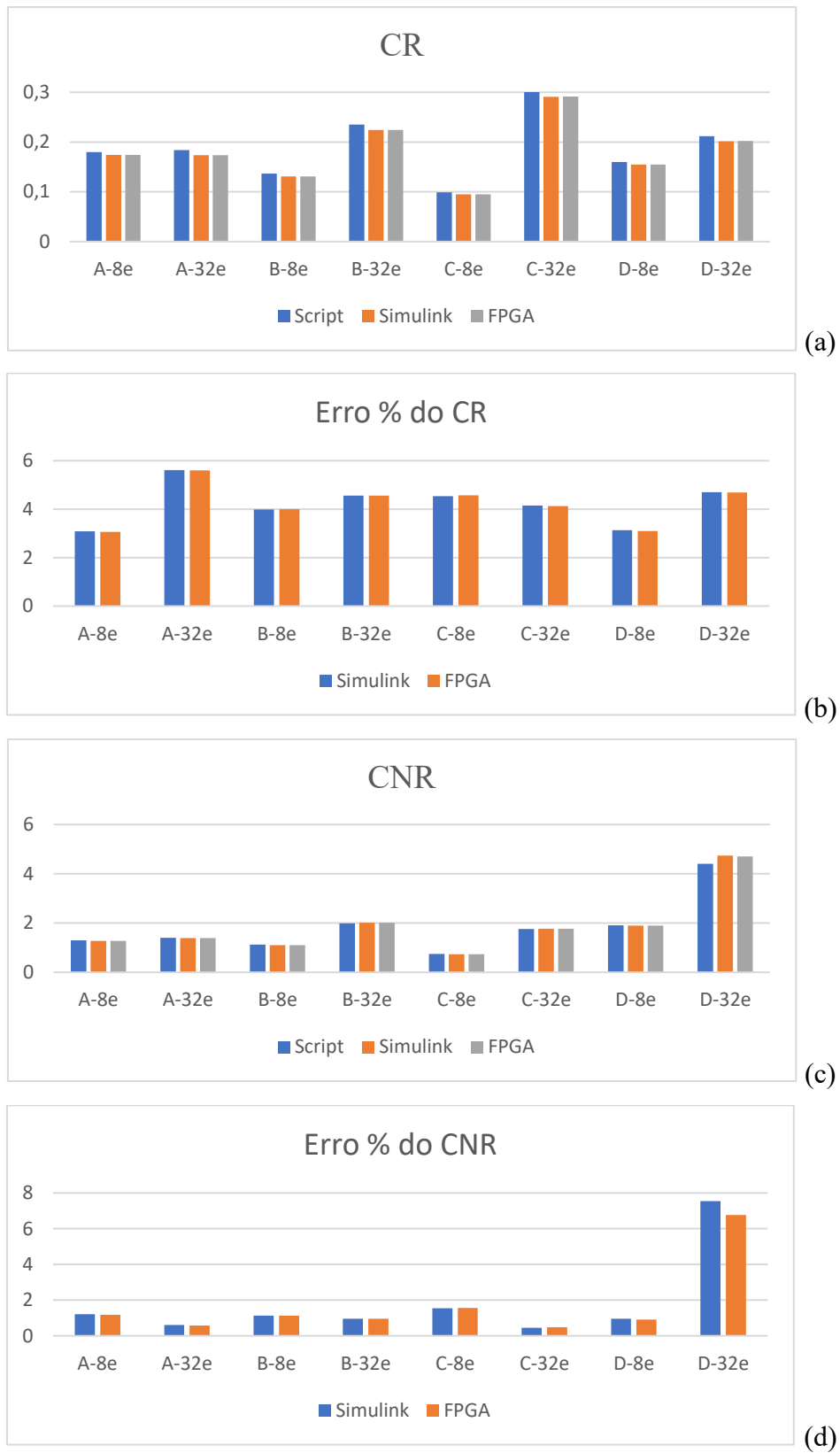
Alvo	Abertura	Procedimento	CR	% Erro	CNR	% Erro
A	8	<i>Script</i>	0,1799	-	1,2920	-
		Simulink	0,1743	3,086	1,2765	1,2024
		FPGA	0,1744	3,065	1,2767	1,1804
	32	<i>Script</i>	0,1841	-	1,3963	-
		Simulink	0,1738	5,6076	1,3878	0,6051
		FPGA	0,1738	5,5969	1,3882	0,5809
B	8	<i>Script</i>	0,1364	-	1,1198	-
		Simulink	0,1310	3,9875	1,1072	1,1255
		FPGA	0,1310	4,0041	1,1071	1,1367
	32	<i>Script</i>	0,2352	-	1,9898	-
		Simulink	0,2245	4,5561	2,0088	0,9548
		FPGA	0,2245	4,5543	2,0089	0,9609
C	8	<i>Script</i>	0,0993	-	0,7458	-
		Simulink	0,0948	4,5402	0,7343	1,5449
		FPGA	0,0948	4,5674	0,7342	1,5622
	32	<i>Script</i>	0,3038	-	1,7593	-
		Simulink	0,2912	4,1538	1,7672	0,4502
		FPGA	0,2913	4,1226	1,7678	0,4849
D	8	<i>Script</i>	0,1600	-	1,9097	-
		Simulink	0,1550	3,1299	1,8915	0,9537
		FPGA	0,1551	3,0997	1,8923	0,9157
	32	<i>Script</i>	0,2118	-	4,4078	-
		Simulink	0,2018	4,7024	4,7404	7,5450
		FPGA	0,2019	4,6910	4,7057	6,7575

Fonte: Autoria própria.

Adicionalmente, na Figura 81 são apresentados os resultados das análises de contraste dos parâmetros CR e CNR na forma de gráficos para melhor visualização. Nesses resultados, os alvos A, B, C e D para as aberturas de 8 e 32 elementos ativos foram nomeados como A-8e, A-32e, B-8e, B-32e, C-8e, C-32e, D-8e e D-32e, respectivamente.

As avaliações qualitativas da Figura 80 e quantitativas da Tabela 8 indicam ótimos resultados para os modelos gerados e implementados em *hardware*. Ambas as análises do CR e CNR indicam erro percentual muito baixo entre as imagens geradas por simulação e de forma experimental. Em todos os alvos, observa-se que o CR e CNR são maiores para as imagens com abertura de 32 elementos, indicando melhor contraste. Esse resultado era esperado, pois mais energia ultrassônica é transferida para o meio com o aumento da abertura.

Figura 81 – Resultados do contraste para os alvos A, B, C e D e aberturas ativas de 8 e 32 elementos. (a) Dados do CR. (b) Erro percentual do CR. (c) Dados do CNR. (d) Erro percentual do CNR.



Fonte: Autoria própria.

Ambos modelos com abertura de 8 e 32 elementos apresentaram resultados coerentes com a análise. No entanto, é notório que o modelo com 32 elementos de abertura apresenta erros de CR e CNR maiores, chegando a aproximadamente 7,5 % no alvo D. Esse comportamento pode ser explicado pela quantidade de informação que é processada em paralelo com abertura de 32 elementos com resolução de 17 *bits* da parte fracionária em comparação com a abertura de 8 elementos. Conseqüentemente, uma diferença é introduzida na etapa de somatório coerente em comparação com o algoritmo do Matlab que utiliza variáveis do tipo *double*, ou seja, 64 *bits*. Isso implica que a quantidade de canais a serem processados tornam a precisão do número de *bits* da etapa do processamento um fator impactante. Supõem-se que aumentando o número de *bits* fracionários os erros sejam mitigados, já que os mesmos influenciam diretamente na qualidade do sinal. Entretanto, a partir de 18 *bits* é necessário o dobro de elementos multiplicadores, elevando o custo do computacional *hardware*.

4.5 TEMPO DE PROCESSAMENTO E CUSTO COMPUTACIONAL

Neste trabalho, a placa DE2-115 foi utilizada para a transmissão dos dados brutos de US, adquiridos através da plataforma ULTRA-ORS, via interface Ethernet para a placa de alto desempenho DE4-230, na qual realizou-se processamento embarcado envolvendo as aberturas de 8 e 32 elementos ativos para geração de imagem por US. Para transmissão de dados foi configurado a módulo Ethernet no modo Gigabit (1000 Mbps).

Para o projeto envolvendo 8 elementos ativos de abertura, o tempo total de processamento, desde o início da transferência até o último dado processado na variável “*data*”, foi de 4 segundos. No projeto com abertura de 32 elementos, o tempo total decorrido foi de 9 segundos. O tempo contabilizado, não envolve a transferência via RS232 emulada por USB para o computador.

Para uma plataforma de pesquisa, houve uma significativa redução de tempo de transferência e processamento, passando de 30 minutos para menos de 10 segundos. Esses resultados ainda não são adequados para geração de imagens por US em tempo real, pois seria necessário, ao menos, gerar 25 imagens em um segundo (HEDRICK *et al.*, 1995), implicando em um tempo mínimo de 40 ms por imagem, comumente chamada de quadro. No entanto, os equipamentos que possibilitam a operação em tempo real, desprendem de mais recursos de *hardware* e *firmware* totalmente embarcado. Por exemplo, Boni *et al.* (2018) apresentaram um estudo acerca dos principais sistemas abertos de pesquisa na atualidade. Naquele estudo, foram comparados sistemas de 128 até 1024 canais de transmissão e recepção com capacidade de

geração de imagem em tempo real e as seguintes tecnologias de transferência de dados: USB 2.0, USB 3.0, PCI Express 3.0 e Ethernet. Tais equipamentos de pesquisa utilizam múltiplas FPGAs, DSPs e sistema em *chip* (SoCs - *System on Chip*) com capacidade de paralelismo, além de GPUs de última geração.

Por outro lado, neste trabalho existe a possibilidade de redução significativa do tempo de transferência dos dados entre as duas placas de FPGA. No futuro, poderão ser avaliadas técnicas de otimização baseadas no modo de operação *burst* com as memórias FIFO e operações de acesso direto à memória (*Direct Memory Access* - DMA) mais avançadas.

Na Tabela 9 é mostrado o custo computacional dos recursos de *hardware* para o projeto de transmissão de dados executado na placa DE2-115 e para os projetos de processamento com 8 e 32 elementos ativos de abertura compreendidos na placa DE4-230.

Tabela 9 – Comparação de custo computacional dos recursos de *hardware* entre as placas DE2-115 e DE-230.

Dispositivo	EP4CE115F29C7	EP4SGX230KF40C2	
Família	Cyclone IV E	Stratix IV	
Abertura	8/32	8	32
Utilização de elementos lógicos	13%	14%	19%
ALUT combinacional	-	15858/182400 (9%)	23662/182400(13%)
ALUT memória	-	335/91200 < (1%)	299/91200 < (1%)
Registros lógicos dedicados	10871/114480 (9%)	19577/182400 (11%)	25263/182400 (14%)
Total de registros	11002	20251	25901
Total de pinos	175/529 (33%)	250/888 (28%)	250/888 (28%)
Total de bits dos blocos de memória	2668345/3981312 (67%)	968992/14625792 (7%)	2167396/14625792 (15%)
Blocos DSP com elementos de 18 bits	-	166/1288 (13%)	598/1288 (46%)
Total de PLLS	2/4 (50%)	4/8 (50%)	4/8 (50%)
Total de DLLS (<i>Delay-Locked Loop</i>)	-	1/4 (25%)	1/4 (25%)

Fonte: Autoria própria.

Como o projeto para transmissão de dados não envolveu as etapas de processamento digital, não foram utilizados blocos DSP para a FPGA da placa DE2-115. Em contrapartida, para o processamento envolvendo 8 canais, foram utilizados 13% dos blocos de DSP com 17 bits de precisão. Já para o projeto com 32 canais, a utilização dos blocos sobe para 46%, mostrando que é ainda possível desenvolver um modelo utilizando mais canais no processamento. O modelo completo do sistema de processamento com abertura de 8 elementos

ativos é apresentado no Apêndice A em um folha tamanho A3. O modelo completo para abertura de 32 elementos não foi incluído devido às suas dimensões.

Em geral, a utilização de elementos lógicos ficou entre 13 e 19% e a utilização de pinos entre 28 e 33%, apresentando uma margem para projetos futuros envolvendo essa mesma vertente. Pode-se citar com um eventual problema o total de *bits* dos blocos de memória disponibilizados pela placa DE2-115, já que para o projeto de transmissão ocupou 67% da memória, sobrando menos liberdade para projetos futuros envolvendo transmissões mais robustas. Para tal, é necessária uma placa com maior capacidade de blocos de memória.

5 CONCLUSÃO

Pode-se concluir que o objetivo principal deste trabalho, que foi a implementação de um sistema de transferência de dados brutos de US via rede Ethernet para realização e avaliação das etapas de processamento digital de sinais usando a técnica DAS na geração de imagens de US em Modo B embarcado em FPGA, foi alcançado com sucesso.

Todas as funcionalidades definidas na especificação de requisitos foram atendidas e as etapas de processamento digital de sinais para reconstrução de imagem em Modo B foram implementadas, apresentando funcionalidade correta. Entre elas, realizou-se com sucesso a avaliação da resposta do filtro FIR com relação ao número de *bits* da parte fracionária e ordem dos coeficientes nas etapas de filtragem digital e do detector de envoltória baseado em uma aproximação da Transformada de Hilbert. Conclui-se então que:

- O número de elementos ativos utilizados para a abertura sintética impacta diretamente no custo computacional e no tempo decorrido tanto para transferência quanto para processamento dos dados;
- Os modelos desenvolvidos no Simulink com a biblioteca DSP Builder apresentaram ótimo desempenho e estão em concordância com os modelos de referência;
- A análise quantitativa mostrou que os maiores erros entre os modelos ocorrem para a abertura de 32 elementos. O máximo erro do CR e do CNR foram 5,61% e 7,55%, respectivamente, ambos para o modelo do Simulink;
- Apesar da diferença do tipo das variáveis, a comparação quantitativa entre o *script* do Matlab e os modelos do Simulink auxiliou na avaliação, verificação e validação dos blocos de processamento implementados com o DSP Builder;
- O número máximo de blocos DSP não atingiu 50% da capacidade da FPGA de processamento, indicando que novas técnicas podem ser adicionadas aos modelos;
- Mesmo não sendo possível a implementação de um sistema robusto o suficiente para geração de imagem em tempo real, houve um avanço significativo em termos de tempo de processamento, reduzindo de 30 minutos para 9 segundos.

5.1 TRABALHOS FUTUROS

Como sugestão para continuidade deste trabalho, pode-se citar:

- Implementação das etapas de Interpolação e Decimação no processamento de recepção para otimizar a focalização;

- Estudo e implementação da etapa de Conversão de Varredura em *hardware* pela placa Rx;
- Avaliar o modo *burst* dos blocos de memória FIFO do DSP Builder;
- Pesquisar um método mais eficiente para aquisição dos dados processados pela placa Rx;
- Estudar estratégias para melhorar o *handshake* entre as placas;
- Realizar comparações da imagem em Modo B utilizando a Apodização com outros tipos de Janelamento;
- Estudar alternativas para otimização de recursos utilizados pelos modelos implementados com o DSP Builder, de forma a melhorar o processamento;
- Explorar a possibilidade da utilização de mais canais, a fim de possibilitar projetos com maior número de elementos ativos de abertura;
- Avaliação de otimizações na implementação proposta e a avaliação de novas interfaces de transferência de dados, como, por exemplo, USB 3.0 e PCI Express.

REFERÊNCIAS

3COM. **Token Ring-in-Fast Ethernet Server Network Interface Card User Guide**. 1998. Disponível em: <http://www.3com.com>. Acesso em: 10 jul. 2020.

AGARWAL, A.; SCHNEIDER, F. K.; YANG MO YOO; BALUYOT, F.; YONGMIN KIM. Single-Chip Solution for Ultrasound Imaging Systems: Initial Results. **IEEE Ultrasonics Symposium**, p. 1563-1566, 2007.

ALI, M.; MAGEE, D.; DASGUPTA, U. **Signal Processing Overview of Ultrasound Systems for Medical Imaging**. Texas Instruments Incorporated, White Paper, 2008. Disponível em: <http://www.ti.com/lit/wp/sprab12/sprab12.pdf>. Acesso em: 10 jul. 2020.

ASEN, J. P. *et al.* Implementing Capon beamforming on the GPU for real time cardiac ultrasound imaging. **IEEE Ultrasonics Symposium**, 2012.

ASSEF, A. A. **Arquitetura de hardware multicanal reconfigurável com excitação multinível para desenvolvimento e testes de novos métodos de geração de imagens por ultrassom**. 2013. Tese (Doutorado em Engenharia Biomédica) - Universidade Tecnológica Federal do Paraná, 2013.

ASSEF, A. A. **Sinais de ultrassom para caracterização de meios biológicos e não biológicos**. 2006. Dissertação (Mestrado em Engenharia Biomédica) - Universidade Tecnológica Federal do Paraná, 2006.

ASSEF, A. A.; MAIA, J. M.; COSTA, E. T. Initial experiments of a 128-channel FPGA and PC-based ultrasound imaging system for teaching and research activities. **Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**, p. 5172–5175, 2016.

ASSEF, A. A.; MAIA, J. M.; OLIVEIRA, J.; FPGA Implementation and Evaluation of an Approximate Hilbert Transform-Based Envelope Detector for Ultrasound Imaging Using the DSP Builder Development Tool. **41st Annual International Conference of IEEE Engineering in Medicine and Biology Society (EMBC)**, p. 2813, 2019.

BERGÉ, J. M. *et al.* **VHDL designer's reference**. [S. l.]: Springer US, 1992.

BERTSEKAS, D.; GALLAGER, R. **Data Networks**. 2nd ed. [S. l]: Prentice Hall, 1992.

BLACK, P E. **Data structure. Dictionary of Algorithms and Data Structures**. U.S. National Institute of Standards and Technology, 2004.

BONI, E. *et al.* 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, 2012.

BONI, E. *et al.* Ultrasound Open Platforms for Next-Generation Imaging Technique Development. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 65, n. 7, p. 1078-1092, 2018.

BONI, E., *et al.* ULA-OP 256: A 256-channel open scanner for development and real-time implementation of new ultrasound methods. **IEEE transactions on ultrasonics, ferroelectrics, and frequency control**, v. 63, n. 10. p. 1488-1495, 2016.

BROADSOUND CORPORATION. **Convex array ultrasound transducer**. Disponível em: <https://www.medicaexpo.com/prod/broadsound-corporation/product-67912-420166.html>
Acesso em: 10 jul. 2020.

BUSHBERG, J. T.; BOONE, J. M. **The essential physics of medical imaging**. [S. l]: Lippincott Williams & Wilkins, 2011.

CÂMARA, D. J. **Plataforma de ultrassom programável dedicada à pesquisa**. 2015. Dissertação (Mestrado em Ciências) - Universidade de São Paulo, 2015.

CHANG, J. H.; YEN, J. T.; SHUNG, K. K. A novel envelope detector for high-frame rate, high-frequency ultrasound imaging. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 54, 2007.

CHANG-HONG HU *et al.* Design of a 64-channel Digital High Frequency Linear Array Ultrasound Imaging Beamformer on a Massively Parallel Processor Array. **IEEE Ultrasonics Symposium**, p. 1266-1269, 2008.

CHRISTENSEN, D. A. **Ultrasonic Bioinstrumentation**. [S. l]: New York: John Wiley & Sons, 1988. 235 p. ISBN 978-0471604969.

CINCOTTI, G. *et al.* Efficient transmit beamforming in pulse-echo ultrasonic imaging. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 46, p. 1450–1458, 1999.

CLARK, A. P. **Principles of Digital Data Transmission**. (2nd Edition). New York: Wiley, 1983. 300 p. ISBN 9780470274583.

COMER, D. E. **Computer Networks and Internets with Internet Application**. 3rd ed. [S. l.]: Prentice Hall, 2001.

COUADE, M. **The advent of ultrafast ultrasound in vascular imaging: a review**. Journal of Vascular Diagnostics and Interventions, 2016.

DA COSTA, C. **Projetos de circuitos digitais com FPGA**. São Paulo: Erica, 2014.

DEBRUNNER L. S.; WANG Y. Optimizing filter order and coefficient length in the design of high performance FIR filters for high throughput FPGA implementations. **IEEE 12th Digital Signal processing Workshop & 4th IEEE Signal Processing Education Workshop**. p. 608-612, 2006.

FELDKAMPER, H. *et al.* Low power delay calculation for digital beamforming in handheld ultrasound systems. **IEEE Ultrasonics Symposium**, 2000.

FERREIRA, B. M. **Modelagem e implementação de um sistema de processamento digital de sinais baseado em FPGA para geração de imagens por ultrassom usando o Simulink**. 2017. Dissertação (Mestrado em Sistemas de Energia) - Universidade Tecnológica Federal do Paraná, 2017.

FISH, P. **Physics and instrumentation of diagnostic medical ultrasounds**. New York: John Wiley & Sons, 199. 250 p.

FLUKE BIOMEDICAL. **84-317 Multi-Purpose Tissue / Cyst Ultrasound Phantom**. Disponível em: <https://www.flukebiomedical.com/products/radiation-measurement/phantoms-test-tools/84-317-multi-purpose-tissue-cyst-ultrasound-phantom>. Acesso em: 10 jul. 2020.

GAMMELMARK, K. L.; JENSEN, J. A. Multielement synthetic transmit aperture imaging using temporal encoding. **IEEE Transactions on medical imaging**, v. 22, n. 4, p. 552-563, 2003.

GOLIO, M. **The RF and Microwave Handbook (Second Edition)**. CRC Press. Jan, 2008. 2208 p. ISBN 978-0849372179

HASSAN, M. A.; YOUSSEF, A. M.; KADAH Y. M. Modular FPGA-based digital ultrasound beamforming, **1st Middle East Conference on Biomedical Engineering**, p. 134-137, 2011.

HAZARD, C. R.; LOCKWOOD, G. R. Developing a high speed beamformer using the TMS320C6201 digital signal processor. **IEEE Ultrasonics Symposium**, v. 2, p. 1775-1758, 2000.

HEDRICK, W. R.; HYKES, D. L.; STARCHMAN, D. E. **Ultrasound physics and instrumentation**. 3rd ed. [S. l.]: St. Louis; London: CV Mosby, 1995. ISBN 0323032125.

HUDA, S.; MALLICK, M.; ANDERSON, J. H. **Clock gating architectures for FPGA power reduction**. International Conference on Field Programmable Logic and Applications, 2009.

IBM. **Ethernet and Token Ring Configuration Guide**. 2001. Disponível em: <ftp://ftp.software.ibm.com/printers/manuals/npref/g5445240.pdf>. Acesso em: 10 jul. 2020.

IEEE. **IEEE Standard for Ethernet**. [S.l.: s.n.], 2012. 400 p. ISBN 9730738173122.

INGLE, V. K.; MANOLAKIS, D. G. **Applied Digital Signal Processing**. Cambridge University Press, 2011. ISBN 9780521110020.

INTEL. **Cyclone IV FPGAs**. Disponível em: <https://www.intel.com/content/www/us/en/products/programmable/fpga/cyclone-iv.html>. Acesso em: 10 jul. 2020a.

INTEL. **Stratix IV FPGAs**. Disponível em: <https://www.intel.com/content/www/us/en/products/programmable/fpga/stratix-iv.html>. Acesso em: 10 jul. 2020b.

JENSEN, J. A. *et al.* A program for simulation ultrasound systems. 10th Nordic-Baltic Conf. Biomed. **Imaging – Med. Bio. Eng. Comp.**, p. 351-353, 1996.

JENSEN, J. A. *et al.* **Synthetic aperture ultrasound imaging**. Ultrasonics, v. 44. SUPPL., 2006.

JENSEN, J. A. *et al.* Ultrasound research scanner for real-time aperture data acquisition. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 52, p. 881–891, 2005.

JENSEN, J. A.; 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, 1998.

JINBO, M. **Imagens ultrassônicas em modo B com técnicas de abertura focal sintética - SAFT**. 2007. Dissertação (Mestrado em Engenharia) - Universidade de São Paulo, 2007.

KANG, J. *et al.* A System-on-Chip Solution for Point-of-Care Ultrasound Imaging Systems: Architecture and ASIC Implementation. **IEEE Transactions on Biomedical Circuits and System**, v. 10, p. 412-423, 2016.

KUMAR, A. *et al.* An Improved Design Technique of Digital Finite Impulse Response Filter for Notch Filtering. In: Rawat B., Trivedi A., Manhas S., Karwal V. (eds) **Advances in Signal Processing and Communication. Lecture Notes in Electrical Engineering**, Springer, Singapore. v. 526, 2019.

KWON, K.; LEE, B. J.; SHIN, B. S. Reliable subsurface scattering for volume rendering in three-dimensional ultrasound imaging. **Computers in Biology and Medicine**, v. 117, p. 103608, 2020.

LEWANDOWSKI, M.; WALCZAK, M.; ZOLEK, N. Optimization of real-time ultrasound PCIe data streaming and OpenCL processing for SAFT imaging. **IEEE International Ultrasonics Symposium (IUS)**, 2013.

MACHADO, R. P. *et al.* **Metodologia para a inspeção instrumentalizada de isoladores de pino de redes de distribuição por RFI e ultrassom**. VI Congresso de Inovação Tecnológica em Energia Elétrica, 2011.

MAIA, J. M. **Apostila de Ultrassom**. 2004.

MATHWORKS. **DSP Builder for Intel FPGAs**. 2018. Disponível em: <https://www.mathworks.com/solutions/fpga-asic-soc-development.html>. Acesso em: 10 jul. 2020.

MAXFIELD, C. **The design warrior's guide to FPGA's: device, tools, and flows.** [S. l]: Boston: Newnes/Elsevier, 2004.

MCCLELLAN, J.; 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, 1973.

MCDICKEN, W. N. **Diagnostic ultrasonics: Principles and use of instruments.** [S. l]: Wiley, 1976. 320 p.

MENOTTI, R. **Implementação de um módulo Ethernet 10/100Mbps com interface Avalon para o processador Nios II da Altera.** 2005. Dissertação (Mestrado em Ciências da Computação e Matemática Computacional) - Universidade de São Paulo, 2005.

MONMASSON, E. *et al.* FPGAs in Industrial Control Applications. **IEEE Transactions on Industrial Informatics**, v. 7, p. 224-243, 2011.

MOROS, E. **Physics of Thermal Therapy: Fundamentals and Clinical Applications.** 1st ed. Imaging in medical Diagnosis and Therapy. [S. l]: CRC Press, 2012. 380 p. ISBN 9781439848906.

MULLER, R. S.; T. I. KAMINS T. I.; CHAN M., **Device Electronics for Integrated Circuits**, 3rd ed. New York, NY: John Wiley and Sons, Inc., 2003.

NATIONAL INSTRUMENTS. **Fundamentos da tecnologia FPGA - National Instruments.** 2013. Disponível em: <http://www.ni.com/white-paper/6983/pt/>. Acesso em: 10 jul. 2020.

NAUGLE, M. G. **The Illustrated Network Book.** [S. l]: VNR, 1994.

NGUYEN, T. Q.; VAIDYANATHAN, P. P. Eigenfilters: A New Approach to Least-Squares FIR Filter Design and Applications Including Nyquist Filters. **IEEE Transactions on Circuits and Systems**, v. 34, p. 976-978 n. 1, 1987.

NIKOLOV, S. I.; JENSEN, J. A.; TOMOV, B. G. Fast parametric beamformer for synthetic aperture imaging. **IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 55, p. 1755-1767, 2008.

OLIVEIRA, J.; ASSEF, A. A.; MAIA, J. M. Cascaded FPGA-based FIR filter for ultrasound imaging applications using the DSP Builder Tool. XII Simpósio de Engenharia Biomédica - IX

Simpósio de Instrumentação e Imagens Médicas, 2019, Uberlândia. **Anais do XII SEB e IX SIIM**, v. 4. p. 1-4, 2019.

OPPENHEIM, A. V. **Discrete-time signal processing**. Pearson Education India, 1999.

OTAKE, T. *et al.* **High-quality/High-resolution Digital Ultrasound Diagnostic Scanner**. Hitachi Review, v. 52, 2003.

PAYNE, P. A. Medical and industrial applications of high resolution ultrasound. **Journal of Physics E: Scientific Instruments**, v. 18, n. 6, p. 465-473, 1985.

PEI-JIE CAO *et al.*, Design of a real time digital beamformer for a 50 MHz annular array ultrasound transducer. **IEEE Ultrasonics Symposium**, v. 2, p. 1619-1622, out. 2002.

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

SANIIE J.; ORUKLU E. Introduction to the Special Issue on Novel Embedded Systems for Ultrasonic Imaging and Signal Processing. **Transactions on Ultrasonics, Ferroelectrics and Frequency Control**, v. 59, n. 7, p. 1329-1331, 2012.

SANTOS, F. D. dos. **Utilização de DSPs e FPGAs em unidades eletrônicas automotivas**. 2008. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - Universidade de São Paulo, 2008.

SCHNEIDER, F. K. *et al.* A fully programmable computing architecture for medical ultrasound machines. **IEEE Transactions on Information Technology in Biomedicine**, v. 14, n. 2, p. 538-540, 2010.

SHATTUCK, D. P. *et al.* Explososcan: A parallel processing technique for high speed ultrasound imaging with linear phased arrays. **The Journal of the Acoustical Society of America**, v. 75, p. 1273-1282, 1984.

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

SIKDAR, S. *et al.* **A Single Mediaprocesor - Based Programmable Ultrasound System.** v. 7, n. 1, p. 64-70, 2003.

SPURGEON, C. E. **Ethernet: The Definitive Guide.** [S. l]: O'Reilly, 2000.

SUETENS, P. **Fundamentals of Medical Imaging.** [S. l]: Cambridge University Press, 2009. 253 p.

SZABO, T. L. **Diagnostic Ultrasound Imaging.** 2nd Edition. [S. l]: Print Book & E-Book. ISBN 9780123964878, 9780123965424.

TER HAAR, G. **Therapeutic ultrasound.** European Journal of ultrasound, v. 9, n. 1, p. 3-9, 1999.

TERASIC, **Complex FPGA Design Service.** Disponível em: <https://www.intel.com/content/www/us/en/products/programmable/fpga/stratix-iv.html>. Acesso em: 10 jul. 2020.

THOMENIUS, K. E. Evolution of ultrasound beamformers. **IEEE Ultrasonics Symposium.** Proceedings, p. 1615-1622, 1996.

WELLS, P. N. T. **Ultrasonic imaging of the human body.** Reports on Progress in Physics, v. 62, p. 671-722, 1999.

WIKISONIX. **Ultrasound Image Computation.** 2013. Disponível em: <http://www.ultrasonix.com/wikisonix/index.php>. Acesso em: 10 jul. 2020.

WILSON, T. *et al.* The ultrasonix 500RP: A commercial ultrasound research interface. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 53, n. 10, p. 1772-1781, 2006.

YOUNG, S. R. **The effect of therapeutic ultrasound on the biological mechanisms involved in dermal repair.** Thesis (Physics Doctorate) - University of London, London, 1990.

ZHENG, Y. F.; ZHOU, H. An efficient quadrature demodulator for medical ultrasound imaging, **Frontiers of Information Technology & Electronic Engineering**, v. 16, n. 4, p. 301-310, 2015.

APÊNDICE A – PROJETO DO SISTEMA DE PROCESSAMENTO PARA GERAÇÃO DE IMAGEM POR ULTRASSOM NO SIMULINK

Projeto de processamento de dados da placa de desenvolvimento DE4-230 envolvendo uma abertura sintética de 8 elementos ativos.
 Autor: Jonathan de Oliveira

