

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**

RENÃ MARQUES PASTORE

**PROJETO, MONTAGEM E TESTE DE UM INVERSOR
TRIFÁSICO EM PONTE UTILIZANDO O MÓDULO
FNA41060B2 ACIONADO POR FPGA**

**VITÓRIA
2019**

RENÃ MARQUES PASTORE

**PROJETO, MONTAGEM E TESTE DE UM INVERSOR TRIFÁSICO
EM PONTE UTILIZANDO O MÓDULO FNA41060B2 ACIONADO
POR FPGA**

Parte manuscrita da Proposta de Projeto de Graduação do aluno **Renã Marques Pastore**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Orientador: Prof. Dr. Walbermark Marques dos Santos

VITÓRIA
2019

RENÃ MARQUES PASTORE

**PROJETO, MONTAGEM E TESTE DE UM INVERSOR TRIFÁSICO
EM PONTE UTILIZANDO O MÓDULO FNA41060B2 ACIONADO
POR FPGA**

Parte manuscrita do Projeto de Graduação do aluno **Renã Marques Pastore**, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovada em 02 de setembro de 2019.

COMISSÃO EXAMINADORA:

Prof. Dr. Walbermark Marques dos Santos
Universidade Federal do Espírito Santo
Orientador

Profa. Dra. Eliete Maria de Oliveira Caldeira
Universidade Federal do Espírito Santo
Examinador

Prof. Dr. Hélio Marcos André Antunes
Universidade Federal do Espírito Santo
Examinador

Prof. MSc. Vinícius Secchin de Melo
Instituto Federal do Espírito Santo
Examinador

AGRADECIMENTO

Agradeço ao meu Orientador por todo o tempo, dedicação, trabalho e pela disposição em me auxiliar, sempre precisei. Agradeço também à minha família sempre me apoiando e me incentivando. E em especial ao meu tio José Leandro Pastore que sempre me aconselhando e ajudando, não vou esquecer “Vai lá e tenta outra vez, se fosse fácil todo mundo fazia”.

RESUMO

Em virtude das diversificadas aplicações de inversores trifásicos, este trabalho apresenta algumas aplicações em diferentes setores industriais, mostrando assim sua importância. Dentro desse contexto, tem-se a proposta de apresentar as etapas de estudo, análise, projeto e implementação experimental de um conversor trifásico CC-CA para alimentar uma carga resistiva. O circuito de controle emprega o *kit* Nexys2 sendo este uma placa de aprendizagem/prototipagem que possui uma FPGA XC3S1200E da Xilinx que é um circuito integrado. Já o circuito de potência é composto pelo filtro LC e o CI FNA41060B2. Por fim, é apresentada a simulação assim como os testes experimentais, sendo estes efetuados com carga resistiva mantendo a frequência constante e variando a amplitude da tensão. No em um segundo teste a amplitude da tensão variando a frequência. Os resultados obtidos demonstram que protótipo foi capaz de variar tanto a tensão aplicada a carga como também a frequência, sendo estes controlados pelo operador.

Palavras-chave: Conversor trifásico CC-CA. Kit Nexys2. CI FNA41060B2. Filtro LC.

LISTA DE FIGURAS

Figura 1 – Estrutura básica de um inversor trifásico	17
Figura 2 – Inversor trifásico de tensão tipo 180°	18
Figura 3 – Diagrama de comando do inversor trifásico de tensão tipo 180°	19
Figura 4 – Formas de onda das tensões de linha e pulsos de comando nas chaves para o inverso 180°	20
Figura 5 – Representação da tensão VAt	21
Figura 6 – Circuito simplificado do inversor trifásico e carga balanceada	22
Figura 7 – Variação da tensão sobre o capacitor e corrente e tensão no indutor.....	29
Figura 8 – Tensão $VAB(t)$ para o semiciclo positivo da moduladora.....	30
Figura 9 – CI do módulo inversor.....	34
Figura 10 – Circuito de acionamento do FNA41060B2	35
Figura 11 – Circuito térmico equivalente.....	36
Figura 12 – Perdas causadas pela comutação nos IGBTs	38
Figura 13 – Impedância térmica entre a junção e a cápsula do IGBT	39
Figura 14 – Circuito térmico equivalente para regime de operação	40
Figura 15 – <i>Kit</i> Digilent Nexys2.....	41
Figura 16 – Programa de um contador destacando a estrutura básica projeto em VHDL	44
Figura 17 – Diagrama de funcionamento do código implementado.....	45
Figura 18 – Efeito <i>bouncing</i>	46
Figura 19 – Estrutura <i>display</i> anodo comum.....	47
Figura 20 – Diagrama de Bode em malha aberta com carga nominal	51
Figura 21 – Diagrama de Bode malha aberta operando a vazio	51
Figura 22 – Modelo do inversor simulado	52
Figura 23 – Bloco SPWM	53
Figura 24 – Simulação do inversor operando carga nominal	53
Figura 25 – Tensão de saída com carga nominal.....	54
Figura 26 – Foto do protótipo montado	54
Figura 27 – Tempo morto no pulso SPWM	55
Figura 28 – Saída do inversor sem filtro.....	55
Figura 29 – Tensão no barramento CC	56
Figura 30 – Carga do inversor	56

Figura 31 – Defasagem da tensão entre fases.....	57
Figura 32 – Tensão de linha	57
Figura 33 – Tensão de fase.....	58
Figura 34 – Variação em amplitude da tensão, sendo (a) 20%, (b) 40 %, (c) 60 %, (d) 90 % e (e) 100 %.....	59
Figura 35 – Variação em frequência, sendo (a) 30 Hz, (b) 50 Hz, (c) 70 Hz, (d) 90 Hz e (e) 100 Hz	61

LISTA DE QUADROS

Quadro 1 – Decodificador BCD para <i>display</i> de sete segmentos	47
Quadro 2 – Especificação para o projeto do inversor trifásico de tensão	49

LISTA DE TABELAS

Tabela 1 – Variação em amplitude	60
Tabela 2 – Comparação entre frequência valor teórico <i>versus</i> experimental.....	61

LISTA DE ABREVIATURAS E SIGLAS

ASIC	<i>Application Specific Integrated Circuit</i>
BCD	<i>Binary Code Decimal</i>
BJT	<i>Bipolar junction transistor</i>
CA	Corrente alternada
CC	Corrente Contínua
CPLD	<i>Complex Programmable Logic Device</i>
FPGA	<i>Field Programmable Gate Array</i>
GTO	<i>Gate Turn-Off Thyristor</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IGBT	<i>Insulated Gate Bipolar Transistor</i>
MOSFET	<i>Metal Oxide Semiconductor Field Effect Transistor</i>
SPWM	<i>Sinusoidal Pulse Width Modulation</i>
VHDL	<i>Hardware description language</i>
VSI	<i>Voltage Source Inverters</i>

LISTA DE SÍMBOLOS

V_I	Tensão de entrada (V)
$d_A(t)$	Razão-cíclica do braço A [Adimensional]
$V_A(t)$	Tensão instantânea no braço A (V)
T_s	Período completo de comutação (S)
Δt_1	Largura de pulso (S)
Δt_2	Largura de pulso complementar (S)
$d_B(t)$	Razão-cíclica do braço B [Adimensional]
$d_C(t)$	Razão-cíclica do braço C [Adimensional]
$V_{L1}(t)$	Tensão instantânea sobre indutor L_1 (V)
V_{1G1}	Tensão instantânea sobre capacitor C_1 com referência do ponto G1 (V)
V_{2G1}	Tensão instantânea sobre capacitor C_2 com referência do ponto G1 (V)
$V_{L2}(t)$	Tensão instantânea sobre indutor L_2 (V)
$V_B(t)$	Tensão instantânea braço no B (V)
$V_C(t)$	Tensão instantânea braço no C (V)
$V_{L13}(t)$	Tensão instantânea sobre indutor L_3 (V)
V_{3G1}	Tensão instantânea sobre capacitor C_3 com referência do ponto G1 (V)
I_{L1}	Corrente instantânea no indutor L_1 (A)
I_{C1}	Corrente instantânea no capacitor C_1 (A)
I_{R1}	Corrente instantânea no resistor R_1 (A)
I_{L2}	Corrente instantânea no indutor L_2 (A)
I_{C2}	Corrente instantânea no capacitor C_2 (A)
I_{R2}	Corrente instantânea no resistor R_2 (A)
I_{L3}	Corrente instantânea in no dutor L_3 (A)
I_{C3}	Corrente instantânea no capacitor C_3 (A)
I_{R3}	Corrente instantânea no resistor R_{13} (A)
L	Indutor de Filtragem (H)
C	Capacitor de Filtragem (F)
R	Resistência de carga (Ω)
$V_{L12}(t)$	Tensão de linha entre os indutores L_1 e L_2 (V)
$V_1(t)$	Tensão instantânea sobre resistor R_1 (V)
$V_2(t)$	Tensão instantânea sobre resistor R_2 (V)

$V_{L23}(t)$	Tensão de linha entre os indutores L_2 e L_3 (V)
$V_{L31}(t)$	Tensão de linha entre os indutores L_3 e L_1 (V)
$V_{AB}(t)$	Tensão de linha entre as fases A e B (V)
$V_{12}(t)$	Tensão de linha entre os resistores R_1 e R_2 (V)
$V_{BC}(t)$	Tensão de linha entre as fases B e C (V)
$V_{23}(t)$	Tensão de linha entre os resistores R_2 e R_3 (V)
$V_{CA}(t)$	Tensão de linha entre as fases C e A (V)
$V_{31}(t)$	Tensão de linha entre os resistores R_3 e R_1 (V)
$d_{AB}(t)$	Razão-cíclica entre os braços A e B [Adimensional]
$d_{BC}(t)$	Razão-cíclica entre os braços B e C [Adimensional]
$d_{CA}(t)$	Razão-cíclica entre os braços C e A [Adimensional]
$[X]_{123}$	Matriz a qual será aplicada a transformada de Clarke
$[X]_{0\infty\beta}$	Matriz após aplicação da transformada Clarke
$[A]^{-1}$	Matriz inversa da transformada Clarke [Adimensional]
V_{TRIPK}	Tensão de pico da portadora triangular (V)
T_S^*	Período da tensão V_{AB} (S)
V_{OPK}	Tensão de pico da saída (V)
ΔI_L	Ondulação de corrente no indutor de filtragem (A)
F_S	Frequência de comutação (Hz)
$\overline{\Delta I_L}$	Ondulação de corrente no indutor de filtragem parametrizada (A)
ΔQ	Variação de carga elétrica (C)
ΔV_C	Variação de tensão no capacitor (V)
Q	Carga elétrica (C)
t	Tempo (S)
C_{BS_MIN}	Capacitor <i>bootstrap</i> (F)
I_{LEAK}	Corrente máxima de descarga do CBS (A)
∇t	Máximo na largura de pulso do IGBT superior (S)
∇V_{BS}	Tensão de descarga admissível do CBS <i>ripple</i> (V)
T_J	Temperatura de junção (°C)
T_C	Temperatura da capsula (°C)
T_D	Temperatura do dissipador (°C)
T_A	Temperatura do ambiente (°C)

P	Potência térmica produzida pela corrente que é transferida para o meio (W)
R_{JC}	Resistência térmica entre a junção e a cápsula ($^{\circ}\text{C}/\text{W}$)
R_{CD}	Resistência térmica entre a cápsula e o dissipador ($^{\circ}\text{C}/\text{W}$)
R_{DA}	Resistência térmica entre o dissipador e o ambiente ($^{\circ}\text{C}/\text{W}$)
R_{JA}	Resistência térmica entre a junção e o ambiente ($^{\circ}\text{C}/\text{W}$)
$P_{\text{IGBT}}^{\text{COND}}$	Perda por condução (W)
$P_{\text{IGBT}}^{\text{COMUTA}}$	Perda por comutação (W)
E_{ON}	Energia dissipada com IGBT ativo (J)
E_{OFF}	Energia dissipada com IGBT inativo (J)
E_{REC}	Energia dissipada durante transitório (J)
T_{COMU}	Período de comutação (S)
F_{COMU}	Frequência de comutação (Hz)
$P_{\text{IGBT}}^{\text{TOTAL}}$	Perdas por comutação e condução (W)
P_E	Potência elétrica (W)
V_{OEFZ}	Tensão eficaz do inversor (V)
I_{OEFZ}	Corrente eficaz do inversor (V)
V_{OPK}	Tensão de pico (A)
F_O	Frequência de ressonância (Hz)
I_{OPK}	Corrente de pico (A)
V_{FASE}	Tensão de fase (V)
V_{LINHA}	Tensão de linha (V)
F_R	Frequência da moduladora (Hz)
V_{REFPK}	Tensão de pico da referência (V)
M	Relação entre as tensões de referência e da portadora (Adimensional)
N	Relação entre os períodos de referência e da portadora (Adimensional)

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	Objetivo geral.....	16
1.1.1	Objetivos Específicos	16
1.2	Estrutura do texto	16
2	INVERSOR DE FREQUÊNCIA TRIFÁSICO VSI	17
2.1	Determinação da tensão em cada braço	21
2.2	Equacionamento do inversor trifásico	22
2.1.1	Tensões sobre os indutores	23
2.2.2	Aplicação da transformada de Clarke.....	25
2.3.1	Filtro LC.....	29
2.3.1.1	CÁLCULO DO INDUTOR.....	30
2.3.1.2	CAPACITOR DE FILTRAGEM.....	33
2.3.2	CI comercial FNA41060B2	34
2.3.3	Dimensionamento térmico	36
2.3.3.1	PERDAS POR COMUTAÇÃO.....	37
2.3.3.2	RESISTÊNCIA TÉRMICA ENTRE A JUNÇÃO E A CÁPSULA, E ENTRE A CÁPSULA E O DISSIPADOR.....	38
3	KIT FPGA NEXYS2.....	41
3.1	Utilizando a Linguagem VHDL	42
3.2	Modelagem de componentes VHDL	43
4	PROJETO DO INVERSOR	49
4.1	Determinação de valores do circuito potência.....	49
4.2	Condição crítica de funcionamento	50
4.3	Simulações e resultados.....	52
4.3.1	Resultados experimentais.....	54
4.3.2	Variando a tensão de saída.....	58
5	CONCLUSÕES FINAIS E SUGESTÕES DE TRABALHOS FUTUROS	62
5.1	Conclusões	62
5.2	Sugestões de trabalhos futuros.....	63
	REFERÊNCIAS BIBLIOGRÁFICAS.....	64
	APÊNDICE A – CÓDIGO UTILIZADO NA FPGA.....	67

1 INTRODUÇÃO

Por muito tempo, na indústria; foram utilizados motores de corrente contínua com a necessidade de acionar cargas com variações em sua velocidade. Entretanto, com o avanço da eletrônica de potência, alguns desses motores estão sendo substituídos, pelo conjunto, motor de indução e inversor de frequência. Os inversores são conversores estáticos CC-CA de tensão ou corrente e têm como função favorecer o trânsito do fluxo de potência entre uma fonte de tensão contínua e uma carga de tensão alternada.

Esses inversores possui um chaveamento em alta frequência de seus interruptores do inversor, os quais provoca a formação de onda na saída com alto conteúdo harmônico, que pode ser minimizado trabalhando-se em técnicas de modulação mais aprimoradas. A tensão ou corrente senoidal na carga é conseguida utilizando-se filtros passivos (indutores e/ou capacitores), o que acarreta em custos financeiros.

No mercado há diversos modelos de interruptores disponíveis, sendo os mais utilizados para o chaveamento do inversor são: BJTs (do inglês *Bipolar Junction Transistor*), MOSFETs (do inglês *Metal Oxide Semiconductor Field Effect Transistor*), IGBTs (do inglês *Insulated Gate Bipolar Transistor*) e GTOs (do inglês *Gate Turn-Off Thyristor*). Todas essas chaves estáticas são controladas tanto na entrada em condução como no bloqueio, sendo as duas últimas recomendadas para potências elevadas (MARTINS; BARBI, 2005, p. 1). Os inversores são aplicados em vários processos modernos, como exemplo, indústria naval, indústria de petróleo, indústria de cana de açúcar, siderurgia e veículos elétricos.

Na indústria naval os inversores são utilizados em bombas de carga e até mesmo como propulsão principal ou em túnel das embarcações, sendo esta operada apenas por períodos limitados durante a atracação e manobra (ESTEVES, 2013, p. 34).

O inversor padrão da WEG - CFW-09, 6 pulsos, na realidade para área naval, é utilizado apenas para acionamento dos motores de bombas de carga, ventiladores, outras cargas com baixa potência que necessitem de variação de velocidade (GURGEL, 2009, p. 96).

Em se tratando da indústria de petróleo, os inversores de frequência, como o CFW-09, são utilizados também em pequenas bombas e ventiladores. Contudo, em sua maioria são empregados conversores com uma alimentação em média tensão. Em determinadas

plataformas, eles são usados nos motores das bombas de combate a incêndio, acionando as mesmas com uma partida suave. Isso elimina a exigência de correntes com valores elevados nos geradores que alimentam estas bombas. Essa ação contribui para a diminuição do desgaste mecânico e elétrico do conjunto motor-gerador (GURGEL, 2009).

Na indústria de cana de açúcar, os inversores da família CFW são os mais adotados pois os valores de potência utilizados nesse tipo de processo situam-se entre 1 cv e 2500 cv com tensões que variam de 380 V a 690 V. Os conversores são aplicados em ventiladores das caldeiras, esteiras transportadoras, moendas e picadores (WEG, 2008).

Para a área siderúrgica apresentam-se em sistema de lingotamento, ponte rolante etc. Nesse âmbito, devido à sua característica, os inversores de frequência regenerativos são usados para obter economia de energia (GURGEL, 2009).

A faixa de potência nestas aplicações pode variar bastante, partindo de motores de 10 kW até motores de 5000 kW, sendo os de maior potência, normalmente em média tensão para reduzir a corrente dos mesmos, fator este que define os custos da instalação e dos acionamentos destes motores. As tensões podem variar de 380 V a 6,6 kV (GURGEL, 2009, p. 99).

Nos veículos elétricos e híbridos, os inversores também encontram aplicação. Destaca-se, por exemplo, o Nissan LEAF, que é um veículo 100% elétrico. Seu módulo (motor, inversor e *driver*) é fabricado pela Remy Electric.

A WEG é uma das empresas brasileiras que atua no mercado de veículos elétricos, produzindo motores e inversores para ônibus com tração elétrica, como pode ser visto no texto abaixo:

O primeiro ônibus elétrico brasileiro movido 100% a bateria foi lançado em São Bernardo do Campo /SP, neste mês e deve entrar em testes no próximo. Com capacidade máxima de cerca de 150 pessoas, é também o primeiro do mundo movido a bateria com 18 metros de comprimento. O veículo que irá operar o dia todo sem emissão de poluentes é equipado pela WEG, com motor de tração refrigerado a água, inversor de tração refrigerado a água, além de equipamentos auxiliares. O ônibus foi concebido pela Eletra, responsável pela engenharia, integração e desenvolvimento do projeto (WEG, 2013, p. 1).

Tendo em vista algumas das aplicações dos inversores de frequência. Pretende-se com este trabalho adquirir conhecimento na área de projeto de um inversor de frequência trifásico.

1.1 Objetivo geral

Estudar, projetar e montar um inversor trifásico, acionado pelo *kit* FPGA Nexys2, utilizando o CI FNA41060B2.

1.1.1 Objetivos Específicos

- Identificar a técnica de modulação mais adequada para a aplicação;
- Conhecer as características e limitações do módulo inversor FNA41060B2;
- Projetar um circuito inversor de tensão capaz de acionar uma carga de 750 W;
- Analisar os resultados obtidos com o projeto.

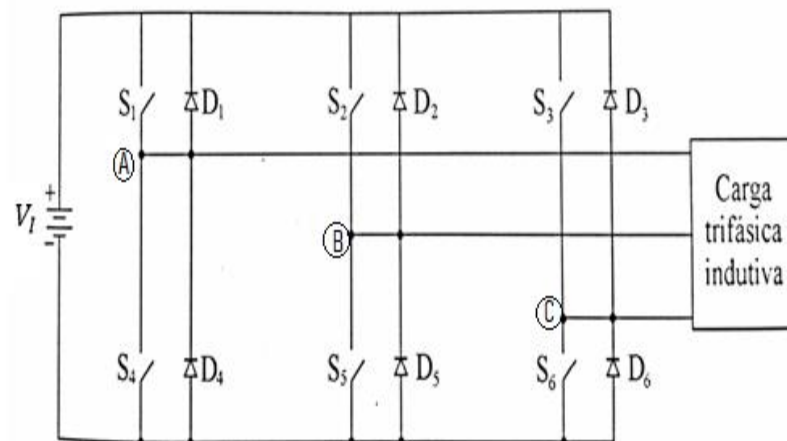
1.2 Organização do texto

No início do texto será apresentada a topologia do inversor de frequência, demonstrando a estrutura básica, assim como a sequência de chaveamento. Em seguida, podem ser vistos o equacionamento dos braços e as formas de ondas atuantes sobre os componentes. Posteriormente, tem-se a apresentação do CI (circuito integrado) e o cálculo térmico, para determinar o seu dissipador. Na sequência, será apresentado o kit FPGA Nexys2, o qual é responsável pelos sinais de comando para abertura ou fechamento das chaves, assim como uma descrição da linguagem de *hardware*. Depois, será realizado o dimensionamento dos componentes bem como a análise matemática para descoberta da condição crítica de funcionamento. E, por fim, tem-se as simulações e os resultados experimentais.

2 INVERSOR DE FREQUÊNCIA TRIFÁSICO VSI

A estrutura interna de um inversor trifásico (VSI, do inglês *Voltage Source Inverters*) é apresentada na Figura 1, onde a tensão do barramento em Corrente Contínua (CC) é representada pela fonte V_I , e os IGBTs operando na zona de saturação, atuando assim como as chaves. O controle sobre as chaves permite obter um sistema trifásico equilibrado.

Figura 1 – Estrutura básica de um inversor trifásico



Fonte: Martins e Barbi (2005).

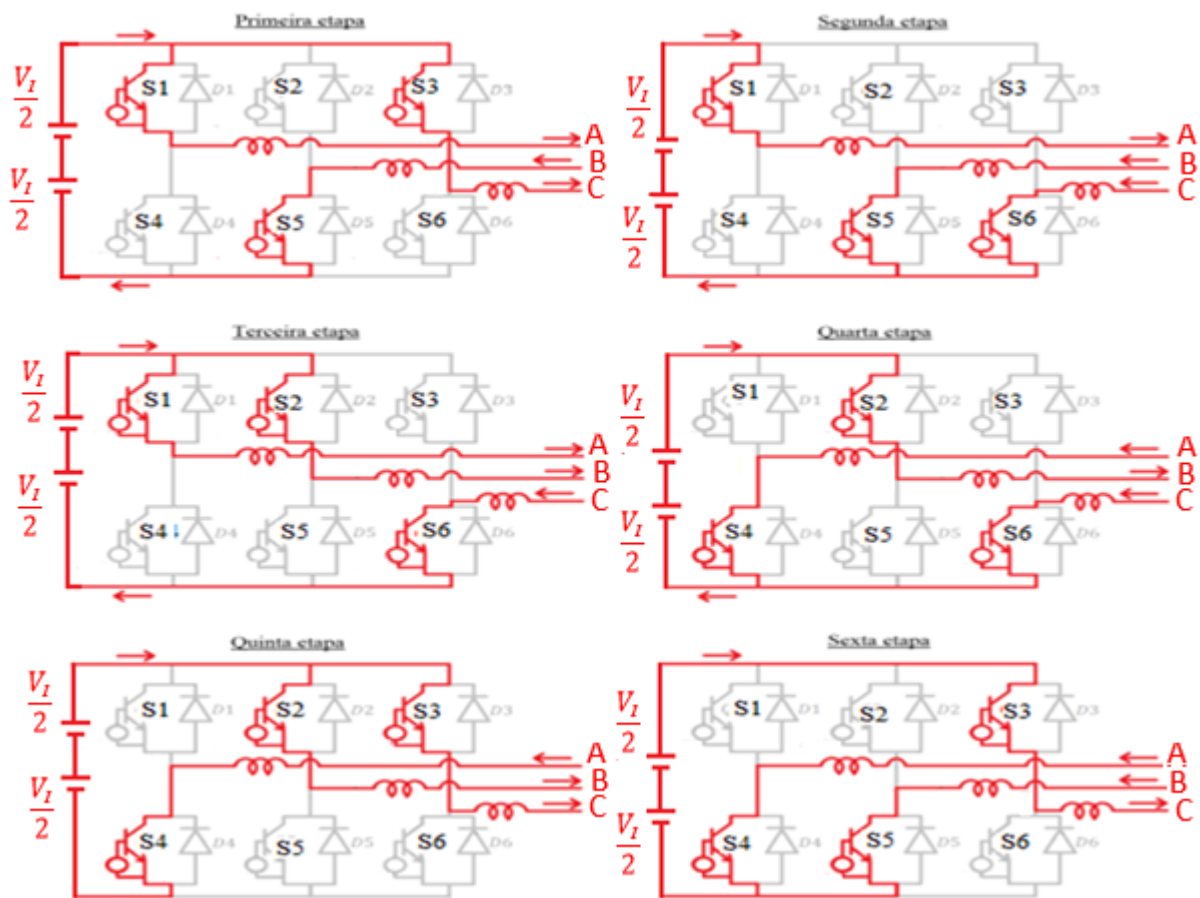
A tensão de saída trifásica ocorre por meio da defasagem imposta em cada comando que ativa o braço, sendo igual a 120° ou 180° . A defasagem de 180° é a mais utilizada, pois as chaves são mais bem aproveitadas visto que a forma da onda não é afetada pela natureza da carga, neste sistema (RANIEL, 2011, p. 33).

Neste caso, cada chave comandada é mantida em condução durante 180° . Os comandos das duas chaves de um mesmo braço do inversor são complementares. Os comandos das chaves de um braço estão defasados de 120° em relação aos comandos das chaves do braço vizinho. A partir dessa estratégia de comando a tensão de saída é imposta a todo instante, qualquer que seja a natureza da carga (PACHECO, 2012, p. 28).

Em acordo com a Figura 1 a estrutura básica de um inversor trifásico. A chave S_1 conduz até a metade do período, após isso, ela é bloqueada. No mesmo instante em que S_1 é bloqueada, a chave S_4 que estava bloqueada passa a conduzir. Este mesmo efeito de alternância acontece no braço B e C. Ao utilizar o braço A como referência, pode-se dizer que a fase B está 120° atrasada em relação à fase A. Assim como ocorre em C que está 120° adiantada.

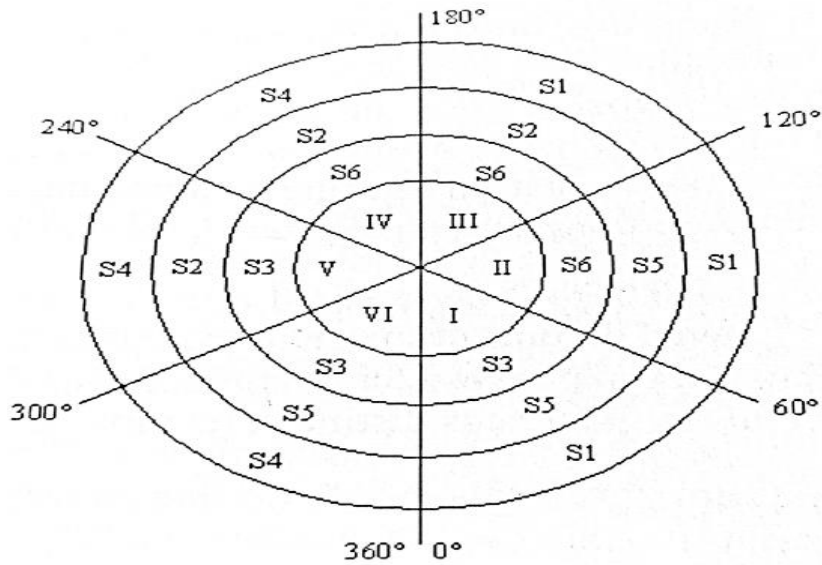
Para facilitar a análise, a fonte CC é dividida em duas. Cada uma com $\frac{V_I}{2}$ como pode ser visto na Figura 2. Note que existem seis sequências de funcionamento, as quais podem ser visualizadas no diagrama de comando na Figura 3. “Em cada sequência existem sempre três chaves em condução, duas no grupo positivo e uma no grupo negativo, ou vice-versa. A duração de cada sequência é de 60°” (MARTINS; BARBI, 2005, p. 46).

Figura 2 – Inversor trifásico de tensão tipo 180°



Fonte: Produção do próprio autor.

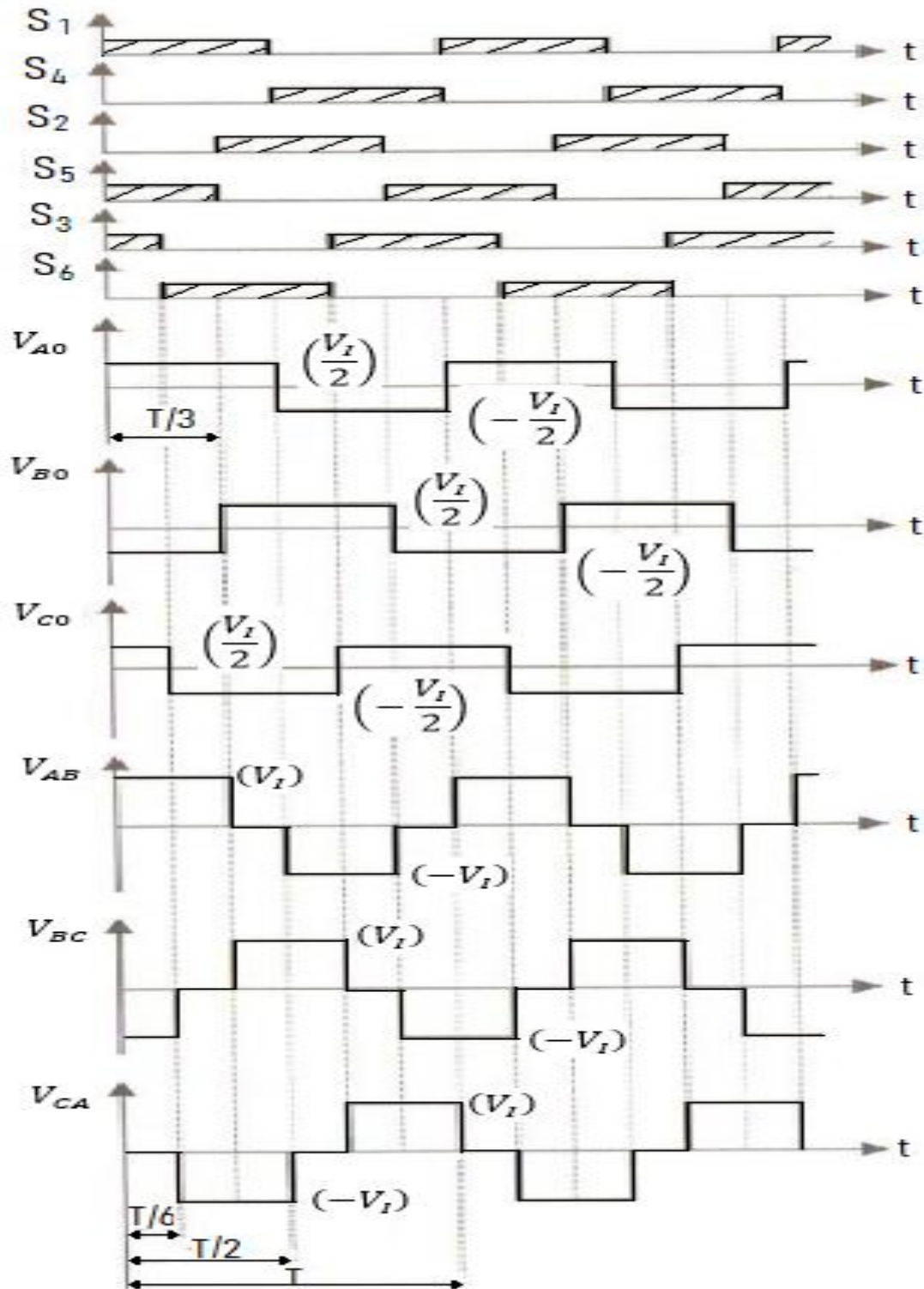
Figura 3 – Diagrama de comando do inversor trifásico de tensão tipo 180°



Fonte: Martins e Barbi (2005).

Cada braço corresponde a uma fase. Nas fases utilizadas A, B e C existe uma defasagem de 120° entre cada uma delas. Como a tensão de linha é dada entre cada fase, temos V_{AB} , V_{BC} e V_{CA} . Como a formação de cada fase é a mesma, o formato da tensão de linha também é o mesmo, das tensões de fase, porém, com tensões em V_L , zero e $-V_L$. Os comandos de cada chave e os formatos de cada onda de cada tensão de linha podem ser vistos na Figura 4.

Figura 4 – Formas de onda das tensões de linha e pulsos de comando nas chaves para o inverso 180°



Fonte: Martins e Barbi (2005).

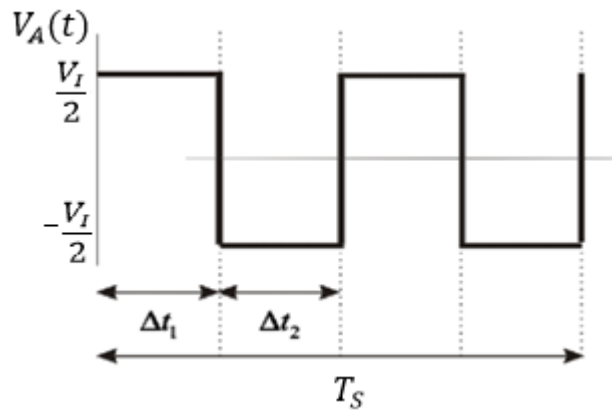
Nota: Adaptado pelo autor.

2.1 Determinação da tensão em cada braço

A modulação adotada neste trabalho é do tipo senoidal, deste modo os pulsos são gerados entre a comparação de três moduladoras senoidais defasadas entre em 120° , que tem como referência a tensão portadora que neste caso é uma onda triangular.

Nesta seção será analisada a tensão do braço A, com o intuito de elucidar a dependência da razão cíclica $d_A(t)$. Para simplificação deste estudo, considera-se a representação da tensão $V_A(t)$, dada pela Figura 5 (HAUSMANN et al., 2008, p. 9).

Figura 5 – Representação da tensão $V_A(t)$



Fonte: Hausmann e outros (2008).

Para período completo de comutação, representado na Figura 5, chega-se à equação (1) da tensão $V_A(t)$ dada por:

$$V_A(t) = \frac{1}{T_S} \left[\int_0^{\Delta t_1} \frac{V_I}{2} dt + \int_0^{\Delta t_2} -\frac{V_I}{2} dt \right] \quad (1)$$

Onde T_S representa o período completo de comutação, já as variáveis Δt_1 e Δt_2 , respectivamente, são a largura de pulso e a largura de pulso complementar as quais encontram-se nas equações (2) e (3).

$$\Delta t_1 = d_A(t) \quad (2)$$

$$\Delta t_2 = 1 - d_A(t) \quad (3)$$

Resolvendo a equação (1) para os intervalos dados nas equações (2) e (3), define-se a tensão do braço A, representada por $V_A(t)$, como mostrado na equação (4).

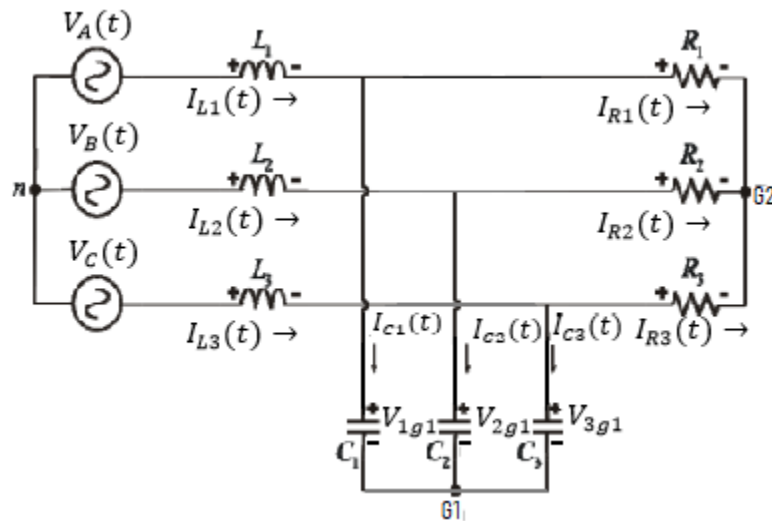
$$V_A(t) = V_I * d_A(t) - \frac{V_I}{2} \quad (4)$$

O mesmo pode ser feito para os demais braços, B e C.

2.2 Equacionamento do inversor trifásico

O circuito do inversor trifásico com carga resistiva é mostrado na Figura 6.

Figura 6 – Circuito simplificado do inversor trifásico e carga balanceada



Fonte: Hausmann e outros (2008).

A modelagem do circuito é feita com o inversor operando com uma carga resistiva. Devido ao fato de as componentes contínuas se anularem em malha fechada, estas não foram representadas nas equações (5), (6) e (7) (HAUSMANN et al., 2008, p. 17).

$$V_A(t) = V_I * d_A(t) \quad (5)$$

$$V_B(t) = V_I * d_B(t) \quad (6)$$

$$V_C(t) = V_I * d_C(t) \quad (7)$$

Para um sistema em equilíbrio a tensão nos pontos G1 e G2, vistos na Figura 6, são iguais (HAUSMANN et al., 2008, p. 7).

Ao aplicar a equação de malha sobre o circuito simplificado, obtém-se três malhas, as quais são apresentadas como malha 1, 2 e 3. Onde $V_{L1}(t)$ é a tensão instantânea sobre o indutor L_1 , $V_{1G1}(t)$ tensão instantânea sobre o capacitor C_1 , $V_{2G1}(t)$ tensão instantânea sobre o capacitor C_2 e $V_{L2}(t)$ é a tensão instantânea sobre o indutor L_2 . A malha 1 traz as equações (8) e (9).

$$V_A(t) = V_{L1}(t) + V_{1G1}(t) - V_{2G1}(t) - V_{L2}(t) + V_B(t) \quad (8)$$

$$\begin{cases} [V_{1G1}(t) - V_{2G1}(t)] = [V_1(t) - V_{G1}(t)] - [V_2(t) - V_{G1}(t)] \\ [V_{1G1}(t) - V_{2G1}(t)] = [V_1(t) - V_2(t)] \end{cases} \quad (9)$$

Reescrevendo a equação (8) aplicando os parâmetros da equação (9) tem-se a equação (10).

$$[V_A(t) - V_B(t)] = [V_{L1}(t) - V_{L2}(t)] + [V_1(t) - V_2(t)] \quad (10)$$

Já a malha 2 fornece a equação (11).

$$V_B(t) = V_{L2}(t) + V_{2G1}(t) - V_{3G1}(t) - V_{L3}(t) + V_C(t) \quad (11)$$

A equação (11) pode ser reescrita como na equação (12).

$$[V_B(t) - V_C(t)] = [V_{L2}(t) - V_{L3}(t)] + [V_2(t) - V_3(t)] \quad (12)$$

A malha 3 provê as equações (13) e (14).

$$V_C(t) = V_{L3}(t) + V_{3G1}(t) - V_{1G1}(t) - V_{L1}(t) + V_A(t) \quad (13)$$

$$[V_C(t) - V_A(t)] = [V_{L3}(t) - V_{L1}(t)] + [V_3(t) - V_1(t)] \quad (14)$$

2.1.1 Tensões sobre os indutores

Com o propósito de projetar o filtro LC, faz-se essencial conhecer a expressão da tensão, a fim de identificar a máxima tensão que o indutor deve suportar. Utilizando a lei Kirchhoff das correntes, tem-se a equação (15), onde as várias I_L , I_C e I_R representam, respectivamente, as correntes instantâneas no indutor, no capacitor e no resistor (15), onde as várias I_L , I_C e I_R representam, respectivamente, as correntes instantâneas no indutor, no capacitor e no resistor.

$$\begin{cases} I_{L1}(t) = I_{C1}(t) + I_{R1}(t) \\ I_{L2}(t) = I_{C2}(t) + I_{R2}(t) \\ I_{L3}(t) = I_{C3}(t) + I_{R3}(t) \end{cases} \quad (15)$$

E as tensões entre os indutores L_1 e L_2 , que são expressas pela equação (16).

$$V_{L1}(t) - V_{L2}(t) = L_1 \frac{dI_{L1}(t)}{dt} - L_2 \frac{dI_{L2}(t)}{dt} \quad (16)$$

Os cálculos são efetuados considerando que os indutores, capacitores e resistores tem os mesmos valores entre si, o que pode ser visto nas equações (17), (18) e (19).

$$L = L_1 = L_2 = L_3 \quad (17)$$

$$C = C_1 = C_2 = C_3 \quad (18)$$

$$R = R_1 = R_2 = R_3 \quad (19)$$

Levando em consideração as equações (15), (17), (18) e (19) e aplicando-as na equação (16), tem-se a equação (20).

$$V_{L1}(t) - V_{L2}(t) = V_{L12}(t) = L \frac{d}{dt} \{ [I_{C1}(t) - I_{C2}(t)] + [I_{R1}(t) - I_{R2}(t)] \} \quad (20)$$

Já a equação da corrente nos capacitores é dada pela equação (21).

$$\begin{cases} I_{C1}(t) = C_1 \frac{d}{dt} V_1(t) \\ I_{C2}(t) = C_2 \frac{d}{dt} V_2(t) \end{cases} \quad (21)$$

E a corrente nos resistores pela equação (22).

$$\begin{cases} I_{R1}(t) = \frac{V_1(t)}{R_1} \\ I_{R2}(t) = \frac{V_2(t)}{R_2} \end{cases} \quad (22)$$

Desta forma, pode-se escrever a equação (20) como nas equações (23) e (24).

$$V_{L12}(t) = L \frac{d}{dt} \left\{ \left[C \frac{d}{dt} V_1(t) - C \frac{d}{dt} V_2(t) \right] + \left[\frac{V_1(t)}{R} - \frac{V_2(t)}{R} \right] \right\} \quad (23)$$

$$V_{L12}(t) = LC \frac{d^2}{dt^2} V_{12}(t) + \frac{L}{R} \frac{d}{dt} V_{12}(t) \quad (24)$$

As demais tensões dos indutores são dadas pelas equações (25) e (26).

$$V_{L23}(t) = LC \frac{d^2}{dt^2} V_{23}(t) + \frac{L}{R} \frac{d}{dt} V_{23}(t) \quad (25)$$

$$V_{L31}(t) = LC \frac{d^2}{dt^2} V_{31}(t) + \frac{L}{R} \frac{d}{dt} V_{31}(t) \quad (26)$$

2.2.2 Aplicação da transformada de Clarke

Ao considerar a variável de monitoramento da tensão de linha, é introduzida a expressão da tensão do indutor, com o sistema operando a vazio, chega-se as equações (27), (28) e (29).

$$V_{AB}(t) = LC \frac{d^2}{dt^2} V_{12}(t) + \frac{L}{R} \frac{d}{dt} V_{12}(t) + V_{12}(t) \quad (27)$$

$$V_{BC}(t) = LC \frac{d^2}{dt^2} V_{23}(t) + \frac{L}{R} \frac{d}{dt} V_{23}(t) + V_{23}(t) \quad (28)$$

$$V_{CA}(t) = LC \frac{d^2}{dt^2} V_{31}(t) + \frac{L}{R} \frac{d}{dt} V_{31}(t) + V_{31}(t) \quad (29)$$

Tem-se que as tensões de linha podem ser escritas como nas equações (30), (31) e (32).

$$V_{AB}(t) = VI * d_{AB}(t) \quad (30)$$

$$V_{BC}(t) = VI * d_{BC}(t) \quad (31)$$

$$V_{CA}(t) = VI * d_{CA}(t) \quad (32)$$

Tais tensões podem ser representadas em forma de matriz, como se encontra nas equações (33) e (34).

$$\begin{aligned}
V_I \begin{pmatrix} d_{AB}(t) \\ d_{BC}(t) \\ d_{CA}(t) \end{pmatrix} &= LC \frac{d^2}{dt^2} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_{12}(t) \\ V_{23}(t) \\ V_{31}(t) \end{pmatrix} + \frac{L}{R} \frac{d}{dt} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_{12}(t) \\ V_{23}(t) \\ V_{31}(t) \end{pmatrix} \\
&+ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_{12}(t) \\ V_{23}(t) \\ V_{31}(t) \end{pmatrix}
\end{aligned} \tag{33}$$

$$V_I \begin{pmatrix} d_{AB}(t) \\ d_{BC}(t) \\ d_{CA}(t) \end{pmatrix} = LC \frac{d^2}{dt^2} [V]_{123} + \frac{L}{R} \frac{d}{dt} [V]_{123} + [V]_{123} \tag{34}$$

Da definição da transformação $0\infty\beta$, obtém-se a equação (35) (HAUSMANN et al., 2008, p. 20).

$$[X]_{0\infty\beta} = [A]^{-1} [X]_{123} \tag{35}$$

Sendo a inversa dada pela equação (36).

$$[X]_{123} = [A]^{-1} [X]_{0\infty\beta} \tag{36}$$

Onde a matriz $[A]^{-1}$ encontra-se expressa pela equação (37).

$$[A]^{-1} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 1 & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{pmatrix} \tag{37}$$

Como a matriz A pode ser encontrada pela equação (38), aplicada equação (34), chega-se à equação (39).

$$[A] = \{[A]^{-1}\}^T \tag{38}$$

$$V_I [A] [d]_{0\infty\beta} = LC \frac{d^2}{dt^2} \{[A] [V]_{0\infty\beta}\} + \frac{L}{R} \frac{d}{dt} \{[A] [V]_{0\infty\beta}\} + [A] [V]_{0\infty\beta} \tag{39}$$

Mas, pode ser escrita como a equação (40).

$$\begin{cases} \frac{d}{dt}\{[A][V]_{0\infty\beta}\} = [A]\frac{d}{dt}[V]_{0\infty\beta} + [V]_{0\infty\beta}\frac{d}{dt}[A] \\ \frac{d^2}{dt^2}\{[A][V]_{0\infty\beta}\} = [A]\frac{d^2}{dt^2}[V]_{0\infty\beta} + [V]_{0\infty\beta}\frac{d^2}{dt^2}[A] \end{cases} \quad (40)$$

Como os termos da matriz da transformada de Clarke são constantes, conclui-se que a derivada desta matriz é nula. Portanto, a reescrita da equação (40); resulta na expressão da equação (41) (HAUSMANN et al., 2008, p. 21).

$$\begin{cases} \frac{d}{dt}\{[A][V]_{0\infty\beta}\} = [A]\frac{d}{dt}[V]_{0\infty\beta} \\ \frac{d^2}{dt^2}\{[A][V]_{0\infty\beta}\} = [A]\frac{d^2}{dt^2}[V]_{0\infty\beta} \end{cases} \quad (41)$$

Substituindo equação (41) na equação (39), tem-se a equação (42).

$$V_I[A][d]_{0\infty\beta} = LC[A]\frac{d^2}{dt^2}[V]_{0\infty\beta} + \frac{L}{R}[A]\frac{d}{dt}[V]_{0\infty\beta} + [A][V]_{0\infty\beta} \quad (42)$$

Ao multiplicar ambos os lados por $[A]^{-1}$ tem-se a equação (43).

$$\begin{aligned} & V_I[A]^{-1}[A][d]_{0\infty\beta} \\ &= LC[A]^{-1}[A]\frac{d^2}{dt^2}[V]_{0\infty\beta} + \frac{L}{R}[A]^{-1}[A]\frac{d}{dt}[V]_{0\infty\beta} + [A]^{-1}[A][V]_{0\infty\beta} \end{aligned} \quad (43)$$

Sabe-se ainda que o produto de uma matriz pela sua inversa resulta na matriz identidade, conforme equação (44).

$$[A]^{-1}[A]=I \quad (44)$$

Substituindo a equação (44) na equação (43), tem-se a equação (45).

$$V_I[d]_{0\infty\beta} = LC\frac{d^2}{dt^2}[V]_{0\infty\beta} + \frac{L}{R}\frac{d}{dt}[V]_{0\infty\beta} + [V]_{0\infty\beta} \quad (45)$$

Onde os termos $[d]_{0\infty\beta}$, $[V]_{0\infty\beta}$ são dados pelas equações (46) e (47).

$$[d]_{0\infty\beta} = [A]^{-1}[d]_{123} \quad (46)$$

$$[V]_{0\infty\beta} = [A]^{-1}[V]_{123} \quad (47)$$

E, ao aplicar a transformada de Laplace na equação (45), obtém-se as equações (48) e (49).

$$V_I[d]_{0\infty\beta} = s^2 LC[V]_{0\infty\beta} + s \frac{L}{R} [V]_{0\infty\beta} + [V]_{0\infty\beta} \quad (48)$$

$$[V]_{0\infty\beta} = \frac{[V]_I[d]_{0\infty\beta}}{s^2 LC + \frac{sL}{R} + 1} \quad (49)$$

A equação (49) pode ser reescrita e chega-se à equação (50).

$$[V]_{0\infty\beta} = V_I Z(s) [d]_{0\infty\beta} \quad (50)$$

Onde $Z(s)$ é dado pela equação (51).

$$Z(s) = \frac{1}{s^2 LC + \frac{sL}{R} + 1} \quad (51)$$

A equação (50) representa o desacoplamento das variáveis do sistema. Assim, as componentes das tensões de linha podem ser representadas através das coordenadas $0\alpha\beta$, permitindo elaborar um controle do inversor a partir destas condições (HAUSMANN et al., 2008, p. 22). A função de transferência do inversor é dada pela equação (52) (HAUSMANN et al., 2008, p. 16).

$$G(s) = \frac{V_I}{V_{TRIPK}} = \frac{1}{s^2 LC + \frac{sL}{R} + 1} \quad (52)$$

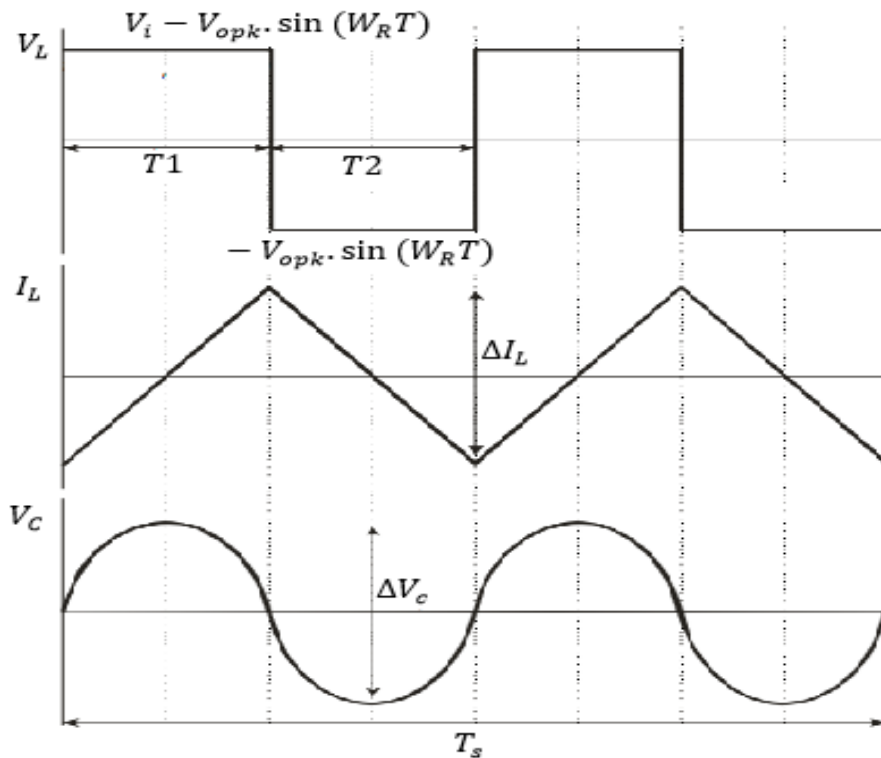
2.3 Filtro LC, CI FNA41060B2 e cálculo térmico

2.3.1 Filtro LC

A implementação do filtro LC faz com que a tensão imposta sobre a carga contenha somente a componente fundamental da tensão de referência. Ao projetar o filtro é essencial que este atenda as condições críticas de trabalho. Para isto, o indutor deve suportar a máxima ondulação de corrente e o capacitor, a máxima ondulação de tensão. A Figura 7 representa a forma simplificada da tensão no indutor, a ondulação de corrente e a sinuosidade de tensão no capacitor.

Considera-se para o dimensionamento do filtro o dobro da frequência de comutação, pois através das equações diferenciais, nota-se que as expressões fazem referência às variáveis de linha. Portanto, como abordado anteriormente, as componentes de linha apresentam o dobro da frequência de comutação, o que será determinante no projeto do filtro LC (HAUSMANN e al., 2008, p. 23).

Figura 7 – Variação da tensão sobre o capacitor e corrente e tensão no indutor



Fonte: Hausmann e outros (2008).

2.3.1.1 CÁLCULO DO INDUTOR

Sabe-se que a frequência de comutação dos componentes de linha é dada pela equação (53).

$$T_s^* = \frac{T_s}{2} \quad (53)$$

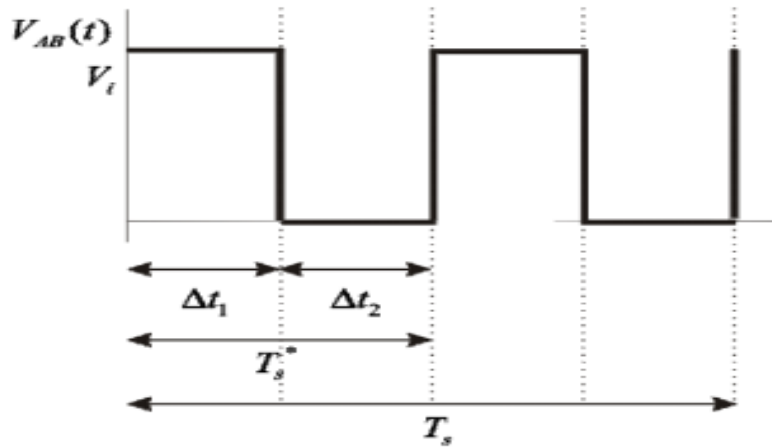
E a tensão V_A pode ser escrita através das equações (54) e (55).

$$V_A(t) = [V_{L1}(t) - V_{L2}(t)] + [V_{1G1}(t) - V_{2G1}(t)] + V_B(t) \quad (54)$$

$$\begin{cases} [V_A(t) - V_B(t)] = [V_{L1}(t) - V_{L2}(t)] + [V_1(t) - V_{G1}(t) - V_2(t) + V_{G1}(t)] \\ V_{AB}(t) = V_{L12}(t) + V_{12}(t) \end{cases} \quad (55)$$

A tensão V_{AB} representa a tensão de linha entre V_A e V_B , cuja frequência de comutação é duas vezes maior que a tensão de fase. A Figura 8 representa a forma simplificada de V_{AB} .

Figura 8 – Tensão $V_{AB}(t)$ para o semiciclo positivo da moduladora



Fonte: Hausmann e outros (2008).

A tensão sobre o indutor é apresentada na equação (56).

$$V_L = L \frac{dI_L}{dt} \quad (56)$$

Utilizando a Figura 8 tem-se que, durante o intervalo t_1 , a tensão no indutor é definida pela equação (57), onde V_I é a tensão barramento CC, $V_{OPK} \sin(w_R t)$ é a tensão sobre a carga. Já intervalo t_2 , tal tensão é dada pela equação (58).

$$L \frac{dI_{L12}(t)}{dt} = V_I - V_{OPK} \sin(w_R t) \quad (57)$$

$$L \frac{dI_{L12}(t)}{dt} = -V_{OPK} \sin(w_R t) \quad (58)$$

As equações (57) e (58) podem ser reescritas, como mostrado na equação (59).

$$\begin{cases} L \frac{\Delta I_{L12}(t)}{\Delta t_1} = V_I - V_{OPK} \sin(w_R t) \\ L \frac{\Delta I_{L12}(t)}{\Delta t_2} = -V_{OPK} \sin(w_R t) \end{cases} \quad (59)$$

Isolando as variáveis Δt_1 e Δt_2 , tem-se as equações (60) e (61).

$$\Delta t_1 = \frac{L \Delta I_{L12}(t)}{V_I - V_{OPK} \sin(w_R t)} \quad (60)$$

$$\Delta t_2 = -\frac{L \Delta I_{L12}(t)}{V_{OPK} \sin(w_R t)} \quad (61)$$

É sabido que a metade do período comutação é soma de Δt_1 mais Δt_2 , como mostrado na equação (62).

$$\frac{T_s}{2} = T_s^* = \Delta t_1 + \Delta t_2 \quad (62)$$

Ao substituir as equações (60) e (61) na equação (62), chega-se à equação (63).

$$\frac{T_s}{2} = \frac{L \Delta I_{L12}(t)}{V_I - V_{OPK} \sin(w_R t)} - \frac{L \Delta I_{L12}(t)}{V_{OPK} \sin(w_R t)} \quad (63)$$

Com o objetivo de encontrar a oscilação sobre o indutor, isola-se o termo Δi_{L12} , como mostrado na equação (64).

$$\Delta I_{L12} = \frac{V_I V_{OPK} \sin(w_R t) - V_{OPK}^2 \sin(w_R t)^2}{2 f_s L V_I} \quad (64)$$

A fim de facilitar o entendimento, a equação (64) é parametrizada, como mostrado na equação (65).

$$\overline{\Delta I L f} = \frac{2 L \Delta I L f(t)}{T_s V_I} = \frac{V_{OPK}}{V_I} \left[\sin(w_R t) - \frac{V_{OPK} \sin(w_R t)^2}{V_I} \right] \quad (65)$$

Com o intuito de encontrar o ponto de máxima ondulação de corrente, deve-se encontrar o ponto de máximo global, calculando a derivada da equação (65) e igualando a zero, o que resulta na equação (66).

$$\frac{d \Delta I_{L12}}{dt} = -\cos(w_R t) - \frac{2 V_{OPK} \sin(w_R t) \cos(w_R t)}{V_I} = 0 \quad (66)$$

Implicando desta forma no seguinte resultado, visualizado na equação (67) (HAUSMANN et al., 2008, p. 26).

$$\frac{d \Delta I_{L12}}{dt} = \begin{cases} \frac{1}{4}; V_{OPK} \leq \frac{V_I}{2} \\ \frac{V_{OPK}}{V_I} \left(1 - \frac{V_{OPK}}{V_I} \right); V_{OPK} > \frac{V_I}{2} \end{cases} \quad (67)$$

Sendo que a ondulação no indutor é definida por meio da equação (68) (HAUSMANN et al., 2008, p. 26).

$$\Delta I_{L12} = \begin{cases} \frac{V_I}{8 f_s L}; V_{OPK} \leq \frac{V_I}{2} \\ \frac{V_{OPK}}{2 f_s L} \left(1 - \frac{V_{OPK}}{V_I} \right); V_{OPK} > \frac{V_I}{2} \end{cases} \quad (68)$$

Pode-se também evidenciar a indutância que, neste caso, fornece a equação (69).

$$L = \begin{cases} \frac{VI}{8f_s \Delta i_{L12}}; V_{OPK} \leq \frac{VI}{2} \\ \frac{V_{OPK}}{2f_s \Delta I_{L12}} \left(1 - \frac{V_{OPK}}{VI}\right); V_{OPK} > \frac{VI}{2} \end{cases} \quad (69)$$

2.3.1.2 CAPACITOR DE FILTRAGEM

A capacitância do filtro deve atender as condições críticas, sendo assim, o capacitor deve suportar a máxima ondulação de tensão, que está ligada à máxima ondulação de corrente sobre indutor. Considerando que todas as componentes alternadas de frequência elevada circulem sobre o capacitor, pode-se calcular a ondulação de tensão como sendo a variação de carga que passa por ele, como mostrado nas equações (70) e (71).

$$\Delta Q = C \Delta V_c \quad (70)$$

Sabendo-se que, a carga elétrica pode ser encontrada pela equação (71).

$$Q = \frac{1}{2} I t \quad (71)$$

A variação de carga pode ser escrita pela equação (72).

$$\Delta Q = \frac{1}{2} \frac{\Delta I_L}{2} \frac{T_s^*}{2} \quad (72)$$

Substituindo a equação (70) na equação (72), tem-se a equação (73).

$$\frac{1}{8} \frac{\Delta I_{L12}}{f_s^*} = C \Delta V_c \quad (73)$$

Assim, a máxima ondulação de tensão imposta ao capacitor é encontrada substituindo a equação (68) na equação (72), resultando na equação (74) (HAUSMANN et al., 2008, p. 27).

$$\Delta V_c = \begin{cases} \frac{1}{128} \frac{V_I}{f_s^2 LC}; V_{opk} \leq \frac{V_I}{2} \\ \frac{1}{16} \frac{V_{opk}}{f_s^2 L \Delta V_c} \left(1 - \frac{V_{opk}}{V_I}\right); V_{opk} > \frac{V_I}{2} \end{cases} \quad (74)$$

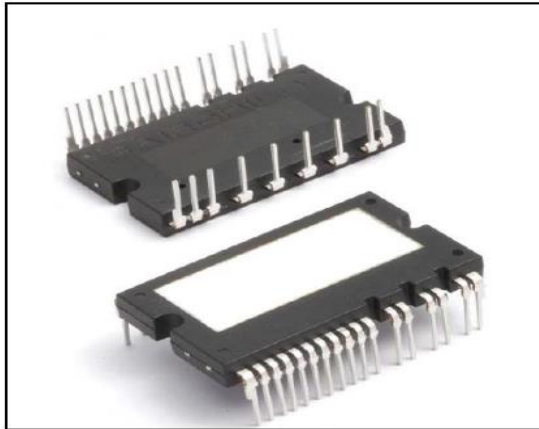
Reescrevendo a equação (74), encontra-se a função que define o valor da capacitância a ser utilizado, como pode ser visto na equação (75).

$$C = \begin{cases} \frac{1}{128} \frac{V_I}{f_s^2 L \Delta V_c}; V_{opk} \leq \frac{V_I}{2} \\ \frac{1}{16} \frac{V_{opk}}{f_s^2 L \Delta V_c} \left(1 - \frac{V_{opk}}{V_I}\right); V_{opk} > \frac{V_I}{2} \end{cases} \quad (75)$$

2.3.2 CI comercial FNA41060B2

Para implementação do protótipo é utilizado o CI (circuito integrado) FNA41060B2, pois facilita a montagem, diminui as conexões por solda. A Figura 9 mostra o módulo usado.

Figura 9 – CI do módulo inversor

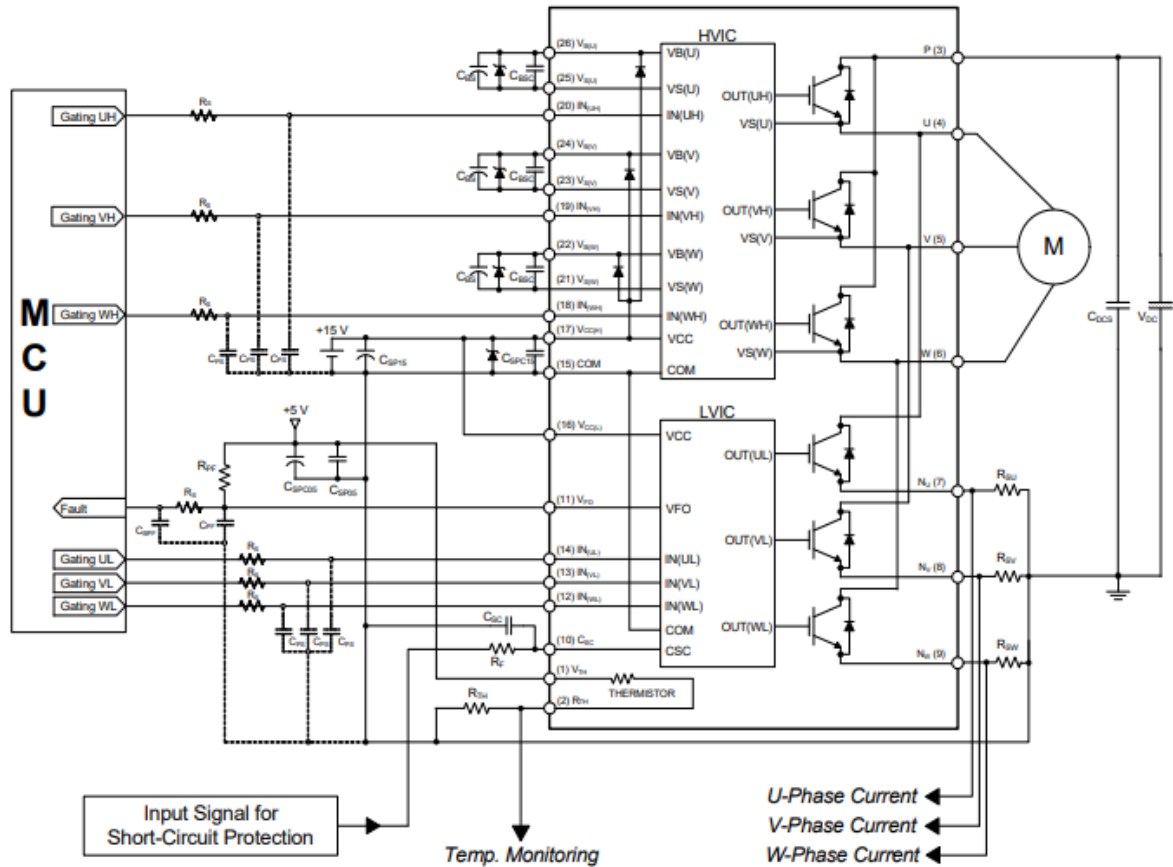


Fonte: Fairchild Semiconducto (2011).

O CI é formado por 6 transistores IGBTs, os quais suportam 600 Volts e 10 Amperes de corrente, sendo sua frequência máxima de comutação de 20 kHz. O dispositivo pode ser alimentado com sinais de 3,3 ou 5 Volts. O CI possui mecanismos de proteção, por meio do desligamento por sobrecorrente, além de um sistema de medição de temperatura para prevenir

sobrecarga. O fabricante disponibiliza o circuito de acionamento do módulo, conforme mostrado na Figura 10.

Figura 10 – Circuito de acionamento do FNA41060B2



Fonte: Fairchild Semiconducto (2011).

É possível observar na Figura 10 que as entradas dos sinais de disparos dos seis IGBTs, pinos 20, 19, 18, 14, 13 e 12, passam por um filtro RC, sendo o valor de $R=100\ \Omega$ e o de $C=10\ \text{nF}$, conforme o recomendado pelo fabricante. O objetivo deste filtro é evitar oscilações indesejadas no disparo de entrada.

No cálculo do capacitor *bootstrap*, o *datasheet* recomenda os valores para I_{LEAK} de 2 mA e V_{BS} de 0,1 V, o valor de Δt depende da frequência de comutação. Assim, a equação (76) fornece o valor do capacitor *bootstrap*.

$$C_{Bs_min} = \frac{I_{LEAK} \Delta t}{\nabla V_{BS}} \quad (76)$$

2.3.3 Dimensionamento térmico

Uma vez escolhido o CI FNA41060b2, agora tem-se que garantir a operação de trabalho recomendada pelo fabricante, como temperatura, tensão de operação e frequência de comutação. O trabalho fora da temperatura adequada pode causar desde mal funcionamento até danos irreversíveis ao componente. Haja vista que, nessas condições, o CI encontra-se impossibilitado de dissipar o excesso de calor, o qual resulta em seu acúmulo, elevando sua temperatura.

O aumento da temperatura é causado pelas perdas nos semicondutores, logo, ao conhecer a corrente que circula por cada um destes, pode-se estimar as perdas durante a condução e, através da frequência de chaveamento, são calculadas as perdas por comutação. Com os valores obtidos, pode ser feito o cálculo do dissipador, o qual tem o papel de facilitar a troca de calor entre o CI e o ambiente externo. Para o cálculo, será utilizado o circuito térmico, o qual pode ser visto na Figura 11.

Figura 11 – Circuito térmico equivalente



Fonte: Knaesel (2016).

As grandezas vistas no circuito são:

T_j – Temperatura de junção (°C).

T_c – Temperatura da cápsula (°C).

T_d – Temperatura do dissipador (°C).

T_a – Temperatura do ambiente (°C).

P – Potência térmica produzida pela corrente que circula no componente sendo transferida ao meio (W).

R_{jc} – Resistência térmica entre a junção e a cápsula (°C/W).

R_{cd} – Resistência térmica entre a cápsula e dissipador ($^{\circ}\text{C}/\text{W}$).

R_{da} – Resistência térmica entre dissipador e ambiente ($^{\circ}\text{C}/\text{W}$).

R_{ja} – Resistência térmica entre a junção e o ambiente ($^{\circ}\text{C}/\text{W}$).

Desta forma, a resistência térmica entre a junção e o ambiente é dada pela equação (77).

$$R_{ja} = \frac{T_j - T_a}{P} \quad (77)$$

O valor de P é calculado a partir das características do componente, as quais são informadas no *datasheet*, e a corrente é obtida via simulação.

As perdas devido à condução são dadas pela equação (78).

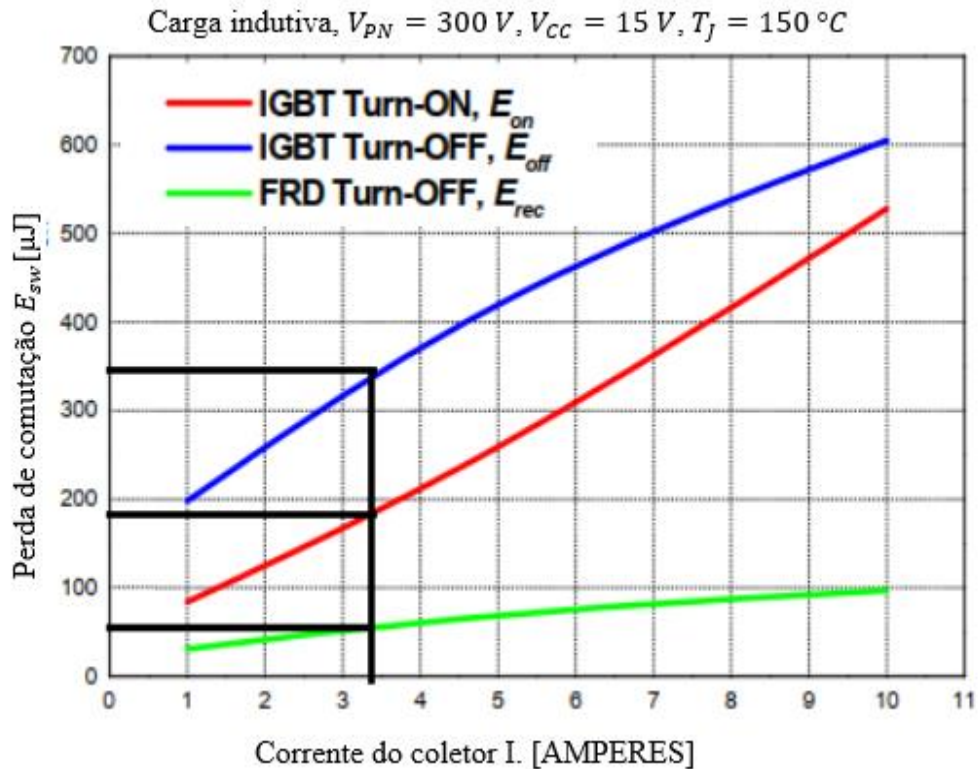
$$P_{\text{cond}}^{\text{IGBT}} = (1,8 \text{ V}) \cdot (1,55 \text{ A}) = 2,79 \text{ W} \quad (78)$$

1.3.3.1 PERDAS POR COMUTAÇÃO

O gráfico apresentado na Figura 12 traz, em função da corrente do coletor, o valor aproximado da energia dissipada durante um ciclo. Desta forma, torna-se possível encontrar a potência dissipada, através da equação (79).

$$\begin{cases} P_{\text{comuta}}^{\text{IGBT}} = \frac{(E_{\text{ON}} + E_{\text{OFF}} + E_{\text{REC}})}{T_{\text{COMU}}} = (E_{\text{ON}} + E_{\text{OFF}} + E_{\text{REC}}) * F_{\text{COMU}_c} \\ P_{\text{comuta}}^{\text{IGBT}} = (350\mu + 190\mu + 60\mu) * 20\text{k} = 12\text{W} \end{cases} \quad (79)$$

Figura 12 – Perdas causadas pela comutação nos IGBTs



Fonte: Fairchild Semiconducto (2011).

Assim sendo, tem-se que a soma das perdas de comutação e condução é dada na equação (80).

$$P_{\text{total}}^{\text{IGBT}} = 14,79\text{ W} \quad (80)$$

2.3.3.2 RESISTÊNCIA TÉRMICA ENTRE A JUNÇÃO E A CÁPSULA, E ENTRE A CÁPSULA E O DISSIPADOR

A resistência térmica entre a junção e a cápsula é fornecida no *datasheet*, como mostrado na equação (81).

$$R_{jc} = 3,6 \frac{^{\circ}\text{C}}{\text{W}} \quad (81)$$

Entretanto, este valor é fornecido para uma condição específica, onde a duração do pulso é maior do que 0,2 segundo, de modo que a utilização do gráfico mostrado na Figura 13 se faz necessário. Assim, faz-se uma aproximação da impedância térmica em função da duração do pulso. Para isso, emprega-se uma frequência de comutação de 20 kHz, o que corresponde a um

período de 50 μ s, lembrando que o sinal de controle possui uma razão cíclica de 50% portanto, a duração do pulso passa a ser de 25 μ s. Dessa forma, em acordo com o gráfico da Figura 13, tem-se que a resistência térmica aproximada é dada pela equação (82).

$$R_{jc} = 0,1 \frac{^{\circ}\text{C}}{\text{W}} \quad (82)$$

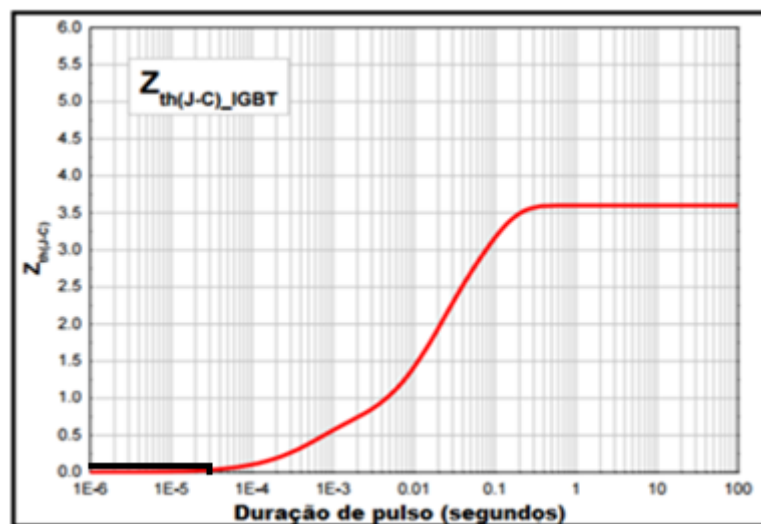
A resistência térmica entre a cápsula e o dissipador é determinada pelo projetista. Desta forma, é utilizado o valor expresso na equação (83) (KNAESEL, 2016, p. 119).

$$R_{cd} = 1,0 \frac{^{\circ}\text{C}}{\text{W}} \quad (83)$$

Ao considerar a temperatura de junção e a temperatura ambiente como sendo 150°C e 40°C, respectivamente, pode-se calcular a resistência térmica entre a junção e a cápsula, por meio da equação (84).

$$R_{ja} = \frac{T_j - T_a}{P} = \frac{150 - 40}{14,79} = 7,44 \frac{^{\circ}\text{C}}{\text{W}} \quad (84)$$

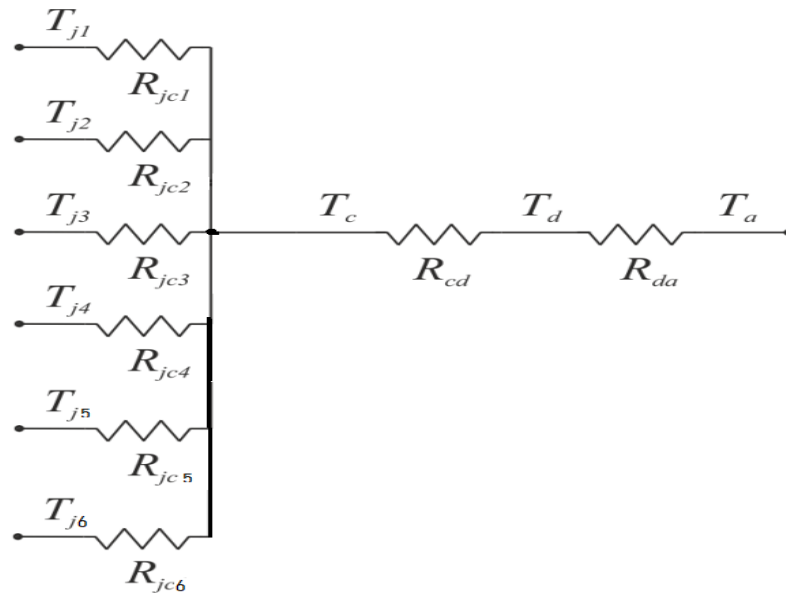
Figura 13 – Impedância térmica entre a junção e a cápsula do IGBT



Fonte: Fairchild Semiconducto (2008).

Devido ao fato de a resistência térmica calculada ser de apenas um IGBT, para o cálculo da resistência térmica entre o dissipador e o ambiente, é necessário considerar que existem seis IGBTs como fonte de calor, como mostrado na Figura 14.

Figura 14 – Circuito térmico equivalente para regime de operação



Fonte: Produção do próprio autor.

A fim de definir a resistência térmica entre o dissipador e meio ambiente, deve-se isolar este termo, tendo assim a equação (85).

$$R_{da} = \frac{7,44 - 1 - 0,1}{6} = 1,06^{\circ} \frac{C}{W} \quad (85)$$

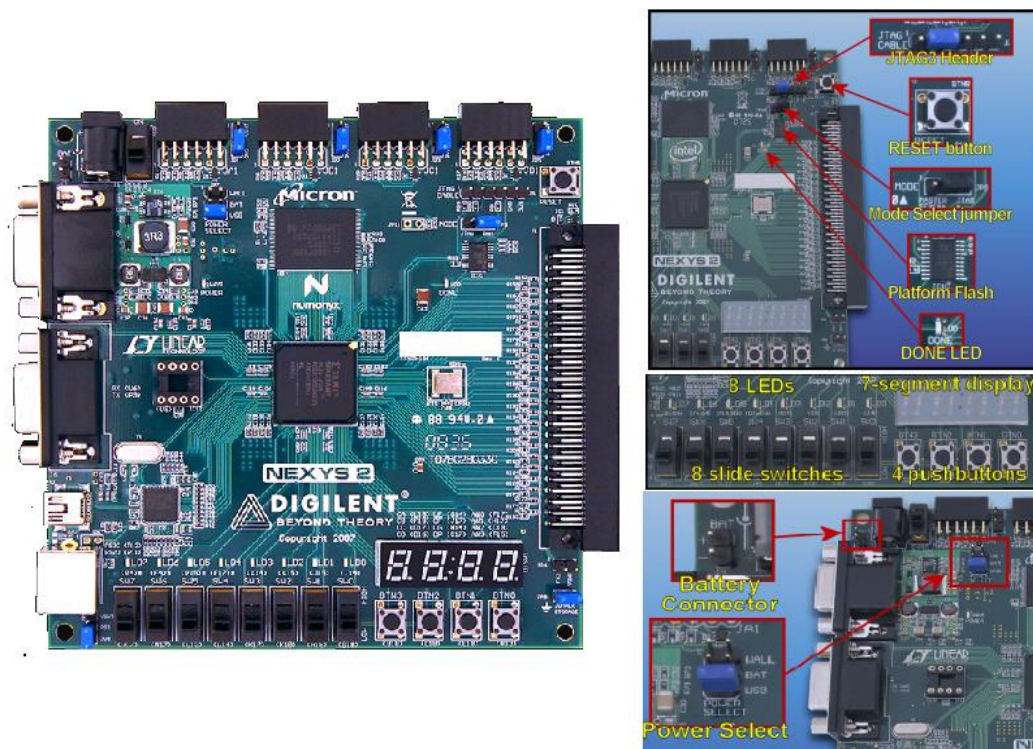
Portanto, $R_{da} \leq 1,06^{\circ} \frac{C}{W}$.

3 KIT FPGA NEXYS2

Um FPGA (do inglês *Field Programmable Gate Array*) é um circuito integrado o qual pode ser reprogramado a nível de porta lógica. A vantagem do FPGA frente a outros sistemas de controle encontra-se no fato de que a reconfiguração das portas modifica o seu funcionamento. Assim, para o desenvolvimento de protótipo este é muito útil, pois, não é necessário o desenvolvimento circuitos integrados de aplicação específica ASIC (do inglês *Application Specific Integrated Circuit*).

A fim de gerar os pulsos por meio da modulação por largura de pulso com referência senoidal SPWM (do inglês *Sinusoidal Pulse Width Modulation*); foi escolhido o kit Nexys2, ilustrado na Figura 15, devido à sua disponibilidade no Laboratório de Sistemas Embarcados do DEL/UFES.

Figura 15 – Kit Digilent Nexys2



Fonte: Digilent (2008).

Suas principais características são:

- É baseado FPGA *Xilinx Spartan 3E*, a qual é otimizada para as aplicações onde a densidade lógica importa mais do que a contagem de E / S. “Ideal para integração lógica, processamento DSP e controle incorporado, exigindo processamento significativo e interfaces estreitas ou poucas” (DIGILENT, 2008);
- Possui 16 MB de PSDRAM e 16 MB de ROM Intel *StrataFlash*; Intel *StrataFlash* armazena dois ou mais *bits* de informação por célula, isso é possível pois o armazenamento é feito com mais de dois níveis de tensão (KINGSTON, 2009);
- Plataforma *Xilinx Flash* para configurações FPGA não voláteis;
- Porta USB 2.0 para programação, que também pode ser usada para alimentar a placa Nexys2;
- Oscilador de 50 MHz mais soquete para o segundo oscilador:

Os sinais de *clock* dos osciladores se conectam aos pinos de entrada de *clock* globais no FPGA para que eles possam acionar os blocos do sintetizador de *clock* disponíveis no FPGA. Os sintetizadores de *clock* (chamados DLLs ou *loops* bloqueados por atraso) fornecem recursos de gerenciamento de *clock* que incluem duplicação ou quadruplicação da frequência de entrada, dividindo a frequência de entrada por qualquer múltiplo inteiro e definindo relações precisas de fase e atraso entre vários sinais de *clock* (XILINX, 2009, p. 13).

- 60 E/S (pinos de entrada e saída) direcionados para conectores de expansão; um conector *Hirose FX2* de alta velocidade e quatro conectores de 6 pinos;
- 8 LEDs, *display* de 7 segmentos de 4 dígitos, 4 botões, 8 interruptores deslizantes;
- Porta VGA de 8 *bits*;
- Inclui cabo USB de programação.

A escrita é realizada no *Suíte design ISE*, sendo instalado juntamente ao ISE o *ISim*, onde é possível simular a linguagem de descrição de *hardware*.

O ISE® Design Suite é o ambiente de *design* da Xilinx®, que permite levar seu *design* desde a entrada do *design* até a programação do dispositivo Xilinx. Com edições específicas para projetistas de sistemas lógicos, processadores embarcados ou Processamento Digital de Sinal (DSP - Digital *Signal Processing*) (XILINX, 2009, p. 1).

3.1 Utilizando a Linguagem VHDL

A linguagem VHDL (do inglês *Hardware Description Language*) foi criada para a descrição de *hardware*, iniciada em 1980 pelo departamento defesa EUA com intuito de substituir os complexos manuais ASIC (do inglês *Application Specific Integrated Circuit*) (MARCON;

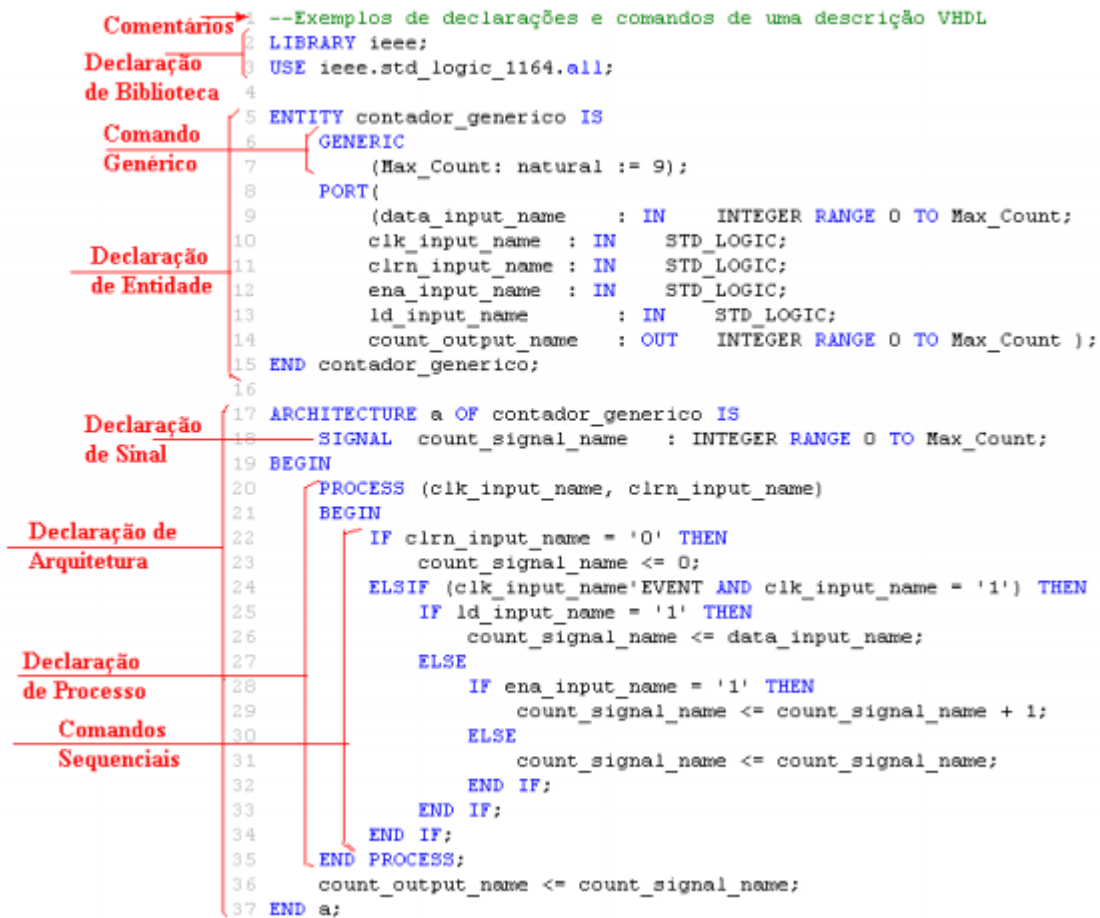
MORENO, 2016). Dos anos de 1983 a 1985, o desenvolvimento da linguagem foi feito pelas empresas Texas Instruments, Intermetrics e IBM. Em 1986 os direitos foram transferidos para o IEEE (do inglês *Institute of Electrical and Electronics Engineers*), tendo sua padronização ocorrendo em 1987, e posteriormente revisado em 1993. A padronização promove uma comunicação entre as partes digital e analógica de maneira eficiente. Assim sendo, o uso do VHDL oferece vantagens, como:

- O projeto pode ser compilado em equipamentos da indústria, pois é comum uso de FPGAs e CPLDs (do inglês *Complex Programmable Logic Device*) .
- “Em sistemas sequenciais, o detalhamento da lógica de controle é realizado pelas ferramentas de automação do projeto, o que evita a trabalhosa e limitada aplicação das técnicas manuais tradicionais” (GIACOMINI; RENATO, 2013, p. 3).
- “O volume de documentação diminui, já que um código bem comentado em VHDL substitui com vantagens o esquemático e a descrição funcional do sistema” (GIACOMINI; RENATO, 2013, p. 3).
- Reduz o tempo de projeto e elimina erros de baixo nível.

3.2 Modelagem de componentes VHDL

A primeira atitude a se tomar é definir as características de interface, ou seja, as entradas e saídas dos componentes, com os modos de sinal *in* e *out*, respectivamente. Feito isso, com o código aberto são definidas as bibliotecas utilizadas, faz-se a identificação das entradas e saídas, dos atributos e, por fim, a descrição do comportamento do circuito. Para facilitar o entendimento deve-se observar as informações da Figura 16.

Figura 16 – Programa de um contador destacando a estrutura básica projeto em VHDL



Fonte: Codá (2014).

A declaração da biblioteca segue o padrão `Library <nome da biblioteca>` e pode-se incluir todos os componentes da mesma, usando `use <nome da biblioteca>.all`.

A declaração genérica é uma declaração opcional que possibilita passar informações externas para entidades de projeto. O emprego do `GENERIC` possibilita a reconfiguração de um circuito pela simples alteração de seus valores, sem alterar o código do projeto deixando-o genérico (CODÁ, 2014, p. 45).

Ainda dentro da entidade do circuito, são definidas as entradas e saídas, assim como seu tipo. Dentro de arquitetura são definidos os objetos, que podem ou não ter um valor armazenado. Eles podem ser de três tipos: *signal*, *variable* e *constant*. “*Signal* representa sinais lógicos sobre um fio no circuito, os quais interligam componentes. Um sinal não tem memória, portanto se a fonte do sinal é removida, o sinal não terá um valor” (CODÁ, 2014, p. 45). Já a *variable*, é utilizada em processos nos quais são atualizadas imediatamente e não correspondem à implementação física, enquanto a *constant* pode ser declarada em *package*, *entity*, *architecture*

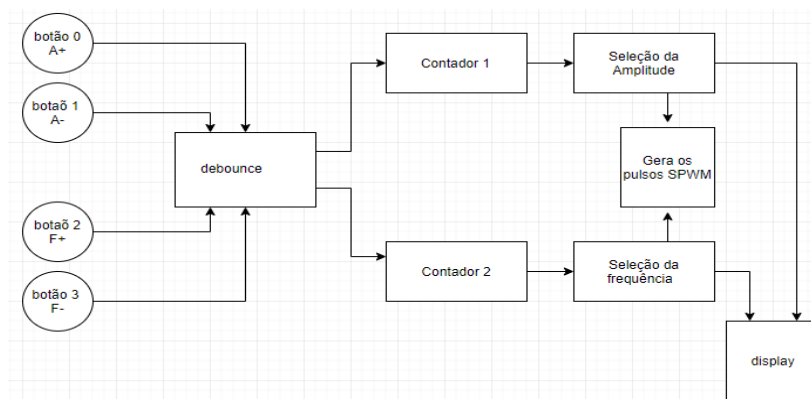
e *process*, sendo validada no contexto onde é declarada. Seu valor, uma vez declarado, não pode ser alterado.

Dentro da estrutura, pode ser utilizado o *component*, o qual corresponde à descrição de um componente. Sua instanciação permite, via *port map*, mapear as entradas e saídas deste componente em um sistema maior.

O *process*, por sua vez, vem seguido de uma lista de sensibilidade, cuja a mudança dessas variáveis ou sinais deve executar o código interno a ele de maneira sequencial. Esta é única parte do código em que isto ocorre, sendo que as demais são executadas em paralelo.

A fim de implementar a modulação SPWM em VHDL, o circuito foi dividido em alguns componentes para facilitar tanto a lógica quanto o entendimento. São eles: **debounce**, **contador**, **disp0e1**, **selecaoFrequencia** e **selecaoAmplitude**. O diagrama de funcionamento entre eles está mostrado na Figura 17.

Figura 17 – Diagrama de funcionamento do código implementado

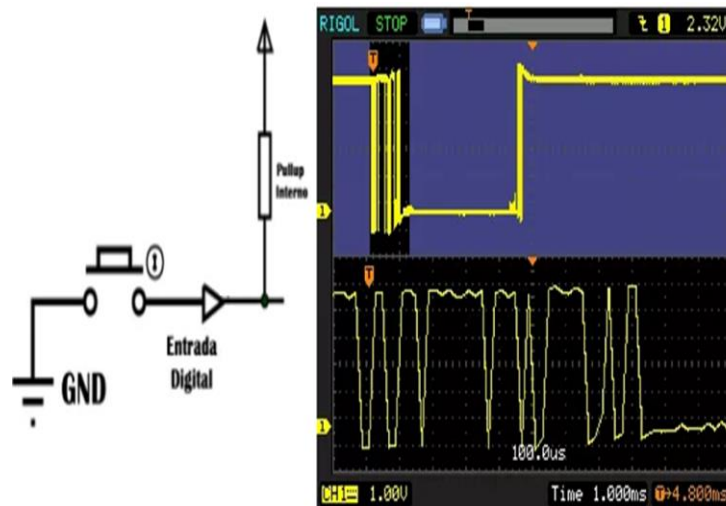


Fonte: Produção do próprio autor.

Debounce se faz necessário, haja vista que as chaves mecânicas ao serem pressionadas oscilam algumas vezes entre circuito aberto e fechado, o que causa mudança no nível lógico. O ruído causado por este efeito é chamado *bouncing*, o qual pode ser visualizado na Figura 18, e a sua filtragem ou eliminação é chamada *debouncing*. Para que não ocorram erros na contagem de apertos do botão, utilizou-se um *flip-flop* responsável por armazenar o estado anterior (entenda,

o estado no pulso do *clock* anterior). Para contar o tempo, é usado um contador de 19 bits. Caso o valor do botão for diferente do valor do pulso do *clock* anterior é reiniciada a contagem; caso contrário, o contador é incrementado. Portanto, quando o último *bit* for 1, já se passou o tempo suficiente para considerar o pulso estável. Neste caso, o valor é conectado a saída do circuito.

Figura 18 – Efeito *bouncing*



Fonte: Mota (2017).

O **contador** recebe o sinal após passar pelo filtro do *debouncing*. Como cada *button* é uma entrada diferente, deve-se somar 1 ao número já existente ou retirar, dependendo de qual *button* foi pressionado. Caso o número ultrapasse os limites inferior ou superior, deve-se prevalecer o valor superior ou inferior especificado. Deve-se lembrar que tanto o contador para a frequência quanto para amplitude da senoide funcionam de forma idêntica.

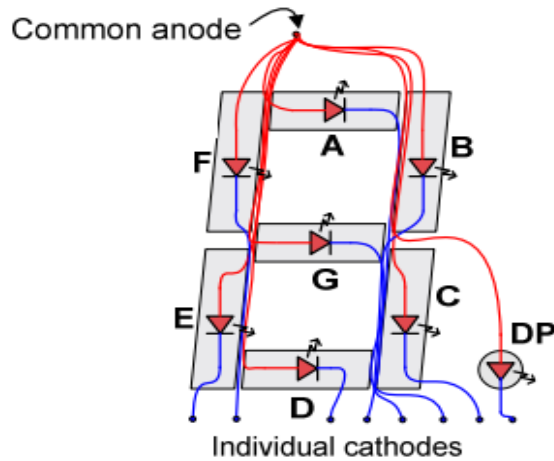
O **disp0e1** tem como função apresentar ao usuário a frequência e o percentual da amplitude da senoide na qual o inversor está trabalhando. Da esquerda para a direita, os dois primeiros dígitos informam a frequência e os dois últimos, o percentual de tensão nominal aplicada à carga.

Para entender adequadamente o funcionamento do bloco *disp0e1* é indicado conhecer o funcionamento do *display* de sete segmentos.

Uma vez que o número é recebido, este deve estar em código BCD (do inglês *Binary Code Decimal*) de quatro *bits* (D0, D1, D2 e D3) e gerar uma saída (a, b, c, d, e, f, g); o nível alto ou

baixo aciona o segmento do *display*. Na Figura 19 pode-se observar a estrutura interna do *display*.

Figura 19 – Estrutura *display* anodo comum



Fonte: Digilent (2008).

Para valores decimais de 0 a 9, o decodificador BCD deve gerar a sequência de *bits* correspondente. Por exemplo, para o número 0 o decodificador BCD deve gerar a seguinte saída 11000000. A tabela verdade do *display* de sete segmentos anodo comum, para valores 0 a 9, pode ser vista na Quadro 1.

Quadro 1 – Decodificador BCD para *display* de sete segmentos

Entrada	Saída (segmentos) g f e d c b a	Dígito decimal
0000	1000000	0
0001	1111001	1
0010	0100100	2
0011	0110000	3
0100	0011001	4
0101	0010010	5
0110	0000010	6
0111	1111000	7
1000	0000000	8
1001	0010000	9

Fonte: Produção do próprio autor.

O segmento de saída será energizado quando a saída estiver em nível baixo, pois esta configuração é anodo comum.

Como o *dis0e1* trabalha com 4 *displays* de sete segmentos, operando de forma multiplexada, onde ele alterna entre o tempo que cada *display* fica ligado um de cada vez. Ou seja, apresenta um valor em um display enquanto os demais estão desligados e, quando desliga este *display*, muda o valor e apresenta este no próximo *display*. Ao fim, retorna ao início. Como isto é realizado em alta frequência para o olho humano (500 Hz), tem-se a impressão que todos estão ligados ao mesmo tempo.

Os blocos **selecaoFrequencia** e **selecaoAmplitude** identificam, através do valor fornecido pelos contadores, os valores desejados de frequência e amplitude, fornecendo assim, tanto os valores que são destinados ao *display* quanto os valores destinados ao programa principal.

A fim de gerar o SPWM é preciso comparar o valor de pico da moduladora com o valor de pico da portadora, para evitar sobremodulação. A amplitude da moduladora foi de 53800 e a portadora de 65891, o que corresponde a um índice de modulação igual a 0,816. Os valores são elevados pois esta FPGA suporta apenas valores inteiros.

Para a programação, tanto os valores da portadora quanto da moduladora foram tabelados e armazenados em um vetor com 200 amostras. Como a Nexys2 trabalha com o *clock* interno de 50 MHz, o vetor possui 200 amostras e a portadora, cuja frequência é de 19,23 kHz. Estas características fazem com que, a cada 13 ciclos de *clock*, a portadora mude para o próximo valor tabelado. O valor da portadora está próximo aos 20 kHz devido ao fato desta apenas trabalhar com números inteiros.

O *clock* de trabalho da moduladora é definido pelo usuário, bem como sua amplitude. A amplitude da moduladora é encontrada no vetor ‘senoide’, cujo valor é multiplicado pelo valor informado pelo usuário, através da quantidade de vezes que apertou os botões, o que define o valor a ser comparado. A saída do SPWM é trifásica, portanto, deve-se ter três senoides, as quais são defasadas em 120° graus, o que corresponde a uma defasagem 66 casas do vetor ‘senoide’.

4 PROJETO DO INVERSOR

Neste capítulo encontram-se os cálculos realizados para o circuito de potência, cujos dados estão disponíveis no Quadro 2.

Quadro 2 – Especificação para o projeto do inversor trifásico de tensão

Grandezas	Valor nominal
Tensão de alimentação	300V
Tensão eficaz de fase do inversor	100V
Potência de saída	750W
Frequência de saída	10-100 Hz
Frequência de comutação	19,3 kHz
Ondulação de corrente do indutor	30%
Ondulação de tensão capacitor	1%

Fonte: Produção do próprio autor.

4.1 Determinação de valores do circuito potência

Para determinar a carga, utiliza-se da equação (86).

$$R = \frac{3 * V_{OEFZ}^2}{P_e} = \frac{3 * 100^2}{750} = 40\Omega \quad (86)$$

Onde:

V_{OEFZ} : tensão eficaz (V);

P_e : potência elétrica (W).

Uma vez definida a carga, calcula-se a corrente de saída do mesmo, conforme mostra na equação (87).

$$I_{oefz} = \frac{V_{OEFZ}}{R} = \frac{100}{40} = 2,5 \text{ A} \quad (87)$$

Dispondo dos critérios de projeto, são calculadas as ondulações de corrente sobre o indutor e de tensão sobre o capacitor de filtragem por meio das equações (88) e (89).

$$\Delta I_L = 0,3I_{OPK} = 0,3 * 2,5 * \sqrt{2} = 1,06 \text{ A} \quad (88)$$

$$\Delta V_C = 0,01V_{OPK} = 0,01 * 100 * \sqrt{2} = 1,41 \text{ V} \quad (89)$$

Note-se que, tanto a corrente como a tensão estão sendo calculadas pelos valores de pico, portanto, sendo multiplicadas por raiz quadrada de dois.

Tendo encontrado o valor da ondulação máxima da corrente sobre o indutor é possível determinar o valor de sua indutância, utilizando a equação (90).

$$L = \frac{V_I}{8f_s \Delta I_L} = \frac{300}{8 * 20000 * 1,06} = 1,76 \text{ mH} \quad (90)$$

O mesmo é feito para o capacitor, valendo-se da equação (91).

$$C = \frac{1}{128 f_s^2 L \Delta V_c} = \frac{300}{128 * (20e^3)^2 * 1,76e^{-3} * 1,41} = 2,34 \text{ } \mu\text{F} \quad (91)$$

A frequência de ressonância é dada pela equação (92).

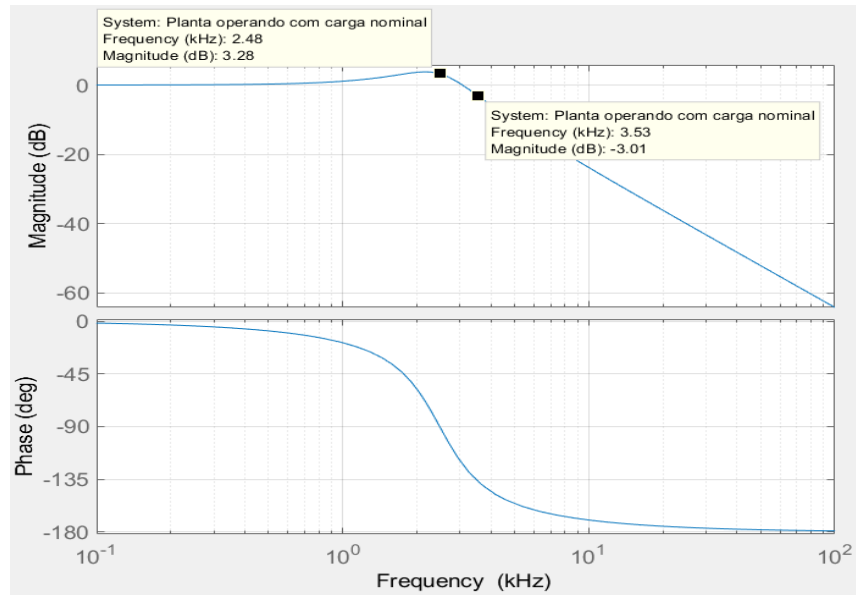
$$F_0 = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{1,76\text{m} * 2,34\mu}} = 2,48 \text{ kHz} \quad (92)$$

4.2 Condição crítica de funcionamento

Definidos os valores de carga e do filtro, em seguida, deve-se analisar qual será o pior valor para a carga. Para isso, utiliza-se o modelo matemático dado pela equação (52).

A seguir, tem-se o diagrama de Bode com a carga nominal, mostrado na Figura 20.

Figura 20 – Diagrama de Bode em malha aberta com carga nominal

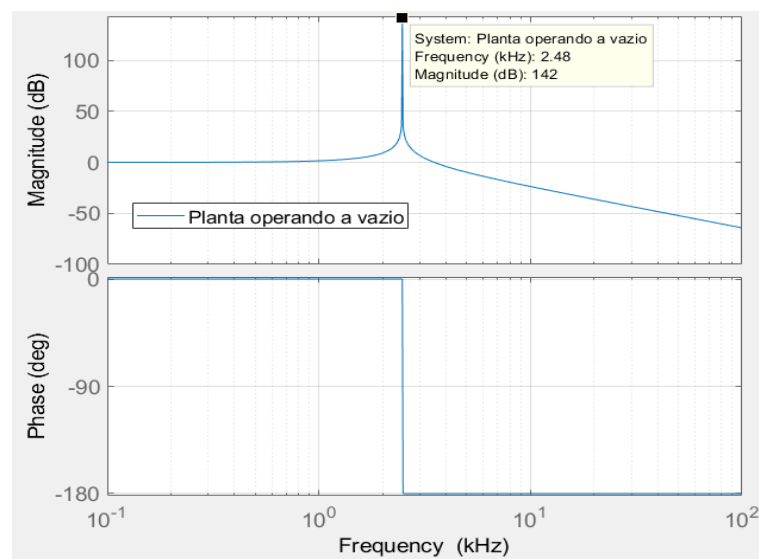


Fonte: Produção do próprio autor.

O diagrama de Bode mostra que a frequência de corte está próxima a 3,53 kHz. Assim sendo, este inversor poderá trabalhar de forma adequada, haja vista que a tensão de saída do mesmo tem frequência máxima 100 Hz.

Quando o valor da resistência for idealmente infinito, este estará a vazio, o que promove a mudança no seu comportamento conforme pode ser visto no diagrama de Bode na Figura 21.

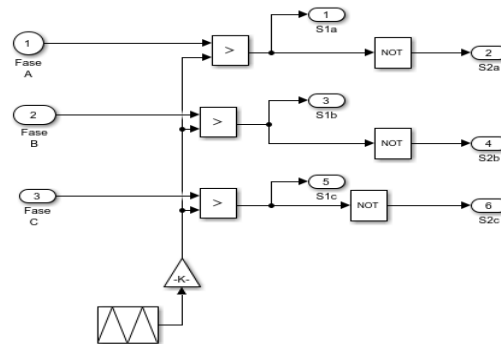
Figura 21 – Diagrama de Bode malha aberta operando a vazio



Fonte: Produção do próprio autor.

Em relação ao bloco SPWM, que é responsável pela comutação dos IGBTs, cujo interior é mostrado na Figura 23, é possível observar a comparação realizada entre a portadora e a moduladora, o qual é responsável por gerar tais comandos.

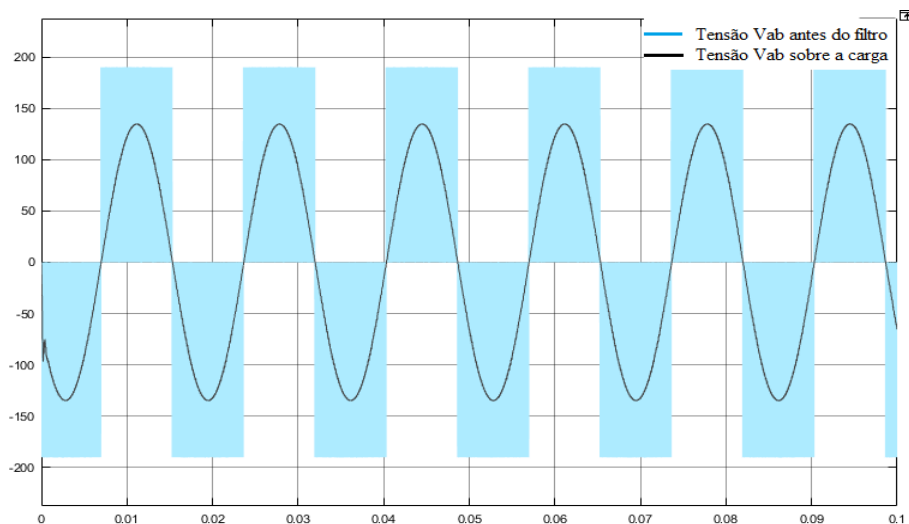
Figura 23 – Bloco SPWM



Fonte: Produção do próprio autor.

A tensão de linha na saída dos IGBTs está representada em azul, como mostrado na Figura 24. Após a passagem pelo filtro LC, com a carga nominal identificada pela cor preta, obtém-se a tensão de linha aplicada sobre a carga.

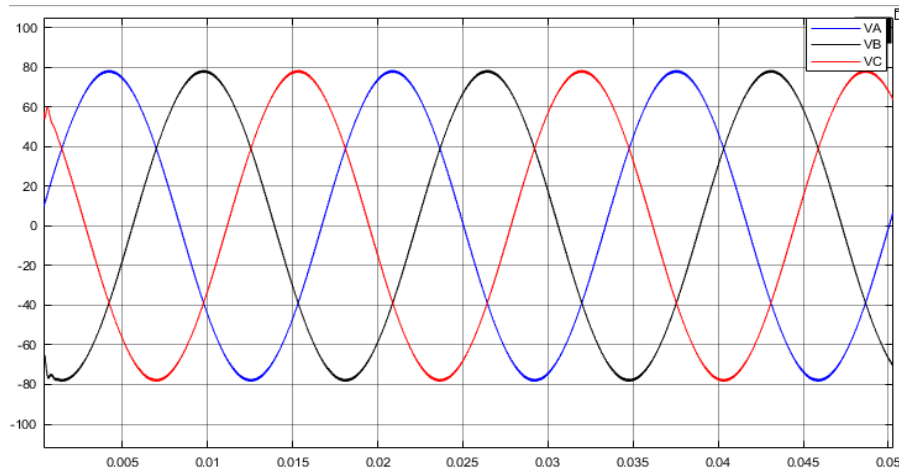
Figura 24 – Simulação do inversor operando carga nominal



Fonte: Produção do próprio autor.

Conectando uma carga resistiva equilibrada de $40\ \Omega$ por fase à saída no inversor, obteve-se a Figura 25, onde é apresentada as três fases defasadas em 120 graus, além da tensão de fase.

Figura 25 – Tensão de saída com carga nominal

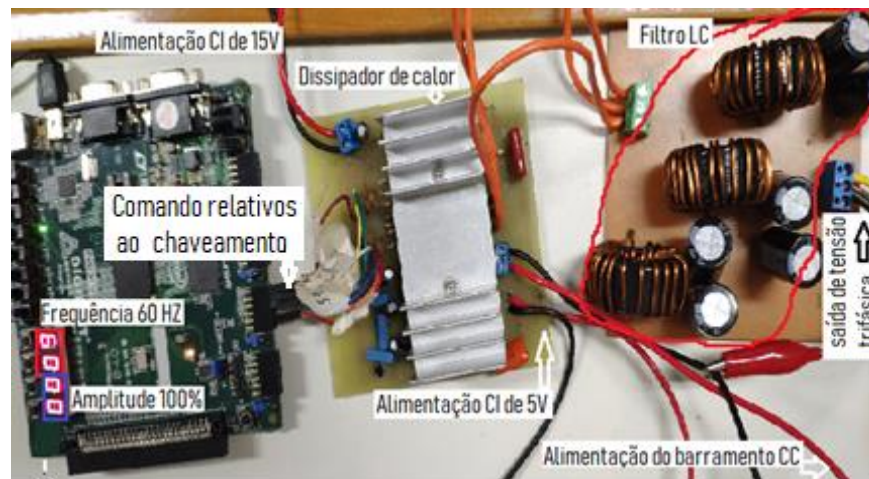


Fonte: Produção do próprio autor.

4.3.1 Resultados experimentais

O protótipo construído foi submetido a alguns testes, a fim de validar seus resultados quando comparados aos simulados. O protótipo montado pode ser visualizado na Figura 26.

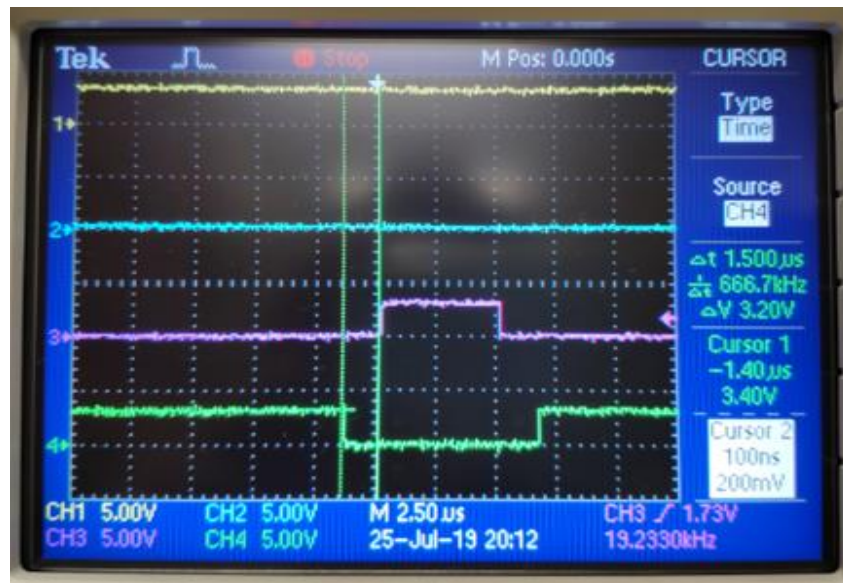
Figura 26 – Foto do protótipo montado



Fonte: Produção do próprio autor.

Para o início da montagem foi necessário verificar se os comandos estavam em acordo com o *datasheet*. Este informa que o tempo mínimo de atraso entre as chaves de um mesmo braço deve ser de $1,5\mu s$, sendo encontrado o valor de atraso de $1,5\mu s$, com as saídas das chaves s2 e s5, conforme mostrado na Figura 27, assim como a frequência de comutação de 19,2330 kHz.

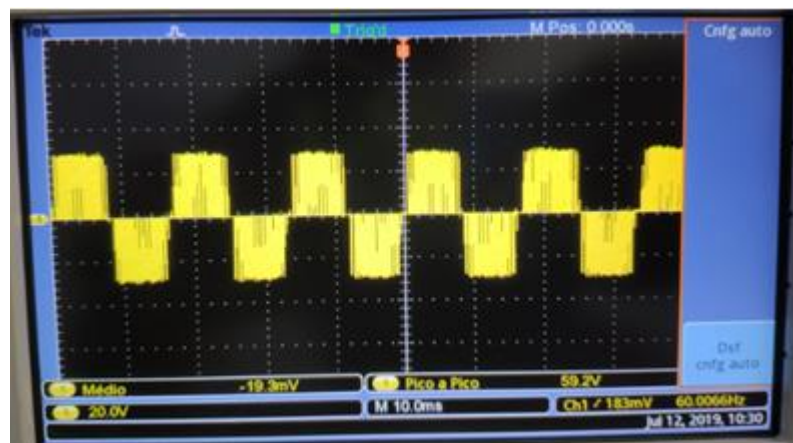
Figura 27 – Tempo morto no pulso SPWM



Fonte: Produção do próprio autor.

Conectando os pulsos gerados pela FPGA à placa do inversor e adicionando a carga resistiva em sua saída, foi a averiguada a forma de onda sem a presença do filtro, como mostrado na Figura 28.

Figura 28 – Saída do inversor sem filtro



Fonte: Produção do próprio autor.

Em seguida, os testes foram realizados com a tensão de barramento em 190V CC, como indicado na Figura 29.

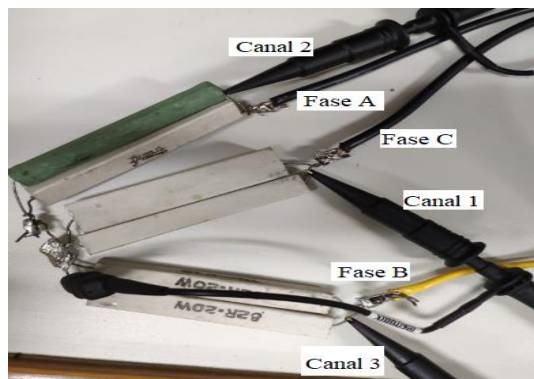
Figura 29 – Tensão no barramento CC



Fonte: Produção do próprio autor.

A primeira verificação foi a defasagem entre as tensões. A Figura 30 mostra como foi realizada esta medida. Já a Figura 32 apresenta a defasagem de 120 graus encontrada.

Figura 30 – Carga do inversor

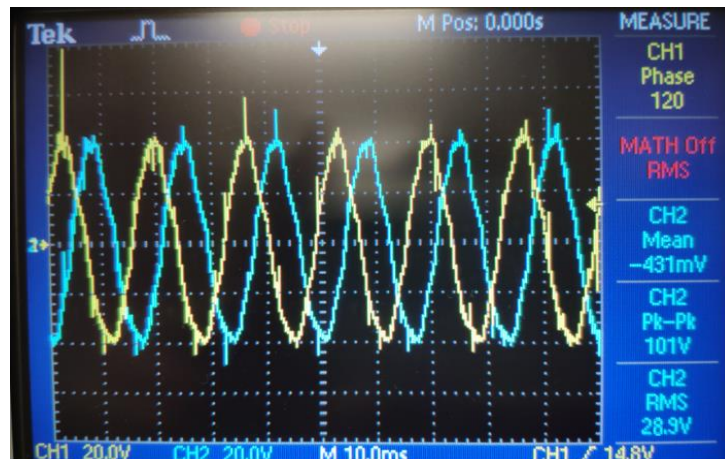


Fonte: Produção do próprio autor.

A Figura 31 mostra a defasagem entre as fases A, em azul (canal 2 do osciloscópio), e a fase C, em amarelo, adiantada em 120 graus, configurando assim, a sequência de fase negativa. Sabe-se que o sistema trifásico equilibrado a separação entre as fases é equidistante, ou seja, de 120 graus entre fases. Logo, a defasagem entre as fases evidência que este sistema é trifásico.

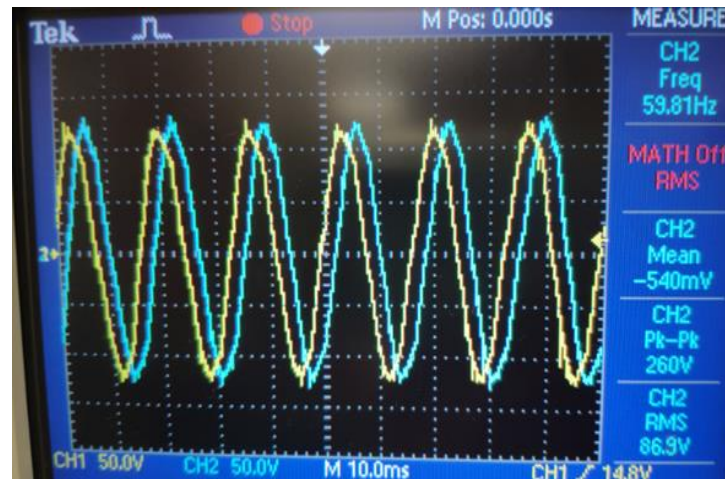
Posteriormente, foi medida a tensão fase-fase como mostrado na Figura 32.

Figura 31 – Defasagem da tensão entre fases



Fonte: Produção do próprio autor.

Figura 32 – Tensão de linha



Fonte: Produção do próprio autor.

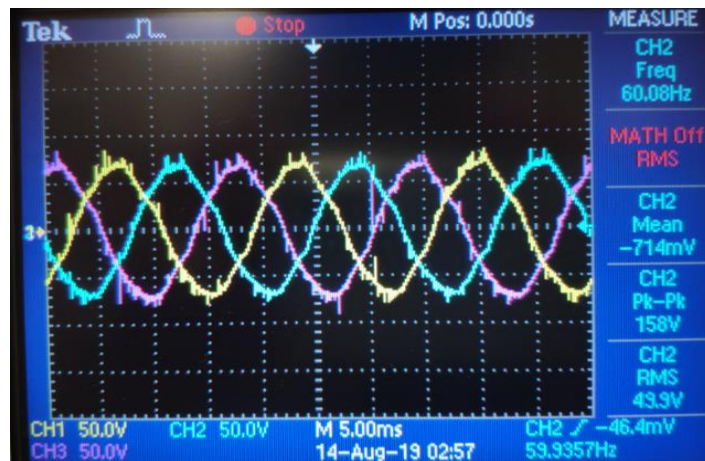
A tensão de linha encontrada é de 86,9 V. Como o sistema é trifásico, ele deve satisfazer à relação entre tensão de linha e tensão de fase, dada pela equação (94).

$$V_{\text{fase}} = \frac{V_{\text{linha}}}{\sqrt{3}} = \frac{86,9}{\sqrt{3}} = 50,1 \text{ V} \quad (94)$$

Calcula-se a tensão de fase dividindo-se a tensão de linha por raiz quadrada de 3, o que resulta no valor de 50,1 V. Desse modo, ao medir a tensão por fase, deve-se encontrar o mesmo valor. A Figura 33 traz as tensões de fase. A tensão de fase encontrada é de 49,9 V, muito semelhante ao calculado. Esta pequena diferença pode ter sido causada pelo número reduzido de amostras presente na Figura 33, pois o valor apresentado pelo osciloscópio é calculado a partir da imagem

exibida na tela. Assim, quanto maior o número de ciclos completos da tensão presentes na tela, melhor será a precisão dos dados.

Figura 33 – Tensão de fase



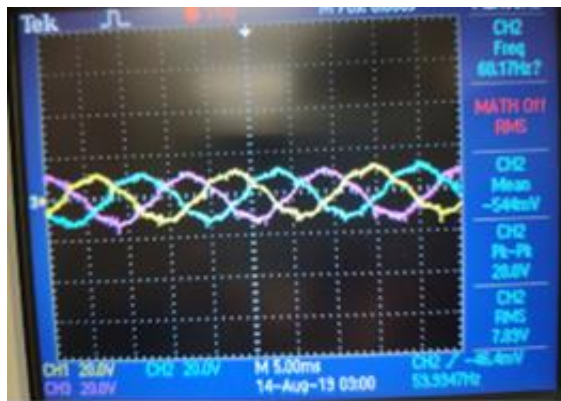
Fonte: Produção do próprio autor.

4.3.2 Variando a tensão de saída

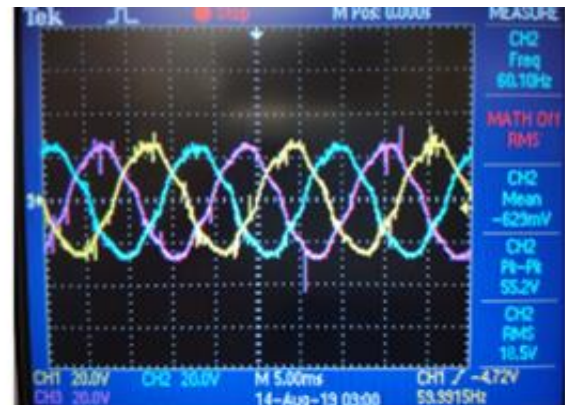
O primeiro método, foi utilizado para verificar a tensão, mantendo a frequência em 60 Hz e variar a tensão de saída.

Para isso, foram pressionados os botões da placa Nexys2 a fim de mudar a amplitude da senoide, internamente programada, resultando na variação da amplitude de saída. Na Figura 34 é mostrada a variação, onde as tensões testadas foram 20%, 40%, 60%, 80% e 100% da amplitude. A Tabela 1 traz comparação entre os valores teóricos e experimentais.

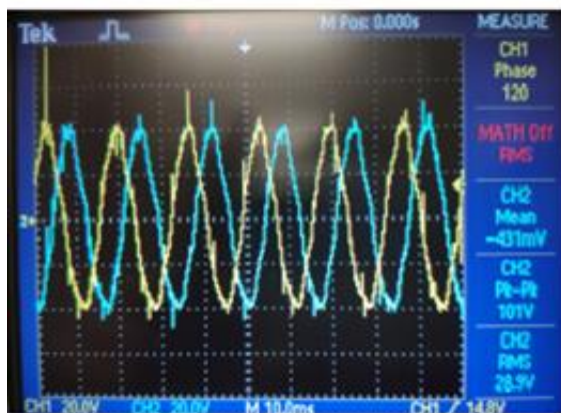
Figura 34 – Variação em amplitude da tensão, sendo (a) 20%, (b) 40 %, (c) 60 %, (d) 90 % e (e) 100 %



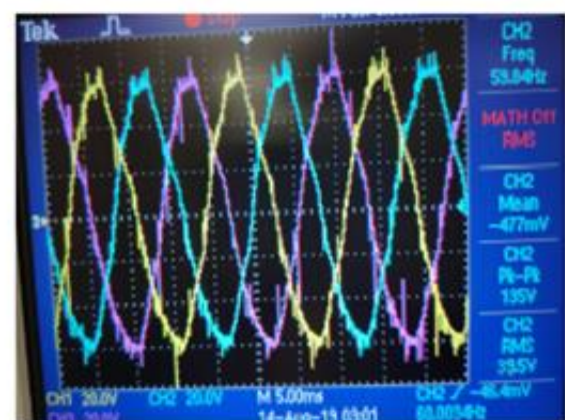
(a)



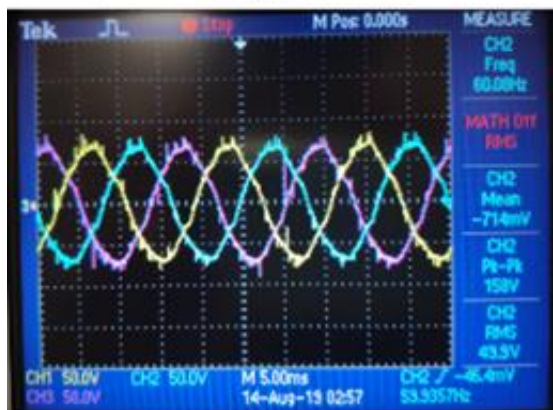
(b)



(c)



(d)



(e)

Fonte: Produção do próprio autor.

Tabela 1 – Variação em amplitude

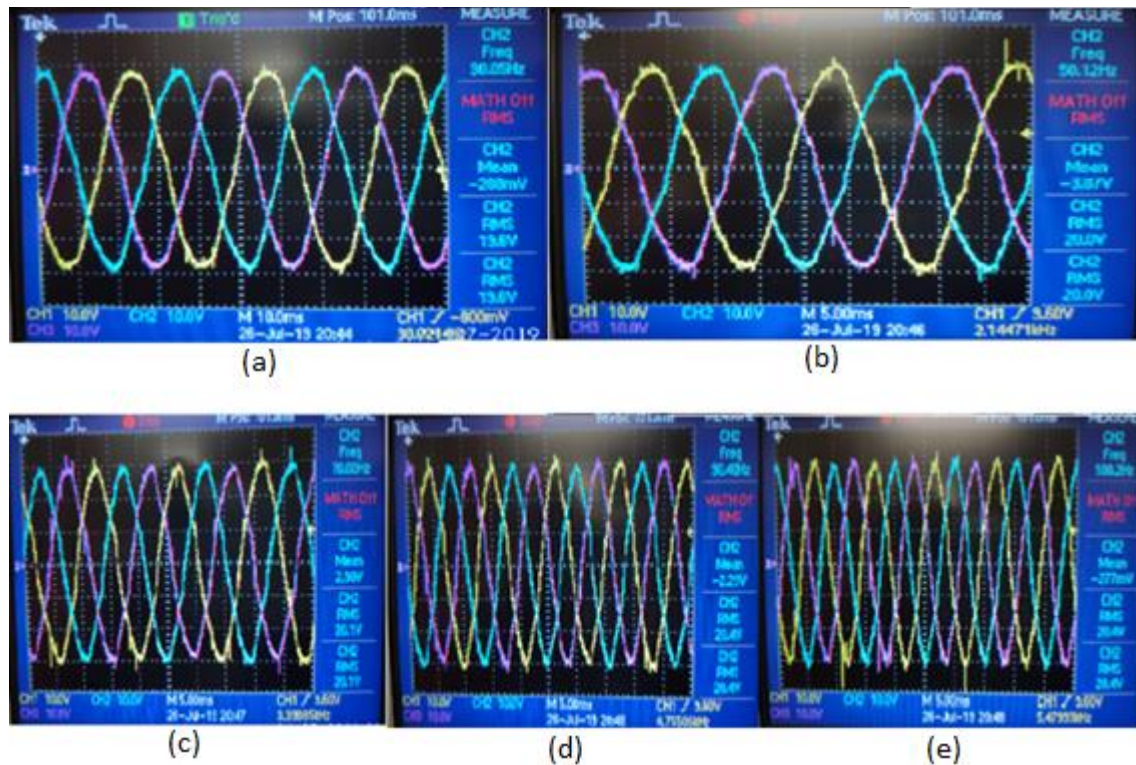
Percentual amplitude 60 (Hz)	Tensão de fase (teórico RMS) (V)	Tensão de fase (experimental RMS) (V)	Erro (V)
20%	10,98	7,89	3,09
40%	21,96	18,5	3,5
60%	32,94	28,9	4,0
80%	43,92	39,5	4,4
100%	54,90	49,9	5,0

Fonte: Produção do próprio autor.

A diferença de tensão entre o valor teórico e o experimental é justificada pela queda de tensão que ocorre entre emissor e coletor quando o IGBT encontra-se operando em saturação, com valor típico 1,7 V em cada IGBT, podendo chegar a 2,2 V, como informa o *datasheet*. À medida que a tensão aumenta, os valores teórico e experimental se distanciam, pois o filtro LC causa uma queda de tensão devido à resistência interna do indutor.

O segundo método é aplicado com finalidade de verificar a variação em frequência, mantendo a tensão em 45% da amplitude e variando em frequência. Para apurar o seu funcionamento, foram coletados os dados das frequências mostrados na Figura 35, os quais são comparados com os valores na Tabela 3.

Figura 35 – Variação em frequência, sendo (a) 30 Hz, (b) 50 Hz, (c) 70 Hz, (d) 90 Hz e (e) 100 Hz



Fonte: Produção do próprio autor.

Tabela 2 – Comparação entre frequência valor teórico *versus* experimental

Frequência (Hz)	Frequência (Hz) (experimental)	Erro (%)
30,00	30,05	0,167
50,00	50,12	0,24
70,00	70,03	0,043
90,00	90,40	0,44
100,00	100,2	0,2

Fonte: Produção do próprio autor.

Em acordo com os dados da Tabela 2, o erro percentual é menor do que 0,5%, mostrando assim uma boa exatidão.

Nota-se, na Figura 35, que os valores da tensão apresentam algumas variações na amplitude, esta variação, ocorre devido à resposta do filtro, que mudar para as diferentes frequências.

5 CONCLUSÕES FINAIS E SUGESTÕES DE TRABALHOS FUTUROS

5.1 Conclusões

Inicialmente, apresenta-se algumas aplicações dos inversores de frequência. Em seguida, foi realizada a modelagem matemática aplicada a seus componentes, assim como, as formas de ondas sobre os mesmos. Por meio da modelagem matemática foi dimensionados os componentes do filtro LC, como também, o dissipador calor do CI.

Visando gerar os comandos SPWM, foi apresentada uma breve descrição do *kit* Nexys2, o qual é composto por uma FPGA. A grande vantagem em se utilizar uma FPGA quando comparada ao microcontrolador, é que essa permitiu a elaboração de cálculos em paralelo, o que torna sua resposta mais rápida. Uma das vantagens em usar dispositivo lógico programável é sua flexibilidade. Por exemplo, a modificação da frequência de comutação, pode ser reprogramada com a finalidade de atender às novas especificações.

Utilizou-se de ferramenta com baixo nível de abstração (estrutural), como é caso da linguagem VHDL, para a programação da FPGA a fim de gerar os comandos de comutação.

Em seguida, foi montado o protótipo e realizados os ensaios para validar o modelo. Onde os resultados obtidos mostram que o hardware desenvolvido pode controlar a tensão trifásica de saída, desejada pelo usuário, como também a frequência aplicada a carga. A variação em amplitude apresentou valores diferentes, quando contraposto ao simulado, pois os componentes empregados na simulação são ideais. Enquanto a varredura em frequência apresentou um erro menor comparado a variação em amplitude. Haja visto que o erro é causado quando se define a quantidade de *clocks*, o qual permanece em determinado valor tabelado. Caso ultrapasse esse número, deve-se utilizar após ultrapassar o próximo valor da tabela. Como nem todos os números calculados para o salto são inteiros, estes valores são aproximados, o que explica a diferença entre o simulado e o dado coletado.

5.2 Sugestões de trabalhos futuros

Para trabalhos futuros, sugere-se a implementação de partida em rampa e controle V/F utilizando FPGA, bem como, a implementação de Controle Vetorial de Motor de Indução utilizando FPGA.

REFERÊNCIAS BIBLIOGRÁFICAS

- CODÁ, R. **Apostila de Introdução a VHDL**. São Carlos, [2014]. Departamento de Engenharia Elétrica e de Computação. Disponível em: https://edisciplinas.usp.br/pluginfile.php/530833/mod_resource/content/1/Apostila%20de%20Introdu%C3%A7%C3%A3o%20a%20VHDL_2014.pdf. Acesso em: 02 jul. 2019.
- DIGILENT. **Digilent Nexys2 Board Reference Manual**. [S. l.], 13 jan. 2013. Digilent. Disponível em: https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-2/nexys2_rm.pdf. Acesso em: 24 abr. 2019.
- ESTEVES, R. C. **MODERNO SISTEMA DE PROPULSÃO**. 2013. Trabalho de conclusão de curso (Bacharel em Ciências Náuticas) - Centro de Instrução Almirante Graça Aranha, Marinha do Brasil, Rio de Janeiro, 2013. Disponível em: <http://www.redebim.dphdm.mar.mil.br/vinculos/000005/00000555.pdf>. Acesso em: 9 ago. 2019.
- FAIRCHILD SEMICONDUCTO. **AN-9070 Smart Power Module Motion SPM® Products in SPM45H Packages**. [S. l.], 28 ago. 2008. Fairchild Semiconducto2. Disponível em: <http://s1.dtsheet.com/store/data/001360856.pdf?key=8a0afa20c9afd9e1b8abadd69a8db94&r=1>. Acesso em: 12 abr. 2019.
- FAIRCHILD SEMICONDUCTO. **FNA41060/FNA41060B2**. [S. l.], 6 jan. 2011. Fairchild Semiconducto, Disponível em: <https://www.mouser.com/datasheet/2/149/FNA41060-1008842.pdf>. Acesso em: 24 jul. 2019.
- GIACOMINI, R. **Apostila básica de VHDL**. Campinas, [2000], [S. l.]. Unicamp. Disponível em: http://www.dsif.fee.unicamp.br/~juvenilj/apostilas_vhdl/Apostila_VHDL.pdf. Acesso em: 15 dez. 2018.
- GUEDES, S D. **Projeto de inversor trifásico aplicado ao acionamento de motor de indução**. 2015. Trabalho de conclusão de curso (Bacharel em Engenharia Elétrica) - Departamento de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2015. Disponível em: <http://webcache.googleusercontent.com/search?q=cache:http://monografias.poli.ufrj.br/monografias/monopoli10014074.pdf>. Acesso em: 18 maio .2019.
- GURGEL, M. **Aplicações de inversores eletrônicos na indústria nacional: a contribuição brasileira**. 2009. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Escola Politécnica da Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2009. Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli10000921.pdf>. Acesso em: 05 maio. 2018.
- HAUSMANN, R.; LARICO, H.S.; LAZZARIN, T.B.; PIAZZA, G.L. **Projetos de inversores**. Dissertação (Mestrado em Engenharia Elétrica) – Departamento de Engenharia Elétrica, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis, 2008. Disponível em: https://docgo.net/document/doDownload/link_rand/J5voLPUEPp2axQyB00WemPRcMUo6PELoA6DbMnFrDxr0eQr4Ge8TTPVvA9sfE3u40Y6Cb. Acesso em: 10 jul. 2019.

KINGSTON TECHNOLOGY. **Guia de Memória flash**. Fountain Valley, 10 mai.2004. Kingston Technology Corporation,. Disponível em: https://media.kingston.com/pdfs/Flash_Memory_Guide_br.pdf. Acesso em: 09 ago. 2019.

KNAESEL, C. **Transformador de Estado Sólido Monofásico Utilizando o Conversor Dual Active Bridge (DAB): Estudo do Bloco DAB+Inversor**. 2016. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Instituto de Eletrônica de Potência, Universidade Federal de Santa Catarina, Florianópolis. 2016.1 Pen Drive.

MARCON, M.; MORENO, I. **Introdução à linguagem vhdl**. São Carlos, [2015]. Disponível em: https://edisciplinas.usp.br/pluginfile.php/3020729/mod_resource/content/0/Aula%20VHDL%20Alternativa.pdf. Acesso em: 06 jun.2019.

MARTINS, D. C.; BARBI, I. **Introdução ao estudo dos conversores cc-ca**. Florianópolis: Edição dos Autores, 2005.

MOTA, A. **Leitura de Botões e o Bounce**. [S. l.], 13 mai. 2017. Vida de silício. Disponível em: <https://portal.vidadesilicio.com.br/leitura-de-botoes-e-o-bounce/>. Acesso em: 04 abr. 2019.

MONTEIRO, J. R. B. A. **Apostila de Eletrônica de Potência**. São Paulo, 2015. Universidade de São Paulo. Disponível em: https://edisciplinas.usp.br/pluginfile.php/302059/mod_resource/content/0/elepota.pdf. Acesso em: 06 maio 2018.

PACHECO, J. O. **Desenvolvimento de um sistema didático para ensino de conversores cc-ca com monitoramento por microcontroladores**. 2012. Monografia (Especialização em Desenvolvimento de Produtos Eletrônicos) – Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Florianópolis, 2012. Disponível em: <http://www.uniedu.sed.sc.gov.br/wp-content/uploads/2014/01/Juliano-de-Oliveira-Pacheco.pdf>. Acesso em: 5 set. 2019.

RANIEL, T. **Desenvolvimento e Implementação de um Sistema de Controle de Posição e Velocidade de uma Esteira Transportadora usando Inversor de Frequência e Microcontrolador**. 2011. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Estadual Paulista, Ilha Solteira, 2011. Disponível em: https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/pos-graduacao/291-dissertacao_thiagoraniel.pdf. Acesso em: 1 set. 2019.

RASHID, M. H. **Eletrônica de Potência: Circuitos, dispositivos e aplicações**. Tradução Carlos Alberto Favato. São Paulo: Makron Books, 1999. E-book. Disponível em: https://www.academia.edu/38932140/Eletr%C3%B4nica_de_pot%C3%Aancia_-_Rashid. Acesso em: 06 ago. 2019.

WEG. **WEG equipa primeiro ônibus elétrico do país movido 100% abateia**. Jaraguá do Sul, 25 set. 2013. Weg notícias. Disponível em:

<http://www.weg.net/institutional/BR/pt/news/produtos-e-solucoes/weg-equipa-primeiro-onibus-eletrico-do-pais-movido-100-a-bateria>. Acesso em: 06 maio 2018.

WEG. **CFW-09 Inversores de Frequência**. Jaraguá do Sul, 2008. WEG Automação, Disponível em: http://www.multiluzbh.com.br/weg/weg_857brand.pdf. Acesso em: 2 maio 2019.

XILINX. **LSIM DESIGN SUITE OVERVIEW**. [S. l.], [s.l.]. Xilinx. Disponível em: <https://www.xilinx.com/products/design-tools/ise-design-suite.html>. Acesso em: 24 abr. 2019.

APENDICE A – CÓDIGO UTILIZADO NA FPGA

```

1  ---Projeto de graduação
2  ---Renã Marques Pastore
3  ---Inversor Trifásico
4  library IEEE;
5  use IEEE.STD_LOGIC_1164.ALL;
6  USE ieee.std_logic_unsigned.ALL;
7  entity inversor is --declaração da entidade
8  port (
9      clk : in std_logic; --declaração da entrada clock
10     button1 : in std_logic:= '0';
11     button2 : in std_logic:= '0';
12     button3 : in std_logic:= '0';
13     button4 : in std_logic:= '0';
14     sw:in STD_LOGIC_VECTOR(1 downto 0):= (others => '0');
15     reset : in std_logic;
16     S1,S2,S3,S4,S5,S6: out std_logic; --declaração das variáveis de saída
17     cur_display: OUT std_logic_vector(3 DOWNT0 0);---identifica anodo do display
18     display: out STD_LOGIC_VECTOR(7 downto 0)---seguimento os quais devem ser ligados
19 );
20 end entity inversor; --fim da entidade
21 architecture estrutura of inversor is --início da arquitetura
22 -----
23     component debounce
24     Port ( clk : IN STD_LOGIC;
25           button : IN STD_LOGIC; --input signal to be debounced
26           result : OUT STD_LOGIC); --debounced signal
27     |
28     end component;
29
30
31     signal result0 : STD_LOGIC;
32     signal result1 : STD_LOGIC;
33     signal result2 : STD_LOGIC;
34     signal result3 : STD_LOGIC;
35
36 -----
37 component digital_clock_top
38 Port (
39
40
41     clk: in STD_LOGIC;
42     display: out STD_LOGIC_VECTOR(7 downto 0); ---saída para o display
43     cur_display: out STD_LOGIC_VECTOR(3 downto 0);---identifica anodo do display
44     dis0:in integer;
45     dis1:in integer;
46     dis2:in integer;
47     dis3:in integer
48 );
49
50     end component;
51 -----
52 component contador1 ---Primeiro contador
53
54 Port ( clk: in std_logic;
55       reset : in std_logic;
56       b0:in STD_LOGIC;
57       b1 : in STD_LOGIC;

```

```

57     b1 : in STD_LOGIC;
58     saida : out integer
59 );
60 end component;
61
62     signal ContSen : integer;
63     signal AmpSen: integer;
64
65 -----
66
67 component contador2 ---segundo contador
68
69 Port ( clk: in std_logic;
70       reset : in std_logic;
71       b2:in STD_LOGIC;
72       b3 : in STD_LOGIC;
73       saida : out integer
74
75 );
76 end component;
77 signal ContFr: integer;
78 signal Fr : integer ;
79
80 -----
81
82 component frequencia
83
84 Port ( clk: in std_logic;
85
86
87 ContFr: in integer;
88 Fr : out integer;
89 dis2: out integer;
90 dis3: out integer
91 );
92 end component;
93
94 -----
95 component amplitude
96
97 Port ( clk: in std_logic;
98       ContSen: in integer;
99       AmpSen : out integer;
100       dis0: out integer;
101       dis1: out integer
102
103 );
104 end component;
105
106 -----
107
108
109 signal c1 : integer :=1; --valor inicial p/ c1 que define n do vetor
110 signal c2 : integer :=67; --defasagem da senoide fase B
111 signal c3 : integer :=133; --defasagem da senoide fase C
112 signal a : integer :=1; -- declaração variavel a de contagem

```

```

112 signal a : integer :=1; -- declaração variavel a de contagem
113 signal saidal : integer:=0; -- declaração saída senoide Fase A
114 signal saida2 : integer:=0; -- declaração saída senoide Fase B
115 signal saida3 : integer:=0; -- declaração saída senoide Fase C
116 signal TempoMorto1a : STD_LOGIC_VECTOR(7 downto 0) :=("00000000");
117 signal TempoMorto1b : STD_LOGIC_VECTOR(7 downto 0) :=("00000000");
118 signal TempoMorto2a : STD_LOGIC_VECTOR(7 downto 0) :=("00000000");
119 signal TempoMorto2b : STD_LOGIC_VECTOR(7 downto 0) :=("00000000");
120 signal TempoMorto3a : STD_LOGIC_VECTOR(7 downto 0) :=("00000000");
121 signal TempoMorto3b : STD_LOGIC_VECTOR(7 downto 0) :=("00000000");
122 signal b1,b2,b3,b4,b5,b6 :STD_LOGIC:='0';--recebem os memos bits das saída S.Usado para o tempo morto
123
124 signal dis0:integer;
125 signal dis1:integer;
126 signal dis2:integer;
127 signal dis3:integer;
128
129
130 type senoid is array (integer range <>) of integer; --declaração 200 valores amostrados para a função de referencia
131 signal senoide:senoid (1 to 200):=(17,34,51,67,84,101,117,134,150,166,182,198,214,229,244,259,274,288,302,316,330,
132 343,356,368,380,392,403,414,425,435,445,454,463,471,479,487,494,500,506,512,517,521,525,528,531,534,536,537,538,538,
133 538,537,536,534,531,529,525,521,517,512,506,500,494,487,480,472,463,455,445,436,425,415,404,393,381,369,356,343,330,
134 317,303,289,274,260,245,230,214,199,183,167,151,135,118,102,85,68,51,35,18,1,-16,-33,-50,-67,-83,-100,-116,-133,-149,
135 -165,-181,-197,-213,-228,-243,-258,-273,-287,-302,-315,-329,-342,-355,-368,-380,-391,-403,-414,-424,-435,-444,-454,-462,
136 -471,-479,-486,-493,-500,-506,-511,-516,-521,-525,-528,-531,-534,-535,-537,-538,-538,-537,-536,-534,-532,-529,-525,
137 -521,-517,-512,-507,-501,-494,-487,-480,-472,-464,-455,-446,-436,-426,-415,-405,-393,-381,-369,-357,-344,-331,-317,-304,
138 -290,-275,-261,-246,-231,-215,-200,-184,-168,-152,-135,-119,-102,-86,-69,-52,-35,-19,-2);
139
140 constant max_refresh_count: INTEGER := 100000;

```

```

140 constant max_refresh_count: INTEGER := 100000;
141 signal refresh_count: INTEGER range 0 to max_refresh_count;
142 signal refresh_state: STD_LOGIC_VECTOR(1 downto 0) := (others => '0');
143 signal display_sel: STD_LOGIC_VECTOR(3 downto 0) := (others => '0');
144
145
146
147 signal saidat : integer:=1; --declaração variavel da portadora 1
148 signal L : STD_LOGIC_VECTOR(3 downto 0) :=("1010"); --declaração variável L de contagem
149 signal m : integer :=1; --declaração variável m de contagem
150 --signal N1: integer :=1; --declaração variável N1
151 signal N2: STD_LOGIC_VECTOR(3 downto 0); --declaração variável N2
152
153
154
155
156
157 type triangular is array (integer range <>) of integer; --declaração 200 valores amostrados para a função da portadora
158
159 signal triangular1:triangular (1 to 200):=(-68310,-66930,-65550,-64170,-62790,-61410,-60030,-58650,-57270,-55890,-54510,
160 -53130,-51750,-50370,-48990,-47610,-46230,-44850,-43470,-42090,-40710,-39330,-37950,-36570,-35190,-33810,-32430,-31050,
161 -29670,-28290,-26910,-25530,-24150,-22770,-21390,-20010,-18630,-17250,-15870,-14490,-13110,-11730,-10350,-8970,-7590,
162 -6210,-4830,-3450,-2070,-690,690,2070,3450,4830,6210,7590,8970,10350,11730,13110,14490,15870,17250,18630,20010,21390,
163 22770,24150,25530,26910,28290,29670,31050,32430,33810,35190,36570,37950,39330,40710,42090,43470,44850,46230,47610,48990,
164 50370,51750,53130,54510,55890,57270,58650,60030,61410,62790,64170,65550,66930,68310,69000,67620,66240,64860,63480,62100,
165 60720,59340,57960,56580,55200,53820,52440,51060,49680,48300,46920,45540,44160,42780,41400,40020,38640,37260,35880,34500,
166 33120,31740,30360,28980,27600,26220,24840,23460,22080,20700,19320,17940,16560,15180,13800,12420,11040,9660,8280,6900,5520,
167 4140,2760,1380,0,-1380,-2760,-4140,-5520,-6900,-8280,-9660,-11040,-12420,-13800,-15180,-16560,-17940,-19320,-20700,-22080,
168 -23460,-24840,-26220,-27600,-28980,-30360,-31740,-33120,-34500,-35880,-37260,-38640,-40020,-41400,-42780,-44160,-45540,

```

```

168 -23460,-24840,-26220,-27600,-28980,-30360,-31740,-33120,-34500,-35880,-37260,-38640,-40020,-41400,-42780,-44160,-45540,
169 -46920,-48300,-49680,-51060,-52440,-53820,-55200,-56580,-57960,-59340,-60720,-62100,-63480,-64860,-66240,-67620);
170
171
172 signal indice:STD_LOGIC_VECTOR(1 DOWNTO 0);
173
174
175 begin
176 db0:debounce port map(clk=>clk,button=>button1,result=>result0);
177 db1:debounce port map(clk=>clk,button=>button2,result=>result1);
178 db2:debounce port map(clk=>clk,button=>button3,result=>result2);
179 db3:debounce port map(clk=>clk,button=>button4,result=>result3);
180
181
182 contador:contador1 port map(clk=>clk,b0=>result0,b1=>result1,reset=>reset,saida=>ContSen);
183 contado2:contador2 port map(clk=>clk,b2=>result2,b3=>result3,reset=>reset,saida=>ContFr);
184
185 disp0el:digital_clock_top port map(clk=>clk,display=>display,cur_display=>cur_display,dis0=>dis0,dis1=>dis1,dis2=>dis2,dis3=>dis3);
186
187 selecaoFrequencia:frequencia port map(clk=>clk,ContFr=>ContFr,Fr=>Fr,dis2=>dis2,dis3=>dis3);
188 selecaoAmplitude:amplitude port map(clk=>clk,ContSen=>ContSen,AmpSen=>AmpSen,dis0=>dis0,dis1=>dis1);
189
190
191
192
193 process(clk) --inicio processamento do clock
194 begin --inicio do programa
195 IF(clk'EVENT and clk = '1') THEN --executa quando há subida do clock
196

```

```

196 -----
197 N2<="1101"; --50000000/(freqtri*200)=12,5 aproximado 13 ; --cálculo de N2 (subidas de clock)
198 -----
199 if a>fr then --- caso a frequência mude garante que valor de "a" seja menor "fr"
200 a<=1;-- Retorna o contador a
201 elsif a=Fr then
202   saida1<= senoide(c1)*(AmpSen);--carrega na variavel saide o elemento c1 do vetor
203   if c1=200 then --quando excutar uma onda retorna ao inicio
204     c1<=1;
205   else
206     c1<=1+c1; --incrementa em 'c1' uma unidade
207   end if;
208   saida2<= senoide(c2)*(AmpSen); --senoide2
209   if c2=200 then --quando excutar uma onda retorna ao inicio
210     c2<=1;
211   else
212     c2<=1+c2; --incrementa em 'c2' uma unidade
213   end if;
214   saida3 <= senoide(c3)*(AmpSen); --senoide3
215   if c3=200 then --retorno para o 0 quando c3=200
216     c3<=1;
217   else
218     c3<=1+c3; --incrementa em 'c3' uma unidade
219   end if;
220   a<=1; -- faz igual a 1
221   else
222     a<=1+a; --incrementa em 'a' uma unidade
223   end if;
224
225 -----
226 if L=N2 then
227   saidat <= triangular1(m);--carrega portadora
228   L<="0001";
229   if m=200 then --limita variável m em 200
230     m<=1; --carrega valor inicial de m
231   else
232     m<=m+1;
233   end if;
234   else
235     L<="0001"+L;--incrementa uma unidade na variavel L
236   end if;
237 if sw(0) ='1' then --chave para habilitar pulsos na saida
238   if saida1 > saidat then --comparaçao moduladora e portadora.
239     S4<='0'; --chave s4 desativa
240     b4<='0';
241     if b4='0' then--conta apartir s4 desativa
242       TempoMortola<=TempoMortola+"0000001";
243       if TempoMortola="1001011" then--TempoMorto=75 clock, tempo de 1,5us
244         S1<='1'; --chave d1 ativa
245         b1<='1';
246         TempoMortola<="00000000";--zera o tempo morto
247       end if;
248     else --Quando houver mudança em s4 zera a contagem
249       TempoMortola<="00000000";--zera o tempo morto
250     end if;
251   else
252     S1<='0'; --chave s1 desativa

```

```

251      S1<='0'; --chave s1 desativa
252      b1<='0';
253      if b1='0' then
254          TempoMorto1b<= TempoMorto1b+"0000001";
255          if TempoMorto1b="1001011" then
256              S4<='1'; --chave s4 ativa
257              b4<='1';
258              TempoMorto1b<="00000000";
259          end if;
260          else TempoMorto1a<="00000000";--zera o tempo morto
261          end if;
262      end if;
263      if saida2 > saidat then --comparaçao moduladora2 e portadora pos.
264          S5<='0'; --chave s2 desativa
265          b5<='0';
266          if b5='0' then--conta apartir s4 desativa
267              TempoMorto2a<=TempoMorto2a+"0000001";
268              if TempoMorto2a="1001011" then--TempoMorto=75 clock
269                  S2<='1'; --chave s5 ativa
270                  b2<='1';
271                  TempoMorto2a<="00000000";--zera o tempo morto
272              end if;
273              else TempoMorto2b<="00000000";--zera o tempo morto
274              end if;
275          else
276              S2<='0'; --chave s2 desativa
277              b2<='0';
278              if b2='0' then
279                  TempoMorto2b<= TempoMorto2b+"0000001";
280                  TempoMorto2b<= TempoMorto2b+"0000001";
281                  if TempoMorto2b="1001011" then
282                      S5<='1'; --chave s5 ativa
283                      b5<='1';
284                      TempoMorto2b<="00000000";
285                  end if;
286                  else TempoMorto2a<="00000000";--zera o tempo morto
287                  end if;
288              end if;
289              if saida3 > saidat then --comparaçao moduladora3 e portadora pos.
290                  S6<='0'; --chave e1 desativa
291                  b6<='0';
292                  if b6='0' then--conta apartir s4 desativa
293                      TempoMorto3a<=TempoMorto3a+"0000001";
294                      if TempoMorto3a="1001011" then--TempoMorto=75 clock
295                          S3<='1'; --chave d1 ativa
296                          b3<='1';
297                          TempoMorto3a<="00000000";--zera o tempo morto
298                      end if;
299                      else TempoMorto3b<="00000000";--zera o tempo morto
300                      end if;
301                  else
302                      S3<='0'; --chave s6 desativa
303                      b3<='0';
304                      if b3='0' then
305                          TempoMorto3b<= TempoMorto3b+"0000001";
306                          if TempoMorto3b="1001011" then
307                              S6<='1'; --chave s6 ativa
308                              b6<='1';
309                              TempoMorto3b<="00000000";
310                          end if;
311                      else TempoMorto3a<="00000000";--zera o tempo morto
312                      end if;
313                  end if;
314              -- end if;
315          end if ;
316      end process;
317  end architecture estrutura; --fim estrutura

```

Componente **debounce**.


```

1  --debounce
2  LIBRARY ieee;
3  USE ieee.std_logic_1164.all;
4  USE ieee.std_logic_unsigned.all;
5  ENTITY debounce IS
6  GENERIC(
7    counter_size : INTEGER := 19); -- tamanho (19 bits dá 10.5ms com clock de 50MHz)
8  PORT(
9    clk: in std_logic; --input clock
10   button : IN STD_LOGIC; --sinal de entrada para debounce
11   result : OUT STD_LOGIC); --resul do signal
12 END debounce;
13 ARCHITECTURE logic OF debounce IS
14   SIGNAL flipflops : STD_LOGIC_VECTOR(1 DOWNTO 0); --inicia flip flops
15   SIGNAL counter_set : STD_LOGIC; -- reset to zero
16   SIGNAL counter_out : STD_LOGIC_VECTOR(counter_size DOWNTO 0) := (OTHERS => '0'); --counter output
17 BEGIN
18   counter_set <= flipflops(0) xor flipflops(1); --determinar quando iniciar / resetar contador
19   process(clk) --inicio processamento do clock
20   begin --inicio do programa
21     if(clk'event and clk='1') then
22       flipflops(0) <= button;
23       flipflops(1) <= flipflops(0);
24       If(counter_set = '1') THEN --redefinir contador porque a entrada está mudando
25         counter_out <= (OTHERS => '0');
26       ELSIF(counter_out(counter_size) = '0') THEN --o tempo de entrada da tabela ainda não foi atingido
27         counter_out <= counter_out + 1;
28       ELSE --tempo de entrada estável é cumprido
29         result <= flipflops(1);
30       END IF;
31     END IF;
32   END PROCESS;
33 END logic;
34

```

Componente Contador1.

```

1  --Contador 1
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  entity contador1 is
5    Port ( clk: in std_logic;
6          reset :in std_logic;
7          b0 : in std_logic;
8          b1 : in std_logic;
9          saida : out integer
10         );
11 end contador1;
12 architecture Behavioral of contador1 is
13   SIGNAL flipflops : STD_LOGIC_VECTOR(1 DOWNTO 0); --input flip flops
14   SIGNAL flipflops2 : STD_LOGIC_VECTOR(1 DOWNTO 0);
15   shared variable contador: integer:=91;--- 100% Amplitude
16   SIGNAL counter_set : STD_LOGIC;
17   SIGNAL counter_set2 : STD_LOGIC;
18 begin
19   counter_set <= flipflops(0) and (flipflops(0) xor flipflops(1)); -- determinar quando iniciar / resetar contador
20   counter_set2 <= flipflops2(0) and (flipflops2(0) xor flipflops2(1));
21   process (clk,reset)
22   begin
23     if reset = '1' then
24       contador :=1;
25     elsif(clk'EVENT and clk = '1') THEN
26       flipflops(0) <= b0;
27       flipflops(1) <= flipflops(0);
28       If(counter_set = '1') THEN -- redefinir contador porque a entrada está mudando
29         contador := contador + 1;
30       end if;
31       flipflops2(0) <= b1;
32       flipflops2(1) <= flipflops2(0);
33       If(counter_set2 = '1') THEN --redefinir contador porque a entrada está mudando
34         contador := contador - 1;
35       end if;
36       if(contador>91) then -- limite superior amplitude
37         contador:=91;
38       elsif(contador<1) then--limite inferior amplitude
39         contador:=1;
40       end if;
41     end if;
42     saida<=contador;
43   end process;
44 end Behavioral;

```

Componente Contador2.

```

1  --Contador 2
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  entity contador2 is
5      Port ( clk: in std_logic;
6            reset :in std_logic;
7            b2 : in std_logic;
8            b3 : in std_logic;
9            saida : out integer
10         );
11  end contador2;
12  architecture Behavioral of contador2 is
13      SIGNAL flipflops : STD_LOGIC_VECTOR(1 DOWNTO 0); --input flip flops
14      SIGNAL flipflops2 : STD_LOGIC_VECTOR(1 DOWNTO 0);
15      shared variable contador: integer:=51;
16      SIGNAL counter_set : STD_LOGIC;
17      SIGNAL counter_set2 : STD_LOGIC;
18  begin
19      counter_set <= flipflops(0) and (flipflops(0) xor flipflops(1));
20      counter_set2 <= flipflops2(0) and (flipflops2(0) xor flipflops2(1));
21      process (clk, reset)
22      begin
23          if reset = '1' then
24              contador :=1;
25          elsif(clk'EVENT and clk = '1') THEN
26              flipflops(0) <= b2;
27              flipflops(1) <= flipflops(0);
28              If(counter_set = '1') THEN --reset counter because input is changing
29                  contador := contador + 1;
30              end if;
31              flipflops2(0) <= b3;
32              flipflops2(1) <= flipflops2(0);
33              If(counter_set2 = '1') THEN --reset counter because input is changing
34                  contador := contador - 1;
35              end if;
36              if(contador>91) then
37                  contador:=91;
38              elsif(contador<1) then
39                  contador:=1;
40              end if;
41          end if;
42          saida<=contador;
43      end process;
44  end Behavioral;

```

Componente **dispoel**.

```

1  --Disp0el
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5  entity digital_clock_top is
6  port (
7  -- Asen: in integer;
8  -- Fr: in integer;
9  clk: in STD_LOGIC;
10 display: out STD_LOGIC_VECTOR(7 downto 0);
11 cur_display: out STD_LOGIC_VECTOR(3 downto 0);
12 dis0:in integer;
13 dis1:in integer;
14 dis2:in integer;
15 dis3:in integer
16 );
17 end digital_clock_top;
18 architecture beh of digital_clock_top is
19 constant max_count: INTEGER := 25000000; -- 50000000/2
20 constant max_refresh_count: INTEGER := 10000; -- 50Mzh/100000=500Hz
21 signal count: INTEGER range 0 to max_count;
22 signal refresh_count: INTEGER range 0 to max_refresh_count;
23 signal refresh_state: STD_LOGIC_VECTOR(1 downto 0) := (others => '0');
24 signal clk_state: STD_LOGIC := '0';
25 signal display_sel: STD_LOGIC_VECTOR(3 downto 0) := (others => '0');
26 function digito(numero:INTEGER) return STD_LOGIC_VECTOR is
27 variable salida: STD_LOGIC_VECTOR(7 downto 0);
28 begin
29 case numero is
30
31 case numero is
32 when 0 => salida := "11000000"; -- 0
33 when 1 => salida := "11111001"; -- 1
34 when 2 => salida := "10100100"; -- 2
35 when 3 => salida := "10110000"; -- 3
36 when 4 => salida := "10011001"; -- 4
37 when 5 => salida := "10010010"; -- 5
38 when 6 => salida := "10000010"; -- 6
39 when 7 => salida := "11111000"; -- 7
40 when 8 => salida := "10000000"; -- 8
41 when 9 => salida := "10010000"; -- 9
42 when others => salida := "11111111";
43 end case;
44 return(salida);
45 end digito;
46 begin
47 cur_display <= display_sel;
48 gen_clock: process(clk, clk_state, count)
49 begin
50 if clk'event and clk='1' then
51 -- contador 1Hz
52
53 if count < max_count then
54 count <= count + 1;
55 else
56 clk_state <= not clk_state;
57 count <= 0;
58 end if;
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

```

```

57
58 -- contador 500Hz (para refresco del display)
59 if refresh_count < max_refresh_count then
60 refresh_count <= refresh_count + 1;
61 else
62 refresh_state <= refresh_state + 1;
63 refresh_count <= 0;
64 end if;
65 end if;
66 end process;
67
68 show_display: process(refresh_state)
69 begin -- selección del display
70 case refresh_state is
71 when "00" => display_sel <= "1110"; -- display 0
72 when "01" => display_sel <= "1101"; -- display 1
73 when "10" => display_sel <= "1011"; -- display 2
74 when "11" => display_sel <= "0111"; -- display 3
75 when others => display_sel <= "1111";
76 end case;
77
78 -- mostrar hora
79 case display_sel is
80 when "1110" => display <= digito(dis0); -- display 0
81 when "1101" => display <= digito(dis1); -- display 1
82 when "1011" => display <= digito(dis2); -- display 2
83 when "0111" => display <= digito(dis3); -- display 3
84 when others => display <= "11111111";
85 end case;
86
87
88
89
90
91
92
93
94
95

```

```

86
87 -- parpadeo del punto
88 if display_sel="1101" then
89 display(7) <= clk_state;
90 end if;
91 end process;
92
93
94
95 end beh;

```

Componente selecaoFrequencia.

```

1  --Seleção frequência
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  entity frequencia is
5      Port (
6          clk: in std_logic;
7          ContFr: in integer;|
8          Fr : out integer;
9          dis2: out integer;
10         dis3: out integer );
11 end frequencia;
12 architecture Behavioral of frequencia is
13 begin
14 process(clk)
15     begin --inicio do programa
16         if (clk'event and clk='1') then
17             if (ContFr =1 ) then
18                 Fr<=25000;
19                 dis3<=1;
20                 dis2<=0;
21             elsif (ContFr =2 ) then
22                 Fr<=22727;
23                 dis3<=1;
24                 dis2<=1;
25             elsif (ContFr =3 ) then
26                 Fr<=20833;
27                 dis3<=1;
28                 dis2<=2;
29             elsif (ContFr =4 ) then
30
31                 elsif (ContFr =4 ) then
32                     Fr<=19231;
33                     dis3<=1;
34                     dis2<=3;
35                 elsif (ContFr =5 ) then
36                     Fr<=17857;
37                     dis3<=1;
38                     dis2<=4;
39                 elsif (ContFr =6 ) then
40                     Fr<=16667;
41                     dis3<=1;
42                     dis2<=5;
43                 elsif (ContFr =7 ) then
44                     Fr<=15625;
45                     dis3<=1;
46                     dis2<=6;
47                 elsif (ContFr =8 ) then
48                     Fr<=14706;
49                     dis3<=1;
50                     dis2<=7;
51                 elsif (ContFr =9 ) then
52                     Fr<=13889;
53                     dis3<=1;
54                     dis2<=8;
55                 elsif (ContFr =10 ) then
56                     Fr<=13158;
57                     dis3<=1;
58                     dis2<=9;
59                 elsif (ContFr =11 ) then
60
61                     elsif (ContFr =11 ) then
62                         Fr<=12500;
63                         dis3<=2;
64                         dis2<=0;
65                     elsif (ContFr =12 ) then
66                         Fr<=11905;
67                         dis3<=2;
68                         dis2<=1;
69                     elsif (ContFr =13 ) then
70                         Fr<=11364;
71                         dis3<=2;
72                         dis2<=2;
73                     elsif (ContFr =14 ) then
74                         Fr<=10870;
75                         dis3<=2;
76                         dis2<=3;
77                     elsif (ContFr =15 ) then
78                         Fr<=10417;
79                         dis3<=2;
80                         dis2<=4;
81                     elsif (ContFr =16 ) then
82                         Fr<=10000;
83                         dis3<=2;
84                         dis2<=5;
85                     elsif (ContFr =17 ) then

```

```

85     elsif (ContFr =18 ) then
86         Fr<=9259;
87         dis3<=2;
88         dis2<=7;
89     elsif (ContFr =19 ) then
90         Fr<=8929;
91         dis3<=2;
92         dis2<=8;
93     elsif (ContFr =20 ) then
94         Fr<=8621;
95         dis3<=2;
96         dis2<=9;
97     elsif (ContFr =21 ) then
98         Fr<=8333;
99         dis3<=3;
100        dis2<=0;
101     elsif (ContFr =22 ) then
102         Fr<=8065;
103         dis3<=3;
104         dis2<=1;
105     elsif (ContFr =23 ) then
106         Fr<=7813;
107         dis3<=3;
108         dis2<=2;
109     elsif (ContFr =24 ) then
110         Fr<=7576;
111         dis3<=3;
112         dis2<=3;
113     elsif (ContFr =25 ) then

```

```

113     elsif (ContFr =25 ) then
114         Fr<=7353;
115         dis3<=3;
116         dis2<=4;
117     elsif (ContFr =26 ) then
118         Fr<=7143;
119         dis3<=3;
120         dis2<=5;
121     elsif (ContFr =27 ) then
122         Fr<=6944;
123         dis3<=3;
124         dis2<=6;
125     elsif (ContFr =28 ) then
126         Fr<=6757;
127         dis3<=3;
128         dis2<=7;
129     elsif (ContFr =29 ) then
130         Fr<=6579;
131         dis3<=3;
132         dis2<=8;
133     elsif (ContFr =30 ) then
134         Fr<=6410;
135         dis3<=3;
136         dis2<=9;
137     elsif (ContFr =31 ) then
138         Fr<=6250;
139         dis3<=4;
140         dis2<=0;
141     elsif (ContFr =32 ) then

```

```

141     elsif (ContFr =32 ) then
142         Fr<=6098;
143         dis3<=4;
144         dis2<=1;
145     elsif (ContFr =33 ) then
146         Fr<=5952;
147         dis3<=4;
148         dis2<=2;
149     elsif (ContFr =34 ) then
150         Fr<=5814;
151         dis3<=4;
152         dis2<=3;
153     elsif (ContFr =35 ) then
154         Fr<=5682;
155         dis3<=4;
156         dis2<=4;
157     elsif (ContFr =36 ) then
158         Fr<=5556;
159         dis3<=4;
160         dis2<=5;
161     elsif (ContFr =37 ) then
162         Fr<=5435;
163         dis3<=4;
164         dis2<=6;
165     elsif (ContFr =38 ) then
166         Fr<=5319;
167         dis3<=4;
168         dis2<=7;
169     elsif (ContFr =39 ) then

```

```

169         elsif (ContFr =39 ) then
170             Fr<=5208;
171             dis3<=4;
172             dis2<=8;
173         elsif (ContFr =40 ) then
174             Fr<=5102;
175             dis3<=4;
176             dis2<=9;
177         elsif (ContFr =41 ) then
178             Fr<=5000;
179             dis3<=5;
180             dis2<=0;
181         elsif (ContFr =42 ) then
182             Fr<=4902;
183             dis3<=5;
184             dis2<=1;
185         elsif (ContFr =43 ) then
186             Fr<=4808;
187             dis3<=5;
188             dis2<=2;
189         elsif (ContFr =44 ) then
190             Fr<=4717;
191             dis3<=5;
192             dis2<=3;
193         elsif (ContFr =45 ) then
194             Fr<=4630;
195             dis3<=5;
196             dis2<=4;
197         elsif (ContFr =46 ) then

```

```

197         elsif (ContFr =46 ) then
198             Fr<=4545;
199             dis3<=5;
200             dis2<=5;
201         elsif (ContFr =47 ) then
202             Fr<=4464;
203             dis3<=5;
204             dis2<=6;
205         elsif (ContFr =48 ) then
206             Fr<=4386;
207             dis3<=5;
208             dis2<=7;
209         elsif (ContFr =49 ) then
210             Fr<=4310;
211             dis3<=5;
212             dis2<=8;
213         elsif (ContFr =50 ) then
214             Fr<=4237;
215             dis3<=5;
216             dis2<=9;
217         elsif (ContFr =51 ) then
218             Fr<=4167;
219             dis3<=6;
220             dis2<=0;
221         elsif (ContFr =52 ) then
222             Fr<=4098;
223             dis3<=6;
224             dis2<=1;
225         elsif (ContFr =53 ) then

```

```

225         elsif (ContFr =53 ) then
226             Fr<=4032;
227             dis3<=6;
228             dis2<=2;
229         elsif (ContFr =54 ) then
230             Fr<=3968;
231             dis3<=6;
232             dis2<=3;
233         elsif (ContFr =55 ) then
234             Fr<=3906;
235             dis3<=6;
236             dis2<=4;
237         elsif (ContFr =56 ) then
238             Fr<=3846;
239             dis3<=6;
240             dis2<=5;
241         elsif (ContFr =57 ) then
242             Fr<=3788;
243             dis3<=6;
244             dis2<=6;
245         elsif (ContFr =58 ) then
246             Fr<=3731;
247             dis3<=6;
248             dis2<=7;
249         elsif (ContFr =59 ) then
250             Fr<=3676;
251             dis3<=6;
252             dis2<=8;
253         elsif (ContFr =60 ) then

```

```

253         elsif (ContFr =60 ) then
254             Fr<=3623;
255             dis3<=6;
256             dis2<=9;
257         elsif (ContFr =61 ) then
258             Fr<=3571;
259             dis3<=7;
260             dis2<=0;
261         elsif (ContFr =62 ) then
262             Fr<=3521;
263             dis3<=7;
264             dis2<=1;
265         elsif (ContFr =63 ) then
266             Fr<=3472;
267             dis3<=7;
268             dis2<=2;
269         elsif (ContFr =64 ) then
270             Fr<=3425;
271             dis3<=7;
272             dis2<=3;
273         elsif (ContFr =65 ) then
274             Fr<=3378;
275             dis3<=7;
276             dis2<=4;
277         elsif (ContFr =66 ) then
278             Fr<=3333;
279             dis3<=7;
280             dis2<=5;
281         elsif (ContFr =67 ) then
282             Fr<=3289;
283             dis3<=7;
284             dis2<=6;
285         elsif (ContFr =68 ) then
286             Fr<=3247;
287             dis3<=7;
288             dis2<=7;
289         elsif (ContFr =69 ) then
290             Fr<=3205;
291             dis3<=7;
292             dis2<=8;
293         elsif (ContFr =70 ) then
294             Fr<=3165;
295             dis3<=7;
296             dis2<=9;
297         elsif (ContFr =71 ) then
298             Fr<=3125;
299             dis3<=8;
300             dis2<=0;
301         elsif (ContFr =72 ) then
302             Fr<=3086;
303             dis3<=8;
304             dis2<=1;
305         elsif (ContFr =73 ) then
306             Fr<=3049;
307             dis3<=8;
308             dis2<=2;
309         elsif (ContFr =74 ) then

```

```

337      Fr<=2809;
338      dis3<=8;
339      dis2<=9;
340      elsif (ContFr =81 ) then
341      Fr<=2778;
342      dis3<=9;
343      dis2<=0;
344      elsif (ContFr =82 ) then
345      Fr<=2747;
346      dis3<=9;
347      dis2<=1;
348      elsif (ContFr =83 ) then
349      Fr<=2717;
350      dis3<=9;
351      dis2<=2;
352      elsif (ContFr =84 ) then
353      Fr<=2688;
354      dis3<=9;
355      dis2<=3;
356      elsif (ContFr =85 ) then
357      Fr<=2660;
358      dis3<=9;
359      dis2<=4;
360      elsif (ContFr =86 ) then
361      Fr<=2632;
362      dis3<=9;
363      dis2<=5;
364      elsif (ContFr =87 ) then
365      Fr<=2604;
366
365      Fr<=2604;
366      dis3<=9;
367      dis2<=6;
368      elsif (ContFr =88 ) then
369      Fr<=2577;
370      dis3<=9;
371      dis2<=7;
372      elsif (ContFr =89 ) then
373      Fr<=2551;
374      dis3<=9;
375      dis2<=8;
376      elsif (ContFr =90 ) then
377      Fr<=2525;
378      dis3<=9;
379      dis2<=9;
380      else
381      Fr<=2500;
382      dis3<=0;
383      dis2<=0;
384      end if;
385      end if;
386      end process;
387      end Behavioral;

```

Componente selecaoAmplitude.


```

1  --Seleção Amplitude
2  library IEEE;
3  use IEEE.STD_LOGIC_1164.ALL;
4  entity amplitude is
5  Port (
6      clk: in std_logic;
7      ContSen: in integer;
8      AmpSen : out integer;
9      dis0: out integer;
10     dis1: out integer );
11 end amplitude;
12 architecture Behavioral of amplitude is
13 begin
14 process(clk) --inicio processamento do clock
15     begin --inicio do programa
16         IF (clk'EVENT and clk = '1') THEN --executa quando há subida do clock
17             if (ContSen = 1 ) then
18                 AmpSen<=10;
19                 dis1<=1;
20                 dis0<=0;
21             elsif (ContSen = 2 ) then
22                 AmpSen<=11;
23                 dis1<=1;
24                 dis0<=1;
25             elsif (ContSen = 3 ) then
26                 AmpSen<=12;
27                 dis1<=1;
28                 dis0<=2;
29             elsif (ContSen = 4 ) then
30                 AmpSen<=13;
31                 dis1<=1;
32                 dis0<=3;
33             elsif (ContSen = 5 ) then
34                 AmpSen<=14;
35                 dis1<=1;
36                 dis0<=4;
37             elsif (ContSen = 6 ) then
38                 AmpSen<=15;
39                 dis1<=1;
40                 dis0<=5;
41             elsif (ContSen = 7 ) then
42                 AmpSen<=16;
43                 dis1<=1;
44                 dis0<=6;
45             elsif (ContSen = 8 ) then
46                 AmpSen<=17;
47                 dis1<=1;
48                 dis0<=7;
49             elsif (ContSen = 9 ) then
50                 AmpSen<=18;
51                 dis1<=1;
52                 dis0<=8;
53             elsif (ContSen = 10 ) then
54                 AmpSen<=19;
55                 dis1<=1;
56                 dis0<=9;
57             elsif (ContSen = 11 ) then
58                 AmpSen<=20;
59                 dis1<=2;
60                 dis0<=0;
61             elsif (ContSen = 12 ) then
62                 AmpSen<=21;
63                 dis1<=2;
64                 dis0<=1;
65             elsif (ContSen = 13 ) then
66                 AmpSen<=22;
67                 dis1<=2;
68                 dis0<=2;
69             elsif (ContSen = 14 ) then
70                 AmpSen<=23;
71                 dis1<=2;
72                 dis0<=3;
73             elsif (ContSen = 15 ) then
74                 AmpSen<=24;
75                 dis1<=2;
76                 dis0<=4;
77             elsif (ContSen = 16 ) then
78                 AmpSen<=25;
79                 dis1<=2;
80                 dis0<=5;
81             elsif (ContSen = 17 ) then
82                 AmpSen<=26;
83                 dis1<=2;
84                 dis0<=6;
85             elsif (ContSen = 18 ) then

```

```

85     elsif (ContSen =18 ) then
86         AmpSen<=27;
87         dis1<=2;
88         dis0<=7;
89     elsif (ContSen =19 ) then
90         AmpSen<=28;
91         dis1<=2;
92         dis0<=8;
93     elsif (ContSen =20 ) then
94         AmpSen<=29;
95         dis1<=2;
96         dis0<=9;
97     elsif (ContSen =21 ) then
98         AmpSen<=30;
99         dis1<=3;
100        dis0<=0;
101    elsif (ContSen =22 ) then
102        AmpSen<=31;
103        dis1<=3;
104        dis0<=1;
105    elsif (ContSen =23 ) then
106        AmpSen<=32;
107        dis1<=3;
108        dis0<=2;
109    elsif (ContSen =24 ) then
110        AmpSen<=33;
111        dis1<=3;
112        dis0<=3;
113    elsif (ContSen =25 ) then
114
115    elsif (ContSen =25 ) then
116        AmpSen<=34;
117        dis1<=3;
118        dis0<=4;
119    elsif (ContSen =26 ) then
120        AmpSen<=35;
121        dis1<=3;
122        dis0<=5;
123    elsif (ContSen =27 ) then
124        AmpSen<=36;
125        dis1<=3;
126        dis0<=6;
127    elsif (ContSen =28 ) then
128        AmpSen<=37;
129        dis1<=3;
130        dis0<=7;
131    elsif (ContSen =29 ) then
132        AmpSen<=38;
133        dis1<=3;
134        dis0<=8;
135    elsif (ContSen =30 ) then
136        AmpSen<=39;
137        dis1<=3;
138        dis0<=9;
139    elsif (ContSen =31 ) then
140        AmpSen<=40;
141        dis1<=4;
142        dis0<=0;
143    elsif (ContSen =32 ) then
144
145    elsif (ContSen =32 ) then
146        AmpSen<=41;
147        dis1<=4;
148        dis0<=1;
149    elsif (ContSen =33 ) then
150        AmpSen<=42;
151        dis1<=4;
152        dis0<=2;
153    elsif (ContSen =34 ) then
154        AmpSen<=43;
155        dis1<=4;
156        dis0<=3;
157    elsif (ContSen =35 ) then
158        AmpSen<=44;
159        dis1<=4;
160        dis0<=4;
161    elsif (ContSen =36 ) then
162        AmpSen<=45;
163        dis1<=4;
164        dis0<=5;
165    elsif (ContSen =37 ) then
166        AmpSen<=46;
167        dis1<=4;
168        dis0<=6;
169    elsif (ContSen =38 ) then
170        AmpSen<=47;
171        dis1<=4;
172        dis0<=7;
173    elsif (ContSen =39 ) then

```

```

169         elsif (ContSen =39 ) then
170             AmpSen<=48;
171             disl<=4;
172             dis0<=8;
173         elsif (ContSen =40 ) then
174             AmpSen<=49;
175             disl<=4;
176             dis0<=9;
177         elsif (ContSen =41 ) then
178             AmpSen<=50;
179             disl<=5;
180             dis0<=0;
181         elsif (ContSen =42 ) then
182             AmpSen<=51;
183             disl<=5;
184             dis0<=1;
185         elsif (ContSen =43 ) then
186             AmpSen<=52;
187             disl<=5;
188             dis0<=2;
189         elsif (ContSen =44 ) then
190             AmpSen<=53;
191             disl<=5;
192             dis0<=3;
193         elsif (ContSen =45 ) then
194             AmpSen<=54;
195             disl<=5;
196             dis0<=4;
197         elsif (ContSen =46 ) then
198
199             AmpSen<=55;
200             disl<=5;
201             dis0<=5;
202         elsif (ContSen =47 ) then
203             AmpSen<=56;
204             disl<=5;
205             dis0<=6;
206         elsif (ContSen =48 ) then
207             AmpSen<=57;
208             disl<=5;
209             dis0<=7;
210         elsif (ContSen =49 ) then
211             AmpSen<=58;
212             disl<=5;
213             dis0<=8;
214         elsif (ContSen =50 ) then
215             AmpSen<=59;
216             disl<=5;
217             dis0<=9;
218         elsif (ContSen=51 ) then
219             AmpSen<=60;
220             disl<=6;
221             dis0<=0;
222         elsif (ContSen =52 ) then
223             AmpSen<=61;
224             disl<=6;
225             dis0<=1;
226         elsif (ContSen =53 ) then
227
228             AmpSen<=62;
229             disl<=6;
230             dis0<=2;
231         elsif (ContSen =54 ) then
232             AmpSen<=63;
233             disl<=6;
234             dis0<=3;
235         elsif (ContSen =55 ) then
236             AmpSen<=64;
237             disl<=6;
238             dis0<=4;
239         elsif (ContSen =56 ) then
240             AmpSen<=65;
241             disl<=6;
242             dis0<=5;
243         elsif (ContSen=57 ) then
244             AmpSen<=66;
245             disl<=6;
246             dis0<=6;
247         elsif (ContSen =58 ) then
248             AmpSen<=67;
249             disl<=6;
250             dis0<=7;
251         elsif (ContSen =59 ) then
252             AmpSen<=68;
253             disl<=6;
254             dis0<=8;
255         elsif (ContSen =60 ) then

```

```

253     elsif (ContSen =60 ) then
254         AmpSen<=69;
255         disl<=6;
256         dis0<=9;
257     elsif (ContSen =61 ) then
258         AmpSen<=70;
259         disl<=7;
260         dis0<=0;
261     elsif (ContSen =62 ) then
262         AmpSen<=71;
263         disl<=7;
264         dis0<=1;
265     elsif (ContSen =63 ) then
266         AmpSen<=72;
267         disl<=7;
268         dis0<=2;
269     elsif (ContSen =64 ) then
270         AmpSen<=73;
271         disl<=7;
272         dis0<=3;
273     elsif (ContSen =65 ) then
274         AmpSen<=74;
275         disl<=7;
276         dis0<=4;
277     elsif (ContSen =66 ) then
278         AmpSen<=75;
279         disl<=7;
280         dis0<=5;
281     elsif (ContSen =67 ) then

```

```

281     elsif (ContSen =67 ) then
282         AmpSen<=76;
283         disl<=7;
284         dis0<=6;
285     elsif (ContSen =68 ) then
286         AmpSen<=77;
287         disl<=7;
288         dis0<=7;
289     elsif (ContSen =69 ) then
290         AmpSen<=78;
291         disl<=7;
292         dis0<=8;
293     elsif (ContSen =70 ) then
294         AmpSen<=79;
295         disl<=7;
296         dis0<=9;
297     elsif (ContSen =71 ) then
298         AmpSen<=80;
299         disl<=8;
300         dis0<=0;
301     elsif (ContSen =72 ) then
302         AmpSen<=81;
303         disl<=8;
304         dis0<=1;
305     elsif (ContSen =73 ) then
306         AmpSen<=82;
307         disl<=8;
308         dis0<=2;
309     elsif (ContSen =74 ) then

```

```

309     elsif (ContSen =74 ) then
310         AmpSen<=83;
311         disl<=8;
312         dis0<=3;
313     elsif (ContSen =75 ) then
314         AmpSen<=84;
315         disl<=8;
316         dis0<=4;
317     elsif (ContSen =76 ) then
318         AmpSen<=85;
319         disl<=8;
320         dis0<=5;
321     elsif (ContSen =77 ) then
322         AmpSen<=86;
323         disl<=8;
324         dis0<=6;
325     elsif (ContSen =78 ) then
326         AmpSen<=87;
327         disl<=8;
328         dis0<=7;
329     elsif (ContSen =79 ) then
330         AmpSen<=88;
331         disl<=8;
332         dis0<=8;
333     elsif (ContSen =80 ) then
334         AmpSen<=89;
335         disl<=8;
336         dis0<=9;
337     elsif (ContSen =81 ) then

```

```

337         elsif (ContSen = 81 ) then
338             AmpSen<=90;
339             dis1<=9;
340             dis0<=0;
341         elsif (ContSen = 82 ) then
342             AmpSen<=91;
343             dis1<=9;
344             dis0<=1;
345         elsif (ContSen = 83 ) then
346             AmpSen<=92;
347             dis1<=9;
348             dis0<=2;
349         elsif (ContSen = 84 ) then
350             AmpSen<=93;
351             dis1<=9;
352             dis0<=3;
353         elsif (ContSen = 85 ) then
354             AmpSen<=94;
355             dis1<=9;
356             dis0<=4;
357         elsif (ContSen = 86 ) then
358             AmpSen<=95;
359             dis1<=9;
360             dis0<=5;
361         elsif (ContSen = 87 ) then
362             AmpSen<=96;
363             dis1<=9;
364             dis0<=6;
365         elsif (ContSen = 88 ) then
366             AmpSen<=97;
367             dis1<=9;
368             dis0<=7;
369         elsif (ContSen = 89 ) then
370             AmpSen<=98;
371             dis1<=9;
372             dis0<=8;
373         elsif (ContSen = 90 ) then
374             AmpSen<=99;
375             dis1<=9;
376             dis0<=9;
377         else
378             AmpSen<=100;
379             dis1<=0;
380             dis0<=0;
381         end if;
382     end if ;
383     end process;
384 end architecture Behavioral;

```

Arquivo.ucf

```

1  ## UCF
2  NET "clk"          LOC = "B8";
3  ## Buttons
4  NET "button1"      LOC = "B18";      # Amplitude +
5  NET "button2"      LOC = "D18";      # Amplitude -
6  NET "button3"      LOC = "E18";      # frecuencia +
7  NET "button4"      LOC = "H13";      # frecuencia -
8  ## 12 pin connectors
9  NET "S1"           LOC = "G15";      NET "S4"           LOC = "H15";
10 NET "S2"           LOC = "J16";      NET "S5"           LOC = "F14";
11 NET "S3"           LOC = "G13";      NET "S6"           LOC = "G16";
12 ##### DISPLAYS #####
13 NET "display<0>"    LOC = "L18";
14 NET "display<1>"    LOC = "F18";
15 NET "display<2>"    LOC = "D17";
16 NET "display<3>"    LOC = "D16";
17 NET "display<4>"    LOC = "G14";
18 NET "display<5>"    LOC = "J17";
19 NET "display<6>"    LOC = "H14";
20 #NET "display<7>"    LOC = "P16";
21 #anodo
22 NET "cur_display<0>" LOC = "F17";
23 NET "cur_display<1>" LOC = "H17";
24 NET "cur_display<2>" LOC = "C18";
25 NET "cur_display<3>" LOC = "F15";
26 ## Switches
27 NET "sw<0>"         LOC = "G18";
28

```