



Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Departamento de Engenharia Elétrica

Ítalo Vladimir Diniz Vilarim

Relatório de Estágio Integrado

Campina Grande, Paraíba, Brasil
17 de maio de 2024

Ítalo Vladimir Diniz Vilarim

Relatório de Estágio Integrado

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Sistemas Embarcados

Orientador: Prof. Gutemberg Gonçalves dos Santos Júnior, Dr.

Campina Grande, Paraíba, Brasil

17 de maio de 2024

Ítalo Vladimir Diniz Vilarim

Relatório de Estágio Integrado

Relatório de Estágio Integrado submetido à Coordenação de Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Aprovado em: ___/___/___

Prof. Gutemberg Gonçalves dos Santos Júnior, Dr.
Orientador

Prof. Danilo Freire de Souza Santos, Dr.
Avaliador

Campina Grande, Paraíba, Brasil
17 de maio de 2024

Agradecimentos

Gostaria de agradecer, primeiramente, aos meus familiares e aos meus amigos por todo o apoio e companhia durante esses 12 meses longe de casa e vivenciando novas experiências, minha jornada até aqui não teria sido a mesma sem eles. À Marília, Yan, Pedro, Mariana e Gustavo por terem sido parte crucial do meu processo de adaptação a uma nova cidade. A Lucas por ter me orientado e supervisionado durante esses meses de estágio, sua ajuda e suporte foram essenciais para que meus conhecimentos na área de sistemas embarcados e FPGA se expandissem.

Também gostaria de expressar minha gratidão ao meu orientador Gutemberg Gonçalves dos Santos Júnior, cuja orientação e apoio foram inestimáveis ao longo de minha jornada na graduação e me auxiliaram na escolha de área de interesse profissional. Seus feedbacks e encorajamento me motivaram a me esforçar ainda mais e buscar a excelência.

Por fim, sou grato aos integrantes das equipes de sistemas embarcados e de desenvolvimento de hardware da empresa Thales, cuja disposição em compartilhar suas experiências e percepções tornou o período de estágio muito mais produtivo. Suas contribuições foram inestimáveis para me ajudar a obter uma compreensão mais profunda em desenvolvimento de hardware.

Resumo

Este relatório tem como objetivo detalhar as principais atividades executadas pelo aluno Ítalo Vladimir Diniz Vilarim, do curso de graduação em Engenharia Elétrica, durante o período de estágio na empresa Omnisys Engenharia, do grupo Thales. O estágio ocorreu entre o período do dia 02/05/23 até 01/05/24, com carga horária de 30 horas semanais, focado na área de desenvolvimento utilizando FPGA para um equipamento de geração de sinais destinados a radares de rota e aproximação.

Palavras-chaves: FPGA, Radares de Rota e Aproximação, Geração de Sinais.

Abstract

This report aims to detail the main activities carried out by student Ítalo Vladimir Diniz Vilarim, from the Electrical Engineering undergraduate course, during his internship period at the company Omnisys Engenharia, part of the Thales group. The internship took place between 02/05/23 and 01/05/24, with a workload of 30 hours per week, focused on the area of development using FPGA for signal generation equipment for route and approach radars.

Key-words: FPGA, Route and Approach Radars, Signal Generation.

Lista de abreviaturas e siglas

UAEE	Unidade Acadêmica de Engenharia Elétrica
CEEI	Centro de Engenharia Elétrica e Informática
UFMG	Universidade Federal de Campina Grande
VHDL	<i>Very high speed integrated circuit Hardware Description Language</i>
ECM	<i>Electronic countermeasure</i>
EW	<i>Electronic Warfare</i>
RANRAP	<i>Random Range Program</i>
ASTERIX	<i>All Purpose Structured Eurocontrol Surveillance Information Exchange</i>
UAP	<i>User Application Profile</i>

Lista de ilustrações

Figura 1 – Fachada da Omnisys Engenharia.	3
Figura 2 – <i>TRAC-NG</i>	6
Figura 3 – <i>Metodologia de Scrum</i>	10
Figura 4 – <i>Exemplo de representação de interferência simulada</i>	12

Sumário

1	INTRODUÇÃO	1
1.1	Objetivos	1
1.2	Organização do Trabalho	2
2	LOCAL DE ESTÁGIO	3
3	FUNDAMENTAÇÃO TEÓRICA	5
3.1	Radar Primário	5
3.2	Contra medidas Eletrônicas (<i>ECM</i>)	6
3.3	<i>Random Range Program (RANRAP)</i>	7
3.4	<i>Asterix</i>	7
4	ATIVIDADES REALIZADAS	9
4.1	Implementação de <i>Test Benches</i>	10
4.2	Desenvolvimento de Interferências Simuladas	11
4.3	Implementação da Interferência <i>RanRap</i>	12
4.4	Implementação de um bloco para Gerenciamento de Memória	13
4.5	Testes em cadeia reduzida de Radar	13
5	CONSIDERAÇÕES FINAIS	15
	REFERÊNCIAS	16

1 Introdução

Este relatório abrange as atividades desempenhadas pelo estudante Ítalo Vladimir Diniz Vilarim, aluno do curso de Engenharia Elétrica da Universidade Federal de Campina Grande (UFCG), durante seu estágio na Omnisys Engenharia, uma empresa pertencente ao grupo Thales. O estágio ocorreu no período de 02 de maio de 2023 a 01 de maio de 2024, com uma carga horária de 30 horas por semana, totalizando 1560 horas. O estágio foi conduzido de forma presencial, de segunda a sexta-feira, com todas as atividades realizadas no setor de engenharia da Omnisys.

Dentre as atividades realizadas neste período, o aluno ficou responsável pelo desenvolvimento de interferidores de radar para treinamento de operadores, desenvolvimento de *test benches* em *SystemVerilog* ¹, implementação de um bloco para gerenciamento de memória e testes em radar.

O estágio ocorreu no Setor de Engenharia da Omnisys, sob a supervisão do Chefe do Setor de Engenharia e engenheiro mecânico José Carlos Freire da Rocha Júnior, com orientação do professor Gutemberg Gonçalves dos Santos Júnior.

1.1 Objetivos

Este trabalho tem como propósito detalhar as atividades realizadas durante o estágio na Omnisys Engenharia, concentrando-se na área de *FPGA* (*Field Programmable Gate Array*). O projeto abordou a simulação de alvos para o treinamento de operadores de radares de vigilância aérea, utilizando linguagens de hardware e software, tais como *VHDL* (*VHSIC Hardware Description Language*), *SystemVerilog*, C++ e *Python*. Além disso, foram utilizadas ferramentas de desenvolvimento da *Xilinx* (*AMD*,), como *Vivado* (*VIVADO...*,) e *Vitis* (*VITIS...*,).

As principais atividades desenvolvidas durante o período de estágio foram:

- Desenvolvimento de interferidores de radiofrequência de radar (*RANRAP*);
- Desenvolvimento de interferidores em protocolo *Asterix* ²;
- Desenvolvimento de *test benches* em *SystemVerilog*;
- Implementação de um bloco para gerenciamento de memória;

¹ Linguagem de descrição e verificação de hardware usada para modelar e implementar sistemas eletrônicos.

² Protocolo estruturado para o intercâmbio de informações do sistema de vigilância de tráfego aéreo

- Testes em cadeia reduzida de radar para validação de produto;

1.2 Organização do Trabalho

Este relatório está estruturado em cinco capítulos, o **Capítulo 1** contém a introdução, onde são apresentados os objetivos deste trabalho de forma sucinta, apresentando também uma visão geral do que será discutido ao longo do relatório. O **Capítulo 2** trata da apresentação da empresa, mostrando suas instalações, bem como alguns dos projetos em que a empresa está envolvida. Em seguida, no **Capítulo 3**, é apresentada a fundamentação teórica do trabalho desenvolvido, incluindo conceitos-chave relacionados à simulação de alvos para o treinamento de operadores de radares de vigilância aérea. O **Capítulo 4** é dedicado à descrição detalhada das atividades realizadas durante o estágio. E no **Capítulo 5** são apresentadas as considerações finais, onde são discutidos os resultados alcançados e as lições aprendidas durante o estágio.

2 Local de Estágio

A Thales se destaca globalmente em tecnologia avançada, direcionando investimentos para inovações digitais e tecnologias-chave, como conectividade, *big data*, inteligência artificial, cibersegurança e tecnologia quântica. Seu compromisso é construir um futuro confiável para o progresso de nossa sociedade. A empresa oferece uma gama abrangente de soluções, serviços e produtos que capacitam seus clientes - sejam eles empresas, organizações ou nações - a superar desafios nos setores de defesa, espaço, transporte, identidade digital e segurança.

Figura 1 – Fachada da Omnisys Engenharia.



Fonte: (A . . . , 2023).

A Omnisys, onde a sede é apresentada na Figura 1, subsidiária do Grupo Thales no Brasil, é uma empresa nacional de alta tecnologia, com expertise técnica e gerencial em áreas cruciais tanto para aplicação civil quanto militar. Fundada em 1997 por engenheiros brasileiros, seu escopo abrange desde vigilância e controle do espaço aéreo, defesa aérea, guerra eletrônica, até sistemas de rastreamento, aviônicos, eletrônica de mísseis e entretenimento a bordo.

Além disso, a empresa atua também no segmento espacial, a Omnisys oferece equipamentos e sistemas embarcados em satélites. Integrando a rede internacional de

Centros de Excelência do Grupo Thales, investe fortemente em tecnologia e infraestrutura industrial para elevar sua capacidade de atender tanto o mercado interno quanto o internacional.

A equipe técnica da Omnisys é composta por engenheiros de todo o país, responsáveis pelo projeto, desenvolvimento e fabricação de equipamentos e sistemas, contribuindo para a redução da dependência brasileira de tecnologias estrangeiras. A empresa também oferece suporte logístico completo por meio de um Centro de Serviços equipado com tecnologia de ponta. Com equipes treinadas e laboratórios bem equipados, este centro garante manutenção corretiva e preventiva, suporte técnico e treinamento, resultando em uma alta disponibilidade operacional de seus equipamentos.

Com sede em São Bernardo do Campo (SP), a empresa expandiu sua presença e atualmente também possui filiais em São José dos Campos (SP), Rio de Janeiro (RJ) e Brasília (DF).

3 Fundamentação Teórica

Com o intuito de embasar as atividades realizadas durante o estágio e fornecer um contexto técnico adequado, este capítulo apresenta a fundamentação teórica que norteou o desenvolvimento das atividades realizadas.

3.1 Radar Primário

Um radar de vigilância primário (do inglês *Primary Surveillance Radar - PSR*) opera emitindo ondas eletromagnéticas através de uma ampla área do espaço e, em seguida, captando os sinais refletidos dos objetos dentro dessa área. Este tipo de radar é usado principalmente no controle de tráfego aéreo para detectar e localizar alvos potenciais ([PRIMARY...](#)).

Ao contrário dos radares secundários, que dependem de aeronaves que transmitem dados através de transponders de radar, os radares primários rastreiam continuamente os alvos sem exigir a sua cooperação. Sendo assim, ele realiza varreduras rápidas de 360 graus com um ângulo de elevação fixo, fornecendo medições precisas de distância e velocidade radial dos alvos. No entanto, determinar a altitude e a velocidade real do alvo geralmente requer sistemas de radar adicionais.

O radar primário tem como vantagem não exigir equipamento de bordo nas aeronaves para detecção e ser capaz de monitorar movimentos de veículos terrestres. Porém, também apresenta suas limitações, como a incapacidade de identificar diretamente os alvos e a altitude. E, também, uma área de cobertura restrita em virtude do seu uso de emissões poderosas.

Originalmente, os radares primários eram usados principalmente para aeronaves que se aproximavam de aeroportos. Contudo, à medida que os volumes de tráfego aéreo aumentam e os céus se tornam mais congestionados, há uma necessidade crescente de alargar a cobertura da vigilância primária para além das áreas de aproximação e para o espaço aéreo em rota, a fim de otimizar a utilização do espaço aéreo.

Em 2017, a Thales anunciou o *TRAC-NG*, o radar de rota mais avançado do mundo para operações civil/militares ([THALES...](#), jun). O *TRAC-NG* é um sistema avançado de radar primário operando na Banda L com alcance de até 250 NM. Ele foi projetado para fornecer cobertura de vigilância primária confiável em grandes extensões de espaço aéreo em rota.

O *TRAC-NG*, representado na Figura 2 oferece recursos opcionais como altimetria e a capacidade de rastrear vários tipos de alvos, incluindo helicópteros. Ele também inclui

recursos de nível militar, como agilidade de frequência, detecção de estroboscópio de interferência e um seletor de frequência menos congestionado, aumentando sua resiliência contra contramedidas eletrônicas.

O uso desse radar oferece benefícios que reduzem os custos do ciclo de vida. Seu design compacto e arquitetura otimizada minimizam o consumo de energia e facilitam a instalação nos locais. Orçamento de energia aprimorado, estabilidade do sistema, faixa dinâmica e largura de banda instantânea melhoram o desempenho de detecção, garantindo uma operação confiável mesmo em cenários operacionais complexos.

Figura 2 – *TRAC-NG*



Fonte: ([THALES...](#), jun).

3.2 Contramedidas Eletrônicas (*ECM*)

A guerra eletrônica (*EW*) envolve a detecção, interpretação, controle ou interrupção de sinais dentro do espectro eletromagnético (*EM*), comumente incluindo radar, rádio ou transmissões infravermelhas. O seu objetivo é salvaguardar os ativos militares, mitigando ameaças potenciais.

As Contramedidas Eletrônicas (*ECM*) desempenham um papel crucial na guerra eletrônica (*EW*), visando perturbar os sistemas de vigilância do adversário, sejam eles baseados em radar ou comunicações por rádio, além de neutralizar armas que utilizam sistemas eletromagnéticos, infravermelhos ou laser para orientação ou direcionamento. Essas contramedidas são realizadas principalmente por dois métodos: interferência e uso de chamarizes ([ELECTRONIC...](#)).

O primeiro método, conhecido como *jamming*, consiste em utilizar transmissões para interromper os canais de comunicação do inimigo ou saturar seu radar, dificultando a detecção de alvos. Embora isso impeça o inimigo de receber informações por esses canais, também impede que a fonte de interferência obtenha inteligência através da leitura dos sinais. Além disso, os modernos sistemas de comunicação com mudança de frequência tornam mais difícil interceptar efetivamente essas transmissões. O bloqueio de ruído simples ainda é amplamente utilizado em conflitos terrestres, sendo uma aplicação importante dos *jammers* operados remotamente, que podem ser empregados manualmente, disparados por artilharia, lançados de aeronaves ou utilizados em veículos aéreos não tripulados, servindo como bloqueadores temporários para operações específicas.

O segundo método de *ECM* envolve o uso de chamarizes, seja para enfrentar ameaças eletromagnéticas ou para contrariar dispositivos infravermelhos. O uso de chamarizes remonta a mais de 50 anos, desde a Segunda Guerra Mundial, e o material utilizado pouco mudou desde então. No entanto, os métodos de dispersão evoluíram, variando de acordo com o tipo de plataforma empregada. Para contramedidas infravermelhas, os cartuchos de chamas são ejetados de distribuidores, muitos dos quais têm a função de transportar tanto os chamarizes quanto às chamas.

3.3 *Random Range Program (RANRAP)*

O *RanRap*, *Random Range Program*, é um tipo de contramedida eletrônica utilizada na guerra eletrônica (*EW*) para confundir radares. Consiste na geração de múltiplas réplicas do alvo original em diferentes posições, de forma aleatória. Essas réplicas, conhecidas como “alvos falsos”, são criadas com o propósito de sobrecarregar ou desorientar o radar inimigo, dificultando a identificação do alvo real. O *RanRap* representa uma estratégia eficaz para interferir na detecção e rastreamento de alvos, sendo uma das diversas técnicas utilizadas na guerra eletrônica contemporânea ([ELECTRONIC...](#)).

3.4 *Asterix*

ASTERIX, uma abreviação para *All Purpose Structured Eurocontrol Surveillance Information Exchange*, é um padrão para a troca de informações de serviços de tráfego aéreo (*ATS*). Desenvolvido e mantido pela organização europeia *Eurocontrol*, o *ASTERIX* é um formato de dados de vigilância que está sendo adotado pela comunidade mundial de usuários como o padrão universal nesse domínio atualmente ([ASTERIX...](#)).

ASTERIX é um padrão extensível com várias categorias diferentes, cada uma lidando com um tipo específico de informação. Isso inclui relatórios de alvos de sensores de vigilância, como radares, bem como informações processadas, como trajetórias de aero-

naves e vários tipos de mensagens de status do sistema. Cada categoria define uma série de itens de dados que podem ser transmitidos em uma mensagem. Quais desses itens são transmitidos são definidos em um Perfil de Aplicação do Usuário (*UAP*), geralmente o *UAP* padrão fornecido pelo documento padrão para a categoria, mas opcionalmente um *UAP* mais especializado negociado entre o remetente e o destinatário separadamente.

4 Atividades Realizadas

Durante o período de estágio, o aluno foi designado para participar de um projeto envolvendo a simulação de cenários destinados ao treinamento de operadores de radares primários de vigilância aérea. Dentro desse contexto, sua atribuição principal era desenvolvimento de hardware. Sob a supervisão e orientação direta de um desenvolvedor júnior, ele recebeu instruções sobre suas responsabilidades e atividades a serem realizadas. Essas tarefas incluíam a elaboração de *designs* de hardware, a implementação de *test benches*, a realização de testes e a resolução de problemas técnicos conforme necessário. Ao longo do estágio, o aluno teve a oportunidade de adquirir experiência prática no desenvolvimento de soluções de hardware, trabalhando em colaboração com colegas mais experientes e contribuindo para o progresso do projeto.

Para a execução das atividades, uma variedade de ferramentas foram utilizadas, o *Vivado* foi uma das principais ferramentas, proporcionando um ambiente de desenvolvimento integrado para compilar e simular os códigos desenvolvidos em *VHDL* e *SystemVerilog*. Também foi usado o *Vitis* para o desenvolvimento de software e a integração com o hardware desenvolvido. O *ModelSim* também foi empregado para a simulação de circuitos digitais, permitindo a verificação do comportamento dos componentes projetados.

Para a edição e gerenciamento dos códigos-fonte, foi adotado o *Visual Studio Code* (*VSCode*). O *Picocom* ([PICOCOM...](#)) foi utilizado para comunicação serial com dispositivos externos, enquanto o *Wireshark* foi empregado para análise de tráfego de rede e depuração de protocolos de comunicação.

Na organização e estruturação das atividades, foram utilizadas algumas plataformas e ferramentas colaborativas para o gerenciamento e documentação do progresso do projeto. O *Jira* ¹, o *Confluence* ² e o *Bitbucket* ³ foram as principais ferramentas utilizadas para esse fim.

O *Jira* foi empregado para o acompanhamento e gestão das tarefas do projeto. Por meio dessa plataforma, foram criados e atribuídos níveis para cada atividade, permitindo uma visão clara das responsabilidades de cada membro da equipe e o status atual de cada tarefa. Além disso, no *Jira* foram criadas *sprints* e a definição de metas de curto prazo para o avanço do projeto.

O *Confluence* foi usado como uma ferramenta de documentação centralizada, onde foram armazenadas informações como requisitos do projeto, documentação técnica, pro-

¹ Disponível em: <<https://www.atlassian.com/br/software/jira>>

² Disponível em: <<https://www.atlassian.com/br/software/confluence>>

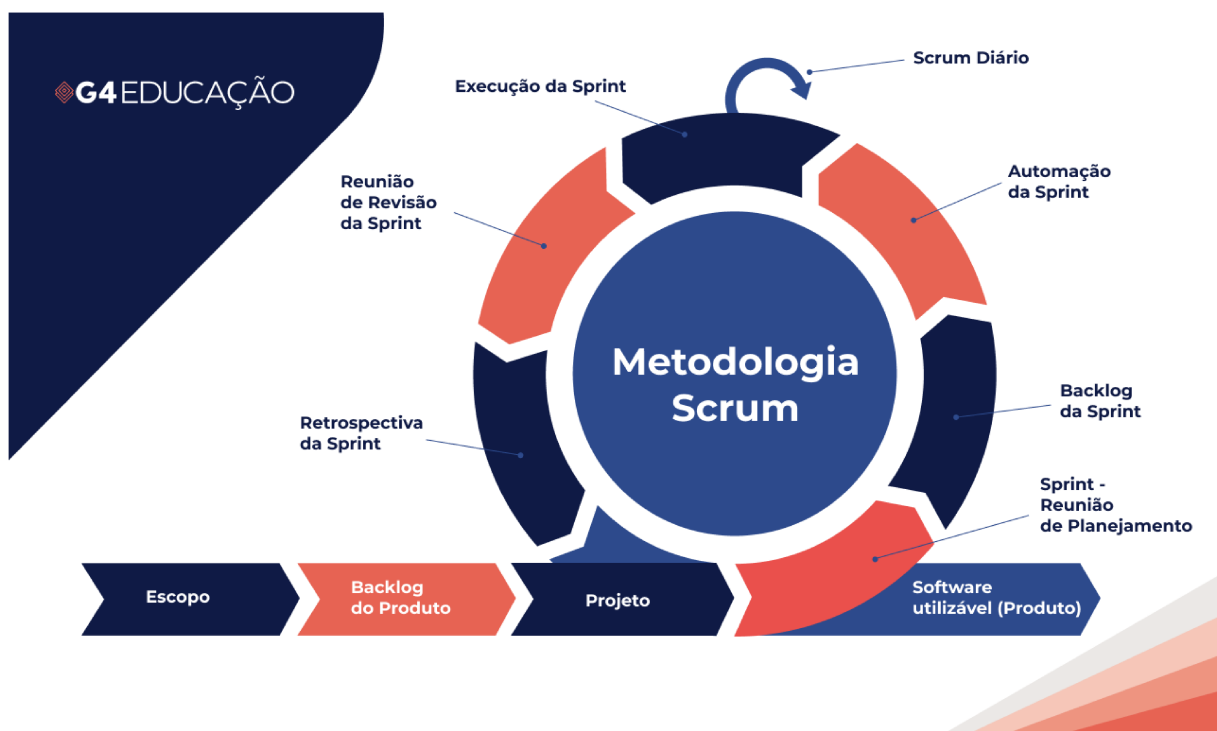
³ Disponível em: <<https://bitbucket.org>>

cedimentos de teste e resultados de análises. Essa plataforma serviu como um repositório de conhecimento para o acesso de todos os membros da equipe, garantindo a integridade das informações do projeto.

Por fim, o *Bitbucket* foi utilizado como um sistema de controle de versão para o gerenciamento do código-fonte do projeto, permitindo o armazenamento e organização do código, além de facilitar a colaboração entre os membros da equipe por meio de recursos de revisão de código e integração contínua.

Quanto à metodologia de gerenciamento de projetos, o método adotado foi baseado no *framework Scrum*, representado no diagrama da Figura 3. Esse método envolveu a realização de três reuniões semanais de acompanhamento das atividades em andamento, uma reunião semanal de revisão de tarefas em teste e uma reunião a cada duas semanas para o planejamento da próxima *sprint*. Cada *sprint* teve uma duração de duas semanas, durante as quais as atividades foram executadas de acordo com o planejamento estabelecido.

Figura 3 – Metodologia de Scrum



Fonte: (METODOLOGIA..., 2024).

4.1 Implementação de *Test Benches*

Uma das atividades iniciais atribuídas ao aluno, foi de implementar *test benches* para diversos *designs* do projeto, além de aprimorar os *test benches* já existentes. Esses *designs* haviam sido elaborados utilizando a linguagem *VHDL*, proporcionando ao esta-

giário a oportunidade de se familiarizar com essa linguagem de descrição de hardware com a qual não havia tido contato previamente.

A implementação dos *test benches* em *SystemVerilog* possibilitou ao estagiário explorar e utilizar as diversas funcionalidades e recursos oferecidos por essa linguagem para facilitar as simulações. A escolha pelo *SystemVerilog* se deu, em parte, devido à sua ampla gama de funções que auxiliam na criação de *test benches* mais eficientes.

Em alguns casos, foi necessário desenvolver *wrappers* para integrar os designs em *VHDL* com os *test benches* em *SystemVerilog*. Esses *wrappers* desempenharam um papel na comunicação entre os componentes desenvolvidos em diferentes linguagens, garantindo a correta interação durante as simulações.

4.2 Desenvolvimento de Interferências Simuladas

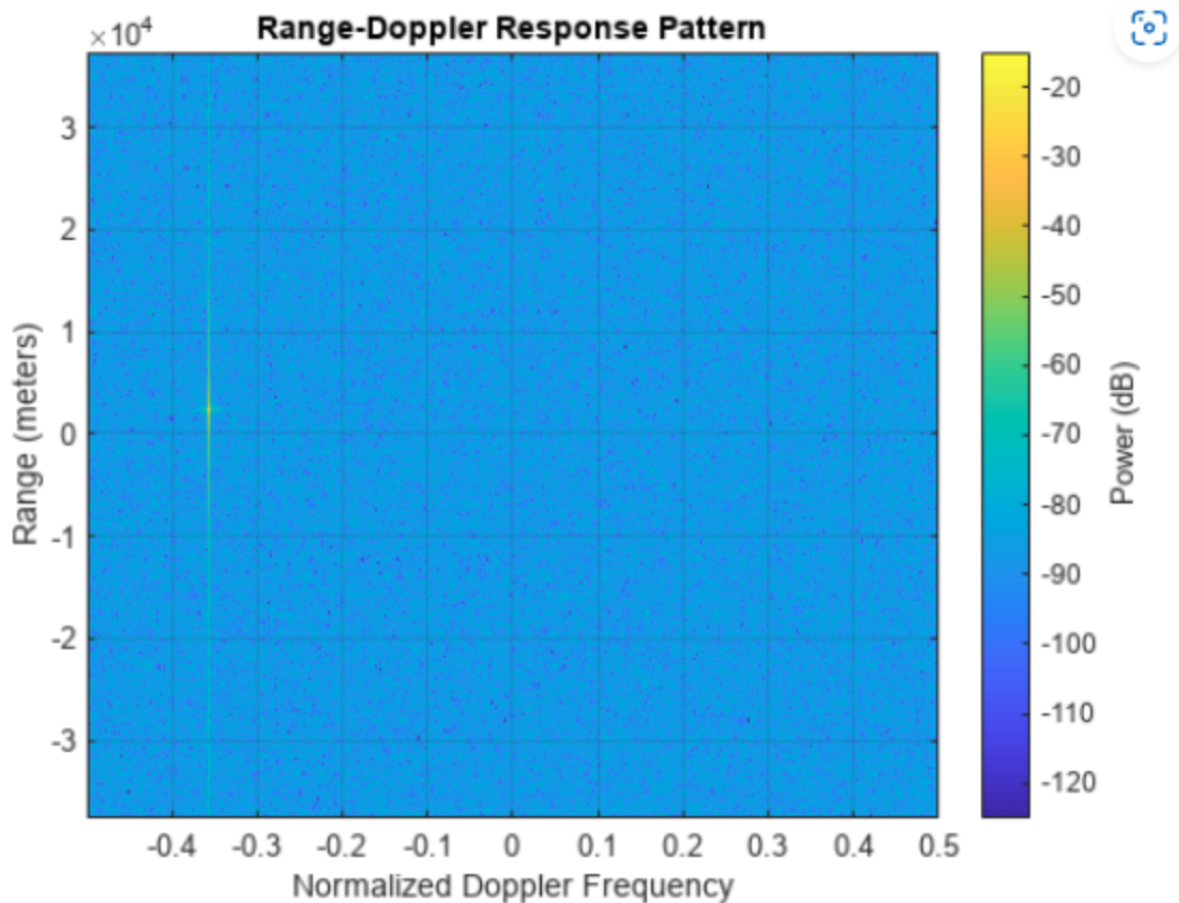
Posteriormente, o estágio progrediu para o desenvolvimento de interferências simuladas destinadas ao treinamento de operadores de radares. Essas interferências foram transmitidas por meio de pacotes *Asterix*, uma estrutura de dados amplamente utilizada na aviação para comunicação de informações de vigilância. Para essa finalidade, o estagiário empregou a linguagem de programação *Python* para gerar os pacotes *Asterix* correspondentes a cada tipo de interferência de radar.

Entre os tipos de interferência simulados estavam o *jamming* contínuo, *jamming* pontual, *jamming* em barragem, *jamming* assíncrono e *jamming* de varredura, cada um com suas características e impactos específicos nos sistemas de radar. O processo de geração dos pacotes *Asterix* envolveu a implementação de algoritmos em *Python* para simular esses diferentes tipos de interferência, levando em consideração seus parâmetros e comportamentos típicos.

Além disso, o estagiário utilizou *Python* para representar graficamente as interferências simuladas, gráficos e diagramas foram criados para ilustrar a forma de onda, intensidade e padrões temporais das interferências, resultando em uma representação visual detalhada dos efeitos que seriam observados pelos operadores de radar durante o treinamento.

A visualização das interferências simuladas apresentava uma interface que representava a relação entre a potência em decibéis por alcance em metros e a frequência em hertz como apresentado na Figura 4.

Figura 4 – Exemplo de representação de interferência simulada



Fonte: (RANGE-DOPPLER...).

4.3 Implementação da Interferência *RanRap*

Durante o período de estágio, a principal atividade empreendida foi a concepção do módulo de interferência *RANRAP* (*Random Range Program*). Esta interferência tem como objetivo simular repetidamente um mesmo alvo em diferentes posições, garantindo que os alvos apresentem as mesmas características do alvo original. Para alcançar esse propósito, foi necessário desenvolver um algoritmo que capturasse os dados do alvo original e os replicasse para os alvos *RANRAP*. O usuário tinha a capacidade de determinar a quantidade de alvos *RANRAP* desejados e a distância entre eles, sendo o número mínimo de alvos igual a 1 e o máximo 30, com uma distância mínima entre cada alvo de 15 km e máxima de 150 km.

Foi utilizada a ferramenta “*draw.io*” para a elaboração da arquitetura do módulo *RANRAP*. A arquitetura foi dividida em vários módulos, cada um responsável por uma função específica. Um módulo foi destinado à implementação de uma Máquina de Estados Finitos (*FSM*) para cada estado de operação da interferência. Outro módulo foi designado para a geração dos sinais do interferidor, garantindo que possuíssem as mesmas caracterís-

ticas do alvo original. Por fim, um arquivo de topo foi desenvolvido para instanciar todos os módulos necessários. Cada um desses módulos foi implementado em *VHDL*, acompanhado de um *test bench* correspondente em *SystemVerilog*. Esses componentes foram simulados utilizando o ambiente de desenvolvimento *VIVADO*.

4.4 Implementação de um bloco para Gerenciamento de Memória

Durante a elaboração do módulo *RANRAP*, uma das principais limitações identificadas foi relacionada ao número de alvos gerados e sua ocupação de memória. Para superar esse desafio, foi reconhecida a necessidade de desenvolver um bloco de gerenciamento de memória, com o intuito de otimizar o uso da memória e reduzir sua ocupação.

A lógica por trás do bloco de gerenciamento de memória envolveu a implementação de uma estrutura que utilizava um número reduzido de memórias. Para isso, foi empregada uma Máquina de Estados Finitos (*FSM*), que controlava o funcionamento da memória em três estados distintos: *IDLE*, *READ* e *WRITE*.

No estado *IDLE*, a memória estava disponível para receber novos dados. Quando um pulso de escrita era recebido, a memória transitava para o estado *WRITE*, onde os dados eram armazenados. Após a conclusão da escrita, a memória retornava ao estado *IDLE*.

Por outro lado, quando uma solicitação de leitura era recebida e a memória estava no estado *IDLE*, ela transitava para o estado *READ*, indicando que estava ocupada realizando a leitura dos dados. Se a memória estivesse ocupada no momento da solicitação de leitura, a entrada correspondente era direcionada para a próxima memória disponível. No caso de nenhuma memória estar disponível, a entrada era simplesmente ignorada. Após a conclusão da leitura, a memória retornava ao estado *IDLE*, pronta para receber novas operações de escrita ou leitura.

Essa abordagem de gerenciamento de memória forneceu uma solução eficiente para lidar com as restrições de ocupação de memória no módulo *RANRAP*. Ao utilizar uma lógica projetada e uma *FSM* bem definida, foi possível otimizar o desempenho e garantir um uso mais eficiente dos recursos de memória disponíveis, contribuindo para elevar a funcionalidade e a escalabilidade do módulo *RANRAP*.

4.5 Testes em cadeia reduzida de Radar

Para validar as implementações realizadas durante o estágio, foram adotadas várias estratégias de teste e verificação. Além das simulações realizadas utilizando o arquivo de topo, foi gerado o bitstream do projeto como um todo. Esse *bitstream* representa a

configuração do hardware programável (*FPGA*) com todas as funcionalidades implementadas.

Em seguida, foram conduzidos testes em cadeia reduzida no radar, também conhecidos como testes de Integração, Verificação, Validação e Qualidade (IVVQ). Esses testes foram realizados utilizando o software interno desenvolvido pela empresa específico para o radar, em conjunto com o ambiente de desenvolvimento integrado (do inglês *Integrated Development Environment - IDE*) *VITIS*.

Durante os testes de IVVQ, o *bitstream* gerado foi carregado no hardware do radar para avaliar o desempenho e a integração das implementações com o sistema real. Esses testes permitiram verificar se as funcionalidades desenvolvidas atendiam aos requisitos especificados e se o sistema funcionava conforme o esperado em condições reais de operação.

Além disso, foram realizados testes de verificação para garantir a correta funcionalidade de cada módulo e componente do sistema. Esses testes foram conduzidos em diferentes níveis de granularidade, desde testes unitários de cada módulo até testes de integração para verificar a interação entre os diversos componentes.

5 Considerações Finais

Este relatório apresenta as atividades realizadas durante o estágio na *Omnisys* Engenharia e ressalta a relevância dessa experiência para formação profissional e acadêmica do aluno. Durante o tempo na empresa, foi possível aplicar os conhecimentos adquiridos na graduação em um ambiente de trabalho prático e desafiador.

O projeto do qual fez parte ativamente aprofundou os conhecimentos na área de desenvolvimento em hardware. Trabalhar em colaboração com uma equipe e receber orientação direta de um engenheiro foi importante para aprimorar habilidades técnicas e compreender melhor o processo de desenvolvimento de sistemas complexos.

Durante o estágio, houve a utilização de uma variedade de ferramentas e tecnologias, desde linguagens de programação como *VHDL*, *SystemVerilog*, *C++* e *Python* até ferramentas de desenvolvimento como *Vivado*, *Vitis* e *ModelSim*. Essa experiência prática ajudou a consolidar o conhecimento teórico e desenvolver a capacidade de resolver problemas de forma eficaz. Além disso, o estágio proporcionou a oportunidade de aprender sobre métodos de gerenciamento de projetos, como o método Scrum, e utilizar ferramentas de colaboração e documentação, como *Jira*, *Confluence* e *Bitbucket*.

Algumas disciplinas cursadas ao longo da graduação em Engenharia Elétrica foram fundamentais para a realização das atividades. As disciplinas de Circuitos Lógicos e Laboratório de Circuitos Lógicos forneceram a base necessária para a compreensão e implementação de módulos em *VHDL* e *SystemVerilog*. Arquitetura de Sistemas Digitais contribuiu para o entendimento das ferramentas de desenvolvimento como *Vivado* e *Vitis*, essenciais para a simulação e verificação de hardware. A disciplina de Redes foi importante para a familiarização com protocolos e estruturas de dados, como o Asterix.

Sendo assim, o estágio na *Omnisys* Engenharia foi uma oportunidade extremamente valiosa que contribuiu significativamente para crescimento pessoal e profissional.

Referências

A empresa brasileira de defesa Omnisys anuncia seu novo presidente. 2023. Disponível em: <<https://www.defesaemfoco.com.br/a-empresa-brasileira-de-defesa-omnisys-anuncia-seu-novo-presidente/>>. 3

AMD. Disponível em: <<https://www.amd.com/en.html>>. 1

ASTERIX – Protocolo Estruturado para o Intercâmbio de Informações do Sistema de Vigilância de Tráfego Aéreo. Disponível em: <<https://sirius.decea.mil.br/en/asterix-protocolo-estruturado-para-o-intercambio-de-informacoes-do-sistema-de-vigilancia-de-trafego>>. 7

ELECTRONIC Warfare Fundamentals. Disponível em: <<https://falcon.blu3wolf.com/Docs/Electronic-Warfare-Fundamentals.pdf>>. 6, 7

METODOLOGIA Scrum: o que é, para que serve e exemplos para aplicar no seu negócio. 2024. Disponível em: <<https://g4educacao.com/portal/metodologia-scrum>>. 10

PICOCOM - Empowering Wireless. Disponível em: <<https://picocom.com>>. 9

PRIMARY SURVEILLANCE RADAR. Disponível em: <<https://www.terma.com/products/radars/primary-surveillance-radar/>>. 5

RANGE-DOPPLER response. Disponível em: <<https://kr.mathworks.com/help/phased/ref/phased.rangedopplerresponse-system-object.html>>. 12

THALES unveils TRAC NG world's most advanced en-route radar for dual civil/military operators. jun. Disponível em: <<https://www.thalesgroup.com/en/worldwide/aerospace/press-release/thales-unveils-trac-ng-worlds-most-advanced-en-route-radar-dual>>. 5, 6

VITIS Integrated Design Environment. Disponível em: <<https://www.xilinx.com/products/design-tools/vitis/vitis-ide.html>>. 1

VIVADO Overview. Disponível em: <<https://www.xilinx.com/products/design-tools/vivado.html>>. 1