UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Teste e diagnóstico de interfaces utilizando FPGA com reprogramação dinâmica e software embarcado para o Espectrômetro Digital de Ressonância Magnética da CIERMag

João Carlos Batista

Dissertação de Mestrado do Programa de Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria (MECAI)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

João Carlos Batista

Teste e diagnóstico de interfaces utilizando FPGA com reprogramação dinâmica e software embarcado para o Espectrômetro Digital de Ressonância Magnética da CIERMag

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria. *VERSÃO REVISADA*

Área de Concentração: Matemática, Estatística e Computação

Orientador: Prof. Dr. Alberto Tannús

USP – São Carlos
Novembro de 2020

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B333t Batista, João Carlos
 Teste e diagnóstico de interfaces utilizando FPGA
 com reprogramação dinâmica e software embarcado para
 o Espectrômetro Digital de Ressonância Magnética da
 CIERMag / João Carlos Batista; orientador Alberto
 Tannús. -- São Carlos, 2020.
 128 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
 em Mestrado Profissional em Matemática, Estatística
 e Computação Aplicadas à Indústria) -- Instituto de
 Ciências Matemáticas e de Computação, Universidade
 de São Paulo, 2020.

 1. FPGA. 2. ARM. 3. Radiofrequência. 4.
 Gradiente. 5. CIERMag. I. Tannús, Alberto, orient.
 II. Título.

João Carlos Batista

Interface testing and diagnostics using FPGA with dynamic
reprogramming and embedded software for CIERMag's
Digital Magnetic Resonance Spectrometer

Master dissertation submitted to the Institute of
Mathematics and Computer Sciences – ICMC-USP, in
partial fulfillment of the requirements for the degree of
the Master – Professional Masters in Mathematics,
Statistics and Computing Applied to Industry. *FINAL
VERSION*

Concentration Area: Mathematics, Statistics and
Computing

Advisor: Prof. Dr. Alberto Tannús

USP – São Carlos
November 2020

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.
Em especial, ao pesquisadores do Instituto de Ciências Matemáticas e de Computação
(ICMC).*

AGRADECIMENTOS

Agradeço a Deus, pela força, saúde, proteção, e por nunca me abandonar nas horas de dificuldade.

A toda minha Família, a minha mãe Inês, pelo carinho, dedicação e provendo ajuda financeira e motivacional para a conclusão do mestrado. Ao ICMC, IFSC e a USP pela oportunidade de estudo e realização, e aos professores pela paciência e dedicação.

À minha esposa Marina Jeaneth Machicao Justo, que muito além de proporcionar diversos momentos felizes ao lado dela, tem me apoiado e dado forças nesta fase difícil.

Agradeço imensamente ao Grupo CIERMag (Centro de Imagens e Espectroscopia in vivo por Ressonância Magnética) no IFSC, onde atuei profissionalmente, e especial aos professores Dr. Alberto Tannús, Mateus José Martins, Edson Luiz Géa Vidoto por todas as oportunidades, pela orientação, atenção, ajuda e suporte no desenvolvimento desse trabalho e todos os profissionais e amigos da (CIERMag).

Agradeço o professor Dr. Antonio Castelo Filho, do ICMC e todos os professores do ICMC.

A *CNPq* (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo apoio financeiro ao longo do curso de mestrado.

Aos profissionais da FIT (*Fine Instrument Technology*) e ao Daniel Martelozo Consalter.

Por último, mas não menos importante, aos meus amigos, Ângela S. M. Cunha, Sama Rouhani e Rafael Silva.

*“As invenções são, sobretudo,
o resultado de um trabalho de teimoso.”
(Santos Dumont)*

RESUMO

BATISTA, J. C. **Teste e diagnóstico de interfaces utilizando FPGA com reprogramação dinâmica e software embarcado para o Espectrômetro Digital de Ressonância Magnética da CIERMag**. 2020. 128 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

O Centro de Imagens e Espectroscopia *in vivo* por Ressonância Magnética (CIERMag) tem uma alta produção no desenvolvimento de placas que compõem o Espectrômetro Digital de Ressonância Magnética. Essas placas operam com circuitos que contêm conversores de radiofrequência e conversores de gradientes que são usados em experimentos de ressonância magnética. Antes de liberar uma placa como produto final, a sua qualidade deve ser testada, pois possíveis falhas de funcionalidade ou operabilidade poderiam danificar os equipamentos ou posteriores experimentos. Atualmente, um especialista realiza um diagnóstico da qualidade das placas de forma manual, sendo assim um procedimento demorado e com risco de queimar os equipamentos durante o manuseio. Neste trabalho foi proposto um sistema para teste e diagnóstico de placas, com o propósito de detectar possíveis falhas de produção. O sistema foi desenvolvido sobre uma arquitetura híbrida e em um sistema embarcado de FPGA, composto da lógica programável em hardware e do software no processador ARM, operando de forma sincronizada. O sistema é controlado e configurado por meio de uma interface gráfica, permitindo realizar testes nas diversas arquiteturas de placas da CIERMag. As duas grandes características do sistema são: (i) testes dedicados para avaliar a qualidade e a estabilidade dos componentes a cada bloco funcional do circuito eletrônico; (ii) testes aplicados para avaliar todas as variações nos pontos de amplitudes que formam um sinal. A robustez e sensibilidade do sistema proposto também foi validada usando dois tipos de experimentos usando datasets reais e sintéticos. Em ambos casos, foi observado que o sistema consegue detectar a presença de ruído durante a reconstrução do sinal. De acordo como os experimentos, o sistema permite avaliar e elaborar um diagnóstico preciso a cada bloco funcional do circuito eletrônico da placa em teste e tudo isso aliado a precisão, praticidade e portabilidade. Assim, o sistema proposto tem a principal vantagem de facilitar para o especialista de aplicar testes personalizados, semi-automatizados ou automatizados para avaliação da qualidade das placas da CIERMag.

Palavras-chave: FPGA, ARM, Radiofrequência, Gradiente, CIERMag.

ABSTRACT

BATISTA, J. C. **Interface testing and diagnostics using FPGA with dynamic reprogramming and embedded software for CIERMag's Digital Magnetic Resonance Spectrometer.** 2020. 128 p. Dissertação (Mestrado – Mestrado Profissional em Matemática, Estatística e Computação Aplicadas à Indústria) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2020.

The Center for Magnetic Resonance Imaging and Spectroscopy (CIERMag) has a high production in the development of circuit boards that compose the Digital Magnetic Resonance Spectrometer. These boards operate with circuits that contain radiofrequency converters and gradient converters that are used in magnetic resonance experiments. Before releasing a circuit board as a final product, its quality must be tested, as possible failures of functionality or operability could damage the equipment or subsequent experiments. Currently, a specialist performs a diagnosis of the quality of the boards manually, thus being a time-consuming procedure with the risk of burning the equipment during handling. In this work, a system for testing and diagnosis of boards was proposed, to detect possible production failures. The system was developed on a hybrid architecture and on an embedded FPGA system, composed of the programmable logic in hardware and software in the ARM processor, operating in a synchronized way. The system is controlled and configured through a graphical interface, allowing tests to be performed on several CIERMag board architectures. The two main features of the system are (i) dedicated tests to evaluate the quality and stability of the components to each functional block of the electronic circuit; (ii) applied tests to evaluate all the variations in the amplitudes points that form a signal. The robustness and sensitivity of the proposed system were also validated using two types of experiments using real and synthetic datasets. In both cases, it was observed that the system can detect the presence of noise during signal reconstruction. According to the experiments, the system allows us to evaluate and elaborate an accurate diagnosis of each functional block of the electronic circuit board under test and all this combined with precision, practicality, and portability. Thus, the proposed system has the main advantage of making it easier for the specialist to apply personalized, semi-automated, or automated tests for evaluating the quality of CIERMag boards.

Keywords: FPGA, ARM, Radiofrequency, Gradient, CIERMag.

LISTA DE ILUSTRAÇÕES

Figura 1 – Visão detalhada da metodologia para sintetizar circuitos em FPGA.	33
Figura 2 – Placa desenvolvida pela CIERMag	37
Figura 3 – Cabo <i>Flex cable</i> para a comunicação da FPGA com a placa da CIERMag versão 2.0 ou 3.0.	38
Figura 4 – Exemplo de modulação de um sinal <i>Cossenoidal</i>	42
Figura 5 – Exemplo da filtragem de um sinal.	43
Figura 6 – Exemplo do funcionamento do misturador	44
Figura 7 – Estrutura do filtro CIC.	45
Figura 8 – Representação do filtro FIR	47
Figura 9 – Simulação do filtro CIC e, em seguida, pelo filtro CFIR.	48
Figura 10 – Fluxograma do sistema das partes do <i>software</i> e <i>hardware</i> e a placa para teste.	51
Figura 11 – Visão detalhada do sistema desenvolvido.	53
Figura 12 – Fluxo de dados bidirecional entre o <i>software</i> e a FPGA	53
Figura 13 – Arquitetura da comunicação entre o <i>software</i> , a FPGA e a placa para teste	55
Figura 14 – Módulos de transmissão e recepção do sinal de RF	57
Figura 15 – Ambiente de experimentação da transmissão de radiofrequência	58
Figura 16 – Visão geral do funcionamento de uma memória dual <i>clock</i>	59
Figura 17 – Módulo de Modulação	60
Figura 18 – Módulo NCO	61
Figura 19 – Fluxo de dados da transmissão de sinais de RF	62
Figura 20 – Módulo Misturador	64
Figura 21 – Módulo Filtro CIC	65
Figura 22 – Módulo CFIR	66
Figura 23 – Estrutura de comunicação de três camadas	69
Figura 24 – Alguns pulsos do sistema que podem ser configurados em amplitude, números de pontos e as formas do sinal listado abaixo.	72
Figura 25 – Visão da interface gráfica do <i>software</i>	72
Figura 26 – Visão da interface no painel de resultados	75
Figura 27 – Fluxo de dados que ocorre na transmissão e recepção do sinal de RF	78
Figura 28 – Exemplos de saturação e não saturação de um sinal	80

Figura 29 – Conjunto de placas para testes avaliadas. Quatro amostras da versão 3.0 e uma da versão 2.0.	84
Figura 30 – Validação do sistema no osciloscópio das saídas RF DACs do sistema	85
Figura 31 – Análise da flexibilidade do sistema via osciloscópio	86
Figura 32 – Validação via <i>software</i> da transmissão e recepção do sinal de RF variando o ganho do VGA	87
Figura 33 – Relatório automático das variações do ganho do VGA	88
Figura 34 – Exemplos de erros na reconstrução do sinal de RF	90
Figura 35 – Plot de 20 sinais de testes de sinal <i>quadrado</i> para identificar variações de amplitude	91
Figura 36 – Bateria de testes nos conversores RF DACs , VGA e RF ADCs	92
Figura 37 – Detecção de sinal ruidoso no conjunto de $N = 40$ sinais de teste aplicados ao canal A.	94
Figura 38 – Teste do sinal S_5 é identificado com uma variação de amplitude $\Delta A = 6$.	94
Figura 39 – Conjuntos de testes aplicado à Placa-1 no canal A destacando o <i>outlier</i> referente ao sinal S_5 modificado.	96
Figura 40 – Teste do sinal S_5 referente à Placa-1 do canal A.	96
Figura 41 – Conjunto de testes aplicado à Placa-1 do canal A.	97
Figura 42 – Comparação da qualidade entre as placas que foram testada. Relação da qualidade entre os canais da mesma placa e uma comparação com os demais canais.	97
Figura 43 – Diagrama do conversor DAC usado na placa para teste	108
Figura 44 – Sinais de controle VGA.	109
Figura 45 – Conversor ADCs.	110
Figura 46 – Exemplo de sinais de RF e gradiente	111
Figura 47 – Arquitetura de comunicação usando sinais de gradientes	114
Figura 48 – Ambiente de experimentação da transmissão de sinais de gradientes	115
Figura 49 – Funcionamento dos sinais de gradientes	116
Figura 50 – Arquitetura do módulo RAM RF do gradiente	117
Figura 51 – A) O módulo Grad DAC controla o fluxo de dados pelo segundo conversor Grad DAC indicado em B. B) Sinais que gerenciam o controle e transmissão dos dados pelo conversor, onde o DAC-A corresponde ao sinal do gradiente Z e DAC-B corresponde o sinal B0.	118
Figura 52 – Representação da forma de onda seguindo o protocolo SPI	119
Figura 53 – Interface de configurações de gradientes	121
Figura 54 – Interface de configurações de gradientes	122
Figura 55 – Exemplo de quatro sinais de gradientes sendo transmitidos pelos conversores Grad DACs	125
Figura 56 – Resultados de dois testes aplicados com sinais de gradientes	127

Figura 57 – Fluxo de dados que ocorre na transmissão e recepção do sinal de RF e transmissão do sinal de gradientes	128
--	-----

LISTA DE TABELAS

Tabela 1 – Terminologia dos conversores.	38
Tabela 2 – Módulos e submódulos de transmissão e recepção de radiofrequência desenvolvidos neste trabalho.	54
Tabela 3 – Lista de barramentos usados na arquitetura desenvolvida.	56
Tabela 4 – Variação de amplitude aceitável para o sinal ser convertido de digital → analógico → digital nos conversores RF ADCs e RF DACs	91
Tabela 5 – Tabela de temporização.	109
Tabela 6 – Terminologia dos conversores.	112
Tabela 7 – Módulos e submódulos de transmissão de gradientes.	113
Tabela 8 – Lista de barramentos de gradientes usados na arquitetura desenvolvida.	113

LISTA DE ABREVIATURAS E SIGLAS

ADC	<i>Analogic to Digital Converter</i>
ARM	<i>Advanced RISC Machine</i>
CI	Circuito Integrado
CIERMag	Centro de Imagens e Espectroscopia in vivo por Ressonância Magnética
DAC	<i>Digital to Analogic Converter</i>
DMRS	Espectrômetro Digital de Ressonância Magnética
FIFO	<i>First In, First Out</i>
FPGA	<i>Field-Programmable Gate Array</i>
HDL	<i>Hardware Description Language</i>
HPS	<i>Hard Processor System</i>
HSMC	<i>High Speed Mezzanine Card</i>
HW	<i>Hardware</i>
I/O	<i>Input/Output</i>
LED	<i>Light-Emitting Diode</i>
LUT	<i>Look-Up Table</i>
MMU	<i>Memory Management Unit</i>
MRI	Imagens por Ressonância Magnética
NCO	<i>Numerically Controlled Oscillator</i>
RF	Sinal de radiofrequência
RMN	Ressonância Magnética Nuclear
RX	Recepção de radiofrequência
SMA	<i>SubMiniature version A</i>
SoC	<i>System-on-a-chip</i>
SPI	<i>Serial Peripheral Interface</i>
SW	<i>Software</i>
TX	Transmissão de radiofrequência
VGA	<i>Variable Gain Amplifier</i>

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Justificativa e motivação	27
1.2	Objetivos	28
1.2.1	<i>Vantagens para a empresa</i>	29
1.3	Organização	29
2	FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS	31
2.1	FPGA (arranjo de portas programáveis em campo)	31
2.2	Operação híbrida com processador e lógica programável	33
2.3	Placas desenvolvidas pela CIERMag	34
2.3.1	<i>Terminologia</i>	35
2.3.2	<i>Conversor de radiofrequência como transmissores RF DACs e receptores RF ADCs</i>	36
2.3.3	<i>Versões de placas a serem testadas</i>	36
2.3.4	<i>Conexões entre a FPGA e a Placa da CIERMag</i>	38
2.4	Gerador NCO	39
2.5	Modulação	40
2.6	Demodulação	41
2.6.1	<i>Misturador de sinais</i>	41
2.6.2	<i>Filtros</i>	45
2.6.2.1	<i>Filtro CIC decimador</i>	45
2.6.2.2	<i>Filtro CFIR</i>	46
2.7	Considerações finais	49
3	DESENVOLVIMENTO DO HARDWARE	51
3.1	Descrição da implementação do <i>hardware</i>	52
3.2	Comunicação de dados entre o <i>software</i>, a FPGA e a placa para teste	54
3.3	Transmissão e recepção do sinal de RF	56
3.3.1	<i>Módulos da transmissão de sinal de radiofrequência</i>	58
3.3.1.1	<i>Módulo Envia</i>	58
3.3.1.2	<i>Módulo Controla RAM</i>	58
3.3.1.3	<i>Módulo RAM RF</i>	59

3.3.1.4	<i>Módulo Modulação</i>	60
3.3.1.5	<i>Módulo NCO</i>	61
3.3.1.6	<i>Módulo Offset DAC</i>	61
3.3.2	<i>Módulos da recepção do sinal de radiofrequência</i>	62
3.3.2.1	<i>Módulo Controle VGA</i>	62
3.3.2.2	<i>Módulo Data ADCs</i>	63
3.3.2.3	<i>Módulo Offset ADCs</i>	63
3.3.2.4	<i>Módulo Misturador</i>	63
3.3.2.5	<i>Módulo Filtro CIC</i>	64
3.3.2.6	<i>Módulo Filtro CFIR</i>	65
3.3.2.7	<i>Módulo FIFO</i>	66
3.3.2.8	<i>Módulo Recebe</i>	67
3.4	<i>Considerações finais</i>	67
4	<i>DESENVOLVIMENTO DE SOFTWARE</i>	69
4.1	<i>Visão geral da estruturação do software</i>	69
4.1.1	<i>Tipos de pulsos</i>	71
4.2	<i>Painel de transmissão e recepção de sinais de RF</i>	71
4.3	<i>Painel de resultados</i>	74
4.4	<i>Restrições e alertas do sistema</i>	77
4.5	<i>Verificação dos componentes RF DACs, VGA e RF ADCs por meio do software</i>	77
4.5.1	<i>Falhas de funcionalidade</i>	79
4.5.2	<i>Falhas de qualidade</i>	79
4.5.3	<i>Interferências geradas por ruídos</i>	79
4.5.4	<i>Saturação do conversor RF ADCs pelo VGA</i>	80
4.6	<i>Considerações finais</i>	81
5	<i>RESULTADOS E DISCUSSÕES</i>	83
5.1	<i>Configuração dos experimentos</i>	83
5.1.1	<i>Notação dos testes</i>	83
5.1.2	<i>Placas para teste a serem analisadas</i>	84
5.2	<i>Validação via osciloscópio da transmissão do sinal de RF nos conversores RF DACs</i>	84
5.2.1	<i>Análise da flexibilidade do sistema</i>	85
5.3	<i>Validação da transmissão e recepção via software</i>	87
5.3.1	<i>Relatório automático das variações do ganho do VGA</i>	88
5.4	<i>Estratégia para a detecção de falhas na reconstrução do sinal de RF</i>	89
5.4.1	<i>Dataset da placa para teste por introdução de ruído</i>	89

5.4.2	<i>Considerações para a classificação de placa “boa” ou “ruim”</i>	90
5.4.3	<i>Análise do sinal reconstruído no conversor RF ADCs</i>	91
5.5	Diagnóstico da qualidade de cada canal das placas em teste	92
5.5.1	<i>Configuração dos sinais</i>	92
5.5.2	<i>Dataset sintético</i>	93
5.5.3	<i>Identificação de outlier</i>	95
5.5.4	<i>Dataset original</i>	95
5.5.5	<i>Resultado de análise da qualidade de cada canal das placas que foram testadas</i>	96
5.6	Considerações finais	98
6	CONCLUSÕES	99
6.1	Contribuições	100
6.2	Questões para trabalhos futuros	101
REFERÊNCIAS		103
APÊNDICE A	APÊNDICE A:	107
A.1	Arquitetura dos circuitos integrados	107
A.1.1	<i>Arquitetura dos conversores RF DACs</i>	107
A.1.2	<i>Arquitetura do controle de ganho VGA</i>	107
A.1.2.1	<i>Arquitetura do conversor RF ADCs</i>	110
APÊNDICE B	APÊNDICE B:	111
B.1	Diferenças entre sinais de RF e sinais de gradientes	111
B.1.0.1	<i>Conversor de gradientes Grad DACs</i>	111
B.2	Módulos de gradientes	112
B.3	Comunicação de sinais de gradientes entre o SW, a FPGA e as placas a serem testadas	112
B.3.1	<i>Fluxo de sinais de gradientes</i>	112
B.3.2	<i>Barramentos que incluem gradientes</i>	113
B.4	Transmissão de sinais de Gradientes	114
B.4.1	<i>Visão geral do funcionamento dos sinais de gradientes</i>	114
B.4.1.1	<i>Módulos memórias RAM Grad e RAM RF</i>	115
B.4.1.2	<i>Módulo Multiplexador</i>	116
B.4.1.3	<i>Módulo Grad DAC e conversores Grad DACs</i>	117
B.4.1.4	<i>Preparação e transmissão de dados no conversor Grad DAC</i>	118
B.5	Interface de usuário de gradientes	120
B.5.1	<i>Arquitetura do gradientes</i>	120
B.6	Transmissão do pulso de Gradientes	121

B.7	Resultado nos conversores Grad DACs	123
<i>B.7.1</i>	<i>Teste nos conversores de Gradientes</i>	<i>123</i>
B.8	Validação dos sinais de RF e gradientes nas saídas do osciloscópio	124
<i>B.8.1</i>	<i>Teste e análise de dados da transmissão dos gradientes nos conversores Grad X, Grad Y, Grad Z e B0</i>	<i>124</i>
<i>B.8.2</i>	<i>Teste da saída do sinal demodulado do Radiofrequência em alguns dos canais do (Gradientes), todos os conversores funcionando</i>	<i>125</i>

INTRODUÇÃO

1.1 Justificativa e motivação

O Centro de Imagens e Espectroscopia por Ressonância Magnética ([CIERMAG, 2019](#)), localizado no Instituto de Física de São Carlos da Universidade de São Paulo (IFSC-USP), é constituído por uma equipe de pesquisadores formada por alunos e professores, que vêm desenvolvendo inúmeros projetos no decorrer de sua história. As pesquisas na área de Imagens por Ressonância Magnética (MRI) foram iniciadas em 1983, com o desenvolvimento de um protótipo de tomógrafo por Ressonância Magnética Nuclear (RMN) para pequenos animais. O Grupo conta atualmente com uma estrutura física que abriga experimentos de relaxometria, imagens e espectroscopia, com um enfoque significativo na instrumentação de ressonância magnética (RM).

A RM é cada vez mais utilizada nas mais diversas áreas do conhecimento e é responsável por avanços significativos na área da medicina, agricultura, química, física e na biologia ([MCLEAN, 2014](#)). A realização de MRIs, por exemplo, é rotina em centros de diagnósticos por todo o mundo e pesquisadores na área agrícola e de alimentos estão utilizando cada vez mais a RM em suas pesquisas.

O novo espectrômetro de ressonância magnética digital desenvolvido pela CIERMag conta com uma tecnologia que permite reconfigurar o hardware a partir de software, com base em lógicas programáveis em *Field-Programmable Gate Array* (FPGA). Esse sistema é desenvolvido pelos cientistas do IFSC, e busca imunidade à obsolescência – daí a sua versatilidade - permitindo criar as funcionalidades que o usuário pretender, de forma muito mais rápida e eficiente quando comparada ao uso do hardware convencional.

Atualmente, o CIERMag tem uma alta produção em desenvolvimento de hardware que precisam ser testados para integrar o conjunto que compõe o Espectrômetro Digital de Ressonância Magnética (DMRS). O projeto de teste e diagnóstico permitirá detectar

vários tipos de erros ou falhas técnicas durante o desenvolvimento e a montagem, seja em componentes que estejam operando fora da especificação de fábrica (que pode ocorrer quando se muda de fornecedor) ou com a mudança de valores dos componentes que é resultante do desgaste da utilização do equipamento por um longo tempo. Todas essas falhas, que são complexas para detectar manualmente, ocasionam lentidão para o desenvolvimento, pouca efetividade de produção, e posterior falta de padronização dos testes, que são demorados e insuficientes. É, portanto, necessária uma forma efetiva e rápida que permita a redução no tempo de produção. Esse processo garantirá que o hardware desenvolvido pelo CIERMag seja padronizado em um ambiente controlado.

Neste projeto é proposto o desenvolvimento de uma plataforma de teste de diagnóstico de interfaces para o Espectrômetro DMRS. Assim, é apresentado o desenvolvimento de um software com interface gráfica implementado em uma linguagem versátil como, por exemplo, Python, para que o operador possa interagir com o sistema de teste e também com o desenvolvimento de toda a lógica de controle e geração de sinais digitais implementado na FPGA em linguagem **HDL** (*Hardware Description Language*) **VHDL** ou **Verilog** (MOORE, 2014) para interagir com o hardware de teste e o sistema de controle. O sistema de Teste de Hardware será formado pelo software embarcado desenvolvido na linguagem de programação Python juntamente com o sistema operacional Linux embarcado na arquitetura de processador ARM (do inglês, *Advanced RISC Machine*) e a programação em hardware em FPGA.

Este trabalho partiu da necessidade de realizar testes rápidos nas placas desenvolvidas pela CIERMag com o objetivo de detectar possíveis falhas nestas placas. Neste trabalho foi elaborado um sistema em FPGA que permite realizar testes de forma rápida, com um ambiente controlável para aplicar testes especializados a cada bloco funcional do circuito eletrônico, e finalmente permitir obter um diagnóstico da qualidade da placa de forma mais precisa, independente das mudanças de *layouts* ou arquiteturas das placas.

1.2 Objetivos

Um tomógrafo de ressonância magnética é um equipamento complexo, e mesmo utilizando sistemas digitais, ainda existe a necessidade de conversão para o sistema analógico. Essas interfaces de conversão variam muito de características, bem como de funcionalidade, e ao desenvolvê-las, surge o problema de como testar e caracterizá-las de forma fácil, sem o óbvio e grande desperdício de tempo em instalar as mesmas no equipamento de RM.

Assim, o objetivo deste trabalho é desenvolver um equipamento de verificação, validação, testes e diagnóstico dessas interfaces. Em virtude da diversidade destas, a utilização de lógica programável em campo (FPGA) torna-se necessária, bem como de

sua reprogramação dinâmica.

1.2.1 Vantagens para a empresa

O projeto facilitará aos operadores desse equipamento, pois terá um software embarcado com uma interface gráfica, com o qual poder-se-á selecionar o tipo de subsistema a ser testado, bem como aferir seus parâmetros e comparar resultados com padrões esperados. Dessa maneira, o equipamento terá controle dos experimentos e também contará com grande facilidade de escalabilidade, visto que a FPGA poderá ser reprogramada dinamicamente pelo software embarcado conforme novas interfaces forem desenvolvidas.

1.3 Organização

Este trabalho foi dividido em seis capítulos incluindo esta introdução. No Capítulo 2, são apresentadas as bases teóricas das tecnologias usadas, incluindo arquitetura da FPGA e placas da CIERMag, filtros, assim como todos os fundamentos necessários para o desenvolvimento deste trabalho. No Capítulo 3 e Capítulo 4 é feita a proposta da integração do uso da tecnologia FPGA num sistema embarcado que inclui *hardware* e *software*, respectivamente. No Capítulo 5 são apresentados os resultados desta metodologia assim como de uma bateria de testes nele aplicado. Finalmente, no Capítulo 6, são apresentadas as conclusões deste trabalho.

FUNDAMENTAÇÃO TEÓRICA E TECNOLOGIAS

Neste capítulo são introduzidos todos os conceitos fundamentais que foram necessários para o desenvolvimento deste trabalho. Na primeira parte são ilustradas as etapas de desenvolvimento de um projeto em **FPGA** e a tecnologia híbrida com o processador. Na segunda parte é descrita a arquitetura do objeto de estudo deste trabalho, a **placa da CIERMag**. Na terceira parte são descritas as técnicas para geração de sinais (Seção 2.4), a **modulação** (Seção 2.5), a qual é comumente usada para realizar a transmissão e recepção de sinais pelos conversores da placa de forma adequada. Finalmente, na quarta parte é detalhado o método de **demodulação** (Seção 2.6), que envolve métodos de **misturador** e a aplicação de filtros de ação rápida onde componentes de frequência indesejadas são eliminadas por filtragem, e para proporcionar uma transição mais suave na reconstrução do sinal.

2.1 FPGA (arranjo de portas programáveis em campo)

Uma **FPGA** (do inglês, *Field Programmable Gate Array*) é um circuito integrado composto por um grande arranjo de células lógicas, ou blocos lógicos, as quais podem ser configuradas através de uma linguagem de descrição de hardware (**HDL**, do inglês *Hardware Description Language*), como são o **VHDL**, ou o **Verilog**. Cada célula contém capacidade computacional para implementar funções lógicas e realizar roteamento de comunicação entre elas. Após esta implementação é possível representar os registradores interligados por lógica combinacional com o uso da ferramenta **RTL** (do inglês, *Register Transfer Level*).

De modo geral, para sintetizar circuitos em FPGA seguem-se os seguintes passos: (Etapa-1) o desenvolvimento do projeto, (Etapa-2) a gravação e (Etapa-3) representação

da estrutura de chaveamento e a LUT (do inglês, *Look-Up Table*) na FPGA (HAUCK; DEHON, 2008; BOBDA, 2007).

A seguir são descritas de forma simplificada cada uma destas etapas, as quais podem ser acompanhadas na Figura 1, da esquerda para a direita.

Etapas-1: o primeiro passo é o desenvolvimento de um projeto em FPGA. Esta etapa envolve uma série de passos (MÉLO; DIAS; STEINBACH, 2011) detalhados conforme os itens abaixo (Figura 1 da Etapa-1):

1. A modelagem do design da lógica do *hardware* é programado pelo projetista usando a linguagem HDL. A partir deste ponto, as demais etapas são gerenciadas pela plataforma de desenvolvimento;
2. A síntese lógica utiliza-se como entrada o código RTL, nas linguagens (VHDL ou Verilog) e a ferramenta de síntese tenta fazer a conversão de um código comportamental que está em alto nível para portas lógicas como *AND*, *OR*, *XOR* e registradores, decodificadores, contadores, memórias, comparadores, *flip-flops*, funções aritméticas e outras funções pertencentes à biblioteca;
3. O mapeamento separa tais portas lógicas em grupos a fim de melhorar a correspondência do uso dos recursos lógicos da FPGA;
4. O posicionamento atribui os agrupamentos lógicos para blocos lógicos específicos;
5. O roteamento permite interligar os blocos lógicos através das estruturas de chaveamentos entre o cruzamento das trilhas verticais e horizontais (ver Figura 1 da Etapa-3). Esse chaveamento corresponde à síntese física do posicionamento dos blocos lógicos. Além disso, esta etapa pode ser muito demorada porque o algoritmo de roteamento tenta minimizar o atraso de propagação desses sinais em todo o circuito do projeto;
6. É gerado o *bitstream*, representado por um arquivo binário. Neste *bitstream* são definidos todos os pontos de programação da FPGA, os blocos lógicos e as linhas de roteamento;
7. Finalmente, uma vez que os endereços dos periféricos de entradas e saídas são definidos, o projeto pode ser gravado na FPGA. Esta gravação implica que o circuito fique mapeado entre os pinos, como por exemplo: os pinos (A), (B) e (C) representados pelas interfaces de botão na FPGA (ver Figura 1 da Etapa-2) e o pino de saída (S) pode ser representado por uma LED (diodo emissor de luz).

Etapas-2: é uma visualização da lógica que está no código fonte que inclui todos os itens da Etapa-1. A maioria das FPGAs permitem o processo de reprogramabilidade, ou seja, alterar as funções lógicas visando maior facilidade no desenvolvimento de protótipos;

Etapa-3: ao ampliar a imagem da FPGA (ver Figura 1 da Etapa-3) pode-se observar o nível de blocos lógicos, configuração por memória, estrutura de chaveamento e a LUT. Todas as conexões internas são compostas por segmentos metálicos com pontos de estrutura de chaveamento programável, para que implementem-se as trilhas desejadas conforme o roteamento. Os blocos lógicos são as funções lógicas que são executadas, já os blocos de interconexão fornecem os caminhos para que os dados possam ir de um nó computacional para outro;

Figura 1 – Visão detalhada da metodologia para sintetizar circuitos em FPGA.

Fonte: Adaptada de [Silva \(2016\)](#).

2.2 Operação híbrida com processador e lógica programável

Os processadores com arquitetura ARM (do inglês, *Advanced RISC Machine*) vem sendo muito utilizados e estão presentes em vários produtos eletrônicos. Esta tecnologia é integrada em kits de desenvolvimento para estudantes e em produtos de última geração, por exemplo, *netbooks*, *smartphones*, carros e TVs. Esta dissertação está focada na arquitetura Cortex-A, mais especificamente no modelo Dual-core ARM Cortex-A9 HPS (do inglês *Hard Processor System*) ([Altera Corporation, 2012](#)), pois trata-se de uma arquitetura que permite a execução de sistemas operacionais mais complexos como o Linux, Android/ChromeOS e Windows. Neste modelo, à unidade de gerenciamento

de memória (MMU, do inglês *Memory Management Unit*) a qual está integrado no processado, gerencia os requisitos de memória e a tradução de endereços de memória virtual que estão associados aos endereços físicos dos periféricos.

O desenvolvimento de *hardware* para o sistema SoC (do inglês *System-on-a-chip*) é feito através das ferramentas do Quartus II, Qsys, SignalTap II, RTL, e ModelSim, as quais foram criadas pela [Altera Corporation \(2015\)](#) e posteriormente foram incorporadas à Intel em 2015 ([Globo, 2015](#)).

O **Quartus II**, hoje em dia chamado de [Intel Quartus Prime \(2015\)](#), é um ambiente de desenvolvimento integrado (IDE) usada para o desenvolvimento de projetos em FPGA. Esta IDE permite configurar a posição dos pinos de I/O (do inglês, *input/output*) por meio da edição visual de circuitos lógicos, compilação e geração de arquivos de código HDL de gravação de um projeto. Além disso, nesta IDE podem ser incluídas outras famílias de FPGAs da própria [Altera Corporation \(2015\)](#) como o Cyclone, Arria, Stratix e entre outros.

O **Qsys** é uma ferramenta gráfica de fácil utilização e flexibilidade que é usado na integração de sistemas SoC. O Qsys facilita a integração e geração de toda a lógica de interconexão de IPs (do inglês *intellectual property*) para uma FPGA ou para um sistema híbrido (FPGA+HPS). O Qsys auxilia na integração de componentes de *hardware* permitindo a configuração e a integração dos componentes da FPGA+HPS. O Qsys permite configurar o HPS, o que implica selecionar os periféricos que serão usados e seus respectivos pinos, ou inclusive a adição e integração de periféricos existentes ao projeto.

No Quartus II também podem ser incluídos recursos de depuração como o SignalTap II. O **SignalTap II** é um analisador lógico implementado na própria FPGA que pode ser usada para monitorar sinais em tempo de execução durante a operação de teste no dispositivo.

Finalmente, uma outra ferramenta importante é o ModelSim ([ModelSim-Intel® FPGA, 2019](#)), o qual é usado para realizar simulações e depurações do sistema para avaliar adequadamente o design planejado, e consequentemente, permitindo aos projetistas uma maneira fácil e econômica de acelerar o desenvolvimento da FPGA.

2.3 Placas desenvolvidas pela CIERMag

A CIERMag tem desenvolvido diversos modelos de placas que são usadas para realizar experimentos de ressonância magnética. Estas placas trabalham com sinais de radiofrequência (RF) e com sinais de gradiente. Por outro lado, as placas trabalham com dispositivos chamados de conversores de RF assim como de conversores de gradientes.

Neste trabalho foi desenvolvido um sistema para transmissão e recepção de

sinais tanto do tipo de radiofrequência como de gradiente. É importante notar que nos seguintes capítulos iremos focar especificamente nos sinais de RF, já os detalhes para sinais de gradientes serão tratados no Apêndice B.

2.3.1 Terminologia

Antes de continuar com os detalhes da placa da CIERMag, é necessário a introdução breve da terminologia usada.

- **Sinal analógico:** é um sinal contínuo que assume diferentes valores de amplitude (tensão elétrica) em um intervalo de tempo, e que possui características como frequência, amplitude e fase;
- **Sinal digital:** é um sinal discreto (descontínuo) que assume diferentes valores a cada instante de tempo.
- **Conversores:** é uma interface que faz a conversão de um sinal entre o analógico e o digital e vice-versa;
- **Sinal modulante:** é um sinal que contém a informação a ser transmitida, esse é o sinal de interesse para a CIERMag que deve estar numa largura de banda de -1MHz a 1MHz ;
- **NCO** (do inglês *Numerically Controlled Oscillator*): é um circuito digital que gera sinais discretos no tempo, com frequência e fase controladas de forma precisa.
- **Sinal da portadora:** é o sinal gerado pelo NCO, este opera numa frequência superior à frequência do sinal modulante;
- **Sinal modulado** ou **portadora modulante em amplitude:** é o resultado da multiplicação do sinal modulante com a portadora (o sinal gerado pelo NCO);
- **Modulação:** É o processo de impor informação contida em um sinal eletrônico com uma frequência inferior a 1MHz do (sinal modulante) a um sinal que está a uma frequência entre 1MHz a 20MHz (sinal portadora);
- **Demodulação:** consiste em detectar e extrair o envoltório do sinal modulado, com objetivo de reconstruir o sinal modulante;
- **Misturador:** realiza a multiplicação de dois sinais digitais, sinal modulado com o sinal da portadora;
- **Filtro digital:** processa uma sequência de sinais de entrada resultando numa outra sequência de sinais de saída, que representa o sinal filtrado na forma digital;

2.3.2 Conversor de radiofrequência como transmissores RF DACs e receptores RF ADCs

Os conversores **ADC** (analogico para digital) e **DAC** (digital para analogico), ambos dos acrônimos em inglês *Analog to Digital Converter*, e *Digital to Analog Converter*, respetivamente, são utilizados para a medição, monitoramento ou controle de grandezas analógicas realizadas por intermédio de sistemas digitais (RADATZ, 1997).

Os conversores **DACs** são responsáveis por fazer a ligação entre o ambiente digital, que corresponde aos valores binários discretos, e o ambiente analógico, que corresponde aos valores contínuos em tensão/corrente. Por sua vez, os conversores **ADCs** funcionam no sentido inverso, convertendo sinais de tensão/corrente para binário. Em ambos conversores existem erros de arredondamento baseado numa tensão de referencia que varia de acordo com a especificação do conversor.

A placa desenvolvida pela CIERMag possui canais de transmissão e recepção de sinal de radiofrequência **RF DACs** e **RF ADCs**, respetivamente. Basicamente, estes conversores atuam como **filtros passa-banda** pois operam com um *clock* fixo de 50MHz definido na FPGA, permitindo assim, a transmissão e a recepção dos sinais da portadora modulante em amplitudes que estejam na banda de frequência de 1MHz a 20MHz.

2.3.3 Versões de placas a serem testadas

A CIERMag tem desenvolvido várias versões de placas, porém este trabalho limitou-se às versões 2.0 e 3.0. Ambas as versões de placas contam com um circuito integrado (CI) que contem dois conversores RF **DAC** de 14 bits que esta encapsulado num mesmo componente e com um controle de ganho do **VGA** (do inglês, *Variable Gain Amplifier*).

Embora ambas as versões possuam layouts *similares*, a versão 2.0 contém dois CI de conversores RF **ADCs** para sinais de RF de 14 bits cada, enquanto que a versão 3.0 contém apenas um CI com dois conversores RF **ADCs** de 16 bits encapsulado num mesmo componente. Além disso, a principal diferença entre estas versões é que somente a versão 2.0 contém conversores de gradientes.

As duas versões das placas da CIERMag podem ser observadas na Figura 2, e nela são indicadas as posições dos seus componentes:

- **RF DACs:** ambas as versões de placa contém um CI com 2 conversores **RF DACs** de 14 bits integrado no mesmo componente, sendo o **RF DAC-A** do canal A e o **RF DAC-B** do canal B, responsáveis pela transmissão do sinal de RF;
- **RF ADCs:** a placa versão 3.0 inclui um CI que contem dois conversores **RF ADC** de 16

Figura 2 – Placa da CIERMag versão 2.0 e 3.0. A placa mostra os componentes de controle de ganho **VGA**, componentes de transmissão e recepção de sinal de RF e transmissão de sinais de gradientes. Além disso, inclui conversores **RF DACs** e **RF ADCs**.

Fonte: Elaborada pelo autor.

bits integrado no mesmo componente, enquanto a versão 2.0 abrange dois conversores **RF ADCs** de 14 bits cada, sendo o **RF ADC-A** do canal A e o **RF ADC-B** do canal B responsáveis pela recepção do sinal de RF. Na Figura 2-B ambos conversores **RF DACs** e **RF ADCs** estão localizados no anverso da placa;

- **VGA:** é o componente de controle de ganho usado na recepção do **RF ADC**. O CI do **VGA** atua no ganho ou na atenuação do sinal de recepção do **ADC**, e o controle é feito via programação no protocolo SPI (do inglês *Serial Peripheral Interface*). As duas placas possuem 2 **VGAs**, que vão para o conversor **RF ADC-A** e **RF ADC-B** do canal A e B, respectivamente;
- **Interface HSMC:** do inglês *High Speed Mezzanine Card*, é o barramento de comunicação bidirecional de sinais digitais entre a placa da CIERMag e a FPGA. Através dessa conexão é possível transmitir e adquirir sinais de RF, transmitir sinais de gradientes e controlar ganho nas **VGAs**;
- **Conector SMA (*SubMiniature version A*):** este conector é usado para transmitir os sinais de RF nos conversores **RF DAC-A/RF DAC-B** e receber nos conversores **RF ADC-A/RF ADC-B**;
- **Loopback:** é um cabo com dois pinos machos que conecta uma entrada do conversor **RF ADCs** com a saída do conversor **RF DACs** via cabo conector SMA. Assim, várias

combinações podem ser feitas, porém não podem ser conectados cabos da mesma funcionalidade (RX-RX ou TX-TX) pois danificaria a placa.

A Tabela 1 resume os componentes que serão usados nos posteriores capítulos.

Tabela 1 – Terminologia dos conversores.

Termo	Descrição
DAC-A	Canal A do conversor de sinal de RF.
DAC-B	Canal B do conversor de sinal de RF.
Conversores de radiofrequência	
RF ADCs	Conjunto de conversores que recebem sinais de radiofrequência.
RF ADC-A	Conversor de recepção de sinal de RF do canal A.
RF ADC-B	Conversor de recepção de sinal de RF do canal B.
RF DACs	Conjunto de conversores que transmitem sinais de radiofrequência.
RF DAC-A	Conversor de transmissão do sinal de RF do canal A.
RF DAC-B	Conversor de transmissão do sinal de RF do canal B.

Fonte: Elaborada pelo autor.

2.3.4 Conexões entre a FPGA e a Placa da CIERMag

A Figura 3 mostra o *flex cable*, que é usado para conectar a FPGA (do lado esquerdo) com a placa da CIERMag, versão 2.0 ou 3.0 (do lado direito). Este tipo de conexão é frequentemente usado para realizar a comunicação entre placas com diferentes arquiteturas.

Figura 3 – Cabo *Flex cable* para a comunicação da FPGA com a placa da CIERMag versão 2.0 ou 3.0.

Fonte: Elaborada pelo autor.

Idealmente, precauções devem ser tomadas quando sinais esta numa frequência entre 1MHz a 20MHz passam pelo *flex cable*. De fato, qualquer erro no *layout* da placa ou no extensor usado para a comunicação entre duas placas poderia comprometer a qualidade do sinal. Por exemplo, a taxa de transmissão de sinais de RF que passam pelo *flex cable* possuem uma taxa de 50MHz, enquanto que os sinais do controle de ganho **VGA** ou dos sinais de gradientes operam com uma frequência menor a 1MHz, e por tanto, não afetam a qualidade da funcionalidade desses CIs.

Para que ocorra a transmissão e a recepção do sinal pelos conversores **RF DACs/RF ADCs**, respectivamente, é necessário incluir um gerador de sinais para atuar tanto na transmissão quanto na recepção e reconstrução do sinal, o que será descrito na seguinte seção.

2.4 Gerador NCO

O gerador **NCO** (do inglês *Numerically Controlled Oscillator*) é um circuito digital que gera sinais discretos no tempo, com frequência e fase precisamente controladas, tendo como referência o *clock* do sistema (RADATZ, 1997). Os NCOs possuem várias vantagens em relação a outros tipos de osciladores baseados em circuitos analógicos do tipo *phase-locked loop* (PLL). Carvalho (CARVALHO, 2016), baseando-se nos trabalhos de Kadam *et al.* (2002), Lim e Kim (2004), cita algumas vantagens: o NCO pode ser usado como um sintetizador com quadratura, permitindo uma diferença de fases entre os dois sinais gerados mais acurado. Além disso, o NCO permite uma varredura de sinais com frequências diferentes.

Existem diversas opções para se implementar um gerador senoidal em um sistema utilizando a FPGA (CARVALHO, 2016). Uma alternativa é fazer uso do NCO *megacore*, que é uma **IP** disponibilizada pela Altera (Altera Corporation, 2015), que permite implementar geradores de sinais complexos.

Este IP é um oscilador controlado numericamente que apresenta a possibilidade de escolha da frequência do sinal de saída através de uma entrada numérica com grande faixa de valores. Existem implementações que operam na faixa de 10KHz a 25MHz com 16 bits de precisão. Esse tipo de implementações suportam saída dupla, podendo assim escolher uma dessas saídas pelo canal conversor **RF DACs/RF ADCs**, por exemplo, *seno* e *cosseno*. O sinal gerado pelo NCO é usado na etapa de modulação (onde ocorre a transmissão do sinal de RF) e na etapa de demodulação (onde ocorre a reconstrução do sinal de RF).

2.5 Modulação

Quando é feita a transmissão de informação, os sinais, conforme fornecidos, não podem ser enviados diretamente pelos canais de transmissão. Por esse motivo, é necessário modificar este sinal usando uma onda eletromagnética portadora. A **modulação** é o conjunto de técnicas para variar um ou mais parâmetros da onda portadora (amplitude, fase ou frequência) de acordo com as variações do sinal de modulação, que contém a informação que se deseja transmitir. Inversamente, para que esta mesma informação possa ser recuperada na outra parte, é usado o processo reverso chamado **demodulação**. Por meio destas técnicas é feito um melhor uso do canal de comunicação, o que possibilita transmitir mais informações simultaneamente, além de melhorar a resistência contra possíveis ruídos e interferências.

Dependendo do parâmetro sobre o qual se variem as características da onda portadora, existem vários tipos de modulação, por exemplo, a modulação em amplitude (AM), modulação em fase (PM), modulação em frequência (FM), entre outros. Na modulação AM, a frequência é constante e a amplitude varia junto com a amplitude do sinal modulante. Já na modulação FM, a amplitude é constante enquanto que a frequência varia de acordo com a amplitude do sinal modulante. Além disso, para cada uma destas técnicas de modulação há o processo inverso de recuperação da informação, denominado de demodulação AM e demodulação FM, respectivamente.

Em processamento digital de sinais, a **modulação** AM com portadora suprimida é uma prática comum prévia à transmissão dos sinais pelos conversores de radiofrequência **RF DACs**. Este processo implica uma série de modificações que incluem a multiplicação do sinal modulante $F_m(t)$ com uma onda portadora $F_p(t)$. Assim, o sinal modulado é a multiplicação de uma onda portadora com um sinal modulante (o qual contém a informação a ser transmitida).

Para exemplificar, é dado um sinal modulante no domínio do tempo representado pela função *Cosseno* $F_m(t) = A_m \cdot \cos(\omega_m \cdot t)$ (Figura 4-B) com as configurações de frequência ω_m entre -3KHz a 3KHz e amplitude $A_m = 20$. Por outro lado, dada uma função portadora cossenoidal, no domínio da frequência, $F_p(t) = A_p \cdot \cos(\omega_p \cdot t + \phi(t))$ com amplitude $A_p = 1$, fase $\phi(t) = 0$, frequência ω_p sintonizada em 1MHz , conforme está na Figura 4-A. Para simplificar, a frequência é mostrada em módulos, representando o lado positivo do espectro de frequência.

A obtenção do sinal da portadora modulada em amplitude $F_{pma}(t)$ é feito por meio da multiplicação da função portadora com a função do sinal modulante, $F_{pma}(t) = F_p(t) \cdot F_m(t)$. De fato, ocorrem as seguintes transformações:

$$F_p(t) \cdot F_m(t) = [A_p \cdot \cos(\omega_p \cdot t + \phi(t))] \cdot [A_m \cdot \cos(\omega_m \cdot t)] \quad (2.1)$$

Considerando a fase $\phi(t) = 0$, para que não ocorra inversão de fase da portadora, logo:

$$\begin{aligned} F_{pma}(t) &= A_p \cdot A_m \cdot t \cdot [\cos(\omega_p) \cdot \cos(\omega_m)], \\ F_{pma}(t) &= A_p \cdot A_m \cdot t \cdot \left[\frac{1}{2} \cdot \cos(\omega_p - \omega_m) + \frac{1}{2} \cdot \cos(\omega_p + \omega_m) \right]. \end{aligned}$$

Dado que A_p é uma constante igual a 1, então:

$$F_{pma}(t) = A_m \cdot t \cdot \frac{1}{2} \cdot [\cos(\omega_p - \omega_m) + \cos(\omega_p + \omega_m)] \quad (2.2)$$

O resultado está ilustrado na Figura 4-C, onde a frequência do sinal modulante (mensagem a ser transmitida) foi transladada ao redor da frequência da portadora $\omega_p - \omega_m < \omega_{pma} < \omega_p + \omega_m$ com banda lateral dupla e portadora suprimida.

2.6 Demodulação

A **demodulação** é o processo inverso à modulação (AM). Essa técnica consiste em extrair o envoltório da portadora modulante em amplitude que contém o perfil do sinal modulante.

Para recuperar o sinal modulante aplicam-se duas etapas. Na primeira etapa é feita a multiplicação de dois sinais, que multiplica o sinal da portadora modulante em amplitude com o sinal do gerador NCO (Figura 5-A), como será detalhado na Seção 2.6.1. A segunda etapa consiste em pegar o sinal resultante dessa multiplicação e passar por dois filtros passa-baixa (Figura 5-B), a qual será descrita na Seção 2.6.2. O resultado da eliminação dos sinais com frequências superiores a 1MHz e o resgate do sinal, que esteja a uma frequência inferior a 1MHz, está representado na Figura 5-C.

2.6.1 Misturador de sinais

O **misturador** é um tipo de circuito que faz a multiplicação de dois sinais.

Seguindo o exemplo dado na Seção 2.5, aqui usaremos o sinal da portadora modulada em amplitude, na Figura 6-B, que é multiplicado como o sinal da portadora NCO conforme a Figura 6-A.

O resultado dessa mistura é um sinal com a frequência transladada em duas novas regiões de frequência, como ilustrado na Figura 6-C. A primeira região encontra-se próximo a duas vezes a frequência da portadora ω_p . A segunda região encontra-se à frequência ω_m , próximo de 0KHz, que corresponde ao sinal modulante.

Figura 4 – Exemplo de modulação de um sinal *Cossenoidal*. A) A função portadora $A_p \cdot \cos(\omega_p \cdot t)$ no domínio do tempo, frequência $\omega_p = 1\text{MHz}$ e amplitude em $A_p = 1$. B) Sinal modulante *Cossenoidal* em função do tempo com amplitude de $A_m = 20$ e frequência $\omega_m = 3\text{KHz}$. Este sinal representa a informação a ser transmitida pelo conversor **RF DAC**. C) Portadora modulante em amplitude, $F_{pma}(t)$, que é o resultado da multiplicação dos dois sinais (A) e (B) no domínio do tempo t . No lado direito de (C) é mostrado o resultante da transformada de Fourier no domínio da frequência, onde o sinal modulante foi transladado ao redor da faixa de frequência da portadora $0,997\text{MHz} < \omega_{pma} < 1,003\text{MHz}$.

Fonte: Elaborada pelo autor.

Figura 5 – Exemplo da filtragem de um sinal. A) Resultado da mistura da portadora modulada em amplitude com o sinal cossenoidal da portadora resultando em um sinal que contém duas frequências, sendo $\omega_{misturador}$ e ω_m . B) Resposta do filtro, a qual deixa passar os sinais com frequências inferiores a 1MHz como ω_m . C) Representa a primeira filtragem e a reconstrução do sinal modulante que está numa frequência ω_m .

Fonte: Elaborada pelo autor.

Após a execução da multiplicação do sinal modulado $F_{pma}(t) = A_p \cdot A_m \cdot t \cdot [\cos(\omega_p) \cdot \cos(\omega_m)]$ com a portadora $F_p(t) = A_p \cdot \cos(\omega_p \cdot t)$, isso resulta que a amplitude do sinal misturado seja dividido pela metade.

Figura 6 – Exemplo do funcionamento do misturador. A) Sinal da portadora com frequência sintonizada em $\omega_p = 1\text{MHz}$. B) O sinal da portadora modulante em amplitude com frequência $0,997\text{MHz} < \omega_{pma} < 1,003\text{MHz}$ e amplitude em 10. C) O resultado da mistura da portadora modulada em amplitude com o sinal cossenoidal da portadora. Trata-se de um sinal que contém duas novas frequências: O sinal envoltório (sinal de interesse) que está com frequência $\omega_m = 3\text{KHz}$ e o sinal misturado que está na banda de frequência de $1,997\text{MHz} < \omega_{misturador} < 2,003\text{MHz}$, no domínio da frequência.

Fonte: Elaborada pelo autor.

2.6.2 Filtros

Na área de processamento digital de sinais, a filtragem é um recurso frequentemente utilizado e está presente na robótica, processamento de áudio e imagens e vídeo, biomedicina, comunicação de informações, entre outras.

Existem diferentes tipos de filtros bem conhecidos. Por exemplo, o filtro passa-baixas é usado para atenuar frequências acima de 1MHz, permitindo a passagem de sinais com frequências inferiores a 1 MHz, abaixo da frequência de corte do filtro conforme ilustrado na Figura 5. De forma inversa, o filtro passa-altas atua eliminando sinais que esteja abaixo da frequência de corte. Já o filtro passa-banda (ou passa-faixa) atua com duas frequências de corte, assim delimitando as bandas de frequências que podem passar. Neste caso deixa passar todas as faixas de frequência que estão entre as frequências de corte e atenuando as demais frequências fora dessa faixa.

Em sistemas embarcados, estes mesmos filtros são desenhados como estruturas digitais. Nesta seção serão detalhados os filtros FIR (do inglês, *Finite Impulse Response*), CIC (do inglês, *Cascaded Integrator-Comb*) e CFIR (ligação em série do filtro CIC e FIR), os quais atuam como filtros passa-baixa. Como mostrado na demodulação é necessário extrair o sinal modulante da modulação, assim há necessidade de filtrar o sinal.

2.6.2.1 Filtro CIC decimador

O filtro **CIC**, conhecido também como filtro de Hogenauer, possui a vantagem de não precisar de multiplicadores, o que resulta numa resposta do filtro de baixa latência. O filtro CIC tem uma resposta bem conhecida no formato pente, este poderá atuar principalmente na zona da região sensível, que é próxima da região de frequência de corte do filtro, evitando cálculos complexos de multiplicação. O CIC é constituído por três partes básicas que são o integrador, a decimação D e o diferenciador (*comb*), que estão representados na Figura 7.

Figura 7 – Estrutura do filtro CIC é composta pelo integrador, decimador e o diferenciador (*comb*).

Fonte: Elaborada pelo autor.

O primeiro bloco, da esquerda para a direita, é o **integrador**, que consiste de um filtro IIR (*Infinite Impulse Response*, (MEDDINS, 2000)) de um polo simples. Sua resposta impulsional é dada pela Equação 2.3, e sua resposta frequencial é de tipo passa-baixa.

$$y(n) = x(n) + y(n-1). \quad (2.3)$$

No segundo bloco é visualizado o **decimador** D . A decimação é uma técnica que permite que um sistema com frequência de amostragem mais baixa possa receber o sinal. Esta técnica consiste na diminuição da frequência de amostragem na saída de um sistema. Para isso, é usado o fator de decimação, que é simplesmente um valor inteiro que representa a razão entre a taxa de entrada e a taxa de saída, garantindo que seja acima da sua taxa de Nyquist (CROCHIERE; RABINER, 1981). A maior motivação para usar esta técnica é a redução do custo do processamento de sinais (MA *et al.*, 2014) que é quase proporcional à taxa de amostragem; portanto, o uso de uma taxa de amostragem mais baixa geralmente resulta em uma implementação mais barata.

O terceiro bloco é o filtro **Comb**, também chamado de circuito **diferenciador**. Este bloco é um filtro FIR do tipo passa-altas e sua arquitetura se encontra na Figura 7 com resposta impulsional:

$$y(n) = x(n) - y(n-1). \quad (2.4)$$

Observe que o diferenciador (*comb*) opera numa frequência de relógio diferente da frequência do relógio do integrador, e ambos os circuitos atuam como blocos individuais separados por um decimador (D). Esse filtro tem resposta rápida ao impulso do sinal e atua eliminando as frequências superiores a 1MHz.

2.6.2.2 Filtro CFIR

A função que define o filtro FIR é dada pela Equação (2.5). Nesta equação, os sinais de entrada interagem com a sequência de resposta ao impulso do filtro através do processo de convolução linear.

$$y(n) = x(n) * h(n) = \sum_{k=0}^{N-1} h(k).x(n-k), \quad (2.5)$$

onde $x(n)$ é o sinal de entrada dos n bits e $y(n)$ é a saída do sistema dos n bits. $h(n)$ é a resposta ao impulso do filtro, ou seja, o coeficiente para cada bit n . A quantidade de coeficientes indica o quanto é abrupta a separação entre as bandas desejadas e a não desejada. Isto é, quanto maior a quantidade de coeficientes, mais rápida ocorre a separação entre as bandas. Além disso, o filtro FIR têm uma duração finita de valores na

saída $y(n)$, portanto, sua resposta ao impulso regressa nula após passar um determinado tempo.

A forma generalizada do filtro FIR com N estágios requer N iterações de multiplicações e somadores. Os seguintes parâmetros estão indicados na Figura 8, onde $h(N-1)$ representa a matriz dos coeficientes do filtro e $x(n-N+1)$ é a matriz dos dados de entrada. Os somatórios são representados por setas apontando para os pontos circulares e as multiplicações são indicadas pelos coeficientes $h(N-1)$.

Figura 8 – Representação da forma transposta da estrutura do filtro FIR.

Fonte: Elaborada pelo autor.

O filtro digital **CFIR** é também um filtro FIR, a diferença é que este filtro opera em conjunto com o filtro CIC, primeiro o CIC e depois o CFIR, nesta ordem. Os filtros CFIRs são estruturas utilizadas tipicamente para transformar atributos de um determinado sinal, seja no domínio da frequência ou no domínio do tempo. Essas estruturas possuem diferentes funções, por exemplo, a separação e a reconstrução de um sinal. O CFIR é um filtro passa-baixa pois permite a passagem das frequências que estão abaixo de um limiar de corte, enquanto que as demais faixas de frequência, acima do limiar de corte, são atenuadas na banda de rejeição do filtro. Essas medidas de banda de passagem, bandas de rejeição e frequência de corte são parâmetros que devem ser bem definidos antes de projetar o filtro (LYONS, 2011).

O processo de filtragem é ilustrado na Figura 9. Nesta figura é mostrada a etapa de modulação que inclui multiplicação do sinal da portadora com o sinal modulante. A etapa de demodulação inclui a multiplicação do sinal da portadora modulante de amplitude com a portadora, resultando assim num sinal misturado. Em seguida, é realizado o processo de filtragem usando o filtro CIC e CFIR e tendo como resultado o sinal recuperado.

Esse exemplo mostra um cenário que testa a transmissão e recepção com o uso do simulador **ModelSim**. Observe-se que todos os sinais estão na mesma escala conforme é

Figura 9 – Simulação do filtro CIC e, em seguida, do filtro CFIR, nesta ordem. O sinal corresponde a um dado manipulado para estímulo que passa pelos dois filtros simultaneamente. O ModelSim foi usado para a simulação.

Fonte: Elaborada pelo autor.

ilustrado na Figura 9, complementado o entendimento da modulação e demodulação.

Na Figura 9-A, a portadora NCO opera com uma frequência superior a 1MHz durante a primeira parte. Já na segunda parte, o NCO opera com uma frequência inferior a 1MHz. Na Figura 9-B, o sinal modulante, que é a mensagem a ser transmitida, encontra-se numa banda de frequência de -1MHz a 1MHz .

Na Figura 9-C, o sinal da portadora modulante em amplitude, que está com a frequência transladada ao redor da frequência da portadora e sua amplitude, está metade da amplitude do sinal modulante (Figura 9-B).

A Figura 9-D representa o sinal misturado, onde ocorreu a multiplicação do sinal da portadora modulante em amplitude com o sinal da portadora. Observe-se que sua amplitude é $1/2$ da amplitude do sinal modulado (Figura 9-C).

Na Figura 9-E é mostrado o sinal resultante ao aplicar o filtro CIC, o qual filtrou as frequências acima de 1MHz do sinal misturado. Esse filtro é de resposta rápida ao impulso e a amplitude desse sinal é $1/4$ da amplitude do sinal modulante (Figura 9-B).

O que complementa a necessidade de usar o controle de ganho VGA para amplificar o sinal modulado que chega nos conversores **RF ADCs**.

Na Figura 9-F é mostrado o sinal resultante do filtro CFIR, que filtra as frequências acima de 1MHz que o filtro CIC deixou passar. A resposta desse filtro gera atraso em sua resposta e atua com filtro corretor do filtro CIC conforme ilustrado na Figura 9-J.

A Figura 9-G é uma região que contém a mensagem a ser transmitida. A Figura 9-I é a região da primeira filtragem do filtro CIC, quando o NCO está configurado com uma frequência acima de 1MHz.

A Figura 9-J é a região da primeira filtragem do filtro CIC quando o NCO está configurado com uma frequência inferior a 1MHz. A frequência do NCO está próximo da frequência de corte do filtro CIC, por isso deixou passar.

A Figura 9-H é região da segunda filtragem do filtro CFIR, atuando como filtro corretor do filtro CIC. Observe-se que o sistema teve um baixo desempenho quando trabalha com o gerador NCO a uma frequência inferior a 1MHz e podemos concluir que o gerador NCO precisa operar numa banda de frequência superior a 1MHz dentro dos limites de operabilidade aceitáveis do conversor **RF DACs** e **RF ADCs**.

A Figura 9-T₁ é a região que representa uma alta latência com relação ao sinal Figura 9-D, sinal misturado.

A Figura 9-T₂ é a região que representa uma baixa latência com relação ao sinal Figura 9-D, sinal misturado.

2.7 Considerações finais

Neste capítulo foram apresentados os principais aspectos teóricos usados para o desenvolvimento deste trabalho. A maior parte dos fundamentos tecnológicos dão enfoque ao desenvolvimento em *hardware* e processamento de sinais (filtros). É importante notar que tanto os filtros CFIR e CIC atuam como filtros passa-baixas, enquanto que os conversores **RF DACs** e **RF ADCs** atuam como filtros passa-banda. Na última parte do capítulo foi detalhada a arquitetura e comunicação das placas da CIERMag que são objeto de estudo deste trabalho. No Capítulo 3 e Capítulo 4 serão apresentadas as metodologias para o desenvolvimento de *hardware* e *software* proposto.

DESENVOLVIMENTO DO HARDWARE

Neste trabalho foi proposto um sistema que está dividido em duas partes: o **hardware** (HW), desenvolvido na FPGA (a ser descrito neste capítulo); e o **software** (SW), executado no sistema operacional do processador, que carrega a interface de usuário (a ser detalhada no Capítulo 4).

A Figura 10 ilustra a abstração do fluxo de dados entre o SW e HW e sua comunicação com a placa da CIERMag. Cada um dos componentes do *software* encarrega-se de diferentes funcionalidades: (i) comunicação entre o SW e o HW; e (ii) transmissão e recepção de dados entre a FPGA e a placa para teste. Por outro lado, o HW é encarregado de: (i) gerenciar os periféricos de entrada e saída (I/O) da FPGA e do processador; e (ii) transmitir e receber os sinais de radiofrequência.

Figura 10 – Fluxograma do sistema das partes do *software* e *hardware* e a placa para teste.

Fonte: Elaborada pelo autor.

3.1 Descrição da implementação do *hardware*

Para o desenvolvimento do hardware foi utilizada a placa FPGA modelo Cyclone V SoC 5CSXFC6D6F31 (ALTERA®, 2015) juntamente com o sistema operacional Linux LXDE Desktop versão Linaro gravado em um *Micro SD card* e inserido no periférico de I/O do processador. Desta maneira, inicia-se carregando o projeto da FPGA, a qual estabelece a comunicação de vídeo através de módulos dedicados a essa funcionalidade do *hardware*, em seguida é carregado o sistema operacional, e conclui carregando o SW (responsável do controle do HW).

A escolha do modelo da FPGA foi motivada por três razões: (i) por estar disponível na CIERMag, (ii) por cumprir com os requisitos de conexão HSMC para comunicar-se com a placa para teste e (iii) por ser um sistema híbrido que permite a expansão do projeto com recurso de lógicas em FPGA e o desenvolvimento do *software* embarcado em um único *kit* e tudo isso aliado à portabilidade.

Na Figura 11 pode-se observar a placa FPGA, dentro dela é destacado em amarelo o *chip* da FPGA + HPS (visualizada usando a função de *Chip Planner* do Quartus II). Ao amplificar a imagem deste *chip* pode-se observar a distribuição dos módulos usados neste sistema, os quais desempenham funções específicas, por exemplo, a comunicação entre o SW e o HW. Esta distribuição é dividida para o módulo **Soc_system** (fornecido pelo fabricante da placa (TERASIC, 2019)), assim como para os submódulos contidos no módulo **RF** implementados neste trabalho.

O reuso do módulo **Soc_system** foi de extrema importância pois serviu de base inicial para apoiar o desenvolvimento deste trabalho, permitindo ao processador controlar alguns periféricos de I/O, assim como dos próprios periféricos de I/O da FPGA. Além disso, este módulo permitiu a configuração e ajuste do módulo de comunicação de dados entre o SW e HW implementados neste trabalho.

Ainda na Figura 11 é apresentada uma visão mais concreta do sistema desenvolvido. Nesta ilustração pode ser observada a conexão da FPGA com a placa para teste por meio dos periféricos I/O, assim como da conexão com um monitor, o qual mostra a interface de usuário. Também podem ser observados os barramentos e sinais de controle que foram criados para possibilitar a comunicação de dados/comandos dos módulos desenvolvidos com a FPGA e o *software*.

Na região da Figura 11-A é visualizada a comunicação de dados entre a FPGA e o processador HPS, para o qual foi usada a ferramenta Qsys para facilitar a integração entre estes dois sistemas. De fato, o Qsys permitiu criar e configurar um barramento bidirecional de fluxo de dados/comandos entre a FPGA e HPS (ver Figura 12). Os sinais *writedata* e *readedata* formam parte do conjunto de sinais que controlam o envio e recepção dos dados. Quando o sinal *writedata* está em nível lógico zero, é habilitado o

Figura 11 – Visão detalhada do sistema desenvolvido.

Fonte: Elaborada pelo autor.

estado de gravação de dados no sentido do SW → FPGA. Entretanto, o estado de leitura ocorre quando o sinal *readeddata* está no nível lógico alto, que é o instante em que ocorre a leitura de dados no sentido da FPGA → SW.

Figura 12 – Fluxo de dados bidirecional entre o *software* e a FPGA usando o módulo **Soc_system** numa visão RTL simplificada.

Fonte: Elaborada pelo autor.

Ainda na Figura 11-B pode-se observar a distribuição dos módulos de transmissão e recepção de dados entre a FPGA e a placa para teste. Na região de cor vermelha estão os módulos responsáveis pelo controle e gerenciamento dos periféricos de I/O e os módulos da comunicação de dados entre a FPGA e o processador. Finalmente, na região de cor roxa está o módulo responsável por fazer a transmissão e a recepção dos sinais de RF. A

seguir, na Tabela 2, é apresentada a lista dos módulos e submódulos desenvolvidos neste trabalho, agrupados por módulos que tratam dos sinais do módulo **RF (Transmissão RF e Recepção RF)**, os quais serão detalhados no decorrer deste capítulo.

Tabela 2 – Módulos e submódulos de transmissão e recepção de radiofrequência desenvolvidos neste trabalho.

Módulo	Submódulo	Descrição	
RF (Seção 3.3)	Transmissão RF (Seção 3.3.1)	Envia	O <i>software</i> envia os dados/comandos para os módulos da FPGA (Seção 3.3.1.1).
		RAM RF	Armazena os dados referentes ao pulso gerado pelo <i>software</i> (Seção 3.3.1.3).
		Controla RAM	Controla o estado de gravação, leitura e outras funções da memória (Seção 3.3.1.2).
		Modulação	Faz a multiplicação dos pontos do pulso com a função selecionada do NCO (Seção 3.3.1.4).
		NCO	Gera as funções de <i>seno</i> e <i>cosseno</i> (Seção 3.3.1.5).
		Offset DAC	Encarrega-se de fazer a conversão do sinal complemento de 2 para sinal binário como <i>offset</i> deslocado (Seção 3.3.1.6).
	Recepção RF (Seção 3.3.2)	Controle VGA	É responsável de programar o ganho ou atenuação no CI do VGA (Seção 3.3.2.1).
		Data ADCs	Recebe os dados dos conversores RF ADCs e transmite para o módulo Offset ADCs (Seção 3.3.2.2).
		Offset ADCs	Converte os sinal binário como <i>offset</i> deslocado para complemento de 2 para ser usado no módulo Misturador (Seção 3.3.2.3).
		Misturador	Faz a multiplicação do sinal modulado com a mesma função do NCO que foi transmitida (Seção 3.3.2.4).
		Filtro CIC	Filtra os dados removendo frequências superiores a 1MHz (Seção 3.3.2.5).
		Filtro CFIR	Filtragem dos dados com correções (Seção 3.3.2.6).
		FIFO	Armazena os dados do teste (Seção 3.3.2.7).
		Recebe	O SW recebe os dados armazenados na FIFO (Seção 3.3.2.8).

Fonte: Elaborada pelo autor.

3.2 Comunicação de dados entre o *software*, a FPGA e a placa para teste

Neste trabalho foram implementados dois tipos de fluxo de sinais de RF, resumidos a seguir:

- **SW → FPGA → Placa para teste:** exhibe todo o percurso do sinal de RF. O SW aciona a transmissão de pulso, em seguida deslocando-se pela FPGA (e seus respectivos módulos de **Transmissão RF**) e, por fim, alcançando a placa para teste.

- **Placa para teste** → **FPGA** → **SW**: trata do retorno do sinal de RF via conexão *loopback*, passando pelo controle de ganho do **VGA**, pelo conversor de **RF ADCs** e pelos módulos de **Recepção RF**, até chegar ao módulo **FIFO** e finaliza quando os dados são enviados para o SW.

Na Figura 13-1 pode-se observar o barramento de comunicação de dados e comandos entre o processador HPS e a FPGA.

O módulo **Envia** encarrega-se de enviar os dados/comandos no sentido de (SW → FPGA); e para inverter esse fluxo é usado o módulo **Recebe** que é responsável por receber os dados/comandos no sentido (FPGA → SW), enquanto que, o conjunto de barramento sentido FPGA ↔ placa para teste estabelece o fluxo de dados/comandos bidirecional que está localizado na Figura 13-2.

Figura 13 – Arquitetura da comunicação entre o *software*, a FPGA e a placa para teste. Os blocos azuis representam os módulos que comunicam-se com outros seguindo o fluxo de dados de transmissão TX (setas brancas) e recepção RX (setas cinzas), e seus respectivos barramentos representados por círculos amarelos. Os blocos pretos representam os conversores e o CI da **VGA**.

Fonte: Elaborada pelo autor.

Observe que cada cor de bloco no diagrama tem uma interpretação específica. Os blocos de cor laranja correspondem aos módulos de megafunção Quartus II. Os blocos azuis correspondem aos módulos que foram desenvolvidos neste trabalho. Os blocos pretos representam os conversores e o circuito integrado **VGA**. Finalmente, os blocos verdes correspondem à placa para teste desenvolvida pela CIERMag, que contém os conversores **RF DACs**, **RF ADCs**, chip controle de ganho do **VGA**. Além disso, os barramentos são representados por círculos em amarelo (A, B, C e D), descritos na Tabela 3.

Tabela 3 – Lista de barramentos usados na arquitetura desenvolvida.

Barramento	Descrição
A	Envio de dados do SW para a memória RAM RF .
B	Transmissão do sinal de RF para os conversores RF DACs .
C	Recepção do sinal de RF dos conversores RF ADCs .
D	Transferência de dados da FIFO para o <i>software</i> .

Fonte: Elaborada pelo autor.

3.3 Transmissão e recepção do sinal de RF

O módulo **RF** trata da transmissão e recepção dos sinais de radiofrequência. Para facilitar o entendimento, na Figura 14-A é ilustrado o percurso do sinal de RF através dos sub-módulos pertencentes ao módulo **Transmissão RF** sentido $1 \rightarrow \dots \rightarrow 6$, enquanto que os módulos de **Recepção RF** seguem o sentido $7 \rightarrow \dots \rightarrow 13$. Complementarmente, na Figura 14-B e na 14-C são ilustrados os resultados dos sinais nas etapas correspondentes da Figura 14-A.

O procedimento da transmissão de RF, sentido SW $\rightarrow$ FPGA, ocorre seguindo as etapas, da esquerda para a direita, na Figura 14-A:

- [1] os dados/comandos inicializam no barramento **A**. Os dados são transmitidos para o módulo **RAM RF** e os comandos para os módulos **Controla RAM**, **NCO** e **Controle VGA**;
- [2] o módulo **Controla RAM** gerencia a leitura e a gravação do sinal de RF armazenando-o no módulo **RAM RF**;
- [3] ocorre o envio dos dados armazenados na **RAM RF** para o módulo **Modulação**. Esse envio é gerenciado pelo módulo **Controla RAM**;
- [4] o módulo **Modulação** faz a multiplicação do sinal modulante de RF, que vem da memória, com a função portadora do **NCO**;
- [5 – 6] o conjunto de sinais é convertido de sinal complemento de 2 para um sinal binário com *offset* deslocado (sinal com valores positivos) ao passar pelo módulo **Offset DAC**. O sinal modulado está em transição da FPGA $\rightarrow$ placa para teste com sentido ao conversor **RF DACs** pelo barramento **B** (TX).

O procedimento de recepção de sinal de RF, desde a transformação do sinal analógico em digital até a recepção do pulso pelo SW, segue as seguintes etapas, da direita para a esquerda, na Figura 14:

Figura 14 – A) Módulos de transmissão e recepção do sinal de RF. B) Ilustração do sinal modulante passando pelos módulos de **Transmissão RF**. C) Ilustração do sinal portadora modulante em amplitude passando pelos módulos de **Recepção RF**. Os números correspondem à ordem do fluxo de dados de 1 a 6 para TX (setas brancas) e de 7 a 13 para RX (setas cinzas). Os barramentos são representados por círculos amarelos.

Fonte: Elaborada pelo autor.

- [7] o sinal de RF ingressa pelo conector HSMC e passa pelo barramento **C** (RX). Em seguida, o módulo **Data ADCs** captura e estabiliza os dados sincronizando-os com o *clock* da FPGA;
- [8] o sinal sofre uma conversão ao passar pelo módulo **Offset ADCs**, passando de binário com *offset* deslocado para sinal complemento de 2 (dados sinalizados);
- [9 – 10] o módulo **Misturador** faz a multiplicação dos dados já modulados pela recepção com a função portadora do **NCO**;
- [11 – 13] o sinal passa pelo **filtro CIC**, em seguida pelo **filtro CFIR**, e logo após, é armazenado em uma **FIFO** e em seguida é transmitido pelo barramento **D** para o *software*.

Na Figura 15 é exemplificada a transmissão de pulso de RF gerado no *software*, passando pela FPGA até a placa para teste, e cuja validação pode ser verificada por meio

do osciloscópio. Na interface de usuário foram gerados dois pulsos com função *seno* e *sinc*, Figura 15-1 e Figura 15-2, e foram transmitidas usando os canais **RF DAC-A** e **RF DAC-B**, respectivamente. As características de duração, frequência e amplitude são controladas pela interface. O osciloscópio mostra as correspondentes saídas, ou seja, a modulação do *seno*, e uma *sinc* modulada.

Figura 15 – Ambiente de experimentação da transmissão de radiofrequência. No lado esquerdo, o monitor com a interface de usuário configurado com dois sinais RF. No lado direito, o osciloscópio mostrando os sinais de RF sendo transmitidos nos dois conversores de **RF DACs**. Na parte inferior está a placa para teste conectada à FPGA.

Fonte: Elaborada pelo autor.

Nas próximas seções serão detalhados cada um dos módulos e submódulos descritos na Tabela 2, agrupados pelos módulos de **Transmissão RF** e **Recepção RF**.

3.3.1 Módulos da transmissão de sinal de radiofrequência

3.3.1.1 Módulo Envia

O módulo **Envia** foi desenvolvido na FPGA para receber os dados/comandos que são transmitidos pelo *software*. Este módulo é responsável por retransmitir os dados para o módulo **RAM RF** e comandos para o módulo **Controla RAM**. O módulo **Envia** roteia o fluxo de dados/comandos para o módulo de destino, pois cada módulo de memórias/controles tem identificador associado a eles, a qual é usada para traçar a rota final de dados/comandos.

3.3.1.2 Módulo Controla RAM

O módulo **Controla RAM** é responsável por gerenciar a gravação/leitura dos dados no módulo **RAM RF**. Além disso, controla as funcionalidades de: (i) *Loop*, exe-

cutando várias leituras que resultam na transmissão de dados para os conversores **RF DACs**; (ii) *Clear*, esvaziando todo o conteúdo da memória **RAM RF** e eliminando possíveis resíduos de pulso gravado; (iii) *Stop*, parando o modo de *Loop* inicializado; e (iv) complementando o gerenciamento de leitura e gravação do módulo **FIFO**. Observe-se que o sistema consegue gerenciar melhor as funcionalidades de *Loop*, *Stop* e *Clear* e outras funções numa memória **RAM RF** dual *clock* ao invés do uso de uma **FIFO**.

3.3.1.3 Módulo RAM RF

O módulo **RAM RF** é uma memória que armazena os sinais modulantes de RF que serão transmitidos pelos conversores **RF DACs**. A **RAM RF** é uma memória dual *clock* que funciona da seguinte maneira: (i) usa a sincronização do *clock* de gravação do processador para receber os dados (SW → FPGA); e (ii) usa a sincronização do *clock* da FPGA quando os dados são enviados para os demais módulos até a saída do conversor **RF DACs**, onde cada canal tem um **RAM RF** dedicado.

A memória da **RAM RF** tem uma profundidade de 16383 (ou 2^{14}), sendo 14 bits para endereçamento de leitura/gravação e 16 bits para dados. A estruturação deste módulo está representada na Figura 16.

Figura 16 – A) Visão geral do funcionamento de uma memória dual *clock*. Os dados chegam do *software* para serem armazenados na memória. Observe que dados e endereços (Dados/Address) estão sincronizados com o *clock* do processador. B) Leitura da memória com o uso do *clock* da FPGA. Observa-se também que os dados e endereços estão sincronizados com o *clock* da FPGA.

Fonte: Elaborada pelo autor.

3.3.1.4 Módulo Modulação

O módulo **Modulação** é uma megafunção do Quartus II que foi adicionada no *hardware* proposto para realizar a multiplicação dos dados do pulso de RF (sinal modulante contido no módulo **RAM RF**) com a função portadora gerada pelo módulo **NCO**. O resultado é um exemplo de aplicação ilustrado na Figura 17, onde é utilizada a ferramenta Signal Tap II Logic Analyzer para analisar as entradas e saídas desses sinais em tempo real. O resultado mostra o módulo de **Modulação** recebendo sinais modulantes do módulo **RAM RF** (Figura 17-A) que está representado por uma *sinc*. Este sinal tem uma frequência central de 0KHz e está sendo multiplicado com a portadora gerada no **NCO** de frequência central de 3MHz (Figura 17-B). Cada ponto do pulso é multiplicado e o resultado é uma *sinc* portadora modulada em amplitude com uma frequência central de 3MHz (Figura 17-C). A magnitude representa a amplitude de cada sinal, neste exemplo não houve variação de amplitude.

Figura 17 – Módulo de **Modulação** exemplificando um pulso multiplicado com a função *seno* gerada pelo **NCO**. A) O módulo **RAM RF** transmite um sinal modulante de *sinc*, com frequência central $f_0=0\text{kHz}$ que esta numa banda de -2kHz a 2kHz . B) O módulo **NCO** gera um sinal de uma portadora senoidal com frequência central $f_1=3\text{MHz}$. C) O resultado é uma portadora modulante em amplitude de um sinal *sinc* com frequência central $f_2=3\text{MHz}$ que esta numa banda de 2.998MHz a 3.002MHz .

Fonte: Elaborada pelo autor.

3.3.1.5 Módulo NCO

O módulo **NCO** foi desenvolvido para gerar as funções portadoras de sinal *seno* e *coseno*. O sinal gerado pelo **NCO** é usado no módulo **Modulação**, onde ocorre a transmissão do sinal; e no módulo **Misturador**, onde ocorre a multiplicação do sinal de recepção.

É possível verificar o resultado do funcionamento do **NCO** na Figura 18, onde pode-se observar um sinal de frequência de 20KHz. Este módulo permite uma varredura de sinais com frequências até 20MHz e também é possível configurar outra função através da interface de usuário como *seno* e *coseno*.

Figura 18 – Sinais do módulo **NCO** via ferramenta Signal Tap II Logic Analyzer. O **NCO** gera duas portadoras, uma *seno*, para modular/misturar o sinal no primeiro canal; e outra *coseno* para modular/misturar o signal no segundo canal. A escolha de configuração pode variar como *seno* e *coseno* para canais independentes onde a frequência da portadora pode ser configurada via SW.

Fonte: Elaborada pelo autor.

3.3.1.6 Módulo Offset DAC

O módulo **offset DAC** pertencente ao módulo **Transmissão RF** é responsável pela conversão dos dados no formato de complemento de 2 (dados sinalizados) para sinal binário com *offset* deslocado (valores positivos de sinais). Esta conversão é necessária pois todos os conversores da placa para teste estão configurados para operar em binário com *offset* deslocado de acordo com o *data-sheet* (Texas Instruments, 2017). Esse percurso de transmissão segue a seguinte sequência:

FPGA (barramento A) → (**Controla RAM** → **RAM RF**) → (**Modulação * NCO**) → **Offset DAC** → barramento B).

A Figura 19 mostra a transformação do sinal desde o início da transmissão, onde os dados (sinal *sinc*) são armazenados no módulo **RAM RF** na Figura 19-A passando pelo conversor **RF DACs** que está localizado na Figura 19-B e finalizando com o sinal de RF sendo analisado no osciloscópio na Figura 19-C. Para essa representação foi realizada

uma validação com o uso do osciloscópio para indicar o correto funcionamento dos conversores **RF DACs**. O detalhamento dos componentes conversores que estão na placa para teste como **RF DACs**, controle de ganho **VGA** e **RF ADCs** serão detalhados no Apêndice A.

Figura 19 – Fluxo de dados da transmissão de sinais de RF. A) O sinal de RF (sinal modulante) flui pelo barramento A e é multiplicado por uma portadora gerada pelo módulo **NCO**. O resultado da multiplicação é um sinal de uma portadora modulante em amplitude que é transmitido até o barramento B da placa para teste, e (C) é visualizado através do osciloscópio. O módulo **NCO** é responsável em deslocar a frequência do sinal modulante.

Fonte: Elaborada pelo autor.

3.3.2 Módulos da recepção do sinal de radiofrequência

Na seção anterior foram abordados todos os módulos da **Transmissão RF**, desde que o SW envia dados/comandos pelo barramento A até o barramento B, indo para as placas para teste por meio do conversor **RF DACs** e sendo visualizado no osciloscópio.

Nesta seção serão abordados os módulos **Recepção RF** que tratam do pré-processamento e análise dos sinais de RF. Observe que neste cenário o módulo **Controle VGA** será apresentado em conjunto com os módulos de **Recepção RF**.

3.3.2.1 Módulo Controle VGA

O módulo **Controle VGA** é responsável por controlar/programar o CI da **VGA**. No início de uma transmissão, este módulo cumpre a função de atenuar ou amplificar o sinal que chega dos conversores **RF DACs** via *loopback*. Este módulo é responsável pela programação no **VGA**, enviando dados e sinais de controle seguindo o padrão de comunicação do protocolo SPI. O módulo **Controle VGA** está indicado na Figura 19-A com asterisco (*). O retorno do sinal segue a seguinte sequência:

FPGA (barramento B) → placa para teste (RF DACs → conexão via *loopback* → VGA → RF ADCs) → FPGA (barramento C).

3.3.2.2 Módulo Data ADCs

O módulo **Data ADCs** é o primeiro a receber o sinal de recepção do RF, como indicado na Figura 19-B. Este módulo foi implementado para receber dados do conversor **RF ADCs** da placa para teste e retransmitir para os submódulos de **Recepção RF** e desempenha a função de sincronizar os dados de entrada com o *clock* dos outros submódulos.

3.3.2.3 Módulo Offset ADCs

O módulo **Offset ADCs** pertencente ao módulo **RF** é uma megafunção que efetua a conversão dos sinais que estão no formato sinal binário com *offset* deslocado (valores positivos de sinais) em que os conversores operam e o transformam para complemento de 2. Essa conversão é necessária pois todos os módulos da FPGA trabalham com sinais em complemento de 2, ou seja, dados sinalizados.

3.3.2.4 Módulo Misturador

O módulo **Misturador** é uma megafunção que foi examinada no Capítulo 2. Este módulo realiza a multiplicação entre dois sinais de entrada. O primeiro sinal é gerado pela função portadora do **NCO** que possui a mesma configuração usada para fazer a modulação do sinal de RF na transmissão. O segundo sinal chega do conversor **RF ADCs** que está modulado, por essa razão, esse sinal dispõe de modificações ao passar do ambiente analógico para o ambiente digital.

Para exemplificar o funcionamento deste módulo são dados dois sinais na Figura 20-A e Figura 20-B). O resultado dessa multiplicação é um sinal de uma *sinc* de 16 bits que tem a metade da amplitude da portadora modulante em amplitude e contém duas frequências (Figura 20-C). Por um lado, a frequência de 0KHz, que corresponde ao sinal de interesse que deseja-se recuperar, e por outro lado, têm frequência de 6MHz, que corresponde à soma de ambos os sinais com $f_2=3\text{MHz}$ e $f_1=3\text{MHz}$.

Durante a aplicação do **Misturador** é possível obter-se a contaminação do sinal, isto se deve a dois possíveis motivos: (i) relação entre o sinal e o ruído, pois, quanto maior é o ganho do **VGA** implica em amplificação do sinal e também do ruído, como consequência a qualidade do sinal de recepção pode variar à medida que o ganho aumenta; (ii) à inserção de ruídos originados do ambiente externo ou da própria eletrônica (ruído branco).

Na Figura 20 também pode ser observado que o sinal que passa pelo módulo **Misturador** com resquícios de ruído branco, considerado muito baixo, porém existente. Contudo, esse ruído somado à variação de erro gerado pela conversão dos conversores RF DACs e RF ADCs resulta em pequenas variações na reconstrução do sinal.

Figura 20 – O módulo **Misturador** realiza a multiplicação dos sinais (A) e (B) resultando no sinal (C). (A) Sinal da portadora modulante em amplitude de 16 bits e frequência de 3MHz vindo do conversor RF ADCs que esta numa banda de 2.998MHz a 3.002MHz . (B) Portadora da função gerada pelo NCO de 16 bits e frequência de 3MHz. (C) O resultado dessa multiplicação é um sinal *sinc* de 16 bits com metade da amplitude de (A) e duas frequências: 0KHz correspondente ao sinal de interesse que esta numa banda de -2KHz a 2KHz ; e 6MHz que esta numa banda de 5.998MHz a 6.002MHz , correspondente à soma da frequências de (A) e (B).

Fonte: Elaborada pelo autor.

3.3.2.5 Módulo Filtro CIC

O módulo **filtro CIC** foi desenvolvido com o uso do Octave (EATON, 2020), na linguagem em VHDL e também com o uso do simulador ModelSim do próprio Quartus II. Este filtro é de ação rápida na sua filtragem por isso está posicionado antes do filtro **CFIR**. Sabe-se que o sinal resultante do módulo **Misturador** chega super-amostrado incluindo a *sinc* com uma frequência maior que a frequência de corte do filtro. O filtro

projetado é do tipo passa-baixas e a faixa da frequência que desejamos denotar deve estar nesta janela.

O sinal chega super-amostrado com uma taxa de amostragem de 50MHz e ao passar pelo filtro ocorre uma decimação, ou seja, a taxa de amostragem passa para 1MHz. Essa frequência resultante de 1MHz controlará a inserção de dados no **filtro CFIR**, bem como a memória **FIFO**. Após 50 ciclos de *clock* ocorre a saída de um ponto por decimação. Na Figura 21 são apresentados alguns exemplos de transformação de dados, onde são exibidas as transformações após a filtragem.

Figura 21 – O módulo **Filtro CIC** é responsável por fazer a primeira filtragem. Esse módulo recebe um sinal misturado que contém duas frequências após passar pelo módulo **Misturador**. A frequência central f_2 de 3MHz com uma banda de 2.998MHz a 3.002MHz é um sinal modulado que chega dos conversores **RF ADC**, e é misturado com a frequência central f_1 do módulo **NCO** de 3MHz. O resultado é um sinal com duas frequência central $(f_2 - f_1) = 0\text{KHz}$ com uma banda de -2KHz a 2KHz , que é a frequência do sinal de interesse e outra de $(f_2 + f_1) = 6\text{MHz}$ numa banda de 5.998MHz a 6.002MHz. Esse sinal ao passar pelo **filtro CIC** sai filtrado com uma frequência central de 0KHz a uma banda de -2KHz a 2KHz .

Fonte: Elaborada pelo autor.

3.3.2.6 Módulo Filtro CFIR

O módulo **Filtro CFIR** é o filtro digital passa-baixas introduzido no Capítulo 2, o qual foi desenvolvido com a ferramenta do Octave (EATON, 2020) na linguagem em VHDL. Este filtro é usado para modificar a característica do sinal de entrada no domínio do tempo, fazendo correções de erros que possam ter passado pelo filtro **CIC**. O **filtro CIC** e o módulo **filtro CFIR** trabalham em conjunto e devem seguir esta ordem de associação.

Na Figura 22 é ilustrada a junção de dois filtros passa-baixas. A resposta do **filtro CIC** é mostrada na Figura 22-A, onde é possível verificar-se algumas imperfeições que podem ser resultado de outras frequências próximas a 1MHz. Após passar pelo **filtro CIC**, o sinal segue para o **filtro CFIR** onde essas imperfeições foram corrigidas, o qual

justifica estar a posição dos filtros nesta ordem. De fato, na Figura 22-A observa-se que ao aplicar o **filtro CIC**, e em seguida o **filtro CFIR** em conjunto, notamos um atraso quase insignificante. Ocorre o contrário quando apenas é aplicado o **filtro CFIR**, obtendo um atraso considerável.

Figura 22 – O sinal em análise é um pulso trapezoidal que passou pelo módulo **Misturador** e contém duas frequências neste sinal, as frequências de 0KHz e de 600kHz. Após passar pelo módulo **Filtro CIC** o sinal de interesse é reconstruído. O sinal após o **filtro CIC** contém um pouco de ruído onde o módulo **filtro CFIR** consegue fazer as correções necessárias. Observe que o **filtro CFIR** apresenta um atraso maior que o **filtro CIC** sendo analisado com uso da ferramenta Signal Tap II.

Fonte: Elaborada pelo autor.

Na Figura 22-A pode-se observar que o sinal filtrado pelo **filtro CIC** está ruidoso e com muitas imperfeições. Para demonstrar esse efeito foi realizado um teste com as seguintes configurações: (i) o sinal de RF foi transmitido com uma amplitude muito baixa, bem próximo do ruído branco; (ii) o número de pontos próximo aos 50 a qual permite visualizar uma janela menor no analisador; (iii) foi ajustada a frequência do **NCO** para 300KHz (frequência aceitável deve estar entre 1MHz a 20MHz); (iv) ganho do **VGA** ajustado para 34 dB. Este cenário força um nível crítico em que é possível verificar que o **filtro CIC** não obteve um bom desempenho, e deixa para o **filtro CFIR** o encargo de suavizar a filtragem (Figura 22-B).

3.3.2.7 Módulo FIFO

A **FIFO** é uma megafunção do Quartus II que é responsável por armazenar os dados da recepção de RF da análise da placa em teste. Esses dados chegam a uma taxa de 1MHz após passar pelo **filtro CFIR**, esse *clock* é responsável em injetar dados na **FIFO**.

Percebe-se que a retransmissão dos dados da FPGA → SW ocorre quando os dados são enviados com o *clock* do processador de $\approx 50\text{MHz}$.

3.3.2.8 Módulo Recebe

O módulo **Recebe** foi desenvolvido para enviar dados da FPGA → SW. Esse envio ocorre com a leitura dos dados armazenados no módulo **FIFO** sincronizado com o *clock* do processador de $\approx 50\text{MHz}$. O início de envio de dados ocorre quando o SW recebe a notificação da FPGA sobre o estado do conversor **RF ADCs** (saturado/não saturado). Essa funcionalidade de monitoramento usa um sinal enviado pelo próprio conversor, indicando se o conversor está saturado ou não. Esse percurso de recepção segue a seguinte sequência:

FPGA (barramento → **Data ADCs** → **offset ADCs** → (**Misturador** * **NCO** → **CIC** → **CFIR** → **FIFO** → barramento).

3.4 Considerações finais

Neste capítulo foi detalhada a arquitetura do *hardware* proposto que está constituído de duas grandes partes: a transmissão de sinais de RF e a recepção de sinais de RF. Na primeira parte foi apresentada uma visão geral da FPGA, apresentando todos os componentes e módulos desenvolvidos. Em geral, FPGAs são utilizadas em virtude da sua flexibilidade e eficiência, as quais permitem reduzir os custos bem como o período de desenvolvimento. Sua capacidade de reconfiguração é importante, pois permite o reaproveitamento dos recursos lógicos da FPGA em outras tarefas. O sistema foi projetado tendo em vista a comunicação com um processador HPS e FPGA, onde o *software* executa as tarefas seguindo um fluxo de execução sequencial e estabelece o sincronismo de comunicação com as lógicas dos módulos que operam de forma semiparalela ou paralela. Na segunda parte foram descritos cada um dos sub-módulos. No Capítulo 4 será introduzido o *software* desenvolvido, salientando os painéis de interface de usuário e ao mesmo tempo realizando a conexão com cada um dos módulos que foram descritos neste capítulo. Deve-se observar que, apesar de fugir do enfoque deste trabalho, também foram elaborados módulos que operam com sinais de gradientes, os quais serão detalhados no Apêndice B.

DESENVOLVIMENTO DE SOFTWARE

4.1 Visão geral da estruturação do *software*

Neste capítulo é apresentado o *software* proposto, o qual foi estruturado seguindo o modelo de três camadas: interface de usuário, regra de negócio e acesso a dados, como visto na Figura 23.

Figura 23 – Estrutura de comunicação de três camadas. A) Interface do usuário onde ocorre o controle de eventos e configurações de parâmetros. B-C) As chamadas de funções e gerenciamento das regras de negócios para que ocorra a comunicação de dados e comandos para a FPGA de forma sincronizada. D) Área de acesso de dados é responsável pela reconfiguração de parâmetros da interface e recebe as configurações do sinal feita pelo usuário.

Fonte: Elaborada pelo autor.

O *software* foi desenvolvido com integração entre várias linguagens. A camada de interface de usuário foi desenvolvida usando a ferramenta do Qt Designer na linguagem Python, enquanto que a camada da regra de negócio está dividida conforme se segue: uma parte em linguagem Python e uma parte em linguagem C, onde é possível ter acesso e controle no dispositivo de I/O da FPGA e a camada de acesso a dados, que são arquivos de armazenamento de dados referentes ao pulso e a configuração de parâmetros da interface.

A primeira camada corresponde à interface gráfica de usuário (Figura 23-A), a qual foi desenvolvida usando a ferramenta QT Designer. Nesta camada ocorrem as principais configurações de controle das operações no *hardware* e é responsável por exibir os dados de resultados do teste. Esta interface gerencia o sincronismo do funcionamento de cada subconjunto de módulos de controle ou conjunto de módulos que executam uma funcionalidade específica, com os módulos de **Transmissão RF** e **Recepção RF**. Este *software* é dedicado à funcionalidade desse *hardware* em específico e está interconectado à funcionalidade das placas desenvolvidas pela CIERMag, ambos trabalham de forma orquestrada.

A segunda camada apresenta as regras de negócio e é encarregada de receber os fluxos das configurações e das chamadas de funções provenientes da interface; bem como organizar e redistribuir a funcionalidade para os arquivos responsáveis. Essa camada está dividida em dois subgrupos (B) e (C).

Na subcamada (B) da Figura 23-B existem quatro arquivos desenvolvidos em Python que se encarregam da administração do fluxo de dados, gerenciam e distribuem funções aos arquivos responsáveis e manipulam os arquivos de geração de pulso. A subcamada da Figura 23-C é responsável pelo controle dos periféricos de I/O do processador e também da FPGA, da gravação dos resultados provenientes da FPGA e da leitura e envio dos pulsos para a FPGA.

- O arquivo *Principal* recebe dados do arquivo *RF* referente ao experimento de teste e retransmite para interface do usuário e há visualização do dado em forma de gráficos dos pontos recebidos do conversor **RF ADCs** no domínio do tempo. O arquivo *Principal* também recebe comandos e parâmetros de configurações e retransmite para o arquivo *RF*. Esses arquivos controlam o conjunto de módulos de **RF** na FPGA.
- O arquivo *RF* recebe parâmetros para a geração de pulso do arquivo *Principal* e envia para o arquivo *Pulso*, onde esse pulso é gerado, e o retorno da função no arquivo *RF* grava os arquivos de *input_DAC_A* e *input_DAC_B* referente aos dois RF DACs. O arquivo *RF* gerencia o recebimento dos resultados do teste automatizado via arquivo *Relatório* e o resultado do teste no arquivo *output_ADC*, e retransmite para o arquivo *Principal* e logo encaminha esses dados para a respectiva visualização na interface.

- O arquivo *Pulso* é responsável por receber os parâmetros dos pulsos e gerá-los para o arquivo *RF*.

A subcamada (C) da Figura 23-C é responsável pelas seguintes atribuições: controle dos periféricos de I/O do processador e também da FPGA; controle da gravação dos resultados provenientes da FPGA; e da leitura e envio dos pulsos para a FPGA e os arquivos *RF*. Esta subcamada é responsável pela leitura e transmissão do *input DAC-A* e *input DAC-B* para a FPGA. A gravação dos dados de retorno da FPGA estão nos arquivos *output ADC* e *Relatório*. Esses arquivos são lidos pelo arquivo *RF* e retransmitidos para o arquivo *Principal* para sua posterior visualização na interface.

Na terceira camada (Figura 23-D) estão todos os acessos a dados, arquivos referentes ao RF e por seguinte o carregamento e envio dos arquivos para a FPGA. A funcionalidade de RF detém os arquivos *input DAC-A* e *input DAC-B*. Para que seja possível a visualização dos dados na camada da interface de usuário, dispõe-se de arquivos gerados via experimentos de *output ADC* e *Relatório*, em que o arquivo *Config* atua diretamente nas configurações de parâmetros, lista de valores e funções da interface.

4.1.1 Tipos de pulsos

O *software* pode gerar oito tipos de sinais conforme a Figura 24, e cada um desses sinais podem conter distintas combinações de configurações como: (i) ajuste de amplitude até o limite máximo de 32767; (ii) ajuste do número de pontos até o limite máximo de 16383; e (iii) pode-se adicionar outros pulsos que o sistema vai conseguir transmitir.

4.2 Painel de transmissão e recepção de sinais de RF

A Figura 25 exibe a interface gráfica do *software* desenvolvido. A primeira aba contém o ambiente de configuração para a transmissão e recepção do sinal de radio-frequência, bem como os painéis para demais configurações: RF, plot, botões, VGA, NCO e pulso.

- **Painel de configuração de sinal de radiofrequência (RF):**
 - Permite escolher a entrada (DAC-A/ADC-A ou DAC-B/ADC-B) a ser utilizada selecionando o botão de opção. Uma vez que o plot do sinal DACs correspondente (painel do plot) é selecionado, ele apresentará uma borda verde indicando que está ativado;
 - Permite selecionar a opção de modulação (*modulation*) entre as diferentes funções do NCO: *seno* e *cosseno* que serão usadas para realizar a modulação na transmissão

Figura 24 – Alguns pulsos do sistema que podem ser configurados em amplitude, números de pontos e as formas do sinal listado abaixo.

Fonte: Elaborada pelo autor.

Figura 25 – Visão da interface gráfica do *software* que permite realizar o controle da configuração do pulso a ser transmitido, parâmetros de configuração da FPGA e ajuste no controle de ganho do VGA.

Fonte: Elaborada pelo autor.

e o uso do módulo Misturador com pulsos selecionados: *quadrado*, *trapézio*, *Gaussiana*, *triângulo*, *senoidal*, *seno*, *coseno quadrado* e *sinc*.

- **Painel de configuração do ganho do VGA:** permite escolher a opção de ganho alto (*High Gain*) ou ganho baixo (*Low Gain*), assim como permite o ajuste fino no ganho em dB na caixa de combinação;
- **Painel de configuração do NCO:** o oscilador NCO permite a configuração na frequência de oscilação do sinal da portadora, apresentando valores no *spinner* entre 10KHz a 20MHz, sendo o valor padrão de 2.5MHz;
- **Painel de configuração do pulso:** permite configurar o painel de pulsos seguindo os seguintes parâmetros:
 - Número de pontos: ajustar entre 10 a 16383 que é a capacidade máxima da memória **RAM RF**;
 - Amplitude: ajustar entre 10 a 32787, valores maiores que esse saturarão o conversor **RF DACs**;
 - Função de pulso: escolher as funções *sinc*, *seno*, *coseno quadrado*, *triângulo*, *trapézio*, *Gaussiana*, *quadrado* ou *senoidal* exibidas na caixa de combinação.
- **Painel de botões de plot:**
 - *Play*: aguardar a ocorrência da transmissão e plotagem do gráfico da recepção na tela correspondente ao canal selecionado **DAC-A/ADC-A** ou **DAC-B/ADC-B**;
 - *Loop*: após configurar o pulso e clicar no botão *Play*, e em seguida pressionar o botão *loop*, ocorrem inúmeras transmissões e recepções desse pulso de RF. Essa função é observável com auxílio do osciloscópio analisando este canal ou no modo de operação de teste automatizado que será detalhado no Capítulo 5;
 - *Stop*: esse botão finaliza todas as operações que a FPGA esteja executando;
 - *Read memory*: Permite verificar todo o conteúdo da memória fazendo uma varredura em todos os endereços;
 - *Clean*: permite reiniciar todos os parâmetros assim como esvaziar o conteúdo das duas memórias **RAM RF** do **DAC-A** e **RAM RF** do **DAC-B**;
 - *Update*: caso tenha alterado a pulso em algumas das opções essa função faz uma atualização do pulso de RF nos dois canais e mostra abaixo de cada janela do pulso as configurações atualizadas.

4.3 Painel de resultados

Na Figura 26 é possível observar as duas opções existentes para testes automatizados utilizadas para análise de resultados. As configurações comuns entre os dois experimentos são:

- Painel de RF, onde ocorre a seleção dos canais (**DAC-A/ADC-A** ou **DAC-B/ADC-B**) pertencente aos experimentos;
- *Number of tests*: É o número do testes (pulsos ou amostras) que será aplicado no experimento;
- O painel de parâmetros (*Parameters*):
 - Botão *Test*: inicia um conjunto de teste de acordo com o valor especificado no campo *Number of tests*;
 - Botão *Stop*: interrompe o teste inicializado;
 - Botão *Report*: logo após o término do teste é possível observar um relatório se como procederam esses testes. Deve-se selecionar uma opção de caixa de seleção, seleção de ganho ou ponto de saturação no painel *Graphic*;
 - Caixa de seleção *Gain*: mostra os valores de amplitude em que o sinal de recepção atingiu, de acordo com a configuração de ganho do **VGA**;
 - Caixa de seleção *Saturation points*: indica o instante que o conversor **RF ADCs** atingiu o ponto de saturação;
 - Caixa de combinação *Pulse*: escolhe dentre as funções *sinc*, *seno*, *cosseno quadrado*, *triângulo*, *trapézio*, *Gaussiana*, *quadrado* e *senoidal*;
 - Painel de *Initial configuration*: são mostrados os parâmetros para determinar a configuração do sinal de RF, por exemplo, amplitude, número de pontos, NCO, VGA e outros. Neste painel são indicados os valores iniciais padrão para o primeiro teste;
 - Painel de *N. steps*: O usuário define um valor positivo que corresponde ao incremento ou um valor negativo que corresponde ao decremento, assim permite efetuar atualizações nos valores de parâmetro (painel *Initial configuration*) a cada teste ocorrido. Quando o valor é zero, mantém-se constante os valores que foram configurados;
 - Painel de *Enable plot*: permite ativar um respectivo parâmetro (painel *Initial configuration*) e visualizar o correspondente plot incluindo os parâmetros selecionados no experimento que está iterando;

Figura 26 – Visão da interface no painel de resultados. No painel à esquerda é possível aplicar uma bateria de teste com configuração de parâmetros dinâmicos a cada teste. A região da direita ocorre o teste de diagnóstico da placa em teste de forma automatizada, assim o sistema coleta o resultado da recepção, avalia e classifica a qualidade da placa, informando *Passed the diagnostic* para indicar “placa boa” ou *Did not pass the diagnostic* para indicar “placa ruim”.

Fonte: Elaborada pelo autor.

- Painel de diagnóstico da placa (*Circuit board diagnosis*):
 - Painel de configurações de pulso quadrado (*Square pulse configuration*): aqui são realçados os valores que empregam o sinal padrão de comparação (“padrão ouro”), ou seja, representa o resultado conhecido para classificar a qualidade da placa. Neste painel são mostrados os seguintes parâmetros:
 - * *Analyzed points*: o número de pontos que serão analisados;
 - * *Variation*: a variação de amplitude aceitável dentro da região de análise;
 - * *N. points*: o número de pontos do pulso limitado pela capacidade máxima da memória **RAM RF**;
 - * *Amplitude*: amplitude máxima de 32767 correspondentes aos 14 bits do conversor **RF DACs**;
 - * NCO e VGA fixados em 2,50MHz e -8dB, respectivamente.

Observe-se que, para os propósitos desta dissertação, os parâmetros do painel de diagnóstico são fixados, não permitindo ao usuário sua alteração, porém em trabalhos futuros isto poderia ser habilitado.

- Painel de *overview*:
 - Exibe uma visão geral da análise de todos os testes aplicados, ou seja, um resumo estatístico;
 - Botão *Summary boxplot*: ativa a visão geral da análise.
- Painel de resultado (Result):
 - Botão *Apply diagnosis*: aplica o teste de acordo com a configuração do pulso e com o número de testes especificado.
 - Exibe o resultado do diagnóstico automático, classificando como placa “boa” informando *Passed the diagnosis* ou “ruim” se *Did not pass the diagnosis* de acordo com a variação de amplitude aceitável;
- Painel de visualização (*General plot*) contém a caixa de combinação com as seguintes opções :
 - *plot ADCs*: mostra todos os sinais de recepção sobrepostos, na mesma escala. Dessa maneira permite visualizar todo o conjunto de pontos próximo de uma região de variação aceitável;
 - *plot error bits*: ilustra para cada pulso recebido o ponto de variação máxima desse sinal. Para todos os sinais de recepção, esta análise é realizada, bem como elabora-se uma plotagem na mesma escala de gráfico;
 - *boxplot*: exibe uma análise estatística sobre cada sinal sendo mínimo, máximo, mediana, primeiro quartil, segundo quartil e terceiro quartil. Todos esses resultados são dispostos lado a lado para que seja feita uma comparação com os demais resultados caso ocorra algum pulso em destaque, com valores de *outlier*;
- Painel de visualização (*Individual plot*) contém a caixa de combinação com as seguintes opções :
 - *Analysis region.*: nesta opção é possível visualizar um sinal específico na região onde ocorre a análise, para isso, é necessário submeter o número do arquivo que se deseja visualizar.
 - *Complete signal.*: nesta opção é possível analisar toda região do sinal, para isso, é necessário submeter o número do arquivo que se deseja visualizar.

4.4 Restrições e alertas do sistema

A seguir é dada uma lista dos alertas que o sistema lança para o usuário caso aconteça alguma restrição.

- Quando o usuário digita os parâmetros de amplitude de um pulso, o sistema é capaz de indicar a possível saturação do conversor de transmissão **RF DACs**;
- O sistema não consegue prever a combinação de amplitude com o ganho do **VGA** quando o conversor está saturado. Logo, um aviso ocorre após o término da recepção;
- Uma possível situação que seria capaz de danificar os conversores **RF ADCs** seria aplicar um sinal de 32676 amplitude *trapézio* e um número de ponto 16383 e um ganho máximo de 34 dB. O que resulta nesses valores permanecendo no máximo por um longo período de tempo, o sistema alerta após receber o sinal de recepção;
- Quando um teste é automatizado, os parâmetros devem ser fixados e as margens de segurança são consideradas para não danificar os componentes, ou seja, o sistema alerta o possível dano físico do conversor;
- O valor de amplitude do pulso não deve superar de 32767, pois pode saturar o conversor **RF DACs** na transmissão, e também é capaz de saturar o conversor **RF ADCs** em situações em que o pulso respeite a faixa entre 10 a 32767 porém, o ganho do **VGA** possa estar saturando o conversor **ADCs**. Ambas as combinações de amplitude e ganho do **VGA** podem saturar o conversor **RF ADCs** na recepção. Nessa circunstância, o *software* envia um aviso no momento em que recebe o sinal de RF.

4.5 Verificação dos componentes RF DACs, VGA e RF ADCs por meio do *software*

Nesta seção é descrita a análise de cada um dos três componentes **RF DACs**, **RF ADCs** e **VGA** para assim validar o funcionamento do sistema. Primeiramente será detalhada a forma de validação de forma teórica e logo serão analisados por meio da análise do fluxo pelos conversores da placa para teste. Para isto será empregado o esquema de fluxo de dados apresentados na Figura 27, que representa os componentes **RF DACs**, **RF ADCs** e **VGA** como um conjunto de sistemas semi-isolados pelos quais os sinais de RF trafegam para sua transmissão (sinal modulante) e recepção (sinal reconstruído).

Figura 27 – Fluxo de dados que ocorre na transmissão e recepção do sinal de RF. O sinal é transmitido (sinal modulante) e recebido (sinal reconstruído) seguindo o fluxo: SW → FPGA → placa para teste (RF DACs → conexão via *loopback* → VGA → RF ADCs) → FPGA → SW. Os blocos pretos representam os componentes (1) RF DACs, (2) VGA e (3) RF ADCs. Os círculos em amarelo representam os barramentos.

Fonte: Elaborada pelo autor.

Verificação do RF DACs: a forma direta de constatar a correto funcionamento dos conversores RF DACs (Figura 27-1) é analisar o sinal na saída do conversor via osciloscópio. Pois o sinal digital que flui no barramento B é o sinal analógico que é mostrado no osciloscópio em ambos devem conter a mesma forma de um sinal com portadora modulante em amplitude. A segunda maneira, de forma indireta, pode ser feita quando o conjunto de CIs são ligados em série através da conexão *loopback*. Assim, se algum CI falhar os demais CIs serão comprometidos;

Verificação do VGA: após ter validado a correta operabilidade dos conversores RF DACs, seja da forma direta ou da indireta, significa que o sinal seguiu seu fluxo apropriado para o controle de ganho VGA (Figura 27-2);

Verificação do RF ADCs: em seguida, o sinal é enviado para o conversor RF ADCs (Figura 27-3). Se os sinais do barramento B e barramento C tiverem aproximadamente a mesma característica de um sinal da portadora modulante em amplitude então é validada a operabilidade de todos os CIs na placa para teste;

Verificação completa: após ter validado a verificação do RF ADCs o sinal segue para os módulos de **Recepção RF** onde ocorre o processamento deste sinal e depois é armazenado na **FIFO**. O software recebe o sinal demodulado da **FIFO** pelo barramento D e avalia a qualidade, onde o resultado fornece um diagnóstico com

relação à qualidade dos CIs ligado sequencialmente na placa para teste (RF DACs → conexão via *loopback* → VGA → RF ADCs).

4.5.1 Falhas de funcionalidade

Considera-se falhas de funcionalidade quando: (i) algum dos três componentes estejam inoperantes; e (ii) quando o modo *loopback* apresente-se desconectado para o canal (RF DAC-A/RF ADC-A) ou para o canal (RF DAC-B/RF ADC-B), pois resultaria em um sinal que não retornará no formato de portadora modulante em amplitude pelo barramento **C**, e portanto não retornará o sinal para ser analisado pelo *software*.

4.5.2 Falhas de qualidade

Para analisar a qualidade dos CIs é importante que estejam funcionando a transmissão e a recepção. Dessa forma, quando o *software* recebe um sinal com ruído acima do esperado é necessário analisar duas opções:

1. Analisar a saída do conversor RF DACs com auxílio do osciloscópio e averiguar a qualidade do sinal usando a sequência de transmissão (Figura 27-1):

FPGA (barramento **B**) → placa para teste (RF DACs)

2. Se o sinal está conforme o esperado, seguiremos para a segunda opção onde é analisado o circuito completo na seguinte sequência:

FPGA (barramento **B**) → placa para teste (RF DACs → conexão via *loopback* → VGA → RF ADCs) → FPGA (barramento **C**)

É mais plausível que a fonte de ruído esteja no conversor RF ADCs (Figura 27-3), pois o controle de ganho do VGA (Figura 27-2) dificilmente introduzirá ruído. Essa avaliação de qualidade pelo *software* é uma consequência dos dados da recepção. No cenário em que a primeira opção de análise apresente-se validada e o resultado ainda esteja insatisfatório, o mais provável é que a causa desse ruído seja o conversor RF ADCs.

4.5.3 Interferências geradas por ruídos

Existem dois barramentos de comunicação de dados/comandos onde podem ocorrer possíveis interferências de ruídos. A primeira interação de fluxo de dados ocorre entre o *software* e a FPGA, onde dificilmente ocorre interferência externa de ruído. A segunda interação é entre a FPGA e a placa em teste, onde a placa para teste fica mais exposta e o comprometimento da qualidade pode ocorrer da inserção de ruído branco em razão da eletrônica, bem como erros de conversão no RF DACs (digital → analógico) e RF ADCs (analógico → digital).

4.5.4 Saturação do conversor RF ADCs pelo VGA

A situação a seguir ilustra duas situações de ganho do VGA: para esta análise foi usado um pulso triangular com amplitude fixa de 32767 (não satura o conversor RF DACs na transmissão).

A primeira situação ocorre quando o sinal não satura o conversor RF ADC-B e a segunda situação ocorre quando o ganho do VGA está em 8 dB. A Figura 28-A apresenta o caso de recepção do sinal de RF dentro do esperado, pois o sinal de *triângulo* não saturou o conversor RF ADC-B. A segunda situação mostrada na Figura 28-B, ocorre na região de pontos de saturação do conversor, ou seja, é um caso em que o conversor foi saturado no ponto de pico do *triângulo*. Logo, o *software* alerta o usuário em duas situações:

- quando é utilizado uma amplitude do sinal maior que 32767, que é considerado um valor máximo em que o conversor RF DACs de 14 bits pode transmitir. O sistema emite com antecedência um alerta ao usuário sobre a saturação do conversor na transmissão do sinal de RF;
- quando ocorre uma configuração de ganho do VGA igual ou superior ao ponto de saturação do conversor RF ADCs. Na recepção é notificado quando já ocorreu a detecção desse ponto de saturação, por isso é necessário um ajuste equilibrado entre a amplitude do sinal a ser transmitido pelos conversores RF DACs e o ajuste de ganho do VGA, pois ambos influenciam diretamente no conversor RF ADCs.

Figura 28 – Exemplos de saturação e não saturação de um sinal. Na esquerda, é um sinal triangular que não está saturado. Na direita, o *software* alerta ao usuário que o sinal está saturado e posiciona a mensagem na região em que ocorre a saturação do conversor RF ADCs.

Fonte: Elaborada pelo autor.

4.6 Considerações finais

No presente capítulo foi proposto uma interface que permite ao usuário avaliar a qualidade das placas a serem testadas. Assim, foi descrita uma abordagem de 3 camadas que considera as interações de I/O através da interface, o gerenciamento das chamadas de funções através da regra de negócios, a geração e recepção do sinais de RF, as configurações de interface e geração de relatório via acesso a dados. O *software* proposto é capaz de orquestrar todo o sincronismo com a FPGA e a placa para teste. Além disso, também foram considerados alertas do sistema para advertir ao usuário final sobre um possível dano físico da placa em teste. Os parâmetros que foram incluídos no sistema estão de acordo com o solicitado por um especialista na área. Por fim, foi verificado o correto funcionamento dos componentes da placa para teste e a correta atuação do *software*

RESULTADOS E DISCUSSÕES

Este capítulo de resultados está dividido em duas partes. A primeira parte mostra a validação do sistema (*hardware* e *software*), onde verifica-se que o mesmo satisfaz os requisitos. A segunda parte apresenta os resultados da análise realizada com o objetivo de detectar as falhas nas placas para teste, a fim de verificar a robustez do sistema desenvolvido.

5.1 Configuração dos experimentos

5.1.1 Notação dos testes

Defini-se que um sinal de teste S_i é composto por um sinal configurado pelos parâmetros: a amplitude de transmissão A_{TX} , o número de pontos p , frequência de **NCO** f , e o ganho do **VGA** representado por g . Quando o sinal é recepcionado a amplitude pode variar sendo representada por A_{RX} . Além disso, um conjunto de sinais de testes $T = \{S_1, S_2, \dots, S_N\}$, onde N é o número total de testes. Para os experimentos, dado um conjunto de testes, é possível calcular a variação de amplitude de recepção ΔA que representa a diferença entre a máxima (A_{max}) e mínima (A_{min}) amplitude. Da mesma forma, a variação de ganho Δg é a diferença entre o ganho máximo (g_{max}) e o ganho mínimo (g_{min}).

Por outro lado, a metodologia de diagnóstico de placas é dado, principalmente, considerando que cada ponto interfere na qualidade de um sinal e cada sinal é um resultado parcial de um teste. Um experimento é um conjunto de N testes e é por meio desse conjunto que é possível obter o resultado do diagnóstico da placa.

5.1.2 Placas para teste a serem analisadas

A Figura 29 mostra o conjunto de placas a serem examinadas, denominadas Placa-1, Placa-2, Placa-3 e Placa-4 que correspondem à versão 3.0 e operam com conversor de RF ADCs de 16 bits; e Placa-5 que corresponde à versão 2.0 que opera com o conversor de RF ADCs de 14 bits.

Figura 29 – Conjunto de placas para testes avaliadas. Quatro amostras da versão 3.0 e uma da versão 2.0.

Fonte: Elaborada pelo autor.

O sistema proposto é adaptável a estas duas versões de placas, sendo apenas necessário alterar os endereços das pinagens referentes a cada versão de placa. Esse endereçamento estabelece as ligações entre os sinais de controle, barramento de dados e as vias destinadas ao *clock* entre a FPGA e a placa para teste.

5.2 Validação via osciloscópio da transmissão do sinal de RF nos conversores RF DACs

Foi realizado um experimento para validar as etapas da transmissão do sinal de RF pelos conversores RF DAC-A e RF DAC-B. Nesta análise foi usada a placa versão 3.0.

Primeiro foi configurado o envio via *software*. Assim, foram gerados dois sinais *Gaussianos* S_1 e S_2 com amplitude e número de pontos válidos. Logo, foi configurada a modulação desses sinais com as funções portadoras *coseno* e *seno*, para serem transmitidos pelo conversor RF DAC-A e RF DAC-B, respectivamente. Ambas as funções portadoras foram geradas pelo módulo NCO com frequência $f = 2\text{MHz}$.

Após ambos os sinais modulantes serem recepcionados na FPGA, ocorre o processamento desse sinais na transmissão seguindo essa sequência na FPGA:

FPGA (barramento A) → RAM RF → (Modulação* NCO) → Offset DAC → barramento B).

Logo, segue a conversão e transmissão via placa para teste. A transmissão é finalizada após passar pelos conversores RF DAC-A e RF DAC-B.

FPGA (barramento **B**) → Placa para teste (**RF DACs**) → Osciloscópio.

No osciloscópio (Figura 30) podem ser visualizados ambos sinais com as seguintes características: o canal RF DAC-A e o canal RF DAC-B está com um sinal *Gaussiana* como portadora modulante em amplitude, a qual uma é modulada com a portadora *cosseno* e outra com a portadora *seno*, respectivamente. Ambos os sinais S_1 e S_2 estão com uma frequência de $f = 2\text{MHz}$ e com desvio de fase de 90° . Assim, o resultado exibido no osciloscópio valida a correta transmissão do sinal de RF pois garante que as configurações feitas no *software* foram executadas corretamente na FPGA.

Figura 30 – Validação do sistema no osciloscópio das saídas RF DACs. A) Dois sinais *Gaussianos* transmitidos pelos conversores RF DAC-A (amarelo) e RF DAC-B (verde) modulados pelo NCO com função *cosseno* e *seno*, respectivamente, ambos com frequência de modulação de 2MHz e desvio de fase próximo dos 90° . B) Ampliação da região marcada em (A).

5.2.1 Análise da flexibilidade do sistema

Para verificar a flexibilidade do sistema foi feito um experimento usando os dois canais (DAC-A/DAC-B) de forma independente. Na transmissão foi habilitado

o modo *loop*, via interface de usuário, assim ocorrem múltiplas repetições de sinais $T = \{S_1, S_2, \dots, S_N\}$ (Figura 31-A), os quais são configurados com diferentes parâmetros. Para o canal **DAC-A** envia-se um conjunto de sinais T_A com função *seno*, com amplitude de $A_{TX} = 32767$ e número de pontos $p = 1024$; para o canal **DAC-B** envia-se outro conjunto de sinais T_B com função *sinc*, com amplitude $A_{TX} = 32760$, porém com número maior de pontos $p = 16084$.

Na Figura 31-A é mostrada a saída dos sinais no osciloscópio. Na região ampliada (Figura 31-B) pode ser feita uma comparação entre ambos canais. Observa-se que, no canal **RF DAC-A**, aproximadamente, treze sinais do conjunto T_A com função *seno*, correspondem ao tamanho de um sinal *sinc* no canal **RF DAC-B**.

De acordo com este resultado, constata-se que cada canal opera de forma independente e, além disso, garante que o sistema proposto é flexível pois disponibiliza vários parâmetros, números de pontos, amplitude do sinal e a função gerada pelo **NCO**. Todos estes parâmetros podem ser combinados e podem variar de acordo com as restrições de cada funcionalidade.

Figura 31 – Análise da flexibilidade do sistema via osciloscópio. A) Canal **RF DAC-A** transmite um conjunto de sinais de teste T_A com função *seno* enquanto o canal **RF DAC-B** transmite um conjunto T_B contendo três sinais de *sinc*. B) Ampliação do sinal *sinc* S_2 do conjunto T_B contendo aproximadamente treze sinais de *seno* do conjunto T_A .

Fonte: Elaborada pelo autor.

5.3 Validação da transmissão e recepção via software

Este experimento faz uso da placa na versão 3.0 e visa validar a transmissão e a recepção. Foi gerado um conjunto de teste, $T = \{S_1, S_2, S_3\}$, contendo três sinais com função *sinc* configurado com amplitude $A_{TX} = 367$, $p = 1024$ pontos e **NCO** na frequência de $f = 2,5$ MHz. Estes parâmetros não são alterados no decorrer do experimento, as mudanças ocorrem somente no valor de ganho g em dB que amplificará a magnitude do sinal na saída do **VGA**.

O comportamento de cada um destes testes são apresentados a seguir. Para o teste S_1 (Figura 32-A) foi aplicado um ganho no **VGA** de $g = 1$ dB, o que resultou na amplitude do sinal de recepção estar próximo a $A_{RX} = 30$, esse sinal está abaixo da amplitude do sinal de transmissão $A_{TX} = 367$. Para o teste S_2 (Figura 32-B), o ganho do **VGA** foi para $g = 22$ dB, com isso podemos observar as duas *sinc* próximas da mesma amplitude $A_{TX} = A_{RX} \approx 367$. Para o teste S_3 (Figura 32-C), o ganho do **VGA** foi de $g = 34$ dB e a amplitude do sinal da recepção ficou próximo de $A_{RX} = 1400$, que é bem superior ao sinal transmitido. Observa-se que esse sinal está bem por baixo do ponto de saturação do conversor **RF ADCs**. Neste caso, é o especialista (usuário) quem passa a controlar esses experimentos de teste, ajustando os parâmetros manualmente.

Figura 32 – Validação via *software* da transmissão e recepção do sinal de RF variando o ganho do **VGA**. A) O sinal de transmissão está com uma amplitude superior ao sinal de recepção e com ganho no **VGA** de 1dB. B) O sinal de transmissão e recepção têm aproximadamente a mesma amplitude com ganho no **VGA** de 22dB. C) O sinal de recepção está com uma amplitude maior que o sinal de transmissão, com uma ganho de **VGA** de 34dB.

Fonte: Elaborada pelo autor.

Este conjunto de testes T complementa a validação de todas as etapas da transmissão e o fluxo de dados pela placa em teste. Este experimento valida o bom funcionamento dos módulos da **Recepção RF** a qual segue esse fluxo até a reconstrução do sinal:

Placa para teste (**VGA** → **RF ADCs**) → **FPGA** (barramento **C**) → **data ADCs** → **offset ADCs** → (**Misturador** * **NCO**) → **CIC** → **CFIR** → **FIFO** → barramento **D**) → **SW**.

5.3.1 Relatório automático das variações do ganho do VGA

Este experimento foi realizado com o uso da placa na versão 2.0. Foi necessário fixar um conjunto de testes $T = \{S_1, \dots, S_{256}\}$ contendo $N = 256$ sinais com função *sinc* de amplitude $A_{TX} = 700$ (Figura 33-B). O ganho foi configurado de forma incremental para cada sinal ($g = \{1, 2, \dots, 256\}$), respectivamente, visando não saturar o conversor quando atingir o valor máximo de ganho do VGA g_{max} , dependendo do conversor da placa versão 2.0 ou versão 3.0 (Seção 4.5.4). O cenário deste teste, onde o *software* controla a transmissão e recepção, segue a sequência:

Software-TX → FPGA (barramento A) → RF → barramento B → placa para teste (RF DACs → conexão via *loopback* → VGA → RF ADCs) → FPGA (barramento C) → RF → barramento D) → *software-RX*.

Após preparar o ambiente de teste seguindo a sequência resumida, o *software* controla a inserção de valores de ganho para o VGA de forma automática para cada transmissão. O resultado dos dois canais (RF ADC-A e RF ADC-B) foram sobrepostos no gráfico da Figura 33-A, onde foi coletado todos os máximos pontos de amplitude A_{max} de cada sinal da recepção.

Figura 33 – Relatório automático das variações do ganho do VGA para $N = 256$ sinais de teste. A) Representa o ponto máximo de cada sinal e também testa todas as 256 escalas de ganho do VGA. B) Sinal fixo com função *sinc* para todo o conjunto de testes na transmissão. C) Mostra dois valores máximos que correspondem ao sinal S_{127} , localizado na região de passos menores de ganho; e um outro do sinal S_{181} , localizado na região com passos maiores, ambos representados em (A).

Fonte: Elaborada pelo autor.

A Figura 33-C mostra os dois sinais coletados em diferentes regiões de ganho,

sendo que o sinal S_{127} representa o seu ponto máximo que está representado na faixa de ganho baixa, onde é permitido um ajuste mais fino entre os ganhos. O outro é o sinal S_{181} que está denotado na faixa de ganho alta, onde ocorre maior variação de ganho.

Note que a programação de controle de ganho para cada **VGA_A** e **VGA_B** são idênticas, pois cada canal tem seu próprio controle de ganho. Neste experimento manteve-se padronizado o sinal de transmissão não ocorrendo variação do sinal no conversor **RF DACs**, porém o conversor **RF ADCs** obteve mais variações de sinal por motivo da variação de ganho aplicado no CI do **VGA**.

5.4 Estratégia para a detecção de falhas na reconstrução do sinal de RF

5.4.1 Dataset da placa para teste por introdução de ruído

De acordo com os experimentos de validação, que foram tratados nas seções anteriores, foi observado que o conjunto de placas para testes (Seção 5.1.2) não possui, falhas de funcionalidade, foi necessária a geração de um dataset com sinais ruidosos. Por outro lado, também foi criado um cenário que proporcionou caracterizar uma placa com defeito, como é apresentado na Figura 34 e na Figura 35-B. Este defeito pode ser gerado possivelmente ao efetuar uma conexão imperfeita entre a placa em teste e a FPGA pelo *Flex Cable* (descrito no Capítulo 2). Hipoteticamente, esse defeito também pode ter se originado pela presença de poeira nos conversores, ou nos componentes. Não obstante, esse problema pode ser resolvido após retirar a placa para teste, ou ao realizar a limpeza dos conectores e partes dos circuitos da placa para teste, usando um pincel, e reconectando-o na FPGA.

Ambos os cenários possuem a mesma situação de ruído, o que altera-se é o tipo de sinal que é aplicado no teste. Mediante esse cenário, foi desenvolvida a funcionalidade de automatizar os testes e oferecer um diagnóstico mais preciso em relação à qualidade da placa. Pois o sistema realiza testes exaustivos para garantir a consistência, estabilidade dos dados e por fim avaliar automaticamente os resultados dos dados recebidos e classifica-os como placa “boa” ou placa “ruim” ao final do teste.

A Figura 34 representa um *dataset* dos sinais de recepção da placa em teste, que inclui os CIs (**RF DACs** → conexão via *loopback* → **VGA** → **RF ADCs**), a qual mostra quatro recepções de um sinal *sinc*. As Figuras 34-A-C apresentam uma distribuição de ruído aleatório em todo o sinal, sendo assim, caracterizando uma placa como “ruim”. A Figura 34-D é um sinal de uma *sinc* sem ruído, caracterizando-a como uma placa “boa”.

Figura 34 – Exemplos de erros na reconstrução do sinal de RF. A), B) e C) apresentam um sinal reconstruído com muita inserção de ruído. Esse erro classificaria a placa como “ruim”. D) Exemplo de sinal de reconstrução que classifica a placa como “boa”.

Fonte: Elaborada pelo autor.

5.4.2 Considerações para a classificação de placa “boa” ou “ruim”

Quando uma placa é considerada “ruim”, o ruído se faz presente em todos os sinais reconstruídos independentemente da forma do sinal gerado. A Figura 35 representa uma análise de $N = 20$ sinais de teste com função de pulso *quadrado*, cada um contendo $p = 16383$ pontos, aplicados ao canal B. Na Figura 35-A é apresentada uma visão completa deste conjunto de sinais. Note que os sinais de recepção dos conversores RF ADC-B estão sobrepostos na mesma escala dos eixos do gráfico.

A Figura 35-B ilustra uma ampliação que corresponde à área de análise referente aos $N = 20$ sinais de testes aplicados. Constata-se que este conjunto de testes obteve muitos sinais que dispuseram grandes variações na sua amplitude ΔA . Para que o sistema possa classificar se uma placa encontra-se no grupo de placas “boas” ou no grupo de placas “ruins” foi necessário estabelecer um limite de cortes, ou seja, uma variação que seja considerada aceitável. Essa variação depende do conversor que esteja sendo usado,

de acordo aos seus respectivos *data-sheets* **RF DACs** (Texas Instruments, 2017) e **RF ADCs** (Analog Devices, Inc, 2006). Os valores aceitáveis usados neste trabalho são dados na Tabela 4.

Tabela 4 – Variação de amplitude aceitável para o sinal ser convertido de digital → analógico → digital nos conversores **RF ADCs** e **RF DACs**

Conversor	Versão	Máximo de variação aceitável
RF DACs + RF ADCs	2.0	$\Delta A \leq 6$
RF DACs + RF ADCs	3.0	$\Delta A \leq 4$

Fonte: Elaborada pelo autor.

Com base na região da Figura 35-B serão efetuadas várias análises para conceder o diagnóstico para cada nova versão de placa.

Figura 35 – A) Plot de $N = 20$ testes com sinais *quadrados* sobrepostos no mesmo gráfico. B) Região mais estável do gráfico em A), onde torna-se mais simples identificar as variações de amplitude ΔA .

Fonte: Elaborada pelo autor.

5.4.3 Análise do sinal reconstruído no conversor RF ADCs

A Figura 36 apresenta a análise do sinal reconstruído no conversor **RF ADC-B** usando um conjunto de $N = 20$ sinais de teste. A Figura 36-A representa uma região de análise que contém a sobreposição dos sinais sobre a mesma escala do gráfico. Nota-se que no eixo da amplitude, os sinais exibem grande variação de amplitude, resultantes da soma dos erros de conversão do **RF DACs** com os **RF ADCs**. A Figura 36-B realça os detalhes do sinal S_5 , em que ocorreu uma variação de $\Delta A = 4$, é considerado que esse sinal encontra-se no limite de tolerância de erro aceitável. Por outro lado, a Figura 36-C aponta que o sinal S_4 com variação de amplitude $\Delta A = 127$ ultrapassou a tolerância

admitida (ver Tabela 4), portanto a qualidade desse sinal caracteriza o canal **RF ADC-B** como “ruim”.

Finalmente, a Figura 36-D expõe um resumo de todos os sinais. Por meio deste gráfico é possível constatar a variação da amplitude ΔA de cada sinal, incluindo os limites mínimo e máximo, mediana e *outliers*.

Figura 36 – Bateria de testes nos conversores **RF DACs**, **VGA** e **RF ADCs**. A) Conjunto de sinais recebidos dos $N = 20$ testes aplicados. B) Ampliação do sinal de teste S_5 que contém o número de pontos distribuídos entre 3014 a 3018, compreendendo uma variação de amplitude de $\Delta A = 4$. C) Sinal de teste S_4 com variação $\Delta A = 127$ ao longo de todo o sinal. D) Resumo da qualidade dos $N = 20$ sinais de testes aplicados, identificando ao sinal em (B) e (C) com variação $\Delta A = 4$ e $\Delta A = 127$, respectivamente.

Fonte: Elaborada pelo autor.

5.5 Diagnóstico da qualidade de cada canal das placas em teste

5.5.1 Configuração dos sinais

As características de um sinal também podem oferecer um diagnóstico da placa mais acurado. Assim, nos seguintes experimentos foi usado um pulso *quadrado*, isto

porque trata-se de uma função simples de analisar, e caracteriza-se pela sua estabilidade e assim é considerado como um padrão ouro. A seguir são descritas as configurações do conjunto de sinais de teste:

- sinal *quadrado* com amplitude máxima de $A_{\max} = 32767$;
- número de pontos $p = 16383$ onde são descartados os 1400 pontos iniciais e 1400 pontos finais desse pulso, restando uma região para análise de $p' = 13583$ pontos;
- controle de ganho do **VGA** sendo programado a $g = -8\text{dB}$;
- frequência do **NCO** a $f = 2,5\text{MHz}$;
- conexão via *loopback* estabelecida para o canal (**RF DAC-A** $\rightarrow$ conexão via *loopback* $\rightarrow$ **VGA** $\rightarrow$ **RF ADC-A**) ou canal (**RF DAC-B** $\rightarrow$ conexão via *loopback* $\rightarrow$ **VGA** $\rightarrow$ **RF ADC-B**) .

Dados estes parâmetros iniciais agora é possível aplicar uma análise mais criteriosa da região estabelecida, pois a amplitude permanece constante no decorrer da análise onde é possível detectar essa variação com maior precisão.

A região de análise está representada na Figura 37-B, onde são exibidos todos os sinais de recepção com amplitude entre $A_{RX} = [1456, 1463]$. Para proceder com a simulação de uma placa com defeito foi considerado o sinal S_5 e alterou-se seu valor de amplitude na região $A_{TX} = [1457, 1463]$, resultando assim em uma variação de amplitude $\Delta A = 6$, o que consequentemente, sobrepassa o valor aceitável de tolerância (Tabela 4) para esta classe de conversor. Portanto, o sistema diagnosticará a placa versão 3.0 como sendo “ruim”.

O sistema também é capaz de detectar as variações entre os valores da amplitude, máximo e mínimo, de cada sinal. No exemplo apresentado na Figura 37-B encontra-se um conjunto de $N = 40$ sinais de testes aplicados ao canal **RF ADC-A**. Esse sinal alterado se destaca no meio do conjunto de sinais e se torna mais visível na Figura 37-C na região de pontos $[3200, 3250]$ em destaque. O *software* analisa todos esses conjuntos de sinais e consegue detectar automaticamente essa variação de amplitude e permite dar um diagnóstico preciso.

5.5.2 Dataset sintético

A Figura 38 refere-se a um conjunto de $N = 40$ sinais de teste aplicados aos canais A da Placa-1. Observa-se que o sinal S_5 atingiu uma variação mínima no seu valor de amplitude A_{RX} , elevando assim a amplitude desse sinal no ponto 1457 para 1463, o que

Figura 37 – Detecção de sinal ruidoso no conjunto de $N = 40$ sinais de teste aplicados ao canal A.

Fonte: Elaborada pelo autor.

Figura 38 – Teste do sinal S_5 é identificado com uma variação de amplitude $\Delta A = 6$.

Fonte: Elaborada pelo autor.

resulta em uma variação de amplitude $\Delta A = 6$. O sistema detectou essas variações e comparou com o valor de variação aceitável desses conversores (Tabela 4).

Foi feita uma outra análise, desta vez dedicada ao sinal S_5 em destaque na Figura 38-A e com mais detalhe na Figura 38-B. O sistema consegue detectar o ponto que está com *outlier*, ponto que escapa aos restantes.

Na Figura 38-B é mostrado o gráfico de resumo produzido pelo sistema, o qual calcula os limites inferiores e superiores (*boxplot*) dos pontos contidos no sinal S_5 visando medir a variação de amplitude. Nessa mesma figura é detectado o ponto *outlier*. Assim é observado que a variação de amplitude $\Delta A = 6$ não é aceitável e portanto o sistema acusará a placa classificada como “ruim”.

O resultado mencionado acima é utilizado para medir a qualidade da placa para um conjunto de $N = 40$ sinais de teste. De fato, a análise pode ser aplicada para $N = 500$ testes ou mais, todos os sinais estão regidos pelo mesmo critério de avaliação e o resultado final classifica a placa em teste como “boa” ou “ruim”.

5.5.3 Identificação de outlier

A Figura 39 ilustra um resumo da análise dos $N = 40$ sinais de teste aplicados ao canal A da Placa-1, onde é possível contar com uma visão prévia referente à qualidade de cada sinal de recepção (ver a Figura 39-A) e por outro lado a Figura 39-B mostra o resumo de todos os testes aplicados a este canal.

O *software* é capaz de proporcionar um diagnóstico imediato após o teste realizado neste canal, bem como permite uma visualização detalhada sobre esse teste, permitindo assim, identificar quais sinais adquiriram um comportamento de destaque em meio aos $N = 40$ testes da Figura 39-A. Na Figura 39-A também é possível observar as particularidades desse sinal e um resumo dos $N = 40$ testes na Figura 39-B.

5.5.4 Dataset original

Foi feito um outro experimento usando o mesmo sinal S_5 dos experimentos anteriores, porém desta vez foi restaurado no seu formato original. Pode-se notar na Figura 40-A que o *outlier* que estava presente na Figura 39-B foi eliminado.

O sinal S_5 volta a apresentar uma variação de amplitude $\Delta A = 1$, sendo que a variação média para o conjunto total foi de $\langle \Delta A \rangle = 3$ que está representado na Figura 39-A. O *boxplot* da Figura 40-A mostra que o máximo de variação foi $\Delta A_{\max} = 1$ o que é considerado aceitável e o sistema classifica essa placa como “boa”.

A Figura 41-A salienta todos os $N = 40$ sinais do conjunto de testes T. Pode-se perceber que os sinais S_2 , S_{35} e S_{39} estão com uma variação de $\Delta A = 3$ enquanto que

Figura 39 – Conjuntos de testes aplicado à Placa-1 no canal A destacando o *outlier* referente ao sinal S_5 modificado.

Figura 40 – Teste do sinal S_5 referente à Placa-1 do canal A.

Fonte: Elaborada pelo autor.

os demais obtiveram uma variação de $\Delta A = 2$. O resumo da Figura 41-B corresponde a este cenário, onde representa o máximo de variação em 3 e o terceiro quartil com uma variação de 2.

5.5.5 Resultado de análise da qualidade de cada canal das placas que foram testadas

O resultado da Figura 42 permite comparar a qualidade das placas entre os mesmos canais de cada placa assim como efetuar comparações entre placas de versão 2.0 e 3.0.

O primeiro procedimento de teste mede a qualidade do canal A, o que inclui a

Figura 41 – Conjunto de testes aplicado à Placa-1 do canal A.

Fonte: Elaborada pelo autor.

análise dos CIs (conversor **RF DAC-A**, controle da **VGA** desse canal e o conversor **RF ADC-A**). O segundo procedimento de teste é realizado para o conjunto de CIs do canal B e, para concluir este conjunto de procedimentos, é aplicado a cada placa em teste.

Pode-se notar que o resultado dos $N = 4$ sinais de teste referentes às placas de versão 3.0 (Placa-1, Placa-2, Placa-3 e Placa-4) se enquadram no perfil de placas “boas”, pois todas estas obtiveram uma variação menor ou igual a $\Delta A = 4$. Destaca-se que a Placa-3, foi a que alcançou o melhor resultado de qualidade entre as demais da mesma versão de placa.

Para o canal B, destaca-se em primeiro lugar a Placa-3, pois a máximo de variação é $\Delta A_{\max} = 3$ e o terceiro quartil está com valor inferior a 2. Em segundo lugar, o canal A também da Placa-3 está com o máximo $\Delta A_{\max} = 3$ e o terceiro quartil está em 2 próximo da sua mediana. A Placa-5 da versão 2.0 conseguiu um resultado considerado aceitável pois a variação é $\Delta A \leq 6$.

Figura 42 – Comparação da qualidade entre as placas que foram testada. Relação da qualidade entre os canais da mesma placa e uma comparação com os demais canais.

Fonte: Elaborada pelo autor.

5.6 Considerações finais

Este capítulo descreveu duas partes de resultados, a primeira parte verifica o bom funcionamento do sistema. A segunda parte foram detalhados vários experimentos para analisar o funcionamento da detecção de falhas das placas. Para isso, foi descrito um conjunto de dados de sinais com diferentes formas e foram utilizadas cinco placas que foram testadas (*dataset* real), e além disso, também foi realizada uma análise com sinais modificados (*dataset* sintético). De acordo com os resultados foi possível comparar a qualidade entre os canais de cada versão de placa e atribuir corretamente as placas “boas” e “ruins”.

CONCLUSÕES

O objetivo desta dissertação foi criar um ambiente para testar as diversas placas desenvolvidas pela CIERMag, bem como auxiliar o especialista na identificação das falhas de qualidade ou falhas de funcionalidade nessas placas. Para realizar a detecção dessas falhas foram criadas várias rotinas de teste personalizado, semiautomático e automatizado. O sistema avalia todas as variações nos pontos de amplitudes que formam um sinal através dos vários testes aplicados, o que resulta em um sistema que consegue avaliar e elaborar um diagnóstico preciso de forma automatizada. Permitindo ao usuário a opção de aplicar testes de diagnóstico e avaliar em poucos minutos a qualidade da placa dispondo como referência a qualidade dos dados da recepção.

O desenvolvimento deste sistema teve dois grandes pontos desafiadores: (i) o desenvolvimento sobre uma arquitetura híbrida de interconexão entre a CPU e a FPGA e (ii) a qualidade das placas pode ser influenciada por vários critérios que foram descobertos durante os estágios de desenvolvimento.

O primeiro desafio da arquitetura híbrida foi na sincronização de dados e comandos para estabelecer a comunicação entre a FPGA e o software. Uma vez superado este desafio, em seguida foi feita a sincronização da transmissão e a aquisição de sinais digitais entre a FPGA e as placas desenvolvidas pela CIERMag. Finalmente, foi sincronizado o sistema na totalidade, a qual inclui o software, FPGA e placa em teste para que trabalhem em conjunto de modo a orquestrar as operações de teste, diagnóstico e análise.

O segundo desafio foi durante a análise dos sinais de recepção. Durante o desenvolvimento do trabalho foi observado que a qualidade das placas pode ser influenciada por vários fatores: pelo layout da própria placa, pelo tipo do conversor (14 ou 16 bits), pelos componentes e conversores dos diferentes fabricantes, os quais alteram as especificações de qualidade desses componentes, e portanto todos esses fatores interferem

diretamente na qualidade final dos resultados. Por outro lado, a região de maior estabilidade encontra-se no conjunto da lógica do processador com a lógica implementada na FPGA onde dificilmente é afetada por ruído externo, porém ao conectar qualquer uma das versões de placas no HSMC (com diferente layout e componentes) pode implicar em resultados diferentes. Em Ciência de Dados, todos esses fatores que influenciam na qualidade do resultado devem ser considerados, pois, as somatórias de erros acumulados quando o sinal passa por todos esses componentes afetam diretamente na qualidade do resultado final.

Por último, pode-se concluir que o sistema cumpre com os requisitos/objetivos propostos, pois os resultados obtidos foram satisfatório para diagnosticar uma placa com defeito e além disso o sistema demonstrou flexibilidade em se adaptar às várias versões de placas. Ademais, este trabalho encaixa-se diretamente com a área de ciência dos dados, pois foram abordados métodos de análise para a detecção de falhas na placa e foi feita uma análise nos módulos que realizam transformações dos dados (Capítulo 3). Também foi efetuada uma análise via software, via simulador e via osciloscópio. Esta análise foi significativa para aplicar as melhorias na implementação de *software/hardware* com o intuito de atingir o resultado esperado.

6.1 Contribuições

As contribuições deste trabalho foram:

- Foi proposta uma solução automatizada que permite ao usuário escolher o número de testes que deseja aplicar em cada canal. O sistema pode operar em duas opções: de forma personalizada (em modo manual) ou de forma automatizada (usando uma configuração por padrão), porém permite o incremento/decremento de parâmetros pelo sistema;
- O sistema abrange várias configurações de teste, o que evita utilizar equipamentos caros, reduzindo o risco de queimar, a verificação é rápida podendo ser personalizada para testar cada conjunto do circuito, seja no conjunto de circuito de gradientes ou RF.
- Esta dissertação também trouxe contribuição com respeito à multidisciplinaridade, ao envolver *hardware*, MRI e os sinais de radiofrequência estudados sob a perspectiva de Ciência de dados. Cada um desses problemas é uma contribuição por si só.
- Foi proposta uma metodologia para a análise de sinais de radiofrequência gerados de forma automatizada.

- Além da multidisciplinaridade, os problemas tratados estão enquadrados na área de Ciência de Dados, uma vez que apresentam informações de diferentes naturezas e tamanhos tratados com a mesma abordagem.
- Este trabalho significa uma das primeiras tentativas de automatização da produção para a CIERMag de modo a detectar falhas de operabilidade, ou falhas de qualidade nas placas, de forma rápida.

6.2 Questões para trabalhos futuros

Várias sugestões podem ser seguidas como próximos desafios:

- Aumentar o número de bits de dados de 16 para 32 bits;
- Ajustar a janela de recepção para que capture o dado no momento exato, sem que haja variações desse tempo de captura dos dados;
- Adicionar módulos de proteção que monitorem e tomem decisões em tempo real quando surgir alguma saturação nos conversores **RF ADCs**, interrompendo instantaneamente a transmissão do sinal de RF e tudo isso aliado ao gerenciamento e controles do *software*.

REFERÊNCIAS

Altera Corporation. **SoC FPGA ARM Cortex-A9 MPCore Processor Advance Information Brief**. [S.l.], 2012. Disponível em: <https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/aib-01020-soc-fpga-cortex-a9-processor.pdf>. Acesso em: 05.12.2019. Citado na página 33.

_____. 2015. Disponível em: <<http://www.altera.com>>. Acesso em: 05.12.2019. Citado nas páginas 34 e 39.

ALTERA®. **Cyclone V SoC Development Board Reference Manual**. 2015. Disponível em: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/manual/rm_cv_soc_dev_board.pdf>. Acesso em: 02/02/2020. Citado na página 52.

Analog Devices, Inc. **Analog-to-Digital Converter AD9254: 14-Bit, 150 MSPS, 1.8 V**. Norwood, USA: [s.n.], 2006. Disponível em: <<https://www.analog.com/media/en/technical-documentation/data-sheets/AD9254.pdf>>. Acesso em: 11.13.2019. Citado nas páginas 91 e 110.

BOBDA, C. **Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications**. 1st. ed. Heidelberg, Germany: Springer Netherlands, 2007. Citado na página 32.

CARVALHO, P. R. B. **Projeto de circuito Oscilador Controlado Numericamente implementado em CMOS com otimização de área**. Dissertação (Mestrado) — Departamento de Engenharia de Energia e Automação Elétrica. Escola Politécnica da Universidade de São Paulo, SP, Brazil, 2016. Citado na página 39.

CIERMAG. **Centro de Imagens e Espectroscopia in vivo por Ressonância Magnética (CIERMAG)**. 2019. Disponível em: <<https://www2.ifsc.usp.br/portal-ifsc/grupo-de-ressonancia-magnetica-espectroscopia-e-magnetismo/>>. Acesso em: 11.13.2019. Citado na página 27.

CROCHIERE, R.; RABINER, L. Interpolation and decimation of digital signals—a tutorial review. **Proceedings of the IEEE**, Institute of Electrical and Electronics Engineers (IEEE), v. 69, n. 3, p. 300–331, 1981. Citado na página 46.

DEVICES, I. A. **Digitally Controlled VGA Data Sheet AD8370: LF to 750 MHz**. Norwood, USA: [s.n.], 2005. Disponível em: <<https://www.analog.com/media/en/technical-documentation/data-sheets/AD8370.pdf>>. Acesso em: 11.13.2019. Citado nas páginas 107, 108 e 109.

EATON, J. W. **GNU Octave: Scientific Programming Language**. 2020. Disponível em: <<https://www.gnu.org/software/octave/>>. Acesso em: 02.03.2020. Citado nas páginas 64 e 65.

- Globo. **Intel compra Altera por US\$ 16,7 bilhões**. 2015. Disponível em: <<https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html>>. Acesso em: 05.12.2019. Citado na página 34.
- HAUCK, S.; DEHON, A. **Reconfigurable Computing: The theory and Practice of FPGA**. [S.l.]: Elsevier Book, 2008. Citado na página 32.
- Intel Quartus Prime. 2015. Disponível em: <<https://www.intel.com/content/www/us/en/software/programmable/quartus-prime/overview.html>>. Acesso em: 05.12.2019. Citado na página 34.
- KADAM, S.; SASIDARAN, D.; AWAWDEH, A.; JOHNSON, L.; SODERSTRAND, M. Comparison of various numerically controlled oscillators. In: **The 2002 45th Midwest Symposium on Circuits and Systems, MWSCAS-2002**. OK, USA: IEEE, 2002. Citado na página 39.
- LIM, I.-G.; KIM, W.-W. A numerically controlled oscillator with a fine phase tuner and a rounding processor. **ETRI Journal**, v. 26, n. 6, p. 657–660, 2004. Citado na página 39.
- LYONS, R. G. **Understanding digital signal processing**. 3rd ed. ed. Upper Saddle River, NJ: Prentice Hall, 2011. Citado na página 47.
- MA, J.; KARADAYI, K.; ALI, M.; KIM, Y. Ultrasound phase rotation beamforming on multi-core DSP. **Ultrasonics**, v. 54, n. 1, p. 99–105, 2014. Citado na página 46.
- MCLEAN, M. A. Fundamentals of MR spectroscopy. In: **Comprehensive Biomedical Physics**. Cambridge, UK: Elsevier, 2014. p. 257–271. Citado na página 27.
- MEDDINS, B. The design of IIR filters. In: **Introduction to Digital Signal Processing**. [S.l.]: Elsevier, 2000. p. 71–101. Citado na página 46.
- ModelSim-Intel® FPGA. 2019. Disponível em: <<https://www.intel.com.br/content/www/br/pt/software/programmable/quartus-prime/model-sim.html>>. Acesso em: 05.12.2019. Citado na página 34.
- MOORE, A. **FPGAs For Dummies, Altera Special Edition**. John Wiley & Sons, 2014. Disponível em: <<http://design.altera.com/New2FPGAeBook>>. Citado na página 28.
- MÉLO, F.; DIAS, R.; STEINBACH, R. FPGA para todos: um projeto para a disseminação da tecnologia de lógica programável. In: **XXXIX Congresso Brasileiro de Educação em Engenharia**. SC, Brazil: COBENGE, 2011. Citado na página 32.
- RADATZ, J. **The IEEE Standard Dictionary of Electrical and Electronics Terms (6th ed)**. [S.l.]: Inst of Elect & Electronic, 1997. ISBN 1559378336. Citado nas páginas 36 e 39.
- SILVA, M. G. **Implementação de um controlador digital em FPGA por síntese em alto nível: Estudo de caso**. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, 2016. Citado na página 33.
- TERASIC. **SoCKit CD-ROM (rev. C Hardware)**. Florida, USA: [s.n.], 2019. Disponível em: <<http://download.terasic.com/downloads/cd-rom/sockit/>>. Acesso em: 11.13.2019. Citado na página 52.

Texas Instruments. **DAC8812 Dual, Serial-Input 16-Bit Multiplying Digital-to-Analog Converter**. Dallas, Texas, USA: [s.n.], 2016. Disponível em: <<http://www.ti.com/lit/ds/sbas349f/sbas349f.pdf>>. Acesso em: 07.02.2020. Citado nas páginas 115, 118, 119 e 120.

_____. **DAC5672 14-Bit 275 MSPS Digital-to-Analog Converter**. Dallas, Texas, USA: [s.n.], 2017. Disponível em: <<http://www.ti.com/lit/ds/symmlink/dac5672.pdf>>. Acesso em: 07.02.2020. Citado nas páginas 61, 91, 107 e 108.

APÊNDICE A:

A.1 Arquitetura dos circuitos integrados

A.1.1 Arquitetura dos conversores RF DACs

A ([Texas Instruments, 2017](#)) fabrica os modelos de DACs que estão integrados na placa da CIERMag. Assim, foi fundamental o bom entendimento dos CIs para poder desenvolver o *hardware*.

O modelo DAC5672 é um conversor DAC monolítico de dois canais, 14 bits e esta configurado para opera a uma frequência de 50MHz. Os sinais gerados na saída da transmissão da FPGA são pulsos modulados de RF que seguem para o conversor, onde ocorre uma nova transformação nos dados, sendo convertido de digital para analógico.

De acordo com o *data-sheet* da ([Texas Instruments, 2017](#)) (ver Figura 43), os dados do sinal de RF passam pelo barramento de *input DAC-A[13:0]/DAC-B[13:0]* e são sincronizados com o *clock CLOCK-A/CLOCK-B* em conjunto com o sinal WRTA/WRTB de 50MHz gerado pela FPGA, resultando na transferência dos dados entre os registradores temporários (LATCH A/LATCH B) até a saída dos DACs correspondentes, conforme ilustrado na Figura 43.

A.1.2 Arquitetura do controle de ganho VGA

O controle de ganho é um CI que está soldado na placa para teste nas versões 2.0 e 3.0. O VGA é de modelo AD8370ARE e segue as especificações do *data-sheet* ([DEVICES, 2005](#)). O VGA cumpre a função de atenuar ou amplificar o sinal analógico do pulso de RF na recepção antes do conversor de RF ADCs.

Este CI é controlado via *software* no instante de início da transmissão do pulso de RF, assim é administrada pelo módulo **Controle VGA** (Seção 3.3.2.1) para enviar

Figura 43 – Diagrama do conversor (esquerda) mostra o *input/output* de dados e sinais de controle, respectivamente; e a estruturação do conversor. A direita mostra o instante da conversão dos dados de digital para analógicos via os **DACs**. Os dados são transferidos e os sinais de controle *CLOCK-A/CLOCK-B*, *WRTA/WRTB* são sincronizados com o barramento de dados *DAC-A[13:0]/DAC-B[13:0]*.

Fonte: Adaptada de [Texas Instruments \(2017\)](#).

uma programação de pulso no formato de protocolo SPI para ajustar o ganho desse componente antes de enviar o pulso de RF.

A Figura 44-A esta representado os sinais/dados que faz a programação de ganho do VGA via protocolo SPI e na 44-B é o componente VGA que amplifica ou atenua o sinal que chega pelos conectores SMA.

O **VGA** deve ser previamente configurado para a recepção do sinal de RF por meio dos seguintes requisitos: (i) o *clock* deve operar a 1MHz; (ii) os dados possuem 16 bits que representa a configuração de ganho em dB e habilita a configuração quando LTCH transita de 1 para 0; (iii) todas as configurações de sinais, *clock*, Data e LTCH são cronometradas no registrador de deslocamento seguindo as temporizações especificadas no *data-sheet* do VGA ([DEVICES, 2005](#)). Quando LTCH estiver em 0 ocorre a configuração e quando o sinal transita para 1 mantendo a configuração de ganho.

O controle do ganho é configurado via interface de usuário e é enviado para o módulo **Envia** onde é roteado/encaminhado esse valor (dado) para o módulo **Controle VGA** desenvolvido para preparar os conjuntos de sinais e reconfigurar o ganho em dB, enviando esse dado serialmente segundo o padrão de protocolo SPI, como representado na esquerda da Figura 44, como diagrama de tempo digital.

Além disso, foi implementado, em linguagem Verilog, uma máquina de estados para ajustar as configurações de cada temporização, as quais estão listadas na Tabela 5. Em cada ciclo de *clock* é enviado 1 bit serializado para o CI.

A.1.2.1 Arquitetura do conversor RF ADCs

O conversor **RF ADCs** recebe o sinal amplificado ou atenuado através do **VGA**. O sinal de *clock* $CLK+/CLK-$ usado no conversor é gerado pela FPGA e esse *clock* é usado para fazer o conversor **RF ADCs** fazer a conversão dos dados de analógicos para digital. E para transferir os dados para a FPGA é usado o *clock* DCO (do inglês, *Data Clock Output*) com frequência de 50MHz que está sincronizado com os dados.

Figura 45 – Conversor ADCs.

Fonte: Adaptada de [Analog Devices, Inc \(2006\)](#).

Os sinais de período de *clock* (T_{clk}), tempo de configuração (t_s e t_h), tempo do *clock* do T_{dco} e outros são gerenciados pelo conversor **RF ADC** e os demais detalhes estão na especificação do *data-sheet* ([Analog Devices, Inc, 2006](#)).

APÊNDICE B:

B.1 Diferenças entre sinais de RF e sinais de gradientes

Na Figura 46 são observadas as diferenças entre sinais de RF e sinais de gradientes. O sinal de gradiente não têm filtros implementados na FPGA e geralmente esse sinal opera com uma frequência abaixo de 1MHz para que possa passar pelos conversores Grad DACs que atua como filtro passa-baixa. Para o sinal de RF possui dois filtros passa-baixa implementado em FPGA, um filtro CIC para eliminar frequências superiores a 1MHz e um outro filtro CFIR para filtragem e correções. Para o sinal de RF que passa pelos conversores de RF DACs tem que estar entre duas frequências de corte, pois estes conversores atuam como filtro passa-banda.

Figura 46 – Exemplo de sinais de radiofrequência (abaixo) e sinal de gradiente (acima).

Fonte: Elaborada pelo autor.

B.1.0.1 Conversor de gradientes **Grad DACs**

A placa para teste também possui conversores de gradientes. A sua principal função é atuar como **filtro passa-baixa**, pois cada conversor dispõe de **DACs** duplos, os

quais operam com uma frequência de TX abaixo de 1MHz; Assim, o sinal modulante deve ficar abaixo de 1MHz (frequência do sinal).

Os conversores **DACs** recebem os dados seguindo o protocolo **SPI**. O *Serial Peripheral Interface* é um protocolo de comunicação serial síncrono utilizado em comunicações a curta distância, e estão presentes principalmente em sistemas embebidos. Os dispositivos SPI comunicam-se em modo *full duplex* com base na arquitetura mestre-escravo com um único mestre. Assim, os dados chegam bit a bit e são transmitidos de forma serial na saída analógica.

Como tinha sido observado na Figura 2, a placa versão 2.0 contém 2 conversores de gradiente **Grad DACs** e cada uma possui 2 **DACs** (**DAC-A** e **DAC-B**). Neste trabalho, cada conversor é responsável por dois canais de gradiente, sendo os canais **Grad X** e **Grad Y** pertencentes a um conversor e os canais **Grad Z** e **Grad B0** pertencentes ao segundo conversor. Os dados de cada um destes canais são transmitidos serialmente da FPGA para o conversor **Grad DAC**. Estes conversores operam com uma taxa de transmissão menor que 1MHz. Além disso, o sinal transmitido não é modulado por um NCO, e pode ser qualquer forma de onda.

Tabela 6 – Terminologia dos conversores.

Conversores de gradientes	
Grad DACs	Conjunto de conversores duplo que transmite sinais de gradientes.
Grad X	Conversor1 que usa canal DAC-A .
Grad Y	Conversor1 que usa canal DAC-B .
Grad Z	Conversor2 que usa canal DAC-A .
B0	Conversor2 que usa canal DAC-B .

Fonte: Elaborada pelo autor.

B.2 Módulos de gradientes

Na Tabela 7 são resumidos a lista de módulo de transmissão dos sinais de **Gradiente**.

B.3 Comunicação de sinais de gradientes entre o SW, a FPGA e as placas a serem testadas

B.3.1 Fluxo de sinais de gradientes

Neste trabalho também foi tratado dos sinais de gradientes os quais seguem o fluxo de dados da seguinte forma:

Tabela 7 – Módulos e submódulos de transmissão de gradientes.

Módulo	Submódulo	Descrição
Gradientes (Seção B.4)	RAM Grad	Armazena o sinal de gradiente de cada canal dos gradientes (Seção B.4.1.1). Existem quatro: RAM Grad X , RAM Grad Y , RAM Grad Z e RAM B0 pelos conversores Grad X , Grad Y , Grad Z e Grad B0 , respectivamente.
	RAM RF	Armazena os dados dos sinais de RF para serem transmitidos ao conversor do gradiente. É uma instância do módulo RAM RF (Seção B.4.1.1).
	Offset Grad	Ajusta a saída de tensão. É uma instância do módulo offset usada pelo Gradiente .
	Multiplexador	Permite selecionar a saída do sinal das memórias RAM Grad ou da RAM RF .
	Grad DAC	Serializa os dados seguindo o protocolo SPI (Seção B.4.1.3).

Fonte: Elaborada pelo autor.

- **SW → FPGA → Placa para teste:** partindo do SW passando pelos módulos de **Gradientes** até a saída dos conversores **Grad DACs** da placa para teste.
- **SW → FPGA → Placa para teste → FPGA → Placa para teste:** esta implementação também permite a retransmissão do sinal de RF demodulado para a saída dos gradientes até os conversores de **Grad DACs**.

B.3.2 Barramentos que incluem gradientes

Tabela 8 – Lista de barramentos de gradientes usados na arquitetura desenvolvida.

Barramento	Descrição
A	Envio de dados do SW para a memória RAM RF .
B	Transmissão do sinal de RF para os conversores RF DACs .
C	Recepção do sinal de RF dos conversores RF ADCs .
D	Transferência de dados da FIFO para o <i>software</i> .
E	Transferência de dados da FIFO para a RAM RF do módulo de Gradientes .
F	Envio de dados do <i>software</i> para a memória RAM Grad .
G	Transmissão do sinal de gradiente para os conversores Grad DACs .

Fonte: Elaborada pelo autor.

A Figura 47 mostra a arquitetura do HW usando sinais de gradientes.

Figura 47 – Arquitetura da comunicação entre o *software*, a FPGA e a placa em teste usando sinais de gradientes.

Fonte: Elaborada pelo autor.

B.4 Transmissão de sinais de Gradientes

A Figura 48 mostra o ambiente de teste do sinal de gradientes para analisar especificamente o funcionamento dos conversores **Grad DACs** que está integrado à placa da CIERMag. O sinal é enviado pela interface de usuário com o intuito de ser processado e propagado até a placa em teste. Para isso o sinal passa pelos módulos de **Gradientes**, pelos dois conversores **Grad DACs**, e finalmente chega ao conector RJ45. A Figura 48 mostra a saída dos quatro canais de gradiente e seus correspondentes sinais exibidos no osciloscópio com a mesma forma de onda que está representada na interface de usuário.

B.4.1 Visão geral do funcionamento dos sinais de gradientes

A Figura 49 apresenta a arquitetura desenvolvida para a transmissão do sinal de gradiente. À vista disso, a Figura 49-A mostra o percurso de módulos pelos quais é transmitido os sinais de gradientes com sentido aos conversores **Grad DACs** (barramento ① da Figura 49-A). Por outro lado, a Figura 49-B) exibe a segunda via de fluxo que envolve a retransmissão do sinal de RF demodulado (ou seja, sinal já filtrado) com direção aos conversores **Grad DACs** (barramento ② da Figura 49-B).

Figura 48 – Ambiente de experimentação da transmissão de sinais de gradientes. No lado esquerdo é mostrada a interface de usuário configurada com os sinais (1) *trapézio*, (2) *triângulo*, (3) *senoidal* e (4) *Gaussiana* as quais foram transmitidas e recebidas nos quatro canais do gradiente (**Grad X**, **Grad Y**, **Grad Z** e **B0**) visualizadas no osciloscópio (direita).

Fonte: Elaborada pelo autor.

B.4.1.1 Módulos memórias **RAM Grad** e **RAM RF**

O *software* gera e configura cada pulso e envia para as quatro memórias de gradientes (ver Figura 50). Os módulos de memórias são independentes para cada pulso, canal **Grad X**, canal **Grad Y**, canal **Grad Z** e o canal **B0**. Todas essas memórias possuem uma capacidade de 1024 endereços e transmitem os dados simultaneamente. Os endereços (*address*) estão sincronizados com o ciclo de *clock clk_memória*, que é gerado na serialização e transmitindo através da saída SDI (do inglês, *Serial Digital Interface*) para os conversores **Grad DACs**, respeitando o protocolo de comunicação SPI e as configurações de temporalização desses conversores, que está indicado na especificação do *data-sheet* (Texas Instruments, 2016).

A memória **RAM RF** do módulo **Gradientes** têm a mesma configuração que foi usada no módulo **Transmissão RF**. A memória **RAM RF** recebe os sinais do pulso de RF (sinal demodulado e filtrado) e retransmite para os conversores **Grad DACs**. O **Multiplexador** direciona para a saída da memória **RAM Grad** ou para a saída da memória **RAM RF** para os conversores **Grad DACs** dos gradientes.

Figura 49 – A) mostra o fluxo de transmissão do sinal de gradiente sentido software para os conversores **Grad DACs**. B) Segue o fluxo para transmitir o sinal de RF demodulado para os conversores de gradientes **Grad DACs**, nesta transmissão testa todos os CIs (RF DACs, VGAs, RF ADCs e **Grad DACs**.)

Fonte: Elaborada pelo autor.

B.4.1.2 Módulo Multiplexador

O módulo **Multiplexador** foi desenvolvido para criar uma opção de teste. O sinal na saída do conversor do gradiente pode vir direto das memórias dos gradientes ou receber os dados na recepção do sinal de RF demodulado que fica armazenado no módulo **RAM RF**. Com esse recurso pode-se injetar um sinal de uma outra fonte geradora direto na recepção (RX) nos conectores SMA, com o objetivo de analisar o resultado na saída dos canais de gradientes via osciloscópio.

O uso do **Multiplexador** possibilitou a criação de 2 opções de sequência de fluxo de dados que pode ser observado na Figura 49-A. Esta figura ilustra o fluxo sentido memórias do **gradiente** para os conversores **Grad DACs**. Na Figura 49-B é possível

Figura 50 – Arquitetura do módulo **RAM RF** do gradiente. Os sinais de gradientes são armazenados nas memórias **RAM RF**, e depois são organizados para serem transmitidos para os conversores. Todos os endereços são incrementados de forma sincronizada.

Fonte: Elaborada pelo autor.

observar o sinal de recepção de RF demodulado sendo retransmitido para os conversores **Grad DACs**. Essa retransmissão do sinal demodulado do RF faz com que o sinal de RF passe por todos os conversores, conversores de RF (**RF DACs**, **RF ADCs**), controle de ganho do VGA e finaliza deslocando-se pelos conversores do gradientes (**Grad DACs**) e este sinal pode ser observado via osciloscópio.

B.4.1.3 Módulo Grad DAC e conversores Grad DACs

O módulo **Grad DAC** recebe os dados das memórias dos gradientes ou da memória de RF, em seguida ocorre um ajuste de nível pelo módulo **Offset Grad**. Este módulo é o mesmo explicado na seção 3.3.1.6 e foi instanciado. Após esta transformação, os dados estão aptos para serem serializados para os conversores **Grad DACs**. O módulo **Grad DAC** é responsável por controlar essa transmissão de sinais pelo conversor **Grad DACs** que está indicado na Figura 51-A. As transmissões ocorrem nas seguintes etapas:

- os dados são guardados no registrador de *input register R* temporário de acordo com a

seleção do DACs (**DAC-A** ou **DAC-B**) na Figura 51-B; (ii) os dados são transferidos para o registrador intermediário sendo (**DAC-A register R**) ou (**DAC-B register R**) indicado na Figura 51-B; e (iii) é finalizado a transmissão com o registrador **DAC-A** na Figura 51-B-1 para a saída analógica Z ou **DAC-B** na Figura 51-B-2 para a saída analógica B0, onde os sinais são convertidos de digital para analógico. A saída SDI' indicada na Figura 51-A-1 pertence ao segundo conversor onde sai os sinais analógicos X e Y dos gradientes. Essas saídas são diferenciadas de acordo com a especificação do *data-sheet* (Texas Instruments, 2016).

Figura 51 – A) O módulo **Grad DAC** controla o fluxo de dados pelo segundo conversor **Grad DAC** indicado em B. B) Sinais que gerenciam o controle e transmissão dos dados pelo conversor, onde o **DAC-A** corresponde ao sinal do gradiente Z e **DAC-B** corresponde o sinal B0.

Fonte: Adaptada de Texas Instruments (2016).

B.4.1.4 Preparação e transmissão de dados no conversor Grad DAC

A Figura 50 apresenta as 4 memórias dos gradientes e cada uma transfere dados serializados para um DAC do conversor (referente a um canal do gradiente). As memórias **RAM Grad X** e **RAM Grad Y** são responsáveis por enviar dados para o Conversor 1 (que contém o **DAC-A** e **DAC-B**) já a **RAM Grad Y** e **RAM B0** é responsável de enviar dados para o Conversor 2 (que contém o **DAC-A** e **DAC-B**), o que totaliza 4 saídas de gradientes.

Os dados são montados da seguinte maneira: (i) ocorre a associação do identificador 01 à placa **DAC-A**; (ii) adição do dado D0 que é referente a memória **RAM Grad X** ou **RAM Grad Z**; (iii) o identificador 10 é associado à **DAC-B**; (iv) adição do dado D0' referente a memória **RAM Grad Y** ou **RAM B0**; e (v) após os dados estarem prontos, a

transmissão ocorre de forma sincronizada (*Address* sincronizado) e serializada com a *clock clk_memória*. Por fim, os dados são transferidos pelo SDI de cada canal.

Figura 52 – Representação da forma de onda seguindo o protocolo SPI. Mostra as temporizações de cada sinal de controle durante a transmissão dos dados bit a bit de forma serial. Sinais que controla dinamicamente o uso de cada conversor durante a transmissão. Módulo na FPGA **Grad DAC** data-sheet (Texas Instruments, 2016).

O *software* envia o comando para selecionar o **multiplexador** para transmitir os sinais dos gradientes **Grad X**, **Grad Y**, **Grad Z** e **Grad B0** e o módulo **Grad DAC** é responsável por gerenciar a transmissão do sinal de gradientes. O módulo **Grad DAC** também controla a leitura da memória dos gradientes, lendo todas de forma sincronizada com o mesmo *clock* de leitura (*clk_memória*). A visualização do funcionamento de cada canal só é possível via osciloscópio. A forma de onda deve ser a mesma que está na interface e com isso valida a transmissão.

A Figura 52 mostra os dados sendo serializados nos pinos SDI X Y. Na Figura 52 também é possível observar o instante em que ocorrem as mudanças do **DAC-A** para o

DAC-B. Esse padrão se repete no decorrer da transmissão.

O sinal de *chip-select* (CS) negado é responsável por ativar o carregamento de dados no primeiro registrador do conversor. Esse carregamento ocorre quando o sinal está em nível lógico alto. Para que ocorra transmissão de dados pelo conversor, duas situações precisam ocorrer concomitantemente: (i) o *chip-select* negado em nível lógico zero e (ii) o sinal LDAC negado está em nível lógico alto. Para transmitir esses dados pelo conversores **Grad DACs** foi implementado uma máquina de estado na linguagem Verilog. Esta máquina é encarregada de ajustar as temporalizações de cada sinal a qual deve seguir as especificações do *data-sheet* do componente (Texas Instruments, 2016).

A Figura 52 ilustra a transmissão de dados em duas situações. A primeira realça os dados somados com o valor de **offset** sendo transmitido o pino SDI X Y e SDI Z B0, verifique que existem inúmeras oscilações de bits sendo transmitida serialmente. Na segunda transmissão é facilmente observada a transmissão do valor do **Offset**, a qual corresponde a uma tensão de 0V no osciloscópio. Também é possível constatar mais claramente as mudanças que ocorrem na seleção dos conversores, e por fim, os sinais de controle do *clock*, *chip-select* negado e o sinal LDAC negado. Esse padrão repete-se no momento em que ocorre uma transmissão.

B.5 Interface de usuário de gradientes

B.5.1 Arquitetura do gradientes

O fluxo de interação desde a interface de gradiente até o envio dos sinal para a FPGA segue a seguinte etapas :

(i) O usuário configura os pulso de gradiente destinado para os canais Grad X, Grad Y, Grad Z e B0 em conjunto com as amplitudes via *interface gráfica* na Figura 54;

(ii) Os pulsos e os parâmetros seguem para a função geradora de pulso que está no arquivo *Pulse* através das chamadas entre os arquivos *Principal* e *Gradiente* e os possíveis pulso e ilustrado na Figura 24;

(iii) Arquivo *Gradientes* recebe as formas e configurações do pulso e grava nos seu respectivo arquivos *Grad_X*, *Grad_Y*, *Grad_Z* e *B0*;

(iv) A transmissão ocorre quando o arquivo *Comunicação* carrega todos os arquivos *Grad_X*, *Grad_Y*, *Grad_Z* e *B0* e transmite nesta ordem através do barramento para o módulo **Envia** na FPGA.

Todos esse fluxo de chamada de função entre os níveis de camadas de interface de usuários, regra de negócio e acesso a dados estão ilustrados na Figura 53.

Figura 53 – Interface de configurações, onde cada gráfico de sinais corresponde a um canal de gradientes. Nesta interface tem a opção de retransmitir o sinal de RF demodulado por um ou em todos os canais do gradiente, e ajuste do tempo de repetições dessa transmissão.

Fonte: Elaborada pelo autor.

B.6 Transmissão do pulso de Gradientes

O ambiente de configuração dos quatro pulsos que correspondem aos gradientes **Grad X**, **Grad Y**, **Grad Z**, e **B0**, visualizados em quatro painéis independentes, como mostra a Figura 54. Além disso, também contém os painéis de configuração de gradiente e painel de atualização de gradientes.

- **Painel de pulso **Grad X**, **Grad Y**, **Grad Z**, e **B0****: Em cada painel de gradiente é exibido os valores de pontos máximos e mínimos da função de pulso escolhido na caixa de combinação na parte inferior. A configuração de pulso dos demais canais também podem ser configurados e serão gravados nas quatro memórias da **RAM Grad** de capacidade 1024 pontos.
- **Painel de configuração de gradiente**:
 - **Campo Amplitude**: Para o exemplo da Figura 54, é mostrada uma amplitude máxima de 32787 correspondendo aos 16 bits do conversor **Grad DAC**. Esta opção deve ser escolhida previamente à escolha do tipo de pulso de cada canal

Figura 54 – Interface de configurações de gradiente, onde cada gráfico de sinais corresponde uma canal de gradientes. Nesta interface tem a opção de retransmitir o sinal de RF demodulado por um ou em todos os canais do gradiente, e ajuste do tempo de repetições dessa transmissão.

Fonte: Elaborada pelo autor.

ou, no caso, permite ir ajustando a combinação de amplitude e pulso de cada canal;

- Para que ocorra a transmissão dos pulsos de gradientes que estão visualizados nos painéis **Grad X**, **Grad Y**, **Grad Z**, e **B0** deve estar selecionado a opção de *Gradients*;
- Botão *Loop*: faz a gravação do pulso dos quatro canais nas suas respectivas memórias na FPGA. Após a gravação ocorre uma transmissão do pulso de gradiente, onde todos os endereços de leitura da memória estão alinhados, ocorrendo várias vezes a retransmissão desse pulso;
- Botão *Clean*: esvazia o conteúdo das quatro memórias;
- Botão *Reset*: restabelece a configuração dos conversores **Grad DACs**.

Logo após pré-configurar a transmissão e recepção do sinal de RF em um dos canais (**DAC-A/ADC-A** ou **DAC-B/ADC-B**) é necessário certificar-se que esteja em modo *loopback*, para que ocorra a recepção do sinal de RF.

- Painel de atualização de gradientes (*Update ADC Grad*):
 - Se na caixa de combinação estiver selecionada a opção **Gradients**, implica que todos os sinais que estão no Painel de pulso **Grad X**, **Grad Y**, **Grad Z**, e **B0** serão transmitidos pelos conversores de **Grad DACs**. Se a caixa de combinação do painel *Update ADC Grad* estiver selecionado *ADC-Grad X*, *ADC-Grad Y*, *ADC-Grad Z*, *ADC-B0* é escolhido um dos canais para sair o sinal de RF demodulado, porém se estiver selecionado *ADC-selectd RF* ocorre a saída de um mesmo sinal de RF demodulado em todos os canais de Gradientes;
 - Campo *N. Test*: indica o número de vezes em que esse evento vai se repetir;
 - Campo *Delay*: adiciona um intervalo de tempo de espera entre um evento e outro;
 - Botão *Test*: inicia uma sequência de eventos de acordo com o número de testes. Quando clicado ocorrerá a transmissão do pulso de RF e sua recepção sairá em um dos canais selecionados ou pode ser um sinal de uma fonte geradora conectado no SMA que liga ao conversor **RF ADC**.

A caixa de combinação referente a *clock* do conversor de gradientes, exibe as opções de frequência de transmissão que o conversor **Grad DACs** pode transmitir os dados, ou seja, controla a taxa de transmissão desse sinal pelo conversor e interfere diretamente na frequência da saída do sinal. Por exemplo, o conversor pode transmitir o sinal numa taxa de *clock* de 1MHz ou pode transmitir esse mesmo sinal em uma taxa de saída de 0,5MHz.

B.7 Resultado nos conversores Grad DACs

B.7.1 Teste nos conversores de Gradientes

1. Esse teste necessita do auxílio do osciloscópio e de um operador analisando esse sinal.
2. O *software* controla o tipo de pulso e as amplitudes aplicado em cada canal de forma independente e essa transmissão ocorre pelo barramento .
3. O *software* controla a velocidade da taxa de transmissão dos dados pelos conversores **Grad DACs** na Figura 27-B-4.
4. O *software* habilita a retransmissão do sinal de RF demodulado pelo barramento pelo conversor **Grad DACs** na Figura 27-B-4. Este cenário envolve testar todos os conversores, a qual inclui nesta ordem o conversor **RF DACs** na Figura 27-A-1, o controle de ganho do VGA na Figura 27-A-2, o conversor **RF ADCs** na

Figura 27-A-3 e finaliza testando o conversor **Grad DACs** na Figura 27-B-4. Na imagem do osciloscópio da Figura 27-B temos o sinal de RF demodulado saindo no canal **Grad X** e os demais canais como **Grad Y**, **Grad Z** e **B0** estão saindo os sinais configurados pelo *software* na aba de configurações de sinais dos gradientes mostrado na (Seção B.6) na Figura 54.

B.8 Validação dos sinais de RF e gradientes nas saídas do osciloscópio

Foram gerados sinais com função *sinc*, *seno*, *cosseno*, *triângulo* e *trapézio* usando diferentes setores, como a parte do circuito que envolve a transmissão do sinal de gradientes e sinais de RF a qual inclui os conversores **Grad DACs** e os conversores **RF DACs**, respectivamente.

Os testes dos gradientes foram analisados com o auxílio do osciloscópio e em seguida foram coletados dados para serem analisados. Os testes referentes à transmissão do sinal de RF demodulado para uns dos conversores de gradiente resulta em uma análise indireta do funcionamento dos CI (**RF DACs**, controle de ganho do **VGA** e **RF ADCs**). O resultado de ambos testes necessitam de um especialista analisando o sinal no osciloscópio.

B.8.1 Teste e análise de dados da transmissão dos gradientes nos conversores **Grad X**, **Grad Y**, **Grad Z** e **B0**

Nesta seção é apresentado um segundo experimento para verificar a correta transmissão de sinais de gradiente nos conversores **Grad X**, **Grad Y**, **Grad Z** e **B0**. Assim, foram gerados 4 sinais com funções de *seno*, *sinc*, *trapézio* e *triângulo*, respectivamente. Todos esses sinais foram ajustados à mesma escala de voltagem de 5V. A visualização foi feita diretamente no osciloscópio, como pode ser observado na Figura 55-A. Após estes ajustes, para uma melhor análise, foi feita uma amostragem de dados em uma pequena região, como mostrado na Figura 55-B, centrado em 0V.

Foi analisado uma amostragem de 2000 pontos correspondentes à região selecionada (Figura 55-B). É simples observar a presença de ruído. Dados que também são mostrados na tabela dentro da Figura 55. Este ruído pode estar associado à resolução de captura de dados no osciloscópio somado à incerteza de erro de bits gerados pelo conversor **Grad DACs** de digital para analógico. O resultado esperado é que o canal **Grad X** tivesse uma variação de tensão aproximadamente de 6V e **Grad Y** com 3,5V e os canais **Grad Z** e **B0** estivesse a mesma tensão, esse resultado confere com a Figura 55-B.

Figura 55 – Exemplo de quatro sinais de gradientes sendo transmitidos pelos conversores **Grad DACs**. Pelo primeiro conversor **DAC-A** um sinal *seno*, pelo **DAC-B** um sinal *sinc*. Pelo segundo conversor **DAC-A** um sinal trapézio e pelo **DAC-B** um sinal *triângulo*. B) Mostra que os sinais *sinc*, *trapézio* e *triângulo* tem aproximadamente a mesma variação em voltagem, sendo 2,5V, enquanto que o sinal *seno* tem uma variação de 5V. Todos eles têm a mesma duração de tempo de forma síncrona.

B.8.2 Teste da saída do sinal demodulado do Radiofrequência em alguns dos canais do (Gradientes), todos os conversores funcionando

Esse experimento é completo pois testa todos os submódulos referentes ao módulo de **Gradientes** e ao módulo de **RF** uma situação onde todos os módulos estão sincronizados entre si e também sincronizados com os conversores e controle de ganho do VGA da placa em teste. Neste teste é apresentado a transmissão do sinal de gradientes, onde o sinal de uma *sinc* de cor verde e transmitido pelo conversor 1 (DAC-B) → **Grad Y**, o sinal de um *trapézio* em cor roxa sendo transmitido pelo conversor 2 (DAC-A) → **Grad Z** e o sinal *triangular* de cor vermelha sendo transmitido pelo conversor 2 → (DAC-B) **B0** conforme é apresentado na Figura 55-B. Esses sinais estão sincronizados entre si

pois o sinal está armazenado em memórias que contêm a mesma capacidade de 1024 pontos. O conversor 1 (DAC-A) → **Grad X** está recebendo um sinal demodulado de RF que chega pelo conversor **RF ADC-A**, neste procedimento é testado os seguintes CIs nesta sequência:

(i) O sinal da transmissão de RF é modulado e transmitido pelo CI conversor **RF DAC-A**; (ii) Via conexão *loopback*, passa pelo controle de ganho do CI VGA onde é amplificado ou atenuado via módulo controle de ganho VGA; (iii) Em seguida é enviado para o CI do conversor **RF ADC-A**; (iv) Passa pelos módulos de processamentos como **Misturador**, Filtro **Filtro CIC**, **Filtro CFIR** e é armazenado no módulo **FIFO**; (v) Após isso, é encaminhado para a memória **RAM RF** contida no módulo **Gradientes**; (vi) Segue pelo mesmo processamento do sinal de gradiente, seguindo o processo de preparação dos dados e serialização desses dados seguindo o protocolo SPI; (vii) E finaliza a transmissão sendo transmitido pelo CI conversor **Grad X** onde está indicado pela figura **RF ADCs** de forma genérica, podendo receber sinal do conversor **RF ADC-A** ou **RF ADC-B**.

Esse sinal de RF retransmitido pelo conversor **Grad X** pode assumir número de pontos variando de 10 até 16383, em contrapartida o número de pontos dos sinais de gradiente é de 1024 números de pontos.

Figura 56 – Resultados de dois testes aplicados com sinais de gradientes. O primeiro teste é a transmissão do sinal de gradientes, sendo o primeiro conversor **DAC-B** o sinal de uma *sinc*, o segundo conversor **DAC-A** um *trapézio* e **DAC-B** um *triângulo*. O segundo teste inclui a retransmissão (que inclui ter passado pelos **RF DACs**, **VGAs** e **RF ADCs**) do sinal de RF demodulado passar pelo primeiro conversor **Grad DACs** (**DAC-A**) o sinal de uma *sinc*.

Fonte: Elaborada pelo autor.

Figura 57 – Fluxo de dados que ocorre na transmissão e recepção do sinal de RF e transmissão do sinal de gradientes. A) O sinal de RF é transmitido e recebido seguindo a seguinte sequência: SW→ FPGA→ placa para teste (RF DACs → conexão via *loopback* → VGA → RF ADCs) → FPGA→ SW. B) O sinal de gradiente é transmitido e segue a sequência: SW→ FPGA → placa para teste (Grad DACs). O teste completo onde é testado todos os CIs ocorre na seguinte sequência de retransmissão do sinal de RF demodulado pelos conversores do gradiente: SW→ FPGA→ placa para teste (RF DACs → conexão via *loopback* → VGA → RF ADCs) → FPGA→ placa para teste (Grad DACs).

Fonte: Elaborada pelo autor.