

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE ENGENHARIA ELETRÔNICA
ENGENHARIA ELETRÔNICA**

**JULIANA INÁCIO DOS SANTOS
PAULO HENRIQUE DELALLO MARTINS LAMPA**

**DESENVOLVIMENTO DE UM GERADOR DE ONDAS ARBITRÁRIAS
UTILIZANDO A TÉCNICA DE SÍNTESE DIGITAL DIRETA E COM
RECURSOS DE MODULAÇÃO**

TRABALHO DE CONCLUSÃO DE CURSO

**TOLEDO
2018**

**JULIANA INÁCIO DOS SANTOS
PAULO HENRIQUE DELALLO MARTINS LAMPA**

**DESENVOLVIMENTO DE UM GERADOR DE ONDAS ARBITRÁRIAS
UTILIZANDO A TÉCNICA DE SÍNTESE DIGITAL DIRETA E COM
RECURSOS DE MODULAÇÃO**

Trabalho de conclusão de curso de graduação apresentado à disciplina Trabalho de Conclusão de Curso 2, da Coordenação de Engenharia Eletrônica – COELE - da Universidade Tecnológica Federal do Paraná – UTFPR, Câmpus Toledo como requisito parcial para a obtenção do título de Engenheiro Eletrônico.

Orientador: Prof. Dr. Felipe Walter Dafico Pfrimer.

Coorientador: Prof. Dr. Alberto Yoshihiro Nakano.

**TOLEDO
2018**

TERMO DE APROVAÇÃO

Título do Trabalho de Conclusão de Curso Nº 77

**DESENVOLVIMENTO DE UM GERADOR DE ONDAS ARBITRÁRIAS
UTILIZANDO A TÉCNICA DE SÍNTESE DIGITAL DIRETA E COM
RECURSOS DE MODULAÇÃO**

por

Juliana Inácio dos Santos
Paulo Henrique Delallo Martins Lampa

Esse Trabalho de Conclusão de Curso foi apresentado às **14:00 do dia 29 de junho de 2018** como **requisito parcial** para a obtenção do título de **Bacharel em Engenharia Eletrônica**. Após a deliberação da Banca Examinadora, composta pelos professores abaixo assinados, o trabalho foi considerado **APROVADO**.

Marcos Roberto Bombacini
UTFPR

Daniel Cavalcanti Jeronymo
UTFPR

Felipe Walter Dafico Pfrimer
Orientador (a)

Fabio Rizental Coutinho
Coordenador (a) da COELE

O termo de aprovação assinado encontra-se na coordenação do curso.

Toledo, 29 de junho de 2018

Dedicamos este trabalho à nossas famílias.

A força não provém da capacidade física. Provém de uma vontade indomável.

Mahatma Gandhi

AGRADECIMENTOS

Somos gratos primeiramente a Deus pela vida e pela oportunidade a nós concedida. A nossa família que durante todo esse trajeto tem nos dado suporte e amor. Ao nosso orientador Felipe Walter Dafico Pfrimer e coorientador Alberto Yoshihiro Nakano, que além de nos mostrar o caminho a seguir nos ajudou a trilhá-lo. E as amizades que fizemos no decorrer deste curso, que não serão esquecidos.

RESUMO

O gerador de funções está entre os equipamentos mais importantes de teste eletrônico. Serve para fornecer sinais elétricos controláveis que são usados em testes de circuitos. Esse dispositivo é capaz de gerar sinais periódicos ou não, de forma contínua ou limitada, sincronizada e modulada. O gerador de ondas comercial possui poucos parâmetros ajustáveis e se limita em formas de ondas senoidais, retangulares e triangulares. O gerador de ondas arbitrárias (*arbitrary waveform generator* ou AWG) permite a configuração quase que total do sinal, sendo possível ao usuário configurar um sinal ponto por ponto. Dessa forma, o objetivo deste trabalho foi o desenvolvimento de um AWG de baixo custo que poderá ser utilizado em laboratórios de instituições de ensino. Para a construção do protótipo foi utilizado o kit Mercurio IV que possui um FPGA (*Field Programmable Gate Array*) e o chip FT245R. A programação do gerador foi baseada na técnica DDS (*direct digital synthesis*), onde o sinal a ser sintetizado é armazenado em uma memória e um contador de passo ajustável carrega os dados da memória em um DAC. Também foi desenvolvido um estágio de saída que permite ajustar o ganho e a linha base (nível em corrente contínua) do sinal gerado. Os resultados obtidos foram satisfatórios visto que o gerador de ondas arbitrária foi capaz de sintetizar as ondas requeridas, realizar as modulações analógicas e digitais e por fim controlar o ganho e *offset* através do estágio de saída.

Palavras-chave: Gerador de funções; Instrumentação eletrônica; AWG; FPGA.

ABSTRACT

The waveform generator is one of the most important measurements equipment. It provide controllable electrical signals that are used in circuit testing. This device is capable of generating periodic or non-continuous signals, in a continuous or limited way, synchronized and modulated. The common waveform generator has few adjustable parameters and is limited to sine, rectangular, and triangular waveforms. The arbitrary waveform generator (AWG) allows almost a total configuration of the signal, allowing the user to configure a signal point-by-point. In this way, the objective of this work was the development of a low cost AWG that can be used in Educational Institutions' laboratories. For the construction of the prototype, it was used a Mercurio IV kit that has a FPGA (Field Programmable Gate Array) and the FTDI FT245 chip. The programming of the generator was based on the DDS (Direct Digital Synthesis) technique, where the signal to be synthesized is stored in a memory and an adjustable step counter loads the data of the memory to a DAC. An output stage that allows adjusting the gain and the base line (direct current level) of the generated signal was also developed. The results were satisfactory since the AWG was able to synthesize the required waves, perform analog and digital modulations, and control the gain and offset in the output stage.

Keywords: Waveform generator; Electronic instrumentation; AWG; FPGA.

LISTA DE FIGURAS

Figura 1 - Formas de ondas: a) senoidal, b) triangular e c) quadrada.....	16
Figura 2 - Gerador de Funções 4040DDS da marca <i>BK PRECISION</i>	16
Figura 3 - Gerador de ondas arbitrárias 33210A – <i>Keysight Technology</i>	19
Figura 4 - <i>Software Keysight 33503A BenchLink Waveform Builder Pro and Basic Software</i>	20
Figura 5 - Estrutura básica da técnica DDS	21
Figura 6 - Endereçamento da LUT	21
Figura 7 - Truncamento de fase aplicado a uma onda dente de serra.....	23
Figura 8 - Modulação ASK.....	26
Figura 9 - Modulação FSK	27
Figura 10 - Arquitetura de um FPGA	28
Figura 11 - Diagrama de blocos conversão digital – analógico	29
Figura 12 - Sinal $X_d[n]$	29
Figura 13 - Sinal $X_s[n]$	30
Figura 14 - Sinal $X_a'[t]$	30
Figura 15 - Sinal $X_a[t]$	30
Figura 16 - Multiplicação por soma de logaritmo	32
Figura 17 - Configuração AD633	32
Figura 18 - Multiplicador de transcondutância básico	33
Figura 19 - Placa de desenvolvimento Mercurio® IV	34
Figura 20 - Conexão dos periféricos ao FPGA.....	35
Figura 21 - Chip FT245R.....	36
Figura 22 - Configuração do DAC7821.....	37
Figura 23 - Informações de tempo DAC7821.....	37
Figura 24 - Configuração pinos.....	38
Figura 25 - OPA727	39
Figura 26 - Diagrama de blocos geral do sistema.....	40
Figura 27 - Software de configuração da onda desenvolvido	40
Figura 28 - Configuração da onda arbitrária	41
Figura 29 - Forma de ondas padrões	41
Figura 30 - Definição das frequências e transmissão dos dados.....	42
Figura 31 - Encapsulamento de dados desenvolvido.....	42
Figura 32 - Diagrama de blocos dos circuitos sintetizados	43
Figura 33 - Diagrama de blocos da onda modulante.....	44
Figura 34 - Saída do acumulador de fases.....	44
Figura 35 - Diagrama de blocos da onda portadora	45
Figura 36 - Diagrama de blocos do Estágio de Saída.....	46
Figura 37 - Circuito conversor com o DAC7821	47
Figura 38 - Multiplicador com AD633.....	48
Figura 41 - Sinais quadrado e senoidal com <i>offset</i>	49
Figura 39 - Resultado amplificador com AD633 sem <i>offset</i>	49
Figura 40 - Resultado amplificador AD633 com <i>offset</i>	50
Figura 42 - Sinais senoidais	51
Figura 43 - Ondas Triangulares	52

Figura 44 - Ondas quadradas	53
Figura 45 - Sinais arbitrários	54
Figura 46 - Modulação em amplitude.....	55
Figura 47 – Simulação da modulação AM com modulante senoidal.....	55
Figura 48 - Modulação FM.....	56
Figura 49 – Simulação modulação FM com modulante senoidal.....	56
Figura 50 - Modulações digitais ASK e FSK.....	57
Figura 51 – Simulação da modulação ASK.....	57
Figura 52 – Simulação da modulação FSK	58

LISTA DE ABREVIATURAS E SIGLAS

AF	Acumulador de Fase
AM	<i>Amplitude Modulation</i>
AM-DSB	<i>Amplitude Modulation Double sideband</i>
AM-DSB-SC	<i>Amplitude Modulation Double sideband-supressed carrier</i>
AM-SSB	<i>Amplitude Modulation Single sideband</i>
AM-VSB	<i>Amplitude Modulation Vestigial sideband</i>
ASK	<i>Amplitude Shift-Keying</i>
AWG	<i>Arbitrary Waveform Generator</i>
CC	Corrente Contínua
CLB	<i>Configurable Logic Blocks</i>
DAC	<i>Digital to Analog Converter</i>
DDS	<i>Direct Digital Synthesis</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
FFT	<i>Fast Fourier Transform</i>
FIFO	<i>Firs-In-First-Out</i>
FM	<i>Frequency Modulation</i>
FPGA	<i>Field-Programmable Gate Array</i>
FSK	<i>Frequency Shift-Keying</i>
FTW	<i>Frequency Tuning Word</i>
GPIO	<i>General Purpose Input/Output</i>
LAN	<i>Local Area Network</i>
LUT	<i>Look-Up Table</i>
PLD	<i>Programmable Logic Devices</i>
PWM	<i>Pulse-Width Modulation</i>
SPI	<i>Serial Peripheral Interface</i>
SRAM	<i>Static Random Access Memory</i>
TJB	<i>Transistor de junção bipolar</i>
USB	<i>Universal Serial Bus</i>
VGA	<i>Variable Gain Amplifier</i>
VHDL	<i>Very High Speed Integrated Circuits Hardware Description Language</i>

LISTA DE SÍMBOLOS

A_1	Amplitude da portadora no estado 1
A_2	Amplitude da portadora no estado 2
F_1	Frequência da portadora no estado 1
F_2	Frequência da portadora no estado 2
f_c	Frequência da portadora.
F_{clk}	Frequência de <i>clock</i> de referência
$f_i(t)$	Frequência instantânea da portadora
F_{out}	Frequência do sinal gerado
FTW	Incremento do acumulador de fase
I_C	Corrente de coletor
I_{C1}	Corrente de coletor 1
I_{C2}	Corrente de coletor 2
k	Constante de Boltzmann
K_f	Sensibilidade à frequência
m	Índice de modulação
M	Número de bits do acumulador de fase
$m(t)$	Sinal da mensagem a ser modulada
p	Número de bits utilizado na FTW
q	Carga do elétron
Q	Número de endereçamentos realizados na memória por ciclo
$s(t)$	Sinal modulado
t	Tempo
T	Temperatura absoluta(dado em Kelvin)
T_b	Duração de bit dado em segundos
T_s	Período de amostragem
v_{AM}	Valor de tensão instantâneo da onda modulada
V_{BE}	Tensão de base-emissor
V_c	Valor de pico da onda portadora

v_c	Valor de tensão instantâneo da portadora
V_m	Valor de pico da onda modulante
v_m	Valor de tensão instantâneo da modulante
V_x	Tensão na entrada X
V_y	Tensão na entrada Y
W	Saída do multiplicador
$X_d[n]$	Sinal digital de entrada
$X_s[n]$	Equivalente em tensão do valor em binário
$X_a'[t]$	Sinal após passar pelo circuito de <i>holder</i>
$X_a[t]$	Sinal convertido em analógico
X_1	Entrada diferencial
X_2	Entrada diferencial
Y_1	Entrada diferencial
Y_2	Entrada diferencial
Z	Entrada somadora
ΔF_{out}	Resolução da frequência de saída
$\theta(t)$	Fase instantânea

SUMÁRIO

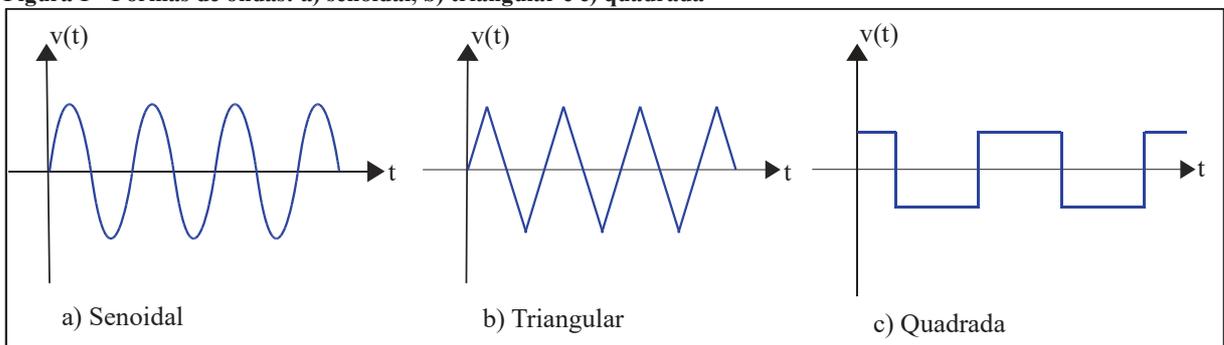
1.0 INTRODUÇÃO.....	16
1.1 OBJETIVOS	18
1.2 JUSTIFICATIVA	18
2.0 FUNDAMENTAÇÃO TEÓRICA	19
2.1 GERADORES DE ONDAS ARBITRÁRIAS.....	19
2.2 SÍNTESE DIGITAL DIRETA.....	20
2.3 MODULAÇÃO EM GERADORES DE ONDAS	23
2.4 FPGAs.....	27
2.5 ESTÁGIO DE SAÍDA DE AWGS.....	28
2.5.1 Conversão Digital para Analógico	29
2.5.2 Amplificador de Ganho Variável	31
3.0 MATERIAIS E MÉTODOS.....	34
3.1 MATERIAIS.....	34
3.1.1 Mercurio IV.....	34
3.1.2 FTDI – FT245R.....	36
3.1.3 DAC7821	36
3.1.4 OPA727.....	38
3.1.5 AD633	39
3.2 METODOLOGIA	39
3.2.1 Softwares de Configurações.....	40
3.2.2 Síntese de Hardware.....	42
3.2.3 Estágio de Saída	46
3.2.3.1 Conversor Digital para Analógico.....	46
4.0 RESULTADOS	48
4.1 ESTÁGIO DE SAÍDA	48
4.1.1 Amplificador de Ganho Variável	48
4.2 SINAIS SINTETIZADOS E MODULADOS	50
4.2.1 Sinais Senoidais.....	51
4.2.2 Sinais Triangulares.....	52
4.2.3 Sinais Quadrados.....	53

4.2.4 Sinais Arbitrários.....	54
4.2.5 Modulação AM	55
4.2.6 Modulação FM	56
4.2.7 Modulações Digitais.....	57
4.3 ANÁLISE E DISCUSSÕES.....	58
5.0 CONCLUSÃO.....	59
5.1 TRABALHOS FUTUROS	59
REFERÊNCIAS	60

1.0 INTRODUÇÃO

Um gerador de funções é um equipamento utilizado para geração de sinais elétricos que podem assumir, comumente, formas senoidais, triangulares ou quadradas. Essas formas de ondas podem ser ajustadas através de parâmetros como: amplitude, frequência e *offset* que são definidos pelo usuário. A Figura 1 apresenta as formas de ondas: a) senoidal, b) triangular e c) quadrada geradas por este equipamento.

Figura 1 - Formas de ondas: a) senoidal, b) triangular e c) quadrada



Fonte: Autoria própria.

É usualmente empregado em testes de circuitos elétricos e eletrônicos para inserir um sinal no circuito em estudo. Para esta análise, osciloscópios também são comumente empregados, podendo ser utilizados outros instrumentos, como, por exemplo, um analisador de espectro. A Figura 2 apresenta o gerador 4040DDS da marca BK PRECISION, presente na Universidade Tecnológica Federal do Paraná - Câmpus Toledo, no laboratório de circuitos elétricos do curso de Engenharia Eletrônica.

Figura 2 - Gerador de Funções 4040DDS da marca BK PRECISION



Fonte: BK Precision (2017).

Visto que o gerador de funções comercial possui um número limitado de formas de ondas e de parâmetros para ajustes limitados, o equipamento não serve para aplicações que necessitem de formas de ondas mais complexas, como por exemplo, cardioides e sinais de giroscópio a fibra óptica.

Uma possível solução é o uso de um gerador de ondas arbitrárias (*Arbitrary Waveform Generator* ou AWG), equipamento que disponibiliza uma gama maior de parâmetros a serem definidos pelo usuário na geração do sinal desejado. Com um AWG, imperfeições do sinal como, por exemplo, ruído e variações randômicas no tempo podem ser simuladas de uma maneira simples e controlada (AGILENT TECHNOLOGIES, 2008).

(REICHERT; MORETO, 2017) utilizaram uma matriz de portas lógicas programáveis (*Field-programmable Gate Array* ou FPGA) para o desenvolvimento de um gerador de formas de ondas arbitrárias por meio da técnica de síntese digital direta (*Direct Digital Synthesis* ou DDS). Esta técnica consiste em gerar um sinal de saída, com especificações de fase e frequência controladas a partir de um *clock* de referência. Ademais, também utilizou-se uma malha resistiva do tipo R-2R como forma de converter os dados digitais em sinais analógicos (conversão digital-analógico).

O presente projeto deu continuidade ao trabalho de (REICHERT; MORETO, 2017), através da adição de diversos melhoramentos e a implementação de novas funções ao gerador. Desta forma, este trabalho desenvolveu e incorporou um modulador digital na estrutura de programação do FPGA; reduziu o tamanho da memória necessária para armazenar os sinais digitais, o que otimizou os recursos do FPGA; utilizou um *clock* de referência mais preciso, que foi obtido de um oscilador externo ao FPGA; substituiu-se a malha R-2R incorporando um conversor digital – analógico (*Digital-to-Analog Converter* ou DAC) no estágio de saída do AWG; inseriu-se um amplificador de ganho variável (*Variable Gain Amplifier* ou VGA) para controle de ganho e *offset*; e desenvolveu-se uma interface gráfica em LabVIEW.

Este trabalho de conclusão de curso foi dividido em cinco capítulos, incluindo esta introdução. O Capítulo 2 contém o referencial teórico no qual o trabalho foi embasado. O Capítulo 3 discute os materiais e métodos que foram utilizados na elaboração e construção do dispositivo. O Capítulo 4 expõe os resultados obtidos na implementação do gerador e, por fim, o Capítulo 5 traz a conclusão e as melhorias futuras.

1.1 OBJETIVOS

O objetivo geral do presente projeto foi o desenvolvimento de um gerador de ondas arbitrárias utilizando a técnica DDS programada em um FPGA, contribuindo com o trabalho de (REICHERT; MORETO, 2017) através da adição de novas funções e melhoramentos. Para alcançar o objetivo geral, os seguintes objetivos específicos foram cumpridos:

- Elaboração de técnicas para otimizar a utilização de recursos do FPGA, incluindo um meio de diminuir a memória onde o sinal digital é armazenado;
- Inclusão, na programação do FPGA, de quatro tipos de modulação (AM, FM, ASK e FSK), recursos comuns em AWGs comerciais;
- Desenvolvimento de uma interface gráfica em LabVIEW;
- Desenvolvimento de um estágio de saída com DAC e elaboração de ganho e *offset* (linha base) controlável;
- Analisar o desempenho do protótipo final, com relação as modulações, tanto no tempo quanto na frequência.

1.2 JUSTIFICATIVA

O gerador de funções é um equipamento muito utilizado para testes em circuitos elétricos e eletrônicos para aplicar o sinal no circuito em estudo.

Com a velocidade em que a eletrônica se desenvolve, geradores de ondas comuns não conseguem suprir as necessidades, uma vez que as tecnologias aplicadas no desenvolvimento de circuitos elétricos e eletrônicos necessitam de sinais diferenciados. Com isso, o gerador de ondas arbitrárias é o equipamento que atende essa demanda, uma vez que é capaz de gerar sinais com frequências extremamente baixas, possui maior imunidade a ruídos e apresenta maior precisão quanto a preservação do sinal pré-configurado.

Visto que o gerador de ondas é um equipamento largamente utilizado tanto na vida acadêmica, nas diversas atividades práticas desenvolvidas durante o curso, quanto na vida profissional, na manutenção e teste de novos equipamentos, o desenvolvimento de um gerador de ondas arbitrárias foi o objeto de estudo escolhido, de forma a oferecer uma nova alternativa de equipamento no meio acadêmico, com menor custo e código aberto (*open source*).

2.0 FUNDAMENTAÇÃO TEÓRICA

Este capítulo traz a fundamentação teórica necessária para o desenvolvimento do trabalho proposto, apresentando uma descrição resumida de geradores de onda arbitrários, teoria da síntese digital direta, conceitos de modulação e amplificadores de ganho variável.

2.1 GERADORES DE ONDAS ARBITRÁRIAS

Um gerador de ondas arbitrárias é um equipamento eletrônico digital para geração de sinais elétricos. Ele gera, além das formas de ondas comuns: senoidal, quadrada e triangular, ondas arbitrárias que são dimensionadas através de uma interface gráfica no próprio aparelho ou em uma aplicação de computador. Essas formas de ondas, podem ser configuradas através de parâmetros como: amplitude, frequência e *offset* (nível CC (corrente contínua)) que são definidos pelo usuário.

Dentre suas principais vantagens, com relação à um gerador de ondas padrão, estão a capacidade de gerar sinais com frequências extremamente baixas (da ordem de μHz), maior imunidade a ruídos e maior precisão quanto a preservação do sinal pré-configurado na entrada.

A Figura 3 apresenta o AWG modelo 33210A da *Keysight Technologies*, sendo este um gerador de ondas arbitrárias de 10 MHz (frequência máxima), com interface de USB (*Universal Serial Bus*), GPIB (*General Purpose Interface Bus*) e LAN (*Local Area Network*) inclusas. Também possui modo gráfico para verificação visual das configurações dos sinais. Além disso o gerador de ondas fornece a opção de modulação AM (*Amplitude Modulation*), FM (*Frequency Modulation*) e PWM (*Pulse-Width Modulation*), formas de onda em rampa, triangulares, ruídos, geração de pulsos com borda variável e ondas CC (AGILENT TECHNOLOGIES, 2008).

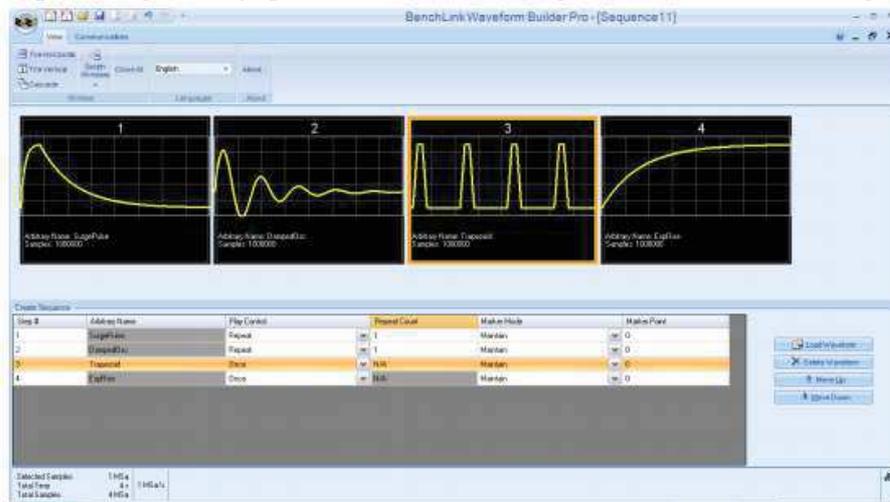
Figura 3 - Gerador de ondas arbitrárias 33210A – *Keysight Technology*



Fonte: *Keysight Technology* (s.d.).

O software Keysight 33503A BenchLink Waveform Builder Pro and Basic Software apresentado na Figura 4 é utilizado para edição de ondas do gerador descrito. Neste aplicativo, o usuário é capaz de ajustar os parâmetros como: amplitude, frequência e *offset*. Dentre suas inúmeras funções, tem-se: criação de ondas arbitrárias através da especificação de pontos ou desenho de linhas, editor de equações para criação de formas de ondas com exatidão polinomial e visualização das características espectrais do sinal a partir da transformada rápida de Fourier (*Fast Fourier Transform* ou FFT) (KEYSIGHT TECHNOLOGIES, 2017).

Figura 4 - Software Keysight 33503A BenchLink Waveform Builder Pro and Basic Software



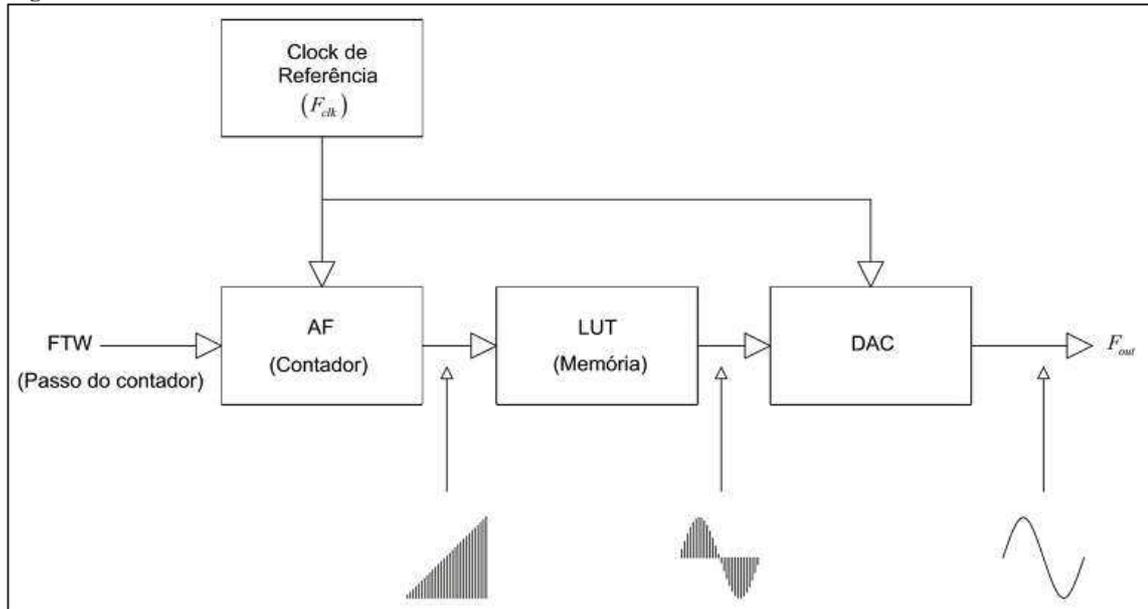
Fonte: Keysight Technology (s.d.).

2.2 SÍNTESE DIGITAL DIRETA

A síntese digital direta é uma técnica que consiste em gerar um sinal de saída analógico, com especificações de fase e frequência controladas a partir de um *clock* de referência, tendo como base um sinal digital. Este *clock* de referência, F_{CLK} , deve ser preciso, geralmente oriundo de um oscilador de cristal. As principais vantagens do uso da técnica DDS são: capacidade de gerar ondas com frequências extremamente baixas, na ordem de μHz , e a possibilidade de programar e reprogramar digitalmente as ondas de saída (CRONIN, 2017).

A Figura 5 apresenta um diagrama de blocos que mostra, resumidamente o funcionamento da técnica DDS.

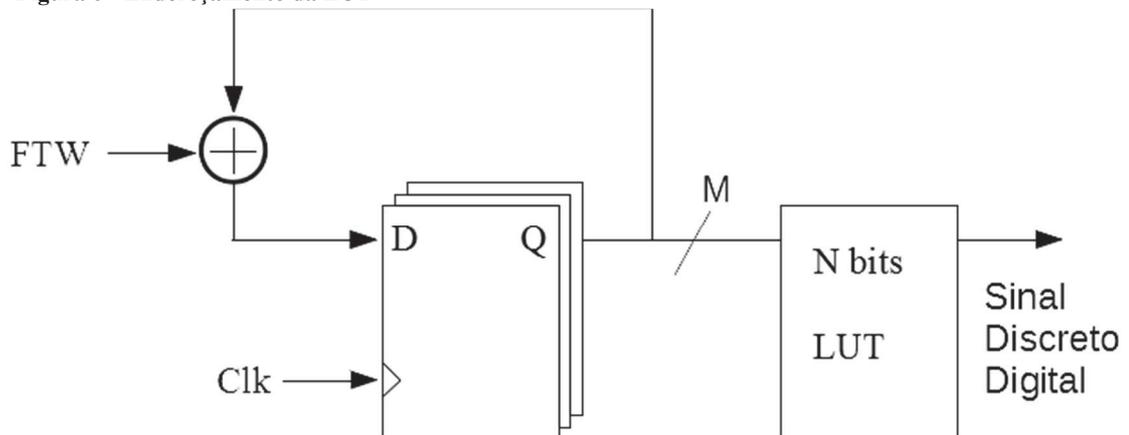
Figura 5 - Estrutura básica da técnica DDS



Fonte: Analog Devices (1999).

A técnica consiste na utilização de um acumulador de fase (designado como AF na Figura 5), de uma memória (*Look-Up table* ou LUT) e de um DAC (*Digital to Analog Converter*). O acumulador de fase funciona como um contador, em que a cada incremento, uma nova amostra do sinal digital é endereçada na memória. O incremento do acumulador de fase pode ser ajustado através de uma palavra de ajuste de frequência (*frequency tuning word* ou FTW), fazendo com que seja possível desviar alguns endereços de memória. Por fim, as amostras armazenadas na memória são enviadas ao conversor digital-analógico, que gera o sinal de saída. A Figura 6 apresenta um diagrama de blocos do endereçamento da LUT.

Figura 6 - Endereçamento da LUT



Fonte: Autoria própria.

Modificando-se a FTW, é possível ajustar a frequência na saída, através da Equação (1):

$$F_{out} = \frac{F_{clk}}{Q}, \quad (1)$$

em que F_{out} é a frequência do sinal gerado, F_{clk} a frequência do *clock* de referência e Q é o número de endereçamentos realizados na memória a cada ciclo de contagem do acumulador de fase. Dessa forma, Q é definido como:

$$Q = \frac{2^M}{FTW}, \quad (2)$$

sendo M o número de bits do acumulador de fase e FTW o incremento do acumulador de fase. Substituindo a Equação (2) na Equação (1), pode se obter a frequência de saída:

$$F_{out} = \frac{F_{clk} \cdot FTW}{2^M}. \quad (3)$$

A resolução da frequência de saída ΔF_{out} , é dada pela Equação:

$$\Delta F_{out} = \frac{F_{clk}}{2^M}. \quad (4)$$

Como exemplo, um AWG que possua $F_{clk} = 50$ MHz, $M = 32$ bits e esteja operando com uma palavra de ajuste $FTW = 5.154$, irá gera um sinal com frequência $F_{out} = (50 \cdot 10^6 \cdot 5.154) / 2^{32} \approx 60,00$ Hz. Esse mesmo gerador possuirá uma resolução de $\Delta F_{out} = \frac{50 \cdot 10^6}{2^{32}} = 11,64$ mHz.

O critério de Nyquist estabelece que a frequência de saída deve ser no máximo menor do que a metade da frequência do *clock* de referência para que não ocorra o fenômeno de serrilhamento (*aliasing*) (OPPENHEIM; SCHAFER, 2012), ou seja:

$$F_{out} < \frac{F_{clk}}{2}. \quad (5)$$

Para satisfazer o critério de Nyquist, a FTW deve ser restringida por:

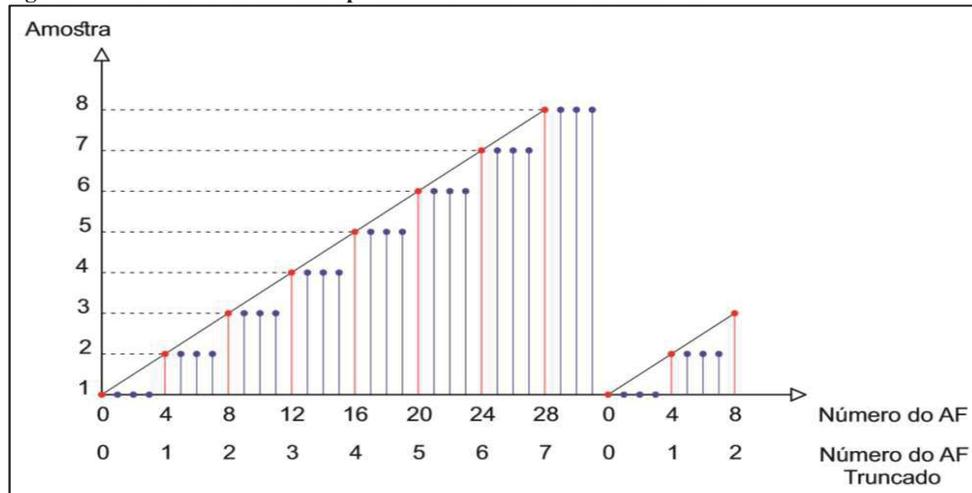
$$p < M - 1, \quad (6)$$

onde p é o número de bits utilizado na FTW e M é o número de bits utilizado no acumulador de fase.

Para o funcionamento da técnica DDS, o módulo do AF deve ser menor ou igual ao número de amostras na LUT, fazendo com que a cada incremento do AF, uma nova amostra seja endereçada. Porém, para se obter frequências baixas, é comum a utilização de um AF com módulo maior que o número de amostras na LUT, considerando apenas os bits mais significativos para endereçar a LUT, truncando o restante, ou seja, fazendo com que a mesma

amostra seja endereçada repetidas vezes. A Figura 7 mostra a repetição de amostras devido ao truncamento de fase em uma onda dente de serra. Neste caso, é possível perceber que o acumulador de fase possui módulo 32 (5 bits) e a LUT possui 8 endereços (3 bits). Dessa forma, dois bits do acumulador de fase foram truncados, fazendo com que cada amostra da LUT seja repetida quatro vezes quando a palavra de ajuste for igual a 1.

Figura 7 - Truncamento de fase aplicado a uma onda dente de serra



Fonte: Reichert e Moreto (2017).

2.3 MODULAÇÃO EM GERADORES DE ONDAS

Modulação é uma técnica empregada em sistemas de comunicações em que as propriedades de um sinal (informação) são modificadas afim de transmiti-lo de um local para o outro através de um meio de transmissão. O sinal a ser transmitido, chamado de portadora, sofre alterações de acordo com um sinal base, chamado de sinal modulador, em que este pode modificar a frequência, amplitude ou a fase da portadora, dependendo do tipo de modulação empregada. Existem diversos tipos de modulações, em que cada uma modifica uma determinada propriedade, possuindo vantagens e desvantagens entre si (HAYKIN, 2001).

As modulações mais comuns em geradores de ondas são: modulação em amplitude (*Amplitude Modulation* ou AM), modulação em frequência (*Frequency Modulation* ou FM), modulação em amplitude por chaveamento (*Amplitude Shift-Keying* ou ASK) e a modulação em frequência por chaveamento (*Frequency Shift-Keying* ou FSK). Geradores de ondas com moduladores podem ser utilizados, por exemplo, para testar um circuito demodulador, e verificar se determinado filtro está funcionando de acordo como esperado.

2.3.1 MODULAÇÃO AM

Na modulação AM, a amplitude da portadora sofre variações de acordo com as variações de amplitude e frequência do sinal modulante, mantendo-se constante a frequência e a fase. Para que não ocorram distorções, a amplitude da portadora deve ser maior do que a amplitude da modulante. A condição ideal é quando o índice de modulação, definido como $m = \frac{V_m}{V_c}$, em que V_m e V_c são os valores de pico da onda modulante e portadora respectivamente, seja unitário, resultando na maior potência de saída do transmissor e na maior tensão de saída no receptor, não havendo distorção (FRENZEL; LOUIS, 2013).

Existem diversos tipos de modulação em amplitude, como AM-DSB (*Amplitude Modulation Double sideband*), AM DSB-SC (*Amplitude Modulation Double sideband-supressed carrier*), AM SSB (*Amplitude Modulation Single sideband*) e AM VSB (*Amplitude Modulation Vestigial sideband*), cada um apresentando vantagens e desvantagens entre si. Neste trabalho, será utilizada a modulação AM DSB, por ser a modulação que necessita de circuitos demoduladores mais simples para recuperar o sinal original.

Considerando o sinal modulante $v_m = V_m \text{sen}(2\pi f_m t)$ e o sinal portadora $v_c = \text{sen}(2\pi f_c t)$ onde v_m e v_c são os valores instantâneos de tensão do sinal modulante e da portadora respectivamente, a modulação em amplitude é dada por:

$$v_{AM} = V_c \text{sen}(2\pi f_c t) + (V_m \text{sen}(2\pi f_m t) \text{sen}(2\pi f_c t)) \quad (7)$$

De outro modo, a modulação em amplitude se resume na multiplicação do sinal modulante pela portadora somado com a portadora.

2.3.2 MODULAÇÃO FM

A modulação em frequência é um tipo de modulação angular em que a frequência instantânea $f_i(t)$ da portadora varia linearmente em função do sinal modulante, mantendo-se constante a amplitude e a fase, ou seja:

$$f_i(t) = f_c + k_f m(t) \quad (8)$$

onde f_c é a frequência da portadora, k_f é a sensibilidade à frequência, dado em Hz/volt , e $m(t)$ o sinal da mensagem que será modulada.

Uma característica importante da modulação em frequência é que ela pode fornecer uma melhor discriminação contra ruído e interferência do que a modulação em amplitude (Haykin & Simon, 2011). Por outro lado, sofre com maior largura de banda de transmissão e maior complexidade do sistema de transmissão/recepção. Um sinal modulado $s(t)$ em fase/frequência é definido por:

$$s(t) = V_c \cos[\theta_i(t)], \quad (9)$$

onde V_c é a amplitude da portadora, e $\theta_i(t)$ a fase instantânea, que varia em relação à mensagem $m(t)$. A frequência instantânea do sinal modulado é dada por $f_i(t) = \frac{1}{2\pi} \frac{d\theta_i(t)}{dt}$.

Para se obter o sinal modulado em frequência, é necessário obter $\theta_i(t)$ a partir da frequência instantânea $f_i(t)$, para isso, integra-se ambos os lados da Equação (8) e multiplica-se por 2π , resultando em:

$$\theta_i(t) = \left[2\pi f_c t + 2\pi k_f \int_0^t m(\tau) d\tau \right]. \quad (10)$$

Substituindo a fase instantânea $\theta_i(t)$ em $s(t)$, encontra-se o sinal modulado em frequência, dado por:

$$s(t) = V_c \cos \left[2\pi f_c t + 2\pi k_f \int_0^t m(\tau) d\tau \right]. \quad (11)$$

2.3.3 MODULAÇÃO ASK

A modulação por chaveamento de amplitude é um tipo de modulação digital que consiste em alterar o nível de amplitude da portadora em função de um sinal de entrada com níveis de amplitude discretos. As principais vantagens da modulação ASK são a facilidade de modular/demodular e a pequena largura de banda necessária para a transmissão, tendo como desvantagem a baixa imunidade a ruídos.

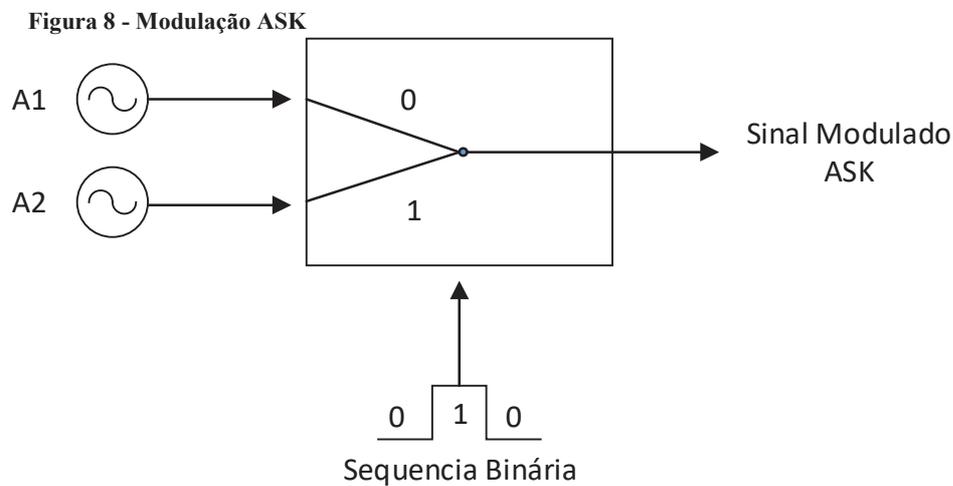
A modulação ASK é indicada apenas em sistemas que apresentem pouca interferência à ruídos devido à sua baixa imunidade a ruídos ou quando o custo é um fator essencial. Em um sistema ASK, o símbolo binário 1 é representado pela transmissão de uma onda portadora senoidal de amplitude e frequência fixas para a duração de bit de T_b segundos, enquanto que o símbolo binário 0 é representado pelo desligamento da portadora por T_b segundos (Haykin & Simon, 2011).

Para o caso de modulação por chaveamento de amplitude binário (*Binary Amplitude Shift Keying* ou BASK) com amplitudes não nulas, o sinal admite dois estados com níveis distintos, chamados de A_1 e A_2 . Portanto, para um sinal temos que:

$$\text{Estado 1: } s_1(t) = A_1 \cos(2\pi f_c t)$$

$$\text{Estado 2: } s_2(t) = A_2 \cos(2\pi f_c t),$$

onde f_c é a frequência da portadora e, A_1 e A_2 a amplitude das portadoras de cada estado. A Figura 8 apresenta o diagrama de blocos da modulação ASK, em que dependendo da entrada binária, uma das ondas senoidais é escolhida.



Fonte: Autoria própria.

2.3.4 MODULAÇÃO FSK

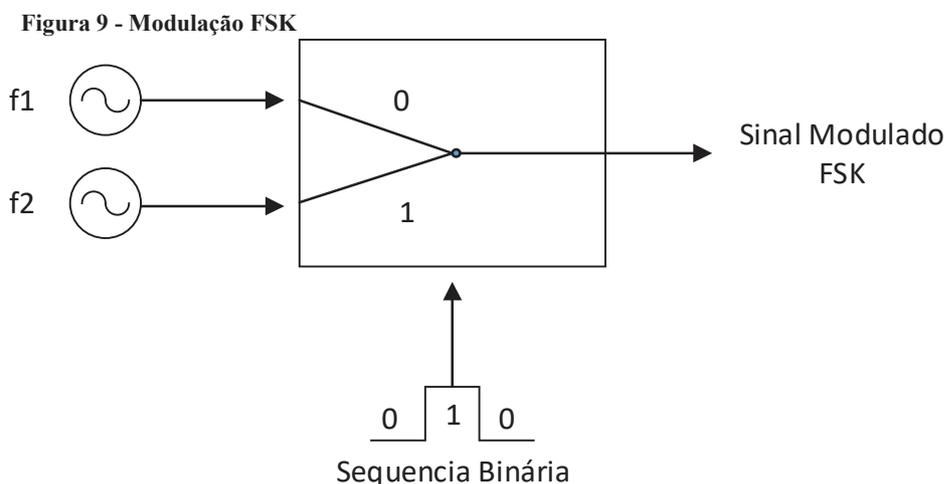
A modulação por chaveamento de frequência (*Frequency Shift Keying* ou FSK) é um tipo de modulação digital que consiste em alterar a frequência da portadora em função de um sinal de entrada com níveis de amplitude discretos. A modulação FSK é mais imune a ruídos externos e apresenta uma menor taxa de erros de transmissão quando comparada com a modulação ASK, por outro lado, a largura de banda aumenta consideravelmente.

Em um sistema FSK, o símbolo binário 1 é representado pela transmissão de uma onda portadora senoidal com frequência f_2 , enquanto que o símbolo binário 0 é representado pela transmissão de uma onda portadora senoidal com frequência f_1 , resultando em:

$$\text{Estado 1: } s_1(t) = A_c \cos(2\pi f_1 t)$$

$$\text{Estado 2: } s_2(t) = A_c \cos(2\pi f_2 t),$$

onde A_c é a amplitude da portadora e, f_1 e f_2 a frequência das portadoras de cada estado. A Figura 9 apresenta o diagrama de blocos da modulação FSK, em que dependendo da entrada binária, uma das ondas senoidais é escolhida.



Fonte: Autoria própria.

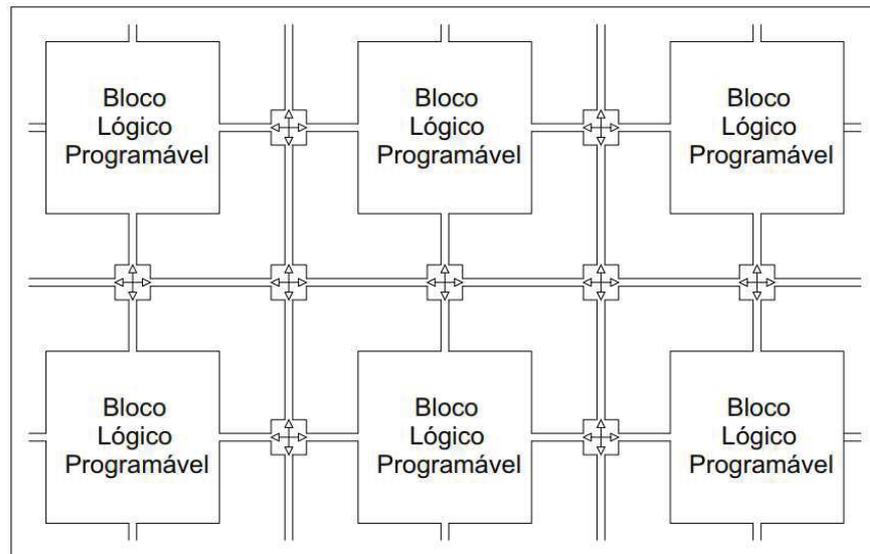
2.4 FPGAs

Os dispositivos lógicos programáveis (*Programmable Logic Devices* ou PLDs) surgiram em meados da década de 1970, com a proposta de serem circuitos combinacionais lógicos que pudessem ser programados. Todavia, diferentemente dos microprocessadores que executam um programa em um *hardware* fixo, a programação dos PLDs se dá no nível do *hardware*. Em outras palavras, os PLDs são chips de uso geral, cujo *hardware* pode ser reconfigurado para atender as especificações particulares (PEDRONI, 2004).

Em meados dos anos 80, a empresa Xilinx criou e comercializou o primeiro FPGA. A arquitetura básica desse tipo de chip consiste em uma matriz de blocos lógicos configuráveis (*Configurable Logic Blocks* ou CLB) conectados através de uma matriz de chaves, apresentado na Figura 10. Estes chips permitem a construção de circuitos sequenciais mais complexos que outros PLDs, pois possuem maior número de flip-flops e elementos lógicos. FPGAs geralmente são utilizados em projetos onde é necessário alto desempenho e velocidade de processamento.

Alguns desses chips podem possuir multiplicadores de *clock*, blocos dedicados como multiplicadores e processadores especializados em processamento digital de sinais (PEDRONI, 2004).

Figura 10 - Arquitetura de um FPGA



Fonte: Pedroni (2004).

A técnica DDS requer uma alta velocidade de processamento e de *hardwares* digitais bem específicos para seu funcionamento. Por esse motivo, é comum a utilização de FPGAs para sintetizar os circuitos necessários e controlar o fluxo de informações em AWGs. Porém, pode ser utilizado um microcontrolador, entretanto, para se obter os mesmos resultados obtidos na utilização de um FPGA, o microcontrolador deve ter um *clock* mais elevado.

2.5 ESTÁGIO DE SAÍDA DE AWGS

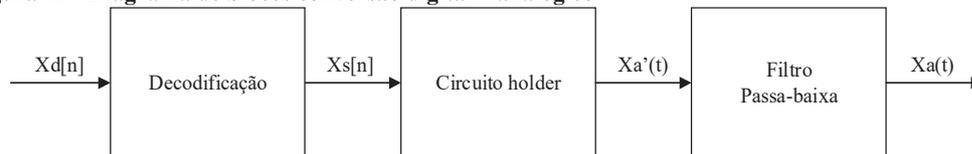
O estágio de saída de um AWG normalmente é composto por um DAC, amplificador de ganho controlável, filtros e um circuito somador de *offset*.

O sinal advindo da técnica DDS está no domínio digital, dessa forma, é necessário a conformação desse sinal para o domínio analógico, e isso é feito através de um DAC. No entanto, a faixa de amplitudes possíveis na saída de um DAC pode ser muito limitada, dado as características do conversor. Dessa forma, é necessário que um amplificador seja adicionado, sendo desejável que esse circuito, de alguma forma, possua o ganho ajustável. O ajuste de *offset* do AWG pode ser obtido através de um circuito somador. Por fim, é comum a construção de um filtro passa baixa com a função de eliminar componentes de alta frequência que surgem durante o processo de amostragem, suavizando o sinal resultante.

2.5.1 Conversão Digital para Analógico

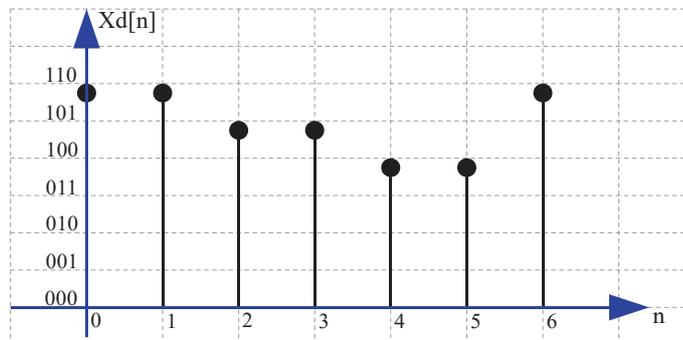
A Figura 11 apresenta o diagrama de blocos do processo de conversão digital para analógico (conversão D/A) dividido em três etapas: decodificação, circuito *holder* e filtro passa-baixa. A entrada $X_d[n]$ é o sinal digital de entrada, onde o índice “n” é um número natural. No contexto de um AWG, construído conforme a técnica DDS, $X_d[n]$ representa uma amostra do sinal que se deseja sintetizar armazenada no endereço “n” da LUT. Dessa forma, $X_d[n]$ está codificado em alguma representação binária, limitado a um número finito de bits, ou seja, a gama de valores possíveis para $X_d[n]$ é quantizada (OPPENHEIM; SCHAFER, 2012). Um exemplo do sinal $X_d[n]$ está representado na Figura 12.

Figura 11 - Diagrama de blocos conversão digital – analógico



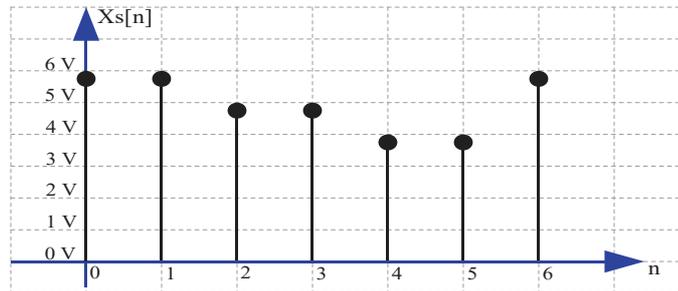
Fonte: Autoria própria.

Figura 12 - Sinal $X_d[n]$



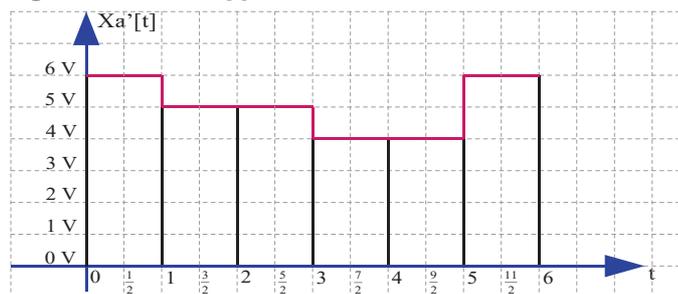
Fonte: Autoria própria.

A etapa de decodificação transforma o sinal $X_d[n]$ em $X_s[n]$, que representa o equivalente em tensão ou corrente do valor binário de $X_d[n]$. Essa decodificação é feita com base em um valor de referência que pode ser tanto de corrente quanto de tensão, dependendo da arquitetura do conversor. Na maioria dos casos, esse valor de referência representa o fundo de escala do conversor. Um exemplo do sinal $X_s[n]$ está representado na Figura 13.

Figura 13 - Sinal $X_s[n]$ 

Fonte: Autoria própria.

Após a decodificação, é necessário um circuito para reter o sinal resultante durante o intervalo entre amostras, trata-se do circuito *holder*, cuja saída é o sinal $X_a'(t)$ indicado na Figura 11, onde $t = T_s \cdot n$, sendo T_s o período de amostragem. Dessa forma, $X_a'(t)$ é contínuo no tempo, mas discreto em amplitude, como pode ser visto na Figura 14.

Figura 14 - Sinal $X_a'[t]$ 

Fonte: Autoria própria.

A última etapa do processo de conversão D/A é a aplicação de um filtro passa-baixas, que tem a função de transformar o sinal $X_a'(t)$ no sinal $X_a(t)$ que é contínuo tanto na amplitude quanto no tempo, assim como pode ser visto na Figura 15. Esse filtro é normalmente especificado com a frequência de corte no limite da banda de Nyquist, e também é conhecido como filtro de suavização (BOYLESTAD; NASHELSKY, 2004).

Figura 15 - Sinal $X_a[t]$ 

Fonte: Autoria própria.

Dessa forma, o sinal obtido após a conversão ($X_a(t)$), observado na Figura 15, possui bordas suaves e está próximo do sinal desejado na forma analógica. É importante ressaltar que, em um AWG, o sinal obtido após a conversão digital para analógico será mais fiel ao que se propôs conforme o número de amostras por período aumenta, proporcionando assim uma melhor aproximação e uma menor perda na fidelidade dos sinais (LATHI, 1998).

DACs comerciais possuem três características importantes: a resolução, a precisão e a frequência máxima, que devem ser observadas na escolha do equipamento. A resolução normalmente é dada pelo número de bits de entrada que o conversor possui, sendo comum dispositivos possuindo de 8 a 16 bits de resolução. A precisão é fornecida pelo fabricante e pode ser analisada através de dois parâmetros principais: o erro de fundo de escala e a não linearidade. A frequência máxima de operação é fornecida pelo fabricante ou pode ser calculada com base no tempo de acomodação (*settling time*), que é o tempo necessário para a saída analógica se estabilizar depois de uma mudança na entrada digital.

Das três etapas da conversão D/A apresentadas no diagrama da Figura 11, apenas as duas primeiras estão presentes em DACs comerciais, sendo opcional a adição do filtro suavizador.

2.5.2 Amplificador de Ganho Variável

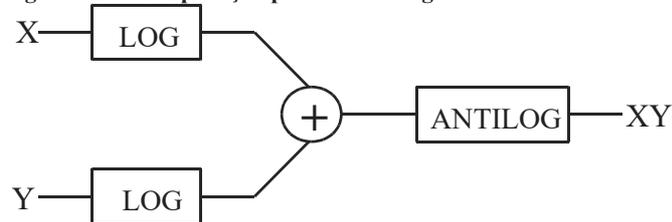
Amplificadores de ganho variável podem ser encontrados em diversas aplicações e são utilizados para maximizar o alcance dinâmico desses sistemas. Eles são importantes no papel de estabilizar a amplitude do sinal de interesse sobre várias condições, providenciando um sinal com amplitude constante no caminho do sinal (CORTES; BAMPI, 2009).

A característica de controle de ganho no projeto do circuito resulta da definição do tipo de amplificador. Uma abordagem que promove a variação do ganho através de uma característica de controle analógica, seja ela um sinal de tensão ou corrente, caracteriza um amplificador de ganho variável – VGA, onde o termo variável está ligado à característica analógica do sinal de controle, que, por definição, pode atingir infinitos pontos de variação do ganho dentro da faixa de variação para a qual foi projetado (MATIAS, 2014).

A variação do ganho destes amplificadores pode ser feita com o uso de um amplificador operacional e de um resistor variável ou com o uso de um circuito com função multiplicador, que são os multiplicadores analógicos. Um multiplicador analógico é um dispositivo que possui duas portas de entrada e uma porta de saída em que, o sinal de saída é o produto entre as duas entradas (ANALOG DEVICES, 2009). Os multiplicadores eletrônicos mais simples utilizam

de amplificadores logarítmico. A razão desse uso está no fato de que o antilog da soma de dois logaritmos é a multiplicação destes números, como pode ser visto na Figura 16.

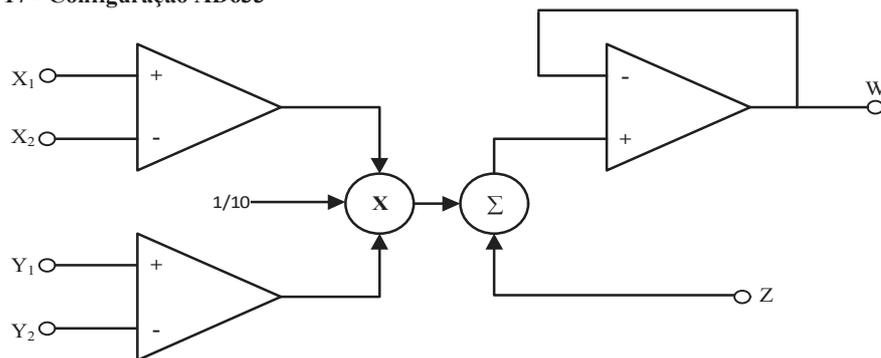
Figura 16 - Multiplicação por soma de logaritmo



Fonte: *Analog Devices* (2009).

Uma forma de desenvolver este amplificador é com o uso de dispositivos analógicos multiplicadores, como é o caso do AD633 cuja configuração é apresentada na Figura 17.

Figura 17 - Configuração AD633



Fonte: *Analog Devices* (2015).

Sua função de transferência é descrita na Equação:

$$W = \frac{(X_1 - X_2)(Y_1 - Y_2)}{10V} + Z, \quad (12)$$

onde: X_1 e X_2 são as entradas diferenciais a serem multiplicadas por Y_1 e Y_2 (também diferenciais); Z é uma entrada somadora, que, no caso do trabalho proposto, funcionará como um controle de *offset* do circuito; e W é a saída obtida como resultado da Equação (12).

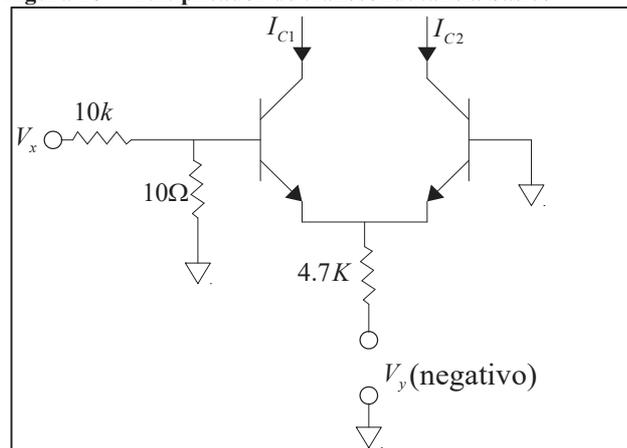
Outra forma é através da “Gilbert Cell”, cuja estrutura esquematizada na Figura 18 foi inventada por Barrie Gilbert no final de 1960. Existe uma relação de linearidade entre a corrente de coletor de um transistor de junção bipolar (TJB) e sua transcondutância. Essa relação pode ser explorada na construção de um multiplicador utilizando um par de TJBs, assim como apresentado na Figura 18. O circuito pode ser equacionado como mostrado na Equação 12 e 13:

$$I_{C1} - I_{C2} = \Delta I_C = \frac{q}{kT} \left(\frac{V_y + V_{be}}{4.7 \times 10^3} \right) \left(\frac{10}{10,010} \right) V_x, \quad (13)$$

$$\Delta I_C = 8.3 \times 10^{-6} (V_y + 0.6) V_x, \quad (14)$$

sendo que a entrada Y tem seu valor de *offset* dado por V_{be} , que muda não linearmente com V_y , a entrada X é não linear como resultado da relação exponencial entre a corrente I_C e a tensão entre emissor e base V_{be} . Algumas desvantagens dessa configuração para a construção do AWG são: a escala de saída varia com a temperatura; existe a necessidade da criação de um circuito para converter a saída $I_{C1} - I_{C2}$ em tensão e é necessário a construção de um circuito somador para adicionar *offset* ao sinal de saída.

Figura 18 - Multiplicador de transcondutância básico



Fonte: Analog Devices (2009).

3.0 MATERIAIS E MÉTODOS

Este capítulo traz os materiais utilizados na implementação do gerador de ondas arbitrárias e a descrição da metodologia utilizada para o desenvolvimento do mesmo.

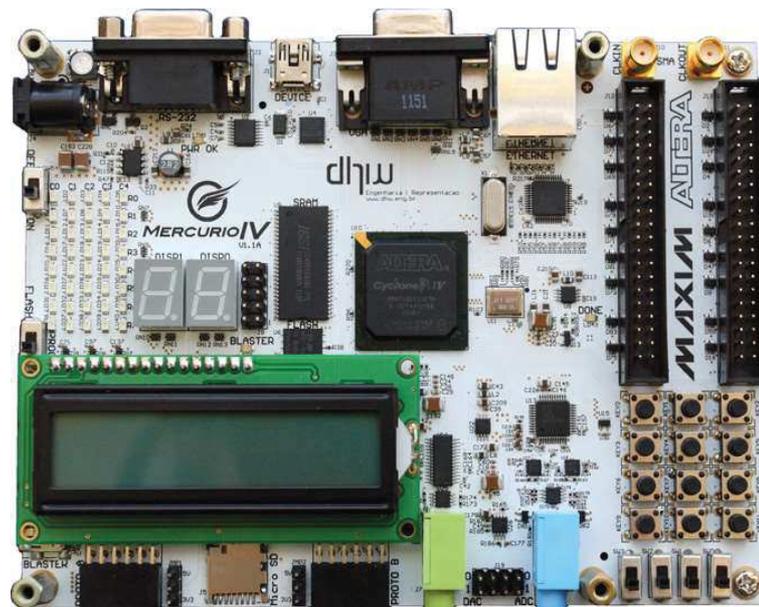
3.1 MATERIAIS

Este tópico traz a especificação dos materiais que foram utilizados no desenvolvimento do projeto assim como suas principais características, sendo estes a placa de desenvolvimento Mercúrio IV, que contém o FPGA e a interface de comunicação USB FT245R, como também do DAC 7821, do amplificador operacional OPA727 e do multiplicador analógico AD633.

3.1.1 Mercurio IV

O kit de desenvolvimento Mercurio® IV, apresentado na Figura 19, pode ser utilizado para o desenvolvimento tanto de produtos quanto de projetos, pois possui os mais diversos periféricos e interfaces para suprir várias necessidades (MACNICA DHW, 2013a).

Figura 19 - Placa de desenvolvimento Mercurio® IV



Fonte: Mansur (2016).

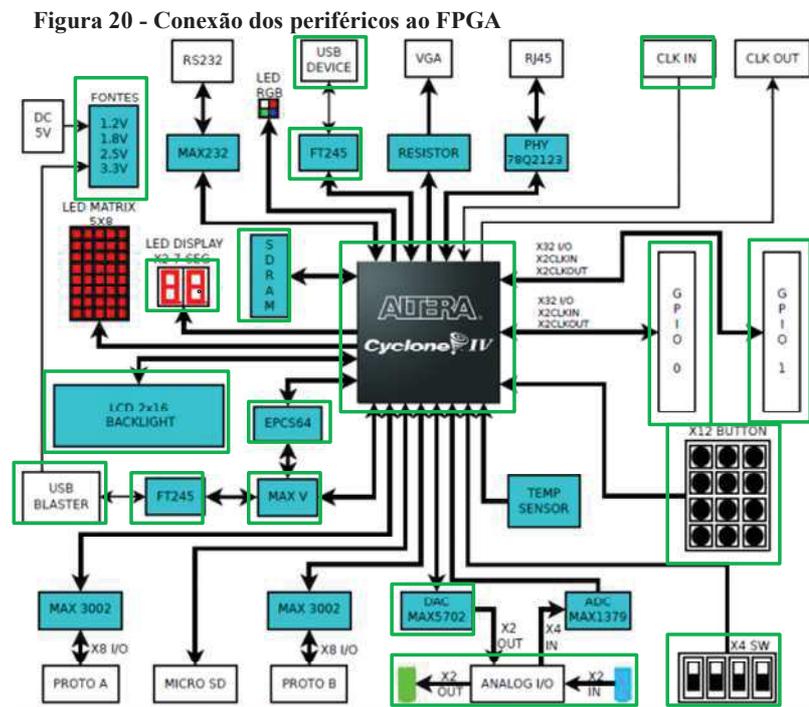
Os itens que foram utilizados com uma breve explicação estão listados a seguir:

- FPGA Cyclone® IV – Utilizado para sintetizar os circuitos necessários;
- Oscilador de 50 MHz – Utilizado como sinal de *clock* de referência;

- GPIOs – Para a comunicação entre a placa Mercurio[®] IV e o circuito analógico desenvolvido;
- USB – Para a comunicação entre a placa Mercurio[®] IV e o software desenvolvido no PC;

A placa Mercurio[®] IV possui o FPGA da família Cyclone[®] IV, desenvolvida pela ALTERA[®], que traz como benefício o baixo custo, consumo de potência reduzida e transceptores integrados (INTEL FPGA, 2017). Esse FPGA permite o desenvolvimento de um processador embarcado, conhecido como NIOS[®] II, que possibilita, com maior facilidade, a implementação de protocolos de comunicação entre periféricos externos à placa com o FPGA.

Para facilitar no desenvolvimento de diferentes sistemas, todos os subsistemas são interligados pelo FPGA Cyclone[®] IV. Assim, o desenvolvedor pode configurar o FPGA para apenas controlar os periféricos necessários, sem se preocupar com os demais itens (MACNICA DHW, 2013a). A Figura 20 apresenta as conexões dos periféricos da placa com o FPGA e em destaque estão os periféricos que foram utilizados.

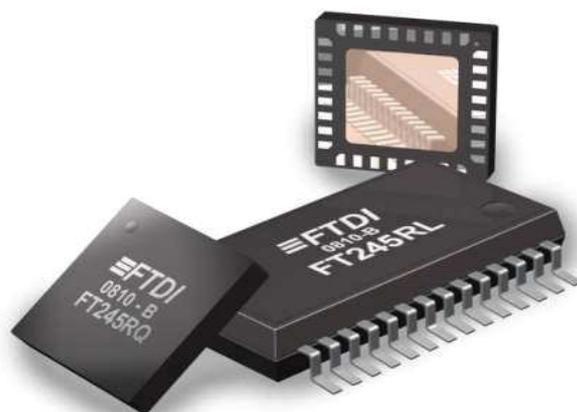


Fonte: Adaptado de Macnica DHW (2017).

3.1.2 FTDI – FT245R

O FT245R é uma interface utilizada para transmissão de dados bidirecional USB para FIFO (First-in-First-Out). Possui uma taxa de transmissão de até 1 Mbyte/s, com buffer de 128 bytes de recepção e 256 bytes de transmissão, utilizando a tecnologia de buffer suavizador, que permite altas taxas de transferências de dados. A Figura 21 apresenta o chip FT245R, contido na placa de desenvolvimento Mercurio® IV. O FT245R possui diversas aplicações como USB de instrumentação, USB de controle industrial, USB áudio e transferência de vídeo de baixa banda de passagem dentre inúmeras outras (FUTURE TECHNOLOGY DEVICES INTERNATIONAL, 2010).

Figura 21 - Chip FT245R

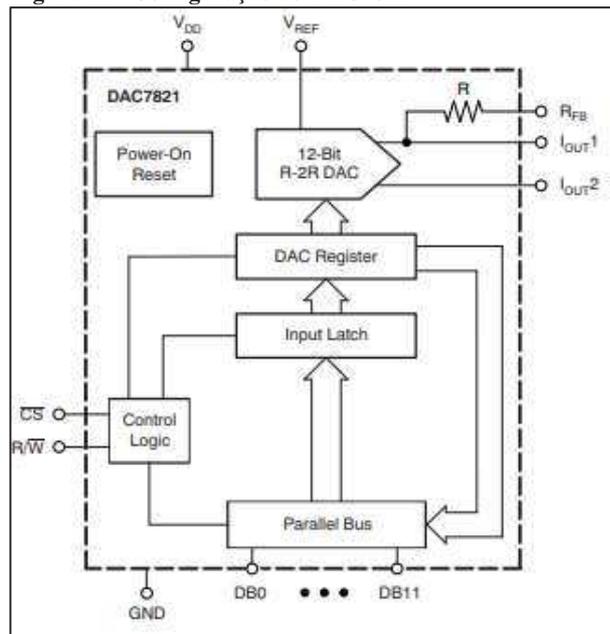


Fonte: Future Technology Devices International (2010).

3.1.3 DAC7821

O DAC7821 é um conversor digital para analógico CMOS (*Complementary Metal Oxide Semiconductor*) de 12 bits com saída em corrente. Este dispositivo opera com uma alimentação de 2,5 V a 5,5 V. O DAC funciona com uma interface paralela rápida e possui um pino R/\bar{W} onde o usuário pode selecionar se deseja fazer a leitura ou escrita dos dados, ou seja, é possível fazer a leitura dos dados previamente enviados através do DAC *Register*, indicado na Figura 21. O dado armazenado neste registrador representa o valor que será convertido em corrente.

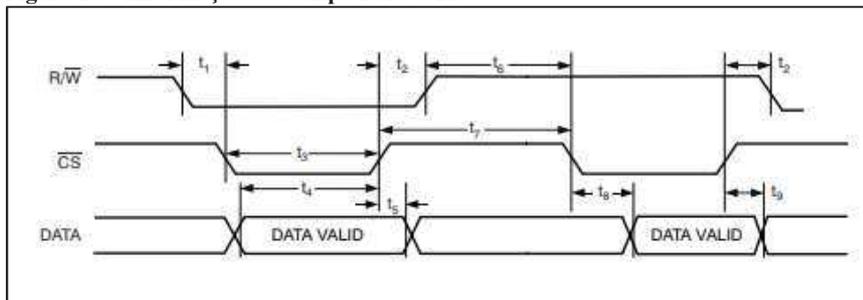
Figura 22 - Configuração do DAC7821.



Fonte: Datasheet DAC7821.

A Figura 23 apresenta o diagrama de tempo do DAC7821, onde note-se a relação do pino R/\bar{W} e do \bar{CS} com os dados a serem lidos, sendo que os dados somente serão lidos após o sinal de borda de descida do pino \bar{CS} seja registrado.

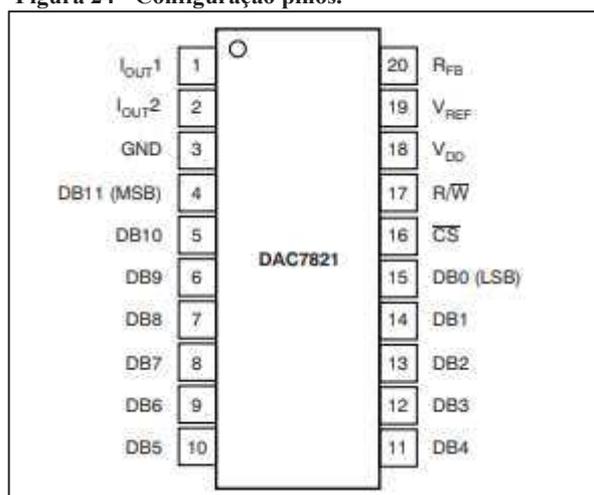
Figura 23 - Informações de tempo DAC7821.



Fonte: Datasheet DAC7821.

A Figura 24 apresenta a relação dos pinos do DAC7821 com sua nomenclatura e o Quadro 1 apresenta as funções de cada um destes pinos, assim como o modo de operação de cada um.

Figura 24 - Configuração pinos.



Fonte: Datasheet DAC7821.

Quadro 1 - Descrição dos pinos do DAC7821.

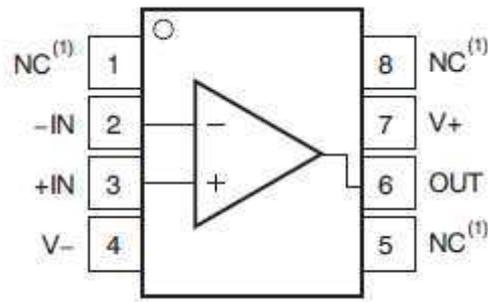
Terminal		Descrição
Nº	Nome	
1	I _{OUT1} ¹	Saída em corrente do DAC.
2	I _{OUT2} ²	DAC terra analógico. Este pino é normalmente conectado ao terra analógico do sistema.
3	GND	Terra.
4-15	DB11 - DB0	Pinos de dados paralelos do 11 ao 0.
16	\overline{CS}	Selação da entrada do chip. Ativo em nível baixo. Utilizado em conjunto com o R/W para carregar dados paralelos a entrada do latch ou ler dados do registrador do DAC. Nível alto desse pino carrega os dados.
17	R/W	Leitura e escrita. Quando em nível baixo, utilizado em conjunto com o pino do \overline{CS} para carregar dados paralelos. Quando em nível alto, utilizado em conjunto com o pino do \overline{CS} para ler de volta os conteúdos do registrador do DAC.
18	V _{DD}	Entrada positiva de tensão. Pode operar de 2,5V a 5,5V.
19	V _{REF}	Tensão de referência do DAC.
20	R _{FB}	Resistor de realimentação do DAC. Estabelece tensão de saída para o DAC através de conexão com a saída de amplificador externo.

Fonte: Adaptado de Datasheet DAC7821.

3.1.4 OPA727

O OPA727 é um amplificador operacional de alta precisão com um *offset* extremamente baixo e largura de banda de 20 MHz, recomendado para a conversão de corrente em tensão para DACs entre outras aplicações. Dessa maneira, foi utilizado para converter a saída em corrente do DAC7821 em tensão.

Figura 25 - OPA727



Fonte: (TEXAS INSTRUMENTS, 2018).

3.1.5 AD633

O AD633 é um multiplicador analógico de baixo custo com quatro quadrantes que inclui alta impedância ($10\text{ M}\Omega$) nas entradas diferenciais X e Y e na entrada somadora Z. Para sua alimentação é necessário uma tensão entre $\pm 8\text{ V}$ e $\pm 18\text{ V}$ sendo que internamente um diodo Zener é responsável pela conformação dessa tensão.

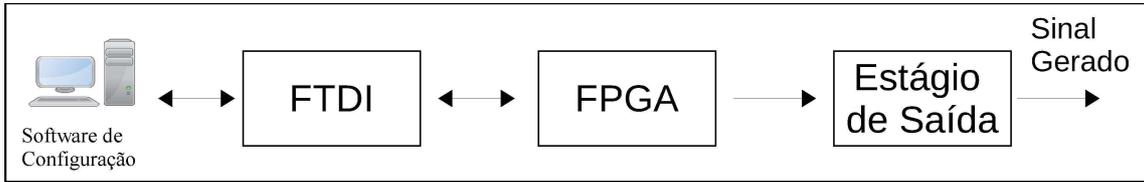
O AD633 é calibrado a laser para garantir uma acurácia de 2% em escala total. A não linearidade das entradas X e Y são tipicamente menores que 0,1% e os ruídos presentes na saída são menores que $100\text{ }\mu\text{V}$ eficazes quando com uma largura de banda entre 10 Hz a 10 kHz. Por possuir construção monolítica, ou seja, uma construção sólida de um bloco só, e calibração a laser se torna um dispositivo estável e confiável.

O AD633 pode ser utilizado para multiplicação, divisão, quadratura, modulação e demodulação, detecção de fase, amplificadores, atenuadores, filtros com controle de tensão (ANALOG DEVICES, 2015).

3.2 METODOLOGIA

Para a construção do gerador de ondas arbitrárias, o proposto trabalho é composto por três etapas distintas, as quais são: o software de configuração em um microcomputador (PC); processamento digital no FPGA presente no kit Mercurio[®] IV; e um estágio de saída, cujo principal componente é um DAC. A Figura 26 apresenta o digrama de blocos geral do sistema. Nessa figura, o bloco intitulado FTDI é responsável pela comunicação entre o PC e o FPGA.

Figura 26 - Diagrama de blocos geral do sistema



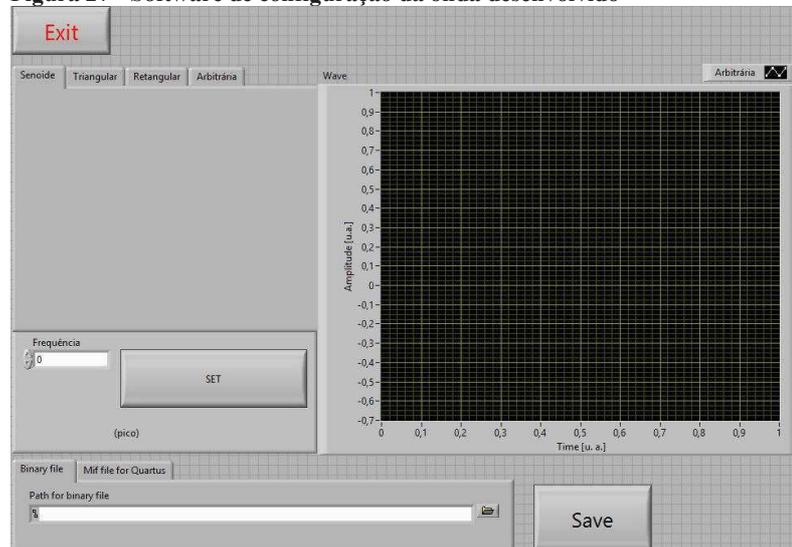
Fonte. Autoria Própria.

3.2.1 Softwares de Configurações

Foram implementados dois *softwares* através da ferramenta LabVIEW, um programa de engenharia de sistemas criado especificamente para aplicações que envolvam teste, medição e controle (NATIONAL INSTRUMENTS, 2018). Um dos softwares é responsável por gerar o sinal (LUT) enquanto que o outro, por transmitir as informações para o FPGA via FTDI-245R.

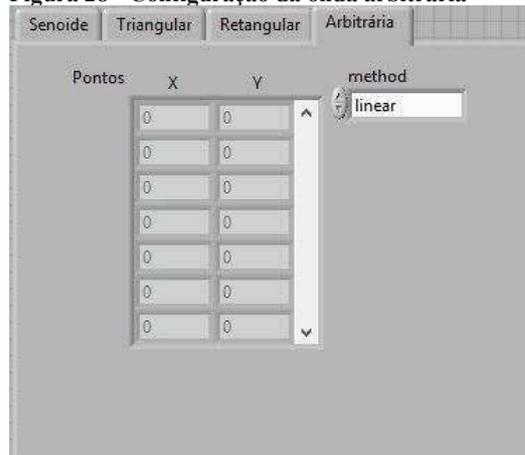
A interface de configuração da onda é apresentada na Figura 27, em que o usuário pode selecionar uma das formas de ondas padrões (senoidal, quadrada ou triangular), ou fornecer pontos para a geração de um sinal arbitrário, conforme apresentado na Figura 28.

Figura 27 - Software de configuração da onda desenvolvido



Fonte. Autoria própria.

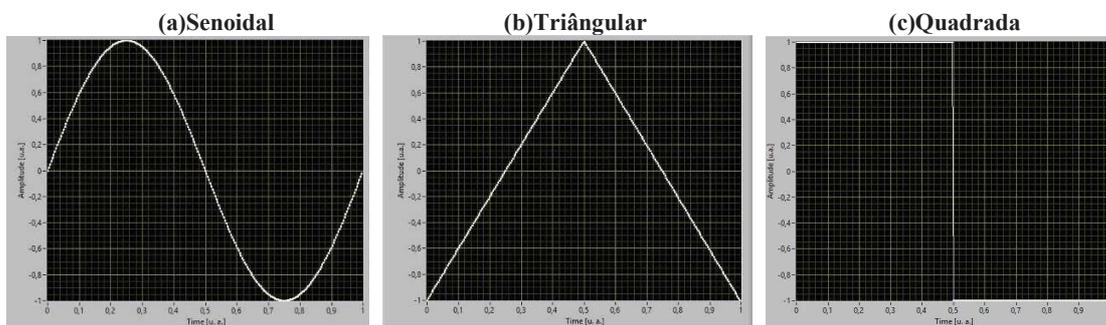
Figura 28 - Configuração da onda arbitrária



Fonte. Autoria própria.

A frequência do sinal pode ser ajustada, além da posição do vértice, para sinais triangulares, e *duty cycle* para sinais quadrados. Um exemplo de cada sinal é apresentado na Figura 29. Para o sinal arbitrário, o usuário pode fornecer até 256 pontos e, caso não fornecer todos os pontos, os restantes são obtidos através de interpolações.

Figura 29 - Forma de ondas padrões



Fonte. Autoria própria.

Por fim, o usuário define se alguma modulação será aplicada, sendo elas as modulações analógicas AM e FM para os sinais modulantes arbitrários, senoidais e triangulares, e as modulações digitais ASK e FSK para os sinais modulantes quadrados, definindo também a frequência da portadora, conforme apresentado na Figura 30.

Figura 30 - Definição das frequências e transmissão dos dados



Fonte. Autoria própria.

Após o término das configurações, os dados são enviados via FTDI-245R para o NIOS II sintetizado no FPGA, conforme o encapsulamento apresentando na Figura 31.

Figura 31 - Encapsulamento de dados desenvolvido

INICIO (4 Bytes)	LUT (1024 Bytes)	FTWA (4 Bytes)	FTWB (4 Bytes)	MODE (4 Bytes)
---------------------	---------------------	-------------------	-------------------	-------------------

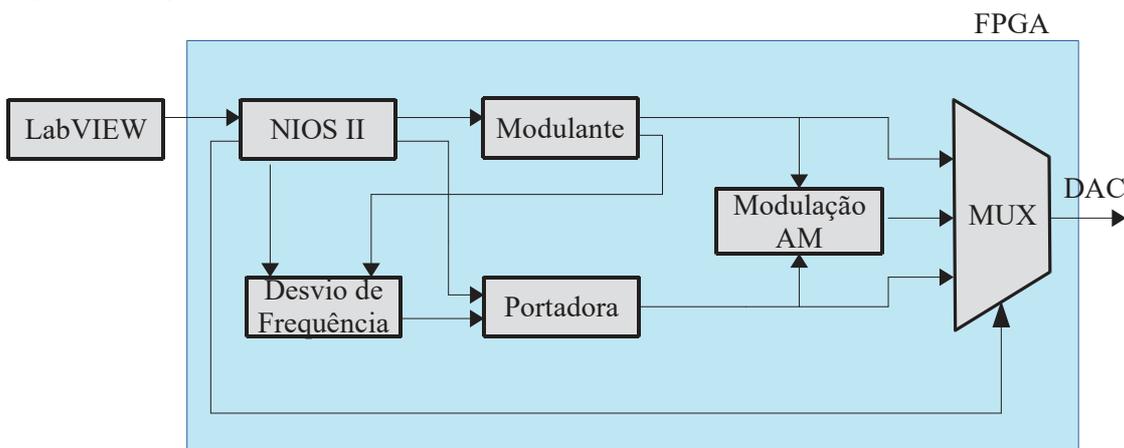
Fonte. Autoria Própria.

Primeiramente é enviado a palavra início que informa ao FPGA que os próximos dados recebidos serão válidos. Em seguida, são transmitidos os 256 pontos (amostras) do sinal para serem gravados na memória (LUT), as palavras de ajuste de frequência da modulante (FTWA) e da portadora (FTWB). Por fim o modo, que define se será aplicada alguma modulação.

3.2.2 Síntese de Hardware

Os circuitos necessários para a confecção do gerador de ondas foram descritos em VHDL (*Very High Speed Integrated Circuits Hardware Description Language*) através do software “Quartus Prime 17.1 Lite Edition”, fornecido gratuitamente pela Altera. A Figura 32 apresenta o diagrama de blocos dos circuitos sintetizados.

Figura 32 - Diagrama de blocos dos circuitos sintetizados



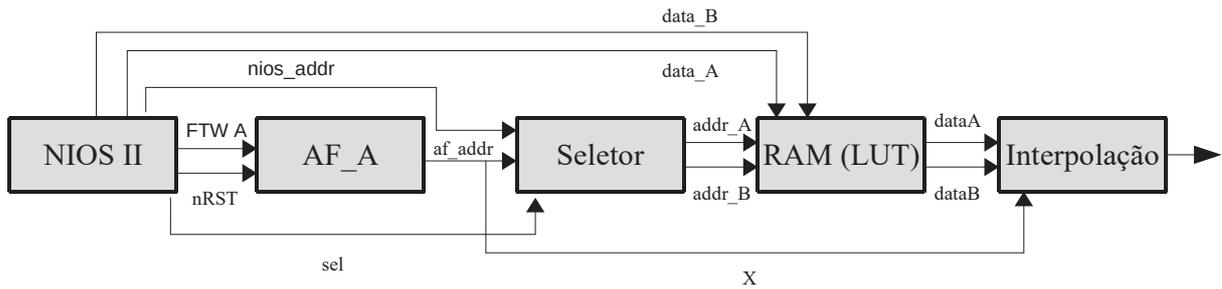
Fonte. Autoria própria.

Primeiramente, foi sintetizado o processador *softcore* NIOS II para receber os dados via FTDI 245R transmitidos pelo PC segundo o encapsulamento apresentando na Figura 31. O processador tem como principal função controlar o fluxo de informações e controlar os circuitos sintetizados no FPGA.

A programação do processador foi realizada em linguagem C (Apêndice A) e para isso dois *buffers* foram criados, um com quatro posições de 8 bits (chamado de *buffer 1*) e outro com 260 posições de 32 bits (*buffer 2*). Quando o *buffer 1* é completado verifica-se se o mesmo possui a palavra de início (0xAA AA AA AA), se sim, os dados do *buffer 1* são passados para o *buffer 2*. De outro modo, as quatro posições do *buffer 1* são salvas em uma única posição do *buffer 2*. Esse procedimento é repetido até que o *buffer 2* seja completado. Quando isso ocorre, as informações são gravadas na LUT, nas palavras de ajuste de frequência (FTWA e FTWB) e no modo. Por fim, os *buffers* são zerados para que novos dados possam ser recebidos.

O bloco modulante (apresentado na Figura 32) é responsável por gerar o sinal da onda desejada. O mesmo é composto por um acumulador de fases, de um circuito seletor, de uma memória RAM (*Random Access Memory*) de dois canais e de um circuito interpolador. A memória de dois canais possibilita gravações e leituras simultâneas em dois endereços no mesmo tempo, resultando em um menor tempo de gravação/leitura. O diagrama completo do bloco modulante é exibido na Figura 33.

Figura 33 - Diagrama de blocos da onda modulante



Fonte. Autoria própria.

O acumulador de fases (Apêndice B), definido como AF_A na Figura 33, funciona como um contador em que a cada borda de subida do sinal de *clock* a contagem é incrementada com o valor da palavra de ajuste de frequência (FTWA). Caso o *reset* seja acionado ($nRST = '0'$), a contagem é zerada. A palavra de ajuste de frequência possui 32 bits e a saída do AF_A possui 16 bits, dessa maneira, os bits menos significativos da contagem são truncados, assim como explicado na seção 2.2.

A saída do acumulador de fases é a entrada para os circuitos seletor (sinal *af_addr*) e interpolador (sinal *X*) apresentados na Figura 33. Dos 16 bits de saída, os mais significativos são utilizados para endereçar a LUT enquanto que os menos significativos para variar o índice de interpolação. A Figura 34 mostra a relação entre a palavra de ajuste de frequência e a saída do acumulador de fases, apresentando os bits truncados, os bits do índice de interpolação e os bits de endereçamento da LUT.

Figura 34 - Saída do acumulador de fases

Endereçamento LUT	Índice de interpolação	Bits truncados
Bits 31...24	Bits 23...16	Bits 15...0

Fonte. Autoria própria.

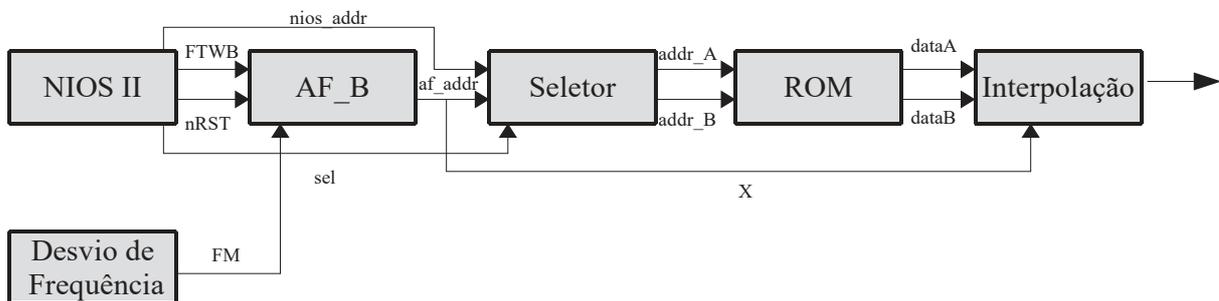
O circuito seletor (Apêndice C), apresentado na Figura 35, define se o acumulador de fases ou o processador NIOS II endereçara a LUT. Essa escolha é baseada na entrada de seleção (*sel*). O NIOS II controla o endereçamento toda vez que há necessidade de gravar dados na memória RAM, caso contrário, o acumulador de fase desempenha essa função para ler as informações previamente gravadas na LUT. A saída do seletor são as entradas de

endereçamento da memória RAM (addrA e addrB) sendo que um dos endereços sempre estará uma posição à frente do outro.

Uma memória RAM de 256 palavras de 18 bits foi sintetizada para funcionar como uma LUT em que cada posição da memória retém uma amostra do sinal. Em modo leitura, o acumulador de fases altera os endereços da LUT, que conseqüentemente, modificam as amostras nas saídas da memória. Essas amostras passam por um circuito interpolador (Apêndice D), que realiza a interpolação linear entre elas, suavizando o sinal de saída. O circuito interpolador foi desenvolvido para diminuir o tamanho necessário da LUT, o que otimiza os recursos do FPGA, e suavizar o sinal de saída.

O bloco portadora, apresentado na Figura 32, é responsável pela geração da onda senoidal da portadora. Seu funcionamento é semelhante ao do bloco modulante, apresentando diferenças no acumulador de fase, o qual possui a entrada FM, e no sistema de armazenamento das amostras (que utiliza a memória ROM (*Read only memory*) ao invés da RAM). O diagrama de blocos completo da onda portadora é apresentado da Figura 35.

Figura 35 - Diagrama de blocos da onda portadora



Fonte. Autoria própria.

O acumulador de fases da portadora (Apêndice E), chamado de AF_B na Figura 35, possui uma entrada a mais do que o acumulador de fases da modulante chamado de FM. Essa entrada é o resultado da multiplicação entre a amplitude da onda modulante com a palavra de ajuste de frequência da onda portadora, calculado previamente pelo bloco desvio de frequência (Apêndice F). A cada borda de subida do *clock*, o contador é incrementado com o valor da palavra de ajuste de frequência (sinal FTWB) e com o resultado do desvio de frequência (sinal FM).

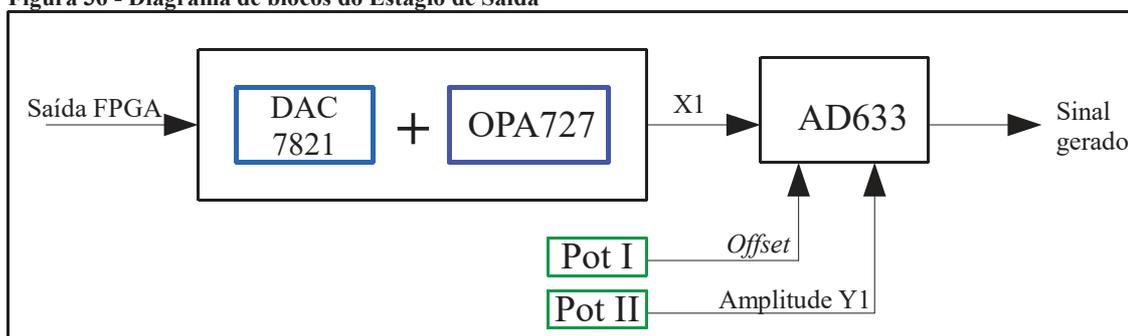
Uma memória ROM foi sintetizada para funcionar como LUT da onda portadora. A mesma é inicializada com as informações de uma onda senoidal. Por fim, o bloco modulação AM (Apêndice G) realiza a modulação em amplitude entre a modulante e a portadora, multiplicando as amplitudes dos dois sinais e somando o resultado com a portadora, assim como descrito na Equação 7. O bloco MUX (Apêndice H) funciona como um multiplexador em que

dependendo da entrada de seleção proveniente do NIOS II, uma das três entradas será a saída, sendo elas as ondas modulante, a modulação AM ou a modulação FM.

3.2.3 Estágio de Saída

A Figura 36 apresenta o diagrama de blocos do estágio de saída do gerador de ondas arbitrárias. Um DAC foi utilizado para converter o sinal sintetizado pelo FPGA. A amplitude da onda de saída do DAC é sempre constante e *offset* nulo, de forma a utilizar todo o fundo de escala do dispositivo. Como a saída do DAC7821 é em corrente, foi necessário a utilização do amplificador operacional OPA727 para a conversão desse sinal para tensão. A saída do amplificador operacional é a entrada X1 do AD633, que é um amplificador multiplicador com circuito somador acoplado, sendo o responsável por ajustar a amplitude e o nível CC do sinal gerado através de potenciômetros. O potenciômetro chamado POT I é ajustado de acordo com o *offset* desejado, enquanto que o POT II controla a entrada Y1 de acordo com o nível de amplitude requerida, conforme Equação 10.

Figura 36 - Diagrama de blocos do Estágio de Saída



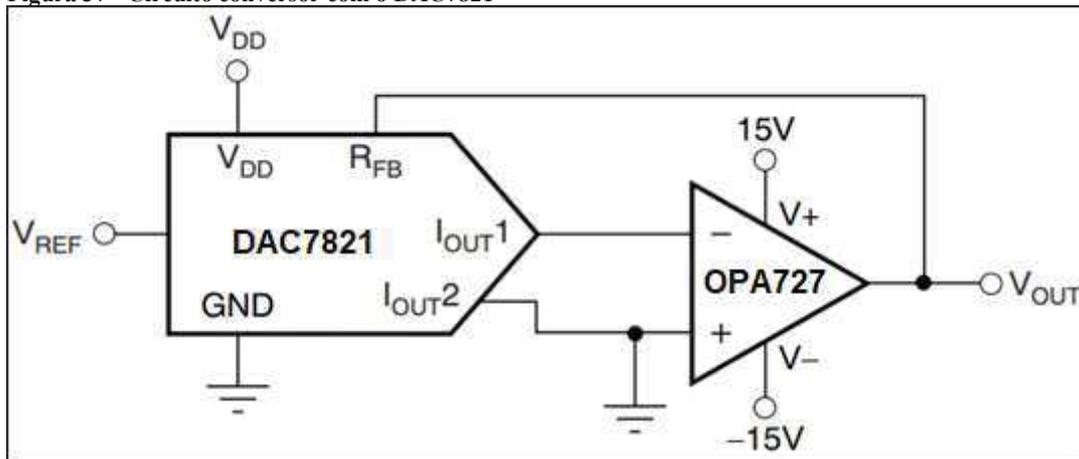
Fonte: Autoria própria.

3.2.3.1 Conversor Digital para Analógico

O circuito utilizado para a conversão digital para analógico foi o recomendado pelo *datasheet* do DAC7821, apresentado na Figura 37, onde verifica-se que a saída em corrente I_{OUT1} e I_{OUT2} do DAC7821 é a entrada de um amplificador operacional que faz a conversão para a tensão de saída V_{OUT} .

O operacional utilizado foi o OPA727 alimentado com +5 e -5 V visto na seção 3.1.4. O pino V_{REF} é a tensão de referência do DAC que no circuito produzido pode ser selecionado por um *jumper* se será +5 e -5 V (APÊNDICE I), V_{DD} é a tensão de alimentação do DAC7821 e R_{FB} é um resistor de *feedback*, que estabelece tensão de saída para o DAC7821.

Figura 37 - Circuito conversor com o DAC7821



Fonte: *Datasheet* DAC7821.

4.0 RESULTADOS

Este capítulo apresenta os resultados obtidos durante os testes do sistema, sendo apresentado separadamente o estágio de saída e os resultados finais coletados do protótipo completo.

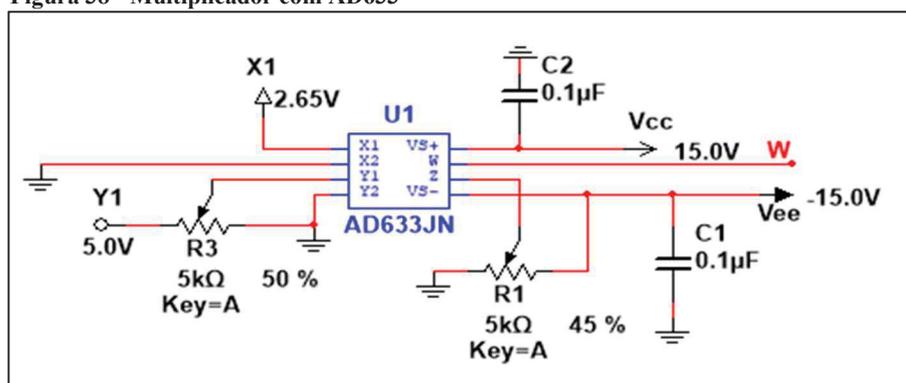
4.1 ESTÁGIO DE SAÍDA

Para o estágio de saída foram confeccionadas duas placas, uma com o DAC e um amplificador operacional, tanto para a conversão do sinal recebido do FPGA como para transformar a corrente de saída do DAC em tensão, e outra com o AD633 para ajuste de amplitude (ganho) e *Offset* (zero), como visto no circuito apresentado na Figura 36.

4.1.1 Amplificador de Ganho Variável

O circuito utilizado na simulação pode ser visto na Figura 38, onde X1 é a saída do DAC, Y1 é a entrada multiplicadora, Z é a entrada de *offset*, Vs+ e Vs- é a alimentação simétrica do *chip* e W é a saída em Volts (V).

Figura 38 - Multiplicador com AD633



Fonte: Autoria própria.

Nota-se no circuito que a entrada de *offset* está conectada à entrada negativa da alimentação, uma vez que o sinal de saída do DAC7821 é apenas positivo, foi necessário o uso de um *offset* negativo para a adequação da excursão total do sinal, e que há capacitores de desacoplamento em paralelo com a alimentação do *chip*, que evitam que oscilações fortes de correntes passem para o resto do circuito.

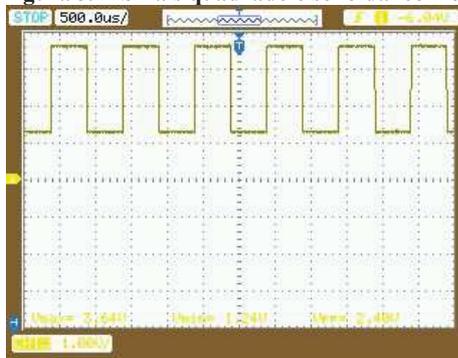
Foram coletados resultados do amplificador multiplicador dispondo de uma entrada na forma quadrada e senoidal, onde a Tabela 1 apresenta os dados e a Figura 39 as imagens retiradas com o uso do osciloscópio modelo DSO1012A da *Agilent Technologies*.

Tabela 1 - Dados dos sinais gerados para teste do circuito amplificador com o DAC7821

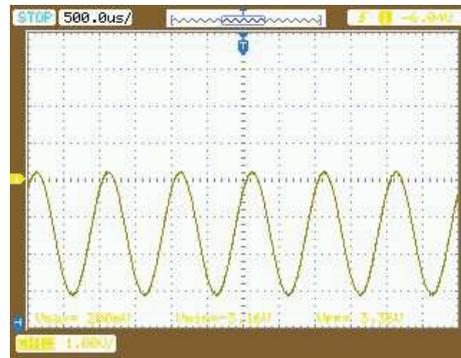
Tipo	X1	Y1	Offset	Vout(pico)
Quadrada	2,65 V	4,25 V	2,50 V	3,64 V
Senoidal	2,65 V	5,60 V	- 1,50 V	0,20 V

Fonte: Autoria própria.

Figura 39 - Sinais quadrado e senoidal com *offset*



a) Onda quadrada 1,0 V/div.

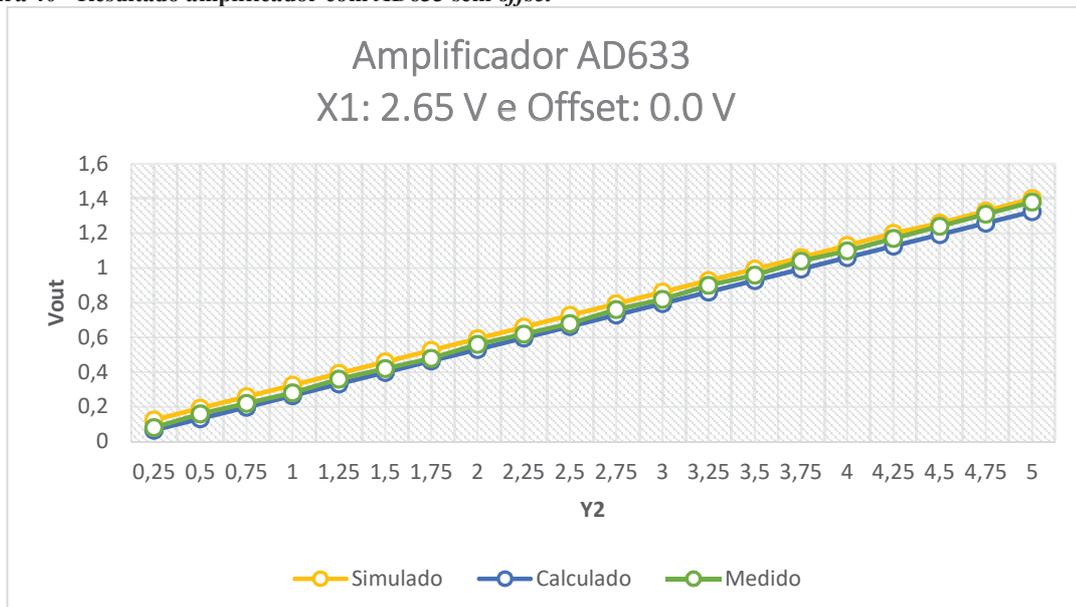


b) Onda senoidal 1,0 V/div.

Fonte: Autoria própria.

A Figura 40 contém os gráficos de resultados sem a adição de *offset*. As três retas no gráfico representam os valores de saída do sinal simulado, calculado e medido .

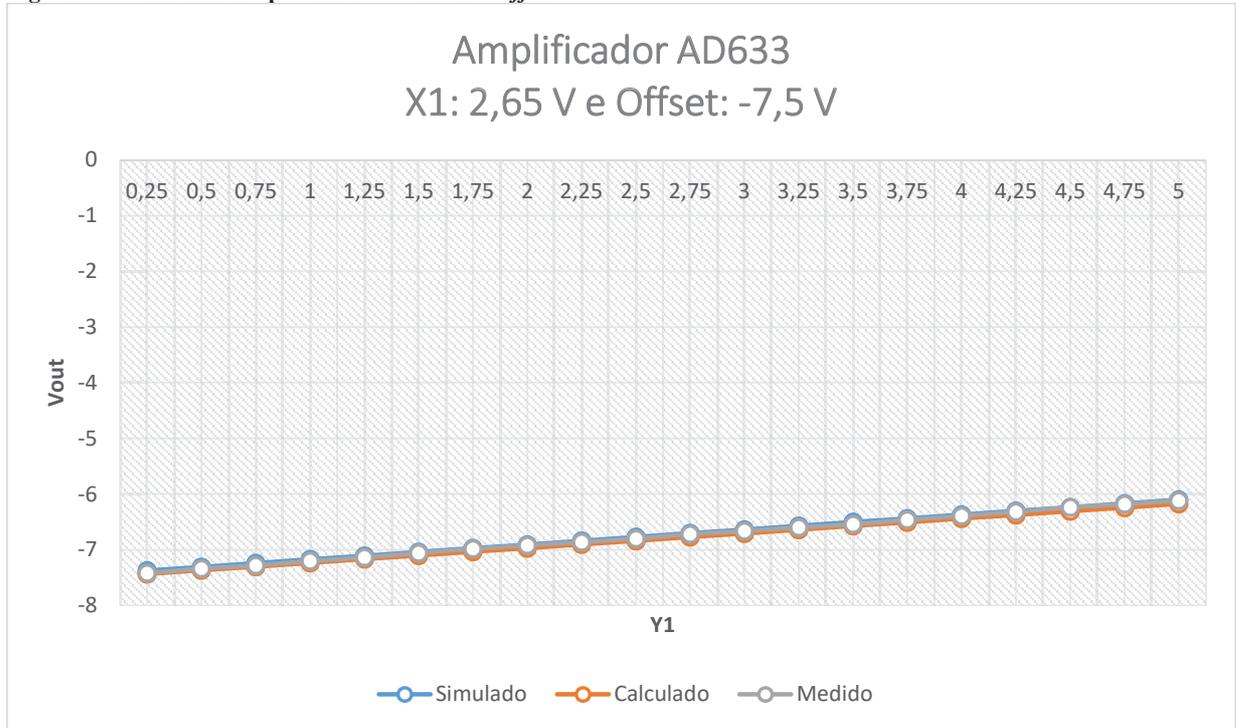
Figura 40 - Resultado amplificador com AD633 sem *offset*



Fonte: Autoria própria.

A Figura 41 apresenta os resultados obtidos para as mesmas entradas com a adição *offset*, obtido a partir do ajuste do POT I, visto na Figura 36, para o valor de -7,50 V.

Figura 41 - Resultado amplificador AD633 com *offset*



Fonte: Autoria própria.

4.2 SINAIS SINTETIZADOS E MODULADOS

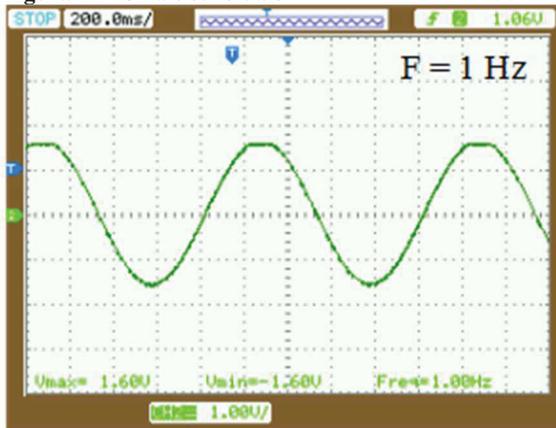
Os sinais foram sintetizados pelo FPGA com uma taxa de amostragem de 5 MHz, o que resultou em uma resolução do gerador de 1,16 mHz, e coletados por meio do osciloscópio. O ganho e *offset* foram ajustados manualmente para que os sinais coletados estivessem centralizados no eixo Y e apresentassem boa visualização.

A fim de se obter uma melhor avaliação das modulações, os resultados também foram simulados, o que possibilitou a visualização da onda modulante e da modulada simultaneamente. A frequência da modulante foi ajustada para 1 kHz e da portadora em 10 kHz para todas as simulações.

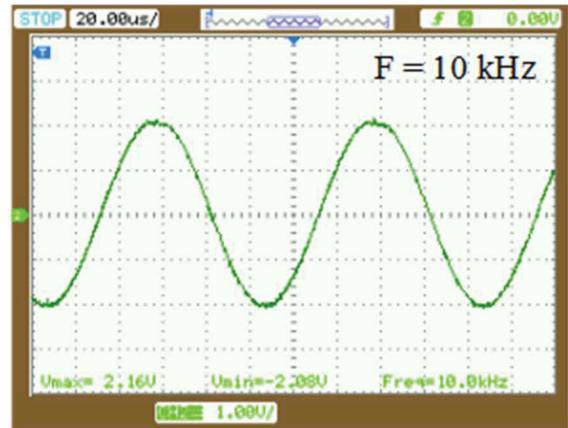
4.2.1 Sinus Senoidais

Alguns sinais senoidais foram gerados utilizando o protótipo do gerador de ondas arbitrárias com frequências variadas (Figura 42), sendo elas 1 Hz, 10 kHz, 25 kHz, 100 kHz, 250 kHz e 350 kHz.

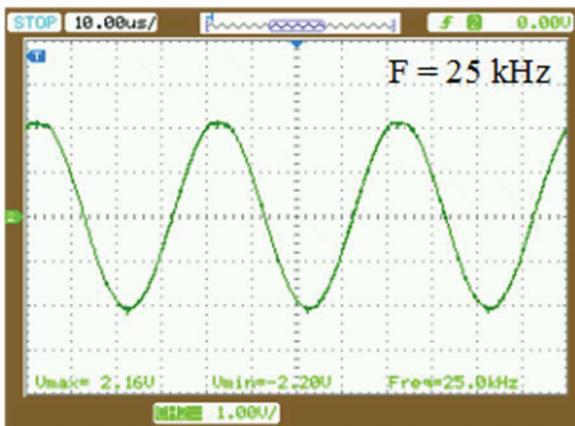
Figura 42 - Sinus senoidais



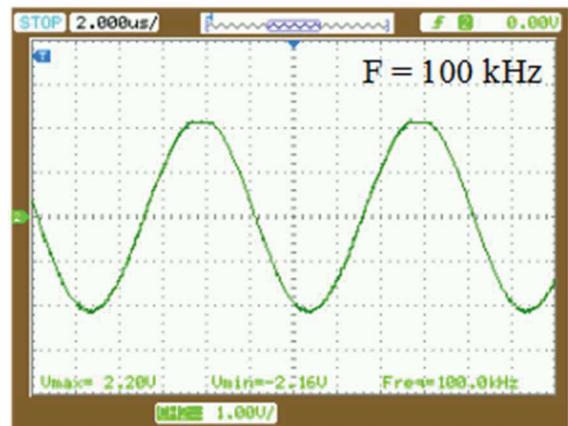
(a) 1 Hz



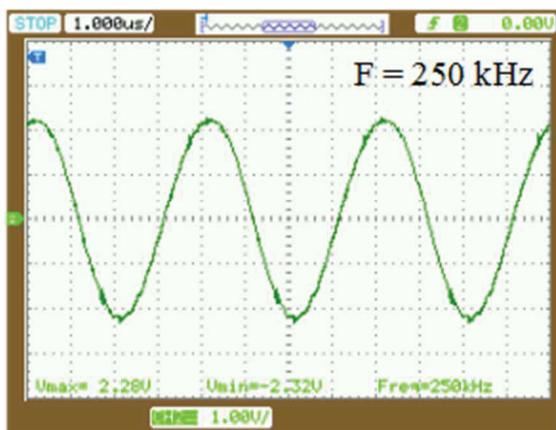
(b) 10 kHz



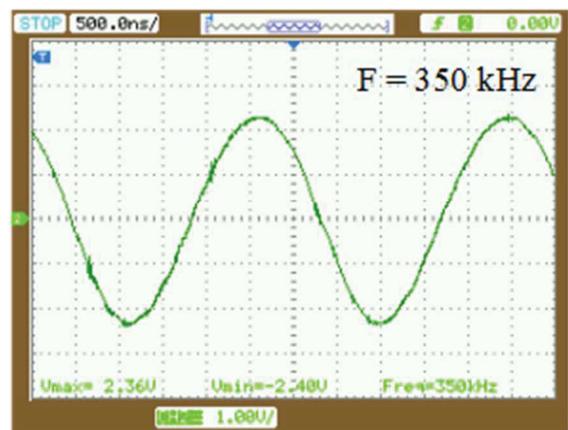
(c) 25 kHz



(d) 100 kHz



(e) 250 kHz



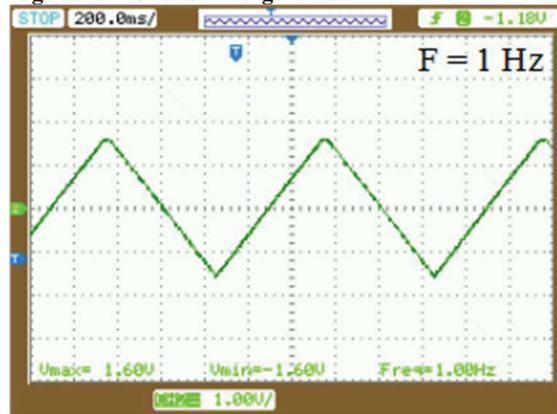
(f) 350 kHz

Fonte. Autoria própria.

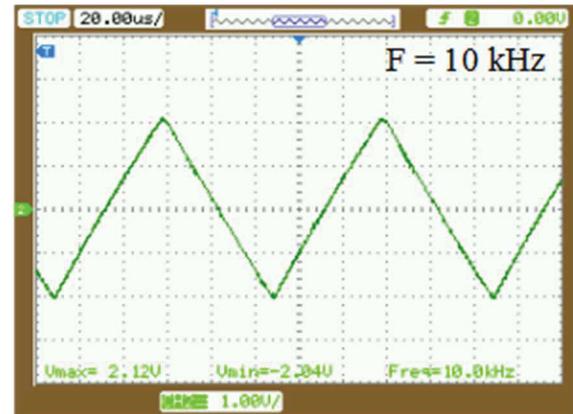
4.2.2 Sinais Triangulares

Alguns sinais triangulares foram gerados utilizando o gerador de ondas arbitrárias com frequências variadas (Figura 43), sendo elas 1 Hz, 10 kHz, 25 kHz, 100 kHz, 250 kHz e 350 kHz.

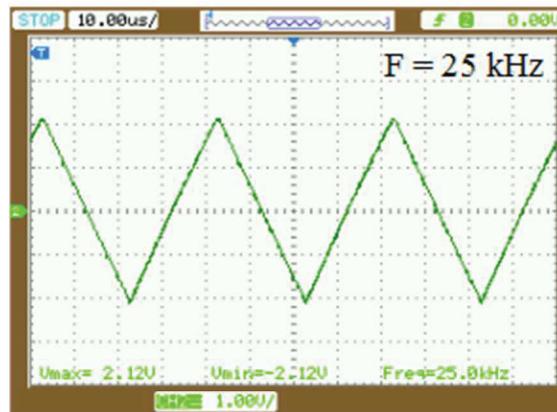
Figura 43 - Ondas Triangulares



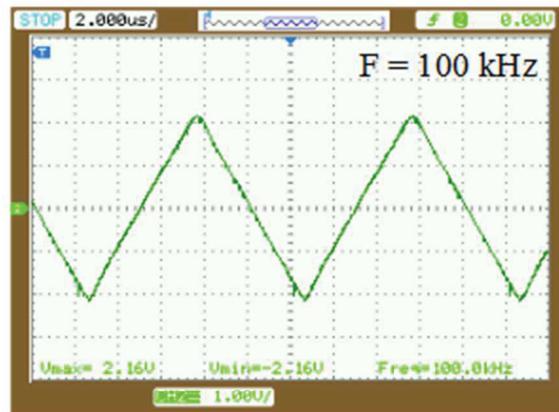
(a) 1 Hz



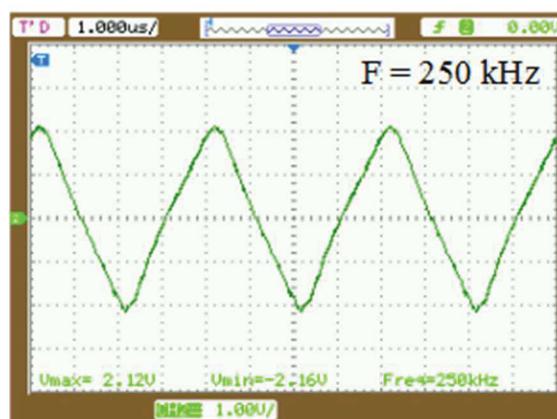
(b) 10 kHz



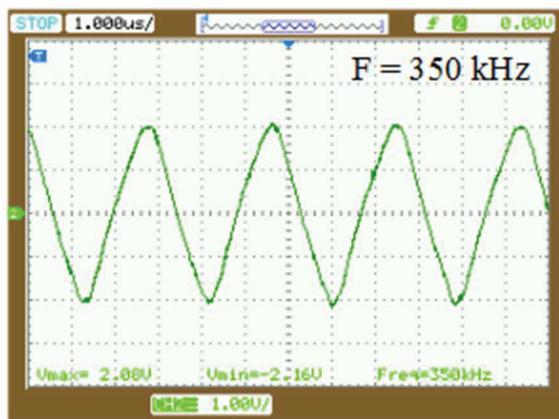
(c) 25 kHz



(d) 100 kHz



(e) 250 kHz



(f) 350 kHz

Fonte. Autoria própria.

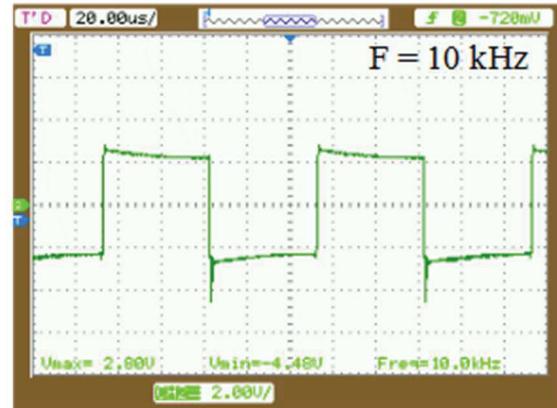
4.2.3 Sinais Quadrados

Alguns sinais quadrados foram gerados utilizando o gerador de ondas arbitrárias com frequências variadas (Figura 44), sendo elas 1 Hz, 10 kHz, 25 kHz, 100 kHz, 250 kHz e 350 kHz.

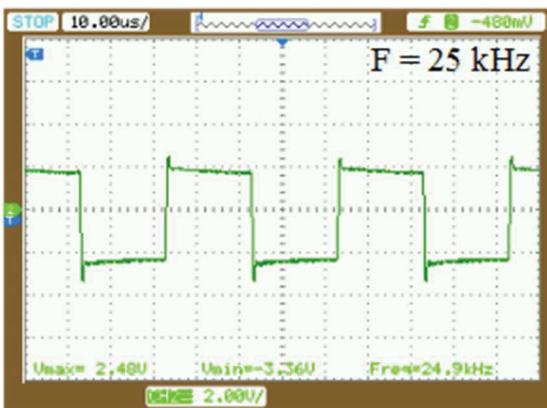
Figura 44 - Ondas quadradas



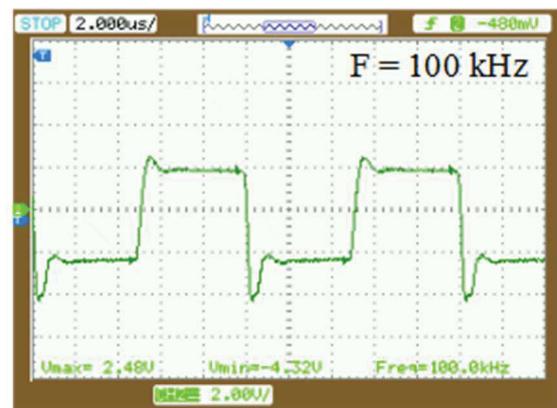
(a) 1 Hz



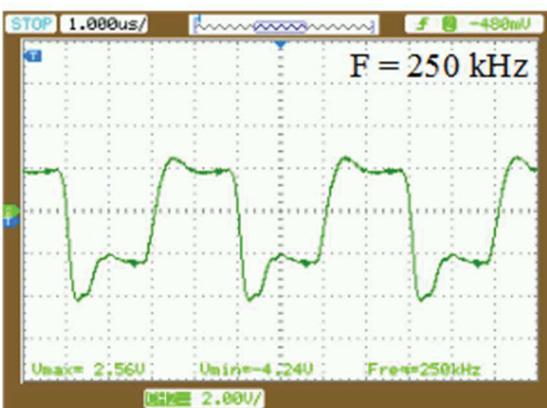
(b) 10 kHz



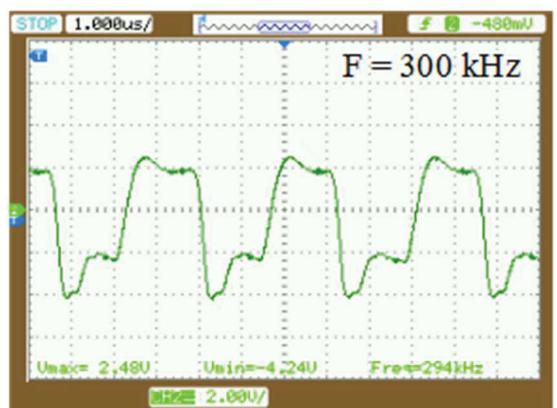
(c) 25 kHz



(d) 100 kHz



(e) 250 kHz



(f) 300 kHz

Fonte. Autoria própria.

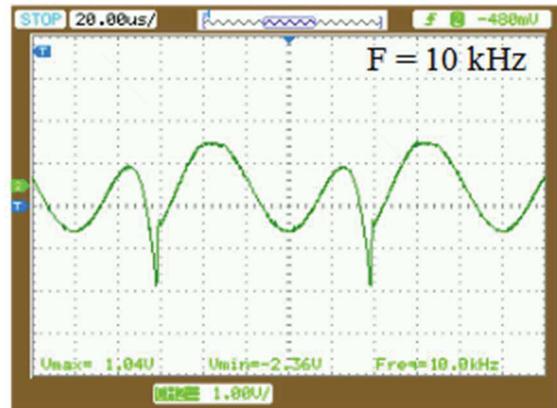
4.2.4 Sinais Arbitrários

Um sinal arbitrário foi gerado utilizando o gerador de ondas arbitrárias com frequências variadas (Figura 42), sendo elas 1 Hz, 10 kHz, 25 kHz, 100 kHz, 250 kHz e 350 kHz.

Figura 45 - Sinais arbitrários



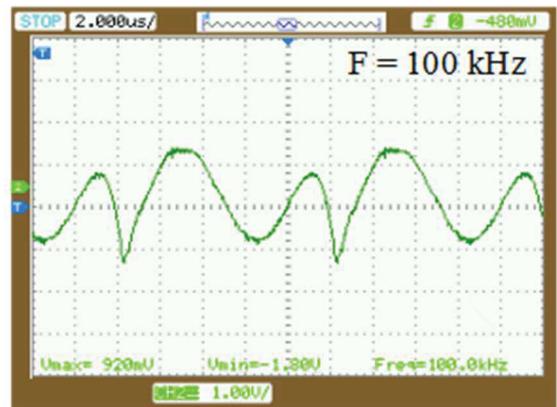
(a) 1 Hz



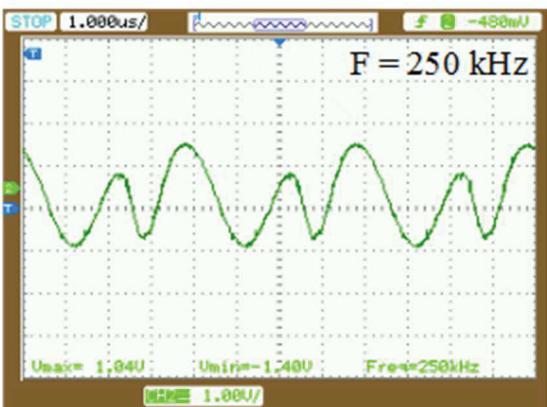
(b) 10 kHz



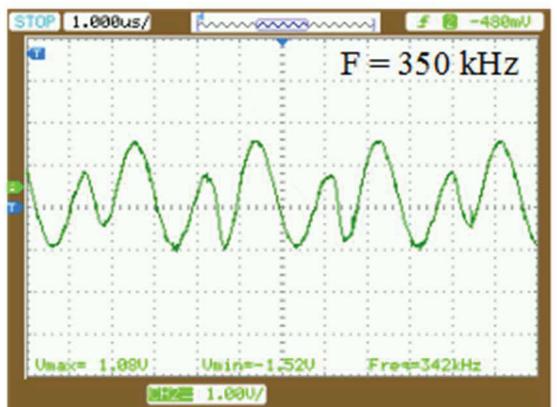
(c) 25 kHz



(d) 100 kHz



(e) 250 kHz



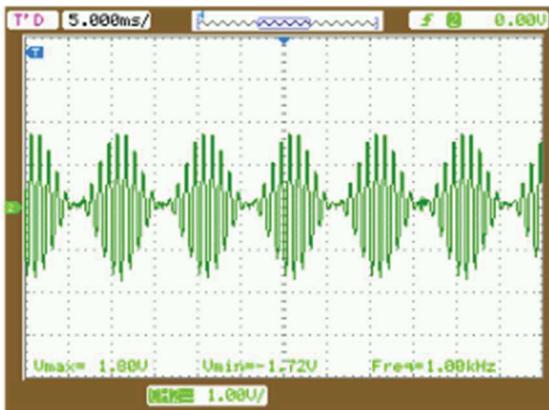
(f) 350 kHz

Fonte. Autoria própria.

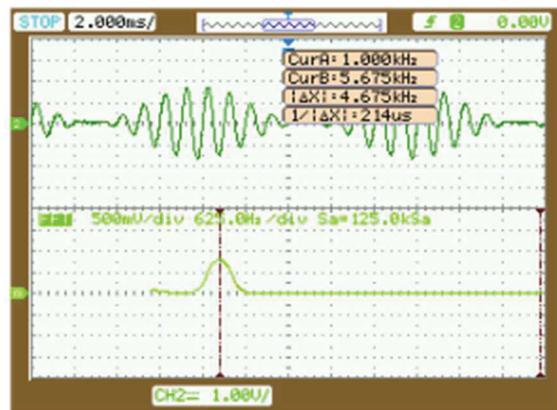
4.2.5 Modulação AM

A modulação em amplitude foi obtida utilizando sinais modulantes senoidais. As frequências utilizadas foram 1 kHz para a modulante e 10 kHz para a portadora (Figura 46 (a) e (b)) e, 10 kHz para a modulante e 100 kHz para a portadora (Figura 46 (c) e (d)). A simulação pode ser visualizada na Figura 47.

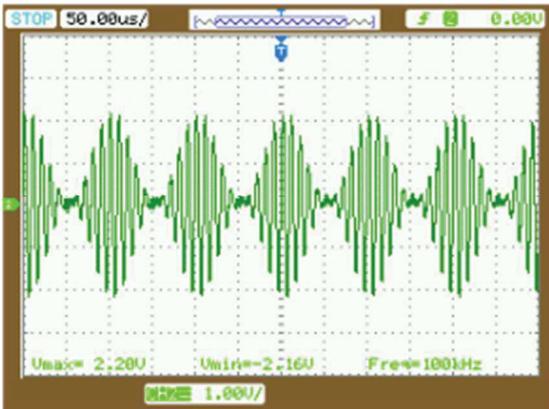
Figura 46 - Modulação em amplitude



(a) frequência modulante = 100 Hz
frequência portadora = 1 kHz



(b) FFT da modulação AM (Figura a)



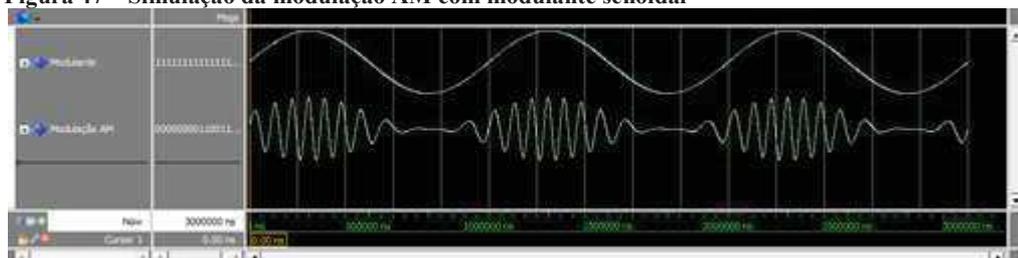
(c) frequência modulante = 1 kHz
frequência portadora = 100 kHz



(d) FFT da modulação AM (Figura c)

Fonte. Autoria própria.

Figura 47 – Simulação da modulação AM com modulante senoidal

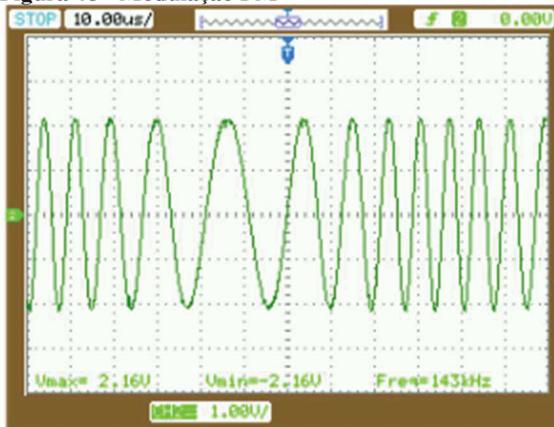


Fonte – Autoria própria.

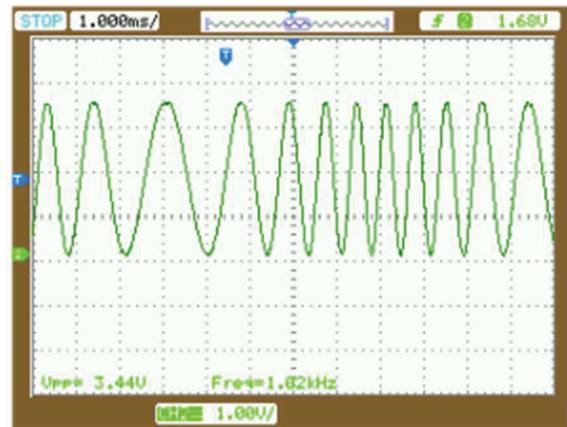
4.2.6 Modulação FM

A modulação em frequência foi obtida utilizando sinais modulantes senoidais. As frequências utilizadas foram 10 kHz para a modulante e 100 kHz para a portadora (Figura 46 (a)), 100 Hz para a modulante e 1 kHz para a portadora (Figura 46 (b)) e 20 Hz para a modulante e 200 Hz para a portadora (Figura 46 (c)). A simulação pode ser visualizada na Figura 49.

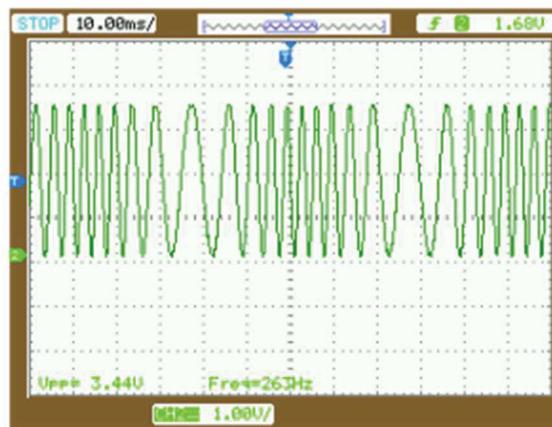
Figura 48 - Modulação FM



(a) Frequência modulante : 10 kHz
 Frequência portadora : 100 kHz



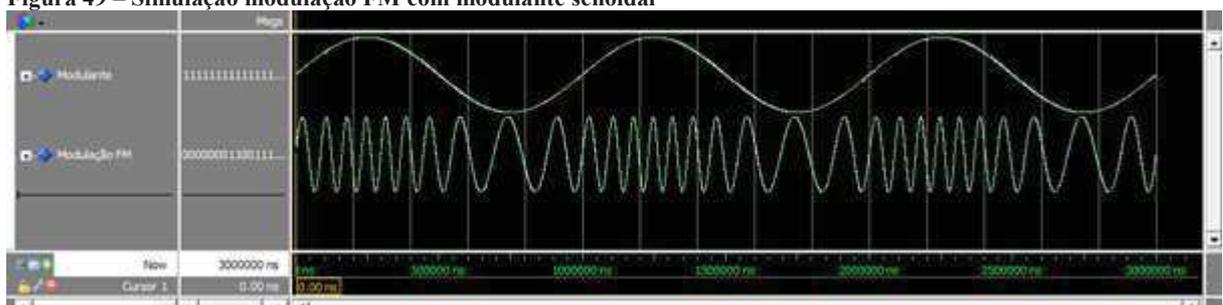
(b) Frequência modulante : 100 Hz
 Frequência portadora : 1 kHz



(c) Frequência modulante : 20 Hz
 Frequência portadora : 200 Hz

Fonte - Autoria própria.

Figura 49 – Simulação modulação FM com modulante senoidal



Fonte. Autoria própria.

4.2.7 Modulações Digitais

Para as modulações digitais, uma onda quadrada com *duty cycle* definido em 50% foi utilizada como sinal modulante. Os itens (a) e (b) da Figura 50 apresentam a modulação ASK, enquanto que os itens (c) e (d) a modulação FSK. As simulações podem ser visualizadas nas Figura 51, para a modulação ASK, e Figura 52 para a modulação FSK.

Figura 50 - Modulações digitais ASK e FSK



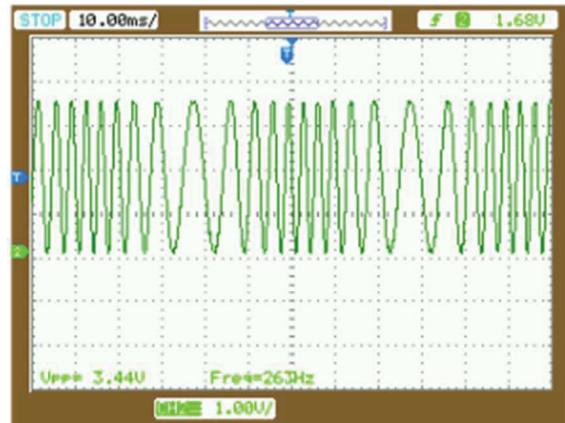
(a) Modulação ASK.
Frequência modulante : 20 Hz
Frequência portadora : 200 Hz



(b) Modulação ASK.
Frequência modulante : 100 Hz
Frequência portadora : 1 kHz



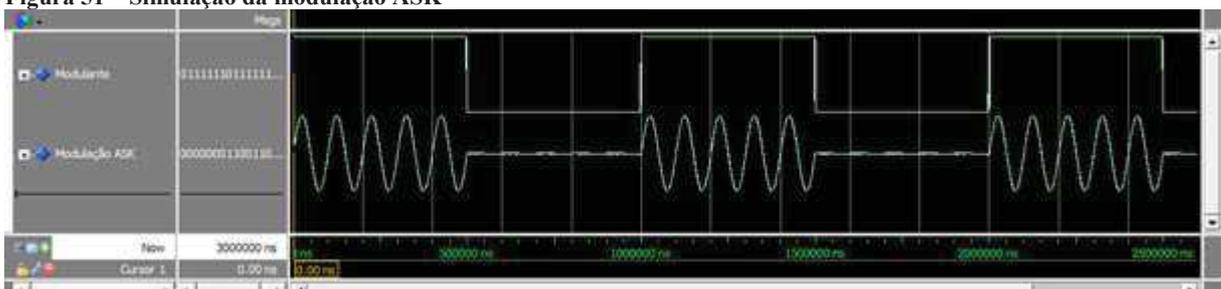
(c) Modulação FSK.
Frequência modulante: 100 Hz
Frequência portadora: 1 kHz



(d) Modulação FSK.
Frequência modulante: 20 Hz
Frequência portadora: 1 kHz

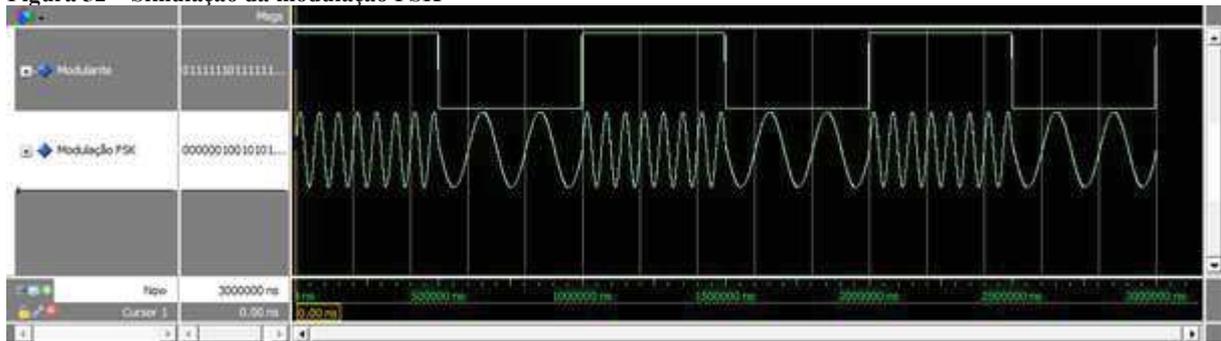
Fonte. Autoria própria.

Figura 51 – Simulação da modulação ASK



Fonte – Autoria própria.

Figura 52 – Simulação da modulação FSK



Fonte – Autoria própria.

4.3 ANÁLISE E DISCUSSÕES

Os sinais senoidais exibidos na Figura 42 apresentaram bons resultados para frequências inferiores à 250 kHz. É possível notar que nas frequências de 1 Hz, 25 kHz e 100 kHz, o pico positivo aponta uma pequena saturação no sinal, ocasionada pela saturação do amplificador operacional utilizado, que foi alimentado com ± 5 V. A diminuição da excursão máxima ou o aumento da alimentação do OPA727 poderia resolver o problema. Para frequências superiores à 350 kHz as distorções apresentadas comprometeram a fidelidade dos sinais.

Os sinais triangulares expostos na Figura 43 apresentaram resultados apropriados até 100 kHz. Acima dessa frequência, o sinal começa a perder suas características conforme ocorre o incremento da frequência. É possível notar que o sinal tende a forma de uma onda senoidal devido a suavização decorrente dos circuitos analógicos, como os fios utilizados, os tempos de atraso, as capacitâncias parasitas e indutâncias das linhas de transmissão.

Os sinais quadrados retratados na Figura 44 na frequência de 1 Hz, 10 kHz e 25 kHz apresentaram conformidade com relação ao sinal quadrado desejado. Já para frequências maiores, especificamente em 100 kHz, 250 kHz e 300 kHz é possível observar distorções características em sinais em que há incidência do fenômeno de Gibbs (OPPENHEIM; SCHAFER, 2012).

O sinal arbitrário apresentou resultado adequado nas frequências baixas em que foi coletado. Em todo caso, para frequências maiores, pelo fato de haver um menor número de amostras por período percebe-se a distorção no sinal, fato também ocorrido para os sinais senoidais, triangulares e quadrados anteriormente citados.

Todas as modulações apresentaram resultados esperados quando contrastados com os resultados simulados. Alterações seriam percebidas caso o sinal modulante e/ou portadora estivessem em frequências em que apresentassem distorções.

5.0 CONCLUSÃO

O objetivo desse trabalho de conclusão de curso consistiu na elaboração de um gerador de ondas arbitrárias utilizando a técnica DDS e com recursos de modulação, sendo este a continuação do trabalho de (REICHERT; MORETO, 2017). O diferencial desse projeto compreendeu na otimização dos recursos do FPGA, diminuindo principalmente o tamanho da memória RAM, a inclusão de quatro tipos de modulação, o desenvolvimento de uma interface gráfica em LabVIEW, desenvolvimento de um estágio de saída com DAC, elaboração de ganho e *offset* controlável.

O gerador de ondas arbitrárias desenvolvido apresentou resultados satisfatórios para as formas de ondas senoidais, triangulares, quadradas e arbitrárias como também para as modulações analógicas e digitais, como visto na Seção 4. Infere-se assim que o *hardware* e *software* desenvolvidos atenderam às expectativas.

Este trabalho ainda possibilitou a interação de várias disciplinas estudadas no decorrer do curso de engenharia eletrônica tais como: Eletrônica analógica II na elaboração de circuitos amplificadores; Medidas e sensores no estudo do circuito INA; Lógica reconfigurável na sintetização dos circuitos utilizando a linguagem VHDL; Sistemas embarcados na programação do processador NIOS II, realizado em linguagem C.

5.1 TRABALHOS FUTUROS

A fim de melhorar o desempenho do gerador de ondas arbitrárias algumas melhorias são sugeridas, tais como: o desenvolvimento da modulação angular (*Phase Modulation* ou PM) e a modulação em fase por chaveamento (*Phase Shift-Keying* ou PSK); aumentar o número de palavras para que seja compatível com o limite disponível em um FPGA de baixo custo, como é o caso dos modelos mais simples da Família Cyclone II, III e IV melhorando assim o sinal configurado e diminuindo o custo do projeto; o controle do ganho e *offset* poderia ser implementado digitalmente através de DAC's o que deixaria o estágio de saída mais complexo, no entanto, mais versátil para a configuração inteiramente digital e por fim, o estudo mais aprofundado da Gilbert Cell que proporcionaria também uma diminuição do custo do projeto.

REFERÊNCIAS

AGILENT TECHNOLOGIES. **User ' s Guide Agilent 33210A 10 MHz Function / Arbitrary Waveform Generator Agilent 33210A at a Glance**, 2008.

ANALOG DEVICES. **A Technical Tutorial on Digital Signal Synthesis**, 1999.

ANALOG DEVICES. **Analog Multipliers**, 2009.

ANALOG DEVICES. **Low Cost Analog Multiplier AD633**, 2015.

BK PRECISION. **20 MHz DDS Sweep Function Generator**. Disponível em:

<<http://www.bkprecision.com/products/discontinued/4040DDS-20-mhz-dds-sweep-function-generator.html>>. Acesso em: 12 jul. 2017.

BOYLESTAD, R. L.; NASHESKY, L. **DISPOSITIVOS ELETRÔNICOS e teoria de circuitos**, 2004.

CORTES, F. P.; BAMPI, S. A 40 MHz 70 dB Gain Variable Gain Amplifier Design Using the gm/ID Design Method. p. 7–12, 2009.

CRONIN, B. DDS Devices Generate High-Purity Waveforms Simply , Efficiently , and Flexibly. p. 1–24, 2017.

FUTURE TECHNOLOGY DEVICES INTERNATIONAL. **Future Technology Devices International Ltd. FT245R USB FIFO IC**, 2010.

HAYKIN, S. **Communication Systems**, 2001.

INTEL FPGA. **Cyclone IV**. Disponível em: <<https://www.altera.com/products/fpga/cyclone-series/cyclone-iv/overview.html>>. Acesso em: 12 set. 2017.

KEYSIGHT TECHNOLOGIES. **33210A Gerador de funções e formas de ondas arbitrárias de 10 MHz**. Disponível em: <<http://www.keysight.com/en/pd-1407410-pn-33210A/function-arbitrary-waveform-generator-10-mhz?cc=BR&lc=por>>. Acesso em: 27 jul. 2017a.

KEYSIGHT TECHNOLOGIES. **33503A Software BenchLink Waveform Builder Pro**. Disponível em: <<http://www.keysight.com/en/pd-1962285-pn-33503A/benchlink-waveform->

builder-pro-software?cc=BR&lc=por>. Acesso em: 27 jul. 2017b.

LATHI, B. P. **Signal Processing and Linear Systems**, 1998.

MACNICA DHW. **MANUAL DE USUÁRIO**, 2013a.

MACNICA DHW. **GUIA RAPIDO MERCURIO IV**, 2013b.

MANSUR, R. **Mercurio-Foto**. Disponível em: <<http://www.decom.ufop.br/imobilis/fpga-flexibilidade-no-projeto-de-hardware-parte-3-de-3/mercurio-foto/>>. Acesso em: 18 set. 2017.

MATIAS, M. L. **AMPLIFICADOR DE GANHO PROGRAMÁVEL PARA APLICAÇÕES DE RADIOFREQUÊNCIA**, 2014.

NATIONAL INSTRUMENTS. **O que é o LabVIEW**. Disponível em:

<[http://www.ni.com/pt-br/shop/labview.html?cid=Paid_Search-701310000011vZhAAI-](http://www.ni.com/pt-br/shop/labview.html?cid=Paid_Search-701310000011vZhAAI-Brazil-)

Google_SW1_LabVIEW_Broad&gclid=EAIAIqobChMIyaeMq6P21gIVyynTCh1uTAc8EAYASAAEgJMEPD_BwE>. Acesso em: 23 set. 2017.

OPPENHEIM, A. V.; SCHAFER, R. W. **Processamento em tempo discreto de sinais**, 2012.

PEDRONI, V. A. **Circuit Design with VHDL**, 2004. Disponível em:

<http://books.google.com/books?hl=en&lr=&id=b5NEgENaEn4C&oi=fnd∓pg=PA1&dq=Circuit+Design+with+VHDL&ots=LNdAwwmJlr&sig=9uI7nQfZkox5rb_prVaVzQLEK8M>

REICHERT, A.; MORETO, R. **DESENVOLVIMENTO DE UM GERADOR DE FORMAS DE ONDAS ARBITRÁRIAS UTILIZANDO A TÉCNICA DE SÍNTESE DIGITAL DIRETA**, 2017.

TEXAS INSTRUMENTS. **OPA727 e-trim 20MHz, High Precision CMOS Operational Amplifier**. Disponível em: <<http://www.ti.com/graphics/folders/partimages/OPA727.jpg>>.

APÊNDICE A – CÓDIGO DA PROGRAMAÇÃO DO PROCESSADOR NIOS II

```

1 #include "sys/alt_stdio.h"
2 #include "system.h"
3 #include "altera_avalon_pio_regs.h"
4 #include "stdio.h"
5 #include "ftdi.h"
6
7 int main()
8 {
9     unsigned char buffer1[4];
10    unsigned char vetorCompleto = 0;
11    unsigned char i = 0;
12    unsigned char contA = 0;
13    unsigned char contB = 0;
14    int buffer2[260];
15
16
17    while(ftdi_verifica_dispositivo());
18
19
20    while (1){
21        if(vetorCompleto == 0) //buffer2 completo?
22            if(ftdi_verifica_rxf()){ //Recebeu algo?
23                buffer1[contA] = ftdi_rchar(); //salva valor recebido
24                contA++; //incrementa contador.
25            }
26            if(contA == 4){ //buffer1 completo?
27                buffer2[contB] = buffer1[contA-4];
28                buffer2[contB] = buffer2[contB]<<8 | buffer1[contA-3];
29                buffer2[contB] = buffer2[contB]<<8 | buffer1[contA-2];
30                buffer2[contB] = buffer2[contB]<<8 | buffer1[contA-1]; //Salva as 4 posições do buffer1 em um posição do buffer2
31                contB++; //incrementa contador.
32                contA = 0; //zera contador.
33                if(buffer2[0] != 0xAAAAAAAA) { //Não recebeu inicio?
34                    contA = 0;
35                    contB = 0; //zera contadores.
36                }
37                if(contB == 260){ //buffer2 completo?
38                    vetorCompleto = 1;
39                    contB = 0;
40                }
41            }
42        else if(vetorCompleto == 1){ //se buffer2 esta completo, tratar os dados.
43            IOWR_ALTERA_AVALON_PIO_DATA(SEL_A_BASE,0); //NIOS II controlara endereçamento.
44            for(i = 1; i<257; i+=2){
45                IOWR_ALTERA_AVALON_PIO_DATA(W_EN_BASE,0); //Desabilita escrita.
46                IOWR_ALTERA_AVALON_PIO_DATA(NIOS_ADDR_A_BASE,0); //Define endereço.
47                IOWR_ALTERA_AVALON_PIO_DATA(DATA_A_BASE,buffer2[i]);
48                IOWR_ALTERA_AVALON_PIO_DATA(DATA_B_BASE,buffer2[i+1]); //Define dados nas entradas da memoria.
49                IOWR_ALTERA_AVALON_PIO_DATA(W_EN_BASE,1); //Habilita escrita.
50            }
51            IOWR_ALTERA_AVALON_PIO_DATA(W_EN_BASE,0); //Desabilita escrita.
52            IOWR_ALTERA_AVALON_PIO_DATA(FTW_A_BASE,buffer2[257]); //Define palavra de ajuste de frequência modulante.
53            IOWR_ALTERA_AVALON_PIO_DATA(FTW_B_BASE,buffer2[258]); //Define palavra de ajuste de frequência portadora.
54            IOWR_ALTERA_AVALON_PIO_DATA(MODE_BASE,buffer2[259]); //Seleciona modo (FM,AM ou normal).
55            IOWR_ALTERA_AVALON_PIO_DATA(SEL_A_BASE,1); //acumulador de fases controlara endereçamento.
56            vetorCompleto = 0;
57            for(i = 0; i<260; i++){
58                buffer2[i] = 0; //zera buffer2.
59            }
60        }
61    }
62    return 0;
63 }

```

APÊNDICE B DESCRIÇÃO DO ACUMULADOR DE FASES (AF_A)

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity AF_A is
6  port(
7      clk          : in std_logic;
8      nRST         : in std_logic;
9      ftw          : in std_logic_vector(31 downto 0);
10     ptr_addr     : out std_logic_vector(15 downto 0)
11 );
12 end entity;
13
14 architecture main of AF_A is
15 begin
16     PROCESS(clk,nRST)
17         variable cont : unsigned(31 downto 0) := (others => '0');
18         BEGIN
19             if (nRST = '0') then
20                 cont := (others=>'0');
21             elsif rising_edge(clk) then
22                 cont := cont + unsigned(ftw);
23             end if;
24             ptr_addr <= std_logic_vector(cont(31 downto 16));
25         end process;
26 end architecture;
```

APÊNDICE C - DESCRIÇÃO DO CIRCUITO SELETOR

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity selector is
6  port (
7      nios_addr      : IN std_logic_vector(7 downto 0);
8      af_addr       : IN std_logic_vector(7 downto 0);
9      sel           : IN std_logic;
10     addr_a,addr_b  : OUT std_logic_vector(7 downto 0)
11 );
12 end entity;
13
14 architecture main of selector is
15     signal addr      : unsigned(7 downto 0);
16     begin
17         addr <= unsigned(nios_addr) when sel = '0' else unsigned(af_addr);
18         addr_b <= std_logic_vector(addr);
19         addr_a <= std_logic_vector(addr + 1);
20
21     end architecture;
```

APÊNDICE D - DESCRIÇÃO DO CIRCUITO INTERPOLADOR

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity interpolator is
6  port(
7      a, b      : in  std_logic_vector(17 downto 0);
8      x        : in  std_logic_vector(7  downto 0);
9      q        : out std_logic_vector(17 downto 0)
10 );
11 end entity;
12
13 architecture main of interpolator is
14
15     signal a_int, b_int      : integer range (-2**17) to (2**17 - 1);
16     signal x_int            : integer range (0) to (2**8 - 1);
17     signal q_int           : integer;
18
19 begin
20     a_int <= to_integer(signed(a));
21     b_int <= to_integer(signed(b));
22     x_int <= to_integer(unsigned(x));
23
24     q_int <= b_int + ((a_int - b_int)*x_int)/256;
25     q      <= std_logic_vector(to_signed(q_int,18));
26
27 end architecture;

```

APÊNDICE E – DESCRIÇÃO DO ACUMULADOR DE FASES (AF_B)

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity AF_B is
6  port(
7      clk, rst    : in  std_logic;
8      ftw, fm, pm : in  std_logic_vector(31 downto 0);
9      saida      : out std_logic_vector(15 downto 0)
10 );
11 end entity;
12
13 architecture main of AF_B is
14 begin
15     process(clk,rst,pm,ftw)
16         variable i      : unsigned(31 downto 0) := (others => '0');
17         variable temp   : unsigned(31 downto 0) := (others => '0');
18     begin
19         if rst = '0' then
20             i := (others => '0');
21         elsif rising_edge(clk) then
22             if fm(31) = '0' then
23                 i := i + unsigned(ftw) + unsigned(fm);
24             else
25                 i := i + unsigned(ftw) - not unsigned(fm) + to_unsigned(1,32);
26             end if;
27         end if;
28
29         if pm(31) = '0' then
30             temp := i + unsigned(ftw) + unsigned(pm);
31         else
32             temp := i + unsigned(ftw) - not unsigned(pm) + to_unsigned(1,32);
33         end if;
34         saida <= std_logic_vector(temp(31 downto 16));
35     end process;
36
37 end architecture;

```

APÊNDICE F – DESCRIÇÃO DO BLOCO DESVIO DE FREQUÊNCIA

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity mod_mux is
6  port(
7      sel          : in  std_logic_vector(1 downto 0);
8      modulante    : in  std_logic_vector(17 downto 0);
9      ftw_mod      : in  std_logic_vector(31 downto 0);
10     fm, pm       : out std_logic_vector(31 downto 0)
11 );
12 end entity;
13
14 architecture main of mod_mux is
15
16     signal temp : signed(49 downto 0) := (others => '0');
17
18 begin
19     desFreq <= signed(modulante)*signed(ftw_mod);
20     process(sel,modulante,desFreq)
21     begin
22         case sel is
23             when "01" =>
24                 pm <= (others => '0');
25                 fm <= std_logic_vector(desFreq(49 downto 18));
26             when "10" =>
27                 fm <= (others => '0');
28                 pm <= std_logic_vector(desFreq(49 downto 18));
29             when others =>
30                 fm <= (others => '0');
31                 pm <= (others => '0');
32         end case;
33     end process;
34 end architecture;

```

APÊNDICE G – DESCRIÇÃO DO BLOCO MODULAÇÃO AM

```
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity am_modulator is
6     port(
7         modulante      : in std_logic_vector(17 downto 0);
8         portadora      : in std_logic_vector(17 downto 0);
9         saida          : out std_logic_vector(17 downto 0)
10    );
11 end entity;
12
13 architecture main of am_modulator is
14     signal temp : signed(35 downto 0);
15 begin
16     process(portadora, modulante)
17         variable temp      : std_logic_vector(17 downto 0) := (others => '0');
18         variable soma      : signed(17 downto 0) := (others => '0');
19         variable produto   : signed(35 downto 0) := (others => '0');
20     begin
21         produto := signed(portadora)*signed(modulante);
22         temp := std_logic_vector(produto(35 downto 18));
23         soma := signed(temp);
24         soma := soma + (shift_right(signed(portadora), 1));
25         saida <= std_logic_vector(soma);
26     end process;
27 end architecture;
```

APENDICE H – DESCRIÇÃO DO MULTIPLEXADOR DE SAÍDA

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.numeric_std.all;
4
5  entity output_mux is
6  port(
7      normal, am, fm_pm    : in std_logic_vector(17 downto 0);
8      mode                 : in std_logic_vector(1 downto 0);
9      saida                : out std_logic_vector(11 downto 0)
10 );
11 end entity;
12
13 architecture main of output_mux is
14 begin
15     process(normal, am, fm_pm, mode)
16         variable saidaMux : std_logic_vector(17 downto 0);
17         variable temp     : unsigned(17 downto 0);
18     begin
19         case mode is
20             when "00" =>
21                 saidaMux := normal;
22             when "11" =>
23                 saidaMux := am;
24             when others =>
25                 saidaMux := fm_pm;
26         end case;
27         temp := unsigned(saidaMux) + to_unsigned(2**17,18);
28         saida <= std_logic_vector(temp(17 downto 6));
29     end process;
30 end architecture;
```

