

**UNIVERSIDADE FEDERAL DE ITAJUBÁ - UNIFEI
INSTITUTO DE SISTEMAS E
TECNOLOGIAS DA INFORMAÇÃO - IESTI**

Maria Clara Ferreira Félix

**PLATAFORMA BASEADA EM DSP E
FPGA PARA CONTROLE DE
CONVERSORES DE POTÊNCIA.**

Relatório Final

Programa: PIVIC 2022-2023

Orientador: Robson Bauwelz Gonzatti

17 de outubro de 2023

Itajubá

Resumo

No contexto atual, um dos mais importantes enfoques de pesquisa na área de Eletrônica de Potência consiste na busca de dispositivos semicondutores de potência que consigam conduzir elevadas correntes e também suportar altos valores de tensão quando bloqueados. Assim, uma das maneiras de se atingir esse objetivo é a associação de conversores estáticos ou associação de células de comutação em série ou em paralelo. Dessa maneira, o estresse de tensão fica dividido por vários componentes e cada um precisa suportar uma tensão menor. Nesse sentido, de acordo com Braga e Barbi (2000), obter níveis intermediários de tensão ou corrente viabilizam a formação de uma forma de onda alternada em degraus, com baixa distorção harmônica, o que permite a redução de perdas e melhoria da estabilidade mecânica dos acionamentos de motores CA, além de minimizar os efeitos sobre o torque da máquina, em comparação com um acionamento empregando uma onda quadrada simples. Com isso, esse projeto apresenta a proposta de uma implementação de um esquema de modulação PWM multinível (por técnica Phase-Shift) para um conversor com três pontes H em cascata, que geram sete níveis de tensão. O modulador foi implementado em uma placa DE0-nano (com FPGA Altera/Intel). Resultados experimentais dos sinais de controle são apresentados para o respectivo conversor.

Palavras-chaves: Conversores multiníveis, FPGA, modulação PWM.

Lista de ilustrações

Figura 1 – Esquemático da modulação PWM	13
Figura 2 – Ponte H inversora de uma fase	14
Figura 3 – Formas de onda referentes a ponte H	14
Figura 4 – Formas de onda da modulação unipolar	15
Figura 5 – Modulação multinível por phase-shift	16
Figura 6 – Três pontes H em cascata para 7 níveis de tensão desenhada no software Kicad 6.0	17
Figura 7 – Vista superior do FPGA Altera Cyclone IV EP4CE22F17C6N	18
Figura 8 – Placa de desenvolvimento DSP F28379D	19
Figura 9 – Placa mãe do Snickerdoodle(Xilinx)	20
Figura 10 – Circuito RTL gerado no Quartus II do bloco PWM	21
Figura 11 – Circuito RTL do bloco Bipolar após compilação no software Quartus II.	22
Figura 12 – Circuito RTL do bloco Unipolar após compilação no software Quartus II	23
Figura 13 – Simulação do Bloco Portadora Triangular - exemplo de contagem até 4 iniciando de 2 na rampa de subida e de 1 na rampa de descida	24
Figura 14 – Simulação do Bloco Portadora Triangular - exemplo de contagem com CntMax=5000, CntIni=2500 e UpDownIni=0	24
Figura 15 – Simulação do Bloco Comparador	25
Figura 16 – Simulação do Bloco PWM - duty-cycle 90% (com polaridades positiva e negativa)	26
Figura 17 – Simulação do Bloco Bipolar	26
Figura 18 – Zoom na simulação do Bloco Bipolar	27
Figura 19 – Simulação do Bloco Unipolar	27
Figura 20 – Simulação do Bloco Três Conversores	28
Figura 21 – Execução no SoC FPGA, com $SW[0]$ em nível alto	29
Figura 22 – Execução no SoC FPGA, com $SW[0]$ em nível baixo	29
Figura 23 – Pulsos de gate para $SW(1:0) = 11$ (90% de duty-cycle)	30
Figura 24 – Cursores indicando período 200 μ s - portanto, frequência de 5kHz	30
Figura 25 – Pulsos de gate para $SW(1:0) = 10$ (50% de duty-cycle)	31
Figura 26 – Pulsos de gate para $SW(1:0) = 00$ (10% de duty-cycle)	31
Figura 27 – Pulsos de gate para $SW(1:0) = 01$ (30% de duty-cycle)	32
Figura 28 – Pulsos de gate para $SW(1:0) = 01$ e polaridade negativa (70% de duty- cycle)	32
Figura 29 – Pulsos de gate cortados por operação na chave <i>reset</i>	33
Figura 30 – Pulsos de gate para $SW(1:0) = 11$ (90% de duty-cycle)	33
Figura 31 – Cursores indicando período 200 μ s - portanto, frequência de 5kHz	34

Figura 32 – Pulsos de gate para $SW(1:0) = 10$ (50% de duty-cycle)	34
Figura 33 – Pulsos de gate para $SW(1:0) = 00$ (10% de duty-cycle)	35
Figura 34 – Pulsos de gate para $SW(1:0) = 01$ (30% de duty-cycle)	35
Figura 35 – Pulsos de gate para $SW(1:0) = 01$ e polaridade negativa (70% de duty-cycle)	36
Figura 36 – Pulsos da ponte $H1$ para $SW(1:0) = 11$ (90% de duty-cycle)	36
Figura 37 – Pulsos da ponte $H2$ para $SW(1:0) = 11$ (90% de duty-cycle)	36
Figura 38 – Pulsos da ponte $H3$ para $SW(1:0) = 11$ (90% de duty-cycle)	37
Figura 39 – Zoom para verificação da defasagem - cursores entre $H1T1$ e $H2T1$. .	37
Figura 40 – Zoom para verificação da defasagem - cursores entre $H2T1$ e $H3T1$. .	37
Figura 41 – Zoom para verificação da defasagem - cursores entre $H3T1$ e $H1T2$. .	38
Figura 42 – Três pontes H em cascata para sete níveis de tensão desenhada no software Kicad 6.0.	39
Figura 43 – Circuito <i>RTL</i> interno ao bloco <i>DDS 3conv.vhd</i>	40
Figura 44 – Fotografia da bancada de testes	41
Figura 45 – Tensões de saída(sinais em amarelo e verde) em duas pontes e soma das tensões em duas pontes(sinal azul)	42
Figura 46 – Sinal de saída na ponte H1 e sinal de saída na ponte H1 filtrado	43
Figura 47 – Espectro de frequências medido na ponte H1	43
Figura 48 – Espectro de frequências da ponte H1 com filtro RC	44

Lista de tabelas

Tabela 3	–	Variação dos Duty-Cycles em relação às chaves $SW[1]$ e $SW[0]$	23
Tabela 4	–	Variação linear de luminosidade de cada LED (<i>Light-emitting diode</i>) com variação logarítmica de duty-cycle		28
Tabela 5	–	Variação dos Duty-Cycles em relação às chaves $SW[1]$ e $SW[0]$	30
Tabela 6	–	Variação dos Duty-Cycles em relação às chaves $SW[1]$ e $SW[0]$	31
Tabela 7	–	Variação dos Duty-Cycles em relação às chaves $SW[1]$ e $SW[0]$	33

Lista de abreviaturas e siglas

DSP	<i>Digital Signal Processor</i>	22
FFT	<i>Fast fourier transform</i>	42
FPGA	<i>Field-programmable gate array</i>	16
IGBT	<i>Insulated Gate Bipolar Transistor</i>	22
LED	<i>Light-emitting diode</i>	5
PWM	<i>Pulse-width modulation</i>	13
RC	<i>Filtro resistor-capacitor</i>	42
SoC	<i>System on chip</i>	16

Lista de símbolos

<i>AC</i>	Alternate current - corrente alternada	40
<i>Bot1</i>	Bottom 1 - Saída de baixo 1 da ponte H	21
<i>Bot2</i>	Bottom 2 - Saída de baixo 2 da ponte H	21
<i>C</i>	Capacitância em Farads	42
<i>CntIni</i>	Contagem inicial	21
<i>ContMax</i>	Contagem máxima	20
<i>DDS</i>	O núcleo DDS IP é uma implementação de um sintetizador de frequência digital direta (DDS) que produz uma onda senoidal na saída com uma frequência e fase especificadas (ajustáveis em tempo de execução).	4
<i>DutyCycle</i>	Ciclo de trabalho - fração de tempo em que um sistema permanece ativo	13
<i>Err1</i>	Entrada de erro 1 proveniente dos drivers	22
<i>Err1DSP</i>	Saída de erro 1 que pode ser enviada a um microcontrolador	22
<i>Err2</i>	Entrada de erro 2 - proveniente dos drivers	22
<i>Err2DSP</i>	Saída de erro 2 - que pode ser enviada a um microcontrolador	22
<i>H</i>	A ponte H é um arranjo de chaves que serve para inverter a polaridade de uma carga sem a necessidade de utilizar uma fonte simétrica	4
<i>H1T1</i>	Sinal de gate 1 da ponte 1	4
<i>H1T2</i>	Sinal de gate 2 da ponte 1	4
<i>H2T1</i>	Sinal de gate 1 da ponte 2	4
<i>H3T1</i>	Sinal de gate 1 da ponte 3	4
<i>OU</i>	Porta lógica que resulta em 1 quando qualquer entrada vale 1, do contrário resulta em 0	22
<i>Pol</i>	Entrada do bloco comparador que define a polaridade	21
<i>Pulso</i>	Polaridade da saída	21
<i>R</i>	Resistência em ohms	42
<i>RTL</i>	Gate-level connections	4
<i>RodaLivre</i>	Sinal resultante da porta OU que controla os quatro sinais de gate da ponte H	22
<i>SW</i>	Slide Switch - Interruptor deslizante	3
<i>Senoide</i>	Sinal ligado a saída do bloco Modulante	40
<i>Top1</i>	Top 1 - Saída de cima 1 da ponte H	21
<i>Top2</i>	Top 2 - Saída de cima 2 da ponte H	21
<i>Tri</i>	Saída do bloco portadora triangular	21
<i>Triangular</i>	Entrada unsigned de 32 bits do bloco comparador	21
<i>UpDownIni</i>	Entrada que define o estado da contagem	21

$Vg1$	Sinal de gate superior 1	13
$Vg3$	Sinal de gate superior 3, paralelo ao sinal superior 1	13
f_{CLK}	Frequência do clock	20
f_{PWM}	Frequência do PWM	20
f_c	Frequência de corte	42
$reset$	Sinal de reinício de contagem	3
t	Tempo	24
$u1$	Instância 1 do bloco PWM	22
$u2$	Instância 2 do bloco PWM	23
$u3$	Instância 3 do bloco PWM	23

Sumário

1	INTRODUÇÃO	11
2	OBJETIVOS PROPOSTOS	12
3	REFERENCIAL TEÓRICO	13
3.1	Princípio da modulação PWM	13
3.2	Princípio da modulação bipolar	13
3.3	Princípio da modulação unipolar	14
3.4	Princípio da modulação multiníveis por phase-shift	15
3.5	FPGA Altera Cyclone IV EP4CE22F17C6N	16
4	DESCRIÇÃO DAS ATIVIDADES DESENVOLVIDAS	19
4.1	Arquitetura em blocos	20
4.1.1	Bloco portadora triangular	20
4.1.2	Bloco comparador	21
4.1.3	Bloco PWM	21
4.1.4	Bloco bipolar	21
4.1.5	Bloco unipolar	22
4.1.6	Bloco três conversores	23
4.2	Simulações no ModelSim	24
4.2.1	Simulação do bloco portadora triangular	24
4.2.2	Simulação do bloco comparador	24
4.2.3	Simulação do bloco PWM	25
4.2.4	Simulação do bloco bipolar	25
4.2.5	Simulação do bloco unipolar	26
4.2.6	Simulação do bloco três conversores	26
4.3	Testes no SoC FPGA	27
4.3.1	Testes do bloco PWM	27
4.3.2	Testes do bloco Bipolar	28
4.3.3	Testes do bloco Unipolar	31
4.3.4	Testes do bloco TresConversores	33
5	RESULTADOS EXPERIMENTAIS	39
5.1	Utilização de core DDS para geração de referências senoidais	40
5.2	Testes em bancada com conversor multiníveis a IGBTs	40
6	CONSIDERAÇÕES FINAIS	45

APÊNDICES	46
APÊNDICE A – ASSINATURA DO ALUNO E DO ORIENTADOR	47
REFERÊNCIAS	48

1 Introdução

Tendo em vista o desenvolvimento da eletrônica de potência em todo o mundo, uma das questões que vem sendo mais aprimorada relaciona-se à busca de dispositivos semicondutores de potência que sejam capazes de conduzir elevadas correntes e paralelamente suportarem altas tensões. Neste sentido foram desenvolvidos novos dispositivos na última década, cada um dos quais apresentando características máximas de tensão, corrente e frequência, funcionando para diferentes aplicações. Entretanto, os novos dispositivos são normalmente mais caros, de difícil acesso e pouco conhecidos. Dessa maneira, é preferível empregar componentes bem estabelecidos no mercado, de baixo custo (Braga e Barbi, 2000)[1].

Se uma determinada aplicação exige valores de tensão ou corrente superiores aos que o dispositivo escolhido é capaz de lidar, recorre-se ao processo de associação de componentes. Associação série para grandes tensões e paralela para grandes correntes. Como apontado por Braga e Barbi (2000), alguns problemas comuns são: dificuldade de sincronismo na entrada em condução e bloqueio, variações bruscas de tensão ou corrente nos dispositivos, instabilidade térmica, entre outros.

A geração de estruturas multiníveis é uma possível solução para alguns dos problemas encontrados nas associações série e paralelo dos componentes semicondutores. A ideia principal consiste em reduzir o estresse aplicado nos conversores a partir da geração de níveis de tensão intermediários, divididos em cada um dos conversores. Nesse sentido, esse trabalho apresenta a proposta de uma implementação de um esquema de modulação PWM para um conversor com três pontes H em cascata, que geram sete níveis de tensão, o qual foi baseado no artigo de Sant'ana (2018e)[2]. Foi gerada uma onda alternada em graus, com baixa distorção harmônica. Assim, o formato dessa onda minimizaria os efeitos sobre o torque dos motores CA, quando se compara com um acionamento empregando uma onda quadrada simples.

Nesse contexto, a baixa distorção harmônica é uma característica positiva que foi atingida com a geração de uma onda senoidal com níveis intermediários de tensão nesse trabalho. A distorção harmônica ocorre quando um sistema gera sinais que contêm componentes de frequência indesejados ou harmônicos que não estavam presentes no sinal original. Em outras palavras, no espectro de frequências gerado, foi possível reduzir essas frequências indesejadas, o que melhoraria a qualidade do equipamento ou sistema em uma aplicação real.

2 Objetivos propostos

- Promover o desenvolvimento de um esquema de modulação PWM multinível implementando-o em FPGA a fim de se gerar uma onda com baixa distorção harmônica;
- Testar modularmente cada bloco escrito em VHDL que constituiu o sistema final;
- Monitorar as ondas de entrada e saída do FPGA da *Terasic* modelo Altera Cyclone IV EP4CE22F17C6N;
- Utilizar a modulação PWM gerada em um conversor baseado em três pontes H em cascata que gerariam sete níveis de tensão, reduzindo o estresse de tensão em cada conversor e tendo uma alta frequência de chaveamento.

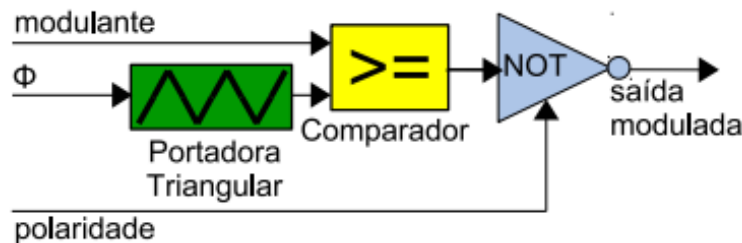
3 Referencial teórico

3.1 Princípio da modulação PWM

O princípio da modulação por largura de pulso *PWM (Pulse-width modulation)* é uma técnica utilizada em eletrônica e engenharia elétrica para controlar a energia entregue a dispositivos elétricos, como motores, lâmpadas e circuitos de controle de potência. O PWM funciona variando a largura dos pulsos em um sinal. O ciclo de trabalho (*DutyCycle*), expresso como uma porcentagem, representa a fração de tempo em que o sinal está em nível alto em relação ao período total do sinal. A principal aplicação do PWM é o controle de velocidade de motores elétricos.

Utilizando como base a figura 1, pode-se concluir que para qualquer valor do sinal modulante maior ou igual ao da portadora triangular, que é uma onda triangular utilizada para comparação com a onda senoidal, a saída apresentará nível alto. E para qualquer valor do sinal modulante menor do que o da portadora triangular, a saída apresentará nível baixo (Sant'ana, 2018a).

Figura 1 – Esquemático da modulação PWM

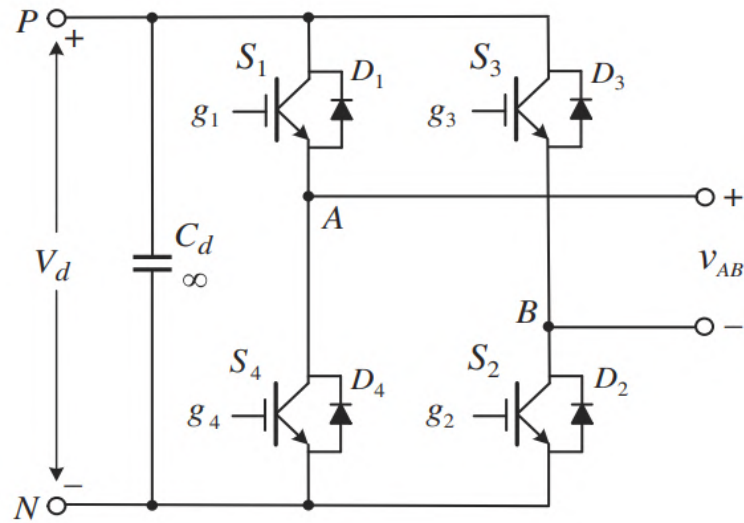


Fonte: Sant'ana, 2018a.[3].

3.2 Princípio da modulação bipolar

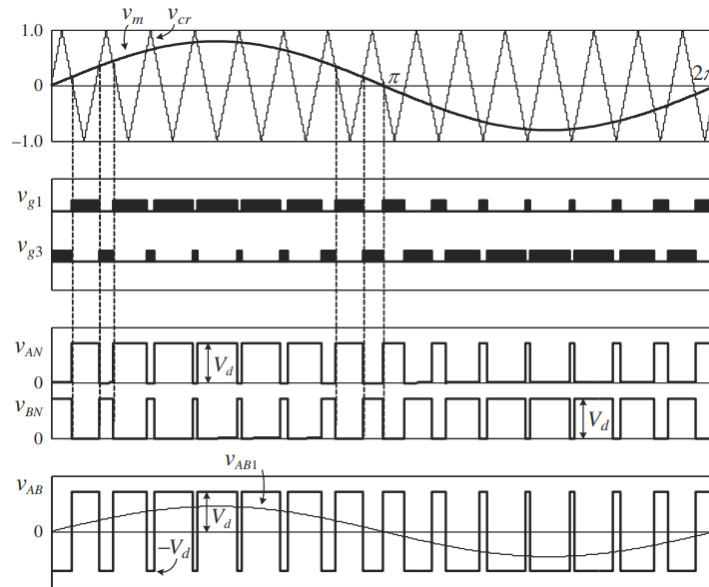
A Figura 3 mostra um conjunto de formas de onda típicas de um inversor ponte H com modulação bipolar, a qual é mostrada na figura 2. Pode-se observar que os sinais V_{g1} e V_{g3} , que são os sinais superiores paralelos na ponte H, são justamente gerados pela comparação da onda moduladora senoidal e da onda portadora triangular. Nesse contexto, esses dois sinais de gate operam de forma complementar na modulação bipolar. Quando a senóide é maior ou igual a onda triangular, tem-se a saída em nível alto. E quando a senóide é menor que a portadora triangular, a saída apresenta nível baixo. Esse esquema é conhecido como modulação bipolar.

Figura 2 – Ponte H inversora de uma fase



Fonte: Wu e Narimani (2017)[4].

Figura 3 – Formas de onda referentes a ponte H



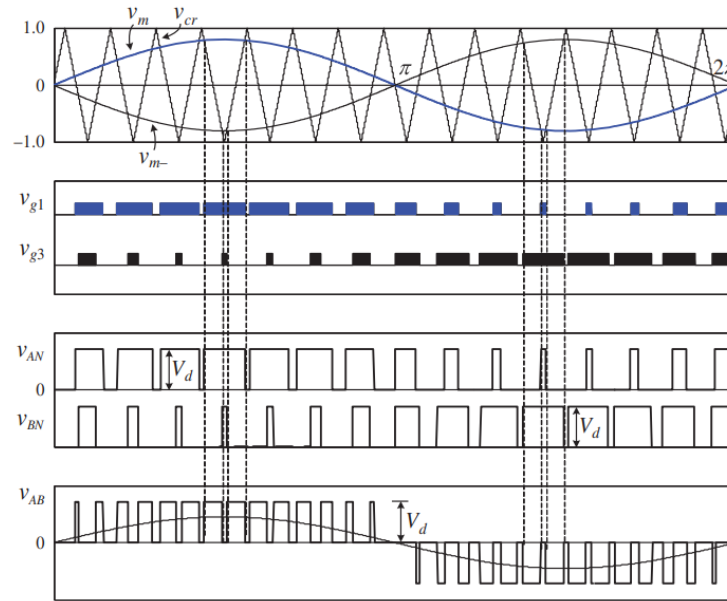
Fonte: Wu e Narimani (2017)[4].

3.3 Princípio da modulação unipolar

Em contraposição com a modulação bipolar, a modulação unipolar normalmente requer duas ondas moduladoras senoidais, que têm a mesma magnitude e frequência, mas estão defasadas em 180 graus, como é exibido na Figura 4. As duas ondas moduladoras são comparadas com uma onda portadora triangular comum, gerando dois sinais de portão, V_{g1} e V_{g3} , para os interruptores superiores. Pode-se observar que elas não comutam simultaneamente, em contrapartida com a modulação bipolar. Dessa maneira, a tensão de saída do inversor alterna entre zero e uma tensão positiva durante o semiciclo positivo ou entre zero e uma tensão negativa durante o semiciclo negativo da frequência fundamental

(Wu e Narimani, 2017). Portanto, esse esquema é conhecido como modulação unipolar.

Figura 4 – Formas de onda da modulação unipolar



Fonte: Wu e Narimani[4].

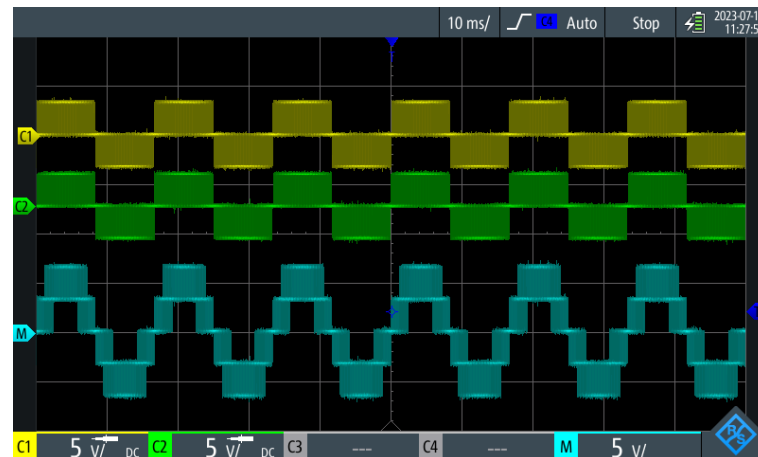
3.4 Princípio da modulação multiníveis por phase-shift

A modulação multinível por phase-shift é uma técnica de controle usada em inversores de tensão multinível, que são dispositivos eletrônicos empregados para converter uma fonte de tensão contínua em uma tensão alternada com níveis de amplitude distintos. Essa técnica é usada para melhorar a qualidade do sinal de saída, reduzir a distorção harmônica e melhorar a eficiência do sistema.

A modulação multinível por phase-shift envolve a geração de múltiplos níveis de tensão de saída, com cada nível sendo gerado com um deslocamento de fase em relação aos outros. Em sistemas multiníveis, em vez de ter apenas dois níveis de tensão de saída (como em inversores convencionais), você pode ter vários níveis intermediários, criando uma forma de onda de saída mais próxima de uma onda senoidal ideal. Na figura 5 pode-se observar o resultado obtido no osciloscópio da modulação multinível gerada a partir do esquema das três pontes H em cascata para sete níveis de tensão, conforme exibe a figura 6.

Cabe ressaltar que só foi possível somar dois canais utilizando este osciloscópio, dessa maneira, a onda em azul, que é a soma dos dois canais, é equivalente à saída de um conversor com duas pontes. Apesar de estar gerando os sinais pra três pontes, não foi possível exibir a saída completa. Dessa maneira, a saída possui cinco níveis de tensão intermediários.

Figura 5 – Modulação multinível por phase-shift



Fonte: Autoria própria.

3.5 FPGA Altera Cyclone IV EP4CE22F17C6N

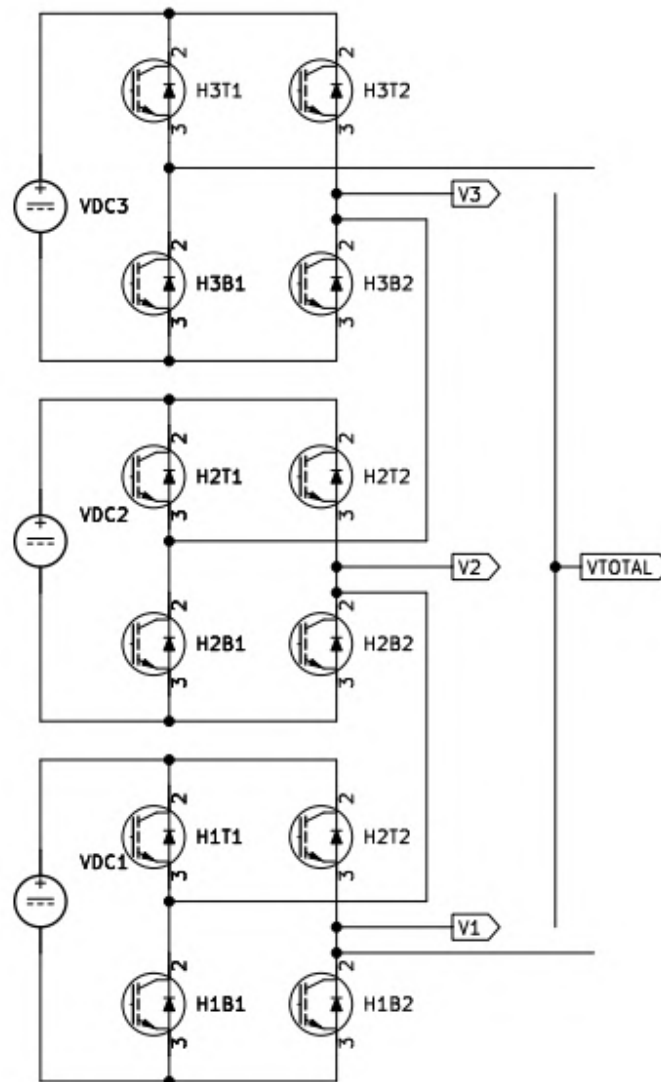
O DE0-nano é uma placa de desenvolvimento baseada em um *FPGA* (*Field-programmable gate array*) produzida pela empresa Terasic Technologies. O FPGA é um dispositivo lógico programável que permite a criação de circuitos digitais. Nele pode-se encontrar o próprio *SoC* (*System on chip*) (Cyclone IV) que pode ser usado para gerenciar o sistema e interagir com os periféricos. Também é possível encontrar recursos de entrada e saída: a placa inclui vários pinos de I/O para conexão de dispositivos externos e periféricos.

Também possui portas USB, interfaces de comunicação como UART e conectividade para VGA, além de outros recursos. Em relação a memória, possui memória FLASH para armazenamento de firmware e memória SDRAM para armazenamento de dados. Neste caso, apenas foi utilizado a memória SDRAM. O software de desenvolvimento utilizado para programar o DE0-Nano foi o Quartus II na versão 13.0. A linguagem de descrição de hardware utilizada foi VHDL, mas os programas também poderiam ter sido escritos em Verilog. Além disso, para programar o microcontrolador ARM, geralmente se utiliza um ambiente de desenvolvimento como o Eclipse.

De acordo com *Terasic* (2019)[5], os recursos oferecidos são:

- Dispositivo Cyclone® V SE 5CSEBA6U23I7;
- Dispositivo de configuração serial – EPCS128;
- USB-Blaster II integrado para programação; Modo JTAG;
- 2 botões;
- 4 interruptores deslizantes;

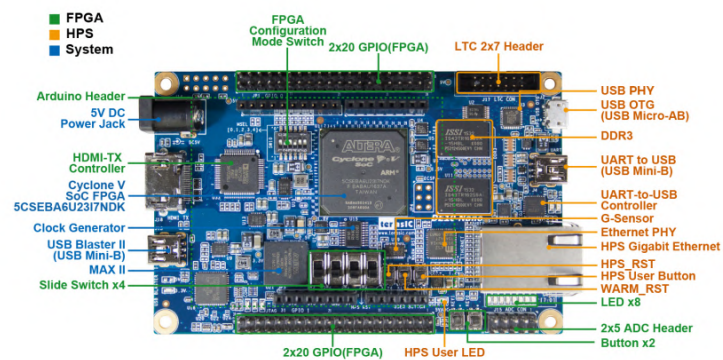
Figura 6 – Três pontes H em cascata para 7 níveis de tensão desenhada no software Kicad 6.0



Fonte: Autoria própria.

- 8 LEDs verdes;
- Três 50MHz fontes de clock do gerador de clock;
- Dois conectores de expansão de 40 pinos;
- Um header de expansão Arduino (compatível com Uno R3), que pode ser conectado com os shields do Arduino;
- Um header de entrada de expansão de 10 pinos. (compartilhado com a entrada analógica do Arduino);
- A/D conversor, interface de 4 fios SPI interface com FPGA;
- HDMI TX, compatível com DVI v1.0 e HDCP v1.4.

Figura 7 – Vista superior do FPGA Altera Cyclone IV EP4CE22F17C6N



Fonte: *Terasic* (2019)[5].

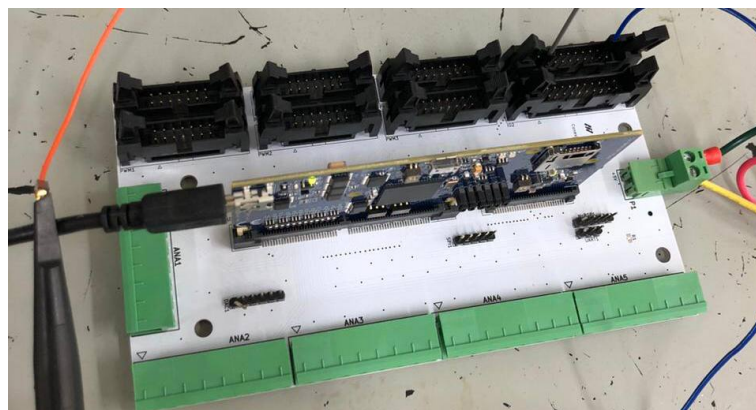
4 Descrição das atividades desenvolvidas

Inicialmente, a ideia deste trabalho era o desenvolvimento de um esquema de modulação PWM para sete níveis de tensão, utilizando três pontes H em cascata no microcontrolador Snickerdoodle(Xilinx/Zynq), como exibe a figura 9. Dessa maneira, utilizando o ambiente de desenvolvimento Vivado, em conjunto com uma placa mãe, seria possível acessar os pinos de entrada e saída desse microcontrolador. A vantagem seria a extensão do projeto, visto que esse microcontrolador possui 180 pinos GPIO, em contrapartida com os 26 pinos GPIO da placa DE0-nano.

Dessa maneira, depois seria feita a comunicação do Snickerdoodle com o DSP F28379D na placa mãe, conforme exibe a figura 8. Assim, foi estudado e testado nesse DSP suas funções principais, como GPIO, Interrupções e Watchdog, ADC e PWM. Nesse sentido, os projetos realizados em VHDL no Quartus II também foram realizados no software Vivado, todavia, não foram compilados no Snickerdoodle por problemas técnicos. Portanto, optou-se pelo caminho mais fácil, que seria utilizar a placa DE0-nano (com FPGA Altera/Intel).

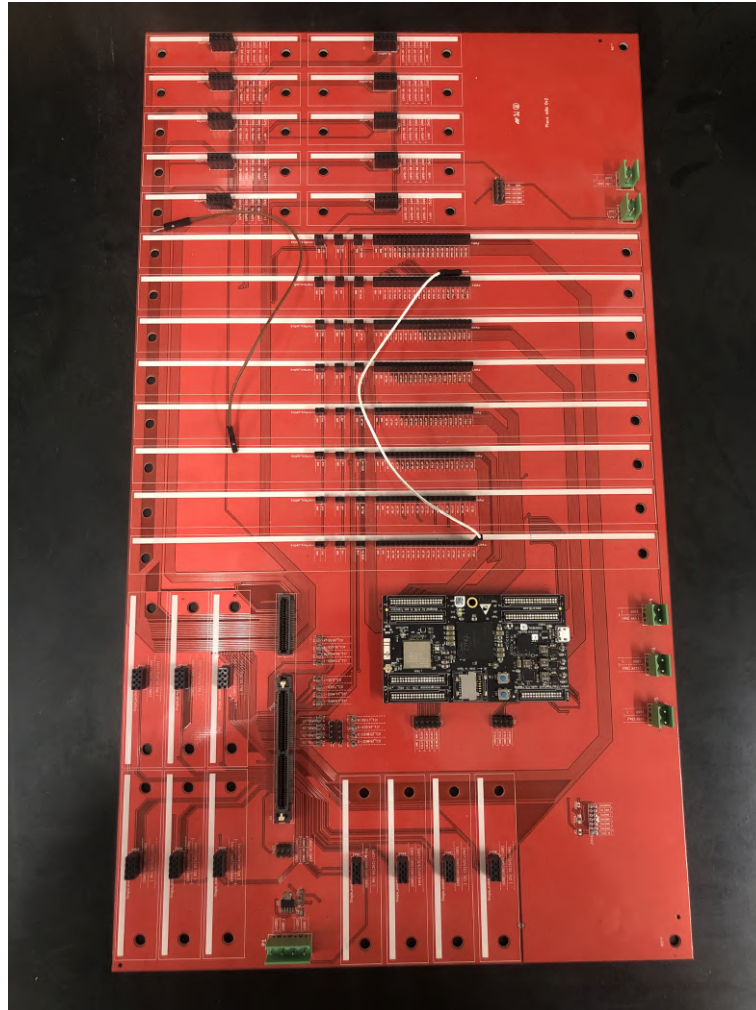
Especificamente, para se atingir o objetivo de desenvolver um modulador PWM, foi utilizada uma arquitetura em blocos. Ou seja, utilizando o software Quartus II foram criados projetos pequenos em VHDL em que cada bloco é responsável por uma função específica, e esses blocos quando combinados geraram o sistema completo. Os blocos criados foram: bloco portadora triangular, bloco comparador, bloco PWM, bloco bipolar, bloco unipolar e bloco três conversores. Cada bloco consiste em um projeto diferente, os quais foram testados separadamente na placa DE0-nano (com FPGA Altera/Intel).

Figura 8 – Placa de desenvolvimento DSP F28379D



Fonte: Autoria própria.

Figura 9 – Placa mãe do Snickerdoodle(Xilinx)



Fonte: Autoria própria.

4.1 Arquitetura em blocos

4.1.1 Bloco portadora triangular

Este bloco implementa um contador de 32 bits do tipo up-down. Ele é um contador bidirecional síncrono que tem a habilidade de contar em ambas as direções, ou seja, para algum valor ou desde um valor pré-definido e também de zero. Nesse caso ele vai incrementar de 0 até um valor máximo que foi definido.

É desejado obter uma determinada frequência(f_{PWM}) de PWM que será escolhida. De zero até a contagem máxima tem-se meio período da onda triangular e a contagem é realizada a cada pulso de clock. Portanto, o valor da contagem máxima($ContMax$) deve ser obtido através de 4.1. Considerando que o microcontrolador utilizado possui clock a cristal de 50MHz(f_{CLK}), como resultado tem-se o valor da contagem máxima.

$$f_{PWM} = \frac{f_{CLK}}{2 \cdot ContMax} [3] \quad (4.1)$$

Também foi necessário escolher um valor de contagem inicial que pode ocorrer tanto na rampa de descida quanto na rampa de subida. Definiu-se uma entrada tal que, para essa entrada em nível alto a contagem começa na rampa de subida e para essa entrada em nível baixo a contagem começa na rampa de descida.

Nesse contexto, a cada quantidade de contagens máximas a portadora se desloca de 180° . Assim, os valores de *CntIni* e *UpDownIni* podem ser obtidos de acordo com as equações 4.2 ou 4.3, dependendo do valor do ângulo desejado. Dessa maneira, quando se desejam portadoras atrasadas no tempo, devem ser utilizados ângulos adjacentes da defasagem desejada.

$$0^\circ \leq \phi < 180^\circ \left\{ \begin{array}{l} CntIni = ContMax \cdot \frac{\phi}{180}; \\ UpDownIni = 1. \end{array} \right. \quad [3] \quad (4.2)$$

$$180^\circ \leq \phi < 360^\circ \left\{ \begin{array}{l} CntIni = ContMax \cdot \frac{360-\phi}{180}; \\ UpDownIni = 0. \end{array} \right. \quad [3] \quad (4.3)$$

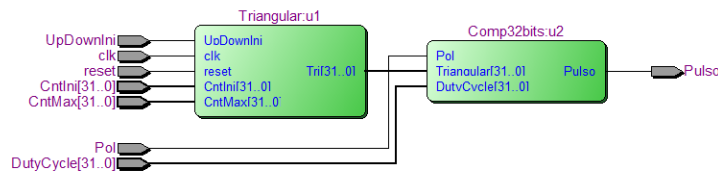
4.1.2 Bloco comparador

Este bloco realiza a comparação entre a portadora triangular e o sinal modulante. As suas entradas (*Triangular* e *DutyCycle*) são números inteiros do tipo unsigned de 32 bits. Uma terceira entrada (*Pol*) determina a polaridade da saída (*Pulso*) de comparação.

4.1.3 Bloco PWM

Este bloco contém uma instância do bloco Portadora Triangular e uma instância do bloco Comparador de 32 bits. Internamente, a saída *Tri* do bloco Portadora Triangular é ligada a entrada *Triangular* do bloco Comparador.

Figura 10 – Circuito RTL gerado no Quartus II do bloco PWM



Fonte: Autoria própria.

4.1.4 Bloco bipolar

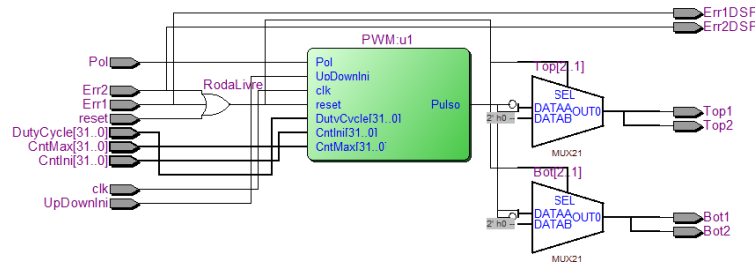
Este bloco consiste em uma instância do bloco PWM acrescido da lógica complementar. O bloco Bipolar herda toda a configuração de entrada do bloco PWM. A diferença está nas quatro saídas (*Top1*, *Bot1*, *Top2* e *Bot2*) para um inversor em ponte H. Alguns

drivers de IGBT (*Insulated Gate Bipolar Transistor*) possuem sistema de detecção de erro. Neste caso os pulsos são cortados e um sinal de erro é ativado (Wilson Cesar Sant’ana, 2018). Como, normalmente, existe um sinal de erro para cada par complementar de IGBTs ($Top1 = Bot1$ e $Top2 = Bot2$), o bloco Bipolar prevê duas entradas ($Err1$ e $Err2$) provenientes dos drivers e duas saídas ($Err1DSP$ e $Err2DSP$) que podem ser enviadas para um microcontrolador/DSP (*Digital Signal Processor*).

A instância do componente PWM é definida como $u1$. Uma operação OU é realizada com as duas entradas de erro e com a entrada de $reset$. O sinal resultante deste OU ($RodaLivre$) controla os quatro sinais de gate da ponte. Quando $RodaLivre$ está em nível alto, seja porque um dos drivers sinalizou erro ou porque o usuário enviou um $reset$, os quatro pulsos são cortados. Nesta condição a ponte opera apenas pelos diodos de roda livre. Quando $RodaLivre$ está em nível baixo, os IGBT $Top1$ e $Bot2$ irão receber a saída do bloco PWM e os IGBT $Top2$ e $Bot1$ irão receber o complementar deste valor (Wilson Cesar Sant’ana, 2018c).

As saídas de erro do bloco Bipolar ($Err1DSP$ e $Err2DSP$) apenas acompanham as entradas recebidas dos drivers ($Err1$ e $Err2$). Desta forma um microcontrolador/DSP pode realizar as operações necessárias para não resultar em erros.

Figura 11 – Circuito RTL do bloco Bipolar após compilação no software Quartus II.



Fonte: Autoria própria.

4.1.5 Bloco unipolar

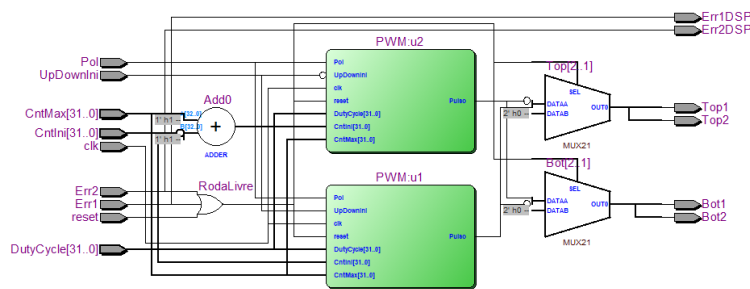
Este bloco consiste em duas instâncias do bloco PWM acrescido da lógica complementar. Ele também herda toda a configuração de entrada do bloco PWM. Assim como o bloco Bipolar, o bloco Unipolar também prevê duas entradas ($Err1$ e $Err2$) provenientes dos drivers e duas saídas ($Err1DSP$ e $Err2DSP$) que podem ser enviadas para um microcontrolador/DSP.

Quando RodaLivre está em nível alto, seja porque um dos drivers sinalizou erro ou porque o usuário enviou um $reset$, os quatro pulsos são cortados. Nesta condição a ponte opera apenas pelos diodos de roda livre. Quando RodaLivre está em nível baixo, o IGBT $Top1$ irá receber a saída do bloco PWM 1 (e $Bot1$ irá receber o valor complementar: $Bot1$

= $Top1$) e o IGBT $Bot2$ irá receber a saída do bloco PWM 2 (e $Top2$ irá receber o valor complementar: $Top2 = Bot2$).

Assim como no bloco Bipolar, as saídas de erro do bloco Bipolar ($Err1DSP$ e $Err2DSP$) acompanham as entradas recebidas dos drivers ($Err1$ e $Err2$). Caso um microcontrolador/DSP esteja utilizando o PWM do FPGA, ele pode realizar as operações necessárias para não resultar em erros. Além disso, as portadoras triangulares dos blocos PWM $u1$ e $u2$ possuem defasagem de 180° , o que está de acordo com o que foi apresentado em 3.3.

Figura 12 – Circuito RTL do bloco Unipolar após compilação no software Quartus II



Fonte: Autoria própria.

4.1.6 Bloco três conversores

Este bloco consiste em três instâncias do bloco Unipolar. As portadoras triangulares dos blocos PWM internos aos blocos Unipolar estão configuradas de forma que apresentem as seguintes defasagens: $u1$ portadora em 0° $to_unsigned(0,32)$ e portadora em 180° configurada automaticamente pelo bloco Unipolar; $u2$ portadora em 60° $to_unsigned(1667,32)$ e portadora em 240° configurada automaticamente pelo bloco Unipolar; $u3$ portadora em 120° $to_unsigned(3333,32)$ e portadora em 240° configurada automaticamente pelo bloco Unipolar (Sant'ana, 2018b) [6].

Tabela 3 – Variação dos Duty-Cycles em relação às chaves $SW[1]$ e $SW[0]$

Duty-cycle	$SW[1]$	$SW[2]$
10%	0	0
30%	0	1
50%	1	0
90%	1	1

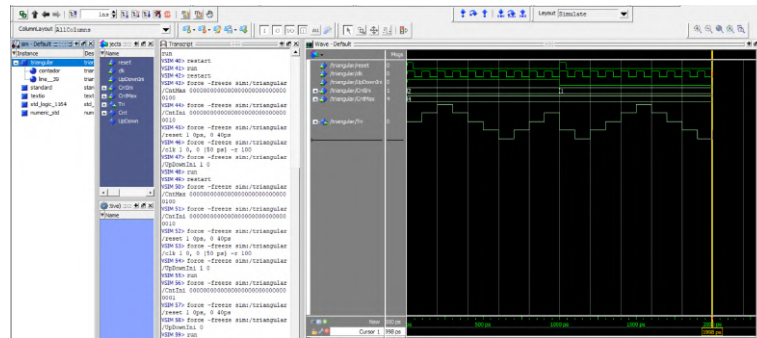
Fonte: Sant'ana (2018b)[6].

4.2 Simulações no ModelSim

4.2.1 Simulação do bloco portadora triangular

Tem-se na Figura 13 um exemplo de funcionamento do bloco para $ContMax=4$. Em $t=40ps$, o sinal reset passa de nível alto para nível baixo, iniciando a contagem. Nesta situação, $CntIni=2$ e $UpDownIni=1$, o que indica que a contagem se iniciará pelo valor 2 na rampa de subida. Em $t=1000ps$, os sinais $CntIni$ e $UpDownIni$ são modificados para $CntIni=1$ e $UpDownIni=0$. Em $t=1000ps$, o sinal reset passa de nível alto para nível baixo, reiniciando a contagem - desta vez do valor inicial 1 na rampa de descida.

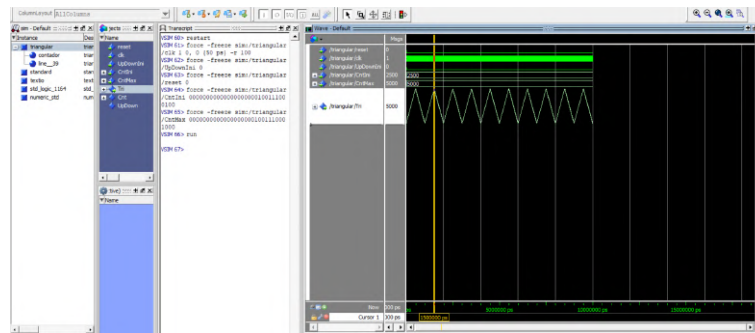
Figura 13 – Simulação do Bloco Portadora Triangular - exemplo de contagem até 4 iniciando de 2 na rampa de subida e de 1 na rampa de descida



Fonte: Autoria própria.

Tem-se na Figura 14 um exemplo de funcionamento do bloco para $ContMax=5000$, $CntIni=2500$ e $UpDownIni=0$ - ou seja, $f_{PWM} = 5kHz$ e portadora atrasada de 90° . O período obtido foi $200000ns=200\mu s$, o que corresponde a uma frequência de 5KHz.

Figura 14 – Simulação do Bloco Portadora Triangular - exemplo de contagem com $ContMax=5000$, $CntIni=2500$ e $UpDownIni=0$



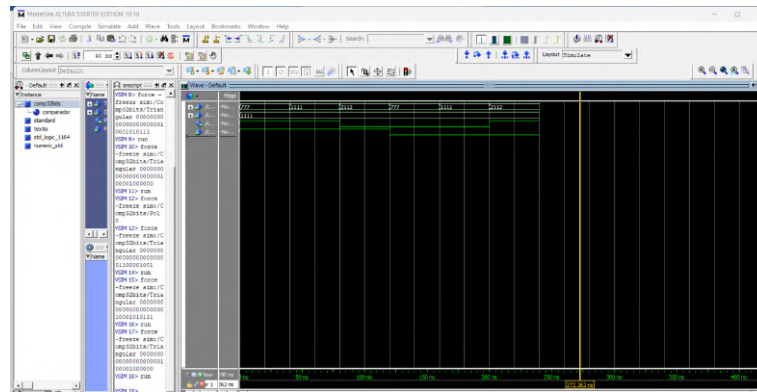
Fonte: Autoria própria.

4.2.2 Simulação do bloco comparador

Tem-se na Figura 15 um exemplo de funcionamento do bloco. Em $t=0ns$, a simulação se inicia com os valores inteiros 1111 e 777 nas entradas $DutyCycle$ e $Triangular$,

respectivamente. A entrada *Pol* se inicia em nível alto. Nesta condição, o teste $DutyCycle \geq Triangular$ é verdadeiro - e como *Pol*=1, a saída *Pulso* receberá nível alto. Em $t=40ns$, *DutyCycle* se mantém em 1111 e *Triangular* muda para 1111. Nesta condição, o teste $DutyCycle \geq Triangular$ continua verdadeiro - e como *Pol* continua em nível alto, a saída *Pulso* continuará em nível alto. Em $t=80ns$, *DutyCycle* se mantém em 1111 e *Triangular* muda para 2112. Nesta condição, o teste $DutyCycle \geq Triangular$ é falso - e como *Pol* continua em nível alto, a saída *Pulso* receberá nível baixo. De $t=120ns$ até o fim da simulação, a mesma sequência em *Triangular* é repetida - porém, desta vez, com *Pol* em nível baixo, obtendo uma saída complementar.

Figura 15 – Simulação do Bloco Comparador



Fonte: Autoria própria.

4.2.3 Simulação do bloco PWM

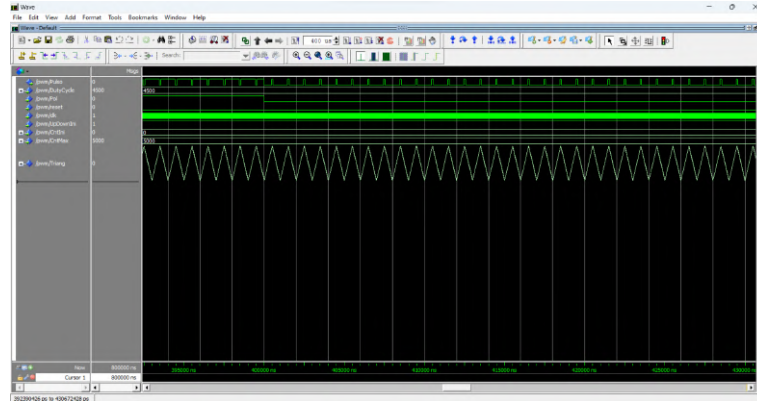
Tem-se na Figura 16 um exemplo de funcionamento do bloco. Em $t=0ns$, a simulação se inicia com a portadora triangular configurada com $ContMax=5000$ (para $f_{PWM} = 5kHz$) e com o valor 4500 na entrada *DutyCycle*. Até $t=400000ns$, a entrada de polaridade da comparação (*Pol*) é configurada em nível alto. Como $4500/5000 = 0.90$, tem-se na saída *Pulso* um sinal com 90% de duty-cycle.

A segunda metade da Figura 16 ocorre a partir de $t=400000ns$, quando a entrada *DutyCycle* continua com os mesmos 4500 mas a entrada de polaridade da comparação (*Pol*) é configurada para nível baixo - resultando, portanto, em um duty-cycle de 10%, que é o complementar de 90%.

4.2.4 Simulação do bloco bipolar

Tem-se na Figura 17 um exemplo de funcionamento do bloco para $ContMax=5000$ (de forma a se obter $f_{PWM} = 5kHz$ com $f_{CLK} = 50MHz$) e $CntIni=0$. Em $t=40ns$, o sinal *reset* passa de nível alto para nível baixo, iniciando a contagem. Nesta situação, a entrada *DutyCycle* contém o valor 4500 (ou seja, 90% de 5000) e a entrada *Pol* contém nível alto. A resultante dos pulsos de saída apresentaria um duty-cycle de 90%. Em $t=400000000ps$

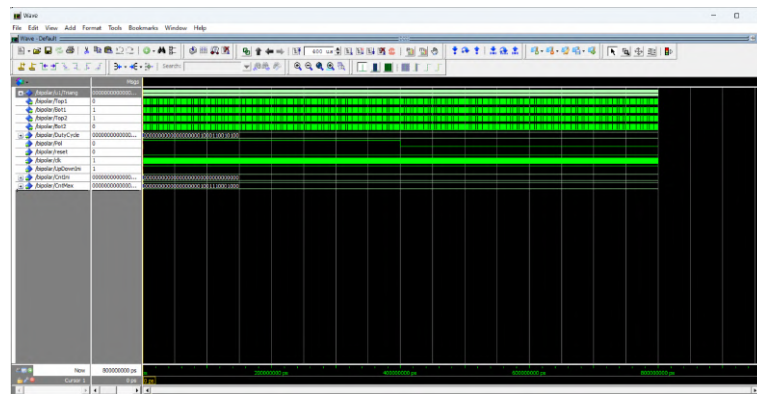
Figura 16 – Simulação do Bloco PWM - duty-cycle 90% (com polaridades positiva e negativa)



Fonte: Autoria própria.

a entrada *Pol* muda para nível baixo enquanto a entrada *DutyCycle* se mantém no valor 4500. A resultante dos pulsos de saída apresentaria um duty-cycle de 10%.

Figura 17 – Simulação do Bloco Bipolar



Fonte: Autoria própria.

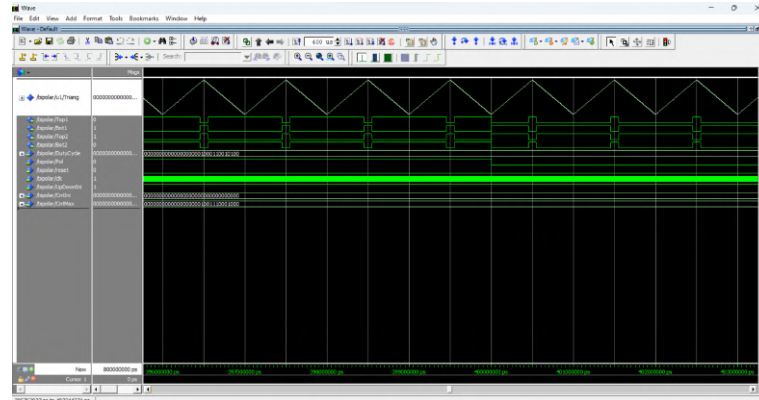
4.2.5 Simulação do bloco unipolar

Tem-se na Figura 19 um exemplo de funcionamento do bloco para *ContMax*=5000 (de forma a se obter $f_{PWM} = 5\text{kHz}$ com $f_{CLK} = 50\text{MHz}$) e *CntIni*=0. Em $t=40\text{ns}$, o sinal *reset* passa de nível alto para nível baixo, iniciando a contagem. Nesta situação, a entrada *DutyCycle* contém o valor 4500 (ou seja, 90% de 5000) e a entrada *Pol* contém nível alto. A resultante dos pulsos de saída apresentaria um duty-cycle de 90%. Em $t=4000\text{ns}$ a entrada *Pol* muda para nível baixo enquanto a entrada *DutyCycle* se mantém no valor 4500. A resultante dos pulsos de saída apresentaria um duty-cycle de 10%.

4.2.6 Simulação do bloco três conversores

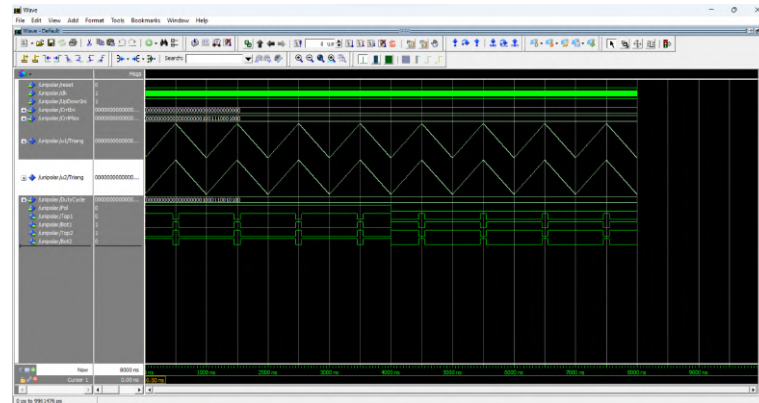
A Figura 20 apresenta um exemplo de funcionamento do bloco para *ContMax*=5000 (de forma a se obter $f_{PWM} = 5\text{kHz}$ com $f_{CLK} = 50\text{MHz}$, e *CntIni*=0.

Figura 18 – Zoom na simulação do Bloco Bipolar



Fonte: Autoria própria.

Figura 19 – Simulação do Bloco Unipolar



Fonte: Autoria própria.

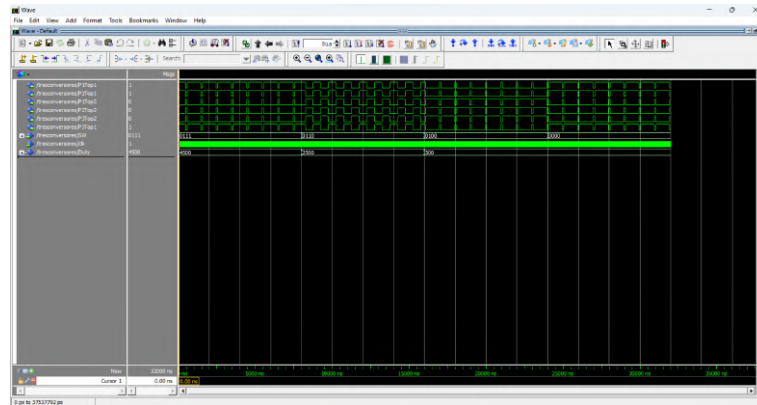
Em $t=20\text{ns}$, o sinal *reset* passa de nível alto para nível baixo, iniciando a contagem. Nesta situação, as chaves *SW*(1:0) contêm o valor 11 (o que resulta no valor 4500 em *DutyCycle* - portanto, 90% de 5000) e a entrada *Pol* contém nível alto. Em $t=8000\text{ns}$, as chaves *SW*(1:0) mudam para o valor 10 (o que resulta no valor 2500 em *DutyCycle* - portanto, 50% de 5000) e a entrada *Pol* continua em nível alto. Em $t=16000\text{ns}$, as chaves *SW*(1:0) mudam para o valor 00 (o que resulta no valor 500 em *DutyCycle* - portanto, 10% de 5000) e a entrada *Pol* continua em nível alto. Em $t=24000\text{ns}$, as chaves *SW*(1:0) continuam com o valor 10, mas a entrada *Pol* muda para nível alto - isto faz com que o duty-cycle resultante seja o complementar de 10%, ou seja, 90%.

4.3 Testes no SoC FPGA

4.3.1 Testes do bloco PWM

Primeiramente, foi feita uma aplicação simples para controlar a luminosidade dos LEDs do microcontrolador, como sugerido em Sant'ana (2018a)[3]. Cada um dos oito LEDs foi comandado de acordo com a Tabela 4. A variação de duty-cycle inserida é

Figura 20 – Simulação do Bloco Três Conversores



Fonte: Autoria própria.

logarítmica - isto é necessário de forma que o efeito visual da luminosidade seja linear - do contrário nossos olhos não conseguiriam ver muita diferença entre o brilho dos diodos. Portanto, foram utilizadas oito instâncias do bloco PWM - sendo que as saídas de cada bloco são ligadas a cada LED. Como os dutycycles precisam ser logarítmicos, a mudança da chave *Pol* para nível baixo apenas vai fazer com que LED [7] se apague e os demais fiquem acesos com uma luminosidade muito difícil de se distinguir.

Tabela 4 – Variação linear de luminosidade de cada LED com variação logarítmica de duty-cycle

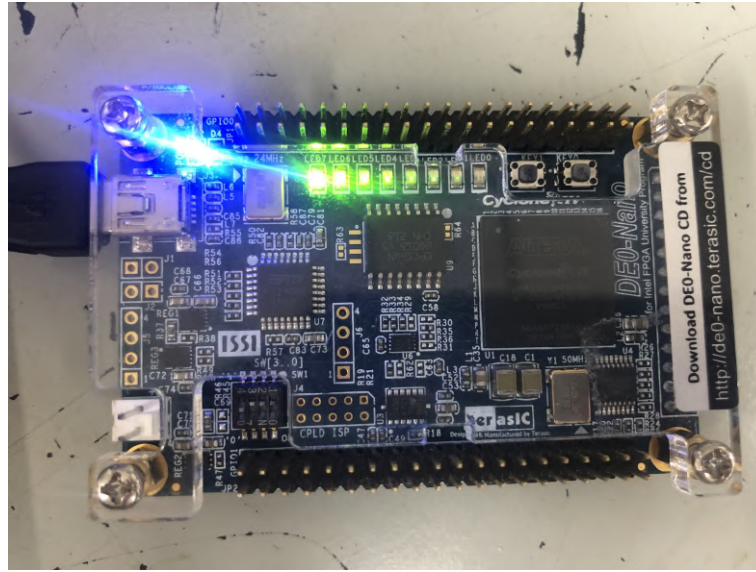
LED	DutyCycle	entrada para $f_{PWM}=5\text{kHz}$
LED [7]	100%	5000
LED [6]	33.8%	1688
LED [5]	11.4%	570
LED [4]	3.8%	192
LED [3]	1.3%	65
LED [2]	0.4%	22
LED [1]	0.1%	7
LED [0]	0%	0

Fonte: Sant'ana (2018a)[3].

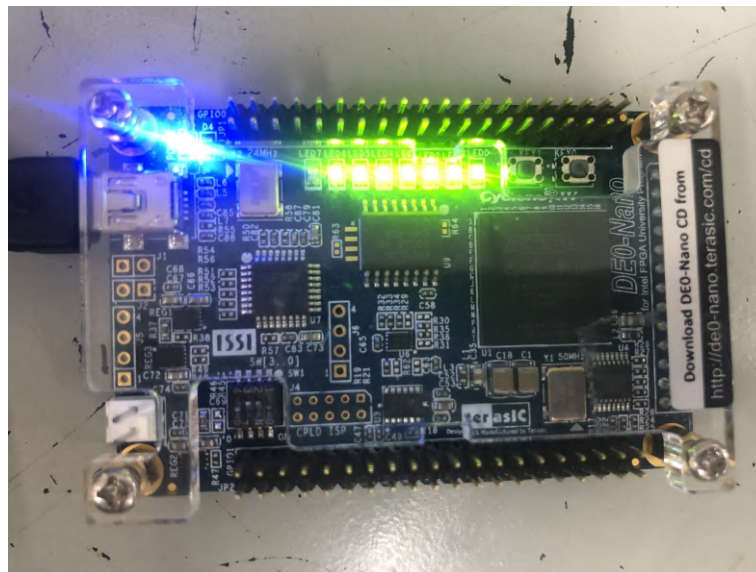
Ao mudar a chave *SW* [0] para nível baixo, pode-se perceber que as luminosidades se invertem, contudo, devido a escala logarítmica para os duty-cycles, não se percebe muita diferença de luminosidade (todos os diodos vão parecer estarem acesos por igual - com exceção de LED [7] que vai estar apagado).

4.3.2 Testes do bloco Bipolar

A seguir será realizado um exemplo de aplicação no microcontrolador, como sugerido em Sant'ana (2018c)[7]. Das quatro chaves deslizantes, uma irá controlar o *reset*, outra a polaridade do PWM e as outras duas irão controlar o DutyCycle de acordo com a Tabela 5. Os dois Push-Buttons controlam as entradas de erro.

Figura 21 – Execução no SoC FPGA, com $SW[0]$ em nível alto

Fonte: Autoria própria.

Figura 22 – Execução no SoC FPGA, com $SW[0]$ em nível baixo

Fonte: Autoria própria.

A Figura 23 mostra os pulsos sendo gerados para $SW(1:0) = 11$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um duty-cycle de 90%.

Da Figura 24, se percebe que os cursores indicam que o período da onda é $200\mu s$ - portanto, frequência 5kHz, conforme projetado.

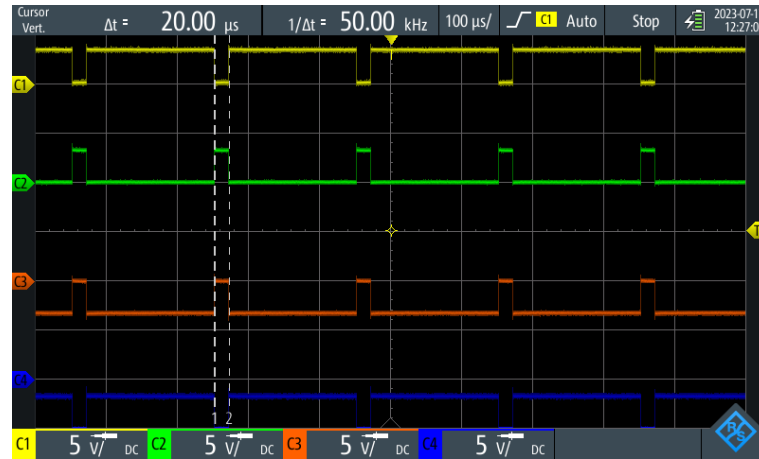
A Figura 25 mostra os pulsos sendo gerados para $SW(1:0) = 10$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um duty-cycle de 50%.

A Figura 26 mostra os pulsos sendo gerados para $SW(1:0) = 00$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um duty-cycle de 10%.

Tabela 5 – Variação dos Duty-Cycles em relação às chaves SW [1] e SW [0]

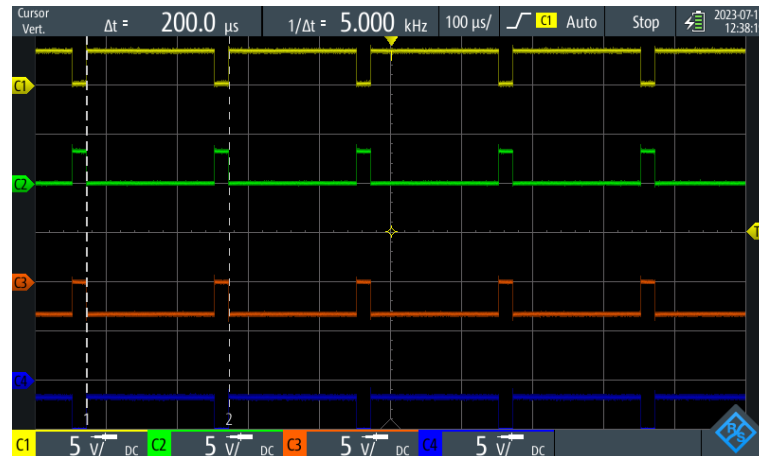
Duty-cycle	SW [1]	SW [2]
10%	0	0
30%	0	1
50%	1	0
90%	1	1

Fonte: Sant'ana (2018c)[7].

Figura 23 – Pulsos de gate para $SW(1:0) = 11$ (90% de duty-cycle)

Fonte: Autoria própria.

Figura 24 – Cursores indicando período 200μs - portanto, frequência de 5kHz

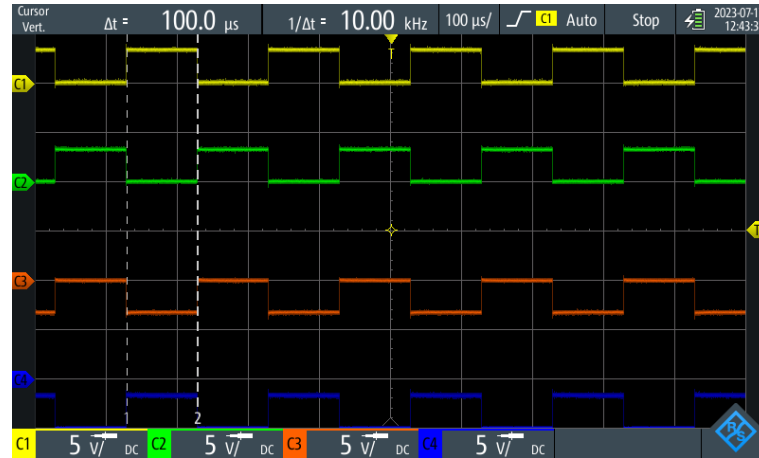


Fonte: Autoria própria.

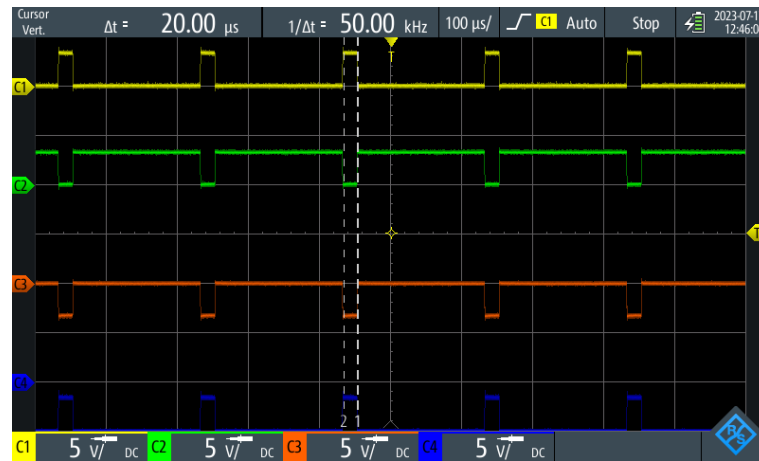
A Figura 27 mostra os pulsos sendo gerados para $SW(1:0) = 01$ com polaridade positiva (SW [2] = 1) - resultando, portanto, em um duty-cycle de 30%.

A Figura 28 mostra os pulsos sendo gerados para $SW(1:0) = 01$ com polaridade negativa (SW [2] = 0) - resultando, portanto, em um duty-cycle complementar de 70%.

A Figura 29 apresenta a situação em que a chave SW [3] (ligada ao pino *reset* do bloco Bipolar) contém nível alto. Isto faz com que os pulsos sejam cortados.

Figura 25 – Pulsos de gate para $SW(1:0) = 10$ (50% de duty-cycle)

Fonte: Autoria própria.

Figura 26 – Pulsos de gate para $SW(1:0) = 00$ (10% de duty-cycle)

Fonte: Autoria própria.

4.3.3 Testes do bloco Unipolar

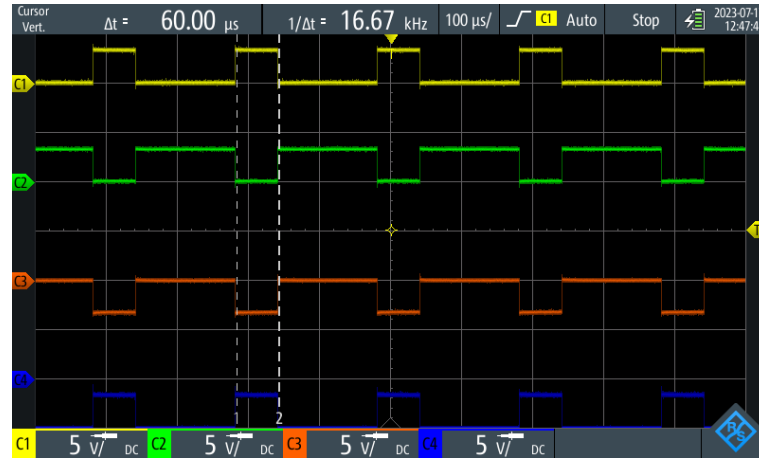
A seguir será realizado um exemplo de aplicação no microcontrolador, conforme sugerido em Sant'ana (2018d)[8]. Das quatro chaves deslizantes, uma irá controlar o *reset*, outra a polaridade do *PWM* e as outras duas irão controlar o *DutyCycle* de acordo com a Tabela 6. Os dois Push-Buttons controlam as entradas de erro.

Tabela 6 – Variação dos Duty-Cycles em relação às chaves $SW[1]$ e $SW[0]$

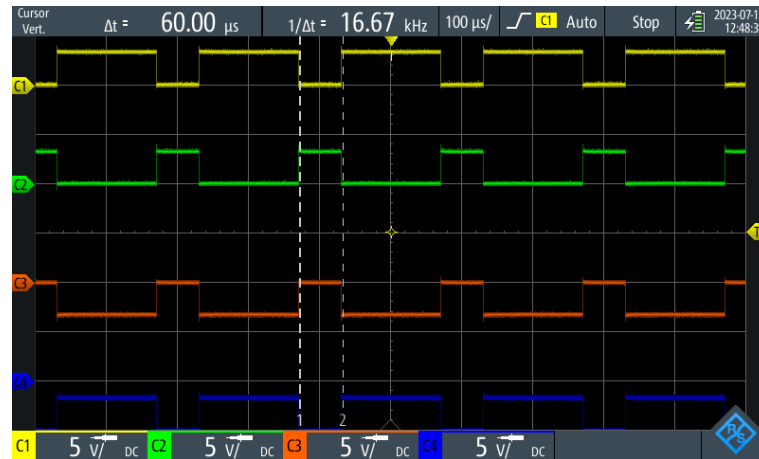
Duty-cycle	$SW[1]$	$SW[2]$
10%	0	0
30%	0	1
50%	1	0
90%	1	1

Fonte: Sant'ana (2018d)[8].

A Figura 30 mostra os pulsos sendo gerados para $SW(1:0) = 11$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um duty-cycle de 90%.

Figura 27 – Pulsos de gate para $SW(1:0) = 01$ (30% de duty-cycle)

Fonte: Autoria própria.

Figura 28 – Pulsos de gate para $SW(1:0) = 01$ e polaridade negativa (70% de duty-cycle)

Fonte: Autoria própria.

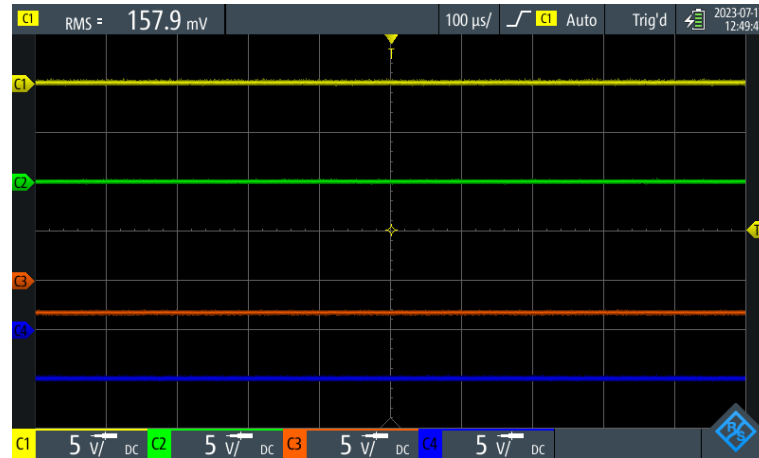
Da Figura 31, se percebe que os cursores indicam que o período da onda é $200\mu s$ - portanto, frequência 5kHz, conforme projetado.

A Figura 32 mostra os pulsos sendo gerados para $SW(1:0) = 10$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um duty-cycle de 50%.

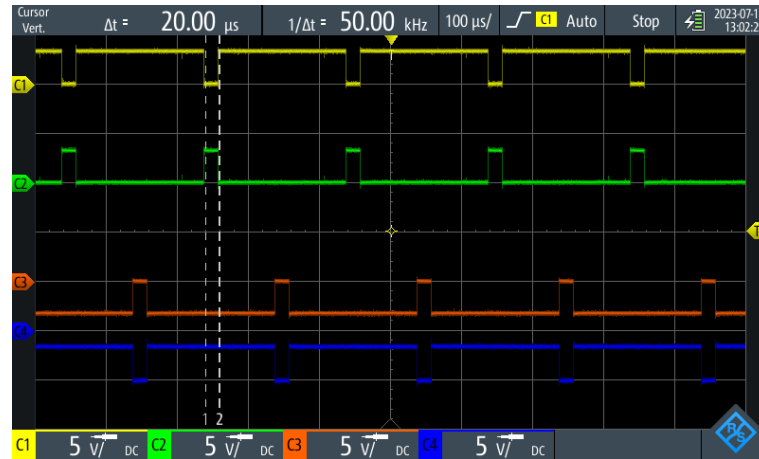
A Figura 33 mostra os pulsos sendo gerados para $SW(1:0) = 00$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um duty-cycle de 10%.

A Figura 34 mostra os pulsos sendo gerados para $SW(1:0) = 01$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um duty-cycle de 30%.

A Figura 35 mostra os pulsos sendo gerados para $SW(1:0) = 01$ com polaridade negativa ($SW[2] = 0$) - resultando, portanto, em um duty-cycle complementar de 70%.

Figura 29 – Pulsos de gate cortados por operação na chave *reset*

Fonte: Autoria própria.

Figura 30 – Pulsos de gate para $SW(1:0) = 11$ (90% de duty-cycle)

Fonte: Autoria própria.

4.3.4 Testes do bloco TresConversores

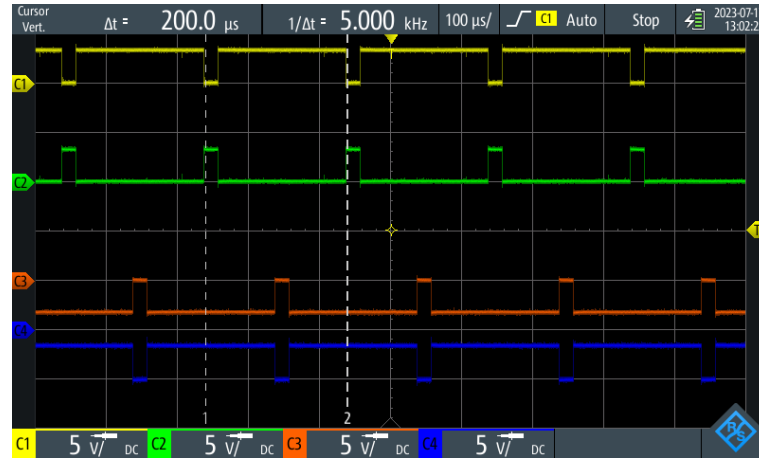
A seguir será realizado um exemplo de aplicação no SoC FPGA, conforme sugerido pelo roteiro em Sant'ana (2018b)[6]. Das quatro chaves deslizantes, uma irá controlar o *reset*, outra a polaridade do PWM e as outras duas irão controlar o *DutyCycle* de acordo com a Tabela 7. Os dois Push-Buttons controlam as entradas de erro.

Tabela 7 – Variação dos Duty-Cycles em relação às chaves $SW[1]$ e $SW[0]$

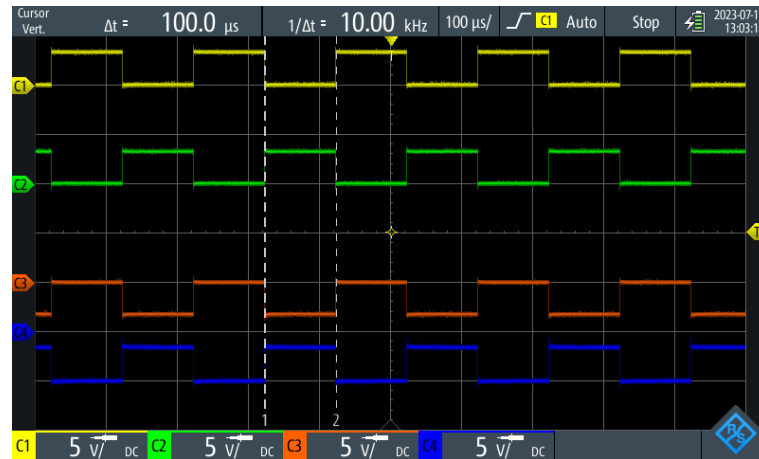
Duty-cycle	$SW[1]$	$SW[2]$
10%	0	0
30%	0	1
50%	1	0
90%	1	1

Fonte: Sant'ana (2018b)[6].

Têm-se nas Figuras de 36 a 38 o resultado da execução do projeto no SoC FPGA para $SW(1:0) = 11$ com polaridade positiva ($SW[2] = 1$) - resultando, portanto, em um

Figura 31 – Cursores indicando período 200 μ s - portanto, frequência de 5kHz

Fonte: Autoria própria.

Figura 32 – Pulsos de gate para $SW(1:0) = 10$ (50% de duty-cycle)

Fonte: Autoria própria.

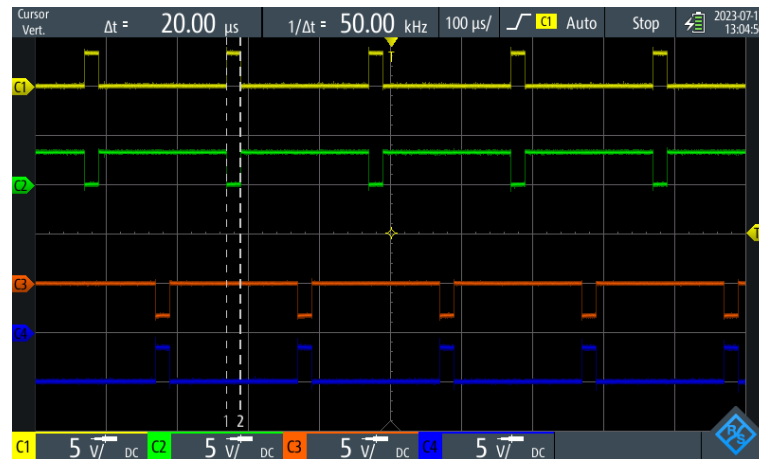
duty-cycle de 90%.

A Figura 36 mostra os pulsos referentes à ponte $H1$. A Figura 37 mostra os pulsos referentes à ponte $H2$. A Figura 38 mostra os pulsos referentes à ponte $H3$. Nota-se que os pulsos das três pontes seguem o padrão unipolar.

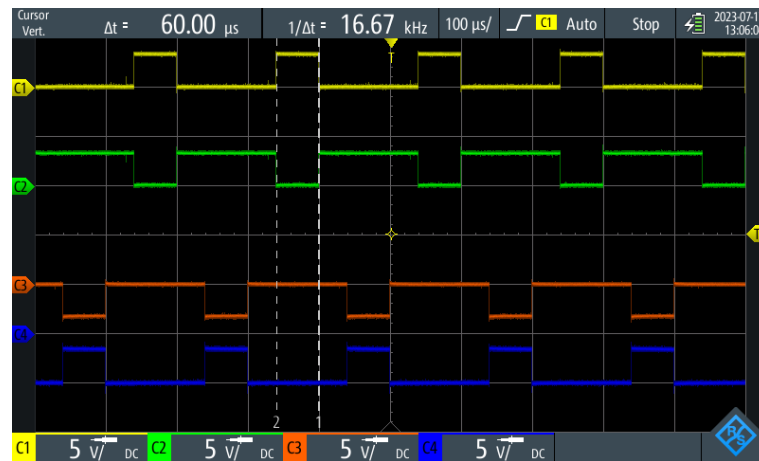
Na Figura 48 os cursores estão posicionados no edge de descida de $H1T1$ e no edge de descida de $H2T1$. Observa-se uma diferença de 34 μ s e o período vale 200 μ s. Considerando que o ajuste dos cursores do osciloscópio não é fino o suficiente, entende-se que a defasagem de 60° foi obtida.

Na Figura 40 os cursores estão posicionados no edge de descida de $H2T1$ e no edge de descida de $H3T1$. Observa-se uma diferença de 33 μ s e o período vale 200 μ s. Considerando que o ajuste dos cursores do osciloscópio não é fino o suficiente, entende-se que a defasagem de 60° foi obtida.

Na Figura 41 os cursores estão posicionados no edge de descida de $H3T1$ e no

Figura 33 – Pulsos de gate para $SW(1:0) = 00$ (10% de duty-cycle)

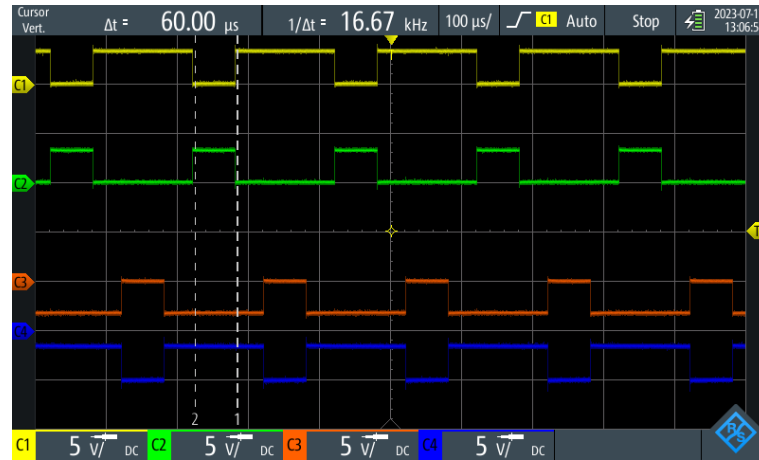
Fonte: Autoria própria.

Figura 34 – Pulsos de gate para $SW(1:0) = 01$ (30% de duty-cycle)

Fonte: Autoria própria.

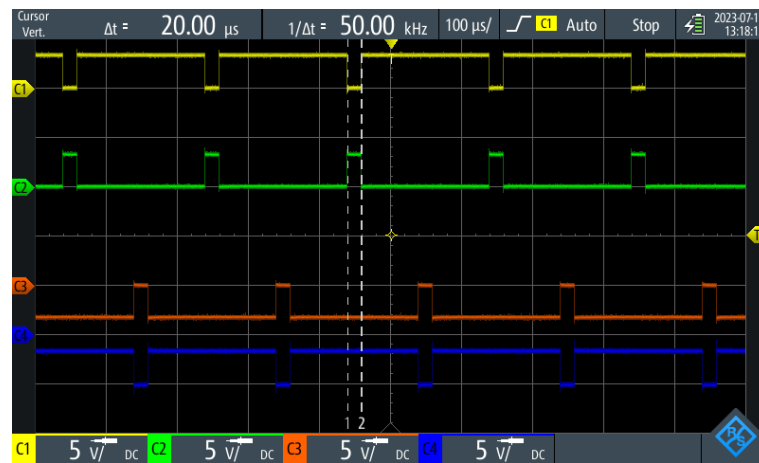
edge de descida de $H1T2$. Observa-se uma diferença de $33\mu s$ e o período vale $200\mu s$. Considerando que o ajuste dos cursores do osciloscópio não é fino o suficiente, entende-se que a defasagem de 60° foi obtida.

Figura 35 – Pulsos de gate para $SW(1:0) = 01$ e polaridade negativa (70% de duty-cycle)



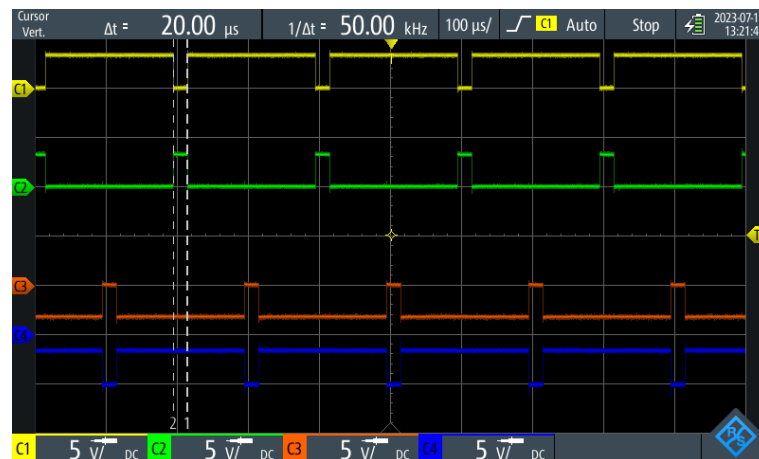
Fonte: Autoria própria.

Figura 36 – Pulsos da ponte $H1$ para $SW(1:0) = 11$ (90% de duty-cycle)



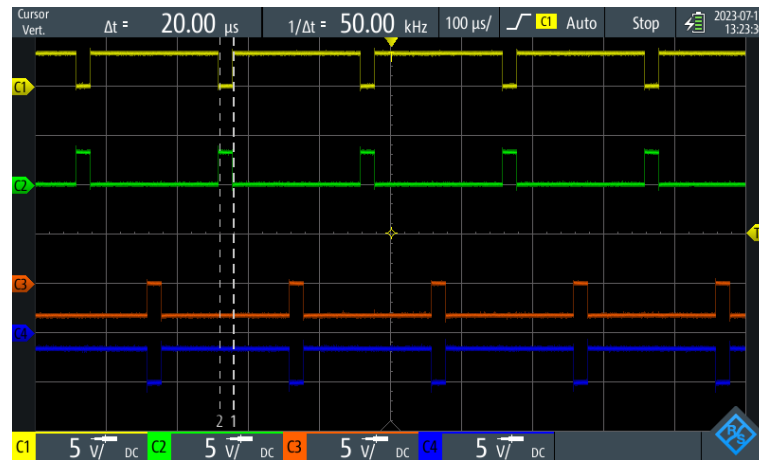
Fonte: Autoria própria.

Figura 37 – Pulsos da ponte $H2$ para $SW(1:0) = 11$ (90% de duty-cycle)



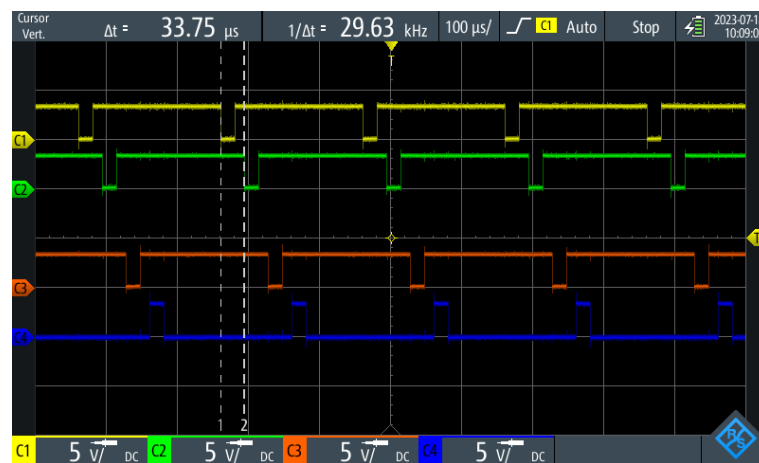
Fonte: Autoria própria.

Figura 38 – Pulsos da ponte $H3$ para $SW(1:0) = 11$ (90% de duty-cycle)



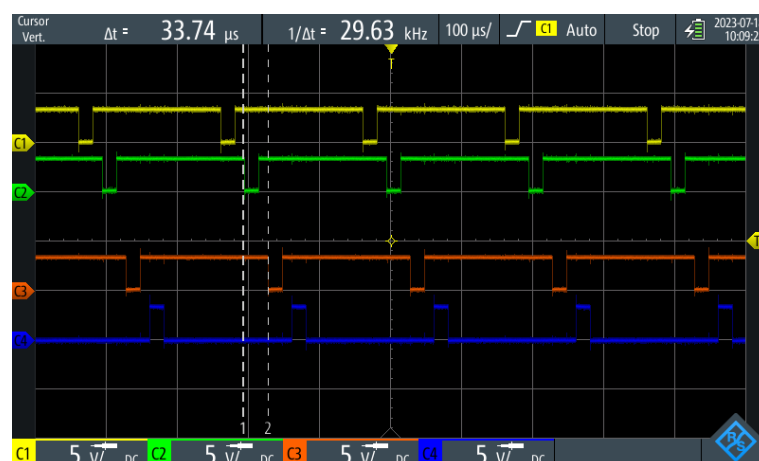
Fonte: Autoria própria.

Figura 39 – Zoom para verificação da defasagem - cursores entre $H1T1$ e $H2T1$

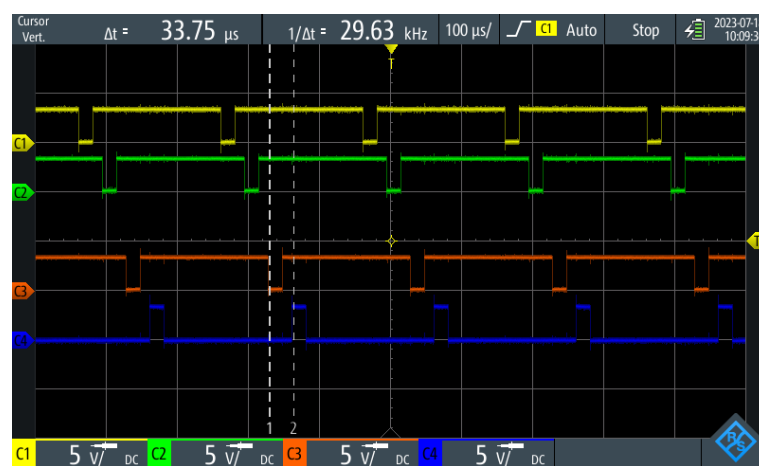


Fonte: Autoria própria.

Figura 40 – Zoom para verificação da defasagem - cursores entre $H2T1$ e $H3T1$



Fonte: Autoria própria.

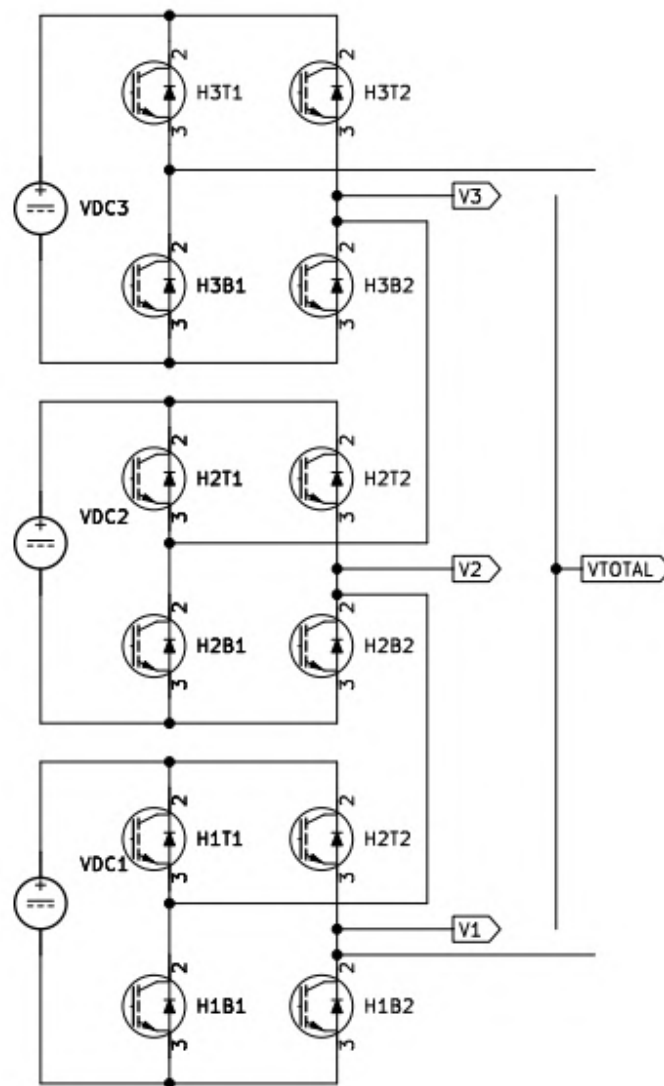
Figura 41 – Zoom para verificação da defasagem - cursores entre $H3T1$ e $H1T2$ 

Fonte: Autoria própria.

5 Resultados experimentais

De acordo com Sant'ana (2018f)[9], para a implementação do modulador PWM no FPGA, foi utilizada uma metodologia “bottom-up”, onde se inicia pelas funcionalidades mais simples - constituindo pequenos blocos para serem utilizados em blocos maiores e assim sucessivamente. Estes blocos são agrupados formando um bloco PWM. Utilizando dois blocos PWM, com defasagens de 180° entre as portadoras, tem-se o bloco Unipolar. Utilizando três blocos Unipolar, com defasagens de 60° entre cada portadora, obtém-se o PWM para três pontes em cascata, conforme exhibe o esquemático da figura 42.

Figura 42 – Três pontes H em cascata para sete níveis de tensão desenhada no software Kicad 6.0.

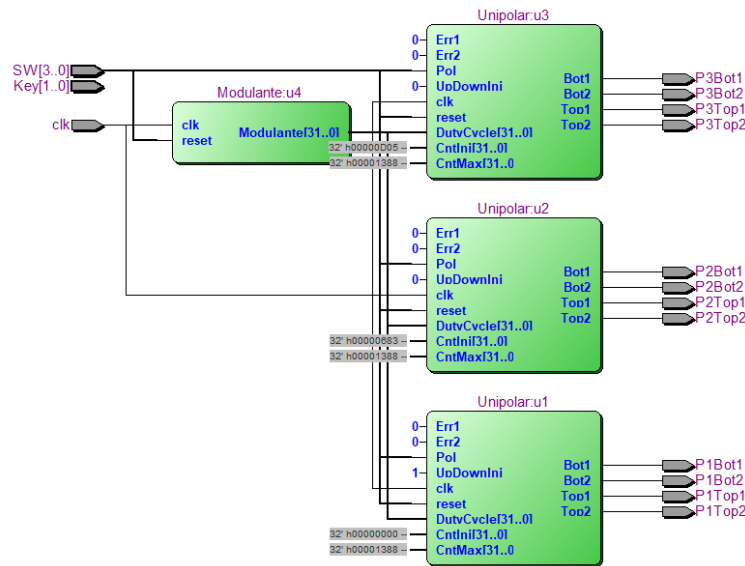


Fonte: Autoria própria.

5.1 Utilização de core DDS para geração de referências senoidais

Como sugerido no artigo em Sant'ana(2018g)[10], é apresentado o uso de um core *DDS* para a geração de um sinal senoidal, que servirá de referência interna para um teste do core *PWM* multiníveis. As instâncias do bloco Unipolar foram feitas respeitando os atrasos de 60° em cada portadora. A diferença consiste na entrada modulante dos blocos unipolar - aqui as entradas destes blocos são todas ligadas ao sinal *Senoide*. Este sinal, por sua vez, é ligado à saída do bloco Modulante.

Figura 43 – Circuito *RTL* interno ao bloco *DDS 3conv.vhd*.



Fonte: Autoria própria.

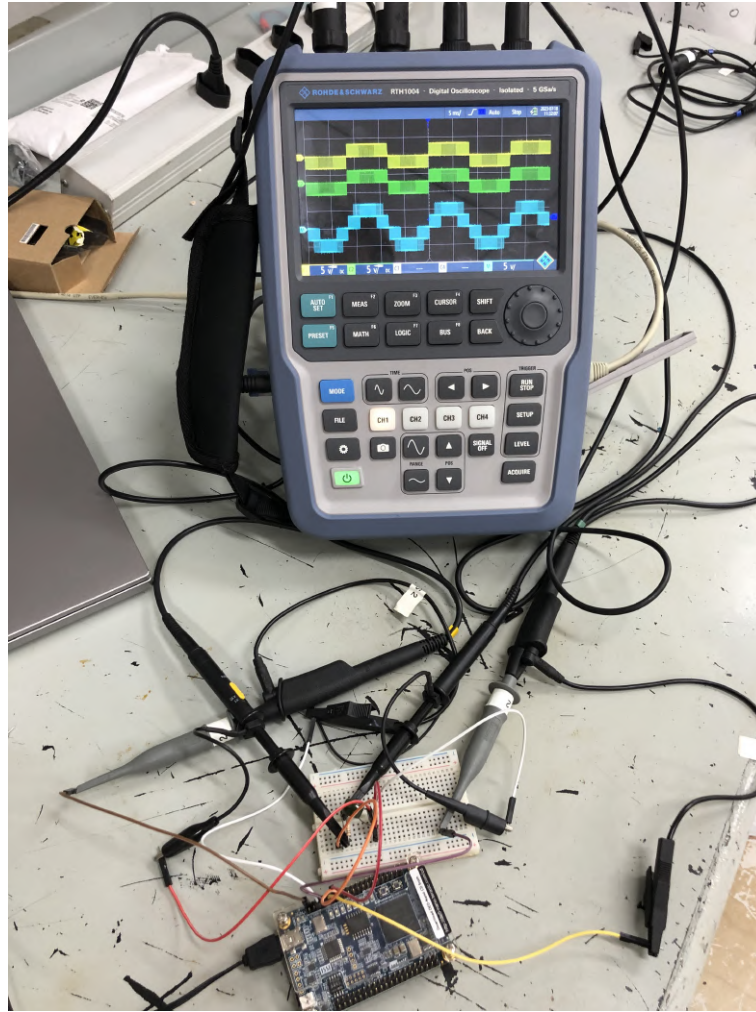
5.2 Testes em bancada com conversor multiníveis a IGBTs

O teste final, realacionado a síntese de sinais de controle para três pontes H resultando em sete níveis de tensão foi baseado no artigo de Sant'ana (2018e)[2]. Cabe ressaltar que não foi gravado na memória FLASH como sugeriu o artigo, logo, foi gravado apenas na memória SDRAM.

Tem-se na Figura 45 as tensões em duas pontes e a tensão total. A onda em amarelo é a tensão *AC* na saída da ponte *H1*. A onda em verde é a tensão *AC* na saída da ponte *H2*. A onda em azul é a tensão *AC* resultante da soma das tensões das duas pontes. Nota-se que a tensão de saída de cada ponte *H* possui um formato de chaveamento unipolar (chaveando entre $+V_{DC}$ e $0V$ e entre $-V_{DC}$ e $0V$), o que está de acordo com o que foi apresentado no gráfico em 3.3.

Vale destacar que só foi possível somar dois canais utilizando este osciloscópio. Assim, a onda em azul, que é a soma dos dois canais, é equivalente à saída de um conversor

Figura 44 – Fotografia da bancada de testes



Fonte: Autoria própria.

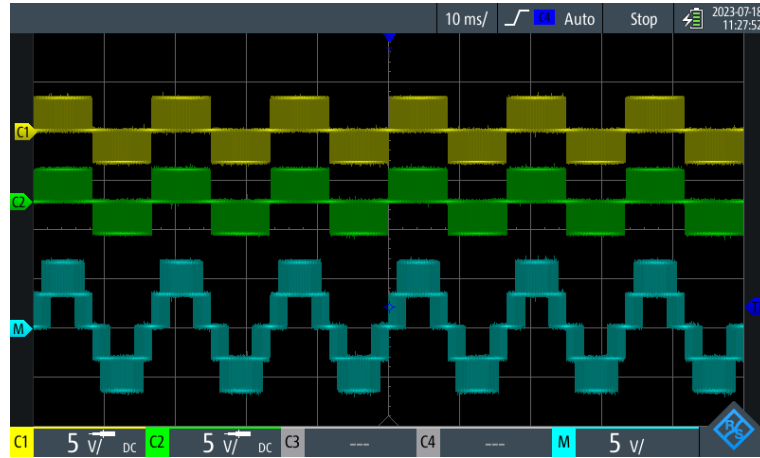
com duas pontes. Apesar de estar gerando os sinais pra três pontes, não foi possível exibir a saída completa. Portanto, a saída possui cinco níveis de tensão intermediários.

O deslocamento nos edges de cada pulso resulta no formato “em escada” da tensão total. Como os edges não ocorrem simultaneamente, a tensão resultante deveria apresentar uma frequência de transição de estados muito maior do que as frequências de chaveamento em cada IGBT.

A frequência resultante do sinal na saída de cada ponte H é o dobro da frequência de chaveamento em cada IGBT (f_{PWM}). E a frequência resultante na saída total do conversor é a frequência de cada ponte unipolar multiplicada pelo número de pontes (considerando que as defasagens estejam corretas) (Sant’ana, 2018e).

$$f_{unip} = 2 \cdot f_{PWM}[2] \quad (5.1)$$

Figura 45 – Tensões de saída(sinais em amarelo e verde) em duas pontes e soma das tensões em duas pontes(sinal azul)



Fonte: Autoria própria.

$$f_{resultante} = H \cdot f_{unip} = 2 \cdot H \cdot f_{PWM}[2] \quad (5.2)$$

Teoricamente, para $f_{PWM} = 5\text{kHz}$, cada ponte H apresentaria um chaveamento em $2 \cdot f_{PWM} = 10\text{kHz}$ e a saída total(três pontes) apresentaria um chaveamento em $6 \cdot f_{PWM} = 30\text{kHz}$ - mesmo que cada IGBT chaveie apenas em 5kHz - portanto com perdas reduzidas.

A figura 46 exibe o sinal de saída de uma das pontes com o PWM e também esse mesmo sinal filtrado. Cabe ressaltar que não foi possível observar no osciloscópio a frequência de chaveamento resultante para três pontes, assim, como foi medido a frequência em apenas uma ponte, foi obtido o 5kHz que cada IGBT chaveia. A Figura 47 apresenta o espectro de frequências da tensão de saída da ponte $H1$ (os espectros das pontes $H2$ e $H3$ são similares). Nesse sentido foi utilizado um filtro RC (*Filtro resistor-capacitor*) passa-baixa, com ele foi possível filtrar a alta frequência do PWM.

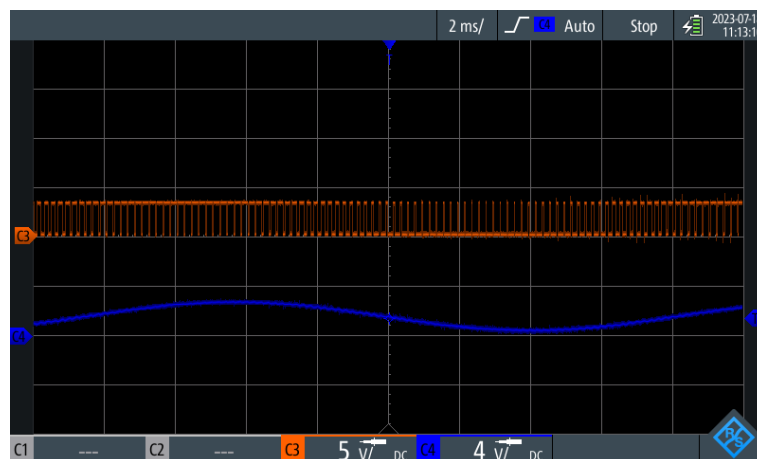
Um filtro RC passa-baixa é um circuito usado para atenuar sinais de alta frequência, permitindo que apenas os sinais de baixa frequência passem através dele. O filtro RC passa-baixa funciona de acordo com o princípio de que o capacitor age como um elemento reativo que permite a passagem de sinais de baixa frequência enquanto bloqueia os sinais de alta frequência. A frequência de corte (f_c) é determinada pelos valores do resistor (R) e do capacitor (C) de acordo com a fórmula:

$$f_c = \frac{1}{2 \cdot \pi \cdot RC} \quad (5.3)$$

Cabe ressaltar que foi utilizado o modo FFT (*Fast fourier transform*) do osciloscópio para ser possível observar o aspecto de frequências - A Transformada Rápida de Fourier (FFT) permite medir o espectro no domínio da frequência dos sinais adquiridos.

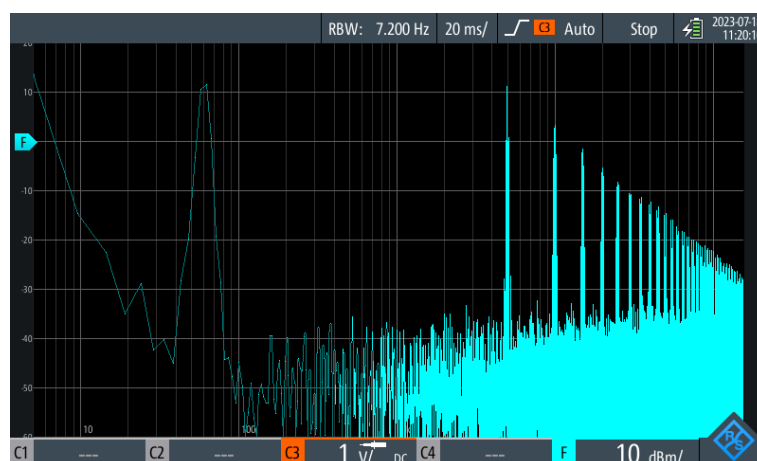
Isso fornece uma perspectiva diferente pois os sinais podem ser vistos como gráficos de amplitude ou fase versus frequência.

Figura 46 – Sinal de saída na ponte H1 e sinal de saída na ponte H1 filtrado



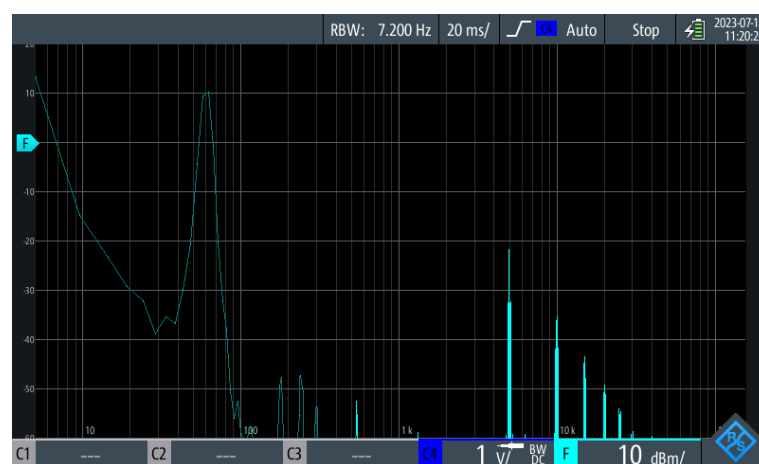
Fonte: Autoria própria.

Figura 47 – Espectro de frequências medido na ponte H1



Fonte: Autoria própria.

Figura 48 – Espectro de frequências da ponte H1 com filtro RC



Fonte: Autoria própria.

6 Considerações finais

Retomando a questão principal deste trabalho na área de eletrônica de potência, a busca baseia-se em dispositivos semicondutores capazes de conduzir altas correntes e aguentar tensões elevadas. Dessa maneira, realiza-se associações em série e em paralelo. Para garantir operação em níveis elevados de tensão ou corrente e melhorar a confiabilidade da associação série e paralela, uma das soluções é a síntese de uma ponte *H* composta por transistores IGBTs. Assim, neste trabalho, utilizou-se uma modulação PWM para três pontes *H* gerando sete níveis de tensão intermediários, operando de forma multinível.

Este trabalho abordou o desenvolvimento de um esquema de modulação PWM para conversores multiníveis em FPGA. Foi apresentada uma arquitetura em blocos, onde blocos menores quando agrupados ajudaram a construir o sistema, que foi a síntese de três pontes *H* em cascata gerando sete níveis de tensão. Apesar de não ter sido possível observar a tensão total das três pontes no osciloscópio, observou-se a soma das tensões em duas pontes que geraram cinco níveis de tensão. Nesse contexto, o sistema poderia ser adaptado para mais pontes, bastando apenas a repetição de blocos (e o cálculo apropriado).

Por fim, não foi possível verificar que a frequência de chaveamento resultante seria seis vezes maior (para o caso do conversor com três pontes) do que a frequência de cada IGBT, o que facilitaria a filtragem e manteria baixas perdas por chaveamento. Isso ocorreu pois apenas foi medida a frequência em um IGBT. Cabe ressaltar que o FPGA DE0-nano possui um número limitado de pinos de entrada e saída. Dessa maneira, seria interessante a realização deste trabalho em um dispositivo com mais pinos, como o microcontrolador Snickerdoodle, o qual possui 180 pinos de entrada e saída. Dessa maneira, todos eles poderiam ser utilizados na construção do PWM.

Pode-se concluir que este projeto de pesquisa realizado na Universidade Federal de Itajubá viabilizou o aprendizado, impulsionou o empirismo e conseguiu instigar estudos posteriores com o objetivo de promover a engenharia, inserindo-a no contexto atual.

Apêndices

APÊNDICE A – Assinatura do aluno e do orientador

Este relatório apresenta os resultados da Iniciação Científica ou Tecnológica intitulada “Plataforma baseada em DSP e FPGA para controle de conversores de potência”, desenvolvida no período de 01-10-2022 a 30-09-2023, sob a orientação do Prof. Robson Bauwelz Gonzatti, tendo em vista as orientações estipuladas pelo Edital PIVIC 2022-2023.

Itajubá, 17 de outubro de 2023:

Maira Clara

Aluno

Documento assinado digitalmente



ROBSON BAUWELZ GONZATTI

Data: 17/10/2023 15:35:32-0300

Verifique em <https://validar.iti.gov.br>

Orientador

Referências

- 1 BRAGA, H. A.; BARBI, I. Conversores estáticos multiníveis - uma revisão. *SBA Controle & Automação*, vol.11, n.1, p.20-28, Jan., Fev., Mar., Abril, 2000. Disponível em: <<https://www.sba.org.br/revista/vol11/v11a262.pdf>>. Acesso em: 04 set. 2023. 11
- 2 SANT'ANA, W. C. "2.6 - Gravação de projeto na memória FLASH e testes em bancada com conversor multiníveis a IGBTs". [S.l.], junho 2018e. Disponível em <<http://doingtechmyself.blogspot.com/2018/06/flash-bancada.html>>. 11, 40, 41, 42
- 3 SANT'ANA, W. C. "2.1 - Desenvolvimento de core PWM por comparação com portadora triangular". [S.l.], abril 2018a. Disponível em: <<http://doingtechmyself.blogspot.com/2018/04/desenvolvimento-de-core-pwm.html>>. 13, 20, 21, 27, 28
- 4 WU, B.; NARIMANI, M. *High-power converters and AC drivers*. 2. ed. Hoboken, New Jersey: John Wiley and Sons, Inc, 2017. 14, 15
- 5 Terasic. *DE0-Nano-SoC User Manual*. [S.l.], 2019. Disponível em: <https://www.terasic.com.tw/attachment/archive/941/DE0-Nano-SoC_User_manual_rev.D0.pdf>. Acesso em: 05 set. 2023. 16, 18
- 6 SANT'ANA, W. C. "2.4 - PWM de sete níveis para três pontes H em cascata". [S.l.], abril 2018b. Disponível em: <<http://doingtechmyself.blogspot.com/2018/04/pwm-sete-niveis-ponte-h-cascata.html>>. 23, 33
- 7 SANT'ANA, W. C. "2.2 - PWM de dois níveis (bipolar) para uma ponte H". [S.l.], abril 2018c. Disponível em: <<http://doingtechmyself.blogspot.com/2018/04/pwm-dois-niveis-bipolar.html>>. 28, 30
- 8 SANT'ANA, W. C. "2.3 - PWM de três níveis (unipolar) para uma ponte H". [S.l.], abril 2018d. Disponível em <<http://doingtechmyself.blogspot.com/2018/04/pwm-tres-niveis-unipolar.html>>. 31
- 9 SANT'ANA, W. C. et al. Implementação em fpga de modulador pwm para conversores multiníveis. *CBA 22º Congresso Brasileiro de Automática, CBA 2018*, João Pessoa - PB, 2018f. Disponível em: <https://www.sba.org.br/open_journal_systems/index.php/cba/article/download/291/253>. Acesso em: 04 set. 2023. 39
- 10 SANT'ANA, W. C. "2.5 - Utilização de core DDS para geração de referências senoidais". [S.l.], maio 2018g. Disponível em: <<http://doingtechmyself.blogspot.com/2018/05/core-dds-ref-senoidal.html>>. 40