

IMPLEMENTAÇÃO DE UM MÓDULO PWM EM UMA PLATAFORMA DE HARDWARE RECONFIGURÁVEL

Adriano Bezerra Pessoa¹, Pedro Thiago Valério De Souza²

¹Departamento de Engenharias e Tecnologia – Universidade Federal Rural do Semi-Árido (UFERSA)

Caixa Postal 15.064 – 59.900.970 – Pau dos Ferros – RN – Brasil

adrianobessoa@gmail.com pedro.souza@ufersa.edu.br

Abstract. *The use of digital applications is highly widespread in engineering. In this work, one of these applications, known as PWM (Pulse Width Modulation), was studied, where its concepts and characteristics were seen, as well as a method for generating signals from digital hardware. The PWM circuit was implemented using a reconfigurable hardware platform and for this purpose, the description language Verilog was used in a CPLD (Complex Programmable Logic Device) MAX III0 EPM240 by Altera. Thus, experimental results were obtained that indicate that the proposed method is efficient in generation of PWM signals, possessing high precision in the control of the work cycle and frequency of the generated PWM signals presenting results equal or very close with an error of less than 2% to those provided by equations.*

Resumo. *A utilização de aplicações digitais é altamente difundida na engenharia. Neste trabalho foi estudado um desses hardwares, conhecido por PWM (Modulação por largura de pulso - Pulse Width Modulation), onde foi visto seus conceitos e características além de método para geração de sinais a partir de um hardware digital. O circuito PWM foi implementado usando uma plataforma de hardware reconfigurável e para isto utilizado a linguagem de descrição Verilog em um CPLD (Complex Programmable Logic Device) MAX III0 EPM240 da Altera. Assim, foi obtido resultados experimentais que indicam que o método proposto é eficiente na geração de sinais de PWM, possuindo alta precisão no controle do ciclo de trabalho e frequência dos sinais PWM gerados apresentando resultados iguais ou muito próximos com erro de menos de 2% com relação aos valores teóricos.*

1. Introdução

Existem inúmeras aplicações de eletrônica digital na engenharia. Uma delas se dá no controle de potências entregue a uma carga, onde se necessita variar ou apenas dosar esta tensão. Neste campo surge o PWM (Modulação por Largura de Pulso - *Pulse Width Modulation*). Conforme Barr (2001, p 103), o PWM é uma técnica poderosa para controlar circuitos analógicos com as saídas digitais de um microprocessador. O PWM é empregado em uma ampla variedade de aplicações, desde medições e comunicações até controle e conversão de energia.

Métodos analógicos ou digitais podem ser utilizados para fornecer uma saída PWM. Ronald et al., (2011) aponta dentre as vantagens das aplicações de sistemas digitais: facilidade em serem projetados, facilidade no armazenamento de informações, o sistema tem maior facilidade de manter a exatidão e precisão, as operações podem ser programadas, são menos afetados por ruídos e os chips digitais podem ser fabricados com mais dispositivos internos.

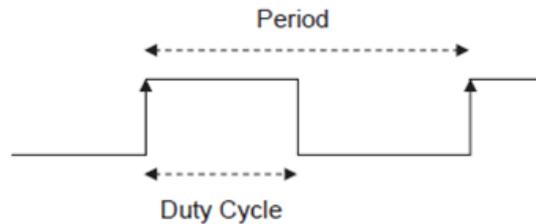
Diante disto, este trabalho tem como objetivo geral implementar um módulo de controle PWM em uma plataforma de *Hardware* reconfigurável utilizando a linguagem

de descrição de *Hardware Verilog* e analisar o comportamento do sistema implementado.

2. Conceitos sobre PWM

Segundo Ibrahim (2011, p. 89) uma saída PWM é basicamente uma forma de onda quadrada com um período e ciclo de trabalho especificados, ou seja, um ciclo de sinal PWM é dividido em um intervalo desligado e outro ligado, como mostra a Figura 01.

Figura 01: Sinal típico de um PWM.



Fonte: (IBRAHIM 2011, p. 89).

Conforme Pomilio (2014) a tensão média de saída depende da relação entre o intervalo em que a chave permanece fechada e o período de chaveamento. Assim, para determinarmos a tensão média temos:

$$V_{med} = V_{DD} \frac{T_{on}}{T_{PWM}} \quad (1)$$

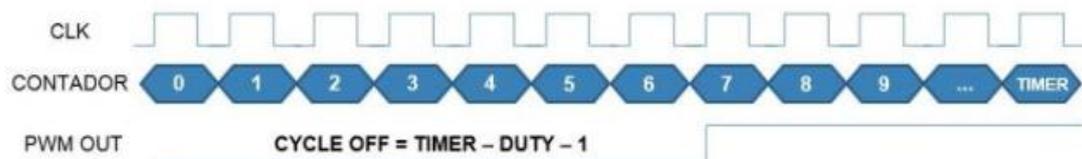
Em que que T_{on} é o tempo que o sinal PWM está em nível lógico alto e T_{PWM} é o período do PWM, V_{med} é a tensão média obtida e V_{DD} é tensão máxima fornecida.

Essa relação é chamada de Ciclo de trabalho. De acordo com Magalhães (2016), o Ciclo de trabalho pode ser determinado por

$$Duty\ Cycle = \frac{T_{on}}{T_{PWM}} \times 100\% \quad (2)$$

Para determinar os valores de *Duty Cycle* e Período deve-se pensar no sistema regido por um relógio e com isso, fazer uma análise discreta do PWM. Este relógio será o *clock* do próprio CPLD que será usado como parâmetro para quantizar o Período e *Cycle On* do PWM. Ao observar a Figura 02, imagina-se um valor de Período e *Duty* carregado no sistema. A saída permanecerá em zero até o contador igualar ao valor do *Cycle Off*, em seguida inverterá seu estado e ficará em alto até o contador completar o valor do Período (TIMER).

Figura 02: Clock e possível saída digital de PWM.



Fonte: MAGALHÃES, (2016).

Conforme Magalhães (2016) a frequência do PWM pode ser determinada pela fórmula abaixo:

$$f_{PWM} = \frac{f_{CLK}}{Período+1} \quad (3)$$

O valor do Período é somado mais 1, pois o Período começa em zero, logo conta-se mais 1 pulso do *clock*.

3. Materiais e Métodos

Das maneiras possíveis para se obter um circuito digital, Vahid (2009, p. 428) expõe que PLDs (Dispositivos Lógicos Programáveis - *Programmable Logic Devices*) não só permitem a obtenção de um projeto digital mais fácil, como também podem ser reprogramados, o que permite modificar o circuito a todo momento. Conforme Tocci et al (2011), os PLDs são subdivididos em três tipos diferentes:

- SPLD – *Simple Programmable Logic Devices*;
- CPLD – *Complex Programmable Logic Devices*;
- FPGA – *Field Programmable Gate Arrays*.

Como opção, para o desenvolvimento deste trabalho, foi utilizado um CPLD da Altera família Max II MAX II EPM240. Além das vantagens inerentes de um CPLD, esse em específico, de acordo com seu *datasheet* oferece alta contagem de entradas e saídas, desempenho rápido, armazenamento não volátil e baixo consumo de energia.

Devido disto, será necessária uma Linguagem de Descrição de *Hardware* ou HDL - *hardware description language*. Segundo Brown (2007, p. 65), uma linguagem de descrição de *hardware* é semelhante a uma linguagem de programação de computador, exceto que uma HDL é usada para descrever o *hardware* em vez de um programa que será executado em um computador. A linguagem utilizada para a implementação deste trabalho foi o Verilog, pois, como expõe Thomas e Moorby (2008 pag. 2), o Verilog proporciona ao projetista de sistema digital a flexibilidade de descrever um sistema digital em vários níveis de abstração e, ao mesmo tempo, fornece acesso às ferramentas de projeto auxiliadas por computador para auxiliar no processo de projeto nesses níveis. Além do mais, conforme Brown (2007. Pag. 65) esta é uma das linguagens padronizadas pela IEEE - *Institute of Electrical and Electronic Engineers*.

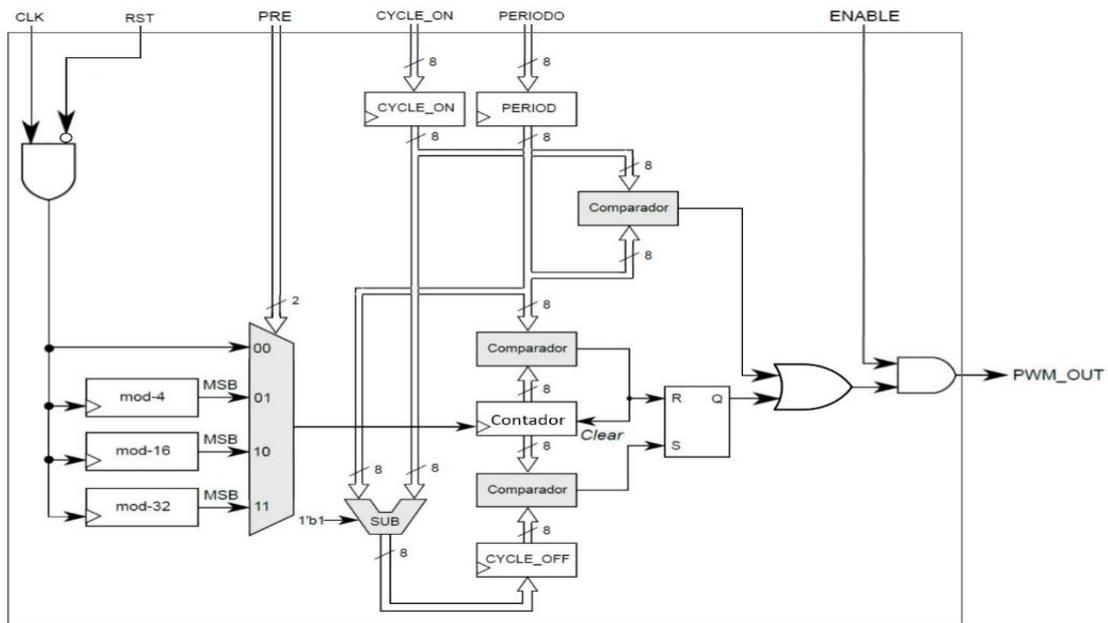
O ambiente de desenvolvimento será o Quartus®II da Altera. De acordo com Oliveira et al., o Quartus além de fornecer ambiente de desenvolvimento, permite a realização de todas as etapas envolvidas no projeto de um sistema digital, desde a descrição de sua lógica, por meio de diagramas esquemáticos ou linguagens de descrição até a simulação do circuito desenvolvido e gravação do projeto em um dispositivo lógico programável.

3.1 Circuito PWM

Com isso, para pensar em um circuito específico que com determinadas entradas forneça uma saída PWM, deve-se analisar o que já foi visto até então. O sinal permanecerá em nível lógico baixo e em seguida ficará em nível lógico alto até completar o valor do período do sinal PWM. O tempo em que a saída fica em zero é chamado de *Cycle Off*. O *Cycle Off*, portanto, é dado pela subtração entre o período e o *Cycle On*.

$$Cycle\ Off = Período - Cycle\ On \quad (4)$$

Para isso, um circuito que com determinadas entradas forneça uma saída PWM é mostrado na Figura 03.

Figura 03: Circuito PWM.

Fonte: Autor (2020).

O circuito recebe o valor do Período (PERIODO) e do Ciclo de Trabalho (CYCLE_ON) ambos de 8 bits que são armazenados em dois registradores. Recebe o Reset (RST) que quando está em valor alto bloqueia o *clock* e assim para o sistema, também recebe *Clock* (CLK) e ENABLE que serve para habilitar a saída. Ao analisar a parte principal do sistema, o contador que é incrementado pelo *clock* recebido envia os valores para os comparadores que irão comparar com os valores do Período e de *Cycle Off*, assim, quando o contador chega no mesmo valor que o *Cycle off*, o comparador inferior fornece valor de saída alto e seta o do *Latch SR*, que faz com que a saída do mesmo seja nível alto, e quando o valor do contador iguala ao valor do período, o comparador do meio vai a nível lógico alto e conseqüentemente Resetando o *Latch* e contador, onde irão a nível baixo e valor zero respectivamente. O comparador superior terá saída alta quando o valor de *Cycle On* e Período forem iguais. Este comparador com a porta *or* se fez necessário quanto se deseja ter um Ciclo de trabalho de 100%, visto que sem eles, neste caso que Período e *Cycle On* são iguais, o *Cycle Off* será zero. Assim, como o Período terá qualquer valor, ocasionará a saída ir a zero sempre que o contador chegar no mesmo valor que o período e ir a nível alto sempre que o contador chegar em zero. Com isso, a Saída PWM ficará em zero por um ciclo de *clock*. Com esta parte do circuito, este problema é solucionado.

Para melhorar a flexibilidade da frequência do PWM, foi implementado um circuito para dividir a frequência do *clock* do CPLD. Este circuito é denominado PRESCALER e dá a possibilidade de optar por quatro PRESCALER's 1, 4, 16 e 32, que são, respectivamente, a frequência do *clock* original, a frequência de *clock* dividida por 4, 16 e 32. O circuito tem uma lógica simples, onde, utiliza-se um Multiplexador – MUX 4x1 em que a entrada PRE de 2 bits que seleciona o *clock* (CLK) desejado na saída do MUX. Os blocos mod-4, mod-16 e mod-32 serão responsáveis pela divisão da frequência do *clock* por 4, 16 e 32 respectivamente. Como consequência da utilização do PRECALER, a frequência do PWM pode ser reescrita como segue:

$$f_{PWM} = \frac{f_{CLK}}{(Período+1)(PRESCALER)} \quad (5)$$

4. Trabalhos Relacionados

Em várias situações é possível encontrar aplicações de um sinal PWM. Uma delas é na área de robótica. Futida & Romero (2008) demonstram a utilização de um sistema PWM em um controlador PIC, com o objetivo de controlar a velocidade dos motores de um robô Very Small Size que é utilizado em times de futebol de robôs. Em um robô deste tipo, é necessário controlar a velocidade dos motores, possibilitando que o mesmo possa fazer curvas, tenha aceleração e possa variar sua velocidade de deslocamento. No sistema, os dois motores são controlados por saídas de PWM's diferentes, entretanto, os dois possuem a mesmas frequências. Para que os motores possam ter velocidades independentes, os Duty Cycles têm controles independentes, possibilitando curvas e manobras, aumentando o desempenho do robô em uma partida.

Outro trabalho em que encontramos a aplicação de PWM é no controle de potência para veículos elétricos e híbridos. Pereira (2012) expõe a utilização de um inversor PWM para o acionamento de um motor síncrono. Utilizando um chaveamento tipo PWM, é possível controlar a frequência e amplitude do sinal alternado de saída. Isso possibilita utilizar uma fonte CC e com esse tipo de chaveamento, controlar e alimentar motores síncronos.

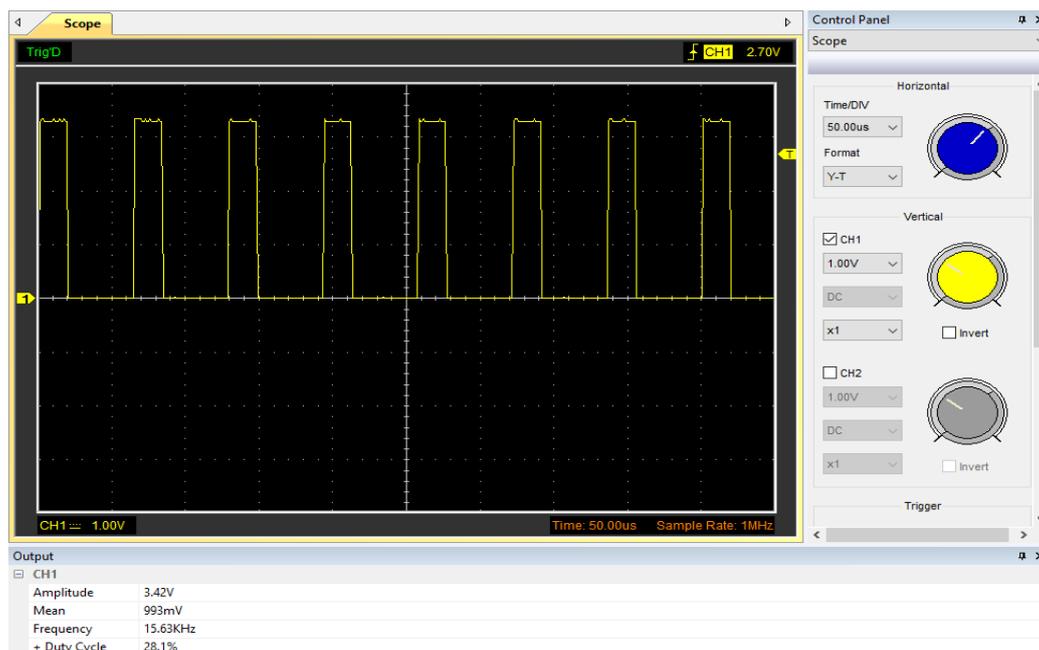
5. Resultados e Discursões

Para analisar a forma de onda de saída do circuito implementado no CPLD, utilizou-se um osciloscópio da Hantek modelo 6022BE que possui largura de banda de 20MHz e taxa de amostragem de 48MS/s.

4.1 Caso 01.

A primeira amostra tem como valores de entrada PERIODO = 200, CYCLE_ON = 60 e PRE = 2. Na Figura 04, é fácil visualizar que a saída irá se manter desligada a maior parte do tempo, o que reflete o comportamento esperado levando em consideração os valores de entrada.

Figura 04: Formato de onda para PERIODO = 200, CYCLE_ON = 60 e PRE = 2.

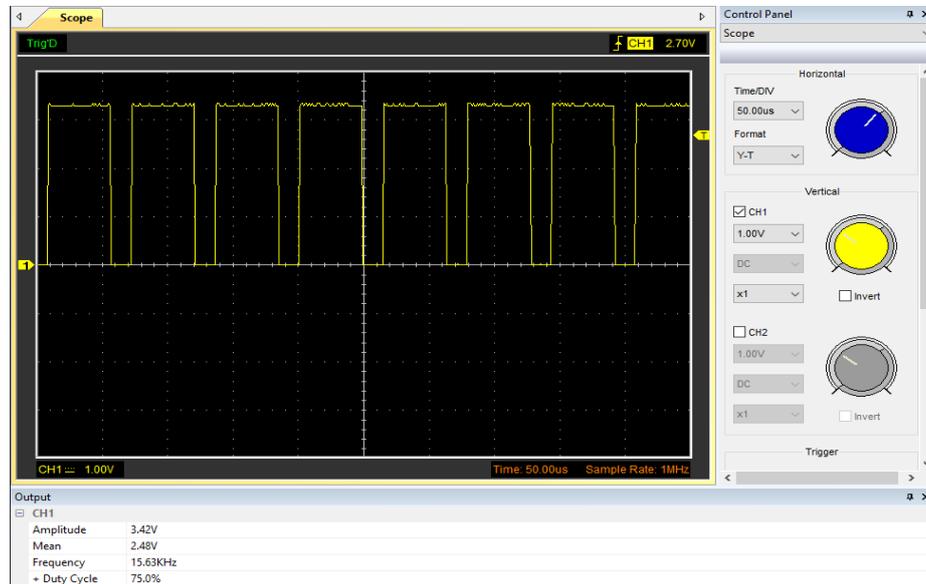


Fonte: Autor (2020).

4.2 Caso 02.

Na segunda amostra, os valores foram: PERIODO = 200, CYCLE_ON = 150 e PRE = 2. Observando a Figura 05, é fácil observar que diferentemente da Figura 04, a saída está em alto por mais tempo, mostrando coesão com os valores de entrada.

Figura 05: Formato de onda para PERIODO = 200, CYCLE_ON = 150 e PRE = 2.

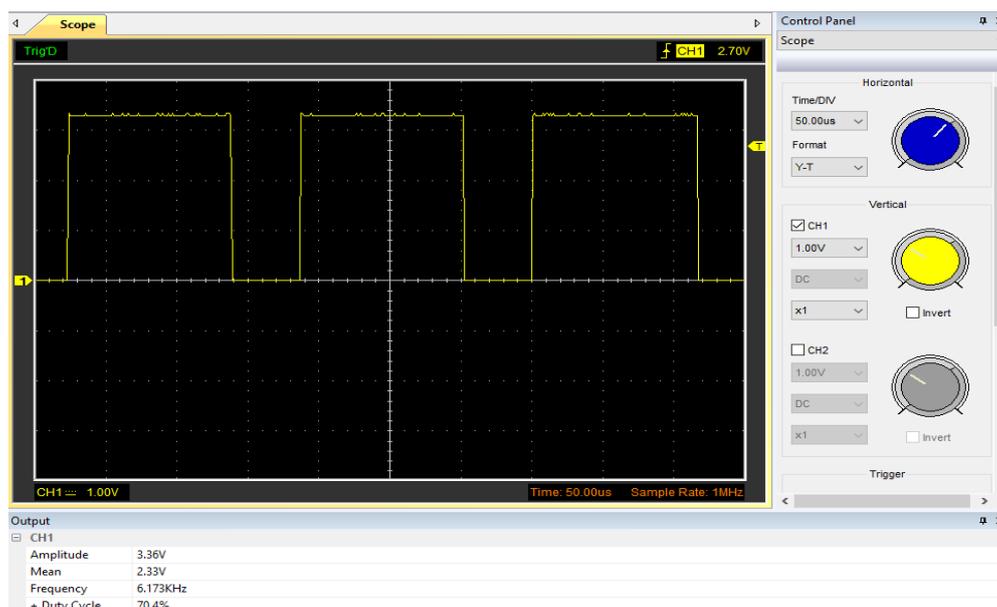


Fonte: Autor (2020).

4.3 Caso 03.

Na terceira amostra os valores de entrada foram com PERIODO = 255, CYCLE_ON = 180 e PRE = 3. Um dos pontos que devem ser observado nesta amostra, é que temos a menor frequência possível neste circuito. Isso se dá devido ter o valor máximo possível de período e o maior valor de PRESCALER.

Figura 06: Formato de onda para PERIODO = 255, CYCLE_ON = 180 e PRE = 3.

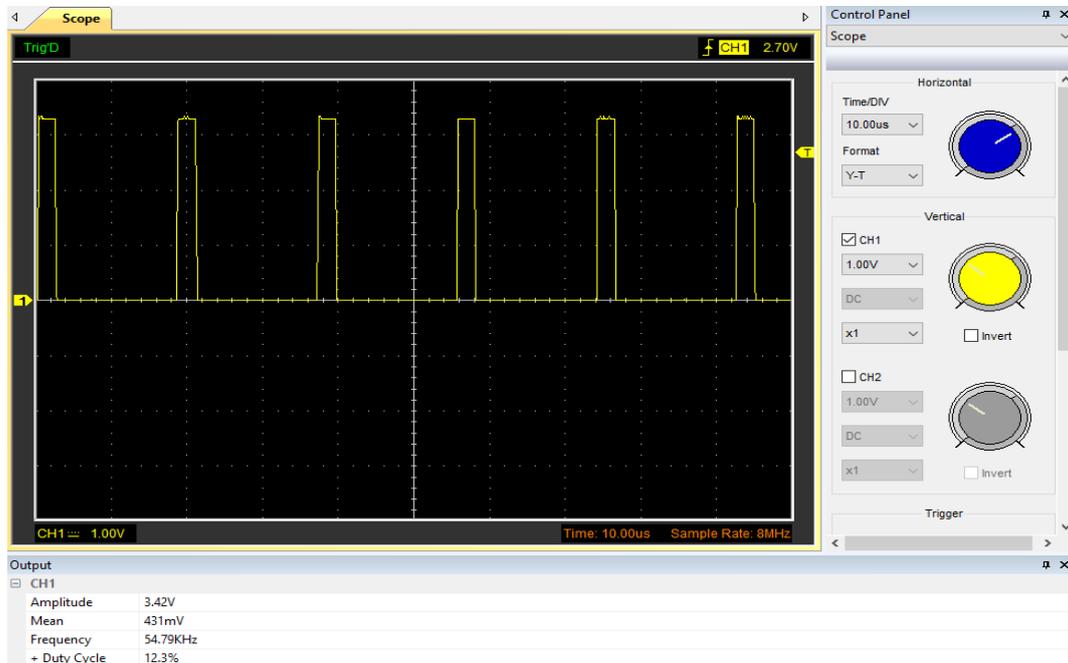


Fonte: Autor (2020).

4.4 Caso 04.

Para a quarta amostra, valores de entrada: PERIODO = 230, CYCLE_ON = 30 e PRE = 1. Nesta amostra é fácil ver que se tem um curto tempo de sinal alto, o que mais uma vez entra em conforme com as entradas fornecidas.

Figura 08: Formato de onda para PERIODO = 230, CYCLE_ON = 30 e PRE = 1.



Fonte: Autor (2020).

Para uma melhor comparação entre os valores práticos e teóricos fornecidos pelas equações de (1) a (5), a Tabela 01 mostra a as tensões médias de saída tanto dos resultados práticos como teóricos e ainda o valor de erro relativo.

Tabela 01: Valores de tensão média e erros relativos.

Amostra	Valor teórico (V)	Valor Prático (V)	Erro Relativo (%)
01	0,990	0,993	0,30
02	2,480	2,480	0,00
03	2,330	2,330	0,00
04	0,430	0,431	0,23

Fonte: Dados experimentais (2020)

Para a frequência.

Tabela 02: Valores de frequência e erros relativos.

Amostra	Valor teórico (KHz)	Valor Prático (KHz)	Erro Relativo (%)
01	15,55	15,63	0,51
02	15,55	15,63	0,51
03	6,10	6,17	1,13
04	54,11	54,79	1,24

Fonte: Dados experimentais (2020)

Ao analisar a Tabela 01, percebe-se que nos casos 02 e 03 os valores práticos e teóricos são iguais, enquanto nos casos 01 e 04 diferem por um valor de tensão menor que $5mV$ e erro menos que 0,5%. Na Tabela 02, os valores de frequência também ficaram muito próximos, apresentando a maior diferença de $0,68kHz$ e erro relativo máximo de 1,24%.

6. Conclusões

Diante dos valores obtidos experimentalmente, é notório que o sistema criado e métodos utilizados para implementação oferece ótimo desempenho, isto porque comparados aos valores teóricos apresentam erros menores que 2%. Além do mais, por utilizar software gratuito e que permite a simulação, tem-se uma melhor viabilidade na hora de criação, tanto no quesito financeiro como na análise do sistema em criação.

Outros fatores relevantes são mostrados na Tabela 01 e Tabela 02, onde os erros relativos de tensão na Tabela 01, ficam abaixo de 05% e em alguns casos fica em 0%. Na Tabela os valores de erro relativo ficaram abaixo dos 2%, um resultado satisfatório e nos leva a concluir que o circuito obtido e método utilizado é viável e oferece desempenho esperado.

Referências

BARR, Michael. Pulse Width Modulation. Embedded Systems Programming. Beginner's Corner. 2001. Disponível: <http://homepage.com.itesm.mx/carbajal/Microcontrollers/ASSIGNMENTS/readings/ARTICLES/barr01_pwm.pdf>. Acesso em: 02 de Jan. de 2020.

NILSSON J. W.; RIEDEL S. A. Circuitos elétricos. 8ª ed. São Paulo: Pearson, 2009.

IBRAHIM, Dogan. Advanced PIC microcontroller projects in C: from USB to RTOS with the PIC 18F Series. Newnes, 2011.

VAHID, Frank. Sistemas Digitais. Bookman Editora, 2009.

BROWN, Stephen D. Fundamentals of digital logic with Verilog design. Tata McGraw-Hill Education, 2007.

THOMAS, Donald, and Philip Moorby. The Verilog Hardware Description Language. Springer Science & Business Media, 2008.

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. Sistemas Digitais Sistemas e Aplicações. 11ª Ed. São Paulo. Pearson 2011.

MAGALHÃES, Gabriel V. Novaes. Uma Arquitetura PWM em VHDL. Embarcados. 2016. Disponível em: <<https://www.embarcados.com.br/uma-arquitetura-pwm-em-vhdl/>>. Acesso em: 18 de Jan. de 2020.

OLIVEIRA, Caio Augusto; AGUIAR, Jéssica Azevedo; FONTANINI, Mateus Galvão Said. Dispositivos Lógicos Programáveis. Universidade Estadual Paulista- UNESP. Disponível em: <<https://www.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/logica-programavel.pdf>> Acesso em: 21 de jan. De 2020. (Apostila).

FUTIDA, Ivo Takao; ROMERO, Roseli Aparecida Francelin. Desenvolvimento mecânico e de Controle PWM para sistema robótico. Disponível em: <

http://conteudo.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_RT_331.pdf >. Acesso em 12 de jan. de 2020.

PEREIRA, Windson Braga. Estudo Da Viabilidade do uso de Alternadores Como Motores em Veículos Elétricos. Universidade Federal do Rio de Janeiro. Rio de Janeiro - RJ. 2012. Disponível em: <https://pt.scribd.com/document/335941285/Estudo-Da-Viabilidade-Do-Uso-de-Alternadores-Como>. Acesso em 12 de jan. de 2020.