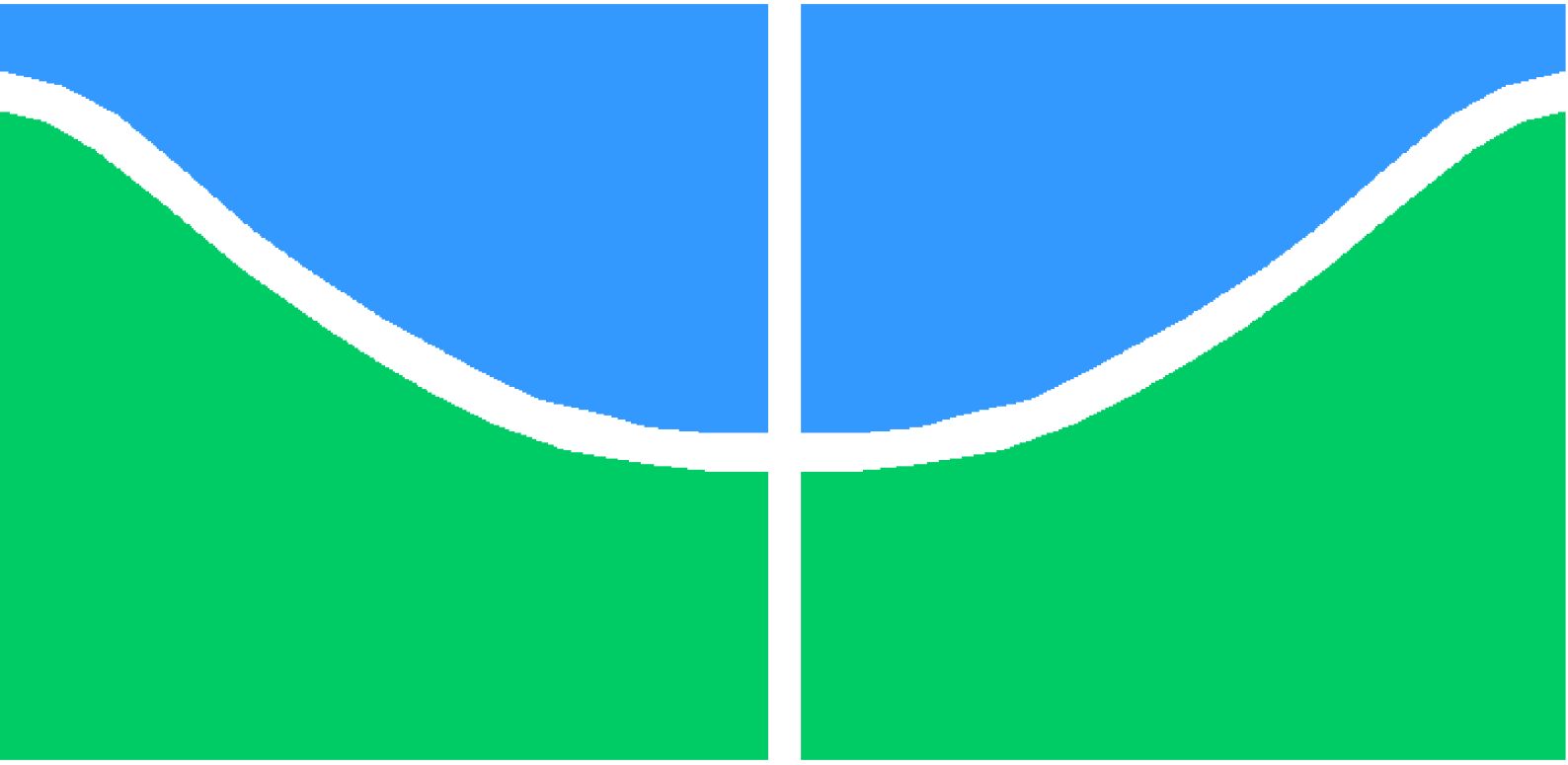

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Módulo Transmissor e Receptor OFDM sintetizado em hardware FPGA

Autor: Filipe Rodrigues da Silva
Orientador: Dr. Leonardo Aguayo
Coorientador: Dr. Daniel M. Muñoz

Brasília, DF
2019

Filipe Rodrigues da Silva

Módulo Transmissor e Receptor OFDM sintetizado em hardware FPGA

Monografia submetida ao curso de graduação em Engenharia eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Leonardo Aguayo

Coorientador: Dr. Daniel M. Muñoz

Brasília, DF

2019

Filipe Rodrigues da Silva
Módulo Transmissor e Receptor OFDM sintetizado em hardware FPGA/ Filipe Rodrigues da Silva. – Brasília, DF, 2019-
82 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Leonardo Aguayo

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2019.

1. Modulação. 2. Multiportadora. I. Dr. Leonardo Aguayo. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Módulo Transmissor e Receptor OFDM sintetizado em hardware FPGA

CDU 02:141:005.6

Filipe Rodrigues da Silva

Módulo Transmissor e Receptor OFDM sintetizado em hardware FPGA

Monografia submetida ao curso de graduação em Engenharia eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 15 de julho de 2019:

Dr. Leonardo Aguayo
Orientador

Dr. Cristiano Jacques Miosso
Convidado1

Dr. Marcelino Monteiro de Andrade
Convidado 2

Brasília, DF
2019

Primeiramente, dedico este trabalho a Deus. O autor e sustentador da minha vida. A Ele a honra, a glória, o louvor e a adoração.

Dedico aos meus pais, com todo amor, gratidão e carinho. Grato pelo apoio, suporte ao longo da minha vida.

Aos meus irmãos por serem o pilar de sustentação. Aos meus tios e primos, os quais são exemplo de esforço e dedicação. E aos meus amigos que sempre estiveram comigo e pelas palavras de motivação.

Agradecimentos

Agradeço a Deus por ter me dado forças em todo a minha vida. Sempre ao meu lado, sendo suporte e me fortalecendo.

Agradeço aos meus pais por terem dado suporte, apoio e recursos. Agradeço a Deus por possibilitá-los a estarem ao meu lado nessa fase importante da minha vida.

Registro também, de forma especial, meu agradecimento aos meus irmãos, tios, primos e amigos, os quais sempre me motivaram e deram forças para permanecer firme na caminhada.

Ao meu orientador Prof. Dr Leonardo Aguayo por ter me recebido e sempre me mostrado o caminho correto a ser seguido, de forma única, admirável e exemplar. Agradeço por ceder recursos, pela paciência, atenção, motivação e dedicação.

Da mesma forma, agradeço o Prof. Daniel Muñoz e Cristiano Miosso pelo auxílio em todas as fases desse trabalho, as observações, conversas, dicas e orientações em momentos importantes do desenvolvimento do projeto.

Agradeço em especial a meu amigo Pedro, que ajudou nas pesquisas e fez observações para melhora deste trabalho.

Por fim, agradeço a todos que indireta ou diretamente fizeram parte dessa pesquisa e que permitiram o desenvolvimento desta tese.

“ Para mim, fé começa com a compreensão de que uma inteligência suprema trouxe o universo à existência e criou o homem. Não é difícil para mim ter esta fé, pois um universo organizado e inteligente testifica a favor da maior afirmação jamais pronunciada: No princípio, Deus...

(Holly Compton,Arthur, Nobel de física, 1927)

Resumo

As dificuldades existentes no sistema de comunicação de única portadora, como por exemplo, condições ruins do meio, atenuação de altas frequências, interferência inter-símbolo, interferência causada por múltiplos caminhos, despertam estudos e análises sobre essas dificuldades. Uma maneira para analisar esse problema é o desenvolvimento, implementação e análise de um transmissor e receptor OFDM na linguagem de descrição de Hardware VHDL. O principal objetivo desse trabalho é implementar um modulador e um receptor OFDM em descrição de Hardware e utilizar um osciloscópio para a análise do sinal modulado. Para o presente projeto, foi necessário um estudo minucioso do sincronismo e a realização de um método, que contribuiu para uma eficiente implementação e visualização dos blocos no osciloscópio. A maioria dos estudos na área de codificação do sistema OFDM não apresenta de forma detalhada a implementação do sistema em linguagem de *hardware*, o que não demonstra a correta elaboração do sincronismo dos subsistemas da OFDM. Os subsistemas IFFT e FFT foram implementadas de forma eficiente, as quais apresentaram uma taxa de erro de apenas 3.2% e apresentaram rapidez nas realizações dos cálculos, embora sejam limitadas ao lidar com quantidades de bits muito elevadas, ou seja, problemas de *overflow* e limitadas quando realizam cálculos em ponto flutuante. Os subsistemas IFFT e FFT foram comparados com os valores adquiridos do software Matlab. Foi possível também, observar o sinal do transmissor no osciloscópio e utilizou a validação por escrita em arquivo na linguagem VHDL para verificação dos resultados do transmissor e receptor. A taxa de erro para o transmissor e o receptor é inferior a 8%. Os resultados obtidos por meio do trabalho realizado contribuirão para aprofundar estudos posteriores nessa área, ampliar a complexidade do subsistema da IFFT e também do transmissor OFDM.

Palavras-chaves: OFDM. IFFT. QAM. VHDL. Transmissor.Modulação.

Abstract

Difficulties in the single-carrier, communication system, such as poor media conditions, high-frequency attenuation, inter-symbol interference, multipath interference, trigger studies and analysis of these difficulties. One way to analyze this problem is the development, implementation and analysis of an OFDM transmitter and receiver in the VHDL Hardware description language. The main objective of this work is to implement a modulator and OFDM receiver in hardware description and to use an oscilloscope to analyzze the modulated signal. For the present project, it was necessary a detailed study of synchronism and the realization of a method, which contributed to an efficient implementation and visualization of the blocks in the oscilloscope. Most studies in the coding area of the OFDM system do not detail the system implementation in hardware language, which does not demonstrate the correct elaboration of OFDM subsystem synchronism. The IFFT and FFT subsystems were implemented efficiently, which had an error rate of only 3.2% and were quick to perform the calculation, although they are limited when dealing with very high bit amounts. The IFFT and FFT subsystems were compared with the values acquired from Matlab software. It was also possible to observe the transmitter signal on the oscilloscope and used file-written validation in the VHDL language to verify the transmitter and receiver results. The error rate for transmitter and receiver is less than 8%. The results obtained through the work performed will contribute to further studies in this area, increasing the complexity of the IFFT subsystem and also the OFDM transmitter.

Key-words: OFDM. IFFT. QAM.8HQAM. Transmitter.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Amplitude Modulation (Modulação em Amplitude)
ASIC	Application Specific Integrated Circuit
ASK	Amplitude-Shift keying
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate (Taxa de Erro de Bit)
BFSK	Binary Frequency Shift Keying
BPSK	Binary Phase Shift Keying
CPLD	Complex Audio Broadcast
CS	Chip Select
DAC	Digital-to-Analog Converter
DFT	Discrete Fourier Transform (Transformada Discreta de Fourier)
DTFT	Discrete Time Fourier Transfom (Transformada de Fourier em Tempo Discreto)
FDM	Frequency Division Multiplexing
FFT	Fast Fourier Transform (Transformada Rápida de Fourier)
FT	Fourier Transform (Transformada de Fourier)
FPGA	Field Programmable Gate Array
FSK	Frequency-Shift Keying
IEEE	Institute of Electrical and Electronic Engineers
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Discrete Fourier Transform
LUT	Look-Up Table
MCM	Multi Carrier Modulation (Modulação de Múltiplas Portadoras)
OFDM	Orthogonal Frequency Division Multiplexing

PAM	Pulse Amplitude Modulation (Modulação em Amplitude de Pulso)
PM	Phase Modulation (Modulação em Fase)
PSK	Phase Shift Keying
ROM	Read-Only Memory
RTL	Register Transfer Language (Nível de Transferência de Registro)
SCM	Single Carrier Modulation (Modulação de Única portadora)
SCLK	Serial Clock

Lista de símbolos

A	Amplitude do sinal senoidal
$s(t)$	Sinal contínuo OFDM no domínio do tempo
ω	Frequência angular do sinal (rad/s)
a	Termo em fase da modulação QAM
b	Termo em quadratura da modulação QAM
E	Energia do sinal QAM
t	Tempo (s)
T_s	Duração de símbolo
f	Frequência (Hz)
f_c	Frequência da portadora
N	Número de canais utilizados na OFDM
k	Índice dos canais no domínio da frequência
n	Índice dos canais no domínio do tempo
s_n	Sinal discreto OFDM no domínio do tempo
$s_c(t)$	Sinal gerado no transmissor OFDM
Θ_i	Fase do i-ésimo sinal modulado QAM
z	Número complexo
j	Unidade imaginária
$\Re$	Parte real de um número complexo
$\Im$	Parte imaginária de um número complexo
S_k	Sinal discreto OFDM no domínio da frequência
$s_{ofdm}(t)$	Sinal de transmissão OFDM
$W_N(nk)$	N-ésimo fator de rotação da DFT de N pontos

Sumário

1.1 Introdução

Os sistemas de comunicação sem fio (*wireless*) estão entre os sistemas tecnológicos mais importantes nos dias atuais (B.PUJITA, 2014). Nesses sistemas trafegam dados sem a utilização de fios ou cabos pelos quais possam fazer interconexões. Eles se destacam por suas vantagens e características importantes na comunicação, dentre as quais podem ser mencionadas o baixo custo de manutenção, instalação e viabilidade de projetos em lugares impossibilitados de habitação como, por exemplo, o espaço sideral. Também possibilitam que os usuários se comuniquem em diferentes lugares, sem a conexão de componentes físicos, como cabos, fios, entre outros (TSE; VISWANATH, 2005).

No contexto atual, há um aumento de acesso a dispositivos que atendam à população quanto a requisitos de projetos em comunicações sem fio. Assim, essa demanda tem favorecido a busca por técnicas cada vez mais eficientes na transmissão e recepção de informações. No entanto, os desafios para a implementação desses projetos são muitos. Dentre os principais problemas, pode-se destacar o efeito Doppler, interferência de canal e interferência intencional em comunicações militares (B.P.LATHI, 2010).

Os sistemas de comunicações sem fio fazem uso de ondas eletromagnéticas. Os principais sistemas de comunicação que fazem o envio e recepção desses dados são os transmissores e o receptores. Esses, a cada dia que passa, têm se tornado mais complexos, mais potentes e mais precisos (CASTRO, 2016). Os dados do transmissor não podem ser enviados diretamente através dos canais de transmissão. Esses sinais, denominados sinais modulantes, são em sua maioria de baixa frequência, o que acarreta a necessidade de uma grande antena para a transmissão e recepção.

Assim, faz-se necessário que esses sinais sejam alterados por uma onda eletromagnética portadora, conhecida como sinal portador, cujas propriedades são mais convenientes aos meios de transmissão. Essa alteração é o que se denomina modulação (B.P.LATHI, 2010).

A modulação conhecida como multiplexação por divisão de frequências ortogonais, ou OFDM, surge nesse contexto como uma solução eficiente para sistemas de transmissão e recepção. OFDM é uma modulação que se caracteriza por utilizar multiportadoras. Em 1971, Weinstein descobriu uma forma eficiente de implementar OFDM. Ele utilizou a transformada discreta de Fourier (DFT) para modular e demodular o sinal. Nessa época, a modulação não se popularizou, devido à necessidade de espaço físico nos *hardwares* (WEINSTEIN; EBERT, 1971).

No entanto, com os recentes avanços das tecnologias *Very Large Scale Integration*(VLSI), a qual tem disponibilizado barateamento e implementações rápidas de *Fast Fourier Transform* FFT, OFDM se popularizou no meio das comunicações móveis (P.Y.TSAI, 2005).

1.2 Formulação do Problema e Proposta

Analisando a grande necessidade do envio de informação e uma eficiente qualidade no sinal enviado, é notória a necessidade de que se faça uso de *hardwares* e *softwares* que possibilitem melhor acesso e rapidez no envio de informações. Hoje, a comunicação enfrenta vários problemas no envio e recepção de informações. Destacam-se interferência entre frequências, ruído impulsivo, multicaminhos, entre outras dificuldades. Assim, buscar um meio eficiente de envio de informação é de primordial importância na área de telecomunicações.

Algumas propostas para se implementar o OFDM são realizadas com microcontroladores. Eles apresentam algumas desvantagens, como a necessidade de memória, de chips periféricos para realizar operações, possui uma saída mais lenta quando comparada a outros dispositivos e usa muito espaço de memória e energia.

Uma outra possibilidade para realizar o sistema OFDM seria o uso do método *Application Specific Integrated Circuit* (ASIC). Este também apresenta limitações, como a inflexibilidade do processo de projeto envolvido e o maior tempo de comercialização para o chip projetado.

Como FPGA possibilita cálculos rápidos, eficientes, flexibilidade ao design do programa e componente de *hardware* de baixo custo em comparação com outros dispositivos, ela pode juntamente com a OFDM alcançar ótimos resultados na comunicação por meios digitais. Assim, realizar um sistema de comunição a OFDM e a FPGA com rápido envio de dados, menor interferência e menor ruído por múltiplos caminhos se faz possível e necessário. Por conseguinte, a proposta deste trabalho é implementar um transmissor e um receptor com a técnica OFDM em descrição de *hardware*, desenvolver um sistema que possa ser analisado de maneira efeciente em um dispositivo de análise de ondas, visto que, a maioria dos trabalhos nessa área não apresentam síntese, análise e o sincronismo dos subsistemas da OFDM de maneira detalhada.

1.3 Justificativa

Em meio às dificuldades apresentadas no envio de mensagens, verifica-se que é vantajoso o uso de OFDM quando comparado a técnicas que utilizam uma única portadora. Para modulações com multiportadoras, pode-se obter a mesma taxa de transferência, devido ao paralelismo de subportadoras de taxas baixas, maior resistência a condições ruins do meio, menor atenuação de altas frequências, menor interferência inter-símbolo, menor interferência causada por múltiplos caminhos e vários outros benefícios em comparação aos sistemas de uma única portadora.

Com base nisso, é importante conhecer e estudar os sistemas multiportadoras e

implementá-los em dispositivos que possam ter um rápido desenvolvimento de cálculos. Para isso, um passo importante é implementar a modulação OFDM. O sistema OFDM possui blocos, que são necessários se conhecer. E assim, pode-se utilizar a codificação em descrição de hardware, já que é um meio rápido e eficiente para cálculos, que pode implementar um modulador OFDM eficiente e robusto.

1.4 Objetivos

1.4.1 Objetivo Geral

Aplicar os conceitos e técnicas desenvolvidas pela engenharia eletrônica na área de telecomunicações para desenvolver, implementar e analisar um transmissor e um receptor OFDM na linguagem de descrição de Hardware VHDL.

1.4.2 Objetivos Específicos

- Descrever as diferentes partes constituintes dos blocos utilizados na modulação e demodulação OFDM.
- Criar um bloco transmissor e um bloco receptor OFDM em linguagem de descrição de *hardware*.
- Analisar os sinais gerados do módulo OFDM em um instrumento de medida de sinais elétricos e eletrônicos, o qual possa apresentar o sinal de saída do transmissor OFDM.
- Obter os valores de saída do bloco transmissor e receptor OFDM por meio da escrita e leitura em arquivo, com linguagem VHDL.
- Validar os dados do transmissor e receptor utilizando o software Matlab.

1.5 Metodologia

Este trabalho teve como ponto de partida pesquisas, leituras e análises de diversos textos relacionados ao tema. Em seguida foram implementados em VHDL cada bloco da modulação e recepção OFDM. Logo após, foram validados, separadamente, cada subsistema por intermédio do seu *testbench*, que funciona como um ambiente para verificar a exatidão ou a integridade de um projeto. Os valores do transmissor e receptor foram escritos em arquivo. Essa escrita possibilitou a comparação dos dados de entrada e de saída. Com a validação de cada bloco, foi possível instanciar na FPGA o código e verificar o sinal de saída em um osciloscópio.

1.6 Esboço do Projeto

O presente trabalho ficou estruturado em seis capítulos, a saber introdução, revisão bibliográfica, metodologia, implementação dos blocos na linguagem VHDL, resultados e discussões e conclusão.

Em termos fundamentais o primeiro capítulo apresenta a ideia geral do projeto, a introdução, a justificativa, o objetivo do projeto e o escopo do projeto.

No capítulo dois é feito um estudo dos temas abordados pelo trabalho de forma a facilitar o entendimento quanto ao que será implementado. São apresentadas algumas vantagens e desvantagens dos sistemas de modulação de única portadora e múltiplas portadoras.

No terceiro capítulo são descritos os materiais e métodos para o trabalho. Assim, são descritos os principais *softwares* utilizados e o passo a passo para a realização do trabalho.

No quarto capítulo é descrita a implementação dos subsistemas da OFDM. É descrita detalhadamente a construção dos submódulos do transmissor e receptor OFDM em linguagem VHDL.

A partir desses módulos, é possível obter os resultados, assim como análises, as quais são comparadas com os resultados obtidos pelas simulações do Matlab. Isso é apresentado no quinto capítulo. Os resultados são descritos em tabelas e imagens que descrevem de maneira clara os resultados obtidos.

Finalmente, o sexto capítulo apresenta a conclusão e propostas para trabalhos futuros.

2 Fundamentação Teórica

2.1 Modulações Digitais

2.1.1 Histórico das Modulações

Ao longo dos anos, utiliza-se a banda de frequência disponível para transmitir os dados. Essa implementação é denominada de *single carrier modulation* (SCM), ou modulação com única portadora. Nesse sistema os dados gerados são transmitidos de forma sequencial. Dentre as modulações que utilizam essa técnica podem-se destacar ASK, FSK, PSK, QAM, PAM (SIQUEIRA, 2004).

Lathi define modulação como o processo no qual varia-se uma ou mais propriedades de uma forma de onda periódica, chamada de portadora, com um sinal de modulação que normalmente contém informações a serem transmitidas . A modulação é um processo de transmissão de sinal da mensagem, como por exemplo, um fluxo de bits digital ou um sinal de áudio analógico, dentro de outro sinal que pode ser fisicamente transmitido (B.P.LATHI, 2010).

Um modulador ou transmissor é um dispositivo que realiza modulação. Um demodulador é um dispositivo que realiza a demodulação, o inverso da modulação. O modem pode realizar ambas as operações. Nessa perspectiva a literatura se baseia em dois tipos, a modulação e demodulação analógica e a modulação e demodulação digital (B.P.LATHI, 2010).

Apesar da grande maioria dos sistemas de tecnologias sem fio utilizarem modulação com única portadora, esse tipo de implementação possui alguns problemas, os quais tornam o projeto muitas vezes inviável. Uma das desvantagens está no fato de que a resposta em frequência não é linear. Dessa forma, faz se necessária a utilização de equalizadores, os quais, na maioria das vezes, são muito complexos e de alto custo de fabricação. Outra dificuldade nesse sistema é o multipercurso, o qual necessita de equalizadores adaptativos para sincronização de sistemas. Como descreve Bahai(2002), esses equalizadores são de difícil fabricação e de custo muito elevado (BAHAI, 2002).

2.1.2 Modulação FSK

A modulação FSK (*Frequency Shift Keying*), uma modulação de única portadora, consiste na variação da frequência da onda portadora em função do sinal digital a ser transmitido. A amplitude da onda portadora é constante durante o processo de modulação e a onda resultante varia a sua frequência conforme os níveis lógicos do sinal modu-

lante (B.P.LATHI, 2010). A tecnologia é usada para sistemas de comunicação como rádio amador, identificador de chamadas e transmissões de emergência. O FSK mais simples é o FSK binário (BFSK), o BFSK usa um par de frequências discretas para transmitir informações binárias (0s e 1s). A técnica de modulação FSK usa duas frequências portadoras diferentes para representar o binário 1 e o binário 0. A frequência de portadora f_{c1} representa os dados binários de valor um e a frequência portadora f_{c2} representa os dados binários de valor zero. A amplitude e a fase da portadora permanecem constantes enquanto a frequência da portadora é alterada. A imagem 1 apresenta a representação do BFSK. O FSK binário (BFSK) pode ser representado seguindo a equação matemática (XIONG, 2000):

$$s(t) = Acos(2\pi f_{c1} t) \tag{2.1}$$

para binário 1

$$s(t) = Acos(2\pi f_{c2} t) \tag{2.2}$$

para binário 0.

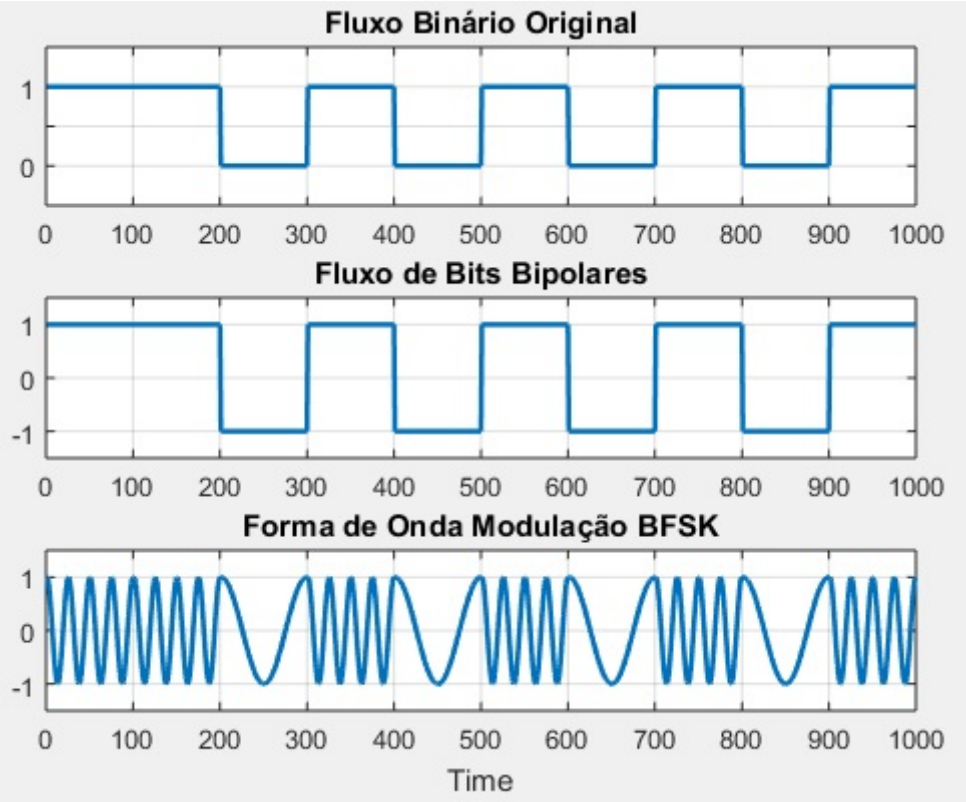

Figura 1 – Exemplo de BFSK com entrada de bits e sinal modulado.

Lathi apresenta a modulação FSK com uma baixa probabilidade de erro (B.P.LATHI, 2010). O FSK tem uma alta imunidade ao ruído devido à envoltória constante, além de

tornar o sistema mais robusto contra a variação na atenuação através do canal. As desvantagens para tais sistemas se caracterizam com o uso de banda maior em comparação com outras técnicas de modulação, como ASK e PSK. Por isso, não é eficiente na largura de banda e a sua performance da taxa de erro de bit (BER) no canal AWGN é pior do que a modulação PSK.

2.1.3 Modulação PSK

A modulação PSK (*Phase Shift Keying*) é aquela em que há variação da fase da onda portadora em função do sinal digital a ser transmitido(B.P.LATHI, 2010). A modulação é realizada variando as entradas de seno e cosseno em um momento preciso. Essa usa um número finito de sinais distintos para representar dados digitais. O PSK usa um número finito de fases, cada uma atribuída a um padrão único de dígitos binários. Normalmente, cada fase codifica um número igual de bits. Cada padrão de bits forma o símbolo que é representado pela fase particular (XIONG, 2000).

Um método conveniente para representar modulações PSK é o diagrama de constelação. A figura 2 apresenta um exemplo para a constelação PSK. Essa constelação mostra os pontos no plano complexo onde, nesse contexto, os eixos real e imaginário são denominados eixos em fase e em quadratura, respectivamente, devido à sua separação de 90 graus. Tal representação em eixos perpendiculares se presta a uma implementação direta. A amplitude de cada ponto ao longo do eixo em fase é usada para modular uma onda cossenoidal e a amplitude ao longo do eixo de quadratura para modular também uma onda cossenoidal. A fase modula o cosseno e a quadratura modula o seno (XIONG, 2000).

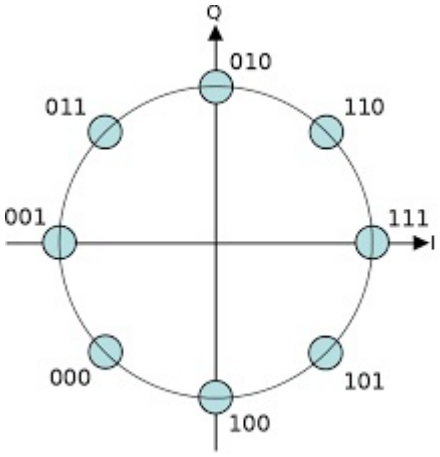

Figura 2 – Representação da constelação PSK.

Em PSK, os pontos de constelação escolhidos são geralmente posicionados com espaçamento angular uniforme em torno de um círculo. Isso proporciona máxima separação de fase entre pontos adjacentes e, portanto, melhor imunidade a ruído. Eles estão posicionados em um círculo para que todos possam ser transmitidos com a mesma energia. Desta

forma, os módulos dos números complexos que eles representam serão os mesmos e assim serão as amplitudes necessárias para as ondas cosseno e senoidal. Dois exemplos comuns são chaveamento de mudança de fase binária (BPSK) que usa duas fases, e chaveamento de mudança de fase de quadratura (QPSK) que usa quatro fases. Embora qualquer número de fases possa ser usado. Como os dados a serem transmitidos são geralmente binários, o esquema PSK é geralmente projetado com o número de pontos de constelação sendo uma potência de dois (XIONG, 2000).

Dentre as vantagens da modulação PSK está o fato de que esse sistema transporta dados sobre o sinal de RF de forma mais eficiente em comparação com outros tipos de modulação, como, por exemplo, ASK e FSK. Dentre as principais desvantagens da PSK está o fato de possuir uma menor eficiência de largura de banda. Ter os dados binários decodificados pela estimativa dos estados de fase do sinal, o que faz esses algoritmos de detecção e recuperação serem muito complexos. E os esquemas de modulação PSK de nível múltiplo, como QPSK, 16QAM são mais sensíveis às variações de fase. É também importante observar que o PSK oferece menor eficiência de largura de banda em comparação com o tipo de modulação ASK(XIONG, 2000). A figura 3 apresenta um exemplo de modulação PSK com uma entrada de bits e sinal modulado.

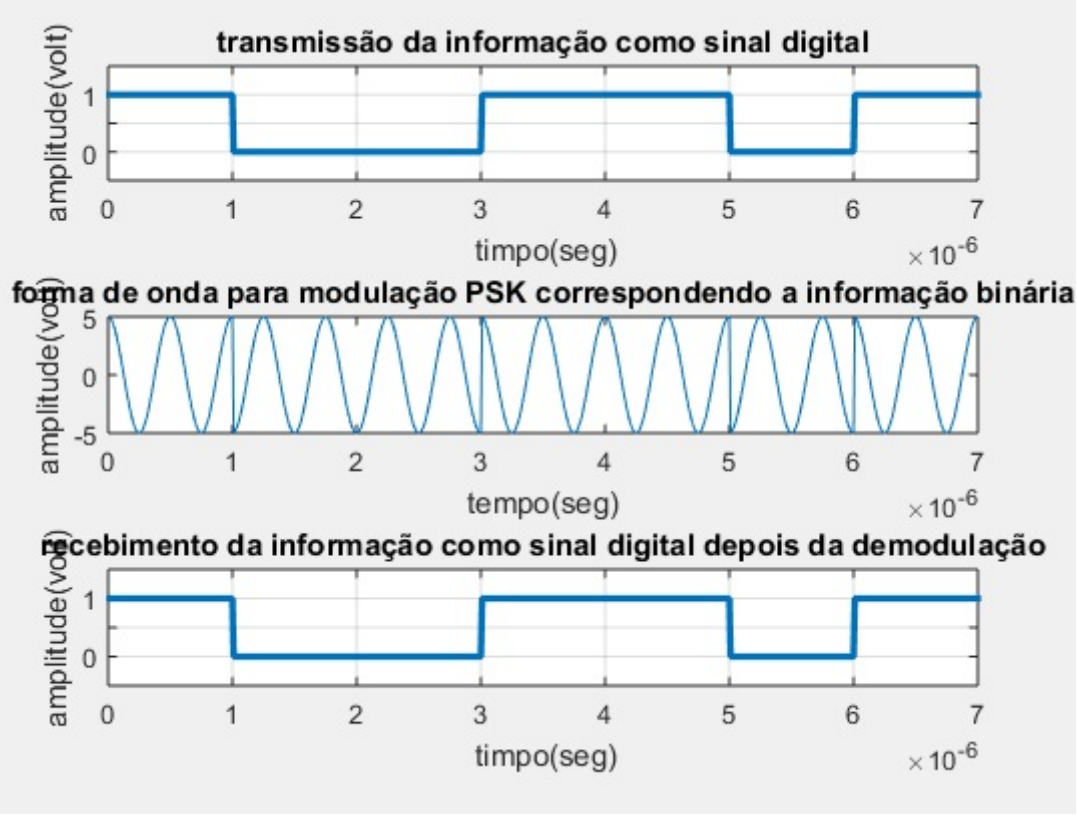

Figura 3 – Representação da modulação PSK.

2.1.4 Modulação ASK

A modulação ASK (*Amplitude Shift Keying*) consiste na modificação do nível de amplitude da onda portadora em função do sinal digital de entrada a ser transmitido. O sinal modulante assume um dos dois níveis discretos da fonte de informação (nível lógico 0 ou 1). Assim, ASK é uma técnica de modulação digital na qual a portadora é analógica e os dados a serem modulados são digitais. A amplitude do sinal da portadora é variada para representar os bits de entrada 1 e 0, enquanto a frequência e a fase do sinal da portadora permanecem constantes. Os níveis de tensão são deixados para os projetistas do sistema de modulação (XIONG, 2000).

Dentre as principais vantagens da modulação ASK, pode-se citar a alta eficiência de largura de banda. Um projeto simples de um receptor pode ser usado para transmitir dados digitais em fibra ótica. A modulação e a demodulação ASK são de custo relativamente baixo. Já entre as principais desvantagens pode-se citar que a ASK é muito suscetível a ruído, o que se deve ao fato de que ela afeta a amplitude(B.P.LATHI, 2010).

2.1.5 Modulação QAM

Wesolowsk (2011) caracteriza a modulação Quadratura em Amplitude, ou QAM, como uma técnica que altera tanto a amplitude como a fase do sinal modulante. Este pode ser expresso da seguinte maneira:

$$s_i(t) = A_i cos(2\pi f_c t + \theta_i), \tag{2.3}$$

para $i = 1, 2, 3, ..., M$, onde A_i é a amplitude e θ_i é a fase do i-ésimo sinal modulado.

Quando levado em consideração um modelo com um pulso de sinal, o sinal QAM será definido pela seguinte expressão:

$$s_i(t) = A_i p(t) cos(2\pi f_c t + \theta_i), \tag{2.4}$$

onde $p(t)$ é um pulso estreito definido em [0, T]. A expressão 2.4 pode ser expandida e reescrita como:

$$s_i(t) = A_{i1} p(t) cos(2\pi f_c t) - A_{i2} p(t) sin(2\pi f_c t), \tag{2.5}$$

$$A_{i1} = A_i cos\theta_i, \tag{2.6}$$

$$A_{i2} = A_i sen\theta_i \tag{2.7}$$

e

$$A_i = \sqrt{A_{i1}^2 + A_{i2}^2}. \tag{2.8}$$

O sinal QAM pode ser escrito como uma combinação linear de duas funções ortonormais. Assim, a expressão acima pode ser escrita como (KRZYSZTOF, 2011):

$$s_i(t) = s_{i1}\phi_1(t) + s_{i2}\phi_2(t), \tag{2.9}$$

onde

$$\phi_1(t) = \sqrt{\frac{2}{E_p}}p(t)cos(2\pi f_c t), \quad \phi_2(t) = \sqrt{\frac{2}{E_p}}p(t)sin(2\pi f_c t), \quad 0 \leq t \leq T \tag{2.10}$$

e

$$s_{i1} = \sqrt{\frac{E_p}{2}}A_{i1} = \sqrt{\frac{E_p}{2}}A_i cos\theta_i, \tag{2.11}$$

$$s_{i2} = \sqrt{\frac{E_p}{2}}A_{i2} = \sqrt{\frac{E_p}{2}}A_i sin\theta_i, \tag{2.12}$$

em que E_p é a energia de $p(t)$ em $[0, T]$.

Constelação QAM

O sinal QAM pode ser representado em uma forma geométrica denominada constelação. Dentre vários tipos de constelação, a que mais tem se destacado é a proposta por Campopiano e Glazer em 1962 (XIONG, 2000). Esta é uma representação do QAM no formato quadrado.

Para os sinais QAM no formato quadrado, as equações 2.5 e 2.9 podem ser escritas como

$$s_i(t) = I_i\sqrt{\frac{E_0}{E_p}}p(t)cos(2\pi f_c t) - Q_i\sqrt{\frac{E_0}{E_p}}p(t)sin(2\pi f_c t) \tag{2.13}$$

e

$$s_i(t) = I_i\sqrt{\frac{E_0}{2}}\phi_1(t) - Q_i\sqrt{\frac{E_0}{2}}\phi_2(t), \tag{2.14}$$

onde E_0 é a energia do sinal com menor amplitude e (I_i, Q_i) são um par de inteiros que determina o ponto do sinal QAM na constelação. Os valores (+/- 1, +/-1) são os mínimos valores de (I_i, Q_i). O par(I_i, Q_i) pode ser representado por uma matriz que descreve de maneira eficiente os valores de uma constelação QAM quadrada. A matriz 1 é representada como

$$1.\ [I_i, Q_i] = \begin{bmatrix} (-L+1, L-1) & (-L+3, L-1) & ... & (-L-1, L-1) \\ (-L+1, L-3) & (-L+3, L-3) & ... & (L-1, L-3) \\ ... & ... & ... & ... \\ (-L+1, -L+1) & (-L+3, -L+1) & ... & (L-1, -L+1) \end{bmatrix}$$

onde

$$L = \sqrt{M}, \quad M = 4^n, \quad n = 1, 2, 3... \tag{2.15}$$

e M denomina a quantidade de pontos de um constelação.

A figura 4 apresenta três exemplos de constelações QAM, sendo a 16 QAM um tipo de constelação QAM quadrada.

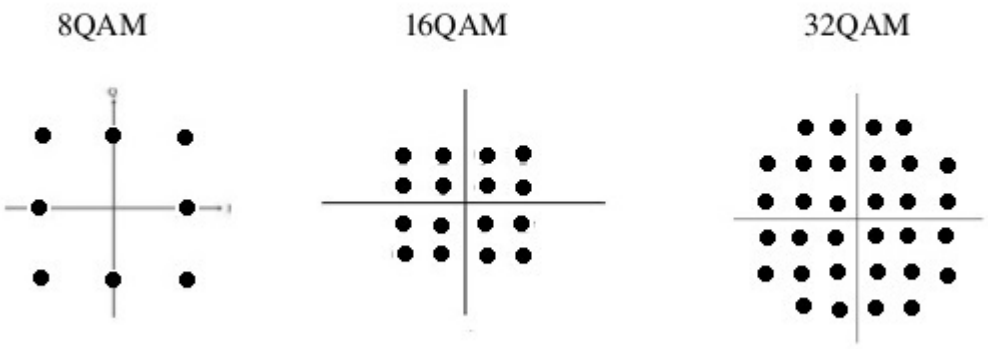

Figura 4 – Exemplos de constelações QAM.

Modulador QAM

Para um modulador digital QAM, os bits sequenciais de entrada que chegam ao transmissor são divididos em duas partes iguais, os quais são codificados separadamente. Esses dados são utilizados para controlar o gerador de nível que fornece o I e o Q do canal. Depois desse processo de separação de bits e devidamente mapeados em uma constelação, um dos sinais é multiplicado por um oscilador que descreve uma função cosseno enquanto o outro sinal é multiplicado por uma função seno. Este processo gera entre esses sinais uma diferença de fase de 90 graus. Depois desse processo, os sinais gerados são somados, formando assim o sinal QAM a ser enviado através do canal (B.P.LATHI, 2010).

2.2 Modulação multi-portadora

A *Multi-carrier modulation*, ou MCM, foi primeiramente usado na década de 50 em comunicações militares analógicas. O MCM é um método de transmissão de dados os quais são modulados paralelamente em sub portadoras. Outra característica importante é a divisão do espectro disponível em vários sub canais ortogonais (RIBEIRO, 2013).

Essa técnica apresenta algumas vantagens em relação ao método SCM (*Single Carrier Modulation*). Dentre elas pode-se citar a não necessidade de complexos equalizadores, tendo em vista que o mesmo divide o canal de modulação em sub canais o que gera uma resposta em frequência linear (BAHAI, 2002). Outra melhoria sobre os outros modelos de modulação está na questão do multi percurso. Esta é contornada com a utilização do intervalo de guarda o qual pode ser implementado de forma bastante simples (SIQUEIRA, 2004).

Devido a essas e outras características, como imunidade a ruídos, diferenças pequenas de complexidade em relação a outros sistemas, a MCM tem se mostrado bastante útil o estudo, análise e implementações desse modelo em sistemas de banda larga e de alta taxa de transmissão (SIQUEIRA, 2004).

O funcionamento básico de uma MCM está esquematizado na figura 5. Esse módulo capta os dados na entrada do transmissor MCM agrupando-os em M bits. Os M bits são divididos em N sub-blocos de m_n bits, onde $M = \sum_{n=0}^{n} n = m_n$, são usados para modular N sub-portadoras. Estas com uma banda igual e com frequência central $f_n = n\Delta f$, $n = 0, 1, ..., N-1$, espaçadas uma da outra, de forma que o espaço entre cada sub portadora não interfira no sinal de uma no da outra. Nos N sub canais, pode-se utilizar qualquer uma das modulações digitais, tais como ASK, FSK, PSK, mas geralmente usa-se a modulação QAM. Depois que os dados são coletados, divididos em blocos, mapeados e selecionados, faz-se a soma desses sinais e assim transmitidos, gerando o sinal MCM (SIQUEIRA, 2004).

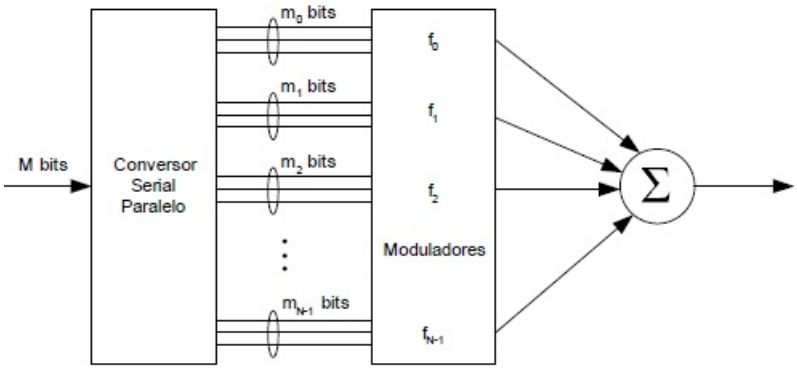

Figura 5 – Transmissor MCM simplificado. Adaptado de (SIQUEIRA, 2004).

2.3 Técnica de Modulação OFDM

O século XXI tem se caracterizado pelo surgimento de novas tecnologias, rápida troca de informações, acesso a informações pela internet, entre outras características. Esse novo padrão de vida é possível graças a pesquisas e estudos, principalmente, nas áreas da eletrônica e de telecomunicações (RIBEIRO, 2013).

Dentro da perspectiva de comunicações, observa-se o surgimento da transmissão de dados pela TV com modulação digital. Isso acentua a necessidade de um estudo cuidadoso a respeito da taxa de bits enviados e a largura de banda do canal (RIBEIRO, 2013).

Nesse quesito da transmissão de bits, velocidade de envio de dados e eficiência dos mesmos, fica evidente que um estudo mais minucioso de uma modulação adequada para certa aplicação se faz necessário. Pode-se notar que vários fatores influenciam no ambiente de comunicação. Por exemplo, no ambiente terrestre existem muitos fatores que fazem refletir as ondas eletromagnéticas criando um caminho de propagação de multipercurso o que causa alterações do sinal transmitido (B.P.LATHI, 2010).

Para resolver essas questões surgiram várias propostas, as quais possuíam suas vantagens e desvantagens. Por exemplo, uma solução no caso dos multicaminhos poderia ser a utilização de equalizadores. Porém, estes requerem altos recursos para implementação e possuem deficiência quando trabalham em altas taxas de envio de dados (ZOU.W.Y, 1995).

Dentro desse contexto, a modulação OFDM digital surge como uma técnica promissora, a qual não se faz necessária a utilização de complexos equalizadores e osciladores quando se utiliza a DFT (BAHAI, 2002).

A técnica OFDM é caracterizada por enviar dados de forma paralela, que faz uso de subportadoras ortogonais. Em cada subportadora é transmitida uma sequência de pulsos modulados digitalmente, os quais podem ser ASK, PSK, FSK, QAM, entre outros (BAHAI, 2002).

Para um envio de dados utilizando única portadora $M = 1$ para modular uma sequência de bits com duração de símbolos T_s o sinal gerado no transmissor $s_c(t)$ é dado pela seguinte expressão:

$$s_c = \Re(m(t)e^{j2\pi f_c t}), \tag{2.16}$$

onde f_c representa a frequência da portadora, $m(t)$ é a envoltória de $s(t)$ em relação a f_c e o termo $\Re$ significa a parte real do sinal $s(t)$ (R.CHANG, 1968).

O sinal $m(t)$ é a envoltória do sinal de transmissão $s_c(t)$ e é dado por

$$m(t) = \sum_{n=-\infty}^{\infty} d_n g(t - nT_s), \tag{2.17}$$

em que $d_n \ \epsilon(A_i + jB_i)$, os quais indicam a constelação correspondente das modulações AM-PM. A função $g(t)$ é o pulso que indica o sinal transmitido.

Para sistemas que enviam dados sequencialmente, denominados de única portadora, a largura de faixa é dada por

$$B_s = \frac{1}{2T_s}. \tag{2.18}$$

Para os sistemas de múltiplas portadoras, nesse caso o OFDM, a largura de faixa compreendida por cada uma das M-subportadoras é dada por

$$B_{sub} = \frac{B_s}{M}. \tag{2.19}$$

A duração dos símbolos T e a largura de faixa B_{ofdm} na transmissão de um sinal OFDM em banda base é dado por:

$$T = MT_s, \tag{2.20}$$

em que T representa o intervalo de duração de um símbolo OFDM, T_s é a duração de M - sub-símbolos de algum esquema de modulação AM-PM (SALTZBERB, 2004).

$$B_{ofdm} = MB_{sub} = B_s. \tag{2.21}$$

Com o resultado das equações acima, é possível ver que, tanto para o sistema OFDM, quanto para o sistema de única portadora, o valor de banda é o mesmo.

Apesar dos dois grupos possuírem o mesmo valor de banda ocupada, a técnica OFDM apresenta maior robustez contra erros em rajada de recepção e menor perda de símbolos durante o período de desvanecimento (BAHAI, 2002).

A técnica OFDM é uma técnica com base na modulação FDM (SALTZBERB, 2004). Esta última possui problemáticas em sua implementação, tais como: interferência cruzada entre os canais adjacentes; necessidade de filtros extremamente robustos, distorção de fase nas bordas dos filtros; entre outros.

Dentro dessa perspectiva, o sistema OFDM surge como uma alternativa de sistema mais eficiente. Pode-se destacar a vantagem desse sistema em apresentar a sobreposição espectral das M-subportadoras , as quais distam entre $\triangle f = \frac{1}{T}$. Isso possibilita uma maior eficiência no sistema, pois tem-se um melhor rendimento da largura de faixa das frequências transmitidas(SALTZBERB, 2004).

Como a relação de ortogonalidade se faz presente no sistema OFDM, pode-se haver sobreposição espectral das bandas sem que haja interferência entre os canais.

Pela condição da ortogonalidade, cada subportadora está localizada nos nulos espectrais das demais. Como o sinal transmitido deve ser um pulso retangular, o espectro de cada um dos sub canais devem ser apresentados no domínio da frequência como uma $sinc(f)$ centralizada na frequência da sub portadora correspondente ao canal. A cada $\frac{1}{T_k}$ (k é um inteiro) acontecem os zeros desses sinais no espectro. Dessa forma, optando-se por um afastamento de $\frac{1}{T}$ de uma sub portadora a outra, não haverá sobreposição do espectro das frequências de centro.

A figura 3 apresenta sinal OFDM no domínio do tempo. Essa apresenta a superposição de várias senóides de cada subportadora que compõe o espectro total.

No domínio do tempo, no interior do espaçamento de um símbolo OFDM, cada subportadora dispõe de um número inteiro de ciclos. Com a premissa de que a ortogonalidade é válida, destaca-se que duas subportadoras do sinal transmitido quaisquer OFDM diferem de um número inteiro de ciclos.

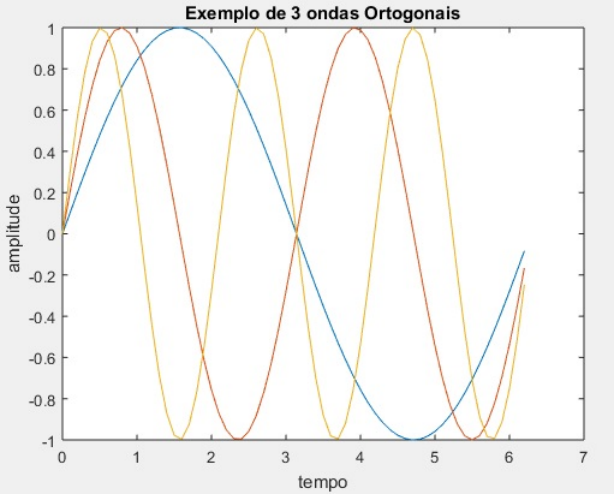

Figura 6 – Ortogonalidade das subportadoras no domínio do tempo.

2.4 Transmissão do sinal OFDM

Na transmissão de um sinal OFDM, os sinais OFDM gerados pelas M-subportadoras são representados da seguinte maneira:

$$s_(t) = \Re(\sum_{n=0}^{N-1} d_n e^{j2\pi f_n t}), \quad t\,\epsilon\,[0, T_s]. \tag{2.22}$$

Substituindo $f_n = f_0 + n\Delta f$, em que f_0 representa a frequência inicial do espectro e Δf o intervalo de frequência entre as subportadoras, temos:

$$s_(t) = \Re(e^{j2\pi f_0 t} \textstyle\sum_{n=0}^{N-1} d_n e^{j2\pi n \Delta f t}) \quad t\,\epsilon\,[0, T]. \tag{2.23}$$

Assim, o sinal de transmissão OFDM, simbolizado por $s_{ofdm}(t)$ em banda básica, é gerado pelas associações das sub portadoras e cada M sub-símbolo modula uma das subportadoras para o sistema OFDM .

$$s_{ofdm}(t) = (\sum_{n=0}^{N-1} d_n e^{j2\pi n\Delta ft}), \quad t\,\epsilon\,[0,T], \tag{2.24}$$

.

Em 1971, com o trabalho de Weinstein e Ebert, a implementação do sistema OFDM teve um grande avanço. A facilidade consistia na possibilidade do uso da transformada de Fourier para descrever a modulação e demodulação do sistema OFDM. A partir de então, essa nova metodologia permitiu evitar a implementação de sistemas OFDM com o uso de grande quantidade de osciladores (TACURI, 2014).

Com essa nova possibilidade de abordagem na transmissão do sistema OFDM, pode-se amostrar o sinal $s_{ofdm}(t)$ M vezes para uma taxa de $\frac{1}{T}$, o qual fica representado da seguinte maneira,

$$s_{ofdm}(k) = s_{ofdm}(kT) = s_{ofdm}(\frac{kT_s}{N}) = (\sum_{n=0}^{N-1} d_n e^{\frac{j2\pi nkT_s\Delta f}{N}}), \tag{2.25}$$

em que T_s é o tempo de símbolo, N o número de subportadoras e T o intervalo de símbolo da sequência de entrada. Substituindo na equação $\Delta f = \frac{1}{T_s}$, obtem a seguinte expressão,

$$s_{ofdm}(k) = (\sum_{n=0}^{N-1} d_n e^{\frac{j2\pi nk}{N}}), \tag{2.26}$$

Por meio de uma análise dessa equação, percebe-se que $s_{ofdm}(k)$ pode ser obtida aplicando o algoritmo da IDFT sobre a sequência d_n.

Como proposto acima, é possível implementar um sinal OFDM com o uso do ferramental IDFT. Sabe-se que o algoritmo que torna o cálculo da IDFT eficiente, rápido e utilizável, por componentes eletrônicos, é a IFFT; diante disso, fazendo uso dessa ferramenta, torna-se possível a implementação de um sinal OFDM bastante eficiente e realizável por descrição de *hardware* (BAHAI, 2002).

2.4.1 Transformada Discreta de Fourier

A transformada de Fourier em tempo discreto (DTFT) de um sinal $X[n]$ é dada por

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} \tag{2.27}$$

A convergência é garantida para sequências absolutamente somáveis, ou seja $\sum |x(n)| \leq \infty$ (OPPENHEIM, 2010).

Pela Transformada de tempo discreta de Fourier se obtem a DFT ,*Discrete Fourier Transform*. A DFT apresenta várias vantagens, entre elas, a de permitir calcular a TF em um computador digital, permitir determinar o conteúdo em frequência de um sinal, permitir desenvolver operações de filtragem no domínio da frequência, dentre várias outras possibilidades.

A DFT é definida como uma sequência discreta de duração finita obtida pela amostragem de um período da TF, em N pontos igualmente espaçados.

Observa-se que a DFT é realizada com 4N multiplicações de valores reais por ponto e que para calcular a transformada em N pontos são necessárias $4N^2$ multiplicações. Para obter-se uma maior eficiência computacional, uma das soluções criadas é algoritmo da *Fast Fourier Transform* (FFT). Esta é aconselhável para quando N for um número elevado e para frequências harmonicamente relacionadas (OPPENHEIM, 2010).

A FFT pode ser derivada da combinação de DFTs de $\frac{N}{2}$ pontos as quais formam uma DFT de N pontos e determinam a contagem de operação. Assim, separando a DFT em duas somas, sequências de índices pares e ímpares, obtêm-se o algoritmo da FFT. A Transformada rápida de Fourier é um algoritmo eficiente para calcular a Transformada discreta de Fourier (DFT) e a sua inversa (OPPENHEIM, 2010).

2.4.2 Transmissor OFDM

Um sinal OFDM, como visto anteriomente, é a soma de vários sinais, os quais são ortogonais entre si, com os dados em banda base sendo modulados por algum tipo de modulação digital, o que na maioria das vezes é realizado por um QAM (SIQUEIRA, 2004).

O primeiro grande bloco do transmissor OFDM é um conversor serial paralelo. Este tem a incumbência de converter o fluxo de dados que chegam em série para paralelo (SIQUEIRA, 2004).

No segundo processo é realizado um mapeamento de dados. Os dados paralelos advindos do conversor serial paralelo são mapeados através de uma modulação específica (SIQUEIRA, 2004).

Os transmisssores OFDM atuais são sintetizados levando-se em conta a simetria hermitiana. Esta permite que o transmissor OFDM seja implementado de forma mais eficiente, já que ela possibilita que seja calculada apenas metade dos valores para a IFFT (SIQUEIRA, 2004).

O bloco da IFFT recebe os dados gerados e realiza a modulação propriamente dita.

Este fluxo de dados é então convertido para serial novamente. Nesse processo é inserido o cyclic-prefix, o qual tem a função de replicar a parte final do bloco a ser transmitido no intervalo de guarda anterior ao bloco (SIQUEIRA, 2004).

Assim, um transmissor básico, que gera um sinal OFDM, possui os seguintes blocos: Um conversor serial-paralelo; um Buffer; um codificador; um bloco que implementa a IFFT ; às vezes um adicionador de prefixo cíclico e um conversor paralelo-serial. A figura 7 apresenta um exemplo de transmissor OFDM.

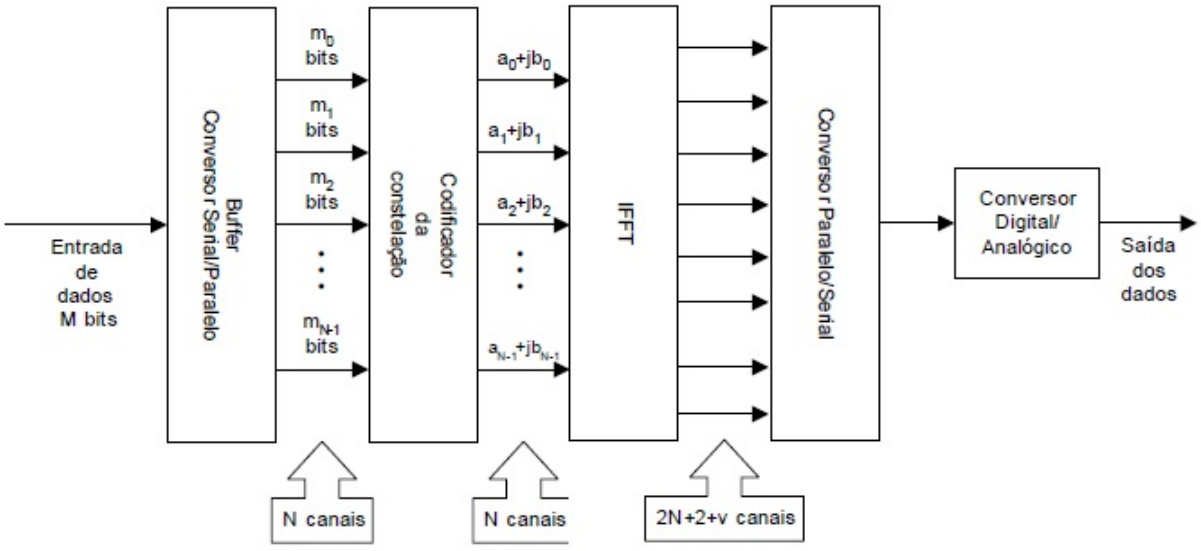

Figura 7 – Representação em bloco do transmissor OFDM. Adaptada de (SIQUEIRA, 2004).

3 Ferramentas básicas Usadas Para Projeto, Implementação e Avaliação

Este capítulo aborda a metodologia utilizada nesse projeto e as ferramentas necessárias utilizadas para completar o trabalho do transmissor e receptor OFDM. Assim, esta seção esclarece cada etapa do projeto e explica os *softwares* utilizados.

Os métodos de pesquisa utilizados na realização deste trabalho foram o qualitativo e o quantitativo. Por meio do método quantitativo foi possível mensurar e testar as hipóteses. Isso foi possível graças ao *software* de programação ISE 14.7 *Xilinx* desenvolvido pela *Xilinx* . E por meio do método qualitativo foi possível verificar a validade dos blocos sintetizados.

O presente trabalho teve como ponto de partida pesquisas bibliográficas, leituras e análises de artigos científicos, de livros e sites relacionados com o tema, coleta de dados, observações experimentais e manuais de alguns produtos como o *PMOD* D/A e o da *nexys3*. A partir dos conhecimentos adquiridos foram analisados conceitos como: modulação ortogonal por divisão de frequência; transformada de Fourier; transformada rápida de Fourier (FFT); linguagem de descrição de hardware (HDL); constelação das principais modulações digitais; entre outros conceitos. Dentre os sites de pesquisas, o *Scielo*, periódicos da CAPES, e o Google *Academic* foram os mais utilizados.

Os principais autores consultados e que contribuíram para o desenvolvimento deste trabalho, foram Haykin (2007), Tonny Mattos Siqueira (2004), Stalzberg (2002) e Wesolowski (2009). Como também, pesquisas de campo com professores da Universidade de Brasília.

O projeto foi dividido em quatro partes, e se inciou com a pesquisa dos conceitos fundamentais, prosseguindo para o processo de design, implementação, testes e análises. Cada uma dessas partes foi subdividida em seções menores que possibilitaram o melhor desenvolvimento e entendimento.

A primeira parte da pesquisa se deu com levantamento de informações e definições de conceitos necessários para a compreensão e desenvolvimento da modulação OFDM. Assim, foi necessária a observação e compreensão do conceito de dados em série e paralelo, mapeamento de constelação, transformada de Fourier , transformada discreta de Fourier, transformada rápida de Fourier, codificação VHDL. Esses conceitos são a base para a correta implementação do sistema OFDM.

Com os conhecimentos adquiridos, mencionados anteriormente, foi possível elaborar códigos que descrevem o sistema transmissor e receptor OFDM. Foram codificados

blocos que fazem parte da modulação OFDM. Cada um desses blocos possui um *Test-Bench* que foi utilizado para a validação da codificação, o qual foi usado para fazer uma comparação com bibliotecas já implementadas no Matlab e em blocos descritos em livros.

No decorrer do trabalho foram utilizados vários *softwares*, os quais são apresentados nessa seção.

Foram construídos dois grandes blocos, o primeiro com uma IFFT de oito entradas e outro com uma FFT de oito entradas para o receptor.

Os resultados obtidos através do *software* ISE 14.7 e analisados por meio das bibliografias e do *TestBench*, possibilitam uma comparação de eficiência, rapidez, robustez em relação ao processamento implementado em FPGA com microprocessadores e microcontroladores.

Após ter validados os blocos de forma separada, foi implementado um *TopLevel* que possibilitou a união dos mesmos. Em seguida foi realizada uma sequência de passos para que fosse possível enviar as codificações para *Nexys3* . Na última parte foi utilizado um conversor D/A, o qual envia o sinal advindo da FPGA para o osciloscópio.

A figura 8 é o fluxograma do transcorrer de todo o trabalho realizado. Como o projeto é dividido em etapas, sendo a primeira o levantamento da parte teórica, fica claro que é necessário todo um levantamento sobre como a transformada de Fourier funciona. Assim, a primeira etapa se baseia em compreender a FFT e a IFFT, a sintaxe da programação VHDL e os conceitos básicos da *Nexys3*. Fica esclarecido também que o funcionamento do algoritmo da FFT é de suma importância para a continuidade do trabalho, já que a segunda parte de elaboração está baseada no processo de *design*.

Quando a parte de estudo teórico foi terminada, o trabalho seguiu para o processo de *Synthesize* dos blocos na linguagem VHDL propriamente dita. Para este processo a codificação foi dividida em várias sub partes. Dentre elas tem-se a *syntaxe* e visão do esquemático RTL . A primeira parte utilizada para a verificação do código VHDL foi o uso do *software* da *Ise Xilinx* que disponibiliza o correto desenvolvimento do código a parte de verificação da sintaxe da codificação em VHDL na diretiva *syntax*. O processo de *Synthese* também possibilita a análise da hierarquia de projeto; através da diretiva *View* RTL, possibilita e reconhece a funcionalidade descrita e garante que seu projeto seja otimizado para a arquitetura do dispositivo selecionado.

Finalizada a parte de *Synthesis* no ISE, o processo de construção do bloco continuou na aba *Implement Design*. Essa parte é subdividida em três sub categorias, as quais são *Translate, Map e Place and Route*. Segundo o manual da *Xilinx*, no capítulo 2, essas subdivisões são para: primeiro mesclar as *netlists* e restrições de entrada em um arquivo de *design Xilinx*; segundo ajustar o desenho aos recursos disponíveis no dispositivo de destino e, opcionalmente, colocar o desenho; e terceiro colocar e direcionar o projeto para

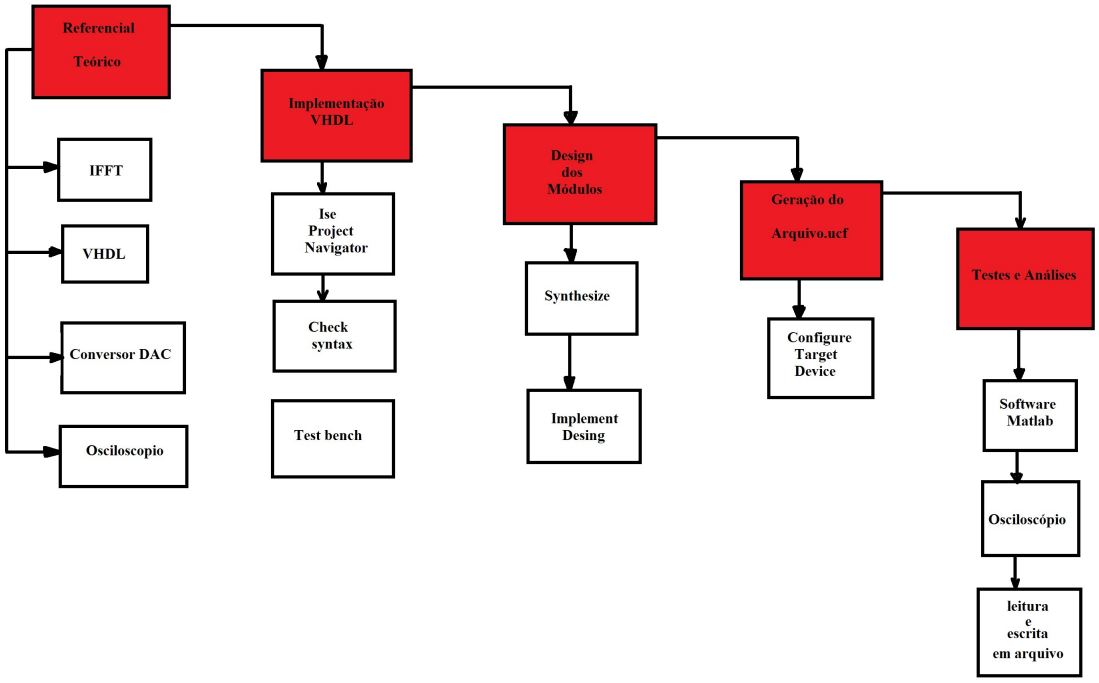

Figura 8 – Fluxograma das atividades realizadas durante a realização do trabalho.

as restrições de tempo.

Terminada a segunda parte, o processo foi encaminhado para a parte denominada *Generate Programing File*. Essa etapa se baseia em preparar suas informações de projeto para a configuração do dispositivo *Xilinx* e exibir os dados nos formatos de arquivo necessários para sua família de dispositivos de destino.

A quarta parte foi dedicada a verificação através do *test bench* de cada módulo em separado e o mapeamento dos pinos de entrada e saída do bloco maior do transmissor OFDM. Para o acionamento do *test bench* foi usado o sub programa denominado *Ise Simulator* no *Ise Xilinx*. Esse gera um arquivo em formato de onda quadrada.

Já a parte de mapeamento é feita pelo acionamento do *configure target* que será explicado em detalhes, posteriormente no decorrer deste capítulo.

3.1 Codificação VHDL

A linguagem VHDL é utilizada para descrever variados projetos em nível de *hardware*, assim como sistemas digitais em variadas camadas de complexidade. A sigla VHDL é um acrônimo para *Very High Speed Integrated Circuit*. A linguagem VHDL está basicamente dividida em duas grandes partes. A primeira é denominada Entidade, na qual descreve todos os sinais de entrada e saída do módulo. Já a segunda parte é nomeada de arquitetura. Essa descreve o funcionamento propriamente dito do circuito em termos de codificação. A codificação em VHDL do transmissor e do receptor OFDM é realizada no

capítulo quatro desse trabalho.

3.2 MatLab

O Matlab é um *software* de uso geral, utilizado para diversas tarefas em cálculos numéricos. Neste trabalho é utilizado como base de verificação para os cálculos elaborados através da codificação em VHDL da IFFT e dos demais blocos do transmissor e receptor OFDM. Assim, o Matlab tem como propósito final a comparação e validação dos blocos desenvolvidos em VHDL.

3.3 Arranjo de Portas Programáveis em Campo FPGA

Um dispositivo FPGA é um circuito integrado, o qual pode ser configurado muitas vezes através de linguagens denominadas de baixo nível como por exemplo a VHDL. FPGAs são dispositivos lógicos diferentes dos CPLDs e SPLDs pelo fato de os últimos apresentarem planos de AND e OR. Já a FPGA consiste em blocos lógicos que possibilitam a implementação de funções requiridas pelo usuário. Uma FPGA é dividida basicamente em três partes: blocos lógicos, blocos de entrada e saída denominados de I/O para conectar os pinos e as interconexões de fios e chaves.

Quando um circuito é implementado em uma FPGA, os blocos lógicos são programados para realizar as funções necessárias e traçam rotas para os canais e fazem a programação de interconexões entre os blocos lógicos. Cada codificação produzida no FPGA forma uma matriz bidimensional. Já as chaves de interconexão são interligações programáveis que permitem conectar blocos lógicos de forma eficiente. As funções lógicas na FPGA são implementadas através de blocos de memória denominadas LUT (Look-UP-Table).Esse blocos são capazes de armazenar um único valor lógico sendo zero ou um (GUTIERREZ, 2001) .

3.4 Conversor Digital Analógico

Um conversor digital analógico é um dispositivo eletrônico capaz de converter um sinal definido como digital para um sinal analógico. Para tais dispositivos é importante observar de forma mais cuidadosa três parâmetros principais: a resolução, a frequência máxima e a precisão.

Este trabalho utiliza o PMOD DA4, o qual é um conversor digital analógico que converte 12 bits em um sinal analógico, possui oito canais, com capacidade de oito saídas simultâneas, compatível com DSP de alta velocidade, capacidade de função de desligamento, baixo consumo de energia e conector PMOD de 6 pinos com interface SPI.

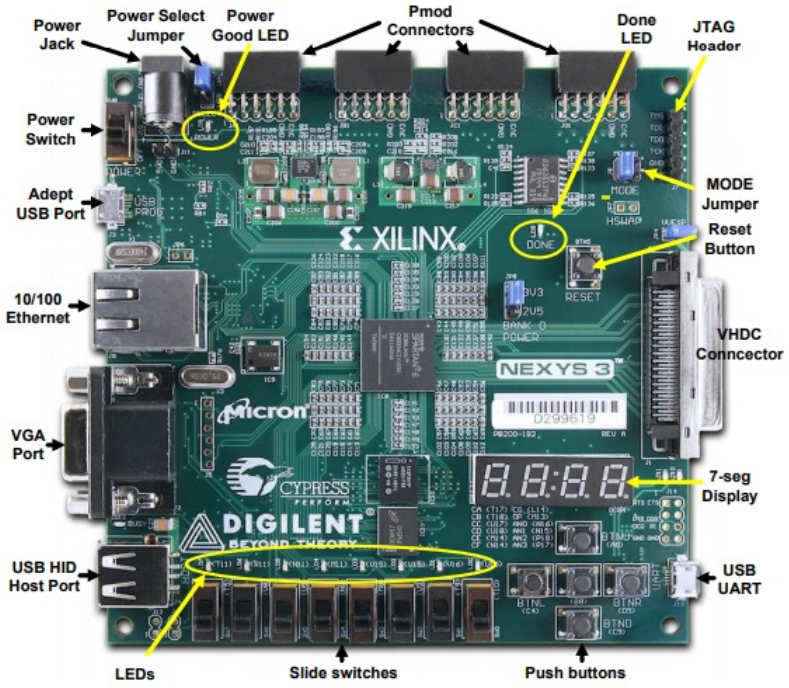

Figura 9 – Dispositivo lógico programável FPGA da Nexys3

O PMOD DA4 se comunica com a placa *host* através do protocolo SPI . Ao direcionar a linha *Chip Select* (CS) para um nível lógico baixo, os usuários podem enviar uma série de 32 bits de informação com os dados registrados no registrador apropriado na borda descendente do Serial *Clock* (SCLK). Uma vez que trigésimo segundo *bit* de informação tenha sido cronometrado, o comando que foi enviado no fluxo de dados é executado.

3.5 Simulação do Transmissor realizada no Matlab

Como o resultado obtido pelo osciloscópio não foi o esperado, houve a necessidade de uma análise mais detalhada dos dados. Assim, a última parte da elaboração do projeto foi a implementação do módulo transmissor e receptor no *software* Matlab. O *software* foi escolhido por ser ótimo na geração de gráficos e possuir funções já implementadas, dentre elas a função da transformada rápida de Fourier(FFT).

Para analisar de forma coerente e comparar os resultados obtidos com a implementação em VHDL, foi elaborado primeiramente um algoritmo que descreve de maneira eficiente os mesmos padrões adquiridos com a implementação do transmissor em VHDL. Para isso, a implementação foi realizada tendo como base o resultado adquirido no domínio do tempo no osciloscópio.

Com a simulação projetada no Matlab, pode-se gerar gráficos com os mesmos padrões de saída obtidos na geração do código em VHDL. Isso permitiu gerar a FFT

do sinal do tempo, assim como avaliar e comparar os resultados obtidos, tanto com as literaturas estudadas, quanto com o sinal adquirido no osciloscópio.

A partir dos resultados obtidos, passou-se a fazer uma série de propostas para obter os mesmos resultados previstos pela literatura. Assim, foram feitas pesquisas e entrevistas a professores e doutores na área, os quais sugeriram caminhos de análises e ideias, que possibilitassem obter os resultados previstos nas bibliografias.

4 Implementação

4.1 Introdução

Este capítulo aborda a implementação dos blocos do transmissor e receptor OFDM. Na literatura corrente, vários tipos de transmissores OFDM são apresentados e implementados em diferentes arquiteturas. Dentre essas, pode-se destacar a projetada com o chip DSP, que implementa a parte principal do bloco OFDM, que é a IFFT. No entanto, como discutido no referencial teórico, é mais vantajoso que esse bloco seja implementado em dispositivos FPGA pelo custo benefício, rapidez e eficiência do processo.

Assim, depois de um estudo aprofundado sobre os blocos constituintes do sistema OFDM em variados trabalhos, artigos e livros, verificou-se que seria adequado a construção dos blocos de Memória ROM, conversor Serial para Paralelo, Mapeador QAM, IFFT, Conversor Paralelo para Serial e uma implementação do conversor digital para analógico como blocos básicos para este trabalho. Esta configuração, para o sistema OFDM, é apresentada na maioria das literaturas atuais, artigos e teses.

4.2 Diagrama do Bloco transmissor Simplificado

A figura 10 apresenta a configuração dos blocos do transmissor OFDM simplificado. É importante observar que a codificação do bloco em VHDL foi realizada bloco por bloco. Essa configuração possibilita que o transmissor realize sua função de forma mais otimizada e facilita o encontro de erros do sistema.

Assim, o bloco transmissor foi inicializado por um conjunto de *bits* que foram armazenados em uma memória do tipo ROM. A cada término de processamento do transmissor, os *bits* da memória ROM são enviados a um bloco que faz a mudança dos *bits* de serial para paralelo. Essa parte se faz necessária, já que o bloco da IFFT faz a computação dos *bits* de forma paralela. Logo após os *bits* serem paralelizados, eles são enviados ao bloco de mapeamento, os quais são passados para os símbolos apropriados para representar os *bits* de dados. Após os símbolos serem mapeados, eles são enviados ao bloco de simetria hermitiana. Há necessidade desse bloco, já que são esperados resultados de números reais na saída da IFFT. Os símbolos então advindos do bloco de mapeamento QAM são transformados do domínio da frequência para o domínio do tempo utilizando a IFFT. Após o término da IFFT o resultado é serializado e enviado a um conversor digital para analógico o qual é analisado por um osciloscópio.

4.3 Implementação Memória ROM

A primeira parte da implementação foi realizada para armazenar bits. Assim, foi utilizada uma memória com 255 posições de vetores, como uma memória ROM, a qual tem a função apenas de ler os dados armazenados nesse vetor. Em cada vetor dessa memória se encontra 8 bits. A sequência escolhida para facilitar a validação e verificação do transmissor foi uma sequência de números começando com os valores de 0 e se estendendo até 255 no valor de inteiro. Como o bloco é muito semelhante a uma memória ROM não se pode alterar ou apagar os valores desses vetores, apenas acessá-los. Esse bloco foi desenvolvido em linguagem VHDL. O esquemático RTL é apresentado na figura 10.

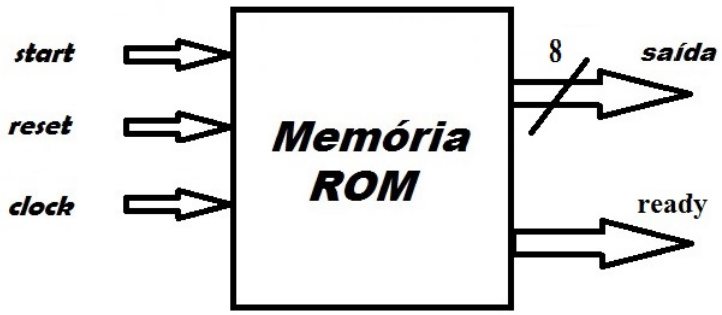

Figura 10 – Representação da entidade da memória ROM

Para a realização desse bloco, foi utilizada uma máquina de estados, que está dividida em três etapas. O primeiro estado é denominado de *esperando*. Neste, o estado espera um sinal de nível alto do *start* para ir para o segundo estado nomeado de *calculando*. Nessa segunda, a saída é atualizada. É importante observar que, para o projeto desse bloco, o estado calculando faz o incremento de um *bit* no endereço de memória. Quando esse valor chega no resultado 255 de memória, há reinício para a posição de memória zero. Essa etapa foi elaborada dessa maneira visando um *loop* infinito desses valores e uma maior possibilidade de análise no final do processo do transmissor OFDM. O último estado da ROM foi o estado de pronto. Nessa etapa há a passagem de nível lógico baixo para nível lógico alto, o que possibilita o acionamento do próximo bloco. A partir de cada atualização dos valores de saída desse bloco, esses dados serão enviados ao bloco serial paralelo. A figura 11 mostra a máquina de estados para o bloco de ROM para esse projeto.

A figura 12 mostra uma simulação em forma de onda para os valores armazenados em memória. Os dados armazenados são enviados a saída e só depois o nível de start é insertado para nível alto. Depois que o start é levado para um, os bits são enviados para a saída e o sinal de ready é insertado em nível alto. O sinal de ready é utilizado para validar o começo do bloco seguinte. Isso deve ser feito para que ocorra um sincronismo eficiente entre os blocos. Os dados vão sendo substituidos a cada acionamento do start.

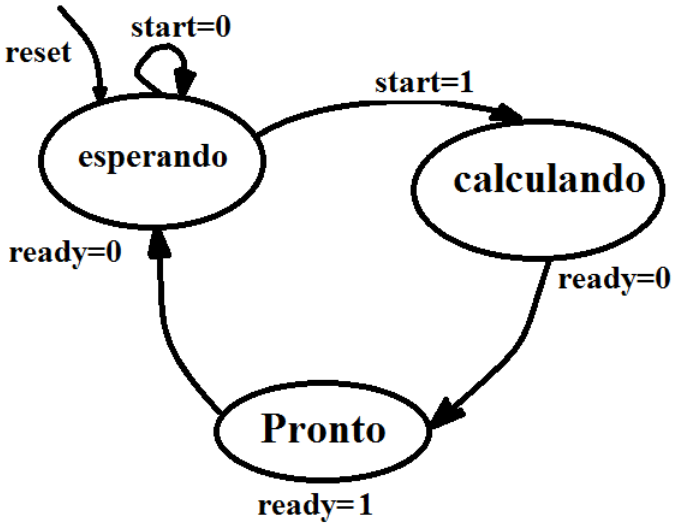

Figura 11 – Máquina de estados para o bloco de ROM.

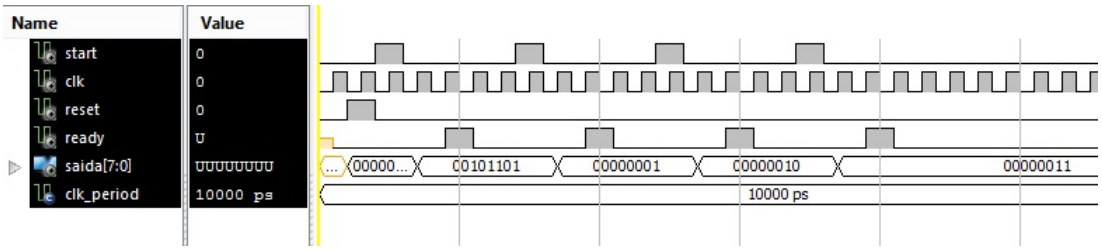

Figura 12 – Simulação do forma de onda para o bloco ROM.

4.4 Serial Paralelo

A segunda parte da implementação VHDL foi destinada a construção do bloco conversor Serial-Paralelo, o qual tem a função de receber os dados de forma serial e modificá-los para saídas em paralelo. Os dados são advindos da memória ROM, e divididos em blocos de M bits, separados em N canais, com m_n bits cada.

O algoritmo proposto considera que para o serial paralelo de uma entrada com oito *bits*, seja implementado a partir de uma máquina de estados, a qual tem como ponto de partida o acionamento em nível alto do *start*. Quando o sinal de *start* é insertado, a máquina de estados vai para o estado de calculando; já quando não há esse acionamento para nível alto, a máquina de estado permanece no estado esperando. Após passar pelo estado de calculando e todas as quatro saídas receberem os devidos *bits*, o fluxo da máquina segue para o estado de pronto. Neste estado é acionado em nível alto um sinal de *ready*. Esse bloco gera uma saída com 2 *bits*, os quais são enviados para os blocos de mapeamento QAM. A figura 13 mostra o esquemático da máquina de estados para esse bloco.

Pode-se perceber que o conversor serial para paralelo corresponde a uma operação meio que inversa do conversor paralelo para serial. Os dados são recebidos em forma serial

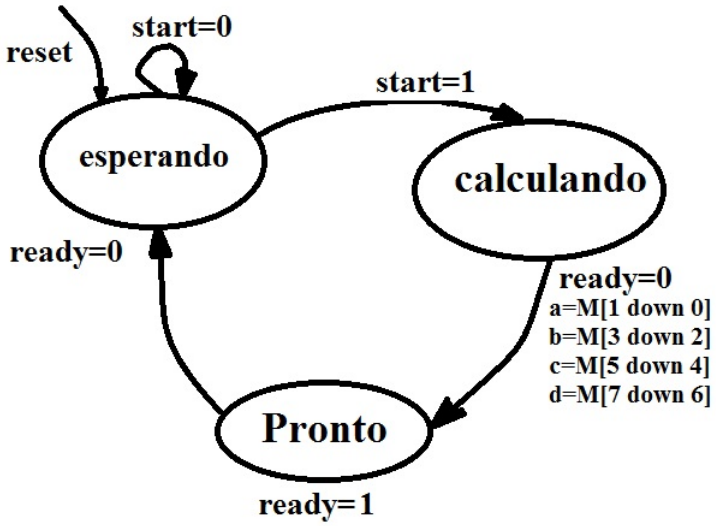

Figura 13 – Máquina de estados para o bloco serial para paralelo.

pela porta denominada M. Os dados de saída são enviados pelas portas nomeadas de 'a', 'b', 'c' e 'd'. Essa porta é acionada só depois que o sinal de *start* é acionado. Para facilitar o sincronismo entre os blocos SerialParalelo e o bloco de mapeamento, há um *ready* que é acionado em nível alto, possibilitando assim o início do próximo bloco. A figura 14 apresenta a entidade do bloco Serial para Paralelo.

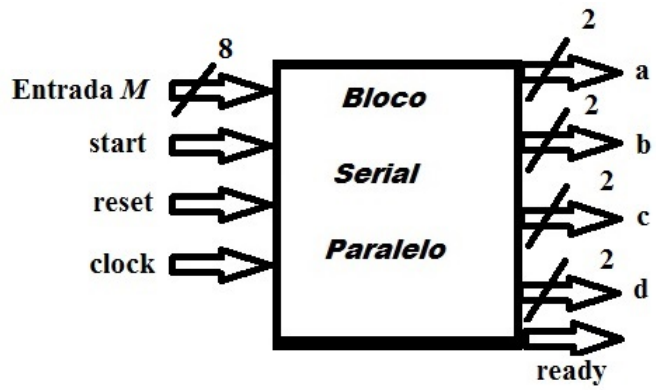

Figura 14 – Entidade do Bloco Serial-Paralelo.

A figura 15 apresenta uma simulação em forma de onda para o módulo serial para paralelo. Pode-se perceber que os *bits* entram através da entrada M de forma serial. Assim, a entrada M recebe os bits advindos do módulo de ROM e converte-os para paralelo quando um sinal em nível alto é adquirido pela entrada de *start*. O sinal de *ready* é usado para informar ao bloco seguinte que a saída está atualizada.

4.5 Mapeamento QAM

A terceira parte da implementação do sistema OFDM foi a codifi

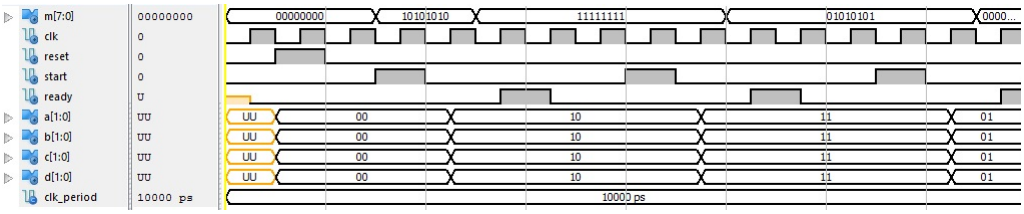

Figura 15 – Resultado do Testbench para o bloco serial paralelo.

-cação do bloco que mapeia os *bits* advindos do bloco Serial-Paralelo. O bloco mapeia os m_n bits do canal em um ponto da constelação $a_n + jb_n$ da modulação em questão.

O mapeamento de bits na modulação OFDM pode ser realizado de diversas maneiras. Pode-se utilizar para cada subportadora um tipo de constelação. Esse fato abre portas para se analisar o desempenho e robustez das constelações. Porém, essa análise foge do escopo deste trabalho. Assim, foi utilizada a constelação 4QAM em todas as subportadoras. Para se ter uma maior desempenho e eficiência do sistema a constelação 4QAM foi mapeada em código de *gray*. Esse tipo de mapeamento ajuda na robustez dos sistemas e na detecção de erros de *bits* do sistema.

O início da codificação do bloco mapeador de 4 bits foi a implementação da entidade em VHDL. A qual foi projetada para se ter uma entrada de *clock*, um sinal que inicializa o bloco denominado *start*, entradas de sinais advindos do bloco serial paralelo e sinais de saídas enviados para o bloco da simetria hermitiana. Esse bloco foi especi-fi

cado para ter duas saídas, as quais denotam o mapeamento correspondente na constelação 4QAM. A figura 16 apresenta o diagrama do bloco mapeador.

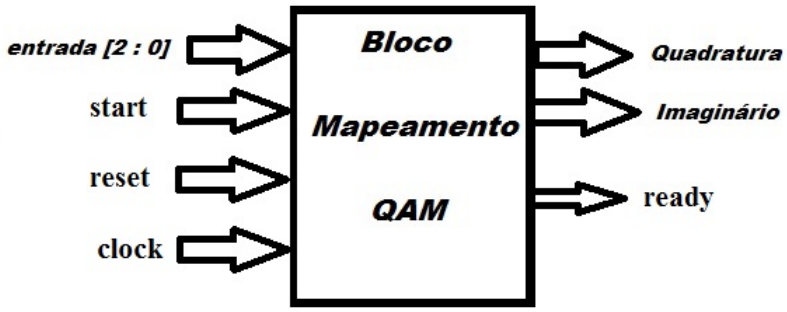

Figura 16 – Arquitetura VHDL mapeamento QAM.

O bloco implementado nessa parte do trabalho foi codificado através de máquinas de estado. O mesmo princípio de realização dos blocos de ROM e serial para paralelo foram usados para a implementação desse bloco. Assim, o bloco de mapeamento utilizou três estados: um para esperar o sinal de inicialização, o qual é insertado em nível lógico alto; o estado de calculando, onde se faz o mapeamento propriamente dito; e o terceiro estado, no qual um sinal de *ready* é levado ao nível lógico 1.

É importante observar que o bloco de mapeamento não realiza nenhuma modulação. A modulação, propriamente dita, é realizada pelo bloco correspondente a IFFT. Esse processo faz apenas o mapeamento dos *bits* e gera números complexos em sua saída.

Uma simulação em forma de onda para melhor entendimento é mostrada na figura 17. O resultado da simulação informa que há uma entrada com dois *bits* e duas saídas *Iout* e *Qout* que informa o valor do mapeamento. Assim, para uma entrada de bits 00 o valor de mapeamento é 10 para a saída Iout e 10 para a saída Qout. O mapeamento continua no sentido anti horário. A matriz 1 apresenta os valores correspondentes a esse mapeamento.

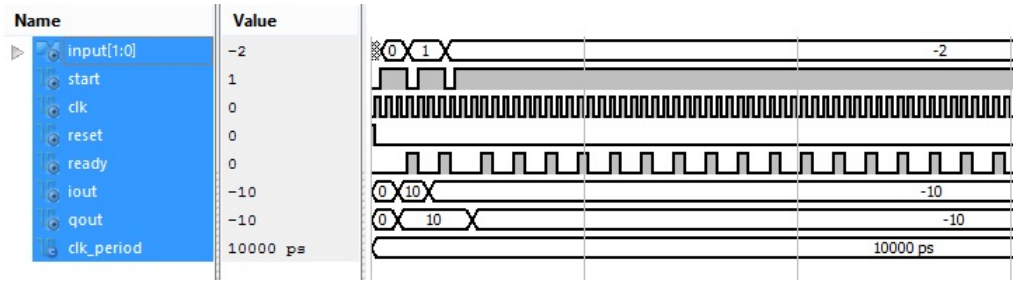

Figura 17 – Simulação em forma de onda para o bloco de mapeamento QAM.

4.6 Simetria hermitiana

Após mapeados os N canais, buscou-se realizar o módulo da simetria hermitiana. Como se está interessado em um sinal real na saída do transmissor OFDM, faz-se necessário que os valores que entrarem na IFFT satisfaçam a simetria hermitiana. Para um sinal de saída OFDM real é necessário que se use uma IFFT de $2N$ pontos, onde os valores de entrada X_k devem satisfazer a propriedade da simetria hermitiana.

A simetria hermitiana pode ser definida para um vetor $N \times 1$ dada por $X_{2N-k} = X_k$, onde $k = 1, 2, 3, ..., N-1$ e $ImX_0 = ImX_n = 0$.

Assim, fica claro que para obter-se os valores reais na saída do transmissor OFDM é necessário que a IFFT seja carregada com valores $2N$ pontos com a condição de que a segunda metade seja carregada com o complexo conjugado da primeira metade, e que as subportadoras 0 e N sejam valores reais.

Seguindo a teoria exposta, foi elaborada uma máquina de estados para fazer a correta relação de dados entre o bloco mapeador, a simetria hermitiana e o bloco da IFFT. A elaboração da máquina de estados foi projetada com quatro estados. O primeiro, denominado esperando, foi utilizado para se iniciar o processo. Esse estado espera um sinal em nível lógico alto para começar a computação dos valores. O segundo estado, denominado calculando, faz o mapeamento dos valores, os quais são multiplicados por -1

no terceiro estado. O quarto estado indica que o processo foi terminado e é insertado um valor alto para a variável *ready*. A máquina de estados pode ser visualizada na figura 18.

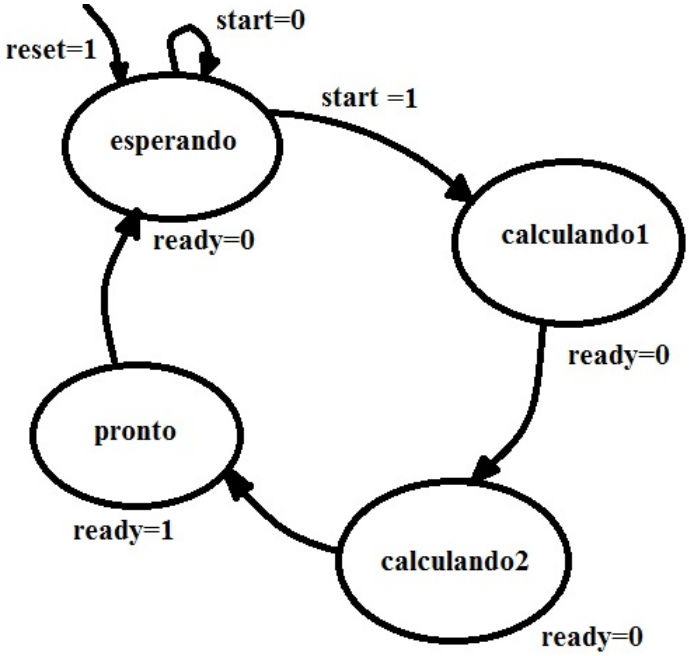

Figura 18 – Máquina de estados para simetria hermitiana.

Para se verificar o comportamento do bloco utilizou-se uma simulação em forma de onda. Essa é apresentada na figura 19. A simulação apresenta quatro entradas divididas em parte real e imaginária denominadas enR e enI com seus respectivos coeficientes indo de 1 a 4. As saídas são apresentadas por s_R para a parte real e s_I para a parte imaginária e também com seus devidos coeficientes indo de 0 a 7. Nota-se que o bloco só se inicializa depois que um sinal de start vai para nível lógico alto. As saídas são mapeadas de acordo com a teoria da simetria hermitiana.

4.7 Inversa da Transformada de Fourier

A quarta parte da implementação dos sistemas da modulação OFDM foi a codificação do bloco IFFT. Para construção desse bloco é de suma importância a compreensão do algoritmo da IFFT. E para melhor entendimento do mesmo, foi elaborado um estudo detalhado na seção da fundamentação teórica. A implementação dos blocos foi baseada no algoritmo *Radix 2*. Como se busca um sinal OFDM real na saída do transmissor no domínio do tempo para as N subportadoras, o bloco da IFFT deve ser composto por 2N pontos, lembrando que a segunda metade dos pontos deve ser o conjugado dos valores da primeira metade.

Para o algoritmo de 8 pontos da IFFT foi realizado um código VHDL, o qual é descrito a seguir passo a passo.

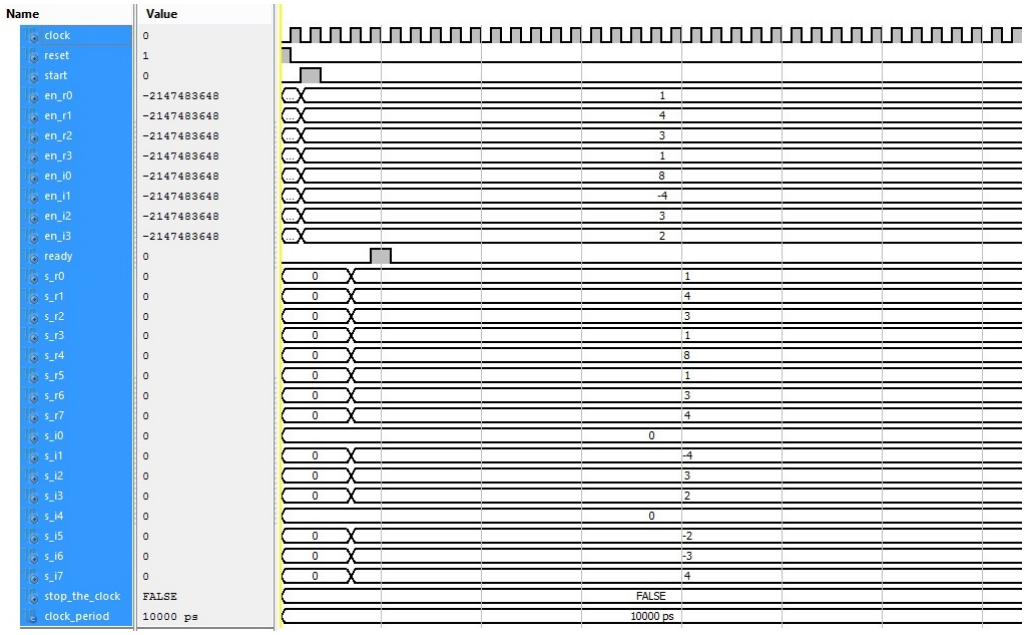

Figura 19 – Simulação em forma de onda para simetria hermitiana.

Ao ser estudado o algoritmo da IFFT, observou-se que a codificação poderia ser implementada através de nove estados. Cada estado faz uma parte do algoritmo da IFFT. O acionamento do *reset* leva o sistema ao estado dedicado ao zeramento das variáveis do sistema. O próximo passo é realizar o primeiro conjunto de somas e subtrações da decimação da IFFT de oito pontos. Nesse estado são realizadas as operações de somas e subtração dos inteiros alocados a entrada da IFFT advindos do bloco de simetria hermitiana. É importante ressaltar que a IFFT possui o dobro de entradas, pois são as entradas da parte real e imaginária dos números imaginários. O resultado das somas do *state0* é então lançada a um vetor de sinal correspondente. O segundo, terceiro e quarto estados foram implementados para fazerem a multiplicação por $sen45graus$. A IFFT implementada faz uma aproximação para esse valor. Na elaboração do código e visando um rápido processamento de algoritmo, decidiu-se aproximar por 0.707. Essa aproximação é realizada nos estados *state1* e *state2*. Nos estados *state1* e *state2* o valor é multiplicado por 707 e no estado *state3* o valor é dividido por 1000. No estado *state4* são realizadas as somas correspondentes ao algoritmo da IFFT. Como para continuar há a necessidade de alguns números serem multiplicados por j, o estado *state5* foi elaborado para essa proposta. Sendo prosseguido pelo *state6*, o qual tem a funcionalidade de realizar as somas do último estágio da IFFT. Por fim, os valores são divididos por um fator 8. O último estado faz a saída *ready* do bloco da IFFT ser levado a nível alto. A figura 20 demonstra a máquina de estados e a entidade para a IFFT desse projeto e a figura 21 apresenta o passo a passo para a realização do bloco em VHDL.

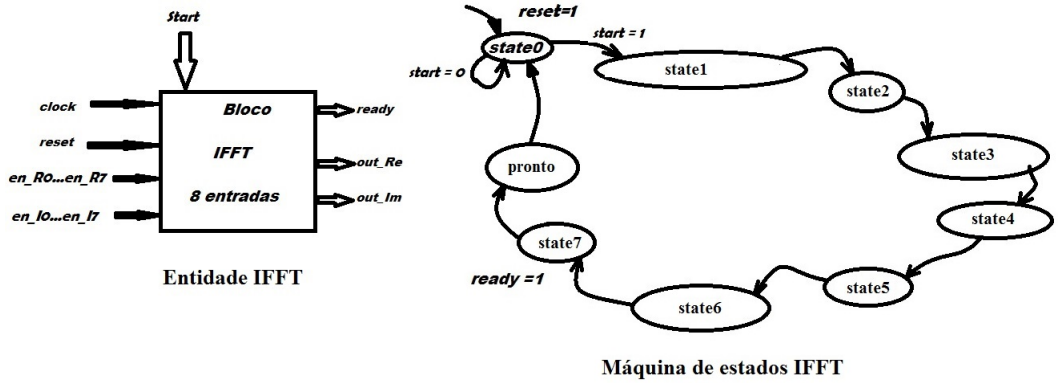

Figura 20 – Máquina de estados e entidade da IFFT.

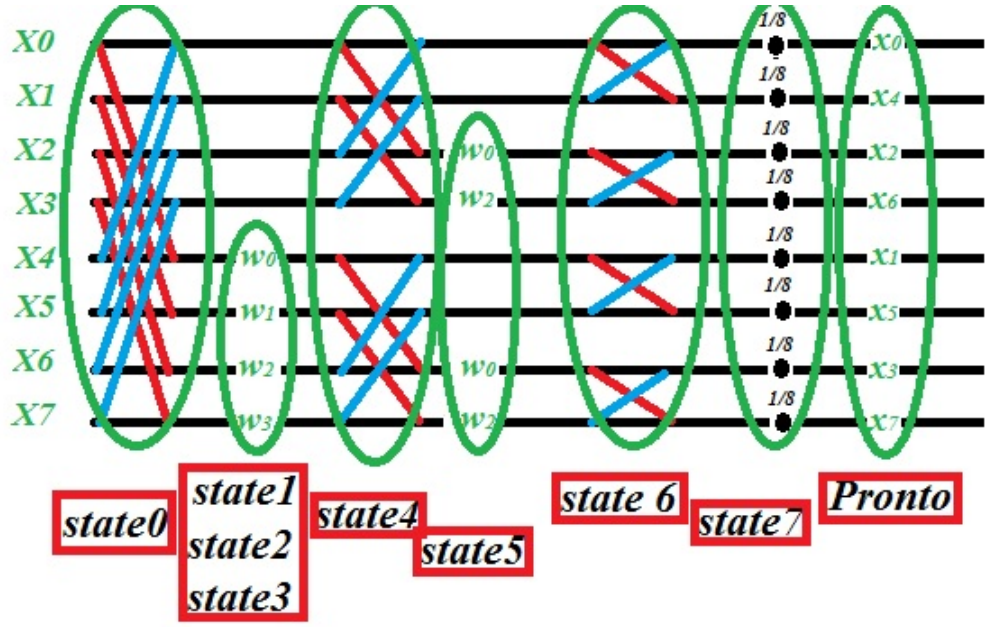

Figura 21 – Diagrama para Transformada Inversa de Fourier de oito pontos usado para codificação em VHDL.

4.8 Módulo Paralelo Serial

Para se obter os valores adquiridos no osciloscópio, foi necessário implementar um bloco paralelo serial. Este possui um comportamento muito semelhante a um multiplexador. O bloco em questão possui oito entradas de valores que são amostrados um de cada vez na saída do bloco. Essa amostragem de valores é selecionada através um acionamento em nível lógico alto. Esse sinal foi denominado de *start*. É importante salientar que esse bloco foi codificado depois da codificação do bloco Conversor D/A. Isso se deve ao fato que, para ser acionado de modo eficiente e usar as corretas variáveis, é necessário conhecer as entradas do bloco e as saídas que serviriam de estímulos para o *PMOD*. Dessa maneira foram utilizadas quatro entradas denominadas de *start*, wrt_done, *reset* e *clock*. O sinal de *start* está interligado aos blocos que advém anteriores ao bloco paralelo serial. Assim, o

início do bloco paralelo serial fica vinculado ao *ready* do bloco anterior a ele. Já o wrt_done é um sinal que advém do bloco conversor D/A. Esse sinal significa que o bloco DAC finalizou sua passagem de digital para analógico e está pronto para a próxima conversão de dados. O sinal advindo do bloco D/A e o start advindo dos blocos anteriores ao Par2Ser fazem com que o bloco tenha um sincronismo eficiente. A figura 22 mostra a entidade desse bloco.

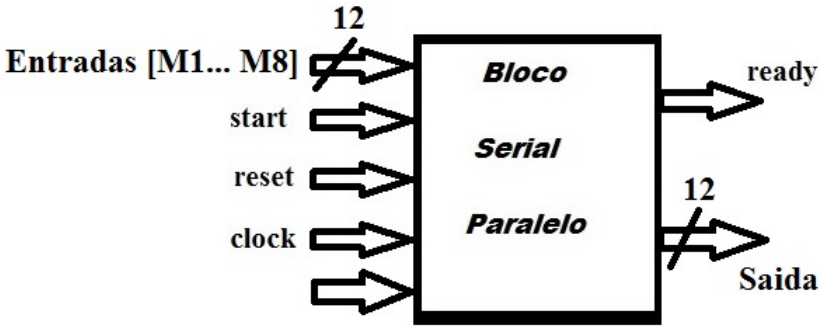

Figura 22 – Entidade do bloco Paralelo para Serial.

A codificação desse bloco se deu a partir de oito máquinas de estados. A cada máquina de estados é lançada uma saída correspondente a um valor de saída do bloco da IFFT. O sincronismo, como já mencionado, é realizado inicialmente pelo *start* , o qual é acionado pelo bloco da IFFT. Após o funcionamento do bloco, o sincronismo se dá pela realização do sinal advindo do bloco do conversor D/A. Dessa maneira, a cada subida de nível do wrt_done há uma mudança de estados. Dentro de cada estado é realizado um pulso para que seja enviado ao bloco conversor para que ele possa começar novamente a conversão de dados. Dessa maneira, observa-se que o Par2Ser funciona como um multiplexador que depende de dois sinais de entrada para iniciar o bloco e possui como saída dados para a conversão e um *ready* para acionar o funcionamento do PMOD. A figura 23 apresenta essa etapa.

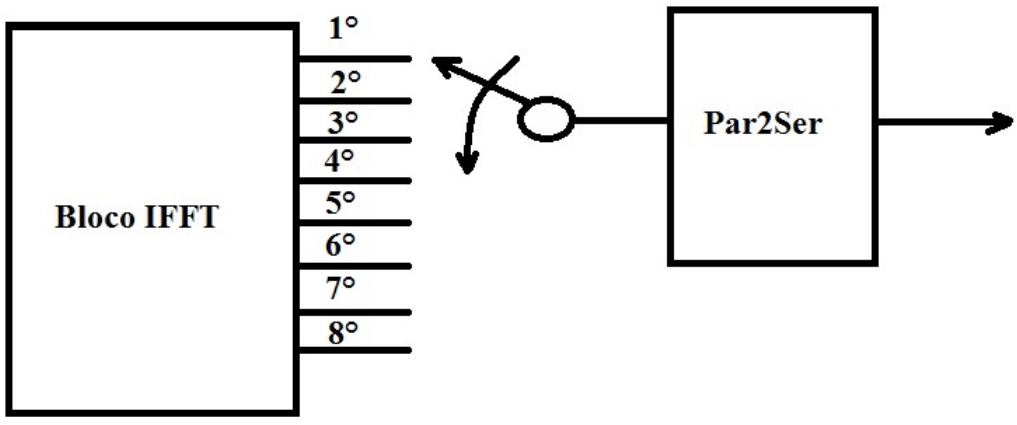

Figura 23 – Funcionamento do bloco Paralelo para Serial

4.9 Conversor D/A

Para a correta elaboração do código do conversor D/A, foi necessário compreender o seu funcionamento. Para isso, foi utilizado como meio de conhecimento e compreensão o datasheet do conversor. Este é disponibilizado pelo site do fabricante. O conversor utilizado foi o PMOD DA4 disponibilizado pela *Xinlinx*. Ele tem uma capacidade de oito canais de conversão de 12 bits de digital para analógico.

O DAC possui seis pinos, dentre os quais cada um possui uma função específica. O primeiro deles é *SYNC*. O *SYNC* tem a função de sincronizar os dados. Este pino é acionado quando há um sinal de nível logico baixo em sua entrada. Quando ele é acionado, ativa o registrador de deslocamento de entrada. Com isso, os dados são transferidos, nas bordas decrescentes dos próximos 32 bits. Uma observação importante é que se o sinal de *SYNC* for elevado a nível lógico alto antes dos 32 ciclos de *clock*, o *SYNC* funcionará como um interruptor de fluxo de dados e os dados não serão armazenados de maneira eficiente no registrador.

O segundo pino é dedicado a entrada de dados. Esse pino tem uma capacidade de entrada de 12 bits. As outras entradas são dedicadas a alimentação do dispositivo, ao terra do sistema e ao acionamento do relógio da placa. Há ainda um outro pino, porém não se faz uso dessa pinagem para o uso do DAC.

O funcionamento do PMOD se inicia acionando a linha SYNC em nível lógico baixo. A seguir é necessária a entrada de 32 bits no registrador de deslocamento de entrada. Os quatros primeiros bits são desprezados. Seguidos por mais quatro bits que acionam o funcionamento do comando do DAC. Como o DAC faz uso de oito canais, os próximos quatro bits fazem a escolha apropriada do canal que se queira trabalhar. Para este trabalho foi utilizado o primeiro canal denominado pelo datasheet de A.

A figura 24 apresenta a entidade para o bloco PMOD descrito em VHDL.

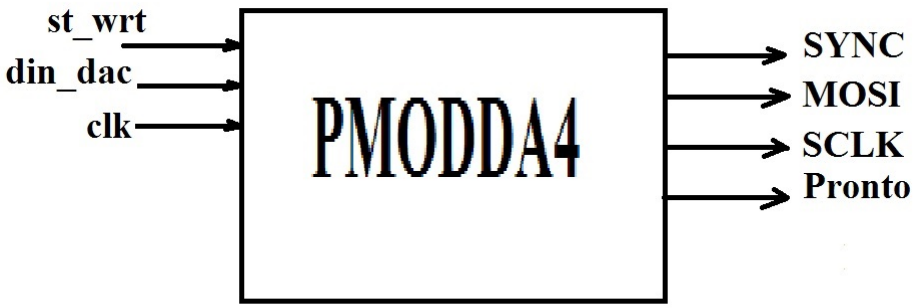

Figura 24 – Entidade para o PMODDA4 descrito em VHDL

A elaboração da codificação desse bloco ficou dividida em cinco máquinas de es-

tados. A primeira funciona como um estado de espera. Nesse estado o DAC permanece no estado inerte até que o módulo PAR2SER envie um nível lógico alto para a entrada do DAC pela entrada st_wrt. Os próximos três estados fazem a inicialização do PMOD e a escrita dos dados de entrada no registrador específico. O quinto estado entra em um aguardo até que um nível st_wrt seja acionado novamente.

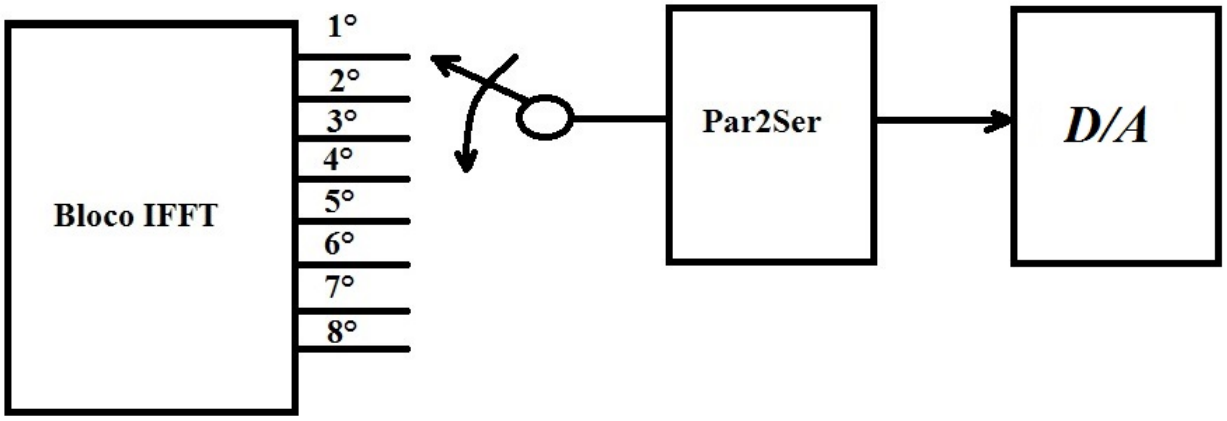

Figura 25 – Representação do sincronismo dos blocos paralelo serial com o bloco do conversor D/A.

4.10 Passos para a codificação do Transmissor em VHDL

A última parte da implementação foi a realização do top levels. Para essa parte, foram utilizados sinais para intercomunicar os blocos. As figuras a seguir mostram o esquemático RTL proposto para o módulo do transmissor OFDM.

A partir dos top levels realizados, passou-se a implentação da comunicação dos módulos OFDM com o conversor D/A. Este é utilizado para converter uma grandeza digital, dados binários, em uma grandeza analógica.

Nesta etapa o primeiro passo foi selecionar no Ise 14.7, na aba view, a configuração Implementation. O Segundo foi clicar duas vezes na diretiva *synthesize*. Esta gerou o esquemático RTL e analisou a sintaxe do programa. No terceiro passo, foi acionado o comando Implement Design. Este comando realizou o mapeamento do esquemático RTL na placa Nexys3. A seguir está o UCF usado para fazer o mapeamento na *nexys 3*.

O quarto passo é dividido em sub-passos, os quais são: primeiro, selecionar o Generate Programing; em seguida selecionar o *Configure Target Device*, o qual possibilita a visualização da tela *ISE Impact*. Na tela que surge, seleciona-se o *boundary scan*. Por meio dessa configuração, foi possível fazer a conexão entre a *nexys* 3 e o computador através do comando *output Cable auto conect*. Após a etapa descrita acima, foi possível selecionar o comando *initialize chain*, o que possibilitou selecionar um arquivo bin. Finalizando este processo, seleciona-se o comando Program para enviar as informações para Nexys3.

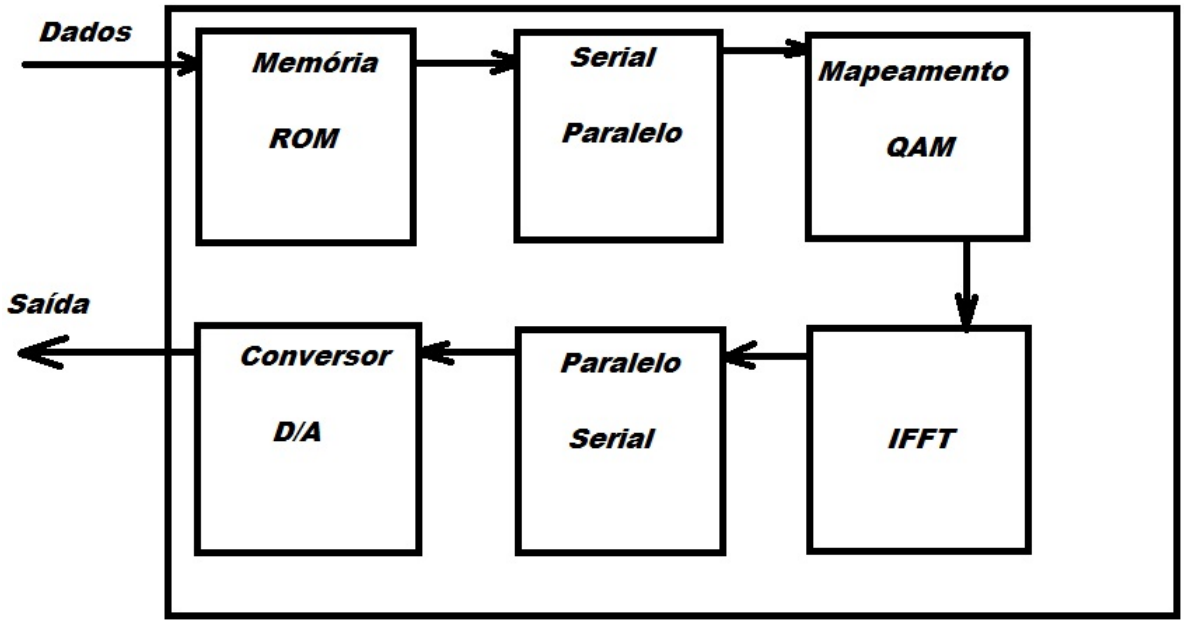

Figura 26 – RTL para o transmissor OFDM

```
1
2  ##Clock signal
3  Net "clk" LOC=V10 | IOSTANDARD=LVCMOS33;
4  #Net "clk" TNM_NET = sys_clk_pin;
5  #TIMESPEC TS_sys_clk_pin = PERIOD sys_clk_pin 100000 kHz;
6
7
8  ## Switches
9  Net start LOC = T10;
0  Net reset LOC = T9;
1
2  ##JB
3  Net "CS" LOC = K2;
4  Net "MOSI" LOC = K1;
5  #Net "JB<2>" LOC = L
6  Net "SCLK" LOC = L3;
7
```

Figura 27 – Mapeamento UCF

4.11 Simulação Matlab OFDM

A última parte do projeto foi realizada no software Matlab. Essa parte foi desenvolvida tendo como base o resultado adquirido no osciloscópio no domínio do tempo. Assim, a primeira fase da implementação do código de simulação foi desenvolver um algoritmo que repetisse valores adquiridos pela IFFT do bloco transmissor. Dessa maneira, valores obtidos são repetidos por um tamanho fixo.

A partir desses pontos, há a união com oito segmentos de reta, os quais foram denominados de T1 na imagem 28. Para se completar essa etapa, há ainda no gráfico um tempo T2 , o qual se faz necessário, já que este corresponde a um intervalo de guarda que também está implementado no código VHDL do transmissor OFDM.

A partir da implementação acima, passou-se a implementar de fato os outros blocos do transmissor OFDM no Matlab. A primeira etapa a ser definida foi uma sequência de números inteiros, os quais são transformados em bits e logo depois enviados ao bloco de mapeamento QAM. Os bits alocados ao bloco de mapeamento são mapeados na constelação 4QAM, da mesma maneira que no bloco em VHDL. É importante notar que há um fator de multiplicativo de 100 nesse bloco.

A próxima estrutura codificada foi a da simetria hermitiana. Essa elaborada para se obter quatro valores de entrada e oito de saída. Como já mencionado na parte da fundamentação teórica, este bloco deve receber os quatro números complexos advindos do bloco mapeador e transformá-los em números complexos conjugados, para assim obter oito valores de saída e encaminhá-los a uma transformação da IFFT.

O *software* Matlab possui um bloco já implementado da função IFFT. Assim, foi utilizado a função IFFT (x,8) para realização da implementação do transmissor. Com os dados adquiridos, foi elaborado um gráfico para verificação e análise. Esse gráfico foi apresentado na imagem 24. Neste gráfico o tempo T2 representa os oito primeiros números obtidos da IFFT e o tempo T1 representa o intervalo de guarda do bloco OFDM.

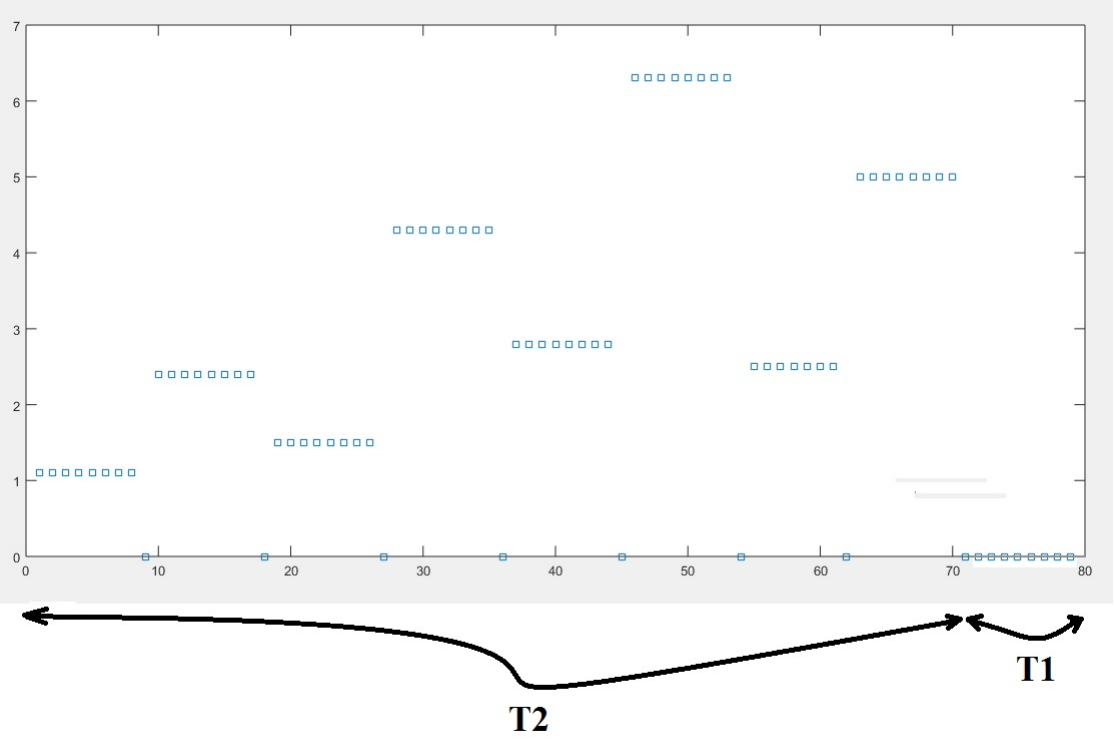

Figura 28 – Implementação do transmissor OFDM antes do filtro passa baixas.

Após a implementação do transmissor OFDM no Matlab, foi utilizado como base o sinal no domínio do tempo apresentado no osciloscópio para a continuação da simulação do transmissor e receptor OFDM. Nessa parte, a codificação ficou primeira determinada da maneira como o osciloscópio realiza a FFT. Assim, os valores da IFFT foram coletados

e usou-se todos os valores obtidos para se calcular uma FFT. Dessa maneira , foi utilizada uma configuração no Matlab com a função *length*, a qual descreve o tamanho da variável. Dessa maneira, utilizou-se todos os valores de pontos adquiridos pela IFFT e fez-se uma FFT. Essa parte da implementação foi realizada pelo fato de ser dessa maneira a implementação utilizando o osciloscópio e o módulo transmissor realizado em VHDL.

5 Resultados e Análises

Este capítulo apresentará as análises e os resultados obtidos na implementação dos blocos realizados no terceiro capítulo. Para uma melhor organização e comparação, os dados são tabelados.

Para que a placa Nexys 3 possa receber os dados enviados pelo computador, é necessário que haja uma configuração precedente. Essa é realizada utilizando a diretiva *Implementation Constraints File* ". Ao se configurar essa opção, pode-se ordenar cada pino de entrada e saída da placa. Ao se mapear os pinos, faz-se necessária a conexão entre a placa e o computador. Essa conexão é realizada por um cabo USB. Após as etapas "Synthesize", "Implement Design"e *Generate Programing* serem concluídas, o *configure Target Device* é acionado. Por conseguinte, o programa *ISE IMPACT* é aberto. Dessa maneira a conexão entre a NEXYS e o computador é realizada. Com a utilização desse programa e com a geração de um arquivo .bit, o *ISE IMPACT* faz o envio entre as saídas da nexys e o conversor digital analógico.

5.1 Realização dos testes

Na realização do trabalho em questão, muitos blocos foram realizados. Para a correta validação desses blocos, uma série de testes foram realizadas. Os dois módulos mais importantes para as análises foram o DAC e o módulo da transformada rápida de Fourier. Para o bloco da IFFT foi realizado um *Test Bench*. A partir desses valores, fez-se comparações entre os valores obtidos no MatLab através de uma função já implementada denominada ifft(y,x), em que y são os valores de um vetor e x a quantidade de valores desse vetor. Já para o conversor digital analógico, fez-se o *Test Bench*, o qual foi utilizado para comparar o sinal de saída no osciloscópio e no *datasheet* do fabricante.

Após os testes da IFFT e do conversor DAC, o próximo bloco verificado foi o transmissor. Esse bloco faz a união de todos os sub blocos implementados. A partir da conexão desses, foi possível verificar a forma de onda de saída e seu respectivo sinal no domínio da frequência.

5.2 Resultado IFFT

Como já mencionado, foi realizado um *Test Bench* para a IFFT. Esses valores são apresentados na Tabela 1. Esta apresenta os valores de XR0 a XR7 como entrada real e XI0 a XI7 como entradas imaginárias. Os valores de saída do módulo, codificado

em VHDL, e os valores correspondentes a saída do Matlab são também apresentados na tabela. Os valores são apresentados em inteiro. É importante observar que o bloco foi implementado para um tamanho de 16 bits. Isso permite que os valores estejam entre -32.768 e 32.767.

Ao observar os valores apresentados na Tabela 1, é possível concluir que há uma certa diferença entre os valores obtidos pelo Matlab e o módulo VHDL. Abaixo são apresentadas algumas justificativas do porquê há essa diferença de valores.

5.2.1 Falta de acurácia

A grande maioria dos blocos implementados em *hardwares* apresentam uma discrepância de valores quando comparados com softwares mais bem robustos. Isso é o que se denomina problema na acurácia do sistema. Para o bloco da IFFT em questão há um problema com a sua acurácia quando comparado com a implementação em MatLab. Um dos motivos que leva a essa não igualdade de valores se deve ao fato de que em *hardware*, na maioria das vezes, se trabalha com o formato de ponto fixo. Já na implementação no software Matlab se faz a computação das etapas da IFFT em ponto flutuante e sofrendo um arredondamento apenas na última parte do seu cálculo, a qual é a divisão. Fica evidente que o fator de giro da borboleta possui um erro de arredondamento. Isso faz com que alguns valores fiquem em discordâncias.

5.2.2 Multiplicações e divisões na IFFT

Além da representação do fator giro ser representado em ponto fixo, outro fator que prejudica a eficiência do bloco IFFT em VHDL é a multiplicação desse fator pelos valores de entrada advindos do bloco de simetria hermitiana.

O processo de divisão também altera o resultado final calculado. Para o módulo realizado utilizou-se de duas etapas, a qual a primeira foi a multiplicação por 707 e a outra foi a divisão por 1000. Apesar de se denominar divisão o que foi realizado foi uma multiplicação pelo fator 10^{-3}. Dessa maneira, o fator de giro sofreu influência tanto na sua multiplicação quanto na divisão de seu valor.

Para os valores especificados no bloco de mapeamento, não se observou problemas de *overflow* no bloco da IFFT. Porém, em alguns realizados com números mais elevados, observou-se que havia um problema de *overflow*. Apesar de ficar evidente esse problema, não há ineficiência no uso do bloco da IFFT quando interligada com todos os outros blocos.

A implementação da IFFT é realizada em sete estágios correspondentes. No entanto, isso não corresponde a quantidade de ciclos de *clocks* utilizado por esse bloco. Isso se deve principalmente a etapa de multiplicação pelo fator 0.707. O bloco consegue gerar

Tabela 1 – Valores de saída para o software Matlab e para o módulo VHDL

Entrada Real	Entrada Imaginário	Matlab Real	Matlab Imaginário	VHDL Real	VHDL Imaginário
XR0=10	XI0=0	83.000	0.000	83	0
XR1=15	XI1=0	-1.8787	-0.1213	-1	0
XR2=6	XI2=0	10.000	-9.000	10	-9
XR3=9	XI3=0	-6,1213	-4.1213	-6	-4
XR4=14	XI4=0	-7.000	0.000	-7	0
XR5=12	XI5=0	-6.1213	4.1213	-6	4
XR6=8	XI6=0	10.000	9.000	10	9
XR7=9	XI7=0	-1.8787	0.1213	-1	0
XR0=10	XI0=0	10	0	10	0
XR1=0	XI1=0	10	0	10	0
XR2=0	XI2=0	10	0	10	0
XR3=0	XI3=0	10	0	10	0
XR4=0	XI4=0	10	0	10	0
XR5=0	XI5=0	10	0	10	0
XR6=0	XI6=0	10	0	10	0
XR7=0	XI7=0	10	0	10	0
XR0=10	XI0=10	80	-80	80	-80
XR1=10	XI1=10	0	0	0	0
XR2=10	XI=10	0	0	0	0
XR3=10	XI3=10	0	0	0	0
XR4=10	XI4=10	0	0	0	0
XR5=10	XI5=10	0	0	0	0
XR6=10	XI6=10	0	0	0	0
XR7=10	XI7=10	0	0	0	0
XR0=10	XI0=0	80	0	80	0
XR1=10	XI1=10	48.2843	0	50	0
XR2=10	XI2=10	0	0	0	0
XR3=10	XI3=0	8.2843	0	10	0
XR410	XR4=0	0	0	0	0
XR5=10	XI5=-10	-8.2843	0	-11	0
XR6=10	XI6=-10	0	0	0	0
XR7=10	XI7=-10	-48.2843	0	-48	0
XR0=10	XI0=0	-60.000	0.000	-60	0
XR1=-10	XI1=-10	-28.2843	0.000	-28	0
XR2=-10	XI=-10	20.000	0.000	20	0
XR3=-10	XR3=-10	11.7157	0.000	13	0
XR4=-10	XI4=0	20.000	0.000	20	0
XR5=-10	XI5=10	28.2843	0.000	29	0
XR6=-10	XI6=10	20.000	0.000	20	0
XR7=-10	XI7=10	68.2843	0	68	0
XR0=10	XI0=0	-60.000	0.000	-60	0
XR1=-10	XI1=10	0.000	0.000	0	0
XR2=-10	XI2=-10	60.000	0.000	60	0
XR3=-10	XI2=-10	40.000	0.000	40	0
XR4=-10	XI4=0	20.000	0.000	20	0
XR5=-10	XI5=-10	0.000	0.000	0	0
XR6=-10	XI6=10	-20.000	0.000	-20	0
XR7=-10	XI7=-10	40.000	0.000	40	0

resultados depois de 12.5 clocks. Pode se concluir que a geração desses resultados é muito rápida. Apesar de haver uma aproximação no fator giro, o resultado final dos valores apresentou um intervalo de erro reduzido, o que permite concluir que o bloco tem um funcionamento bastante rápido e eficiente.

5.2.3 Resultados do DAC

A primeira parte dos resultados do PMOD foi adquirida ao se conectar a placa Nexys 3 e fazer o correto mapeamento de pinos. Através dessa relação de pinos, foi possível verificar os sinais de saída e a saída de verificação do PMOD. Para a medição de saída do conversor, foi utilizado o terra em uma ponta de prova, pino 5 do PMOD e feita a medição através da segunda ponta de prova nas saídas requeridas.A figura 29 demostra esse processo.

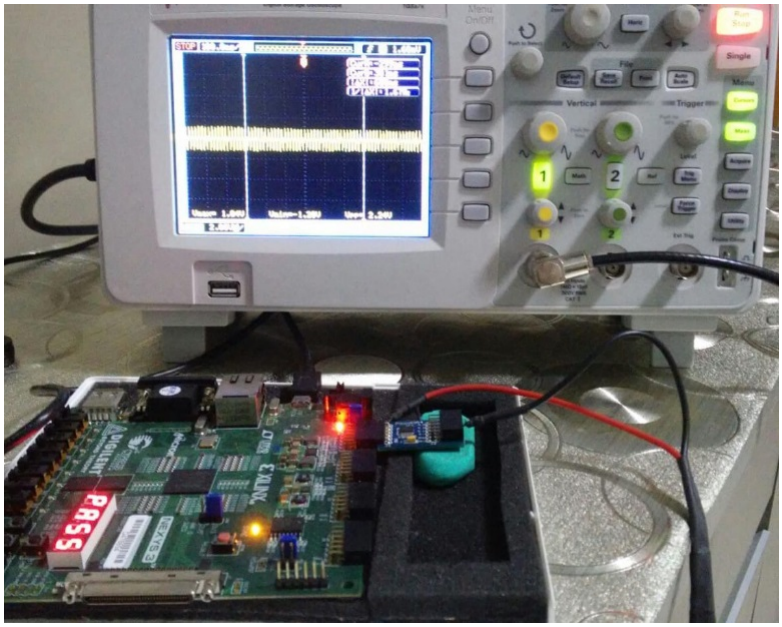

Figura 29 – Medições realizadas nos pinos do conversor digital analógico.

Para o correto funcionamento do DAC, foram realizados alguns testes. Esses possibilitaram a visualização, entendimento e compreensão dos seus recursos. A primeira parte da verificação do PMOD se deu com a observação dos sinais de saída para os pinos 1, 2, 4, 5 e 6. Esses fazem a correlação entre as entradas SYNC, DIN, SCLK, ground e alimentação em 3.3 volts. A figura 30 apresenta os sinais observados na tela do osciloscópio para as entradas dos pinos 1,2 e 4.

Pode se observar que o sinal SCLK apresenta uma frequência de 1.67 MHZ. Essa frequência está de acordo com o elaborado pela codificação para essa parte da verificação do DAC. Outro fator importante a ressaltar é a frequência do sinal SYNC. Esse possui uma frequência de 37.9 KHZ. Segundo o manual, essa frequência está de acordo com o

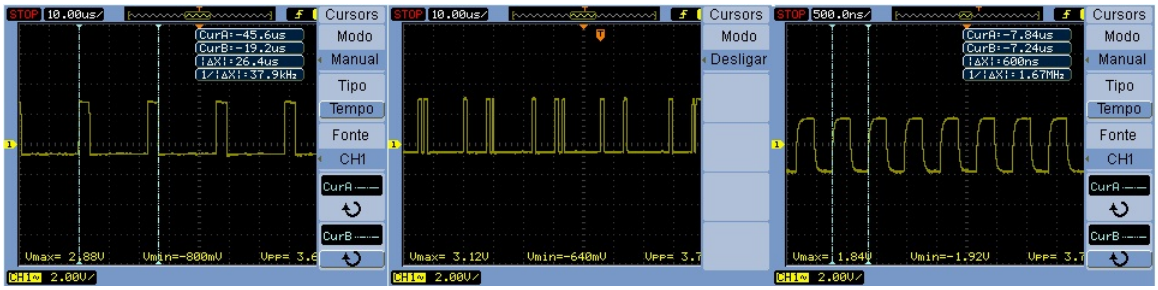

Figura 30 – Sinais SYNC, DIN, SCLK de excitação para o conversor.

correto funcionamento para o PMOD. Observando e tomando como base a forma de onda elaborada pelo fabricante, dada na página 24 do manual, é possível verificar que o PMOD possui as corretas excitações para seu correto funcionamento. Os pinos 5 e 6 são o terra e a alimentação do PMOD. Na figura é possível verificar que sempre há um acionamento inicial na entrada MOSI. Isso se deve ao fato de como foi elaborada a codificação do PMOD. Sempre que há um novo envio de dados ao PMOD há um acionamento para nível lógico alto para o correto envio ao registrador de entrada do PMOD.

Para verificar o sinal de saída do PMOD, foi utilizada uma saída no formato dente de serra. A programação foi realizada através de um código, o qual possui uma máquina de estados. Esta, a cada entrada, atualiza a saída em um *bit*. Observa-se que o nível mais baixo de tensão está representado em -1.52 Volts e o nível de tensão mais alto está representado em 1.20 Volts. Isso significa que a tensão de pico a pico é dada por 2.72 Volts. Essa saída de tensão está de acordo com o apresentado no manual do fabricante na página 22. A figura 31 apresenta o sinal dente de serra.

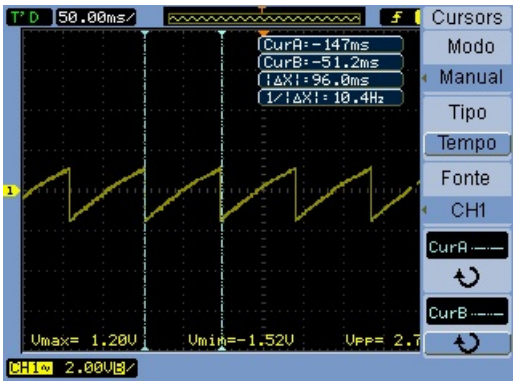

Figura 31 – Sinais SYNC, DIN, SCLK de excitação para o conversor.

5.2.4 Resultados do transmissor OFDM

Após armazenar os códigos na NEXYS3 um sinal de saída foi enviado ao módulo Digital/Analógico. Este sinal serviu para ser enviado ao osciloscópio. A primeira sequência

de *bits* enviada através do modulador foi "11111111". Na figura 32 pode ser visto uma das entradas para ser convertida para analógico.

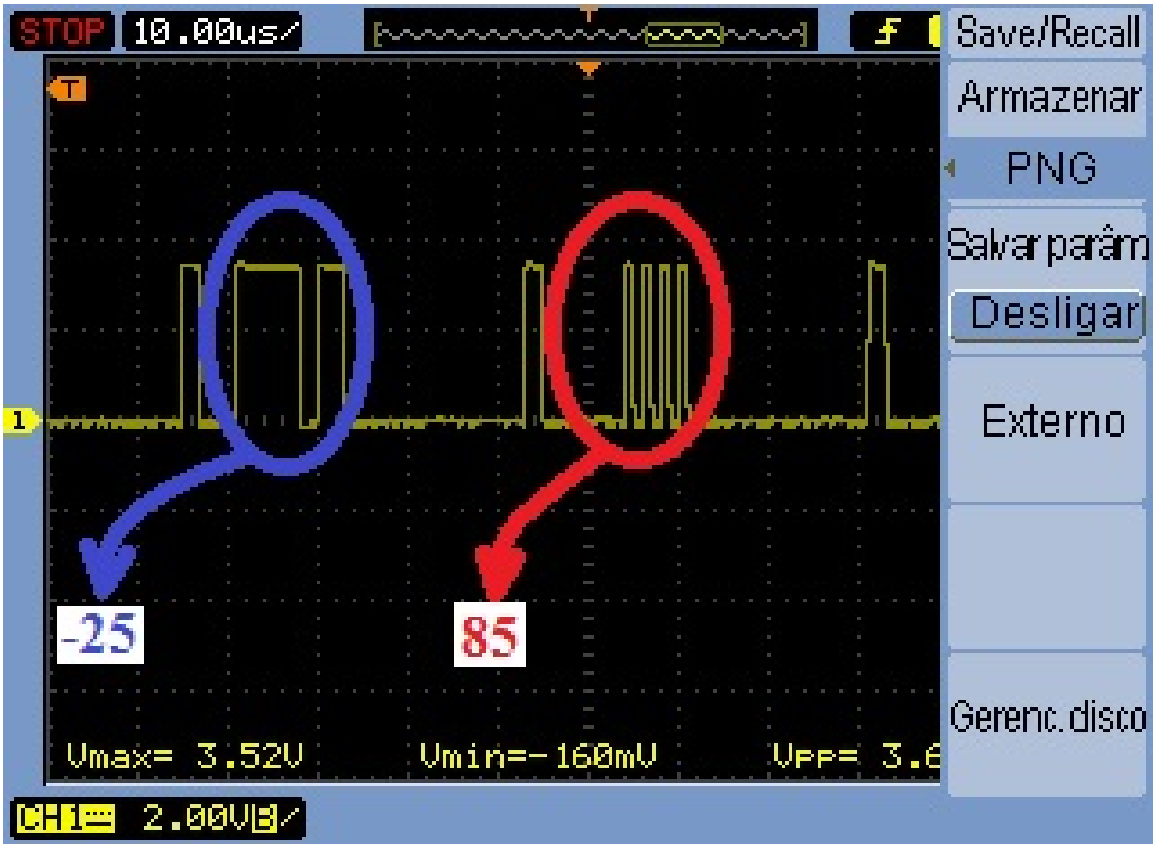

Figura 32 – Exemplo da entrada -25 e 85 no pino DIN de entrada do conversor D/A.

Sabendo do resultado adquirido pela entrada $(100, 100 + 100 \cdot i, 100 + 100 \cdot i, 100 + 100 \cdot i, 100, 100 - 100 \cdot i, 100 - 100 \cdot i, 100 - 100 \cdot i)$, é possível saber que ao passar pela IFFT os resultados obtidos serão apenas do tipo real. Dessa maneira, é possível verificar na figura 33 os resultados convertidos para analógico pelo módulo. Ao passar o vetor mencionado acima pela IFFT o resultado obtido é $75, -25, -25, -25, 85, 14, 35, -35$. Como foi utilizado complemento 2 para os resultados, os valores para números negativos estão apresentados na parte de maior tensão dos resultados apresentados na figura. Como os números estão apresentados em 12 *bits* os valores estão em um intervalo de -2048 a 2047. Para os resultados apresentados com o mapeamento não houve ocorrência de números fora desse intervalo.

A análise e verificação dos resultados no domínio da frequência para os valores de entrada são feitas a seguir: primeiramente, verificou-se quantos ciclos de *clock* eram necessários para que houvesse uma atualização dos valores da memória ROM, ou seja , de quanto em quanto tempo o transmissor é atualizado com a série de *bits*. Essa quantidade de *clock* foi obtida pela contagem separada de cada bloco e contabilizou-se em 38 ciclos de *clock* para cada amostragem de valor. Porém para todo o bloco é necessário que se

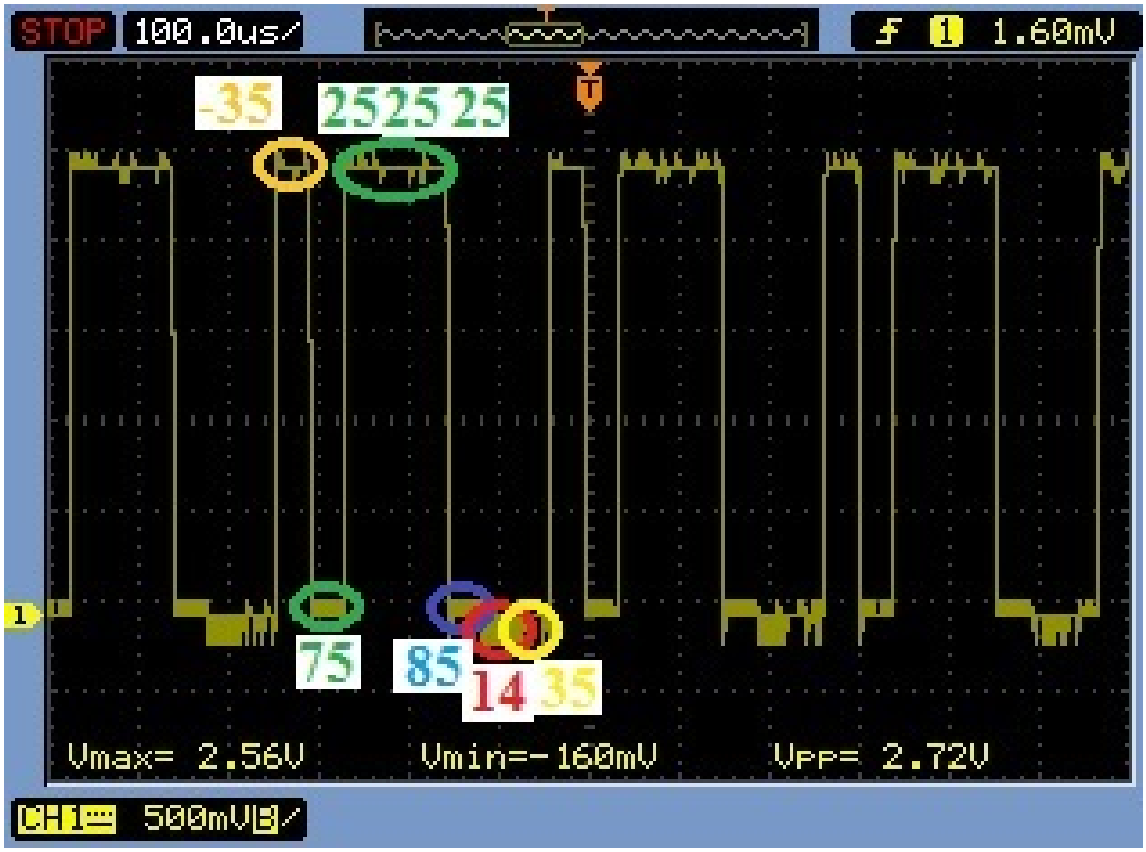

Figura 33 – Conversão dos valores obtidos pela entrada "11111111".

faça uma espera para 8 valores, contabilizando no total $38 \cdot 8 = 304$ ciclos de *clock*. A partir desse valor e conhecendo a velocidade de ciclos de *clock* para o sistema escolhido de 1MHZ por ciclos de clocks, pode se fazer as devidas análises e comparar com o resultado obtido pelo sinal enviado ao osciloscópio. Assim, o primeiro pico de sinal, segundo a base teórica estudada no capítulo inicial fica da seguinte maneira:

$$f_1 = \frac{1MHz}{304} = 3.28kHz \tag{5.1}$$

Os próximos valores para o sistema OFDM, no domínio da frequência, são múltiplos inteiros da primeira frequência. Assim, os valores teóricos são dados por :

$$f_2 = f_1 \cdot 2 = 3.28kHz \cdot 2 = 6.57kHz \tag{5.2}$$

$$f_3 = f_1 \cdot 3 = 3.28kHz \cdot 3 = 9.86kHz \tag{5.3}$$

$$f_4 = f_1 \cdot 4 = 3.28kHz \cdot 4 = 13.15kHz \tag{5.4}$$

$$f_5 = f_1 \cdot 5 = 3.28kHz \cdot 5 = 16.44kHz \tag{5.5}$$

$$f_6 = f_1 \cdot 6 = 3.28kHz \cdot 6 = 19.73kHz \tag{5.6}$$

$$f_7 = f_1 \cdot 7 = 3.28kHz \cdot 7 = 23.02kHz \tag{5.7}$$

$$f_8 = f_1 \cdot 8 = 3.28kHz \cdot 8 = 26.31kHz \tag{5.8}$$

A figura 34 apresenta os valores no domínio da frequência para o transmissor OFDM codificado.

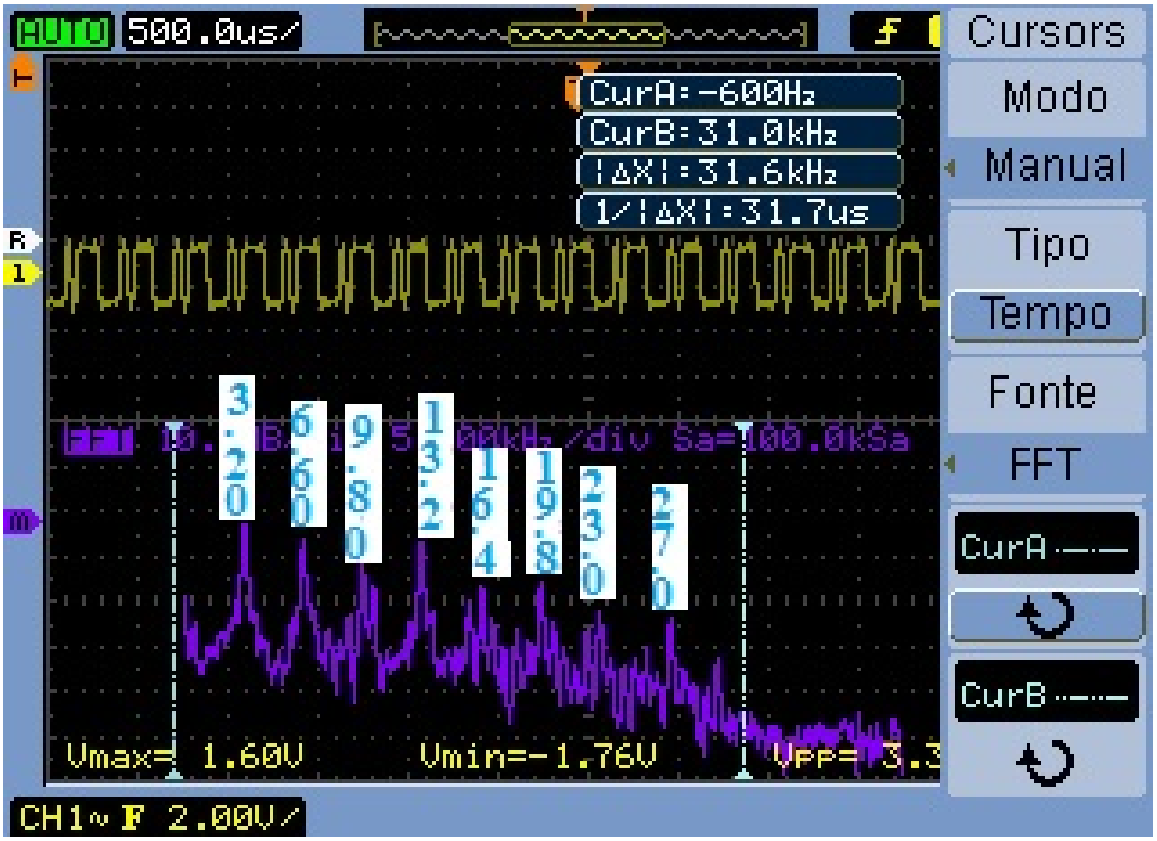

Figura 34 – Resultado dos valores de saída do modulador para o domínio da frequência.

5.2.5 Leitura e Escrita em arquivo VHDL

Esta etapa dedicou-se a análise do bloco transmissor através da escrita e leitura em arquivo na codificação VHDL. Nessa etapa, projetou-se um receptor em linguagem de descrição de *hardware*. Para a escrita em arquivo, utilizou-se a biblioteca *textio*. Como é necessário gerar um arquivo executável, este foi formado em formato TXT. Um para o bloco transmissor e outro para o módulo receptor. Os resultados obtidos pelo transmissor foram lidos pelo receptor. Com os valores lidos, o receptor gerou os valores e fez a escrita em outro arquivo TXT.

Tabela 2 – Tabela para os valores de saída do transmissor e receptor.

Entrada ROM Base 2	Entrada ROM Base 10	Saída Transmissor Base 10	Saída Receptor Base 2	Saída Receptor Base 10
00000001	1	100 -60 0 -10 0 10 0 60,	00000001	1
00000010	2	75 -85 -25 -35 -25 -14 -25 35	00000010	2
00000011	3	50 -60 -50 -10 -50 10 -50 60	00000011	3
00000100	4	75 -35 -25 14 -25 35 -25 85	00000100	4
00000101	5	50 -95 0 25 50 45 0 25	00000101	5
00000110	6	25 -120 -25 0 25 20 -25 0	00000110	6
00000111	7	0 -95 -50 25 0 45 -50 25	00000111	7
00001000	8	25 -70 -25 50 25 70 -25 50	00001000	8
00001001	9	50 -60 50 60 50 10 -50 -10	00001001	9
00001010	10	25 -85 25 35 25 -14 -75 -35	00001010	10
00001011	11	0 -60 0 60 0 10 -100 -10	00001011	11
00001100	12	25 -35 25 85 25 35 -75 14	00001100	12
00001101	13	100 -25 50 25 0 -25 -50 25	00001101	13
00001110	14	75 -50 25 0 -25 -50 -75 0	00001110	14
00001111	15	50 -25 0 25 -50 -25 -100 25	00001111	15
00010000	16	75 0 25 50 -25 0 -75 50	00010000	16
00010001	17	50 -60 50 -10 -50 10 50 60	00010001	17
00010010	18	25 -85 25 -35 -75 -14 25 35	00010010	18
00010011	19	0 -60 0 -10 -100 10 0 60	00010011	19
00010100	20	25 -35 25 14 -75 35 25 85	00010100	20
00010101	21	0 -95 50 25 0 45 50 25	00010101	21
00010110	22	-25 -120 25 0 -25 20 25 0	00010110	22
00010111	23	-50 -95 0 25 -50 45 0 25	00010111	23
00011000	24	-25 -70 25 50 -25 70 25 50	00011000	24
00011001	25	0 -60 100 60 0 10 0 -10	00011001	25
00011010	26	-25 -85 75 35 -25 -14 -25 -35	00011010	26
00011011	27	-50 -60 50 60 -50 10 -50 -10	00011011	27
00011100	28	-25 -35 75 85 -25 35 -25 14	00011100	28
00011101	29	50 -25 100 25 -50 -25 0 25	00011101	29
00011110	30	25 -50 75 0 -75 -50 -25 0	00011110	30

A Tabela 2 apresenta os valores de entrada da ROM, a saída do transmissor e a saída do receptor. No apêndice há a continuação dessa tabela.

Ao observar e comparar os resultados de entrada e saída, chega-se a conclusão de que há 10 valores de saída incoerentes com a entrada. Isso resulta em uma taxa de erro de 7.87%. A análise dos prováveis erros obtidos são apresentadas na primeira parte desse capítulo.

Foi feita uma comparação entre os valores de entrada da ROM e os valores de saída do bloco receptor. Verificou-se correspondência significativa entre esses resultados. Dessa maneira, pode se concluir que os blocos foram projetados de maneira eficiente.

5.3 Resultado da Simulação do transmissor e Receptor OFDM

A partir da implementação realizada do bloco transmissor e Receptor no *software* Matlab, pode-se obter os gráficos adquiridos pela implementação no tempo e no domínio da frequência. O primeiro gráfico obtido foi o do domínio do tempo. Esse gráfico apresenta uma função com varias funções degraus. Isso está de acordo com o estabelecido pela implementação em VHDL e demonstrado no osciloscópio. O resultado da geração do primeiro gráfico se deu com o envio de números com valores inteiro 1. Para a simulação foram utilizadas 500 amostras, as quais geraram um gráfico que pode ser analisado. A imagem do gráfico está disponível na figura 35.

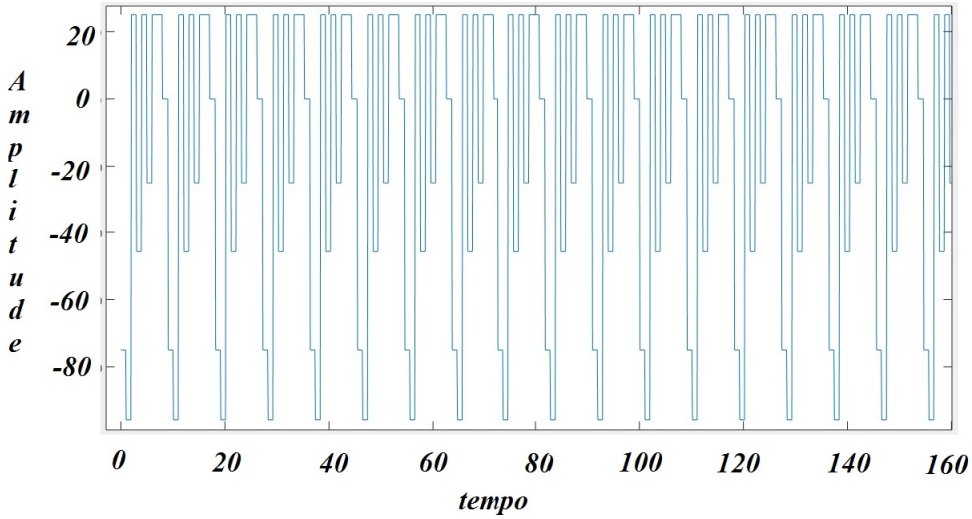

Figura 35 – Resultado dos valores de saída da simulação no Matlab para o transmissor OFDM.

Com o resultado da imagem 35, passou–se a obter os valores da transformada rápida de Fourier. Nessa parte utilizou-se um filtro, o qual é um janelamento para simular de forma mais eficiente aquilo que se obtém no osciloscópio. O resultado obtido é semelhante com aquele observado no domínio da frequência na figura 30. Pelas análises dos resultados obtidos, os resultados são equivalentes, tanto da simulação realizada no Matlab, quanto a estabelecida pela codificação em VHDL. Porém, não houve uma correspondência com os resultados das bibliografias estudadas quanto ao resultado obtido pelo osciloscópio no domínio da frequência. O resultado obtido com transformada rápida de Fourier do gráfico da imagem 35 está abaixo:

A partir da geração do gráfico no domínio da frequência, observou-se raias intercaladas por valores fixos e constantes. Esses espaçamentos estão de acordo com os estabelecidos na imagem em VHDL. Esses valores estão relacionados com o tamanho de cada valor da função degrau.

A partir dos resultados obtidos, passou-se a analisar quais seriam os fatores que

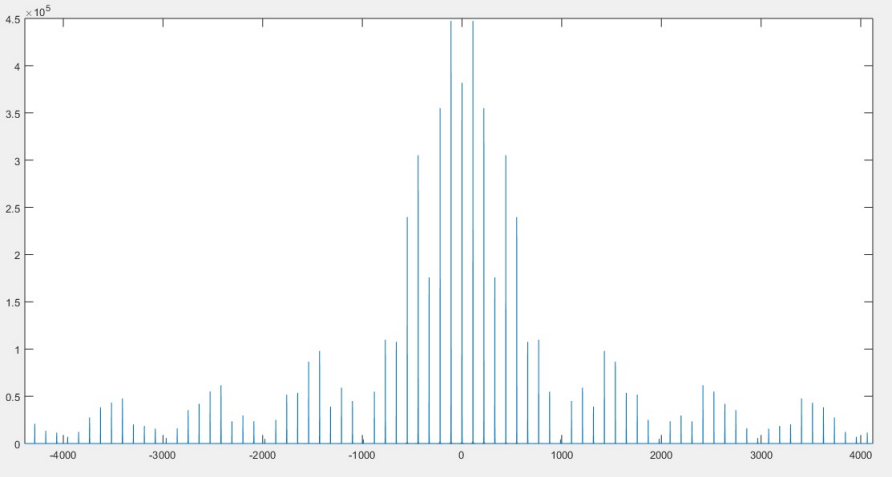

Figura 36 – Resultado dos valores de saída da simulação no Matlab para o transmissor OFDM no domínio da frequência.

estariam influenciando o resultado das raias no domínio da frequência a terem valores diferentes. Em consequência disso, a partir dos resultados obtidos, foram feitas mudanças na simulação realizada no Matlab. Isso foi realizado para se adequar com os estudos estabelecidos na fundamentação teórica.

A primeira mudança a ser realizada foi a do tamanho da constelação. Passou-se a ter um mapeamento de 512QAM e 1024QAM. Porém, essa mudança não demonstrou nenhuma alteração significativa nos picos de raias do espectro em frequência. O resultado dessa primeira tentativa é bem semelhante com a figura 34.

A segunda possibilidade a ser verificada foi a utilização de vários bits de entrada com valores, utilizando uma função que gera valores aleatórios. Assim, foi utilizada no matlab uma função já implementada denominada randint. Essa possibilita gerar valores aleatórios inteiros. Para utilizar essa função foi utilizado um intervalo de números naturais entre zero e cem. O resultado obtido no domínio da frequência também não afetou as raias do espectro. O espectro continuou com valores de raias de módulos distintos.

A terceira análise decorreu da passagem de um filtro sobre o gráfico do domínio do tempo. A partir da utilização de um filtro passa baixas, já implementado no *software* Matlab, observou-se que o gráfico ficou com um formato mais suave nas mudanças de um valor para outro. A partir desse gráfico mais suave, pode-se obter a FFT, porém não se obteve sucesso na geração de raias de amplitudes aproximadas. A figura 37 apresenta o gráfico para os valores de saída no domínio do tempo utilizando o filtro passa baixas.

Com o aumento da resolução do gráfico pela ferramenta utilizada no programa Matlab, notou-se que o gráfico possuía mudanças abruptas de um valor para outro, ou seja, picos entre os valores do gráfico. Dessa maneira, optou-se por fazer uma interpolação polinomial. Essa interpolação suavizou de modo eficiente o gráfico da figura 35. A partir

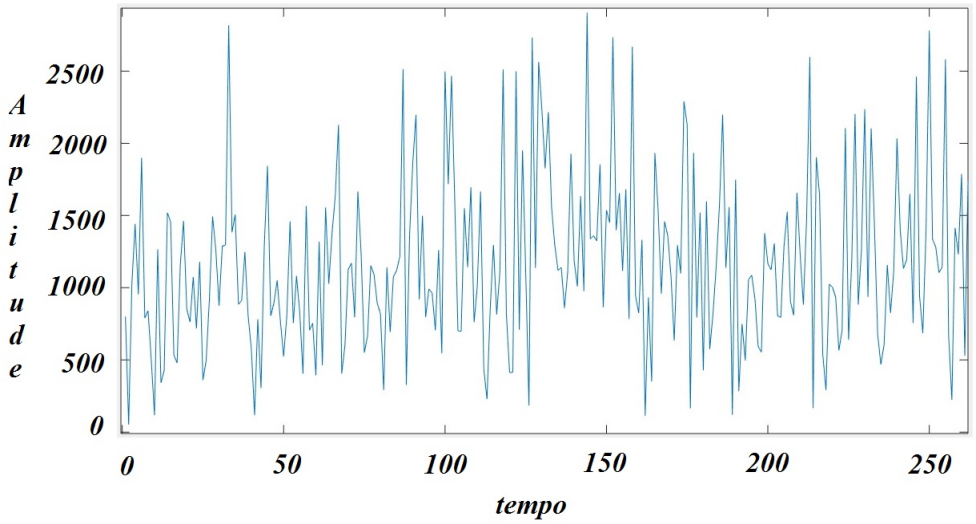

Figura 37 – Resultado dos valores de saída com a utilização de um filtro passa baixas.

desse novo gráfico, pode-se utilizar a FFT e obter um resultado, porém novamente, não se obteve resultados satisfatórios, os quais pudessem ser parecidos entre teoria e a simulação no domínio da frequência.

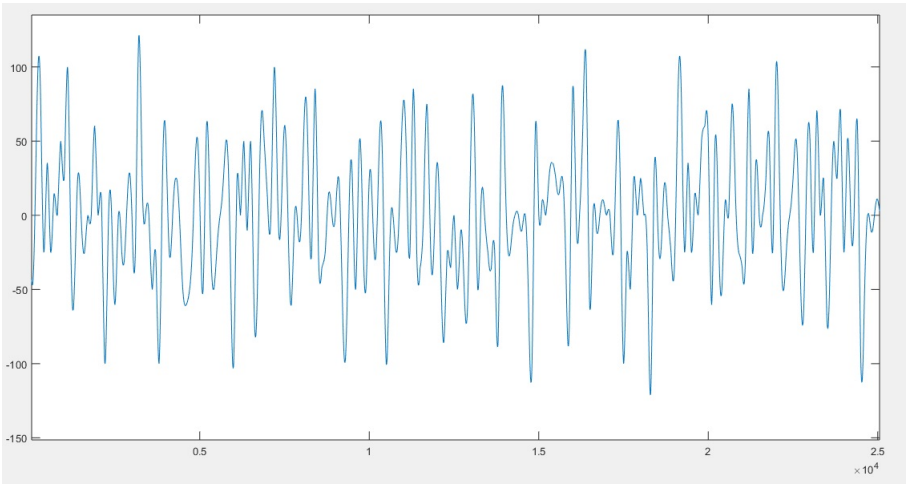

Figura 38 – Resultado dos valores de saída com a utilização de uma interpolação.

Todos os resultados das quatro primeiras possibilidades resultaram em um espectro com amplitudes bem diferentes.

A última análise realizada foi o sincronismo entre os tamanhos dos blocos. Nessa última abordagem, os blocos foram realizados com uma perspectiva de espelhamento, ou seja, tanto os blocos do transmissor e do receptor foram realizados com tamanhos de pontos iguais. Nessa parte do trabalho, os módulos foram projetados para fazerem uma correspondência de tamanho de pontos. Assim, a FFT do bloco receptor implementada passou a realizar a transformada rápida de Fourier com valores de oito pontos, os quais são enviados ao bloco de anti-simetria hermitiana e assim obtidos os valores desse bloco. A

partir da geração desses valores em um gráfico, o espectro obtido revelou consistência com a parte teórica. A imagem 39 representa os oito valores de raias do espectro na frequência obtidos pela modulação do bloco OFDM.

É de extrema importância ressaltar o sincronismo e correspondência entre os blocos em uma modulação OFDM. Para se obter valores de raias no osciloscópio de amplitudes com médias iguais, o transmissor deverá trabalhar com pontos de constelação iguais ao estabelecido pelo módulo IFFT desenvolvido em VHDL. Há, portanto, uma fase importante no desenvolvimento de transmissores e receptores, os quais devem estar não apenas sincronizados em tempo, mas em tamanho.

Figura 39 – Resultado dos valores de saída com a utilização de uma interpolação.

Pode-se observar que os valores são coerentes com o mapeamento QAM. O mapeamento realizado, conforme visto na seção de implementação do bloco 4QAM, foi mapeado com valores definidos em cem para quadratura e fase. O módulo desses valores corresponde a aproximadamente 141, o que corresponde aos valores da imagem 35. Os valores nos pontos um e cinco estão com amplitude menor pelo fato da separação da parte real e imaginária no bloco de simetria hermitiana.

6 Conclusão

Com o atual avanço tecnológico das FPGAs, tornou-se possível, juntamente com as técnicas de transmissão digital de dados, a implementação de sistemas de transmissão digital integrados em um únicos chip. No presente trabalho foram realizados dois blocos, um transmissor OFDM e um receptor. Estes compostos pela IFFT e FFT para fazer a modulação propriamente dita. As análises das implementações demonstram que o tipo de implementação é eficiente, robusta, rápida. Com o sucesso da implementação foi possível a verificação dos resultados através da Placa Nexys 3 no analisador de onda, porém este aparelho não se mostrou eficiente para validação do espectro no domínio da frequência.

O transmissor possui sete blocos. Dentre eles estão Memória ROM, Serial Paralelo, Mapeador QAM, Simetria Hermitiana, IFFT, Paralelo para Serial e conversor digital para analógico. A IFFT utilizou 16 entradas. É importante salientar que a IFFT de 16 entradas funcionou de forma adequada para valores advindos do bloco simetria hermitiana. Na etapa de validação da IFFT foi utilizado o software Matlab para a comparação dos valores. Isso facilitou a análise, já que o Matlab possui uma função IFFT já implementada em ponto flutuante. Comparando os resultados obtidos, pode-se concluir que o módulo implementado em hardware é eficiente e não possui um intervalo que prejudique drasticamente os resultados. Assim, concluiu-se que o bloco da IFFT pode ser utilizado de forma viável no bloco transmissor. Em cada bloco do transmissor e receptor, foram feitas análises e testes através do software ISE SIMULATOR. Dessa forma, é possível concluir, pela quantidade de clocks do transmissor e receptor, que os módulos possibilitam rápido envio de dados. Com a validação dos blocos separadamente, foi possível juntá-los e obter o sinal de saída no osciloscópio.

Para o módulo receptor, foi feito o mesmo procedimento de design, análise e verificação dos dados. Assim como a IFFT, a FFT foi analisada através de uma função já implementada no Matlab. Através da comparação dos resultados, pode-se concluir que o módulo trabalha de forma eficiente e que os valores de saída estão bem próximos dos resultados obtidos pelo Matlab. A discussão para as diferenças encontradas entre o Matlab e o bloco desenvolvido em VHDL está no capítulo de análise, porém é importante ressaltar que a diferença de resultados está muito baseada no fato de que o Matlab implementa essa função em ponto flutuante, enquanto o bloco VHDL foi projetado para se trabalhar em ponto fixo. Diferentemente da IFFT, a FFT foi projetada com oito entradas. O bloco foi projetado dessa maneira para economizar espaço de hardware e pelo fato do resultado do transmissor gerar apenas valores reais. Assim, pelos resultados obtidos para o bloco FFT, este pode ser utilizado de maneira viável para o módulo receptor.

A observação dos valores no analisador de sinais, possibilitou concluir que o sistema está com raias devidamente espaçadas. Porém o osciloscópio mostrou-se ineficiente para obter valores das raias. Fato já discutido na seção de análise. Ficou observado que há um janelamento desconhecido do osciloscópio e uma quantidade de utilização de pontos da FFT diferente da usada no projeto do transmissor OFDM. A verificação, realizada no capítulo de análise, mostrou que as portadoras em seus devidos valores de frequência são coerentes. É importante ressaltar que foi necessária uma codificação bem elaborada de sincronismo entre os módulos do transmissor e do conversor digital analógico. Esse sincronismo foi realizado de maneira eficiente. Isso foi possível através da escolha de um sinal de start em cada bloco e de um sinal de ready na saída dos blocos. Isso permitiu a cada bloco trabalhar de maneira adequada, o que não permite que haja sempre a atualização indesejada dos blocos e nem um processo de atualização antes da hora para os blocos.

6.1 Proposta para Trabalhos Futuros

Nessa etapa são feitas algumas sugestões para a melhora do projeto. A primeira recomendação é usar uma quantidade de bits maior para a representação numérica dos bits nas Transformadas de Fourier. Como se utilizou 12 bits, uma melhora considerável seria a utilização de 16 bits. Isso resultaria em um intervalo maior de números com a probabilidade de erro de *overflow* menor.

Outro fator importante a sugerir é a mudança de ponto fixo para ponto flutuante. Isso resultaria em uma melhor aproximação na saída dos módulos, principalmente, para as transformadas de Fourier, já que utilizam uma aproximação para o fator Twiddle. Isso resultaria em uma acurácia muito melhor nos resultados dos módulos.

Além disso, sugere-se criar um módulo com uma quantidade maior de entradas. É possível ter um modulador eficiente, através do transmissor realizado, com uma quantidade de 1028 subportadoras por exemplo. Isso resultaria em melhor performance e melhor análise no espectro visualizado no osciloscópio.

Para os trabalhos futuros, é essencial também a elaboração de outros módulos que compõem o sistema OFDM. Pode-se destacar o interleaving, código de correção de erros, outras módulações como QPSK, PSK, FSK, ASK entre outras, prefixo cíclico, filtro passa banda. Com a elaboração desses módulos, o sistema OFDM será muito eficiente e robusto.

Para uma melhor eficiência dos blocos transmissor e receptor e com a utilização dos sinais de sincronismo dos blocos já implementados, é possível construir os módulos com uma arquitetura pipeline. Os benefícios para esse tipo de implementação são enormes.

Referências

BAHAI, A. *Multi Carrier Digital Communications Theory and Applications of OFDM*. BOOKMAN COMPANHIA ED, 2002. ISBN 405059953. Disponível em: <https://books.google.com.br/books?id=52YRRsLie1IC>. Citado 5 vezes nas páginas 29, 36, 37, 38 e 40.

B.P.LATHI. *Sistemas de Comunicação Analógica e Digitais Modernas*. BOOKMAN COMPANHIA ED, 2010. ISBN 4050553. Disponível em: <https://books.google.com.br/books?id=52YRRsLie1IC>. Citado 7 vezes nas páginas 25, 29, 30, 31, 33, 35 e 37.

B.PUJITA, S. Robust implementation of OFDM system using VHDL. Internation Journal of Reasearch, 2014. Citado na página 25.

CASTRO, A. P. de. Tecnologias de comunicação sem fio: conceitos e aplicações de redes locais wireless. Encontro de Pesquisa e Extensão, 2016. Citado na página 25.

GUTIERREZ, F. M. *Implementation of a Tx/Rx OFDM System in a FPGA MASTER DEGREE: Master in Science in Telecommunication Engineering Management*. New York, United States, 2001. 13 p. Citado na página 46.

KRZYSZTOF, W. *Introduction To Digital Communication System*. Wiley, 2011. ISBN 4050599533. Disponível em: <https://books.google.com.br/books?id=52YRRsLie1IC>. Citado na página 34.

OPPENHEIM, A. V. *Sinais e sistemas*. São Paulo, Brasil: [s.n.], 2010. 428 p. ISBN 85-87090-90-9. Citado na página 41.

P.Y.TSAI, Y. K. Joint weighted least-squares estimation of carrier-frequency offset and timing offset for OFDM systems over multipath fading channels. IEEE, 2005. Citado na página 25.

R.CHANG, C. A theoretical study of performance of an orthogonal multiplexing data transmission scheme. IEEE, 1968. Citado na página 37.

RIBEIRO, J. B. B. Telecomunicações volume 7. Rio de Janeiro, 2013. Citado 2 vezes nas páginas 36 e 37.

SALTZBERB, A. R. *Multi Carrier Digital Comm OFDM*. BOOKMAN COMPANHIA ED, 2004. ISBN 405059953. Disponível em: <https://books.google.com.br/books?id=52YRRsLie1IC>. Citado na página 38.

SIQUEIRA, T. Implementação de um modem OFDM em FPGA. Universidade Federal do Espírito Santo,Departamento de Engenharia Elétrica, 2004. Citado 5 vezes nas páginas 15, 29, 36, 41 e 42.

TACURI, A. B. H. Desempenho de sistemas OFDM em canais não-lineares variantes no tempo. Rio de Janeiro, julho de 2014, 2014. Citado na página 40.

TSE, D.; VISWANATH, P. *Fundamentals of Wireless Communication*. [S.l.]: Cambridge University Press, 2005. Citado na página 25.

WEINSTEIN, S.; EBERT, P. Data transmission by frequency-division multiplexing using the discrete fourier transform. IEEE Transactions on communication Technology, 1971. Citado na página 25.

XIONG, F. *Digital Modulation Techniques*. Artech House, 2000. ISBN 4050599533. Disponível em: <https://books.google.com.br/books?id=52YRRsLie1IC>. Citado 5 vezes nas páginas 30, 31, 32, 33 e 34.

ZOU.W.Y. COFDM: an overview, vol 41. IEEE Transactions on Communications, 1995. Citado na página 37.