

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**MATHEUS JOSE DA SILVA RUZYK**

**SISTEMA DE PROCESSAMENTO DIGITAL DE SINAIS ULTRASSÔNICOS  
BASEADO EM SYSTEM ON A CHIP (SOC)**

**CURITIBA**

**2023**

**MATHEUS JOSE DA SILVA RUZYK**

**SISTEMA DE PROCESSAMENTO DIGITAL DE SINAIS ULTRASSÔNICOS  
BASEADO EM SYSTEM ON A CHIP (SOC)**

**Digital ultrasound signal processing system based on a system on a chip  
(SoC)**

Trabalho de Dissertação apresentado como requisito para obtenção do título de Mestre em Engenharia Biomédica do Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI) da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Amauri Amorin Assef

**CURITIBA**

**2023**



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



**Ministério da Educação**  
**Universidade Tecnológica Federal do Paraná**  
**Campus Curitiba**



MATHEUS JOSE DA SILVA RUZYK

**SISTEMA DE PROCESSAMENTO DIGITAL DE SINAIS ULTRASSÔNICOS BASEADO EM SYSTEM ON A  
CHIP (SOC)**

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciências da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Engenharia Biomédica.

Data de aprovação: 14 de Dezembro de 2023

Dr. Amauri Amorin Assef, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Joaquim Miguel Maia, Doutorado - Universidade Tecnológica Federal do Paraná

Dr. Nivaldo Theodoro Schiefler Junior, Doutorado - Instituto Federal Santa Catarina - Ifsc

Documento gerado pelo Sistema Acadêmico da UTFPR a partir dos dados da Ata de Defesa em 14/12/2023.

## AGRADECIMENTOS

De forma geral agradeço a todos os que contribuíram para a realização deste trabalho, seja com alguma ajuda direta ou mesmo com uma palavra de encorajamento.

Agradeço ao Prof. Dr. Amauri Amorin Assef pela maestria com que me conduziu por esse percurso de mestrado e por sua paciência e compreensão sempre presentes em nossas interações e durante sua orientação.

Aos meus amados pais, Edson Ruzyk e Elizabete Pereira da Silva, que proporcionaram todas as condições necessárias para investir todo meu tempo e esforço em minha formação, permitindo-me chegar à esta enorme conquista.

À minha querida Nicole Polityto Cremasco pelos valiosos conselhos, por entender meus momentos de foco exclusivo no mestrado e, principalmente, por ajudar a organizar meus artigos e entregas importantes.

Gostaria de deixar registrado o meu reconhecimento à minha família e amigos, pois sempre me deram suporte e foram compreensíveis com minhas ausências para os estudos, desde os tempos de adolescência. Assim agradeço à minha madrinha Lourdes Pereira e tia Selma A. P. Gomes, aos amigos Giovanni Cruzara, Cesar A. S. Ribas e a todos os que fizeram parte dessa importante fase da minha vida.

Agradeço à AMD Xilinx por fornecer as ferramentas de desenvolvimento de projeto disponibilizadas através do *Xilinx University Program*, sem as quais este trabalho não seria possível.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

## RESUMO

O ultrassom (US) é uma técnica amplamente difundida e com aplicações tanto na medicina diagnóstica como na indústria. Suas capacidades para analisar meios de forma não invasiva, sem características ionizantes, a tornam adequada para exames de imagem. A pesquisa na área explora novas técnicas de utilização dos sinais de radiofrequência em US, o que requer acesso e manipulação de sinais e etapas intermediárias na cadeia de processamento, geralmente não acessíveis em plataformas comerciais. Plataformas abertas impulsionam avanços na pesquisa em US. O advento da tecnologia de *System on a Chip* (SoC) acelerou o desenvolvimento da área, permitindo sistemas abertos com características de maleabilidade do *beamforming* de transmissão, acesso aos dados do ADC e imageamento em tempo real, em software ou hardware, ou em SoC, que associam os dois. O objetivo deste trabalho foi implementar um sistema aberto para processar os dados de *back-end* de US em uma plataforma SoC da AMD Xilinx, a ZedBoard, realizando o *beamforming* de recepção e a reconstrução de imagens em Modo B, até a conversão de varredura. As etapas foram projetadas utilizando blocos do Model Composer no Simulink, implementadas no Vivado e Vitis, e validadas utilizando dados de US reais adquiridos de um *phantom*. Resultados de etapas intermediárias foram comparados com simulações do MATLAB/Simulink usando as métricas *Normalized Root Mean Square Error* (NRMSE) e *Normalized Residual Sum of Squares* (NRSS) para comprovar concordância entre os modelos. Já os resultados de imagem do sistema implementado empiricamente indicam que ele é adequado para atuar como plataforma de processamento de sinais de US, com erros quantitativos de 7,14% de *Contrast-to-Noise Ratio* (CNR) e 17,52% em *Contrast Resolution* (CR), levando 0,36 s para exibir resultados em um display VGA e consumindo menos de 50% dos recursos do SoC. Apesar de apresentar uma pequena degradação do CR em comparação com a imagem de referência, as imagens reconstruídas conseguiram mostrar características importantes da imagem. Além disso, a ocupação de recursos do dispositivo indica que novas etapas podem ser implementadas e otimizadas, incluindo os tempos de processamento que já diminuiram em relação a trabalhos anteriores. Como exemplo de possível aplicação do sistema implementado, tem-se o desenvolvimento de futuros sistemas abertos com processamento embarcado em SoC para atividades de pesquisas inovadoras do US.

**Palavras-chave:** ultrassonografia; soc; conversão de varredura; processamento digital de sinais; plataformas abertas de ultrassom.

## ABSTRACT

Ultrasound (US) imaging is a widely spread technique, spanning applications from diagnostic medicine to the general industry. Its non-invasive and non-ionizing characteristics make it suitable for various medical exams. Research in this field often explores novel radiofrequency US techniques, which require full access to intermediate signals and the capacity to make changes in processes that are commonly not accessible in commercial US platforms. Open platforms emerge as a solution, facilitating research in the US domain. The proliferation of the System on a Chip (SoC) technology has accelerated advancements in this field, enabling the creation of open systems with malleable characteristics in transmission beamforming, access to Analog to Digital Converter (ADC) raw data, and real-time imaging, either in hardware, software or in hybrid platforms SoCs. This work aimed to implement an open system to process back-end US data in a SoC platform from AMD Xilinx, specifically the ZedBoard, utilizing the reception beamformer and B-mode image reconstruction with scan conversion. Each processing step was modeled with Model Composer blocks in Simulink, implemented utilizing Vivado and Vitis software, and validated with empirical US data acquired from a commercial phantom. The results of intermediate steps were compared with MATLAB/Simulink simulations, utilizing Normalized Root Mean Square Error (NRMSE) and Normalized Residual Sum of Squares (NRSS) metrics to validate the agreement between models. Image results of the system's empirical implementation indicate that it serves as a suitable US processing platform with quantitative errors of 7.14% for Contrast-to-Noise Ratio (CNR) and 17.52% for Contrast Resolution (CR), and with a processing time of 0.36 seconds to display the resulting image on a VGA screen, utilizing less than 50% of the available SoC resources. Although the results show a minor degradation in the contrast resolution compared to the reference, they suggest that the resulting images retain essential information. Moreover, the device's resource utilization indicates potential optimization for additional processing steps, including changes to enhance processing time, which is already faster than previous works. Possible applications of this work are in the further development of future US research open systems to enable innovative research in the ultrasound area.

**Keywords:** ultrasound; soc; scan conversion; digital signal processing; ultrasound open platforms.

## LISTA DE FIGURAS

Figura 1 – Comportamento de ondas em interfaces entre meios com diferentes impedâncias acústicas. . . . .	24
Figura 2 – Comparação entre focos de abertura em transdutor eletrônico. . . . .	25
Figura 3 – Processo de <i>beamforming</i> de recepção. . . . .	26
Figura 4 – Processamento digital em Modo B. . . . .	27
Figura 5 – Estrutura de um filtro FIR. . . . .	28
Figura 6 – Exemplos de perfis de apodização. . . . .	28
Figura 7 – Detector de envoltória através da Transformada de Hilbert. . . . .	29
Figura 8 – Conversão de varredura. . . . .	30
Figura 9 – Diagrama do fluxo de dados entre módulos para as simulações e implementação. . . . .	33
Figura 10 – Transdutor matricial convexo AT3C52B. . . . .	35
Figura 11 – Ensaio de aquisição com o ULTRA-ORS utilizando transdutor AT3C52B e <i>phantom</i> . . . . .	36
Figura 12 – Plataforma ZedBoard Zynq-7000 ARM/FPGA SoC. . . . .	37
Figura 13 – Principais subsistemas e periféricos disponíveis na placa ZedBoard. . . . .	37
Figura 14 – Etapas de projeto. . . . .	38
Figura 15 – Visão geral do projeto em Simulink com a ferramenta Model Composer. . . . .	39
Figura 16 – Projeto em Simulink com Model Composer do <i>beamformer</i> digital e seus subsistemas. . . . .	39
Figura 17 – Ilustração dos parâmetros de projeto do filtro FIR. . . . .	40
Figura 18 – Resposta em frequência da magnitude (cor azul) e fase (cor vermelha) do filtro FIR passa-baixa. . . . .	41
Figura 19 – Resposta ao impulso do filtro FIR passa-baixa. . . . .	41
Figura 20 – Implementação do filtro FIR passa-baixa com blocos do Model Composer. . . . .	42
Figura 21 – Implementação dos 8 canais de filtro passa-baixa com blocos do Model Composer. . . . .	43
Figura 22 – Projeto em Model Composer da etapa de atraso variável. . . . .	43
Figura 23 – Perfil de apodização com janelamento Hanning com exclusão dos valores nulos nas extremidades. . . . .	44

Figura 24 – Projeto em Model Composer da etapa de apodização. . . . .	44
Figura 25 – Projeto em Model Composer da etapa de somatório coerente. . . . .	45
Figura 26 – Projeto em Model Composer mostrando os subsistemas do detector de envoltória. . . . .	45
Figura 27 – Resposta em frequência da Transformada de Hilbert em magnitude (cor azul) e fase (cor vermelho). . . . .	46
Figura 28 – Resposta ao impulso da Transformada de Hilbert. . . . .	46
Figura 29 – Projeto em Model Composer da Transformada de Hilbert aproximada utilizando estrutura de filtro FIR. . . . .	46
Figura 30 – Projeto em Model Composer do detector de envoltória. . . . .	47
Figura 31 – Projeto em Model Composer da etapa de compressão logarítmica. . . . .	47
Figura 32 – Exemplo de transação do tipo <i>stream</i> através do barramento AXI4. . . . .	48
Figura 33 – Projeto em Model Composer da interface com o barramento AXI4 atra- vés de memórias FIFO. . . . .	48
Figura 34 – Bloco de controle do sinal de esvaziamento da memória FIFO de entrada.	48
Figura 35 – Projeto em Model Composer do controlador de memórias FIFO parale- las e sinais de controle internos. . . . .	49
Figura 36 – Projeto em Model Composer do validador de escrita nas memórias FIFO paralelas. . . . .	50
Figura 37 – Bloco IP do <i>Beamformer</i> exportado para o Vivado. . . . .	51
Figura 38 – Arquitetura e barramentos do sistema implementado. . . . .	52
Figura 39 – Fotografia do sistema exibindo resultados em tela e com descrição de componentes. . . . .	54
Figura 40 – Comparação entre sinal de ultrassom antes (cor azul) e após (cor ver- melha) o filtro FIR passa-baixa utilizando o Model Composer no Simulink.	55
Figura 41 – Comparação entre a simulação de sinal de ultrassom de 8 canais an- tes (lado esquerdo) e após (lado direito) a etapa de atraso variável e apodização utilizando o Model Composer. . . . .	57
Figura 42 – Sinal resultante após o somatório coerente utilizando o Model Composer.	58
Figura 43 – Comparação de envoltórias centrais geradas por demodulador em MA- TLAB, Simulink e Model Composer a partir de dados de RF gerados no Field II. . . . .	59



<b>Figura 44 – Ampliação da comparação de envoltórias com dados de RF gerados no Field II. . . . .</b>	<b>59</b>
<b>Figura 45 – Comparação de envoltórias simuladas e alimentadas com dados empíricos em: (a) MATLAB e Model Composer, e (b) sua ampliação entre 22 e 25,5 <math>\mu</math>s. . . . .</b>	<b>60</b>
<b>Figura 46 – Comparação entre imagens reconstruídas em Modo B com compressão logarítmica a partir de dados simulados no Field II em: (a) <i>script</i> do MATLAB, (b) simulação em Simulink e (c) simulação com Model Composer. Os retângulos nas cores amarelo e verde representam a região de interesse e a região de fundo, respectivamente, para as análises de contraste. . . . .</b>	<b>62</b>
<b>Figura 47 – Comparação de envoltórias com faixa dinâmica de -40 dB entre simulações em MATLAB e Model Composer com dados empíricos do ULTRA-ORS. . . . .</b>	<b>63</b>
<b>Figura 48 – Comparação entre imagens reconstruídas em Modo B com compressão logarítmica e coordenadas polares a partir de dados gerados pelo ULTRA-ORS e processados em: (a) <i>script</i> do MATLAB e (b) simulação do Model Composer. Os retângulos nas cores amarelo e vermelho representam a região de interesse e a região de fundo, respectivamente, para as análises de contraste. . . . .</b>	<b>64</b>
<b>Figura 49 – Imagem simulada resultante da etapa de conversão de varredura em MATLAB para dados processados em Model Composer. Quatro regiões de interesse são exibidas com retângulos na cor amarelo e a região de fundo na cor vermelho para as análises de contraste. . . . .</b>	<b>65</b>
<b>Figura 50 – Comparação de envoltórias com faixa dinâmica de -40 dB entre a simulação em MATLAB e a implementação experimental na ZedBoard. . . . .</b>	<b>66</b>
<b>Figura 51 – Imagem resultante da conversão de varredura realizada na plataforma ZedBoard utilizando dados simulados com Model Composer. . . . .</b>	<b>68</b>
<b>Figura 52 – Imagens resultantes da conversão de varredura em: (a) Simulação completa em MATLAB. (b) Simulação no Model Composer com conversão de varredura no MATLAB. (c) Resultados da ZedBoard. . . . .</b>	<b>69</b>
<b>Figura 53 – Gráfico de uso de recursos do SoC Zynq-7000. . . . .</b>	<b>71</b>

## LISTA DE TABELAS

Tabela 1 – Parâmetros de propagação de sinais de ultrassom em diferentes meios, à 1 MHz. . . . .	24
Tabela 2 – Parâmetros para aquisição de dados de RF avaliados no trabalho. . . .	35
Tabela 3 – Parâmetros de configuração do Model Composer. . . . .	38
Tabela 4 – Parâmetros do filtro FIR passa-baixa. . . . .	40
Tabela 5 – Resultados das métricas de comparação para simulação de detector de envoltória gerada com dados do Field II . . . . .	60
Tabela 6 – Métricas NRMSE e NRSS para envoltórias geradas a partir de dados empíricos do ULTRA-ORS sem compressão logarítmica. . . . .	61
Tabela 7 – Resultados das métricas de contraste para imagens geradas a partir de sinais de RF no Field II e processados no MATLAB, Simulink e Model Composer. . . . .	62
Tabela 8 – Métricas NRSS e NRMSE comparando envoltórias com compressão logarítmica geradas em MATLAB e Model Composer a partir de dados empíricos do ULTRA-ORS. . . . .	63
Tabela 9 – Resultados das métricas de contraste para imagens com compressão logarítmica geradas em simulações do MATLAB e Model Composer a partir de dados empíricos capturados com transdutor convexo (sem conversão de varredura). . . . .	64
Tabela 10 – Resultados das métricas de contraste CR e CNR de imagens com conversão de varredura em MATLAB para dados provindos do <i>script</i> de referência e simulação em Model Composer. . . . .	65
Tabela 11 – Métricas de comparação entre envoltórias com compressão logarítmica geradas pelo <i>beamformer</i> implementado na ZedBoard e por <i>script</i> do MATLAB antes da conversão de varredura. . . . .	67
Tabela 12 – Resultados das métricas de contraste para a etapa isolada de conversão de varredura, comparando implementação na ZedBoard com o <i>script</i> do MATLAB. . . . .	67
Tabela 13 – Resultados das métricas de contraste para a imagem com conversão de varredura gerada na plataforma ZedBoard. . . . .	70

<b>Tabela 14 – Utilização de recursos pós implementação no SoPC. . . . .</b>	<b>71</b>
<b>Tabela 15 – Resultados de tempo de processamento na plataforma ZedBoard. . . .</b>	<b>72</b>

## LISTA DE QUADROS

<b>Quadro 1 – Especificação de requisitos do sistema proposto. . . . .</b>	<b>32</b>
<b>Quadro 2 – Parâmetros de simulação com Field II para etapa de detecção de en- voltória. . . . .</b>	<b>34</b>
<b>Quadro 3 – Configurações e comentários sobre os IPs utilizados no Vivado. . . .</b>	<b>83</b>

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

A	Modo Amplitude
ADC	<i>Analog to Digital Converter</i>
B	Modo Brilho
CNR	<i>Contrast-to-Noise Ratio</i>
CORDIC	<i>Coordinate Rotation Digital Computer</i>
CR	<i>Contrast Resolution</i>
DAS	<i>Delay-and-Sum</i>
DDR	<i>Double Data Rate</i>
DMA	<i>Direct Memory Access</i>
DSP	<i>Digital Signal Processor</i>
DWT	<i>Discrete Wavelet Transform</i>
END	Ensaio Não Destrutivo
FDATool	<i>Filter Design and Analysis Tool</i>
FIFO	<i>First-In First-Out</i>
FIR	<i>Finite Impulse Response</i>
FPB	Filtro passa-baixa
FPGA	<i>Field Programmable Gate Array</i>
FPS	<i>Frames per second</i>
FWHDR	<i>Full Width at Half Dynamic Range</i>
FWHM	<i>Full Width at Half Maximum</i>
GPIO	<i>General Purpose Input/Output</i>
HT	Transformada de Hilbert

HW	Hardware
IP	Propriedade Intelectual
LP	Lógica Programável
LUS	Laboratório de Ultrassom e Instrumentação Biomédica
LUT	<i>Lookup Table</i>
MATLAB	MATrix LABoratory
MC	<i>Model Composer</i>
Np	Neper
NRMSE	<i>Normalized Root Mean Square Error</i>
NRSS	<i>Normalized Residual Sum of Squares</i>
PC	Computador Pessoal
PS	Sistema de Processamento
RAM	<i>Random Access Memory</i>
RF	Radiofrequência
RX	Recepção
SNR	<i>Signal to Noise Ratio</i>
SO	Sistema Operacional
SoC	<i>System on a Chip</i>
SoPC	<i>System on a Programmable Chip</i>
SSD	<i>Solid-State Drive</i>
SW	Software
TGC	<i>Time Gain Compensation</i>
TX	Transmissão
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
ULTRA-ORS	<i>Ultrasound Open Research System</i>
US	Ultrassom

UTFPR Universidade Tecnológica Federal do Paraná

VGA *Video Graphics Array*

## LISTA DE SÍMBOLOS

### Notações

$A_{pass}$	Peso da banda de passagem
$A_{stop}$	Peso da banda de parada
$\alpha$	Coeficiente de atenuação da onda
$b_i$	Coeficiente de filtro FIR
$c$	Velocidade de propagação da onda no meio
$D$	Espessura dos elementos
$dens$	Fator de densidade do filtro
$E(n)$	Envoltória do sinal
$f$	Frequência da onda propagada
$f_c$	Ponto de foco
$F_s$	Frequência de amostragem
$F_{pass}$	Frequência da banda de passagem
$F_{stop}$	Frequência da banda de parada
$FR_{max}$	Taxa de atualização máxima em quadros por segundo
$ht$	Envoltória analisada
$\overline{ht}$	Valor médio das envoltórias analisadas
$I$	Intensidade da onda resultante
$I_0$	Intensidade da onda emitida
$\rho$	Densidade do meio
$Z$	Impedância característica do meio genérico
$Z_1$	Impedância característica do meio 1
$Z_2$	Impedância característica do meio 2
$R$	Fração da onda refletida em interface entre meios diferentes
$i$	Índice de amostra
$I(n)$	Sinal em fase
$kerf$	Kerf
$\lambda$	Comprimento de onda
$\mu_t$	Intensidade média na região de interesse
$\mu_b$	Intensidade média na região de fundo
$n$	Número de linhas de varredura ou abertura
$N$	Número de amostras por canal
$N$	Ordem do filtro FIR
$Ne$	Número de elementos do transdutor
$Q(n)$	Sinal em quadratura
$r$	Raio em coordenadas polares



$r$	Distância percorrida pela onda
$R_{convex}$	Raio de curvatura
$\theta$	Ângulo em coordenadas polares
$x$	Coordenada cartesiana horizontal
$x(n)$	Sinal de ultrassom
$x_{db}(n)$	Sinal de ultrassom com compressão logarítmica
$x_{max}$	Valor máximo mapeado na escala logarítmica
$y$	Coordenada cartesiana vertical
$y(n)$	Sinal de ultrassom resultante

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>18</b>
<b>1.1</b>	<b>Objetivos</b>	<b>21</b>
1.1.1	Objetivos específicos	21
<b>1.2</b>	<b>Estrutura do trabalho</b>	<b>22</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>23</b>
<b>2.1</b>	<b>Fundamentos para obtenção de sinais de ultrassom em modo pulso-eco</b>	<b>23</b>
<b>2.2</b>	<b><i>Beamforming</i> do tipo <i>Delay and Sum</i> para transdutores multielementos</b>	<b>25</b>
<b>2.3</b>	<b>Formação de imagens em Modo B</b>	<b>26</b>
2.3.1	Filtragem digital	27
2.3.2	Apodização	28
2.3.3	Detecção de envoltória	29
2.3.4	Compressão logarítmica e ajuste de faixa dinâmica	29
2.3.5	Conversão de varredura	30
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>32</b>
<b>3.1</b>	<b>Especificação de requisitos</b>	<b>32</b>
<b>3.2</b>	<b>Softwares de projeto e simulação</b>	<b>33</b>
3.2.1	Dados de radiofrequência simulados com Field II	34
<b>3.3</b>	<b>Dados de radiofrequência do sistema ULTRA-ORS</b>	<b>34</b>
<b>3.4</b>	<b>Plataforma ZedBoard para processamento de sinais de ultrassom</b>	<b>36</b>
<b>3.5</b>	<b>Projeto do sistema de processamento de dados de ultrassom</b>	<b>38</b>
3.5.1	Projeto em Model Composer do Filtro Passa-Baixa	40
3.5.2	Projeto em Model Composer da etapa de atraso variável	43
3.5.3	Projeto em Model Composer da etapa de apodização	44
3.5.4	Projeto em Model Composer da etapa de somatório coerente	44
3.5.5	Projeto em Model Composer da etapa de detecção de envoltória	45
3.5.6	Projeto em Model Composer da etapa de compressão logarítmica	47
3.5.7	Projeto em Model Composer do bloco para adequação de sinais ao barramento AXI4	47
3.5.8	Projeto em Model Composer do bloco de controle de sinais e memórias FIFO paralelas do projeto de ultrassom	49

3.5.9	Projeto de hardware no Vivado . . . . .	51
3.5.10	Software do sistema de ultrassom e conversor de varredura . . . . .	52
<b>3.6</b>	<b>Métricas quantitativas para comparação de imagens e sinais de ultrassom</b>	<b>53</b>
<b>3.7</b>	<b>Outros elementos do sistema implementado . . . . .</b>	<b>53</b>
<b>4</b>	<b>RESULTADOS E DISCUSSÃO . . . . .</b>	<b>55</b>
<b>4.1</b>	<b>Resultados simulados das etapas de processamento . . . . .</b>	<b>55</b>
4.1.1	Simulação de sinais do filtro passa-baixa FIR . . . . .	55
4.1.2	Simulação de sinais apodizados e com atraso variável . . . . .	56
4.1.3	Simulação do sinal resultante do somatório coerente . . . . .	57
4.1.4	Simulação de sinais da detecção de envoltória e teste do demodulador com Field II . . . . .	58
4.1.5	Simulação de sinais da Compressão logarítmica . . . . .	61
4.1.6	Simulação da Conversão de varredura . . . . .	64
<b>4.2</b>	<b>Resultados empíricos pós implementação em SoC . . . . .</b>	<b>65</b>
4.2.1	Comparação entre envoltórias com compressão logarítmica . . . . .	66
4.2.2	Resultados isolados da conversão de varredura na ZedBoard . . . . .	67
4.2.3	Resultado da implementação total na ZedBoard e comparação com simulações	68
<b>4.3</b>	<b>Recursos utilizados na implementação em SoPC . . . . .</b>	<b>71</b>
4.3.1	Tempos de processamento da ZedBoard . . . . .	72
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>75</b>
<b>5.1</b>	<b>Trabalhos futuros . . . . .</b>	<b>75</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>77</b>
	<b>APÊNDICE A PROJETO DO SISTEMA DE ULTRASSOM NO VIVADO .</b>	<b>83</b>
	<b>APÊNDICE B PUBLICAÇÕES REFERENTES AO TRABALHO DE MES- TRADO . . . . .</b>	<b>86</b>

## 1 INTRODUÇÃO

O Ultrassom (US) é uma técnica de imageamento não invasiva e não destrutiva que tem muitas aplicações na medicina e indústria. A técnica é baseada em ondas sonoras que possuem frequências superiores às audíveis, ou seja, acima de 20 kHz, que são exploradas na área médica para detectar as características dos tecidos biológicos, geralmente em interfaces com diferentes impedâncias acústicas, onde ocorrem reflexões da onda ultrassônica. Como o tamanho das estruturas observadas através da técnica são da ordem de milímetros, as frequências usuais para imageamento médico são na faixa de mega-Hertz (MHz), o que permite obter uma resolução espacial adequada ao diagnóstico clínico (HEDRICK; HYKES; STARCHMAN, 2005).

As ondas de US são produzidas por transdutores piezoelétricos a partir de circuitos eletrônicos responsáveis pelo controle das etapas de Transmissão (TX), onde o pulso ultrassônico é emitido, e Recepção (RX) dos sinais de eco de Radiofrequência (RF). Os sinais de pulso de US são comumente gerados e recebidos pelo mesmo transdutor, forma de aquisição conhecida como pulso-eco, sendo os dois principais modos de imageamento para aplicações médicas o Modo Amplitude (A) e o Modo Brilho (B) (SZABO, 2004; SHUNG, 2005).

A técnica de tratamento e reconstrução dos sinais de eco de US é conhecida como *beamforming*. A partir da excitação e recepção ultrassônica utilizando transdutores matriciais, as etapas de filtragem, atraso de focalização, apodização e somatório coerente são tradicionalmente realizadas através da técnica de processamento conhecida como *Delay-and-Sum* (DAS) (SHUNG, 2005; ASSEF, 2013). Esse método é necessário para gerar os sinais a serem processados digitalmente através das etapas de detecção de envoltória, seguida da compressão logarítmica, para posterior exibição em Modo A, ou geração de imagens para várias aquisições, resultando no Modo B. Existem também outras técnicas para geração de imagem por US, como o uso de ondas planas com composição coerente, elastografia e variações do Modo Doppler (TANTER *et al.*, 2002; HEDRICK; HYKES; STARCHMAN, 2005; NEVES, 2020).

As aplicações do US para o auxílio ao diagnóstico médico são diversas. Um dos principais focos de aplicações do US é a especialidade de obstetrícia, sendo essa técnica também adequada para observação de estruturas internas do corpo humano sem causar danos aos tecidos, isto é, dentro dos limites de segurança de aplicação (WEBSTER, 2010). Como outros exemplos de aplicações na medicina, pode-se citar: medidas sinus paranasais, detecção de fluídos intracranianos, avaliação de propriedades dos fluídos, avaliação vascular como de diâmetro, avaliação da porosidade de ossos, avaliação de características fisiológicas de partes do corpo humano, como medições na bexiga e análise de movimentos, neuromodulação, adaptação do transdutor para ser embutido em cápsulas para análise interna no corpo, *wearables* ou roupas para comunicação com implantes, entre outros. Já na indústria, a técnica é usada principalmente em aplicações de qualidade e controle de estruturas e elementos mecânicos através do método pulso-eco em Ensaios Não Destrutivos (END) (JONVEAUX *et al.*, 2022).

Em um cenário de tecnologias de diagnóstico avançadas, como os *scanners* de ressonância magnética, a busca de novas formas de se utilizar o US se justificam, principalmente por ser um exame relativamente barato e não invasivo. Soma-se a isso que sistemas portáteis de US são necessários em regiões de difícil acesso ou que não tenham a infraestrutura necessária, como no caso de desastres naturais ou em regiões pouco desenvolvidas. Dessa forma, melhorias nessa técnica de auxílio ao diagnóstico podem representar um grande avanço para os sistemas de saúde. Em casos de emergência, em que os sistemas de saúde se encontram esgotados, esforços pelo uso de técnicas de diagnóstico, como o US, tornam-se necessárias, o que também motivou diversas pesquisas na área durante a pandemia de COVID-19 (MENTO *et al.*, 2020; OLIVEIRA *et al.*, 2020).

A pesquisa na área de US se beneficia de forma direta com a idealização de novos equipamentos de arquitetura aberta, que permitem investigação de novas técnicas e visualização de resultados intermediários. Sistemas de US comerciais são vendidos em sua maioria em plataformas fechadas, de modo que o usuário tem acesso a apenas sinais delimitados e pouca flexibilidade em técnicas de processamento de sinais e reconstrução de imagens. Tal fato exige que na pesquisa em US se desenvolvam plataformas abertas e modulares, que permitam modificações nas etapas de processamento, obtenção de sinais intermediários e inclusive o teste de novos algoritmos (BHARATH *et al.*, 2018).

Uma solução para o desenvolvimento de novas plataformas são os dispositivos reconfiguráveis *Field Programmable Gate Array* (FPGA), que apresentam como vantagem sua facilidade em configuração, confiabilidade e precisão programável. Entretanto, para a maior otimização e abrangência de possibilidades, o uso de sistemas processados e dispositivos *Digital Signal Processor* (DSP), em conjunto com a Lógica Programável (LP) de componentes reconfiguráveis, também se faz necessário. Os avanços atuais na tecnologia permitem que se junte todos estes atributos de Hardware (HW) em apenas um chip, também chamado de *System on a Chip* (SoC), ou *System on a Programmable Chip* (SoPC). No mercado há dois fabricantes principais, a AMD Xilinx e a Intel, as quais possuem soluções SoPC de diversas configurações, além de kits de desenvolvimento em conjunto com empresas parceiras, para facilitar a prototipagem e integração de sistemas (XILINX, 2023; INTEL, 2023).

Existem diversos trabalhos que têm explorado arquiteturas abertas de US para implementação de tendências atuais. Novos sistemas abertos são discutidos por Boni *et al.* (2018), que analisaram o histórico e as tendências na área médica, definindo três fatores chave para essas plataformas: possibilidade de modificação do *beamforming* de TX, ou seja, da forma e tipo de impulso emitido pelo transdutor; acesso aos dados brutos adquiridos pelo módulo de RX; e a implementação de imageamento em tempo real, sendo todos esses processos feitos em Software (SW) ou HW, além da tendência atual de juntar ambos em sistemas de processamento com FPGA integrada.

Jonveaux *et al.* (2022) discutiram diversas plataformas abertas e suas tendências atuais, além de identificar que a disponibilidade de SoPCs teve um efeito positivo na disrupção de

sistemas de US, ou seja, há uma tendência de avanços devido ao uso dessa nova tecnologia. Uma das plataformas abertas que se enquadra nos requisitos dessa categoria é a *Ultrasound Advanced Open Platform* (ULA-OP), desenvolvida por pesquisadores da Universidade de Florença (Itália), que possui recursos computacionais abrangentes, incluindo 4 FPGAs de elevado desempenho e um DSP *multicore*, capaz de processar simultaneamente até 256 canais de TX/RX ultrassônica (BONI *et al.*, 2015). Outros projetos buscam centralizar o processamento em plataformas menores, como mostrado por Hassan e Kadah (2013), no qual um sistema implementado em uma plataforma AMD Xilinx com FPGA Virtex-5 realiza todo o processamento de *beamforming* até a geração de imagem em Modo B. Outro exemplo foi apresentado por Govindan *et al.* (2015), através do sistema RPUTS (*Reconfigurable and Programmable Ultrasonic Testing System*), que usa a placa de testes ZedBoard (DIGILENT, EUA) com FPGA AMD Xilinx para controle de módulos *front-end* de aquisição e processamento de dados de RF em 3 dimensões.

Com foco na pesquisa de múltiplas implementações de algoritmos de US, Wang *et al.* (2021) apresentaram um sistema aberto que permite migração de projeto entre FPGA, placa de vídeo e processadores a partir do uso de várias arquiteturas para US em tempo real. Por outro lado, a popularização de métodos de aprendizado de máquinas também tem incentivado pesquisas que tentam extrair características de imagens de US, que podem ter algoritmos implementados em plataformas configuráveis (JIANG; VIRUPAKSHAPPA; ORUKLU, 2017; MICUCCI; IULA, 2022).

Com o propósito de implementar um sistema aberto para atividades de pesquisa do US, o Grupo do Laboratório de Ultrassom e Instrumentação Biomédica (LUS) da Universidade Tecnológica Federal do Paraná (UTFPR) vem desenvolvendo trabalhos relacionados ao processamento de dados de RF embarcado em plataformas reconfiguráveis. Inicialmente, Assef (2013) implementou o sistema *Ultrasound Open Research System* (ULTRA-ORS), que realiza todo o processamento *front-end* de TX e RX de dados de US de maneira aberta, reprogramável e modular com 128 canais. Trabalhos posteriores do grupo exploraram estruturas do tipo *Finite Impulse Response* (FIR) para a implementação de filtros digitais e detector de envoltória com processamento *back-end* para reconstrução de imagens em Modo B embarcados em duas placas de desenvolvimento: plataforma Terasic DE4-230 com a FPGA Stratix IV; e placa Terasic DE2-115 com a FPGA Cyclone IV, ambas com processador *soft core* (FERREIRA, 2017; OLIVEIRA, 2020). Esses trabalhos ainda necessitavam da etapa final de processamento em um Computador Pessoal (PC) com a plataforma MATrix LABoratory (MATLAB) para realização da etapa final de conversão de varredura (*scan conversion*), ou troca de coordenadas para correta exibição da imagem adquirida com transdutor convexo.

Neste trabalho, buscou-se contribuir com as pesquisas do LUS através da implementação e avaliação do *beamforming* de RX, incluindo como diferencial todas as etapas de processamento digital de sinais até a conversão de varredura para reconstrução de imagem em Modo B, totalmente embarcado em um dispositivo SoPC.

## 1.1 Objetivos

O objetivo principal deste trabalho foi implementar e avaliar um sistema de processamento digital de sinais de ultrassom através do *beamforming* DAS para reconstrução de imagem em Modo B em plataforma reconfigurável SoPC, incluindo as etapas de filtragem digital, apodização, ajuste de atraso de focalização, somatório coerente, detecção de envoltória, compressão logarítmica e conversão de varredura.

### 1.1.1 Objetivos específicos

Complementando o objetivo principal, os objetivos específicos são:

- Utilizar dados brutos de RF de US provindos da plataforma ULTRA-ORS para o desenvolvimento do sistema embarcado;
- Adotar a placa de desenvolvimento ZedBoard (DIGILENT, EUA) para armazenamento dos dados de US, realizar as etapas de processamento em SoPC e apresentar os resultados de reconstrução de imagem em monitor com saída de vídeo do tipo *Video Graphics Array* (VGA);
- Utilizar a plataforma MATLAB/Simulink e a ferramenta de FPGA *Model Composer* (MC) (Advanced Micro Devices Inc., EUA) para implementar e validar as técnicas de processamento de sinais de US;
- Utilizar o ambiente de desenvolvimento Vivado (Advanced Micro Devices Inc., EUA) para implementar e integrar os módulos de processamento com Lógica Programável (LP), o Sistema de Processamento (PS) e o Sistema Operacional (SO);
- Implementar o controle de processamento/periféricos e a etapa de conversão de varredura utilizando a linguagem de programação C no ambiente Vitis (Advanced Micro Devices Inc., EUA);
- Utilizar funções de alto desempenho, como *Direct Memory Access* (DMA), para transferência de dados entre periféricos, processador e memória;
- Validar qualitativamente e quantitativamente os resultados das etapas de processamento através de sinais/imagens simuladas e métricas de avaliação de desempenho;
- Analisar o desempenho da plataforma projetada com base em resultados de tempo de processamento e recursos utilizados.

## 1.2 Estrutura do trabalho

Esta dissertação está subdividida em 5 capítulos. O capítulo 1 apresenta a introdução e os objetivos. O capítulo 2 discorre sobre fundamentação teórica dos princípios utilizados para a geração de imagens em Modo B através da técnica de *beamforming* DAS de RX. O capítulo 3 descreve os materiais e métodos utilizados para projeto e execução do sistema proposto e as métricas adotadas para avaliação. O capítulo 4 apresenta os resultados de simulações e de testes práticos das etapas projetadas e do sistema implementado, além de fazer a discussão do trabalho. O capítulo 5 apresenta as conclusões da pesquisa e as sugestões de trabalhos futuros.



## 2 REFERENCIAL TEÓRICO

Neste capítulo são abordados os conceitos básicos sobre as técnicas de ultrassom utilizadas nesta pesquisa para reconstrução de imagens em Modo B através do *beamforming* DAS. Além deste tipo de *beamforming*, são apresentadas as principais informações sobre as técnicas de processamento de sinais de RF: filtragem digital, apodização, detecção de envoltória, compressão logarítmica com ajuste de faixa dinâmica, e conversão de varredura.

### 2.1 Fundamentos para obtenção de sinais de ultrassom em modo pulso-eco

Ondas de RF na faixa do US são aquelas que possuem frequência acima da audível. Tipicamente, para aplicações de imageamento ultrassônico, são utilizadas frequências entre 1 MHz e 15 MHz para uma melhor resolução espacial dos alvos observados, que é limitado pelo comprimento de onda  $\lambda$ . Frequências mais altas, apesar de possibilitarem a observação de estruturas menores com melhor resolução espacial, também sofrem grande atenuação, o que limita seu uso (SHUNG, 2005; WEBSTER, 2010). Na Equação 1, é mostrada a relação entre comprimento de onda  $\lambda$ , velocidade de propagação do som no meio  $c$  e frequência  $f$ .

$$\lambda = \frac{c}{f} \quad (1)$$

A atenuação da onda ultrassônica segue a relação mostrada na Equação 2, em que  $I_0$  é a intensidade inicial da onda emitida,  $I$  é a intensidade resultante após uma distância percorrida  $r$  e  $\alpha$  é o coeficiente de atenuação de amplitude em Neper (Np) por centímetro (HEDRICK; HYKES; STARCHMAN, 2005). O termo multiplicando  $\alpha$  por 2 adequa o coeficiente de atenuação para ser utilizado com a intensidade, e isso se deve ao fato de que esta é proporcional à amplitude ao quadrado (HEDRICK; HYKES; STARCHMAN, 2005).

$$I = I_0 e^{-2\alpha r} \quad (2)$$

Como ocorre atenuação das ondas durante a sua propagação, há também uma impedância acústica  $Z$  relacionada ao meio, definida pela Equação 3, na qual  $\rho$  é a densidade do meio.

$$Z = \rho c \quad (3)$$

A porção da onda que é refletida e que, portanto, determina a intensidade dos ecos em interfaces entre dois meios com diferentes  $Z$  segue a relação da Equação 4, em que  $R$  é a fração refletida,  $Z_1$  e  $Z_2$  são as impedâncias características dos meios 1 e 2, onde há tal interface (HEDRICK; HYKES; STARCHMAN, 2005). Complementarmente, valores usuais para parâmetros de propagação de ondas acústicas são mostrados na Tabela 1.

$$R = \left( \frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \quad (4)$$

**Tabela 1 – Parâmetros de propagação de sinais de ultrassom em diferentes meios, à 1 MHz.**

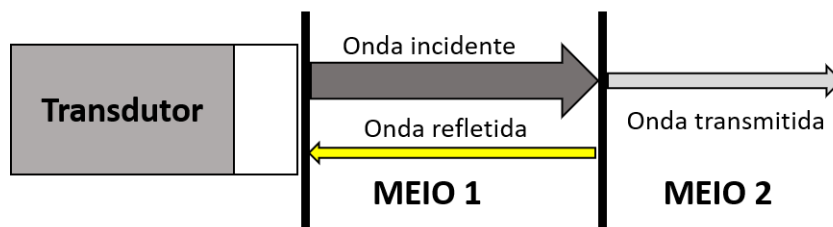
Meio	Coefficiente de Atenuação (Np/cm)	Velocidade (m/s)	Impedância Acústica (kg/(m <sup>2</sup> s) x 10 <sup>6</sup> )
Água (20°C)	2,53x10 <sup>-4</sup>	1480	1,48
Ar	8,63x10 <sup>-1</sup>	330	4x10 <sup>-4</sup>
Tecido mole	8,63x10 <sup>-2</sup>	1540	1,63
Osso	1,73	4080	7,8
Gordura	7,25x10 <sup>-2</sup>	1459	1,38

**Fonte: Adaptado de Hedrick, Hykes e Starchman (2005), Shankar e Pagel (2011).**

Um transdutor emitindo ondas acústicas pode também receber os ecos destas ondas e assim é possível determinar propriedades do meio analisado. Os transdutores são baseados no efeito piezoelétrico, no qual um cristal é excitado com um pulso elétrico e produz uma contração ou expansão, gerando assim um pulso acústico (SHUNG, 2005; WEBSTER, 2010). O efeito contrário também é válido, assim o mesmo transdutor atua como sonógrafo, recebendo intensidades sonoras e as transformando em sinais elétricos. Portanto, ao emitir e receber sinais de US, pode-se analisar os meios de propagação, sendo esse modo de operação denominado de pulso-eco (HEDRICK; HYKES; STARCHMAN, 2005).

Na Figura 1, é ilustrado um diagrama com transdutor operando em modo pulso-eco, no qual as ondas refletidas contêm informação das interfaces entre os meios analisados. Quando essa informação é utilizada para se fazer uma imagem bidimensional cujos eixos são a amplitude e a profundidade das interfaces, tem-se o Modo Amplitude, ou Modo A de operação. Neste, o valor da amplitude se refere à intensidade do sinal refletido nas interfaces entre meios e a distância dos alvos é associada com os tempos de propagação da onda.

**Figura 1 – Comportamento de ondas em interfaces entre meios com diferentes impedâncias acústicas.**



**Fonte: Adaptado de Hedrick, Hykes e Starchman (2005).**

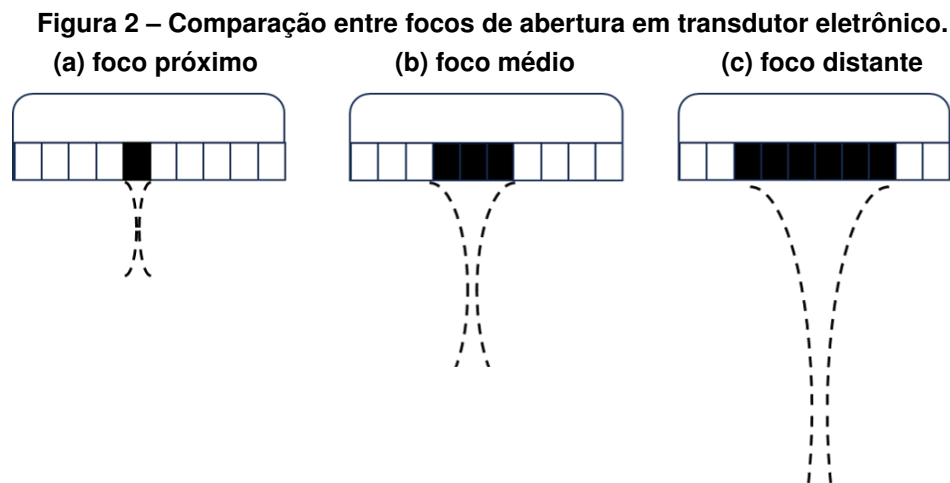
No modo pulso-eco é possível calcular a velocidade máxima da aquisição de quadros, ou a quantidade máxima de *Frames per second* (FPS) do sistema. A Equação 5 mostra a razão de atualização de quadros por segundo, sendo  $c$  a velocidade de propagação da onda no meio,  $f_c$  é o ponto de foco, em metros, e  $n$  é a quantidade de linhas de varredura do transdutor (HEDRICK; HYKES; STARCHMAN, 2005).

$$FR_{max} = \frac{c}{2f_c n} \quad (5)$$

## 2.2 Beamforming do tipo *Delay and Sum* para transdutores multielementos

A utilização de transdutores multielementos permite a manipulação do sinal produzido ou recebido, em um processo conhecido como *beamforming* e traz vantagens para o resultado, como o aumento do *Signal to Noise Ratio* (SNR) da linha de varredura resultante e melhorias na resolução espacial (ASSEF, 2013). Esse processo, quando constituído por operações de atraso e soma, ou DAS, é chamado de somatório coerente.

Transdutores eletrônicos formados por matrizes com múltiplos elementos podem se utilizar dessa característica para manipular as zonas focais e movimentar a zona de visão. A Figura 2, mostra o processo de focalização eletrônica de abertura com três exemplos (focos próximo, médio e distante), que se utiliza do princípio de soma das ondas geradas pelos diversos cristais para criar regiões de focos em diferentes distâncias/profundidades. O foco também pode ser alterado pelo perfil de atrasos dos canais ativos na emissão dos sinais (ASSEF, 2013).

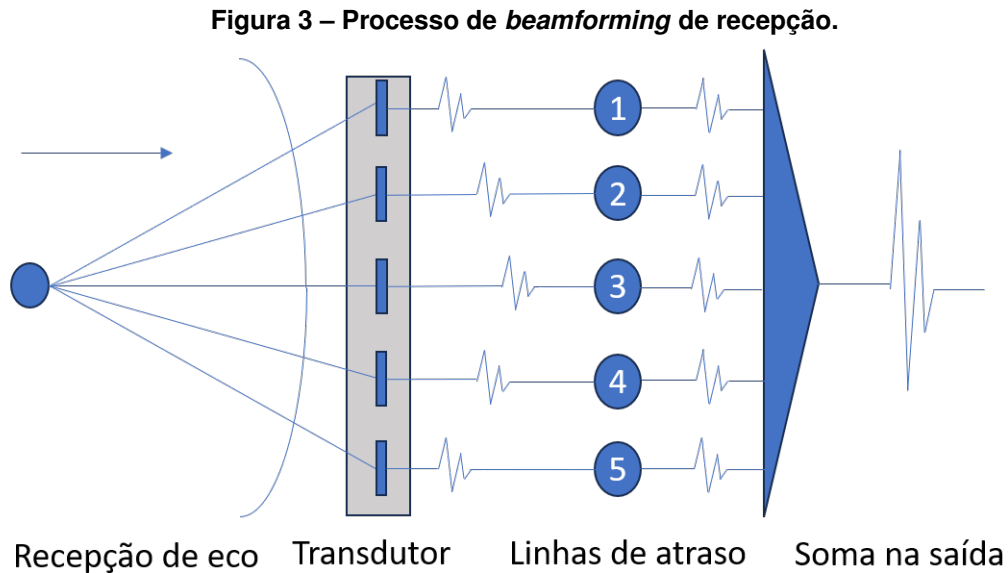


Fonte: Adaptado de Hedrick, Hykes e Starchman (2005).

No momento da recepção, pode-se sincronizar todos os  $n$  canais ativos com as frentes de onda recebidas, adicionando atrasos em cada um deles e criando um foco dinâmico de RX. Em seguida, é realizado alinhamento temporal e a soma de cada canal de forma coerente, formando o *beam*, ou feixe resultante, em um processo conhecido como *beamforming* DAS de RX (HEDRICK; HYKES; STARCHMAN, 2005; ASSEF, 2013). Este processo pode ser associado também com a etapa de apodização, em detalhes na subseção 2.3.2, e é descrita pela Equação 6, na qual  $S(t)$  é o sinal de RF resultante,  $w_i$  é o peso da apodização,  $S_i$  é o sinal do elemento  $i$  com atraso de focalização  $\tau$ , para a abertura  $n$  (FRAZIER; O'BRIEN, 1998).

$$S(t) = \sum_{i=0}^{n-1} w_i S_i(t - \tau_i). \quad (6)$$

A Figura 3 ilustra o processo de somatório coerente.



Fonte: Adaptado de Hedrick, Hykes e Starchman (2005).

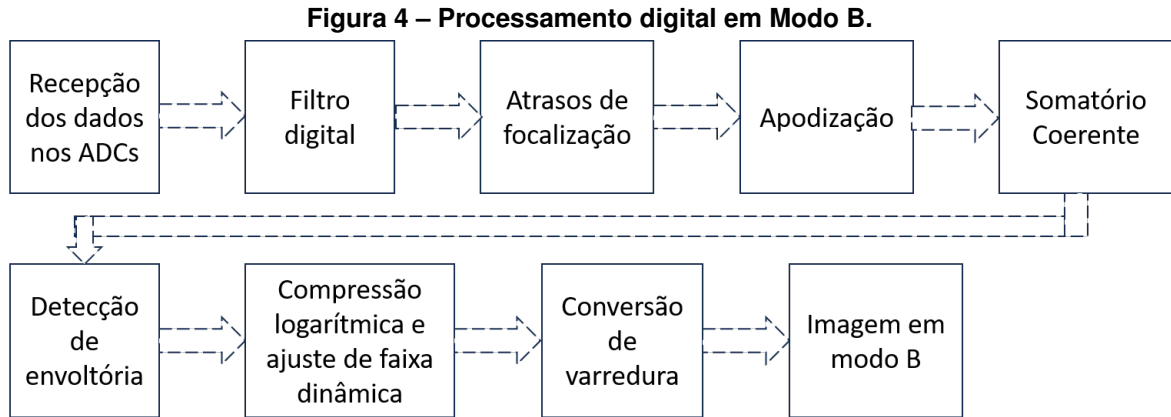
### 2.3 Formação de imagens em Modo B

A partir de transdutores de varredura mecânicos e transdutores de matriz eletrônica é possível se obter diversas linhas de varredura e, com o correto processamento de sinais e imagens, gerar uma imagem resultante em Modo B (HEDRICK; HYKES; STARCHMAN, 2005; WEBSTER, 2010). A quantidade de linhas de varredura adquiridas por um transdutor de matriz multielementos,  $n_{sc}$  é determinada pela quantidade de canais ativos  $na$  durante a aquisição do sinal, do total  $ne$  de elementos disponíveis, como na Equação 7 (ASSEF, 2013).

$$n_{sc} = ne - na + 1. \quad (7)$$

Após a obtenção de linhas de varredura em diversos canais e compensando as perdas de propagação através do módulo *Time Gain Compensation* (TGC), no conjunto de etapas chamado de *front-end*, os sinais passam para o processamento *back-end* com as seguintes etapas: conversão analógico para digital, filtragem, focalização, apodização, somatório coerente, detecção de envoltória e, no caso em que são utilizados transdutores de matriz convexa, a última etapa adicional de conversão de varredura é necessária para a correta exibição das imagens em Modo B (SCHNEIDER *et al.*, 2010; JONVEAUX *et al.*, 2022; ASSEF, 2013). No diagrama da Figura 4, são ilustradas as principais etapas do *beamforming* de RX necessárias

para se gerar uma imagem em Modo B pelo método DAS. Etapas adicionais, como interpolação e decimação, também podem ser aplicadas para otimizar a qualidade da imagem (FERREIRA, 2017; OLIVEIRA, 2020).



**Fonte: Adaptado de Assef (2013).**

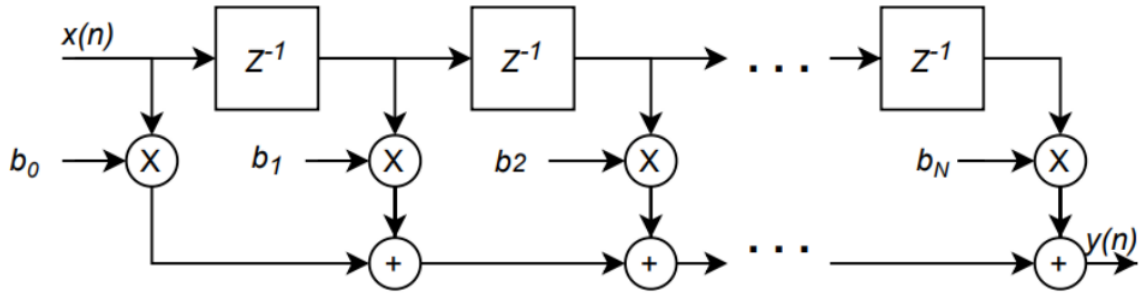
### 2.3.1 Filtragem digital

A filtragem digital é uma forma de manipular sinais com o intuito de excluir ou mitigar componentes de frequência indesejadas. Para tal, filtros FIR são geralmente utilizados, pois apresentam as características desejadas de resposta em fase linear e estabilidade (WINDER, 2002; OPPENHEIM; SCHAFER, 1975). O filtro do tipo FIR pode ser descrito pela convolução discreta da Equação 8, em que a saída  $y(n)$  é a soma ponderada do sinal  $x(n)$ , atrasado em  $i$  amostras, e  $b_i$  é o valor que assume o coeficiente do filtro no intervalo de 0 a ordem  $N - 1$ .

$$y(n) = \sum_{i=0}^{N-1} b_i x(n - i). \quad (8)$$

A estrutura do filtro FIR pode assumir a forma direta da Figura 5, e esta pode ser facilmente implementada em microcontroladores, dispositivos DSP e FPGA. Normalmente, esses filtros assumem alguma simetria, seja com coeficientes repetidos, espelhados ou com valores nulos, o que resulta em implementações com simplificações e economia de recursos, como memória e blocos de DSP (ZHOU; ZHENG, 2015; ASSEF *et al.*, 2016; ASSEF *et al.*, 2018).

Figura 5 – Estrutura de um filtro FIR.

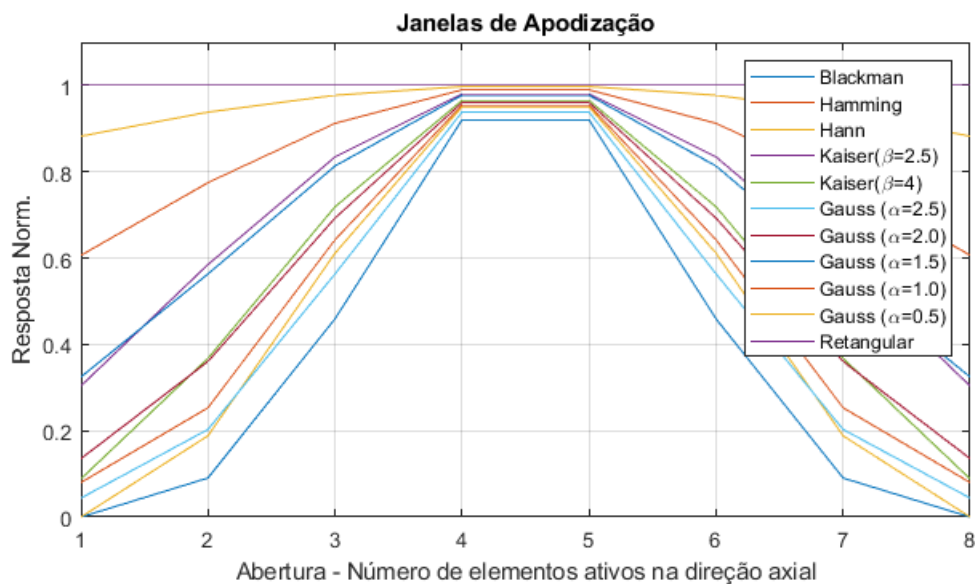


Fonte: Adaptado de Assef *et al.* (2019).

### 2.3.2 Apodização

A apodização é a etapa de processamento que busca melhorar a qualidade dos sinais antes do somatório coerente, através da utilização de janelas de ponderação que podem assumir vários formatos, como as janelas Blackman, Hamming, Hann, Kaiser, Gauss e retangular (OPPENHEIM; SCHAFER, 1975; WINDER, 2002). Na Figura 6, é mostrada a comparação de diversos perfis de apodização para 8 coeficientes. Ferreira (2017) e Oliveira (2020) testaram implementações com diversas janelas de apodização, e Assef (2013) analisou o impacto desses perfis em imagens em Modo B. Os efeitos da apodização são principalmente a redução de lóbulos laterais do feixe de US e atenuação de sinais indesejados que não fazem parte do lóbulo central (ASSEF, 2013; HEDRICK; HYKES; STARCHMAN, 2005).

Figura 6 – Exemplos de perfis de apodização.

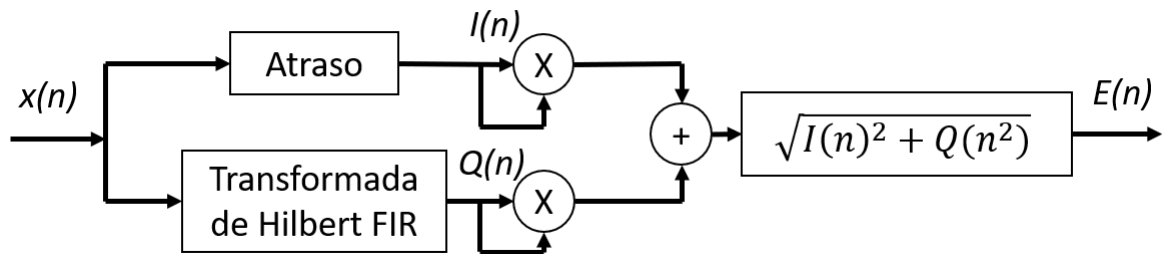


Fonte: Autoria própria (2024).

### 2.3.3 Detecção de envoltória

A etapa de detecção de envoltória é necessária para extração da magnitude das ondas de US (ASSEF *et al.*, 2018). A envoltória é detectada através de diversos métodos, como demonstrado por Chang, Yen e Shung (2007) e Lyons (2017), sendo que dentre eles a Transformada de Hilbert é preferível, pois apresenta pouco custo computacional com resultados precisos (HASSAN; KADAH, 2013; AKKALA *et al.*, 2014; ASSEF *et al.*, 2019). Nessa técnica, obtém-se o módulo da transformada, ou seja, a raiz da soma dos quadrados da resposta em fase, representado por  $I$ , e quadratura da transformada, representada por  $Q$ , transformando o sinal de entrada  $x(n)$  na envoltória  $E(n)$ , conforme demonstrado na Figura 7.

**Figura 7 – Detector de envoltória através da Transformada de Hilbert.**



Fonte: Adaptado de Ruzyk *et al.* (2021).

### 2.3.4 Compressão logarítmica e ajuste de faixa dinâmica

A etapa de compressão logarítmica é necessária para uma exibição adequada da imagem de US em Modo B. Por apresentar valores com uma amplitude alta de variação de intensidade, geralmente esses pontos não são corretamente mapeados em escala de cinza para manter os detalhes na exibição em um display. A compressão logarítmica dimensiona então esses sinais em uma faixa adequada à visão, melhorando o contraste de forma geral (ASSEF, 2013). A Equação 9, descreve a função à qual cada amostra do sinal é submetida, em que o termo  $x_{db}(n)$  é o sinal de US resultante, em dB,  $x(n)$  é o sinal anterior à esta etapa e  $x_{max}$  é o valor máximo a ser mapeado na escala logarítmica.

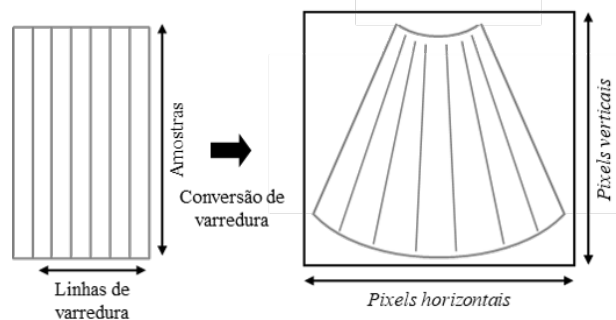
$$x_{db}(n) = 20 \log_{10} \left( \frac{x(n)}{x_{max}} \right) \quad (9)$$

A escolha de uma faixa dinâmica em equipamentos de US permite evidenciar detalhes de regiões específicas da imagem. Isto é feito limitando os valores entre 0 dB e a intensidade mínima desejada. Assim, após aplicar a compressão logarítmica, seus valores são truncados para intensidades mínimas usualmente na faixa de -20 a -40 dB (HEDRICK; HYKES; STARCHMAN, 2005).

### 2.3.5 Conversão de varredura

A conversão de varredura é a última etapa necessária para geração de imagens em Modo B. Ela tem por objetivo a correta exibição de dados adquiridos através de transdutores convexos e se resume a realizar uma conversão de coordenadas na forma polar bidimensional para coordenadas retangulares cartesianas, conforme exibido na Figura 8.

**Figura 8 – Conversão de varredura.**



**Fonte: Adaptado de Assef et al. (2014).**

A transformação de coordenadas é expressa pela Equação 10 e 11, em que  $r$  e  $\theta$  são coordenadas polares. A primeira é a distância radial do transdutor e a segunda é o ângulo entre a linha de varredura e a referência central do transdutor. As coordenadas cartesianas são expressas por  $x$  e  $y$  (LI, 2009).

$$r = \sqrt{x^2 + y^2} \quad (10)$$

$$\theta = \tan^{-1} \frac{x}{y} \quad (11)$$

Além dessas etapas, o processo de projetar um distanciamento angular entre as linhas de varredura gera espaços vazios, onde pixels (elementos que compõem a imagem) assumem um valor nulo. Isso gera um padrão de interferência na imagem, chamado de padrão de Moiré, e é responsável por uma degradação na qualidade da imagem, causando também escurecimento da mesma. Esse problema pode ser solucionado através de diversos métodos, dentre os quais a interpolação (ROBINSON; KNIGHT, 1982).

Vários trabalhos exploraram mecanismos para melhorar a qualidade da conversão de varredura e tornar seus cálculos e interpolações mais rápidos, como o uso de *Lookup Table* (LUT) e algoritmos como o *Coordinate Rotation Digital Computer* (CORDIC) com estratégias de implementação em FPGA (SIKDAR et al., 2001; CHANG; YEN; SHUNG, 2008; QIU et al., 2012; KASSEM; SAWAN; BOUKADOUM, 2005; WANG et al., 2021). A forma mais simples de interpolação é conhecida como vizinho próximo, ou *nearest neighbor*, e ela determina que o



valor a ser dado ao pixel analisado será o mesmo do pixel mais próximo (BERKHOFF *et al.*, 1994).

### 3 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os materiais e métodos utilizados para a implementação do sistema de processamento de dados de ultrassom em um SoPC. Entre os materiais, são expostas as ferramentas e elementos de HW utilizados. Os métodos são demonstrados através da exposição das etapas de projeto e simulação, juntamente com as métricas utilizadas para as avaliações dos sinais e das imagens resultantes.

#### 3.1 Especificação de requisitos

A especificação do sistema de processamento de dados de US, do tipo *beamforming* digital DAS, segue os parâmetros identificados no Quadro 1. Esses requisitos foram baseados no formato de dados adquiridos pelo sistema ULTRA-ORS (ASSEF, 2013; FERREIRA, 2017; OLIVEIRA, 2020), que foi usado como base para aquisição dos dados experimentais de RF utilizados neste trabalho.

**Quadro 1 – Especificação de requisitos do sistema proposto.**

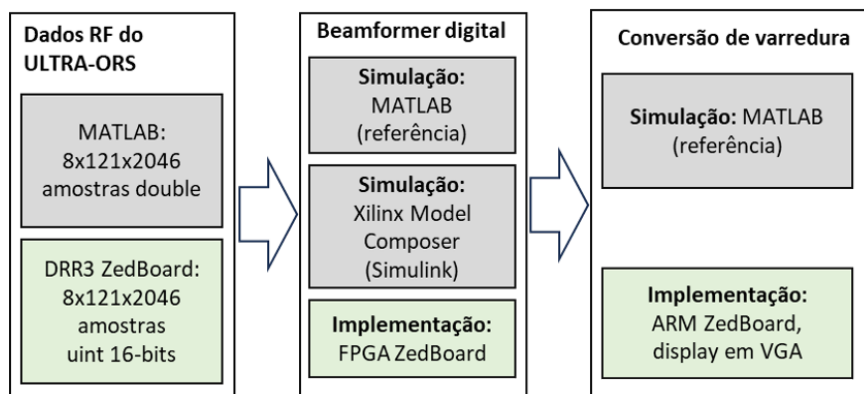
Item	Descrição
1	Utilização de dados de RF de US provenientes do sistema ULTRA-ORS
2	Reconstrução de imagem de transdutor convexo de 128 elementos, modelo AT3C52B
3	Kit de desenvolvimento ZedBoard para processamento em HW/SW embarcado
4	Reconstrução de imagem de US em Modo B
5	Frequência de amostragem de 40 MHz e resolução mínima de 16 bits na LP
6	Armazenamento de 121 aquisições de 8 canais com 2046 amostras na DDR3 da ZedBoard
7	Utilização do barramento AXI4 e DMA para troca de dados de US entre a LP e o SO
8	Fornecimento de dados de RF em 8 canais paralelos ao <i>beamformer</i> da FPGA
9	Filtro digital FIR passa-baixa com frequência de corte em 8 MHz
10	Uso da técnica <i>beamforming</i> para processamento dos sinais com o método DAS
11	Perfil de ajuste de atraso para focalização com profundidade de 25 mm
12	Apodização com perfil do tipo Hanning
13	Aproximação do tipo FIR da Transformada de Hilbert para obtenção de envoltória
14	Compressão logarítmica de -40 dB
15	Conversão de varredura em SW embarcado no ARM Cortex-A9 do SoC à 669 MHz
16	Utilização de display VGA para exibição de imagens de 500×320 pixels em resolução 480p
17	Módulo UART para envio dos dados da imagem ao PC com 8 bits/pixel (total de 160 kB)

**Fonte: Autoria própria (2024).**

No diagrama da Figura 9, são apresentados os principais subsistemas do projeto proposto, juntamente com o direcionamento dos respectivos dados, evidenciando as diferenças entre as simulações e a implementação na ZedBoard, que é feita na plataforma de desenvolvimento de FPGA SoC ZedBoard (DIGILENT, EUA). Os subsistemas se caracterizam pela divisão entre ambientes de projeto e simulação, através da plataforma MATLAB/Simulink e da ferramenta de bloco do Simulink fornecida pela AMD Xilinx, o *Model Composer* (MC), e de implementação prática. Os resultados de avaliação dos sinais e imagens gerados pelas diversas etapas de processamento do sistema foram comparados qualitativamente e quantitativamente com os resultados gerados por *scripts* da plataforma MATLAB, que foram adotados como padrão de referência.

O direcionamento de dados, parte dos resultados da plataforma de aquisição de sinais ULTRA-ORS, resultante do trabalho de Assef (2013), em um conjunto de 121 aquisições em 8 canais simultâneos com 2046 amostras, ver Equação 7 que descreve a relação entre os 128 elementos do transdutor e 8 canais ativos. Estes foram disponibilizados no MATLAB como um vetor do tipo *double*, e na implementação embarcada foram salvos na memória *Double Data Rate* (DDR) do tipo DDR3 da ZedBoard, sendo declarados em software como um *array* de *uint* de 16 bits tridimensional. A última etapa de processamento, conversão de varredura, teve duas implementações: uma em MATLAB e outra programada em linguagem C para o processador Dual-core ARM Cortex™-A9 da ZedBoard.

**Figura 9 – Diagrama do fluxo de dados entre módulos para as simulações e implementação.**



Fonte: Autoria própria (2024).

### 3.2 Softwares de projeto e simulação

São listados a seguir os softwares e ferramentas utilizados neste trabalho:

- MATLAB e Simulink versão R2020b;
- Vivado Design Suite 2020.2, que inclui o Vitis 2020.2 e o Model Composer and System Generator 2020.2 (MC);

- Simulador de campo acústico Field II para MATLAB (JENSEN, 1996).

A plataforma MATLAB foi utilizada como padrão para as comparações qualitativas e quantitativas apresentadas no próximo capítulo. Já o Simulink foi empregado para o projeto e simulação dos sistemas, utilizando recursos disponibilizados pela ferramenta de complemento (*add-on*) do MC. O projeto foi feito com blocos diversos de operações matemáticas e de manipulação de sinais, que por estarem no ambiente Simulink foram também utilizados como recursos de simulação, para visualização de resultados simulados antes da implementação.

Após completar esta etapa, os softwares da AMD Xilinx foram utilizados para projeto e implementação na ZedBoard.

### 3.2.1 Dados de radiofrequência simulados com Field II

O Field II é uma ferramenta de simulação de campo acústico que opera dentro do ambiente do MATLAB. O Field II, permite a simulação de diversos tipos de sinais de US, parametrizados e gerados a partir de transdutores configuráveis (JENSEN, 1996). A ferramenta também fornece funções usuais para processamento de sinais de US, como focalização e apodização, e foram utilizadas como parâmetro de avaliação em diversos trabalhos na área (CHANG; YEN; SHUNG, 2008; RUZYK *et al.*, 2021; JIANG *et al.*, 2022; XU *et al.*, 2022). Neste trabalho, utilizou-se o Field II para gerar dados de US necessários para o teste e validação da etapa de detecção de envoltória. Os parâmetros utilizados para tal são resumidos no Quadro 2.

**Quadro 2 – Parâmetros de simulação com Field II para etapa de detecção de envoltória.**

Parâmetro	Descrição
Frequência de amostragem	40 MHz
Frequência central do transdutor	5 MHz
Tipo de transdutor	Linear de 128 elementos
Elementos ativos do transdutor (abertura)	64
<i>Phantom</i> simulado	10 alvos na vertical com separação de 10 mm
Técnicas de processamento	DAS (apodização e somatório coerente)

Fonte: Adaptado de Ruzyk *et al.* (2021).

### 3.3 Dados de radiofrequência do sistema ULTRA-ORS

Os dados de US foram provenientes da plataforma ULTRA-ORS, desenvolvida pelo Grupo do LUS da UTFPR (LUS, 2023). A plataforma aberta de US possui 128 canais e faz todo o controle dos circuitos de TX e RX de ecos brutos de RF, fornecendo dados digitaliza-

dos com resolução de 12 bits e taxa de amostragem de 40 MSPS (*mega sample per second*) (ASSEF; MAIA; COSTA, 2016).

Os dados aplicados neste trabalho foram obtidos para aplicação do método DAS com abertura ativa de 8 canais com 2046 amostras cada. Dessa forma, considerando o transdutor de 128 elementos, foram geradas 121 linhas de varredura (*scanlines*), conforme a Equação 7. O transdutor utilizado para captura dos dados foi o modelo AT3C52B (BroadSound Corp., Taiwan) de 128 elementos, apresentado na Figura 10. O transdutor possui formato convexo com frequências de operação na faixa de 2 a 5 MHz e frequência central de 3,2 MHz.

**Figura 10 – Transdutor matricial convexo AT3C52B.**



**Fonte: BroadSound (2023).**

Para aquisição dos dados brutos de RF, utilizou-se um corpo de prova mimetizador de tecidos e cistos biológicos, o *phantom* modelo 84-317 da empresa Fluke Corporation, conforme a Figura 11 (ASSEF; MAIA; COSTA, 2016).

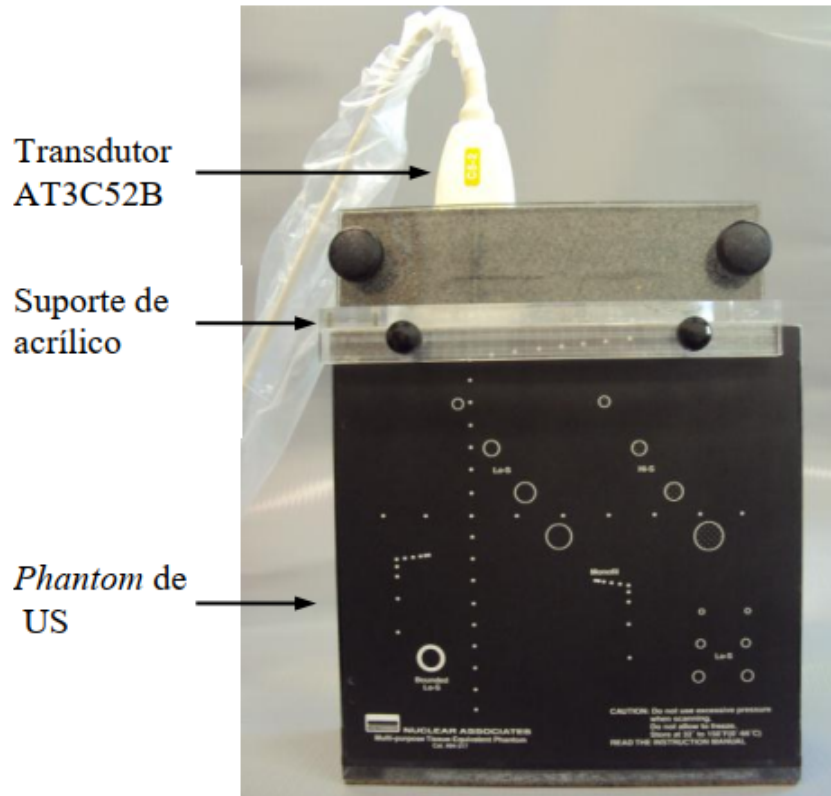
Os parâmetros utilizados na aquisição dos sinais, incluindo dados da geometria do transdutor (JENSEN, 1996), são resumidos na Tabela 2. Além disso, o Field II também foi usado para o cálculo do atraso de focalização de RX, necessário para o ajuste temporal dos dados antes do somatório coerente para formação da linha de varredura.

**Tabela 2 – Parâmetros para aquisição de dados de RF avaliados no trabalho.**

<b>Parâmetro</b>	<b>Valor</b>
Velocidade do som ( $c$ )	1540 m/s
Frequência de amostragem ( $F_s$ )	40 MHz
Número de amostras por canal ( $N$ )	2046
Número de elementos do transdutor ( $N_e$ )	128
Abertura ativa ( $n$ )	8
Kerf (espaço entre os elementos)	115 $\mu$ m
Espessura dos elementos ( $D$ )	0,41 mm
Raio de curvatura ( $R_{convex}$ )	40 mm

**Fonte: Adaptado de Assef (2013).**

Figura 11 – Ensaio de aquisição com o ULTRA-ORS utilizando transdutor AT3C52B e *phantom*.



Fonte: Assef (2013).

### 3.4 Plataforma ZedBoard para processamento de sinais de ultrassom

A plataforma ZedBoard utilizada nesta pesquisa é apresentada na Figura 12. A placa de desenvolvimento ZedBoard é fabricada pela empresa Digilent e conta com um SoPC da AMD Xilinx da linha ZYNQ7000. Esta linha de dispositivo apresenta diversos recursos em um mesmo chip, como processador *hardcore* de dois núcleos ARM Cortex A9, 85.000 células de lógica programável (LP), 53.200 LUT, 106.400 *Flip-Flops*, 4.9 MB de *Block Random Access Memory* (RAM) e 220 DSP *slices* (XILINX, 2018).

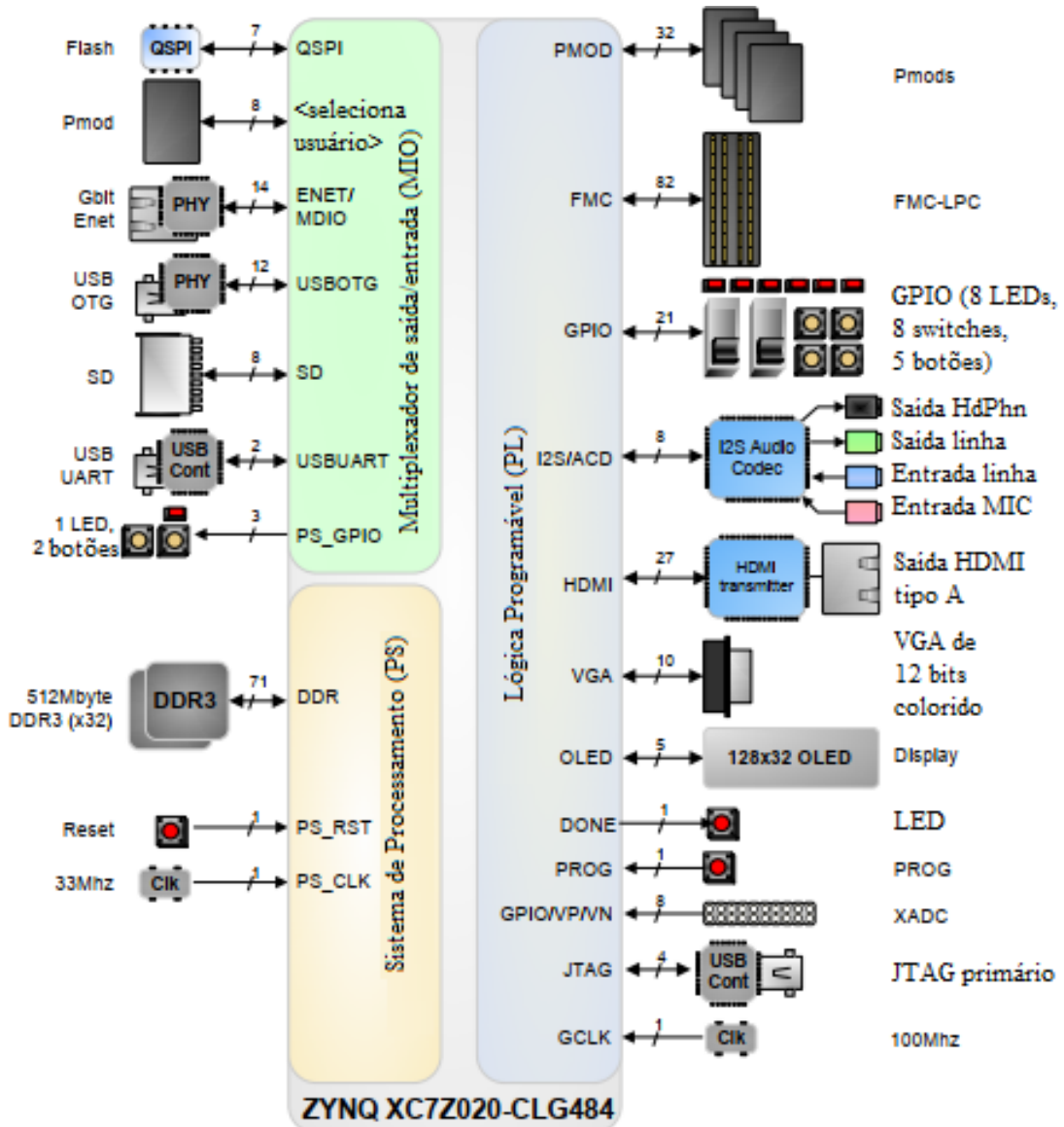
Já a plataforma ZedBoard possui diversos periféricos para facilitar a prototipagem e dar flexibilidade aos projetos de SoPC. Na Figura 13, são mostrados os diversos periféricos disponíveis na placa, entre eles os que foram utilizados neste trabalho: *Universal Asynchronous Receiver/Transmitter* (UART), memórias DDR3 de 512 MB, o ZYNQ7000 de modelo XC7Z020-1CLG484CES EPP, módulo VGA de 4-bits por cor, memória Flash para programação e a interface de programação do dispositivo USB-JTAG (DIGILENT, 2014). Mais detalhes sobre a ZedBoard, suas possibilidades e passo a passo de projeto podem ser encontrados em Crockett *et al.* (2014).

Figura 12 – Plataforma ZedBoard Zynq-7000 ARM/FPGA SoC.



Fonte: Adaptado de Digilent (2014).

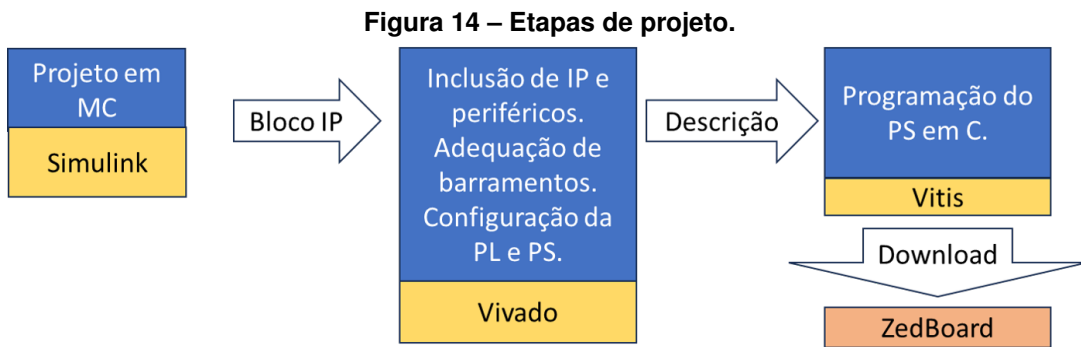
Figura 13 – Principais subsistemas e periféricos disponíveis na placa ZedBoard.



Fonte: Adaptado de Digilent (2014).

### 3.5 Projeto do sistema de processamento de dados de ultrassom

O diagrama apresentado na Figura 14, representa as três principais etapas de projeto: a primeira é a descrição dos elementos implementados em blocos de Propriedade Intelectual (IP), seguida pelo projeto em LP e sua interface com o PS. A terceira etapa é a elaboração do *firmware*, sendo detalhadas nesta seção.



**Fonte: Autoria própria (2024).**

O projeto foi desenvolvido usando um complemento (*add-on*) do MC no Simulink, sendo dividido em 3 blocos, conforme a Figura 15. O primeiro é o *beamformer* que recebe oito canais de dados brutos de RF de US e gera um sinal de envoltória de saída. Essa quantidade de canais foi limitada em 8 para testes iniciais da plataforma, mas poderá ser ampliada como nos trabalhos de Ferreira (2017) e Oliveira (2020). O bloco seguinte realiza o controle das memórias *First-In First-Out* (FIFO) internas, sendo responsável pelo fornecimento dos oito canais de dados em paralelo. O último bloco é a interface entre a LP e o PS. Por motivos de restrição de tempos de propagação na FPGA, foram adicionados blocos de atraso entre as etapas de processamento, mantendo o parâmetro *worst negative slack*, isto é, o tempo de propagação máximo, dentro do requisito para o *clock* definido.

A configuração do bloco *System Generator* segue os parâmetros mostrados na Tabela 3. Esse bloco é responsável por gerar e configurar o IP do sistema, que é exportado para o SW Vivado, descrito na subseção 3.5.9.

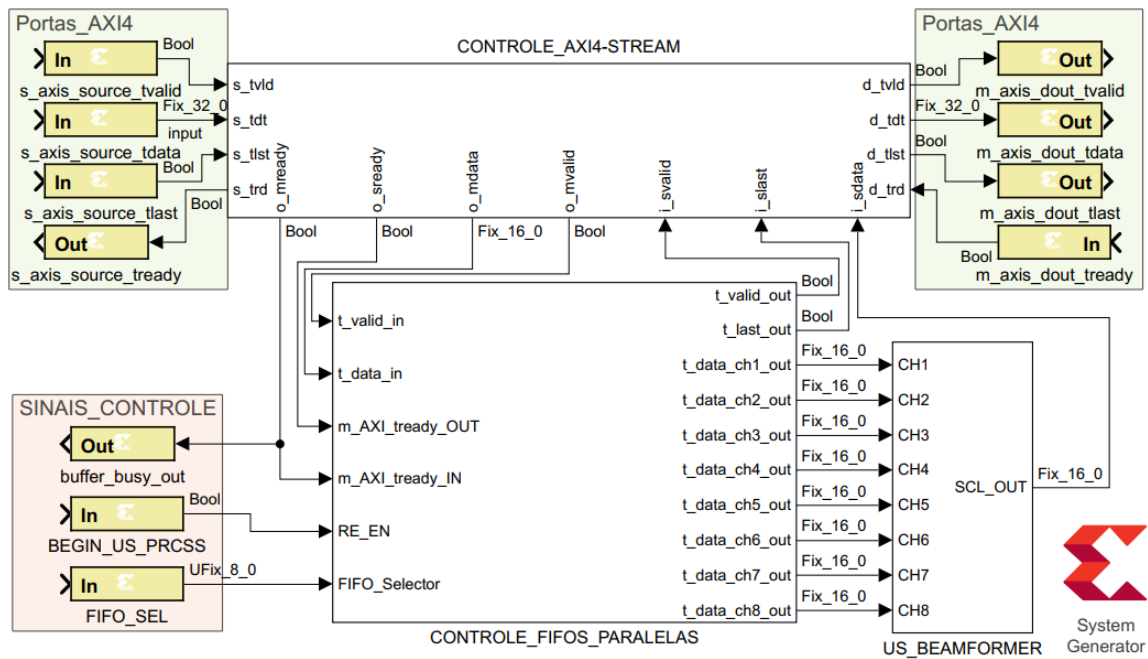
**Tabela 3 – Parâmetros de configuração do Model Composer.**

Parâmetro	Descrição
Placa	ZedBoard ZYNQ <i>Evaluation and Development Kit</i>
Dispositivo	Zynq xc7z020-1clg484
Compilação	IP Catalog
Linguagem	VHDL
Estratégia de síntese	Padrão Vivado
Implementação	Padrão Vivado
FPGA <i>clock</i>	25 ns

**Fonte: Autoria própria (2024).**



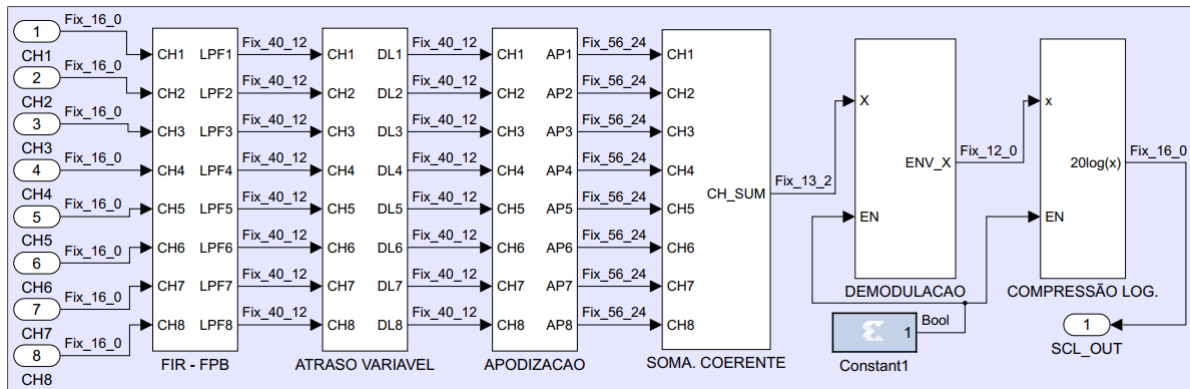
Figura 15 – Visão geral do projeto em Simulink com a ferramenta Model Composer.



Fonte: Autoria própria (2024).

O *beamformer* digital projetado no Simulink com o auxílio da ferramenta MC é apresentado na Figura 16. Esse projeto contém as diversas etapas de processamento em blocos representando os subsistemas. Além disso, é indicada a precisão utilizada para transmissão dos sinais entre os subsistemas, que é feita em complemento de dois e com ponto fixo. O complemento de dois é utilizado para representar números binários negativos e a precisão de ponto fixo mantém a quantidade de casas decimais fixas. A designação "Fix\_y\_z" representa o total de *y* bits, com ponto após o bit de índice *z*. Ao fim, o sinal disponibilizado necessita apenas de uma última etapa de processamento, que é feita em SW, conforme descrito na subseção 3.5.10. Parte dos sinais têm a precisão variando de etapa em etapa, pois a precisão de diversos blocos de processamento é configurada como *full*, ou seja, aumentando conforme as operações são realizadas para preservar a informação decimal.

Figura 16 – Projeto em Simulink com Model Composer do *beamformer* digital e seus subsistemas.



Fonte: Autoria própria (2024).

### 3.5.1 Projeto em Model Composer do Filtro Passa-Baixa

O Filtro passa-baixa (FPB) projetado segue as características mostradas na Tabela 4. Sua estrutura é baseada em filtro FIR e a metodologia de projeto utilizou a técnica *Equiripple*. A estrutura FIR permite explorar a simetria dos coeficientes da resposta do filtro para reduzir pela metade o número de blocos DSP multiplicadores, conforme abordado em diversos trabalhos do Grupo do LUS da UTFPR (ASSEF *et al.*, 2016; OLIVEIRA; ASSEF; MAIA, 2019; MEDEIROS *et al.*, 2022). Para sua implementação, utilizou-se o MATLAB com o *toolbox Filter Design and Analysis Tool* (FDATool), que é parte do System Toolbox 9.11. A ordem 15 com simetria par foi escolhida, mostrando-se suficiente para filtrar os sinais de US e adequada para os requerimentos de HW de um sistema portátil (OLIVEIRA; ASSEF; MAIA, 2019).

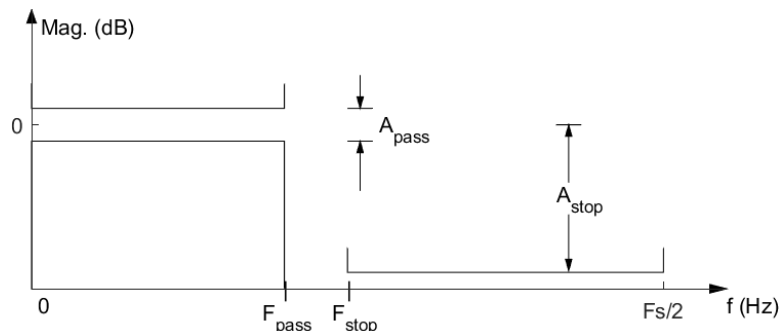
**Tabela 4 – Parâmetros do filtro FIR passa-baixa.**

Parâmetro	Valor
Ordem ( $N$ )	15
Frequência de amostragem ( $F_s$ )	40 MHz
Frequência da banda de passagem ( $F_{pass}$ )	3,2 MHz
Frequência da banda de parada ( $F_{stop}$ )	8 MHz
Peso da banda de passagem ( $A_{pass}$ )	-1 dB
Peso da banda de parada ( $A_{stop}$ )	-50 dB
Fator de densidade ( $dens$ )	20

**Fonte: Adaptado de Ruzyk *et al.* (2021).**

Os parâmetros do filtro FIR são também ilustrados na Figura 17.

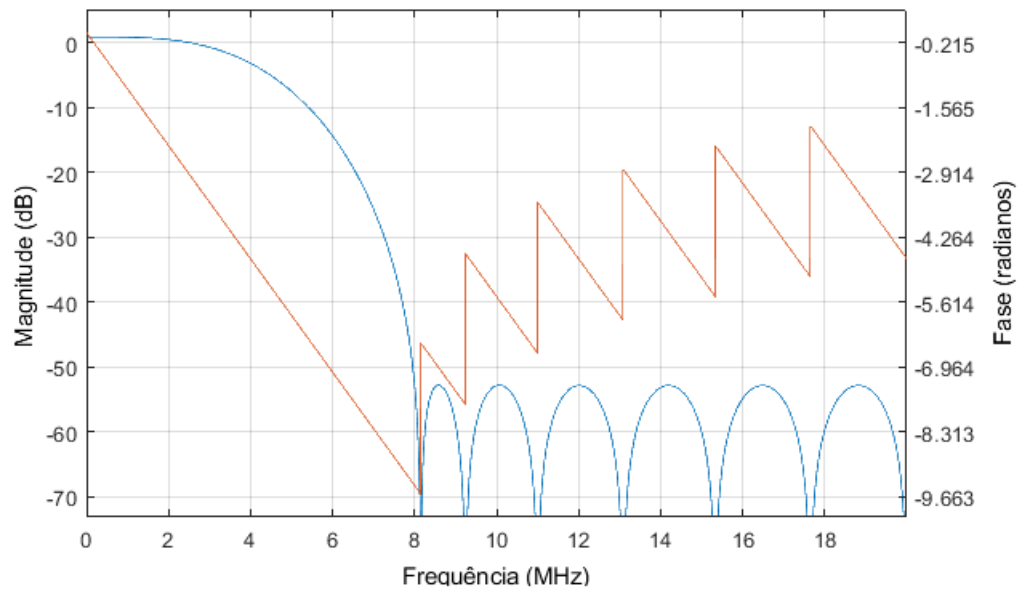
**Figura 17 – Ilustração dos parâmetros de projeto do filtro FIR.**



**Fonte: Autoria própria (2024).**

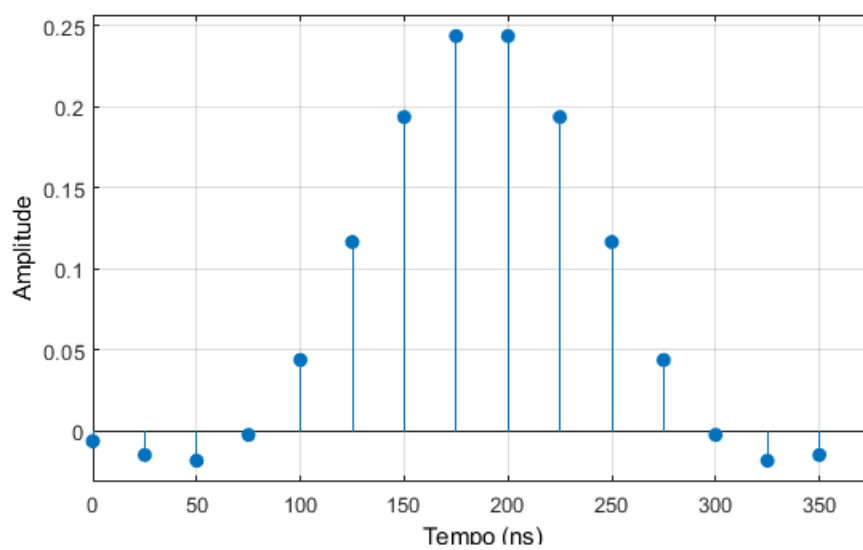
A ferramenta FDATool também permite obter as respostas em frequência de magnitude e fase do sinal filtrado, além da resposta ao impulso, que coincide com os coeficientes utilizados para implementar o filtro FIR com estrutura simétrica, conforme a Figura 18 e Figura 19, respectivamente.

**Figura 18 – Resposta em frequência da magnitude (cor azul) e fase (cor vermelha) do filtro FIR passa-baixa.**



Fonte: Autoria própria (2024).

**Figura 19 – Resposta ao impulso do filtro FIR passa-baixa.**

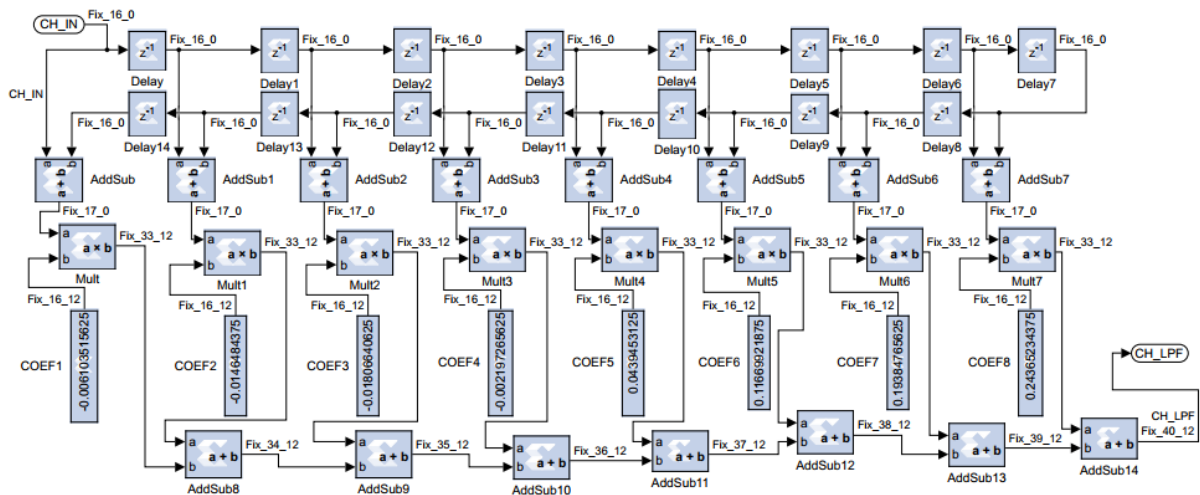


Fonte: Autoria própria (2024).

Na Figura 20, é apresentado o projeto com blocos do MC de um dos oito filtros idênticos implementados. A simetria par foi caracterizada por coeficientes se apresentarem idênticos em pares, como os valores de índice 8 e 9, 7 e 10, e assim sucessivamente até 1 e 16.

Na implementação, essa simetria é expressa pela substituição de multiplicadores, que seriam necessários na estrutura FIR padrão, por oito somadores que recebem os sinais com atraso, o que resulta em economia de elementos lógicos DSP da FPGA. A precisão dos sinais foi inicialmente fixada em 16 bits, mas é aumentada consecutivamente ao passar pelos blocos de soma "AddSub" e multiplicação "Mult", que foram configurados na precisão "full". Isso resulta em um aumento de bits após cada operação, que é otimizado automaticamente pelo MC.

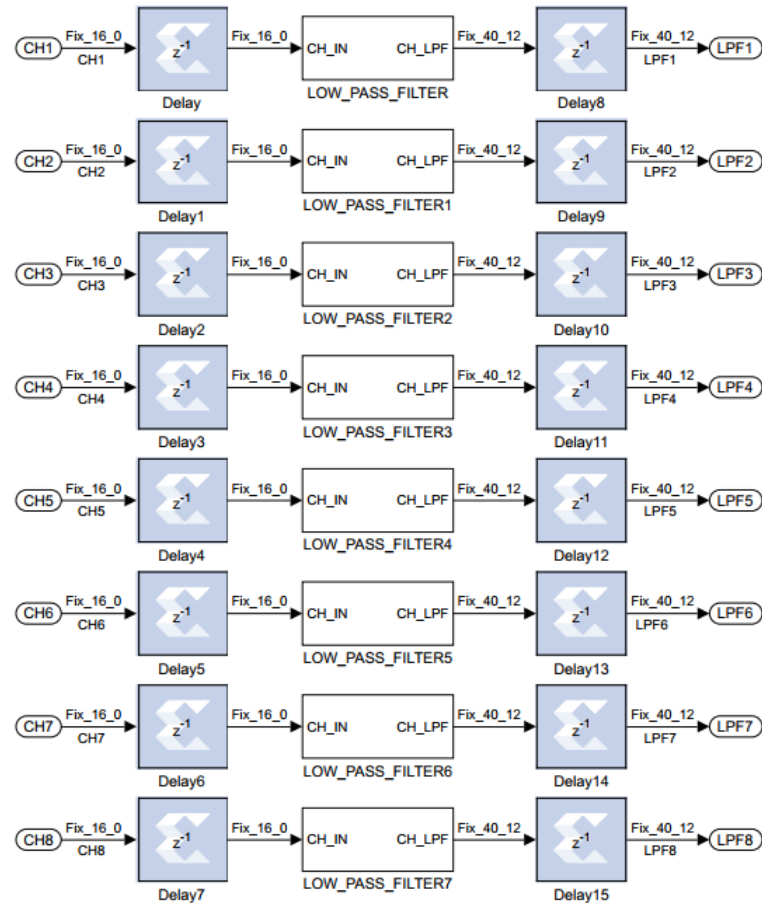
**Figura 20 – Implementação do filtro FIR passa-baixa com blocos do Model Composer.**



**Fonte: Autoria própria (2024).**

A visão geral dos 8 filtros é mostrada na Figura 21, incluindo os atrasos extras necessários para cumprir as restrições de tempo da síntese em FPGA.

**Figura 21 – Implementação dos 8 canais de filtro passa-baixa com blocos do Model Composer.**

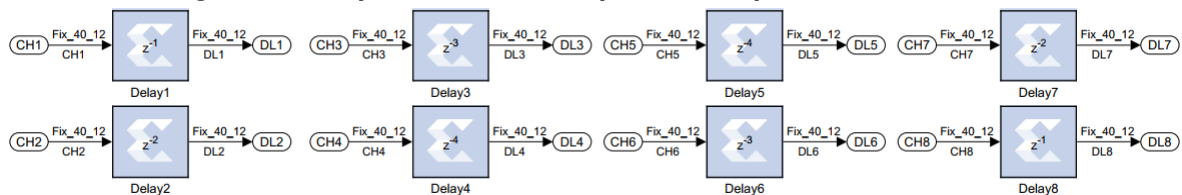


Fonte: Autoria própria (2024).

### 3.5.2 Projeto em Model Composer da etapa de atraso variável

O subsistema de atraso variável é composto por blocos de atraso que têm seus valores provindos do perfil de foco desejado. Neste projeto, o perfil de foco selecionado foi de 25 mm (profundidade) e os dados de atraso foram gerados com o Field II para o transdutor AT3C52B. O procedimento necessário para determinação dos atrasos de focalização para as mesmas configurações de projeto utilizadas e a análise dos perfis de foco são explicados por Assef (2013) e utilizados nos trabalhos de Oliveira (2020) e Ferreira (2017). O projeto do subsistema de atraso variável é apresentado na Figura 22.

**Figura 22 – Projeto em Model Composer da etapa de atraso variável.**

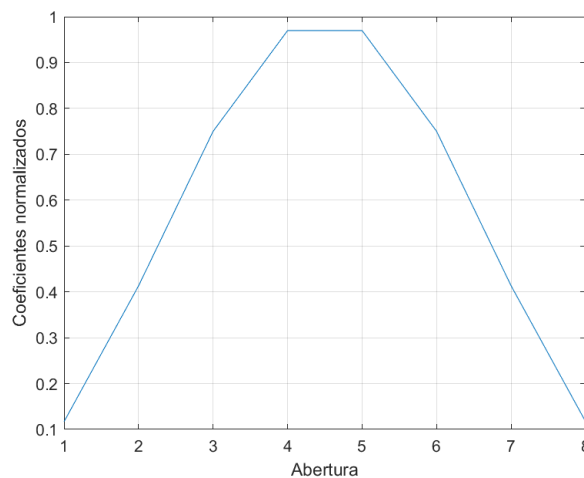


Fonte: Autoria própria (2024).

### 3.5.3 Projeto em Model Composer da etapa de apodização

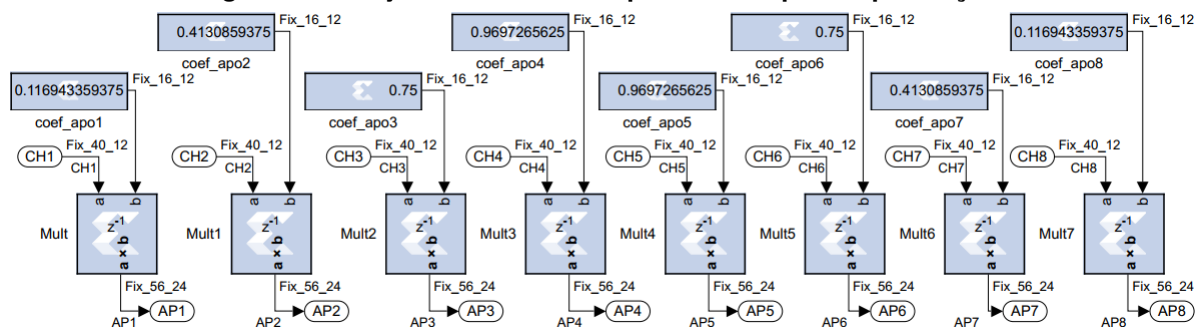
A apodização é feita com 8 coeficientes que representam o perfil de janelamento do tipo Hanning, conforme ilustrado na Figura 23, e foi escolhido por ser um dos mais usuais (JENSEN, 1996). Os coeficientes da função de janelamento foram obtidos através do comando *hann(10)* em *script* do MATLAB, porém sem considerar os valores nulos nas extremidades. Na Figura 24, é apresentado o projeto da etapa de janelamento de apodização, sendo composto de blocos de multiplicação e constantes com precisão configurada como "full".

**Figura 23 – Perfil de apodização com janelamento Hanning com exclusão dos valores nulos nas extremidades.**



Fonte: Autoria própria (2024).

**Figura 24 – Projeto em Model Composer da etapa de apodização.**

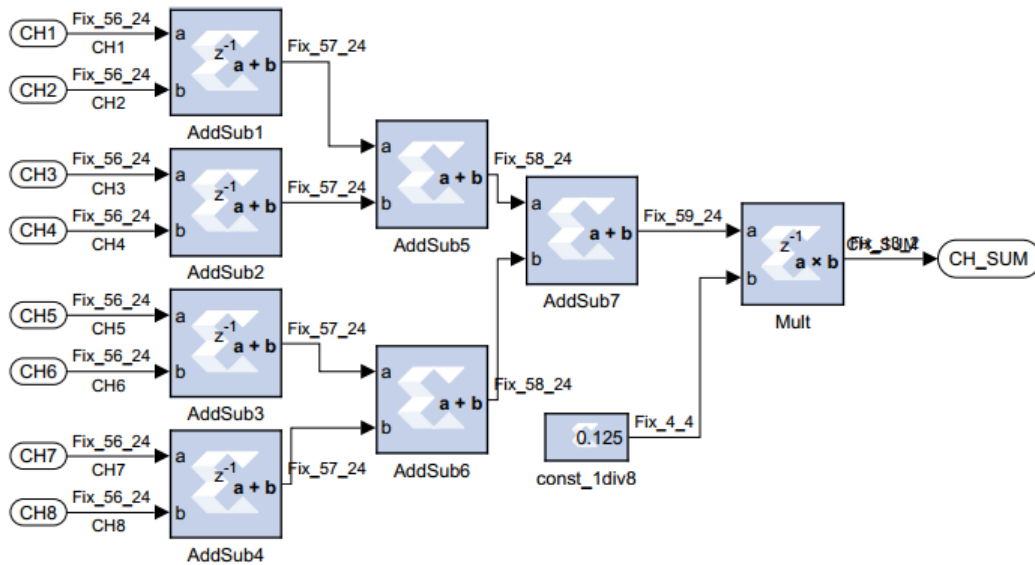


Fonte: Autoria própria (2024).

### 3.5.4 Projeto em Model Composer da etapa de somatório coerente

Em seguida à apodização, ocorre a etapa de somatório coerente. Nessa etapa é realizada a soma dos sinais dos oito canais e posterior normalização através da divisão do resultado por 8 para manter o nível de amplitude semelhante aos canais de entrada. Na Figura 25 é mostrado o projeto do somador coerente usando blocos do MC.

**Figura 25 – Projeto em Model Composer da etapa de somatório coerente.**

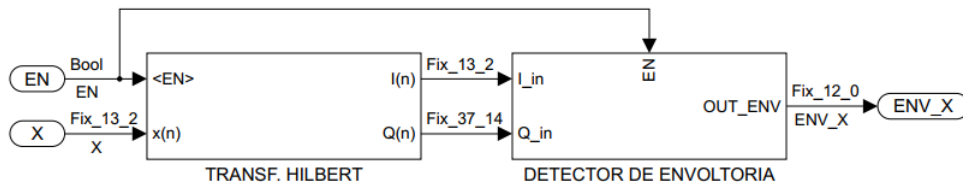


**Fonte: Autoria própria (2024).**

### 3.5.5 Projeto em Model Composer da etapa de detecção de envoltória

A detecção de envoltória foi computada através do valor absoluto da Transformada de Hilbert, sendo utilizada uma aproximação neste trabalho. O projeto mostrado na Figura 26, é composto por duas etapas, realizando primeiramente a aproximação da Transformada de Hilbert com uma estrutura de filtro FIR, seguida pelo cálculo do valor absoluto das componentes em fase  $I(n)$  e quadratura  $Q(n)$ .

**Figura 26 – Projeto em Model Composer mostrando os subsistemas do detector de envoltória.**

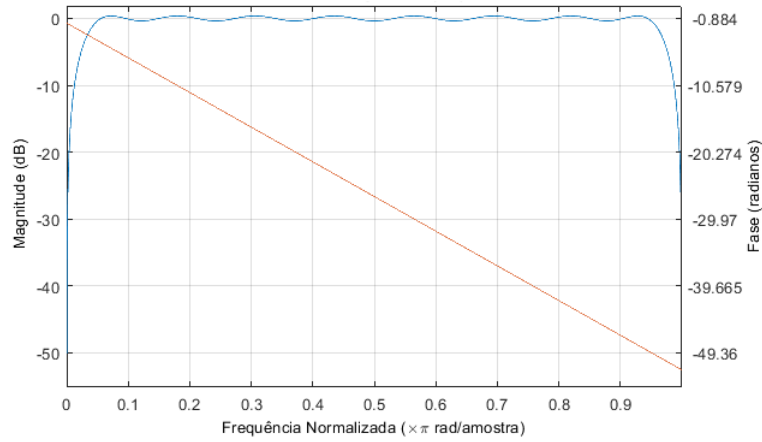


**Fonte: Autoria própria (2024).**

O projeto da Transformada de Hilbert aproximada foi implementado com o auxílio da ferramenta FDATool. Foram gerados os coeficientes da estrutura FIR com ordem 32 pelo método de projeto FIR *Equiripple*, conforme o trabalho de Assef *et al.* (2019). As respectivas respostas em frequência da magnitude/fase e a resposta ao impulso antissimétrica com coeficientes nulos intercalados são mostradas na Figura 27 e Figura 28.

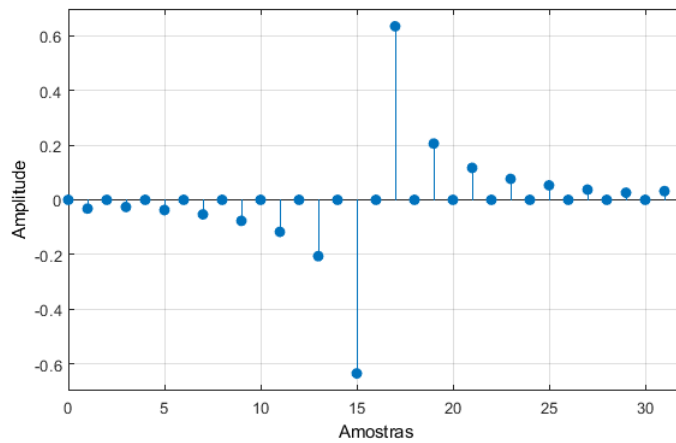
Na Figura 29, é mostrada a implementação da Transformada de Hilbert aproximada em MC. Esta estrutura explora a simetria negativa e ímpar dos coeficientes, além dos valores nulos intercalados, para economizar blocos de operações matemáticas, principalmente multiplicadores e memória.

Figura 27 – Resposta em frequência da Transformada de Hilbert em magnitude (cor azul) e fase (cor vermelho).



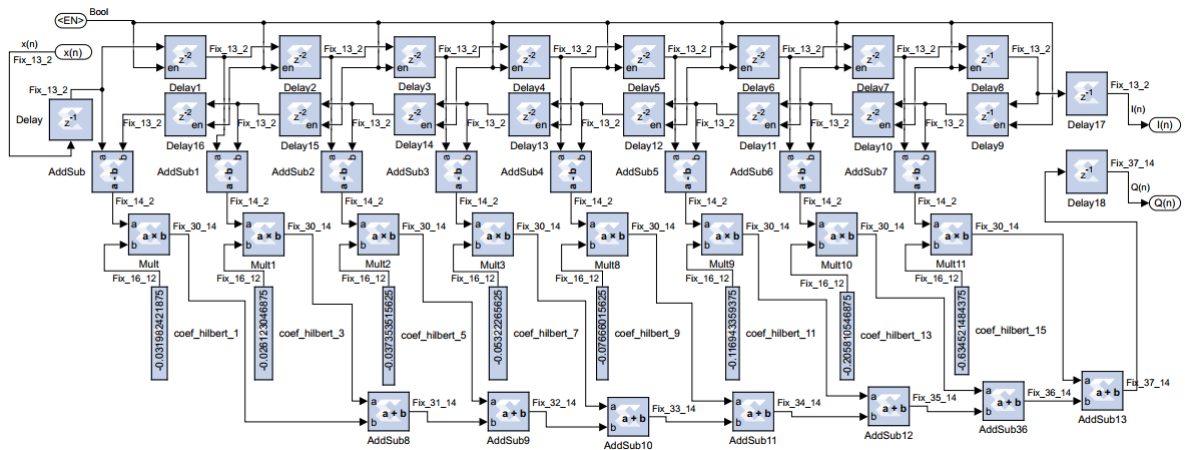
Fonte: Autoria própria (2024).

Figura 28 – Resposta ao impulso da Transformada de Hilbert.



Fonte: Autoria própria (2024).

Figura 29 – Projeto em Model Composer da Transformada de Hilbert aproximada utilizando estrutura de filtro FIR.

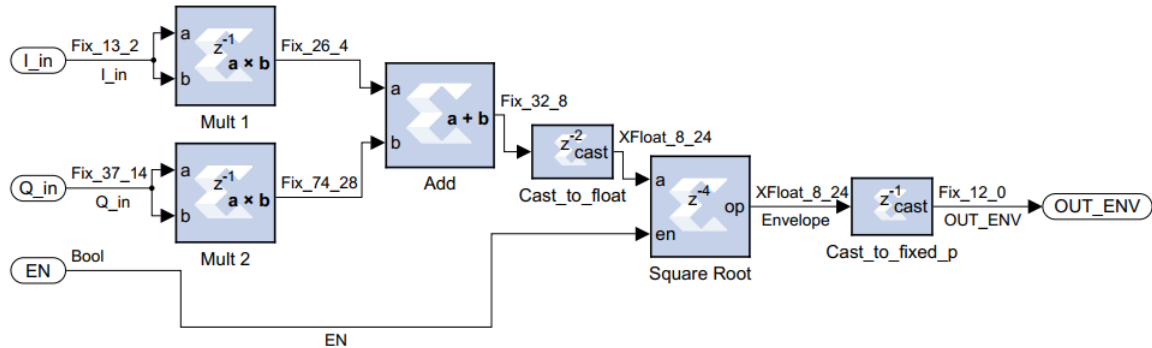


Fonte: Autoria própria (2024).



Conforme ilustrada na Figura 26, a etapa de obtenção da envoltória é feita pelo bloco expandido mostrado na Figura 30, que obtém o módulo das componentes  $I(n)$  e  $Q(n)$ .

**Figura 30 – Projeto em Model Composer do detector de envoltória.**

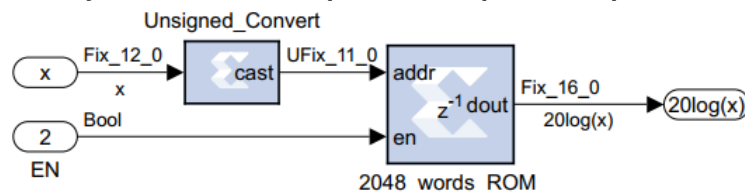


Fonte: Autoria própria (2024).

### 3.5.6 Projeto em Model Composer da etapa de compressão logarítmica

A etapa de compressão logarítmica foi implementada com o auxílio de uma LUT, na qual valores pré-calculados do intervalo de -40 dB a 0 dB são mapeados com normalização de 11 bits (0 a 2047), isto é, com o mesmo comprimento dos dados provindos da etapa anterior. A compressão logarítmica segue a Equação 9, sendo que a amplitude máxima do sinal utilizado corresponde ao valor máximo dos dados de US recebidos do sistema ULTRA-ORS. O projeto em MC da etapa de compressão logarítmica é mostrado na Figura 31.

**Figura 31 – Projeto em Model Composer da etapa de compressão logarítmica.**

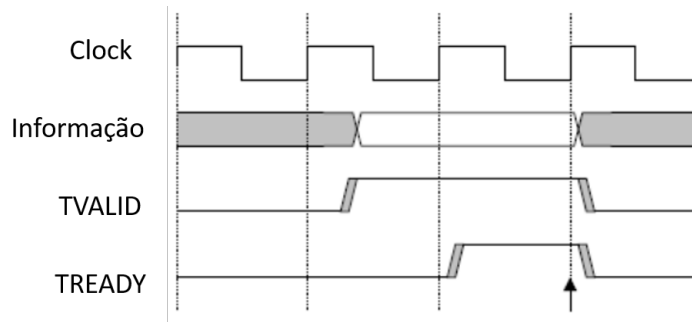


Fonte: Autoria própria (2024).

### 3.5.7 Projeto em Model Composer do bloco para adequação de sinais ao barramento AXI4

A comunicação com o microprocessador no ZYNQ7000 é feita através do barramento AXI4 para aproveitamento de funções de leitura e escrita em *streams* de dados, estabelecidas por IPs controladores em HW. O protocolo utilizado em modo *burst* de dados é descrito por Xilinx (2016) e se baseia na coordenação dos sinais *tvalid* e *tready*, com transações válidas quando ambos estão em nível alto, conforme ilustrado na Figura 32, com detalhe na última borda de subida de *clock*.

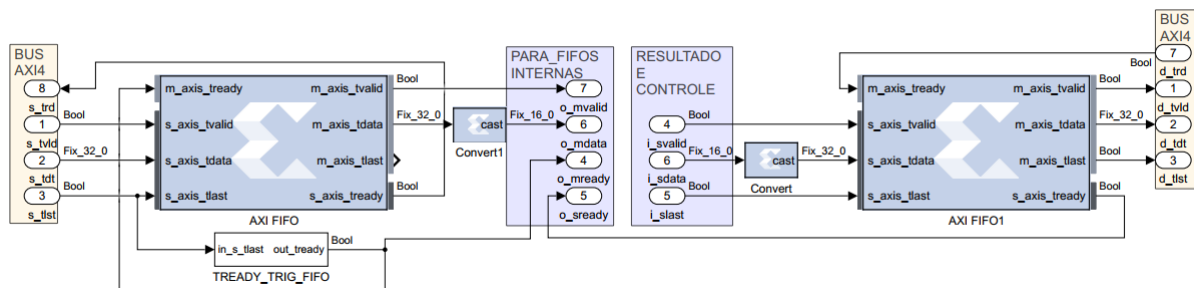
Figura 32 – Exemplo de transação do tipo *stream* através do barramento AXI4.



Fonte: Adaptado de Xilinx (2016).

O módulo projetado está inserido no sistema da Figura 15 e é ilustrado na Figura 33, que mostra as memórias FIFO de entrada e saída de dados do sistema, juntamente com a derivação dos sinais para os outros módulos. Adicionalmente, há uma *flag* opcional *tlast* que indica a última transação de um pacote de dados.

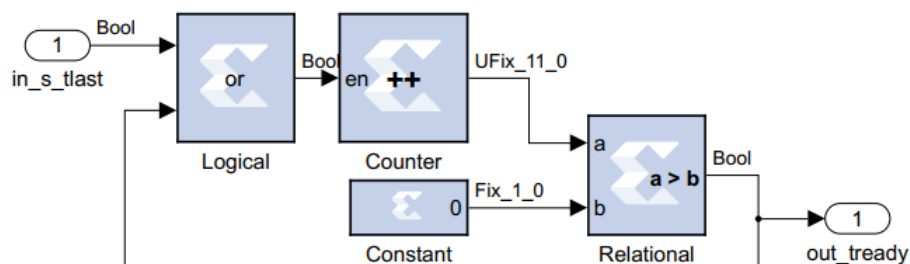
Figura 33 – Projeto em Model Composer da interface com o barramento AXI4 através de memórias FIFO.



Fonte: Autoria própria (2024).

Há também um bloco que estabelece o sinal de controle *trready* para ativar o esvaziamento da memória FIFO de entrada após ser preenchida. Para isso, é verificada a *flag* do *tlast* para ativar um contador que mantém o sinal *trready* em nível alto por 2046 ciclos. Essa implementação é mostrada na Figura 34.

Figura 34 – Bloco de controle do sinal de esvaziamento da memória FIFO de entrada.

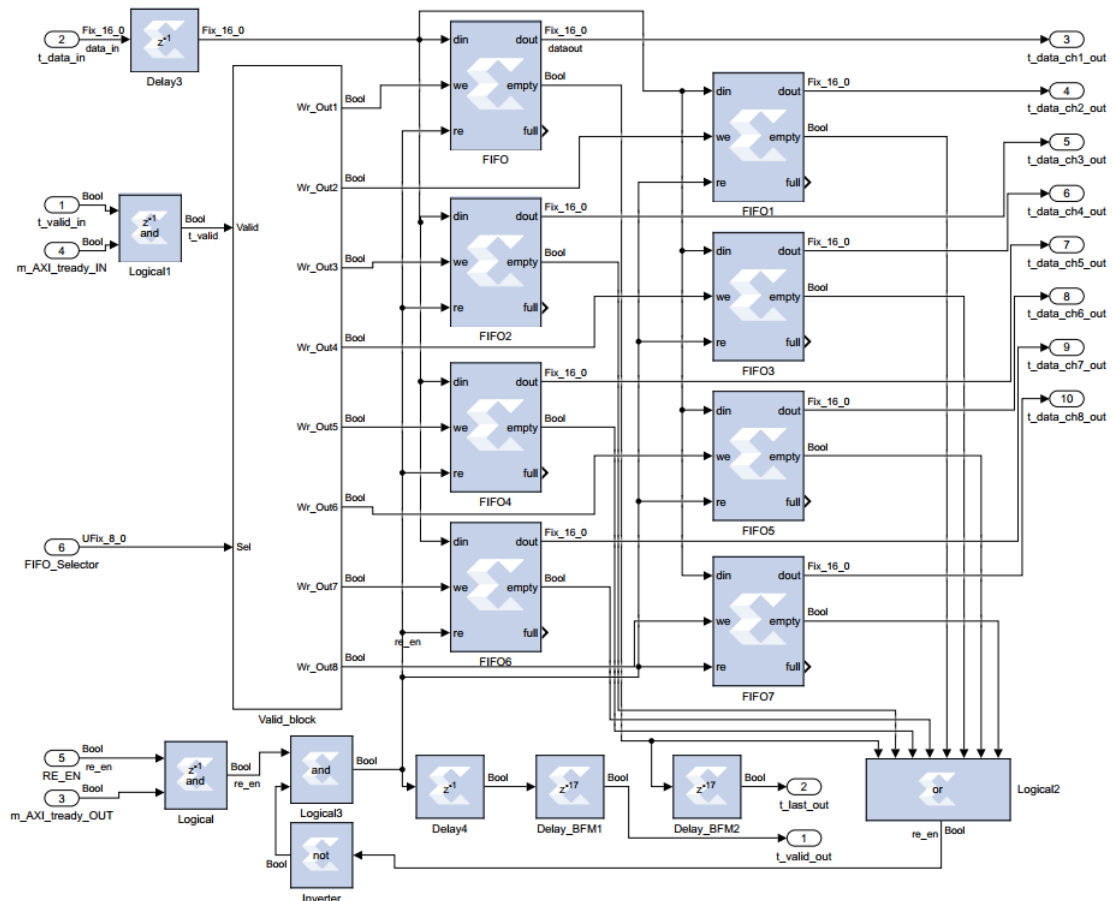


Fonte: Autoria própria (2024).

### 3.5.8 Projeto em Model Composer do bloco de controle de sinais e memórias FIFO paralelas do projeto de ultrassom

Com o propósito de fornecer 2046 amostras simultaneamente para cada um dos 8 canais do *beamformer*, projetou-se um sistema para gerenciar o carregamento de 8 memórias FIFO paralelas, sendo um para cada canal. Após o preenchimento das memórias FIFO (memória cheia), essas são descarregadas e alimentam o *beamformer* em paralelo. Os sinais do barramento AXI4 também são processados e coordenados, tanto para o carregamento das memórias como para a passagem de resultados para a FIFO de saída. O projeto do bloco de controle de sinais e FIFO paralelas é mostrado na Figura 35 e está inserido no sistema da Figura 15.

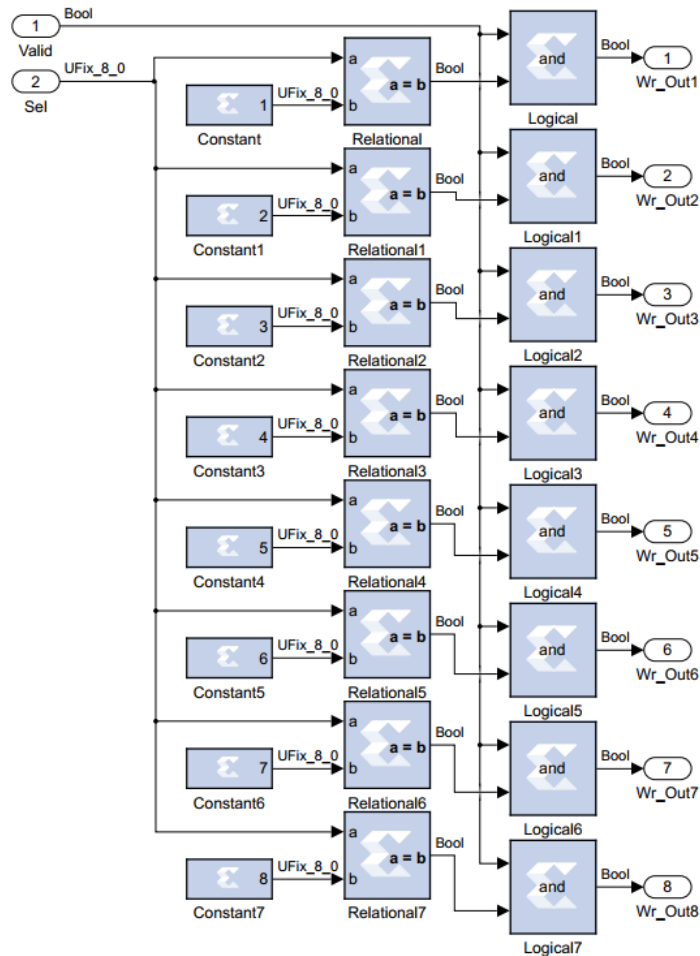
**Figura 35 – Projeto em Model Composer do controlador de memórias FIFO paralelas e sinais de controle internos.**



Fonte: Autoria própria (2024).

A escolha da memória FIFO a ser carregada com dados e a determinação da validade do sinal de US disponível nas entradas das FIFO é feita pelo bloco *valid\_block*, que é expandido na Figura 36. O sinal de entrada *Sel* é utilizado para a seleção da FIFO a ser escrita e o sinal *Valid* é utilizado para indicar que um sinal válido está disponível.

Figura 36 – Projeto em Model Composer do validador de escrita nas memórias FIFO paralelas.



Fonte: Autoria própria (2024).

De forma geral, o funcionamento desse sistema é resumido nas seguintes etapas:

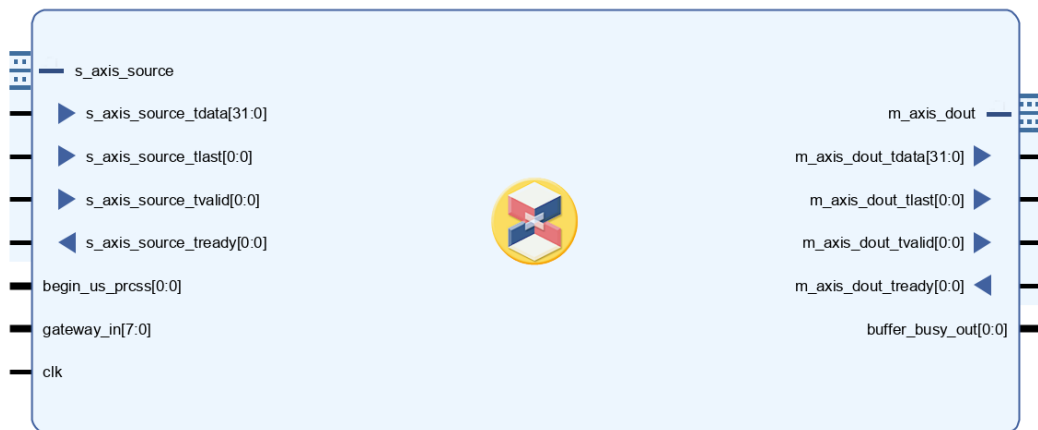
- É necessário começar pela escolha entre 1 a 8 de qual canal será carregado, através da variável FIFO\_Selector;
- A FIFO de interface é carregada por SW através do barramento AXI4;
- Após a conclusão do recebimento, a FIFO interface gera dois sinais válidos, o\_m\_tready e o\_mtvalid, que no subsistema de controle assumem os nomes m\_AXI\_tready\_IN e t\_valid\_in, respectivamente. Quando essa condição é satisfeita, a escrita na FIFO interna selecionada é permitida. A cada ciclo de *clock*, a FIFO habilitada armazenará o sinal disponível na sua entrada *din*. Os dois sinais de validação ficam positivos até o fim do carregamento;
- O processo se repete até as 8 memórias FIFO serem carregadas;
- Após isso, a ativação do *beamformer* ocorre quando o comando RE\_EN é solicitado por SW, e quando a FIFO de saída informa que está pronta para receber dados através do sinal “m\_AXI\_tready\_OUT”;

- Após todos os dados serem fornecidos ao *beamformer*, os sinais *t\_last\_out* e *t\_valid\_out* indicam o fim do processamento e a leitura dos dados é feita por SW.

### 3.5.9 Projeto de hardware no Vivado

Após a validação em Simulink e MC, o projeto do *beamformer* foi exportado como um IP para o SW Vivado, através do bloco *System Generator*, mostrado na Figura 15. Este então pode ser acessado dentro do ambiente de projeto em alto nível do Vivado, sendo disponível como um bloco, conforme ilustrado na Figura 37. Informações sobre o projeto com esta ferramenta são expostos por Xilinx (2023) e Crockett *et al.* (2014).

**Figura 37 – Bloco IP do *Beamformer* exportado para o Vivado.**



**Fonte: Autoria própria (2024).**

As portas para transações de dados foram configuradas com a nomenclatura adequada para integração com o barramento AXI4. A função dos demais sinais têm sua descrição a seguir:

- *begin\_us\_prccs[0:0]*: variável que autoriza o envio de dados das memórias FIFO internas para o *beamformer*;
- *gateway\_in[7:0]*: variável para seleção do canal de US a ser carregado individualmente na memória FIFO interna;
- *clk*: sinal de *clock* do *beamformer* alimentado com fonte de 40 MHz;
- *buffer\_busy\_out[0:0]*: variável que indica se a FIFO auxiliar do barramento AXI4 está sendo utilizada para carregar alguma FIFO interna.

Para a integração com o PS foi necessária a utilização do sistema ZYNQ7. Esse sistema possui os recursos e conexões para a utilização do ARM interno, memórias DDR3, *clock*, barramentos AXI, *General Purpose Input/Output* (GPIO), e demais periféricos (XILINX, 2017). O projeto completo desenvolvido no Vivado é apresentado no Apêndice A.

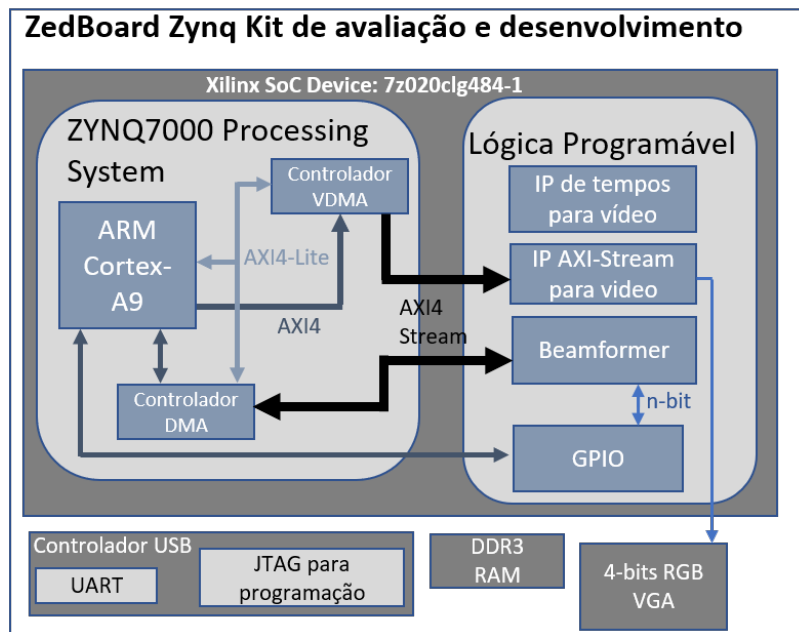
### 3.5.10 Software do sistema de ultrassom e conversor de varredura

A análise da conversão de varredura foi realizada inicialmente gerando imagens através de *script* do MATLAB para comparação, através das funções *meshgrid*, *pol2cart* e *surf*, conforme descrito por Assef *et al.* (2014) e Ruzyk *et al.* (2022).

O software do sistema proposto controla todos os elementos da Figura 38, que contém módulos de LP e PS, além de periféricos disponibilizados pela ZedBoard. Todo o acesso aos controladores de fluxo DMA é feito através dos barramentos AXI e, após configurados, estes operam em paralelo com o PS.

O módulo *beamformer* contém todas as etapas de processamento projetadas, com exceção do conversor de varredura, que é implementado em SW. Além disso, o *beamformer* recebe comandos através do GPIO e envia/recebe dados através de *streams* de dados, que são direcionados para as memórias DDR3 através do controlador de DMA. Para a transferência de dados para monitor de vídeo por meio de sinal analógico do padrão VGA, foram utilizados os blocos de controlador VDMA, controlador de temporização da escrita de pixels e AXI-Stream para vídeo, que por sua vez já fornece os bits necessários para exibição no display.

**Figura 38 – Arquitetura e barramentos do sistema implementado.**



**Fonte: Autoria própria (2024).**

A implementação do conversor de varredura foi feita dentro do PS ZYNQ7000 e é baseada no cálculo da posição dos elementos da matriz resultante, que é guardado em memória, e de um interpolador do tipo *nearest neighbor*, que completa os pixels nulos com o valor do pixel mais próximo (BERKHOFF *et al.*, 1994; SIKDAR *et al.*, 2001).

### 3.6 Métricas quantitativas para comparação de imagens e sinais de ultrassom

Para a comparação entre envoltórias, além de ser feita de forma qualitativa, utilizou-se as métricas de *Normalized Residual Sum of Squares* (NRSS), ou soma normalizada de quadrados residual, e *Normalized Root Mean Square Error* (NRMSE), ou erro quadrático normalizado (ASSEF *et al.*, 2019; CHANG; YEN; SHUNG, 2007). O NRSS e NRMSE são obtidos através da Equação 12 e Equação 13, respectivamente, em que  $E(n)$  é o *envelope* ou envoltória de referência,  $ht$  é a envoltória analisada e  $\overline{ht}$  é o valor médio das envoltórias analisadas.

$$NRSS = \frac{\sum_{n=0}^{M-1} |E(n) - ht(n)|^2}{\sum_{n=0}^{M-1} |ht(n)|^2}, \quad (12)$$

$$NRMSE = \sqrt{\frac{\sum_{n=0}^{M-1} |E(n) - ht(n)|^2}{\sum_{n=0}^{M-1} |ht(n) - \overline{ht}|^2}}, \quad (13)$$

Para a análise da qualidade das imagens produzidas, as métricas de resolução de contraste, *Contrast Resolution* (CR), computada pela Equação 14, e relação contraste-ruído, *Contrast-to-Noise Ratio* (CNR), obtida pela Equação 15, foram empregadas (FRAZIER; O'BRIEN, 1998; GAMMELMARK; JENSEN, 2003). Quanto maior o CNR significa que mais detalhes se sobrepõe e se distinguem do ruído da imagem e, no caso do CR, quanto mais próximo de 1 indica uma melhor resolução (GAMMELMARK; JENSEN, 2003). Os termos  $\mu_t$  e  $\mu_b$  representam a intensidade média nas regiões de interesse e de fundo, respectivamente, e o termo  $\sigma_b$  é o desvio padrão da intensidade da região de fundo. Ambas as relações são adimensionais.

$$CR = 1 - \frac{\mu_t}{\mu_b} \quad (14)$$

$$CNR = \frac{|\mu_t - \mu_b|}{\sigma_b} \quad (15)$$

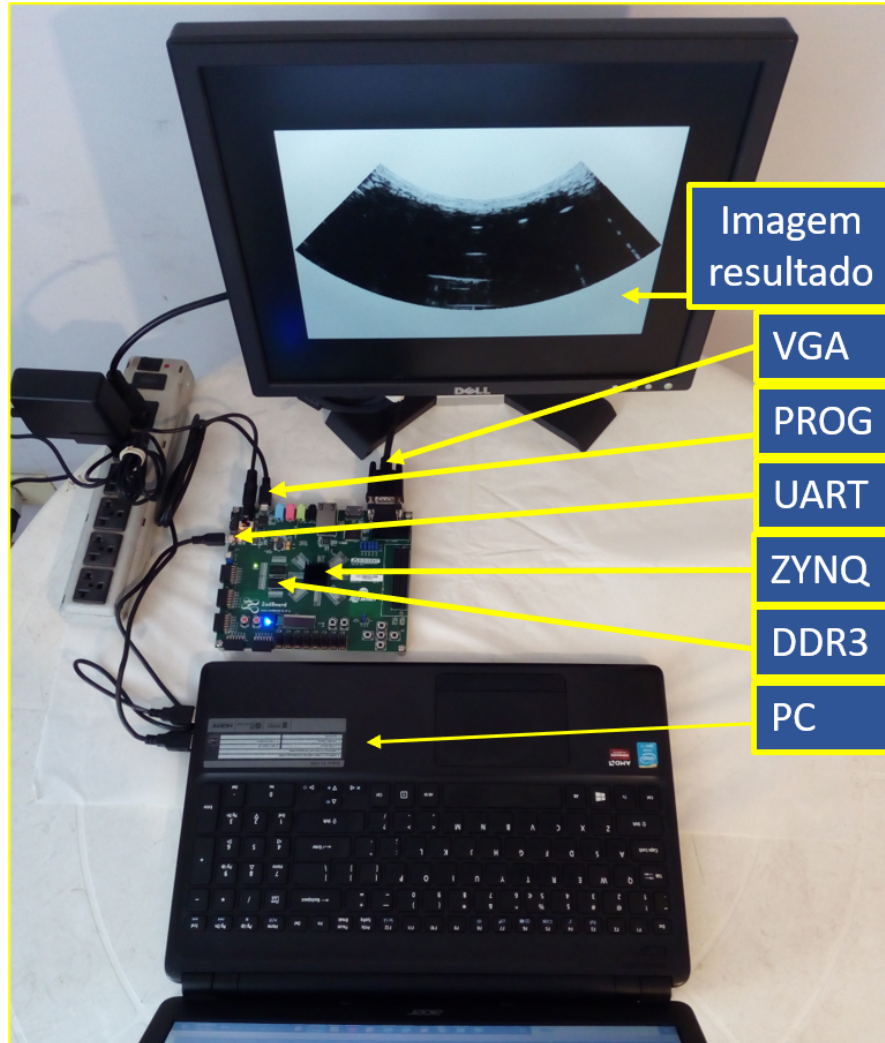
Rodriguez-Molares *et al.* (2020), Esmailian e Mohammadzadeh Asl (2022) e Neves (2020) apresentam outras métricas válidas para comparações quantitativas de dados de US. Porém, a escolha das métricas descritas nesta seção foi baseada na possibilidade de comparações com trabalhos correlatos, como Ferreira (2017), Assef *et al.* (2019) e Oliveira (2020), que utilizaram o mesmo conjunto de dados de RF provindo do sistema ULTRA-ORS.

### 3.7 Outros elementos do sistema implementado

A estrutura utilizada para realização dos testes experimentais com o sistema implementado é apresentada na Figura 39. Aqui são incluídos os principais itens do sistema de processamento e apresentação de resultados: tela modelo Dell LCD E176FPc, placa ZedBoard com a indicação dos principais módulos, como o SoC ZYNQ-7000, DDR3, UART e interface de pro-

gramação *PROG*, além do *notebook* da marca Acer modelo Aspire E1-572G com processador Intel Core i7+4500U 1.8 GHz, 8 GB de RAM, *Solid-State Drive* (SSD) 480 GB Kingston modelo A400, e placa de vídeo AMD Radeon R7 M265 com 2 GB de memória RAM dedicada.

**Figura 39 – Fotografia do sistema exibindo resultados em tela e com descrição de componentes.**



Fonte: Autoria própria (2024).



## 4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados e discutidos os resultados desta pesquisa, iniciando pelas simulações de validação e comparação entre *scripts* de referência em MATLAB e o modelamento no Simulink com blocos nativos e, posteriormente, o projeto de cada etapa de processamento com blocos do MC. Em seguida, são mostrados os resultados pós implementação em SoC com dados de US do sistema ULTRA-ORS e a descrição dos recursos utilizados da FPGA.

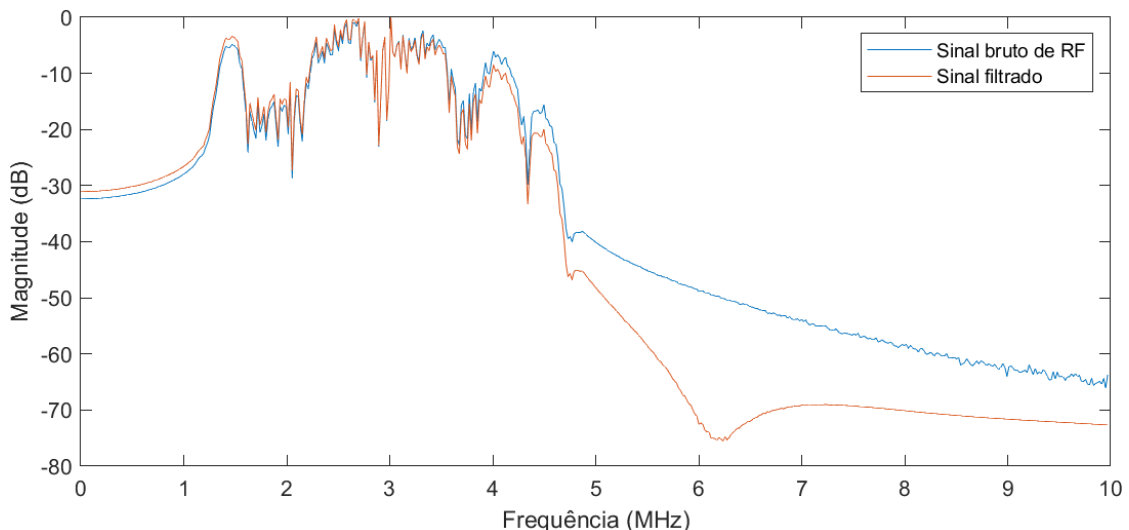
### 4.1 Resultados simulados das etapas de processamento

Nesta seção são apresentados os resultados de simulações de todas as etapas de processamento de sinais de US modeladas e implementadas.

#### 4.1.1 Simulação de sinais do filtro passa-baixa FIR

O projeto do FPB FIR foi validado em Simulink com a obtenção do espectro dos sinais de RF antes e após o filtro. Como exemplo, na Figura 40, são mostrados os sinais correspondentes à avaliação da linha de varredura de índice 61 (linha central, considerando as 121 linhas de varredura geradas). O sinal em azul indica o sinal bruto de RF com o total de 2046 amostras e o sinal vermelho corresponde ao sinal filtrado após essa etapa de processamento.

**Figura 40 – Comparação entre sinal de ultrassom antes (cor azul) e após (cor vermelha) o filtro FIR passa-baixa utilizando o Model Composer no Simulink.**



**Fonte: Autoria própria (2024).**

Pode-se notar que o sinal filtrado do eco captado pelo transdutor de frequência central de 3,2 MHz possui espectro em frequência com perfil similar ao sinal bruto e alteração máxima de 2 dB dentro da banda passante de interesse. Na frequência de banda de parada de 8 MHz,

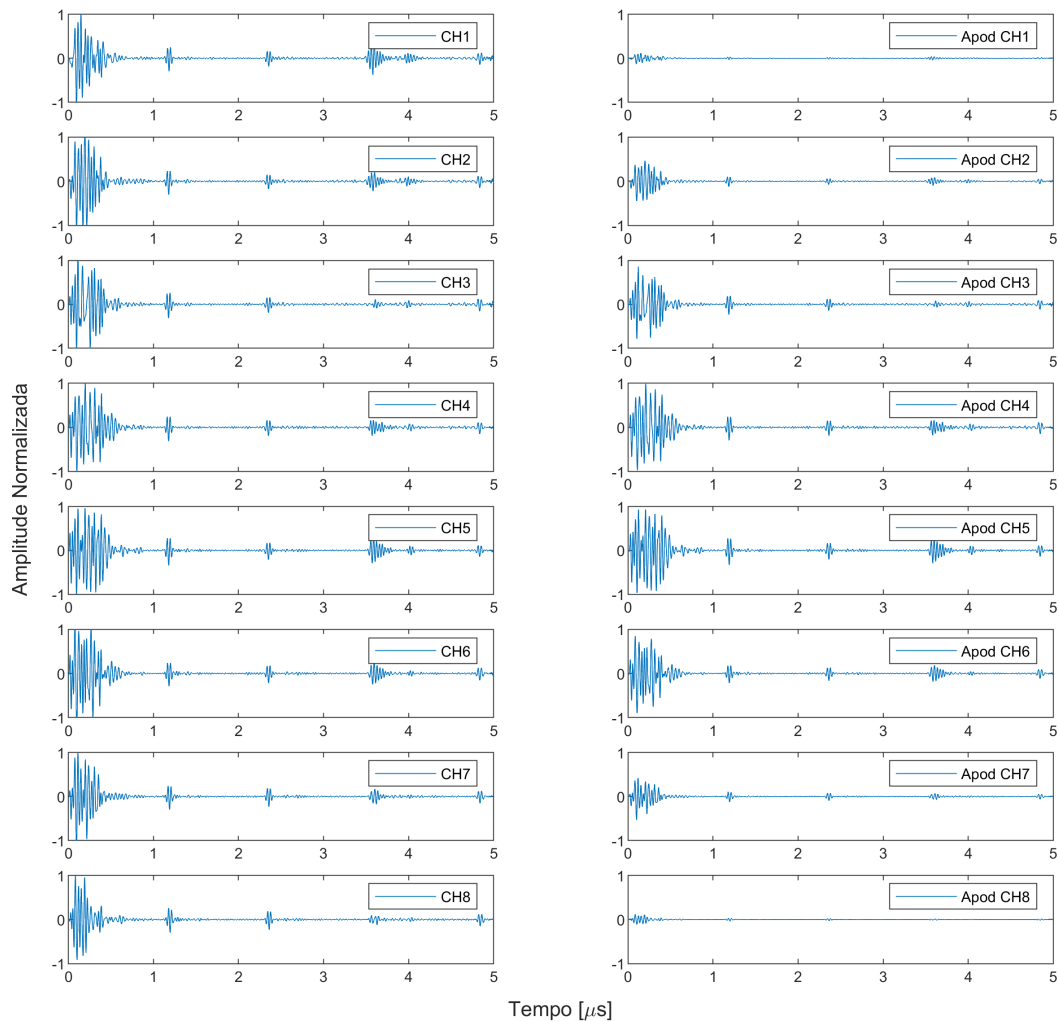
o sinal filtrado possui magnitude de -70 dB, enquanto o sinal bruto apresenta a magnitude de aproximadamente -60 dB.

Esses resultados indicam que o comportamento do filtro condiz com o esperado, reduzindo os componentes de ruído fora do intervalo de interesse.

#### 4.1.2 Simulação de sinais apodizados e com atraso variável

Na Figura 41, são apresentados os resultados combinados das etapas de apodização e atraso de focalização, comparando os 8 sinais da abertura ativa para formação da linha de varredura de índice 61. No lado esquerdo da figura são mostrados os sinais de entrada e no lado direito os sinais de saída de tais etapas. Os resultados simulados condizem com o comportamento esperado de atenuação nos canais das extremidades, que ocorre devido ao perfil de apodização Hanning, que é um dos mais aplicados para essa função (JENSEN, 1996). Porém, o sistema permite que outras funções de janelamento possam ser exploradas, como exemplificado na Figura 6, e discutido por Ferreira (2017) e Oliveira (2020). O atraso de focalização não é visualmente perceptível, pois opera em ciclos de *clock* de 40 MHz da FPGA.

**Figura 41 – Comparação entre a simulação de sinal de ultrassom de 8 canais antes (lado esquerdo) e após (lado direito) a etapa de atraso variável e apodização utilizando o Model Composer.**

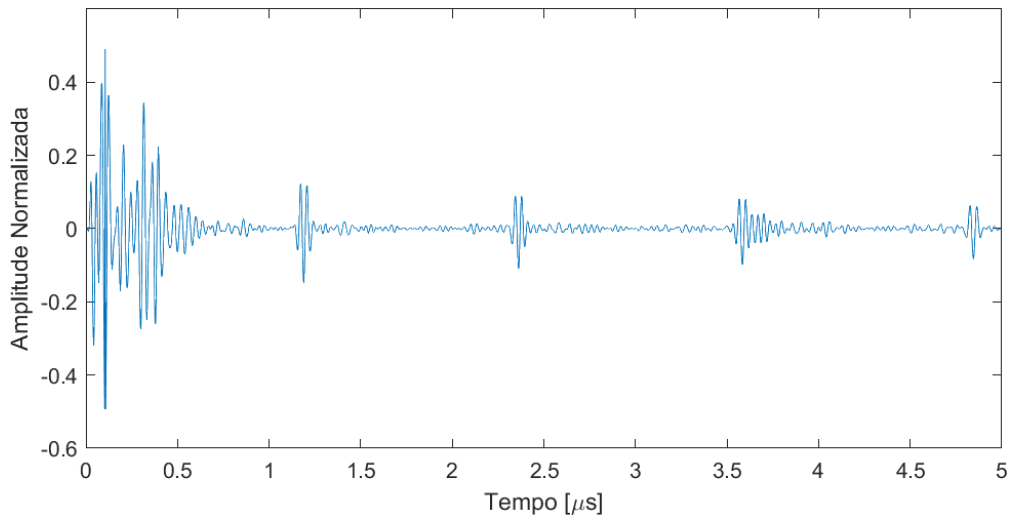


**Fonte: Autoria própria (2024).**

#### 4.1.3 Simulação do sinal resultante do somatório coerente

Após a apodização e o alinhamento dos sinais de eco, o resultado simulado da etapa de somatório coerente é apresentado na Figura 42. Essa imagem corresponde à linha de varredura de índice 61, sendo resultado da soma dos oito canais apresentados na Figura 41, após normalização para adequação da amplitude do sinal.

**Figura 42 – Sinal resultante após o somatório coerente utilizando o Model Composer.**



**Fonte: Autoria própria (2024).**

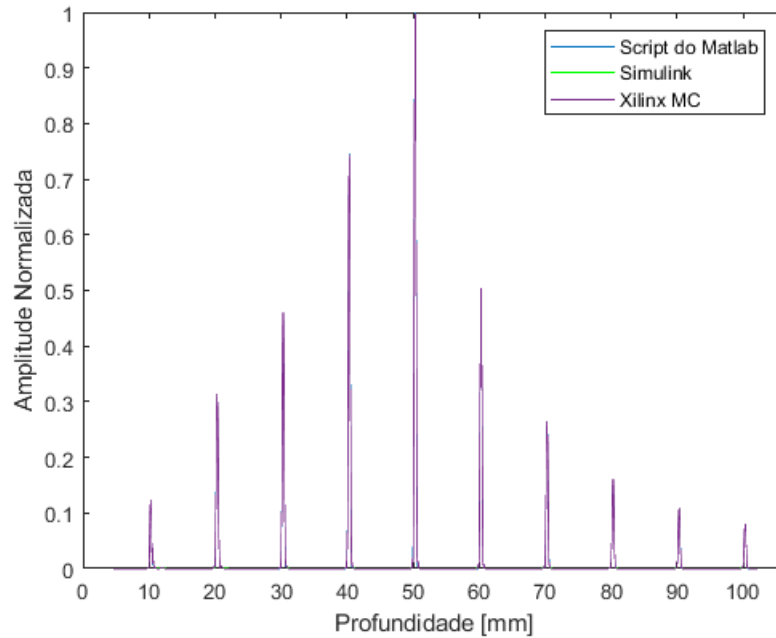
#### 4.1.4 Simulação de sinais da detecção de envoltória e teste do demodulador com Field II

As simulações de validação da detecção de envoltória ocorreram em duas etapas. Na primeira etapa, utilizou-se um conjunto de alvos simulados através do software Field II, que representa um cenário com transdutor linear e alvos precisamente alinhados, conforme o Quadro 2. A segunda simulação, por sua vez, utilizou dados de US práticos adquiridos por transdutor convexo, conforme seção 3.3, na sequência dos módulos já apresentados, para validação completa do sistema de processamento.

A simulação realizada com o Field II, para geração e processamento do *beamforming* digital de sinais de eco, alimentou a etapa de detecção de envoltória. Essa avaliação foi realizada através da comparação dos resultados obtidos por três técnicas: *script* do MATLAB, Simulink com blocos nativos e modelo do Simulink usando blocos do MC para validações iniciais do projeto. Os resultados desta etapa foram apresentados no 2021 IEEE Latin America Ultrasonics Symposium e são exibidos na Figura 43. Nesse resultado, é mostrada a comparação entre envoltórias simuladas para o transdutor linear de 128 canais e abertura de 64 elementos ativos, conforme descrito na subseção 3.2.1 (RUZYK *et al.*, 2021). Percebe-se qualitativamente, através das imagens simuladas, uma excelente concordância entre as envoltórias dos 10 alvos para as três técnicas aplicadas.

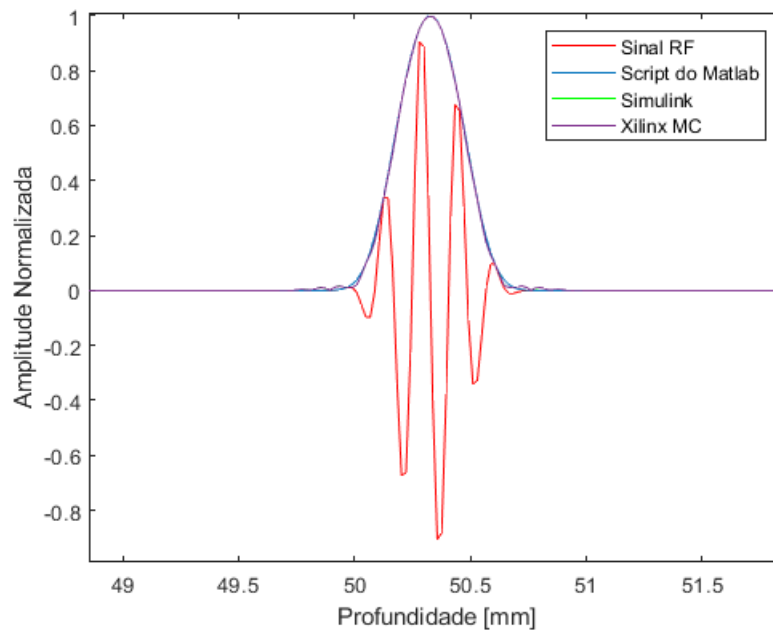
Adicionalmente, a Figura 44, mostra os detalhes das mesmas envoltórias em uma ampliação para melhor visualização, incluindo o sinal de RF de US utilizado como entrada dos algoritmos. Nota-se na imagem que ambos os métodos produzem envoltórias que abarcam o sinal de US, mas que os sinais gerados pelo Simulink e MC apresentam um leve comportamento oscilatório, devido à resposta do filtro FIR. Tais resultados também são comparados de forma quantitativa com as métricas NRMSE e NRSS (Tabela 5). Os valores baixos demonstram que ambos se aproximam da curva de referência gerada por *script* no MATLAB.

**Figura 43 – Comparação de envoltórias centrais geradas por demodulador em MATLAB, Simulink e Model Composer a partir de dados de RF gerados no Field II.**



Fonte: Adaptado de Ruzyk *et al.* (2021).

**Figura 44 – Ampliação da comparação de envoltórias com dados de RF gerados no Field II.**



Fonte: Adaptado de Ruzyk *et al.* (2021).

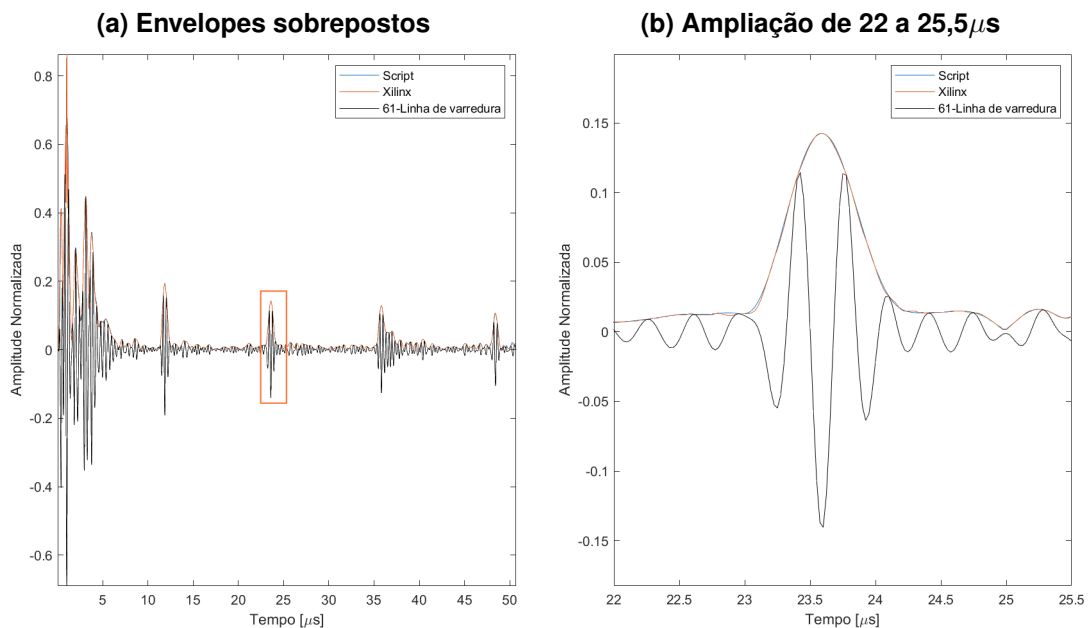
**Tabela 5 – Resultados das métricas de comparação para simulação de detector de envoltória gerada com dados do Field II**

Método	Simulink	Xilinx MC
NRMSE [%]	1,70	1,71
NRSS	$284,10 \times 10^{-6}$	$284,86 \times 10^{-6}$

Fonte: Adaptado de Ruzyk *et al.* (2021).

Após a validação com dados gerados pelo Field II, o mesmo módulo demodulador foi integrado na simulação completa com o MC e submetido aos dados práticos, com resultados apresentados no evento conjunto *17th International Symposium on Medical Information Processing and Analysis (SIPAIM)* e décimo Simpósio de Instrumentação e Imagem Médica (SIIM). Os resultados dessa avaliação para linha de varredura de índice 61 são exibidos na Figura 45, que mostra a comparação entre envoltória de referência simulada através de *script* do MATLAB com a envoltória gerada pelo modelo do MC. Na Tabela 6, são mostrados os resultados quantitativos de NRMSE e NRSS para o projeto usando MC em comparação com o padrão gerado por *script* do MATLAB (RUZYK *et al.*, 2021).

**Figura 45 – Comparação de envoltórias simuladas e alimentadas com dados empíricos em: (a) MATLAB e Model Composer, e (b) sua ampliação entre 22 e 25,5  $\mu$ s.**



Fonte: Adaptado de Ruzyk *et al.* (2021).

**Tabela 6 – Métricas NRMSE e NRSS para envoltórias geradas a partir de dados empíricos do ULTRA-ORS sem compressão logarítmica.**

Métrica	Envoltória número 61	Máximo	Mínimo	Média	Mediana	Variância	Desvio Padrão
NRSS	$1,43 \times 10^{-2}$	$2,81 \times 10^{-2}$	$5,37 \times 10^{-4}$	$5,16 \times 10^{-3}$	$3,12 \times 10^{-3}$	$2,91 \times 10^{-5}$	$5,40 \times 10^{-3}$
NRMSE [%]	13,12	17,37	2,60	6,95	6,30	12,96	3,60

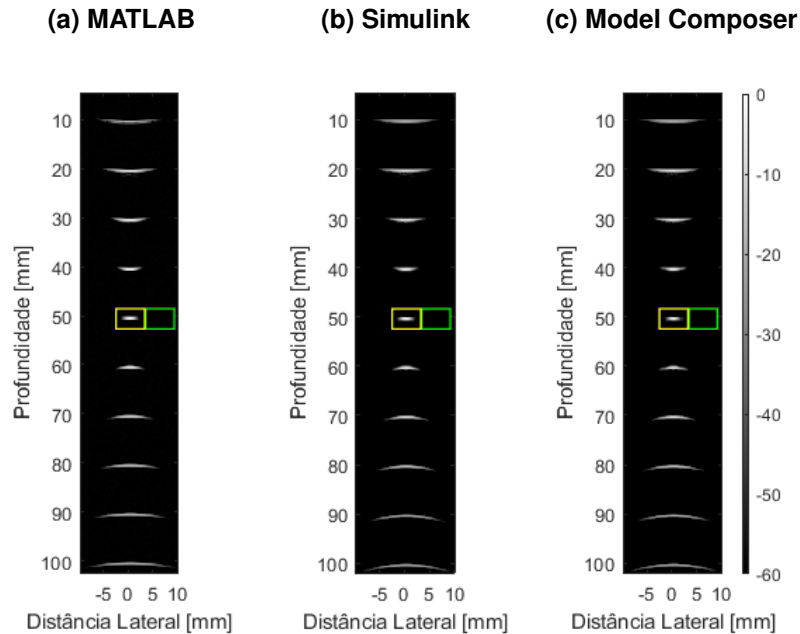
**Fonte: Adaptado de Ruzyk *et al.* (2021).**

O trabalho de Assef *et al.* (2018) que implementou o mesmo módulo em FPGA obteve um resultado de NRMSE para a mesma configuração de filtro de Transformada de Hilbert (HT) de 0,42%, porém utilizando o módulo isolado, sem somar erros de outras etapas de processamento. Outros trabalhos com dados simulados que utilizaram todo o *beamforming* para gerar os dados para a HT apresentaram valores de 3,04% (OLIVEIRA, 2020) e 6,19% (FERREIRA, 2017). Isso demonstra que o NRMSE obtido nas simulações está dentro do esperado com média de 6,95% e desvio padrão de 3,60%, apesar da grande diferença entre os valores totais máximos e mínimos, causados por regiões específicas com ruído elevado (CHANG; YEN; SHUNG, 2007).

#### 4.1.5 Simulação de sinais da Compressão logarítmica

A compressão logarítmica e a escolha de uma faixa de intensidade do brilho da imagem permitem a exibição das linhas de varredura em formato adequado para observação de detalhes através do olho humano. Para o caso da simulação alimentada com os dados gerados no Field II, a utilização de transdutor linear permitiu que se gerasse a imagem final já após a etapa de compressão logarítmica, exibida na Figura 46, com faixa dinâmica de -60 dB. Essa imagem representa uma comparação de resultados entre um *frame* de cada uma das 3 simulações de processamento de envoltória: em Simulink e MC, que fazem a aproximação da Transformada de Hilbert com filtros FIR, e o padrão ouro, em MATLAB, que calcula o valor absoluto da Transformada de Hilbert.

**Figura 46 – Comparação entre imagens reconstruídas em Modo B com compressão logarítmica a partir de dados simulados no Field II em: (a) *script* do MATLAB, (b) simulação em Simulink e (c) simulação com Model Composer. Os retângulos nas cores amarelo e verde representam a região de interesse e a região de fundo, respectivamente, para as análises de contraste.**



**Fonte: Adaptado de Ruzyk *et al.* (2021).**

Na Tabela 7, são apresentados os resultados das métricas de contraste de CR e CNR para as regiões retangulares de cores amarelo e verde, que representam a região de interesse e a região de fundo, respectivamente. Os valores obtidos apresentam erros inferiores a 7%, o que indica uma proximidade dos modelos em relação às imagens finais.

**Tabela 7 – Resultados das métricas de contraste para imagens geradas a partir de sinais de RF no Field II e processados no MATLAB, Simulink e Model Composer.**

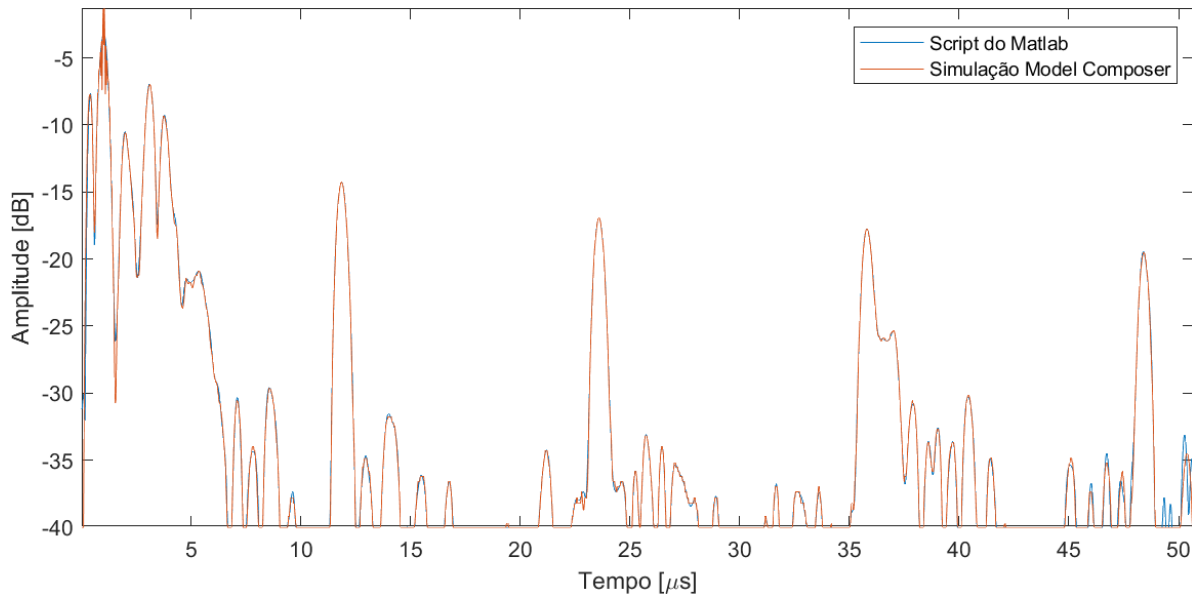
Método	CNR	Erro CNR [%]	CR	Erro CR [%]
<i>Script</i> do MATLAB (referência)	13,58	-	$6,56 \times 10^{-2}$	-
Simulink	14,51	6,84	$7,01 \times 10^{-2}$	6,86
Xilinx MC	14,49	6,7	$6,99 \times 10^{-2}$	6,55

**Fonte: Adaptado de Ruzyk *et al.* (2021).**



Outra simulação foi realizada considerando um *frame* dos dados de RF empíricos do sistema ULTRA-ORS, de 121 amostragens de 8 canais e abertura convexa, e utilizando os demais módulos de processamento propostos e implementados no MC, incluindo a compressão logarítmica. Na Figura 47, é apresentado o sinal resultante da linha de varredura central, índice 61, que é comparada com a simulação do MATLAB. Resultados quantitativos de métricas de proximidade são apresentados na Tabela 8. O valor de 8,18% para o NRMSE da linha central e o NRSS máximo de  $5,15 \times 10^{-4}$  demonstram que as envoltórias estão em excelente concordância, tanto pelo erro ser menor que 10% quanto pelo NRSS ser muito menor que 1 (ASSEF *et al.*, 2018).

**Figura 47 – Comparação de envoltórias com faixa dinâmica de -40 dB entre simulações em MATLAB e Model Composer com dados empíricos do ULTRA-ORS.**



**Fonte: Autoria própria (2024).**

**Tabela 8 – Métricas NRSS e NRMSE comparando envoltórias com compressão logarítmica geradas em MATLAB e Model Composer a partir de dados empíricos do ULTRA-ORS.**

Métrica	Envoltória número 61	Máximo	Mínimo	Média	Mediana	Variância	Desvio Padrão
NRSS	$4,14 \times 10^{-4}$	$5,15 \times 10^{-4}$	$1,65 \times 10^{-5}$	$2,47 \times 10^{-4}$	$2,46 \times 10^{-4}$	$1,08 \times 10^{-8}$	$1,04 \times 10^{-4}$
NRMSE [%]	8,18	11,12	1,59	7,46	7,43	3,49	1,87

**Fonte: Adaptado de Ruzyk *et al.* (2021).**

A imagem resultante, formada por 121 envoltórias com compressão logarítmica de -40 dB, é mostrada na Figura 48. A imagem é apresentada em coordenadas polares e ainda não está em sua forma final, pois foi capturada com o transdutor convexo. Com isso, é necessária uma etapa extra de processamento, apresentada na subseção 4.1.6. As áreas identificadas com retângulos amarelo e vermelho para alvo e plano de fundo, respectivamente, tiveram tam-

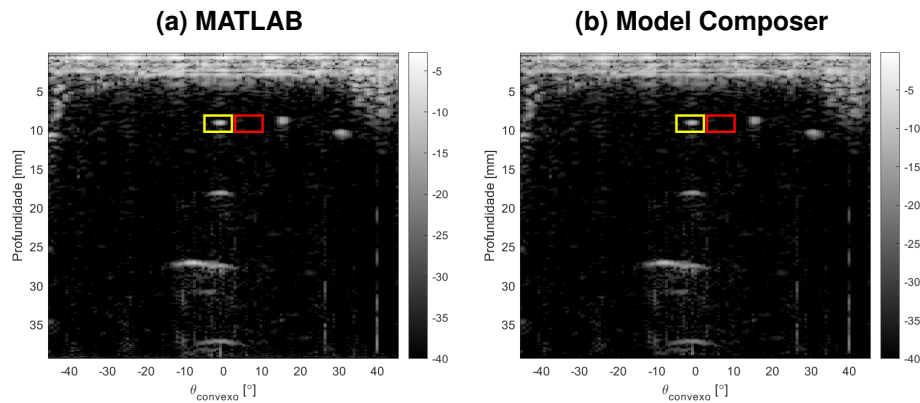
bém as métricas de CR e CNR computadas, conforme a Tabela 9, que apresenta erros da ordem de 1%.

**Tabela 9 – Resultados das métricas de contraste para imagens com compressão logarítmica geradas em simulações do MATLAB e Model Composer a partir de dados empíricos capturados com transdutor convexo (sem conversão de varredura).**

Método	CNR	Erro [%]	CR	Erro [%]
<i>Script</i> do MATLAB	2,18	-	$9,15 \times 10^{-2}$	-
Xilinx MC	2,16	0,81	$9,06 \times 10^{-2}$	1,01

Fonte: Adaptado de Ruzyk *et al.* (2021).

**Figura 48 – Comparação entre imagens reconstruídas em Modo B com compressão logarítmica e coordenadas polares a partir de dados gerados pelo ULTRA-ORS e processados em: (a) *script* do MATLAB e (b) simulação do Model Composer. Os retângulos nas cores amarelo e vermelho representam a região de interesse e a região de fundo, respectivamente, para as análises de contraste.**



Fonte: Adaptado de Ruzyk *et al.* (2022).

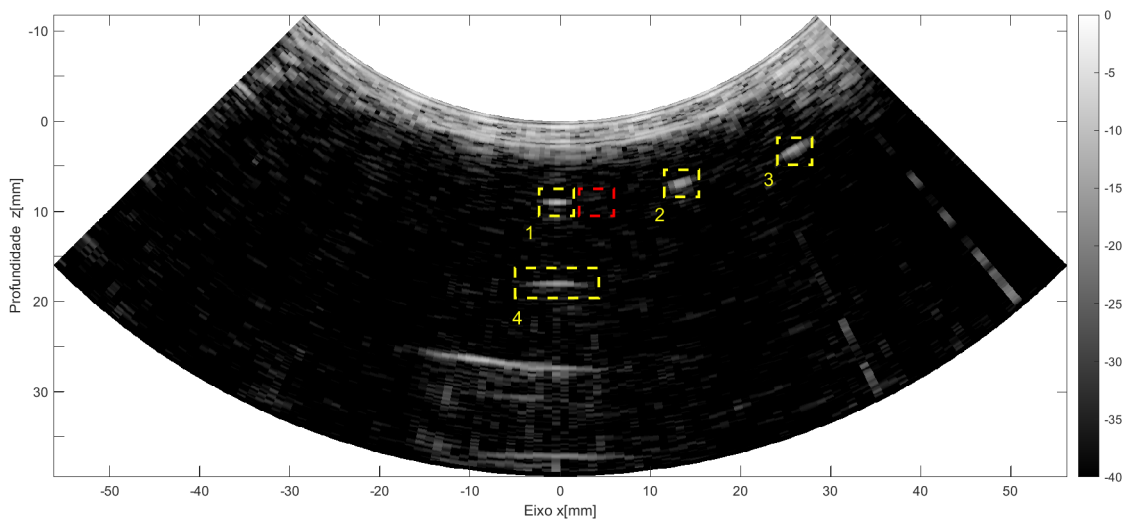
#### 4.1.6 Simulação da Conversão de varredura

A correta exibição de imagens capturadas através de transdutores convexos requer a etapa de conversão de varredura. Assim, as simulações utilizando dados empíricos da plataforma ULTRA-ORS também necessitam dessa etapa, vide a subseção 2.3.5. A imagem processada da Figura 48, utilizando o MC até a etapa de compressão logarítmica, é então submetida à conversão de varredura com o auxílio de funções do MATLAB, conforme indicado na subseção 3.5.10.

Na Figura 49, é apresentado o resultado da conversão de varredura, que apresenta os alvos em conformidade com o corpo de prova utilizado, ver Figura 11. Também são indicadas quatro regiões de interesse denotadas por retângulos em amarelo, e um em vermelho selecionando a região de fundo. As métricas CR e CNR foram calculadas para cada uma dessas

regiões e seus resultados são exibidos na Tabela 10, em conjunto com os erros em comparação com os resultados do processamento completo em *script* do MATLAB. As métricas apontam que há uma leve degradação dos resultados do MC, com erros menores que 10% entre as duas simulações. Uma análise mais aprofundada é feita na subseção 4.2.3, que compara estes com resultados práticos.

**Figura 49 – Imagem simulada resultante da etapa de conversão de varredura em MATLAB para dados processados em Model Composer. Quatro regiões de interesse são exibidas com retângulos na cor amarelo e a região de fundo na cor vermelho para as análises de contraste.**



Fonte: Autoria própria (2024).

**Tabela 10 – Resultados das métricas de contraste CR e CNR de imagens com conversão de varredura em MATLAB para dados providos do *script* de referência e simulação em Model Composer.**

Alvo	Métrica	Referência (MATLAB)	Simulação MC	Erro [%]
1	CR	$10,08 \times 10^{-2}$	$9,26 \times 10^{-2}$	8,17
	CNR	1,98	1,97	0,32
2	CR	$14,34 \times 10^{-2}$	$13,12 \times 10^{-2}$	8,50
	CNR	2,82	2,80	0,68
3	CR	$15,78 \times 10^{-2}$	$14,43 \times 10^{-2}$	8,55
	CNR	3,10	3,08	0,73
4	CR	$6,13 \times 10^{-2}$	$5,64 \times 10^{-2}$	7,99
	CNR	1,20	1,20	0,13

Fonte: Autoria própria (2023).

## 4.2 Resultados empíricos pós implementação em SoC

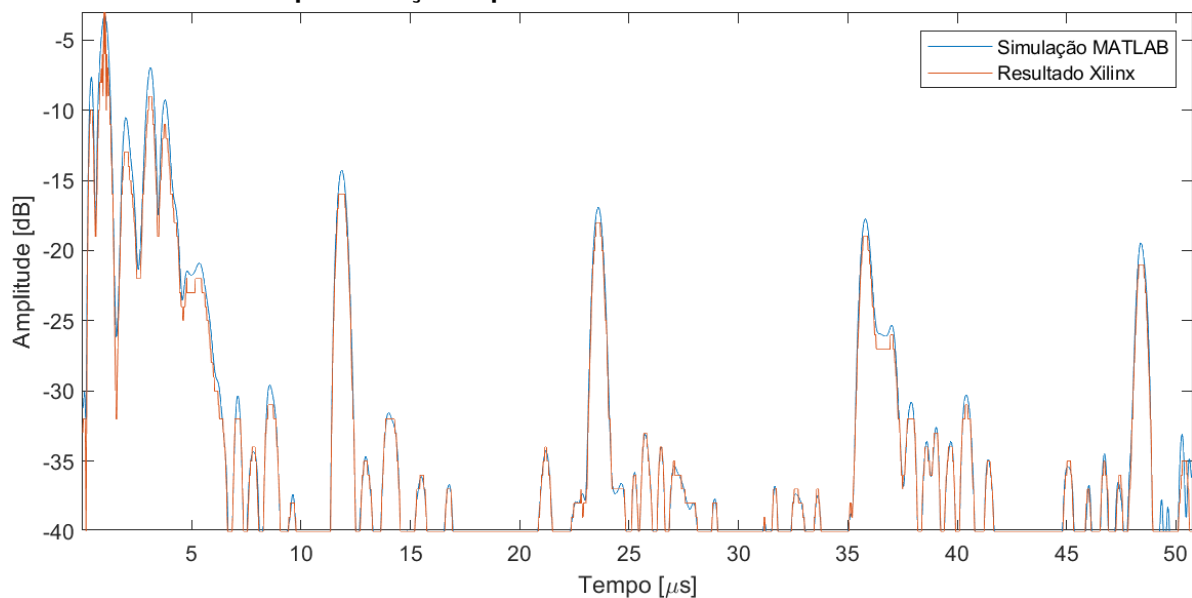
Nesta seção são apresentados os resultados da implementação na ZedBoard, iniciando pela análise em separado das etapas com processamento em FPGA e no ARM embarcado, que

são o *beamforming* e a conversão de varredura, respectivamente. Após caracterização desses elementos, uma análise global é feita, incluindo todo o processamento e análise de tempos de processamento e recursos utilizados.

#### 4.2.1 Comparação entre envoltórias com compressão logarítmica

As envoltórias centrais adquiridas imediatamente após a etapa de compressão logarítmica foram analisadas para os resultados de simulação em MATLAB e implementação experimental na placa ZedBoard. A Figura 50, fornece os resultados das envoltórias de índice 61 para comparação qualitativa. Tais valores foram adquiridos após a compressão logarítmica, porém antes da etapa final de conversão de varredura. Percebe-se que, de forma geral, a envoltória empírica acompanha o formato da referência, porém apresenta artefatos caracterizados por não se atingir totalmente o valor dos picos dos alvos, mas que são de magnitude pequena, não influenciando no comportamento geral.

**Figura 50 – Comparação de envoltórias com faixa dinâmica de -40 dB entre a simulação em MATLAB e a implementação experimental na ZedBoard.**



**Fonte: Autoria própria (2024).**

As envoltórias podem ser comparadas quantitativamente através das métricas NRSS e NRMSE e os resultados são exibidos na Tabela 11, que engloba também resultados estatísticos caracterizando todo o conjunto de 121 envoltórias. Além disso, a linha de varredura de índice 61 foi evidenciada, pois é a mesma exibida na Figura 50. Os resultados apontam que as duas envoltórias apresentam similaridade, com NRSS na ordem de  $10^{-4}$ , indicando erros baixos entre amostras, e NRMSE com mediana de 8,57%, caracterizando que a imagem de forma geral se encontra em uma faixa aceitável, e apenas as amostras centrais, evidenciadas pelos maiores picos de intensidade se aproximam do NRMSE máximo de 10,77%. Isto indica que a imple-

mentação da ZedBoard apresenta desempenho adequado. Como comparação, Oliveira (2020) utilizou uma FPGA Stratix IV e obteve NRMSE de 8,31% para processamento relativamente semelhante. Além disso, estar próximo da faixa de 10% de NRMSE não resulta em degradação significativa das características da imagem em Modo B (CHANG; YEN; SHUNG, 2007). Uma forma de melhorar o valor de NRMSE é diminuir a faixa dinâmica, como apresentado por Ferreira (2017), que obteve NRMSE de 0,45% após diminuir a faixa dinâmica para -30 dB ou retirar os valores de campo próximo da região de interesse (OLIVEIRA *et al.*, 2022).

**Tabela 11 – Métricas de comparação entre envoltórias com compressão logarítmica geradas pelo *beamformer* implementado na ZedBoard e por *script* do MATLAB antes da conversão de varredura.**

Métrica	Envoltória número 61	Máximo	Mínimo	Média	Mediana	Variância	Desvio Padrão
NRSS	$6,85 \times 10^{-4}$	$7,60 \times 10^{-4}$	$1,56 \times 10^{-4}$	$3,29 \times 10^{-4}$	$3,18 \times 10^{-4}$	$1,14 \times 10^{-8}$	$1,07 \times 10^{-4}$
NRMSE [%]	10,58	10,77	6,78	8,62	8,57	0,67	0,82

Fonte: Autoria própria (2023).

#### 4.2.2 Resultados isolados da conversão de varredura na ZedBoard

Testes isolados da conversão de varredura foram necessários para caracterizar o erro gerado pelo modelo, quando implementado na forma de SW embarcado no sistema (RUZYK *et al.*, 2022). Os dados utilizados foram os mesmos da subseção 4.1.5, ou seja, dados empíricos da plataforma ULTRA-ORS com *beamforming digital* simulado utilizando o MC até a etapa de compressão logarítmica. Assim, a conversão de varredura foi testada empiricamente, de forma isolada no ARM, que faz parte do ZYNQ, e comparada com a conversão utilizando funções do MATLAB, conforme subseção 3.5.10. Na Figura 51, é mostrada a imagem gerada pela ZedBoard após a conversão de varredura.

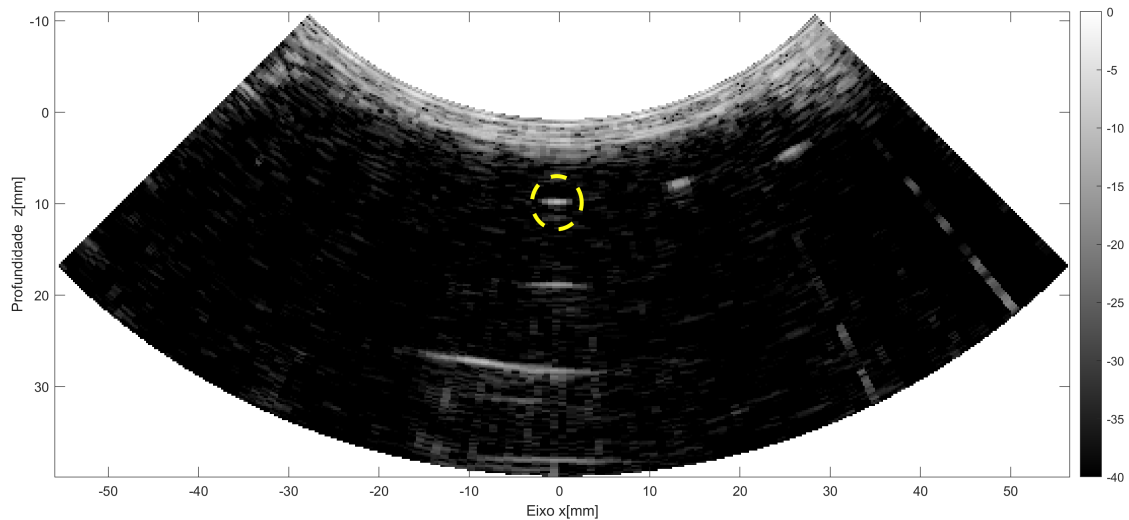
Os resultados quantitativos das métricas de contraste da Figura 51 são apresentados na Tabela 12, que mostra os valores computados de CR e CNR para a região do alvo 1 da Figura 49.

**Tabela 12 – Resultados das métricas de contraste para a etapa isolada de conversão de varredura, comparando implementação na ZedBoard com o *script* do MATLAB.**

Método	CNR	Erro [%]	CR	Erro [%]
<i>Script</i> do MATLAB	1,98	-	$9,24 \times 10^{-2}$	-
ZedBoard	1,91	3,38	$9,00 \times 10^{-2}$	2,62

Fonte: Adaptado de Ruzyk *et al.* (2022).

**Figura 51 – Imagem resultante da conversão de varredura realizada na plataforma ZedBoard utilizando dados simulados com Model Composer.**



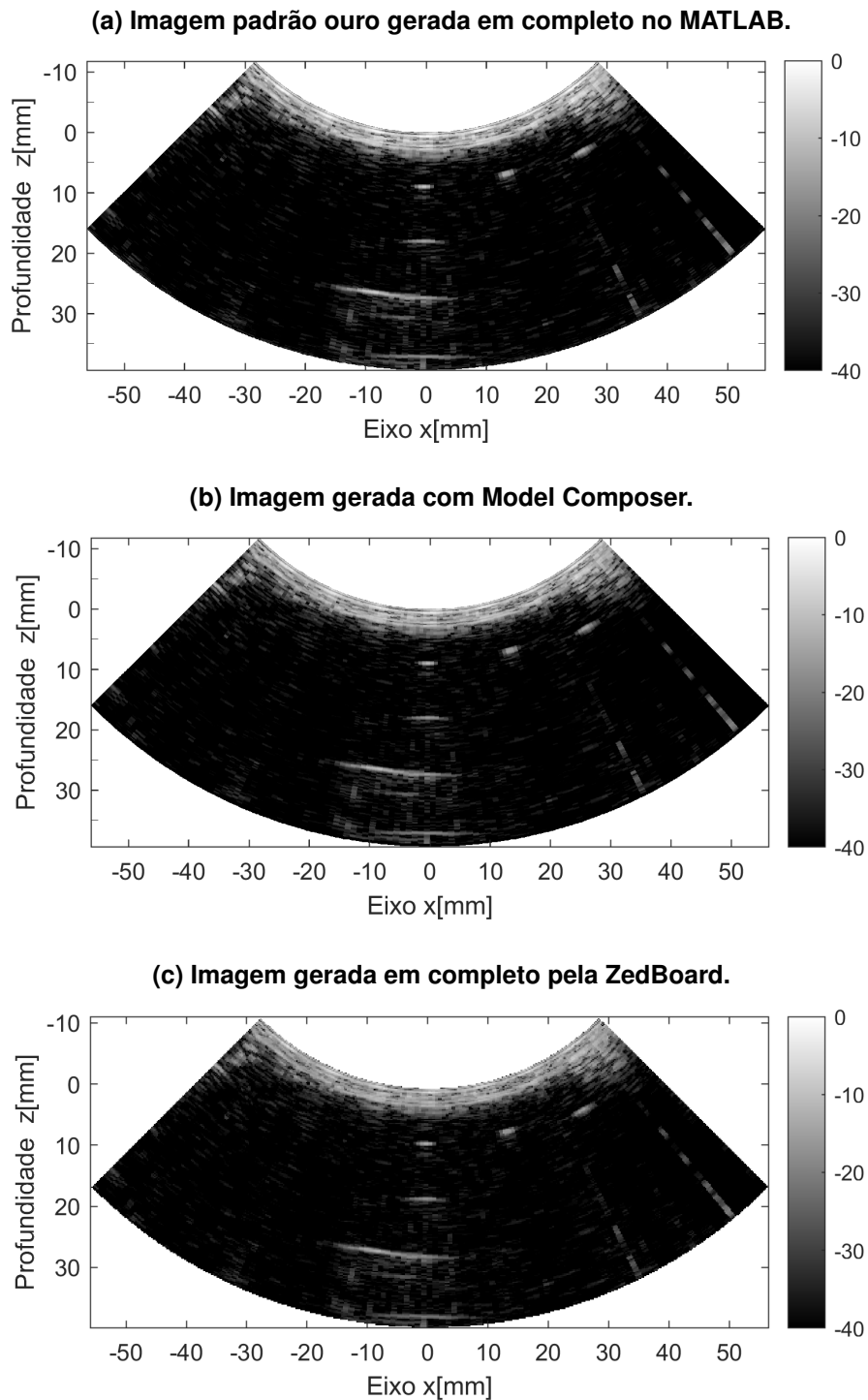
**Fonte: Adaptado de Ruzyk *et al.* (2022).**

Tanto a análise visual como a de erros mostram que a imagem resultante mantém qualidade em relação ao padrão do MATLAB, apesar de apresentar uma perda de resolução visível nas bordas superiores e inferiores. Os erros com valor máximo de 3,38% indicam que a perda de resolução é mínima. Assim, a etapa de conversão de varredura foi caracterizada de forma isolada, e mesmo utilizando uma técnica de interpolação inferior à do MATLAB, e se apresentando em menor resolução,  $320 \times 500$  pixels, ainda é adequada para o processamento, pois não incluiu erros acentuados que geralmente provocam falsos artefatos na imagem final.

#### 4.2.3 Resultado da implementação total na ZedBoard e comparação com simulações

Além de disponibilizar a imagem resultante no display VGA, o dispositivo enviou o resultado, uma matriz em escala de cinza com resolução de  $320 \times 500$  pixels, para o computador através da interface UART, o que demandou a adequação dos valores à resolução de 8 bits *unsigned* (0-255). Após o recebimento, os dados são reprocessados para formar uma imagem com faixa dinâmica de -40 dB. Tal imagem resultante da plataforma Xilinx é exibida em conjunto com a imagem referência, ou padrão ouro, do MATLAB e com a imagem simulada a partir de dados do MC na Figura 52. De forma geral, há divergências visíveis apenas entre a simulação e os resultados práticos. O resultado da ZedBoard apresenta os mesmos artefatos resultantes da perda de resolução discutidos na subseção 4.2.2.

Figura 52 – Imagens resultantes da conversão de varredura em: (a) Simulação completa em MATLAB. (b) Simulação no Model Composer com conversão de varredura no MATLAB. (c) Resultados da ZedBoard.



Fonte: Autoria própria (2024).

A análise é aprofundada pela utilização de métricas de contraste nas mesmas regiões de interesse exploradas na Figura 49, e os resultados são também comparados com os obtidos na subseção 4.1.6. Os valores de CR e CNR foram calculados para cada uma dessas regiões

e seus resultados são exibidos na Tabela 13, a qual também apresenta os valores de erro para comparações dos resultados práticos com as duas simulações.

**Tabela 13 – Resultados das métricas de contraste para a imagem com conversão de varredura gerada na plataforma ZedBoard.**

Alvo	Métrica	ZedBoard	MATLAB	Simulação MC	Erro ZedBoard [%] (Ref. MATLAB)	Erro ZedBoard [%] (Ref. MC)
1	CR	$8,33 \times 10^{-2}$	$10,08 \times 10^{-2}$	$9,26 \times 10^{-2}$	17,34	9,98
	CNR	1,84	1,98	1,97	6,94	6,64
2	CR	$12,48 \times 10^{-2}$	$14,34 \times 10^{-2}$	$13,12 \times 10^{-2}$	13,00	4,92
	CNR	2,76	2,82	2,80	2,05	1,38
3	CR	$13,02 \times 10^{-2}$	$15,78 \times 10^{-2}$	$14,43 \times 10^{-2}$	17,52	9,81
	CNR	2,88	3,10	3,08	7,14	6,46
4	CR	$5,38 \times 10^{-2}$	$6,13 \times 10^{-2}$	$5,64 \times 10^{-2}$	12,13	4,50
	CNR	$11,90 \times 10^{-1}$	$12,03 \times 10^{-1}$	$12,01 \times 10^{-1}$	1,07	0,94

**Fonte: Autoria própria (2023).**

De forma geral, a implementação na ZedBoard apresentou resultados com leve degradação da imagem, observados também nas métricas quantitativas, sobretudo no CR, cujo erro em relação ao MATLAB passou de 10% em todas as amostras, chegando a 17,52% no alvo 3. O CNR apresentou erros menores, de no máximo 7,14% em relação ao MATLAB e se manteve próximo dos valores da simulação do MC. Isso indica que mesmo perdendo resolução, conforme os valores de CR, os alvos ainda se evidenciam em relação ao ruído, indicado pelo CNR. Assim, otimizações de CR podem focar em aumentar a matriz resultante, que é pequena comparada com a quantidade de pontos que o MATLAB fornece, e otimizações para melhorar o CNR podem focar no design do *beamformer* e em técnicas de processamento, além de aumento de precisão interna. O algoritmo de conversão de varredura pode contribuir também para o aumento do CNR. Porém, conforme apresentado na Tabela 10, esta etapa isolada não contribuiu significativamente para o aumento do CNR e é dependente da resolução, notada no decréscimo do CR.

Oliveira (2020) implementou as etapas de processamento similares a esta pesquisa para a mesma abertura e transdutor, porém com a conversão de varredura realizada através de *script* do MATLAB e faixa dinâmica menor. Em comparação com uma faixa dinâmica de -30 dB, o autor obteve erros de CR na faixa de 3,07% a 4,57% e de CNR na faixa de 0,92% a 1,56%, com valores absolutos de CR na faixa de 0,09 a 0,17 e CNR entre 0,74% e 1,89%. Estes valores se assemelham aos resultados de simulações do MC mostrados na subseção 4.1.6, que indicou erro máximo de CR e CNR de 8,55% e 0,73 %, respectivamente. Esses resultados evidenciam que o *beamformer* projetado se assemelha a este em desempenho, mas conforme esperado, a implementação total em SoPC gerou uma limitação no tamanho da matriz imagem resultante e de estratégia de conversão de varredura, deteriorando os resultados.



### 4.3 Recursos utilizados na implementação em SoPC

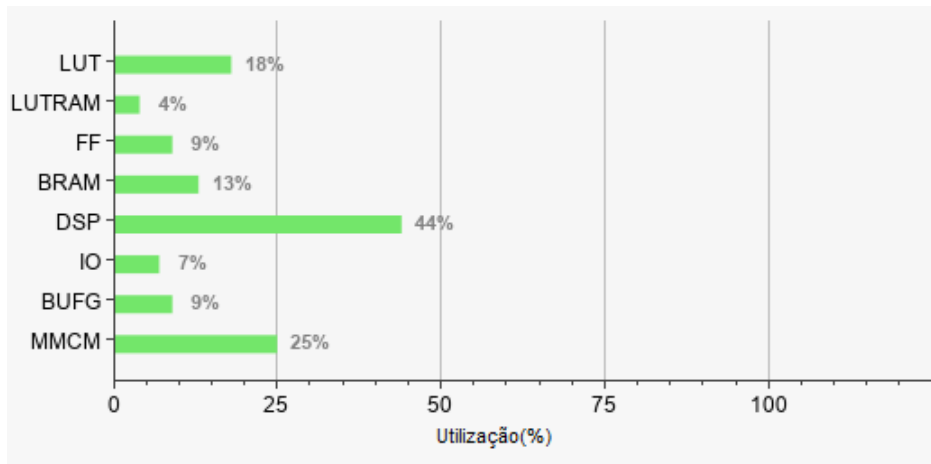
O sistema proposto foi implementado em um SoPC da família Zynq-7000 e ocupou menos da metade dos recursos da mesma, quantificados na Tabela 14, em sua maioria não chegando a 25%, conforme o gráfico da Figura 53. Detalhes sobre os tipos de recursos podem ser consultados em XILINX (2018).

**Tabela 14 – Utilização de recursos pós implementação no SoPC.**

Tipo de recurso	Total disponível	% Utilização total	% Utilização <i>beamformer</i> em FPGA
Slice LUTs	53200	18,08	10,18
Slice Registers	106400	9,29	3,91
Slice	13300	28,13	14,08
LUT as Logic	53200	16,76	9,39
LUT as Memory	17400	4,04	2,44
Block RAM Tile	140	12,86	8,93
DSPs	220	44,09	44,09
Bonded IOB	200	7	0
Bonded IPADs	130	0	0
Bonded IOPADs	130	100	0
BUFGCTRL	32	9,38	0
MMCME2_ADV	4	25	0

Fonte: Autoria própria (2023).

**Figura 53 – Gráfico de uso de recursos do SoC Zynq-7000.**



Fonte: Autoria própria (2024).

Em comparação com outros trabalhos que utilizam a ZedBoard, há a possibilidade de inclusão de outras etapas de processamento, como interpolação, decimação, controle do processamento de *front-end*, inclusão de compressão de dados para transmissão, e outras formas de aquisição de dados como em 3 dimensões (WANG; SANIIE, 2016; GOVINDAN *et al.*, 2015; BHARATH *et al.*, 2016).

#### 4.3.1 Tempos de processamento da ZedBoard

Testes foram executados para determinar o desempenho do sistema quando implementado na plataforma ZedBoard. Na Tabela 15, são indicados os diversos testes realizados. A análise se inicia com a contagem do tempo do processamento das 121 linhas de varredura e de sua exibição no display VGA. Para as mesmas etapas, outra aquisição de tempo foi tomada sem trechos de programação que incluem declarações e inicializações. Outro teste foi realizado para computar o tempo necessário para executar apenas o *beamforming* digital, que envolve as etapas de inserir 8 linhas de varredura (*scanlines*), aguardar o processamento em HW e ler a saída de dados processados, por 121 vezes. Foi também aferido o tempo necessário para o processador executar apenas a conversão de varredura, com e sem trechos de inicializações. O tempo da inicialização do VGA e exibição da imagem foi determinado. Obteve-se também o tempo de transferência utilizado pela UART. Finalmente, determinou-se o tempo total de processamento, desde a inicialização do sistema até a total transferência de dados para o computador.

**Tabela 15 – Resultados de tempo de processamento na plataforma ZedBoard.**

Testes executados	Tempo (s)
Processamento e exibição em VGA	$359,52 \times 10^{-3}$
Processamento e exibição em VGA após inicializações	$359,47 \times 10^{-3}$
Carregamento das memórias FIFO, processamento em FPGA, leitura saída FIFO para as 121 linhas de varredura	$195,21 \times 10^{-3}$
Conversão de varredura completa em software	$164,24 \times 10^{-3}$
Conversão de varredura sem cálculo da matriz de índices	$26,63 \times 10^{-3}$
Inicialização dos módulos VGA e exibição da imagem	$31,41 \times 10^{-3}$
Transferência de dados ( $121 \times 2046$ bytes) por UART	80,03
Processamento total com transferência de resultados	80,42

**Fonte: Autoria própria (2023).**

A partir da análise da Tabela 15, pode-se constatar que o tempo total para a produção de um quadro da imagem de US na tela com VGA é de aproximadamente 0,36 s, com ou sem inicializações, o que resulta em aproximadamente em 2,7 quadros por segundo. A limitação de velocidade por quadro para o caso de 121 linhas de varredura em um sistema operando em meio biológico, considerando a velocidade de propagação do som de 1540 m/s e utilizando a Equação 5, é de 25 quadros por segundo, e é limitado pelo tempo que a onda demora para se propagar e retornar ao transdutor (HEDRICK; HYKES; STARCHMAN, 2005).

Apesar de o sistema não processar os dados na mesma taxa em que foram adquiridos, a razão de FPS já permite a análise de fenômenos que se enquadrem no intervalo de tempo fornecido. Em comparação, há ocasiões como análise cardíaca de animais, em que se necessita de taxas de atualização maiores de até 100 quadros por segundo e há implementações que chegam a utilizar 130, dependendo da aplicação (JONVEAUX *et al.*, 2022).

O tempo de processamento sem a conversão de varredura, que inclui apenas o carregamento de dados na FPGA, processamento, e a leitura dos resultados para as 121 amostras de

um quadro é de 0,20 s, chegando a 5 FPS. Essa etapa de carregamento e leitura de dados com o *beamforming digital* pode ser comparada com trabalhos anteriores, como o de Oliveira (2020) que utilizou a plataforma DE4-230 da Terasic com FPGA Intel para leitura dos dados de RF através da Ethernet e para todo o processamento, leitura e gravação de dados, levando 4 segundos para processar os mesmos 121 canais com 2046 amostras. Este tempo é em maior parte limitado pela transferência dos dados para o *beamformer* e plataformas de desenvolvimento, como a ZedBoard, podem fazer também a aquisição direta do *Analog to Digital Converter* (ADC), ou seja, atuando no *beamforming* de TX e RX (BHARATH *et al.*, 2018; WANG; SANIIE, 2016; GOVINDAN *et al.*, 2015).

A velocidade de processamento do *beamformer* DAS é adequada e se encontra em consonância com um outro trabalho semelhante do Grupo do LUS, mas otimizações podem ser feitas nos deslocamentos de dados entre a LP e PS para se buscar um processamento em tempo real. Ferreira (2017) analisou os atrasos de processamento em FPGA com a placa DE4-230, com um *beamformer* de arquitetura semelhante à implementada neste trabalho, levando 33 ciclos de *clock* à 40 MHz para fazer o processamento de uma amostra, ou 6,19 ms para processar todo o conjunto de 121 canais de 2046 amostras. Por outro lado, os resultados obtidos através da ZedBoard se encontram na faixa de 195 ms, obtido na implementação proposta. Isso se deve a etapas adicionais de leituras e escritas em memória, tanto da RAM para a FIFO de interface, ver subseção 3.5.7, quanto na escrita em cada uma das FIFOs paralelas, ver subseção 3.5.8. Essas etapas adicionais para lidar com os dados correspondem por ao menos 2046 ciclos de *clock* extra para cada carregamento de cada canal, correspondendo a cerca de 49 ms gastos apenas para deslocamento até as FIFOs paralelas, para o total de 121 amostragens de 8 canais. Outras estratégias como o carregamento direto da DDR para as fifos paralelas podem economizar esses 49 ms, e uma arquitetura diretamente interligada com o ADC, para aquisição e processamento em tempo real, não necessitaria tais FIFOs e reduziria o tempo total de processamento para menos de 100 ms.

Na conversão de varredura, que é feita no PS, o maior tempo do processamento foi exigido para o cálculo dos coeficientes índice da matriz final, totalizando 164,24 ms. Percebe-se que, após esse cálculo, o tempo de geração da matriz convertida é de apenas 26,63 ms, e este é o tempo necessário para converter as coordenadas dos próximos quadros. Outros trabalhos chegaram a resultados de tempo de processamento semelhantes ou implementaram parte do processamento em HW para agilizar os cálculos (SIKDAR *et al.*, 2001; KASSEM; HAMAD; SAWAN, 2006; CHANG; YEN; SHUNG, 2008). Implementações com uso de LUT em FPGA e interpolação linear pode chegar à 0,5 ms para imagens do mesmo tamanho utilizado neste trabalho (CHANG *et al.*, 2009). Isso demonstra que próximos avanços devem seguir pela implementação em HW, o que inclusive permitirá o uso de técnicas de interpolação superiores.

A transferência de dados de um quadro de 200 kBytes pela UART leva aproximadamente 80 s, sendo que este tempo não é otimizado e é definido em SW. Futuras implementações que precisem manter um envio de dados em taxas superiores para um computador podem

utilizar transferências em taxas elevadas ou até mesmo com outras abordagens, como a de Oliveira *et al.* (2022) que utiliza Ethernet Gigabit para rápida transferência de dados de US, obtendo transferência de 4 s para  $121 \times 8 \times 2046$  amostras à 100 Mbps, ou a de Govindan *et al.* (2015) que utilizou a ZedBoard para comprimir os dados em 97% antes da transferência, utilizando a compressão do tipo *Discrete Wavelet Transform* (DWT) em SW e HW, levando um minuto em SW ou 0,25 s em HW, para comprimir 33 MBytes em 0,4 MBytes. Esse aspecto de compressão é importante inclusive para inserir a ZedBoard em um sistema com *front-end*, pois neste trabalho os dados de RF já haviam sido adquiridos e salvos nas memórias da placa. Assim, a transferência de dados entre módulos e para o computador é de grande importância devido sua sempre presente quantidade elevada de dados a serem enviados em curtos períodos de tempo para a busca de imageamento em tempo real.

## 5 CONCLUSÃO

Neste trabalho foi proposto o desenvolvimento e avaliação de um sistema de processamento de sinais de US para reconstrução de imagem em Modo B, pela técnica DAS, embarcado em um dispositivo Zynq-7000 ARM/FPGA SoC. Ambos os resultados qualitativos e quantitativos, usando dados brutos de RF da plataforma ULTRA-ORS, apresentaram concordância com modelos adotados para avaliação, e corroboram que o sistema implementado alcançou os objetivos propostos.

As etapas de processamento de filtragem digital, janelamento de apodização, correção de atraso de focalização, somatório coerente, demodulação com detecção de envoltória, compressão logarítmica e conversão de varredura foram implementadas, simuladas e traduzidas em códigos de descrição de hardware e linguagem C para o dispositivo AMD Xilinx Zynq-7000 AP SoC XC7Z020-CLG484 com processador dual-core ARM Cortex™-A9, presente na placa de desenvolvimento ZedBoard.

A ferramenta Model Composer em conjunto com o MATLAB/Simulink permitiu a utilização de blocos de IP que facilitaram a implementação e validação do sistema através de simulações. As ferramentas de projeto da AMD Xilinx, dentro do ambiente Vivado, possibilitaram a inclusão do HW em um sistema com múltiplos estágios, facilitando a integração com o sistema de processamento, periféricos e memória através da técnica de transferência de dados DMA.

Como principal diferencial deste trabalho em comparação com pesquisas anteriores, pode-se citar a incorporação da etapa de conversão de varredura em um dispositivo híbrido embarcado que integra a programabilidade de SW de um processador baseado em ARM com a reconfigurabilidade e flexibilidade de HW de uma FPGA.

Os principais resultados deste trabalho, usando as métricas de comparação de modelos NRMSE e NRSS, além das razões de contraste de imagens geradas CR e CNR, foram avaliados por pares e apresentados em eventos científicos nacionais e internacionais, conforme listado no Apêndice B, que indica os nomes das publicações nos anais dos eventos.

Como exemplos de possíveis aplicações do sistema implementado, pode-se citar o desenvolvimento de futuros sistemas abertos de pesquisa do US nas áreas médica e industrial. Além disso, abre-se uma nova oportunidade para realização de pesquisas envolvendo processamento embarcado em plataformas de SoC para atividades do LUS e para comunidade científica.

Conclui-se que o sistema projetado é adequado para o processamento de sinais de US e geração de imagens em modo B e que sua contribuição se dá tanto no aspecto de exploração de uma nova tecnologia, o SoPC, quanto na avaliação quantitativa de resultados, permitindo avanços na busca por plataformas abertas de US.

### 5.1 Trabalhos futuros

Para trabalhos futuros, sugere-se:

- Incluir e avaliar opções de janela de apodização através de botões do ZedBoard, como Gaussiana, Kaiser, Tukey, entre outras;
- Implementar a transferência de dados para visualização de imagem processada em monitor com interface HDMI e resolução de 1080p, que é superior aos 480p e 4 bits da interface VGA da ZedBoard;
- Avaliar uma interface de comunicação para transferência massiva de dados entre o sistema ULTRA-ORS e a placa Zedboard;
- Otimizar o projeto para permitir reconstrução de imagem utilizando o *beamformer* DAS com outras aberturas ativas: 16, 32 e 64 (ASSEF, 2013);
- Explorar outras métricas para análise de qualidade da imagem, como *Full Width at Half Maximum* (FWHM) e *Full Width at Half Dynamic Range* (FWHDR) (NEVES, 2020; HARPUT *et al.*, 2014)
- Desenvolver outras técnicas *beamforming* usando abordagens não adaptativas e adaptativas (NEVES, 2020);
- Implementar a etapa de conversão de varredura em HW para a busca de processamento *back-end* em tempo real e melhoria da qualidade da imagem, através de técnicas de interpolação avançadas;
- Explorar usos avançados de DMA, como *pipelining*, para acelerar as transferências para o PS e aumentar a taxa de quadros.

## REFERÊNCIAS

- AKKALA, V. *et al.* FPGA based ultrasound backend system with image enhancement technique. **ISSNIP Biosignals and Biorobotics Conference, BRC**, IEEE, 2014. ISSN 23267844.
- ASSEF, A. *et al.* Projeto de um filtro digital fir passa-baixa em fpga para aplicações de processamento de sinais de ultrassom. 04 2016.
- 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) — Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial, 2013.
- ASSEF, A. A. *et al.* FPGA Implementation and Evaluation of an Approximate Hilbert Transform-Based Envelope Detector for Ultrasound Imaging Using the DSP Builder Development Tool. **2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)**, IEEE, p. 2813–2816, 2019.
- ASSEF, A. A. *et al.* Modeling and FPGA-based implementation of an efficient and simple envelope detector using a Hilbert Transform FIR filter for ultrasound imaging applications. **Research on Biomedical Engineering**, v. 34, n. 1, p. 87–92, 2018. ISSN 24464740.
- 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. *In: . [S.l.: s.n.]*, 2016. v. 2016-October, p. 5172–5175. ISBN 9781457702204. ISSN 1557170X.
- ASSEF, A. A. *et al.* Apresentação De Imagens Setoriais Por Ultrassom. *In: . [S.l.: s.n.]*, 2014. p. 324–327.
- BERKHOF, A. *et al.* Fast scan conversion algorithms for displaying ultrasound sector images. **Ultrasonic Imaging**, v. 16, n. 2, p. 87–108, 1994. ISSN 0161-7346.
- BHARATH, R. *et al.* Compact and Programmable Ultrasound Front-End Processing Module for Research Activities. *In: . [S.l.: s.n.]*, 2018. v. 2018-July, p. 921–924. ISBN 9781538636466. ISSN 1557170X.
- BHARATH, R. *et al.* Novel architecture for wireless transducer based ultrasound imaging system. *In: . [S.l.]: IEEE*, 2016. p. 432–436. ISBN 9781467377911.
- BONI, E. *et al.* ULA-OP 256: A portable high-performance research scanner. **2015 IEEE International Ultrasonics Symposium, IUS 2015**, IEEE, n. 324257, p. 21–24, 2015.
- 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. ISSN 08853010.
- BROADSOUND. **AT3C52B (C5-2 for Verasonics)**. 2023. Disponível em: <https://www.broadsound.com.tw/replacement-probe-139-product366>. Acesso em: out. 2023.
- CHANG, J. H. *et al.* Low-cost, high-speed back-end processing system for high-frequency ultrasound B-mode imaging. **IEEE**, v. 56, n. 7, p. 1490–1497, 2009.
- 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, n. 9, p. 1792–1801, 2007.

- CHANG, J. H.; YEN, J. T.; SHUNG, K. K. High-speed digital scan converter for high-frequency ultrasound sector scanners. **Ultrasonics**, v. 48, n. 5, p. 444–452, 2008. ISSN 0041624X.
- CROCKETT, L. *et al.* **The Zynq Book: Embedded Processing With the ARM® Cortex®-A9 on the Xilinx® Zynq®-7000 All Programmable SoC**. [S.l.]: Strathclyde Academic Media, 2014. ISBN 9780992978709.
- DIGILENT. **Zynq Evaluation and Development Hardware User's Guide**. [S.l.], 2014. Disponível em: <https://digilent.com/reference/programmable-logic/zedboard/reference-manual>. Acesso em: fev. 2022.
- ESMAILIAN, K.; Mohammadzadeh Asl, B. Correlation-based modified delay-multiply-and-sum beamforming applied to medical ultrasound imaging. **Computer Methods and Programs in Biomedicine**, v. 226, p. 107171, 2022. ISSN 0169-2607. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0169260722005521>.
- 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. 115 p. Dissertação (Mestrado) — Programa de Pós-Graduação em Sistemas de Energia da Universidade Tecnológica Federal do Paraná, Curitiba, 2017.
- FRAZIER, C.; O'BRIEN, W. Synthetic aperture techniques with a virtual source element. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 45, n. 1, p. 196–207, 1998.
- GAMMELMARK, K.; JENSEN, J. Multielement synthetic transmit aperture imaging using temporal encoding. **IEEE Transactions on Medical Imaging**, v. 22, n. 4, p. 552–563, 2003.
- GOVINDAN, P. *et al.* Reconfigurable and programmable System-On-Chip hardware platform for real-time ultrasonic testing applications. **2015 IEEE International Ultrasonics Symposium, IUS 2015**, IEEE, p. 31–34, 2015.
- HARPUT, S. *et al.* New performance metrics for ultrasound pulse compression systems. *In*: **2014 IEEE International Ultrasonics Symposium**. [S.l.: s.n.], 2014. p. 440–443.
- HASSAN, M. A.; KADAH, Y. M. Digital Signal Processing Methodologies for Conventional Digital Medical Ultrasound Imaging System. **American Journal of Biomedical Engineering**, v. 2013, n. 1, p. 14–30, 2013. Disponível em: <http://journal.sapub.org/ajbe>.
- HEDRICK, W. R.; HYKES, D. L.; STARCHMAN, D. E. **Ultrasound Physics and Instrumentation fourth edition**. [S.l.]: Elsevier Mosby, 2005.
- INTEL. **FPGAs Intel e FPGAs SoC**. 2023. Disponível em: <https://www.intel.com.br/content/www/br/pt/products/details/fpga.html>. Acesso em: set. 2023.
- JENSEN, J. A. FIELD: A program for simulating ultrasound systems. **Medical and Biological Engineering and Computing**, v. 34, n. SUPPL. 1, p. 351–352, 1996. ISSN 01400118.
- JIANG, L. *et al.* Flexible ultrasound-induced retinal stimulating piezo-arrays for biomimetic visual prostheses. **Nature Communications**, v. 13, 2022.
- JIANG, Y.; VIRUPAKSHAPPA, K.; ORUKLU, E. FPGA implementation of a support vector machine classifier for Ultrasonic flaw detection. **Midwest Symposium on Circuits and Systems**, v. 2017-Augus, p. 180–183, 2017. ISSN 15483746.
- JONVEAUX, L. *et al.* Review of current simple ultrasound hardware considerations, designs, and processing opportunities. **Journal of Open Hardware**, Feb 2022.



- KASSEM, A.; HAMAD, M.; SAWAN, M. An efficient SoC dedicated to ultrasonic digital imaging systems. **Proceedings - The 6th IEEE International Workshop on System on Chip for Real Time Applications, IWSOC 2006**, p. 165–168, 2006.
- KASSEM, A.; SAWAN, M.; BOUKADOUM, M. A NEW DIGITAL SCAN CONVERSION ARCHITECTURE FOR ULTRASONIC IMAGING SYSTEMS. **Journal of Circuits, Systems and Computers**, v. 14, n. 02, p. 367–382, apr 2005. ISSN 0218-1266. Disponível em: <https://www.worldscientific.com/doi/abs/10.1142/S0218126605002374>.
- LI, X. **Ultrasound Scan Conversion on TI's C64x+ DSPs**. [S.l.], 2009. 1–15 p. Disponível em: <https://www.ti.com/lit/an/sprab32/sprab32.pdf>.
- LUS. **Laboratório de Ultrassom e Instrumentação Biomédica**. 2023. Disponível em: <https://utfpr.curitiba.br/lus/>. Acesso em: out. 2023.
- LYONS, R. Digital Envelope Detection: The Good, the Bad, and the Ugly [Tips and Tricks]. **IEEE Signal Processing Magazine**, v. 34, n. 4, p. 183–187, 2017. ISSN 10535888.
- MEDEIROS, R. *et al.* B-mode ultrasound imaging system using raspberry pi. *In: XXVII Brazilian Congress on Biomedical Engineering*. [S.l.]: Springer International Publishing, 2022. p. 909–916. ISBN 978-3-030-70600-5.
- MENTO, F. *et al.* Quantitative lung ultrasound spectroscopy applied to the diagnosis of pulmonary fibrosis: The first clinical study. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 67, n. 11, p. 2265–2273, 2020.
- MICUCCI, M.; IULA, A. Recent advances in machine learning applied to ultrasound imaging. **Electronics**, v. 11, n. 11, 2022. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/11/11/1800>.
- NEVES, L. C. **Beamformer adaptativo combinado com generalized sidelobe canceler e filtros adaptativos para geração de imagens de ultrassom com ondas planas**. 2020. 87 p. Dissertação (Mestrado) — Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, Curitiba, 2020.
- OLIVEIRA, J. de. **Implementação de um sistema de transferência de dados de ultrassom via Ethernet e processamento embarcado em dispositivo FPGA**. 2020. 108 p. Dissertação (Mestrado) — Programa de Pós-Graduação em Sistemas de Energia da Universidade Tecnológica Federal do Paraná, Curitiba, 2020.
- OLIVEIRA, J. de; 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**, p. 8–11, 2019.
- OLIVEIRA, J. de *et al.* Implementation of an ultrasound data transfer system via ethernet with fpga-based embedded processing. *In: BASTOS-FILHO, T. F.; CALDEIRA, E. M. de O.; FRIZERA-NETO, A. (Ed.). XXVII Brazilian Congress on Biomedical Engineering*. Cham: Springer International Publishing, 2022. p. 897–903. ISBN 978-3-030-70601-2.
- OLIVEIRA, R. R. d. *et al.* Lung ultrasound: an additional tool in covid-19. **Radiologia Brasileira**, Publicação do Colégio Brasileiro de Radiologia e Diagnóstico por Imagem, v. 53, n. 4, p. 241–251, Jul 2020. ISSN 0100-3984. Disponível em: <https://doi.org/10.1590/0100-3984.2020.0051>.
- OPPENHEIM, A.; SCHAFFER, R. **Digital Signal Processing**. [S.l.]: Prentice-Hall, 1975. (MIT video course). ISBN 9780132146357.

- QIU, W. *et al.* An FPGA-based open platform for ultrasound biomicroscopy. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, IEEE, v. 59, n. 7, p. 1432–1442, 2012. ISSN 08853010.
- ROBINSON, D. E.; KNIGHT, P. C. Interpolation scan conversion in pulse-echo ultrasound. **Ultrason. Imaging**, v. 4, n. 4, p. 297–310, 1982. ISSN 01617346.
- RODRIGUEZ-MOLARES, A. *et al.* The generalized contrast-to-noise ratio: A formal definition for lesion detectability. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 67, n. 4, p. 745–759, 2020.
- RUZYK, M. J. da S. *et al.* Modelling and simulation of an ultrasound reception beamforming using Xilinx model composer. *In*: RITTNER, L. *et al.* (Ed.). **17th International Symposium on Medical Information Processing and Analysis**. SPIE, 2021. v. 12088, p. 120881J. Disponível em: <https://doi.org/10.1117/12.2606292>.
- RUZYK, M. J. da S. *et al.* Ultrasound scan converter implemented on xilinx zynq-7000 all programmable systems-on-chip. *In*: **IX Latin American Congress on Biomedical Engineering (CLAIB 2022) and the XXVIII Brazilian Congress on Engineering Biomedical Engineering (CBEB 2022)**. [S.l.: s.n.], 2022. p. 1–12.
- RUZYK, M. J. da S. *et al.* Comparison of three simulation methods for rf ultrasound signal envelope detection. *In*: **2021 IEEE UFFC Latin America Ultrasonics Symposium (LAUS)**. [S.l.: s.n.], 2021. p. 1–4.
- SCHNEIDER, F. K. *et al.* A Fully programmable computing architecture for Medical Ultrasound Machines. **Technology**, v. 14, n. 2, p. 2009–2011, 2010.
- SHANKAR, H.; PAGEL, P. Potential adverse ultrasound-related biological effects a critical review. **Anesthesiology**, v. 115, p. 1109–24, 08 2011.
- SHUNG, K. K. **Diagnostic ultrasound: Imaging and blood flow measurements**. [S.l.]: CRC press, 2005.
- SIKDAR, S. *et al.* Programmable Ultrasound Scan Conversion on a Mediaprocessor-based System. v. 4319, p. 699–711, 2001.
- SZABO, T. L. **Diagnostic ultrasound imaging: inside out**. [S.l.]: Academic press, 2004.
- TANTER, M. *et al.* Ultrafast compound imaging for 2-d motion vector estimation: application to transient elastography. **IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control**, v. 49, n. 10, p. 1363–1374, 2002.
- WANG, B.; SANIIE, J. Ultrasonic Signal Acquisition and Processing platform based on Zynq SoC. *In*: . [S.l.]: IEEE, 2016. v. 2016-Augus, p. 448–451. ISBN 9781467399852. ISSN 21540373.
- WANG, Y. *et al.* Developing medical ultrasound beamforming application on GPU and FPGA using oneAPI. *In*: . [S.l.: s.n.], 2021. p. 360–370. ISBN 9781665435772.
- WEBSTER, J. G. **Medical Instrumentation: Application and Design**. [S.l.]: John Wiley and Sons, Inc., 2010. ISBN 139780471676003.
- WINDER, S. **Analog and Digital Filter Design**. [S.l.]: Elsevier Science, 2002. (EDN Series for Design Engineers). ISBN 9780080488332.
- XILINX. **AXI4-Stream FIFO v4.1. LogiCORE IP Product Guide**. [S.l.], 2016. Disponível em: <https://docs.xilinx.com/v/u/4.1-English/pg080-axi-fifo-mm-s>. Acesso em: jun. 2023.

XILINX. **Processing System 7 v5.5, LogiCORE IP Product Guide**. [S.l.], 2017. Disponível em: <https://docs.xilinx.com/v/u/en-US/pg082-processing-system7>. Acesso em: out. 2023.

XILINX. **Zynq-7000 SoC Data Sheet: Overview**. [S.l.], 2018. Disponível em: <https://docs.xilinx.com/v/u/en-US/ds190-Zynq-7000-Overview>. Acesso em: out. 2023.

XILINX. **Vivado Design Suite User Guide. Designing with IP**. [S.l.], 2023. Disponível em: [https://docs.xilinx.com/viewer/book-attachment/eUWrI3yPwf6JnaJW1zwdvQ/SFNFCza2Ni1o3\\_TTqtTIFA](https://docs.xilinx.com/viewer/book-attachment/eUWrI3yPwf6JnaJW1zwdvQ/SFNFCza2Ni1o3_TTqtTIFA). Acesso em: out. 2023.

XILINX, A. **AMD Adaptive SoCs: Enabling Differentiated, Customizable Products**. 2023. Disponível em: <https://www.xilinx.com/products/silicon-devices/soc.html>. Acesso em: set. 2023.

XU, C. *et al.* Design and simulation of a ring transducer array for ultrasound retinal stimulation. **Micromachines**, v. 13, n. 9, 2022. ISSN 2072-666X. Disponível em: <https://www.mdpi.com/2072-666X/13/9/1536>.

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

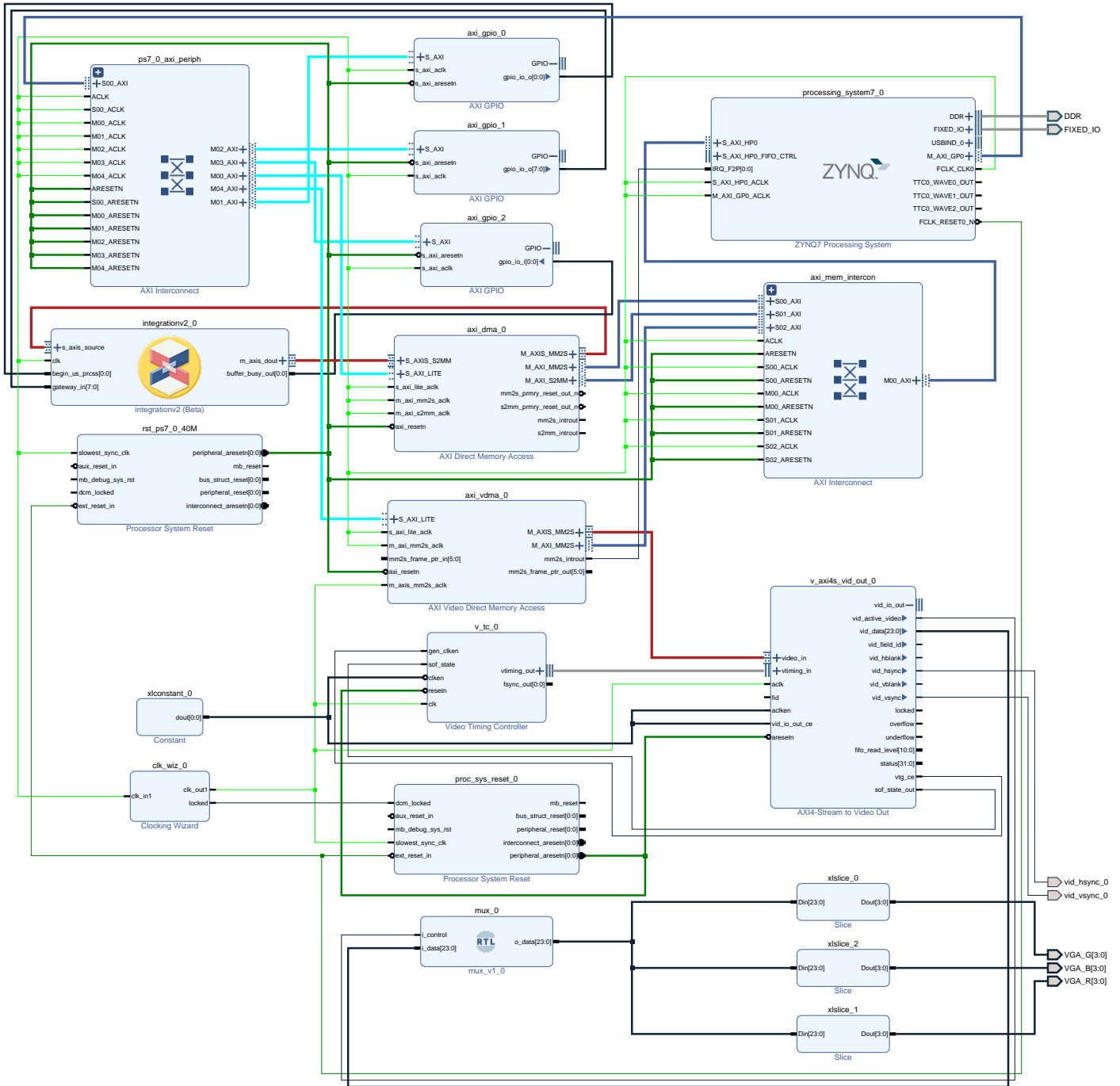
## **APÊNDICE A – Projeto do sistema de ultrassom no Vivado**

O modelo completo do sistema de US no Vivado é apresentado na próxima página e inclui todos os elementos utilizados na forma de blocos IP e sua descrição no Quadro 3.

**Quadro 3 – Configurações e comentários sobre os IPs utilizados no Vivado.**

IP	Configurações	Comentários
processing_system7_0	S_AXI_HPO_interface	Interface de alta performance AXI
	Periféricos: Quad SPI Flash; USB 0; UART1; TTC0; Timer 0;	Periféricos do PS e ZedBoard
	Clock CPU 667 MHz; DDR 533 MHz; PL Fabric Clock FCLK_CLK0 40 MHz;	Clocks padrão e clock do <i>beamformer</i> de 40 MHz
	Interrupção IRQ_F2P[15:0]	Interrupção do VDMA
axi_dma_0	Registrador buffer 14 bits; endereços 32 bits; sem transferências desalinhadas; mapa de memória 32 bits; dados stream 32 bits, burst 16 bits	Controlador de DMA para leitura e escrita de dados no IP do <i>beamformer</i>
Video Timing Controller	Video mode 480p;	Gera sincronismo horizontal e vertical para VGA
axi_vdma_0	Endereço 32 bits; apenas leitura; memória 64 bits, burst 8 bits, pacotes 24 bits e 512 bits buffer	Controlador de VDMA para envio de informações ao VGA
AXI4-Stream to Video Out	1 pixel por clock; vídeo RGB 8 bits; profundidade de FIFO de 1024; clock comum e sincronismo escravo	Converte stream de dados de vídeo em 24 bits VGA
clk_wiz_0	Tipo de gerador de clock MMCM; transforma 40 MHz em 25 MHz	Clock blocos de vídeo 25 MHz =800x525 pixels x60 Hz (480p)
xlconstant_0	Valor 1	Enable clocks vídeo
mux_0	Mux em VHDL de 1 canal	MUX de sincronismo
xlslice_0 , 1 e 2	4 bits de tamanho	Recorta bits para VGA
axi_gpio 0 - 2	-	Sinais <i>beamformer</i>
rst_ps7_0_40M	Configuração automática	Sincronismo clock 40 MHz
ps7_0_axi_periph	Configuração automática	Interconexão AXI4, controle dos GPIOs e axi_vdma_0
proc_sys_reset_0	Configuração automática	Sincronismo de reset de vídeo
axi_mem_intercon	Configuração automática	Interconexão da AXI4_HP0 do PS com controladores DMA
integrationv2_0	Configuração automática	IP do <i>beamformer</i> de US

**Fonte: Autoria própria (2024).**



## **APÊNDICE B – Publicações referentes ao trabalho de mestrado**

1. RUZYK, M. J. S.; ASSEF, A. A. ; MAIA, J. M. ; COSTA, EDUARDO T. Implementation of B-mode Ultrasound Imaging System on Xilinx ZedBoard. In: **XI Symposium on Instrumentation and Medical Imaging 2023**, 2023, Ribeirão Preto. XI SIIM 2023, 2023.
2. RUZYK, M. J. S.; ASSEF, A. A. ; MAIA, JOAQUIM M. ; COSTA, EDUARDO T. Ultrasound scan converter implemented on Xilinx Zynq-7000 all programmable systems-on-chip.. In: **IX Latin American Congress on Biomedical Engineering (CLAIB 2022) and the XXVIII Brazilian Congress on Engineering Biomedical Engineering (CBEB 2022)**, 2022, Virtual. CLAIB & CBEB 2022, 2022.
3. DA SILVA RUZYK, MATHEUS JOSE; ASSEF, AMAURI A. ; MAIA, JOAQUIM M. ; DE OLIVEIRA, JONATHAN ; ABREU DE SOUZA, MAUREN ; COSTA, EDUARDO T. . Modelling and simulation of an ultrasound reception beamforming using Xilinx model composer. In: **Seventeenth International Symposium on Medical Information Processing and Analysis**, 2021, Campinas. 17th SIPAIM, 2021. p. 57.
4. DA SILVA RUZYK, MATHEUS JOSE; MARTINEZ, AMANDA COSTA ; ASSEF, AMAURI AMORIN ; DE OLIVEIRA, LUCAS RIBEIRO ; MAIA, JOAQUIM MIGUEL ; COSTA, EDUARDO TAVARES . Comparison of three simulation methods for RF ultrasound signal envelope detection. In: **2021 IEEE UFFC Latin America Ultrasonics Symposium (LAUS)**, 2021, Gainesville. LAUS, 2021. p. 1-31.