EVOLUTION IN ELECTRICAL AND ELECTRONIC ENGINEERING
e-ISSN: 2756-8458

EEEE

Vol. 5 No. 1 (2024) 115-122
https://publisher.uthm.edu.my/periodicals/index.php/eeee

FPGA Implementation of BLDC Motor Control

Adilah Maizan Mohd Zulkhairi[1], Mohamad Hairol Jabbar[1]*

[1] *Faculty of Electrical and Electronic Engineering,*
Universiti Tun Hussein Onn Malaysia, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: hairol@uthm.edu.my
DOI: https://doi.org/10.30880/eeee.2024.05.01.015

Article Info

Received: 14 January 2024
Accepted: 19 February 2024
Available online: 30 April 2024

Keywords

BLDC motor control, FPGA, Simulink MATLAB

Abstract

Permanent Magnet Brushless DC (BLDC) motors are widely used in industries like electric bicycles, industrial robots, and CNC welding machines. Due to the efficient electronic control system and compact design, they have become a key component in the automation industry. However, they face significant problems when obtaining accurate motor movement and performing speed control. This work concentrates on designing a BLDC motor control system and generating hardware architecture design using PWM and PID controller. The performance of the BLDC motor control in terms of hardware implementation and optimization effects on the target FPGA device was evaluated. This work used Simulink from MATLAB software for motor control circuit design and Vivado software for performance verification on the Zynq-7000 target devices. Hardware control system performance, which includes power consumption, clock speed, and utilization of HDL, is being evaluated in this work. Simulink MATLAB is also used to construct the entire control system process for optimization and generate the VHDL code using the HDL coder from the designated block diagram. Vivado software is used to verify Simulink's simulation and analyze the HDL optimization by implementing it on the target FPGA device. The results show the comparison between modeling simulation on Simulink and Vivado regarding the generated output waveform and the hardware architecture design performance after enabling various optimization options. Both simulation outcomes validate the effectiveness of utilizing Simulink MATLAB for circuit diagram design and HDL code generation. The evaluated performance of on-chip power consumption is enhanced by 47% and 90% faster than the non-optimization option regarding the clock speed. Worst Negative Slack (WNS) for pipelining and resource sharing is 95% better than the Baseline option.

1. Introduction

The brushless motor (BLDC) motor has been widely used in many variable speeds and applications of servo motors due to its attractive features of being simple to operate and easy to control [1]. It is commutated by integrating an inverter, which produces an AC signal from a DC power source to drive the motor electronically, resulting in significantly lower rotor losses [2]. Field Programmable Logic Arrays (FPGA) have recently been widely used in motor control applications. The advantage of using this controller is the flexibility in programming [3]. This controller works in the PWM technique, where it has become an advanced, cost-efficient digital controller and has been improved [4]. Next, Xilinx FPGA system generator is used rather than ModelSim as it can integrate

seamlessly with MATLAB/Simulink, hence producing more optimal performance [3]. This work focuses on model construction of BLDC motor using MATLAB/Simulink and FPGA control. The model construction motor control is constructed via MATLAB/Simulink, and the model is transferred into VHDL hardware language for hardware implementation on the target FPGA device. This development process is used as it reduces engineering debugging systems and workload in FPGA programming [5]. Welding Computer Numerical Control (CNC) machines require a precise, accurate, and tight tolerance range of machine movement. However, current welding CNC systems often need help to achieve the necessary precision in motor movement and encounter challenges in simultaneously controlling speed [6]. A study by Xuesong Wang et al.1 investigated how CNC machine tool feed systems are designed to match their dynamics. They discovered that the usual static design methods need to be revised to keep up with the need for both high speed and high precision. They recommended optimizing each part of the system to improve its overall performance without using too many resources [7]. Research work [8] showed how to use Field Oriented Control (FOC) on FPGA to control 3-phase motors with high switching frequency. They made a single design that works for both Brushless DC motors (BLDC) and Permanent Magnet Synchronous Motors (PMSM). According to the journal [9], it created a Fuzzy PID controller for a BLDC motor using FPGA technology. The tests they conducted showed that the Fuzzy PID controller performs better than the regular PID controller in controlling the motor. These studies offered helpful information about designing and setting up precise and accurate control systems for welding CNC machines.

2. Methodology

2.1 Block Diagram

A block diagram of the whole process, starting from the power supply as the input, is shown in Fig. 1. The purpose of the simulation using the design model block is to confirm the functioning design parameters of the system before transferring to VHDL hardware language.

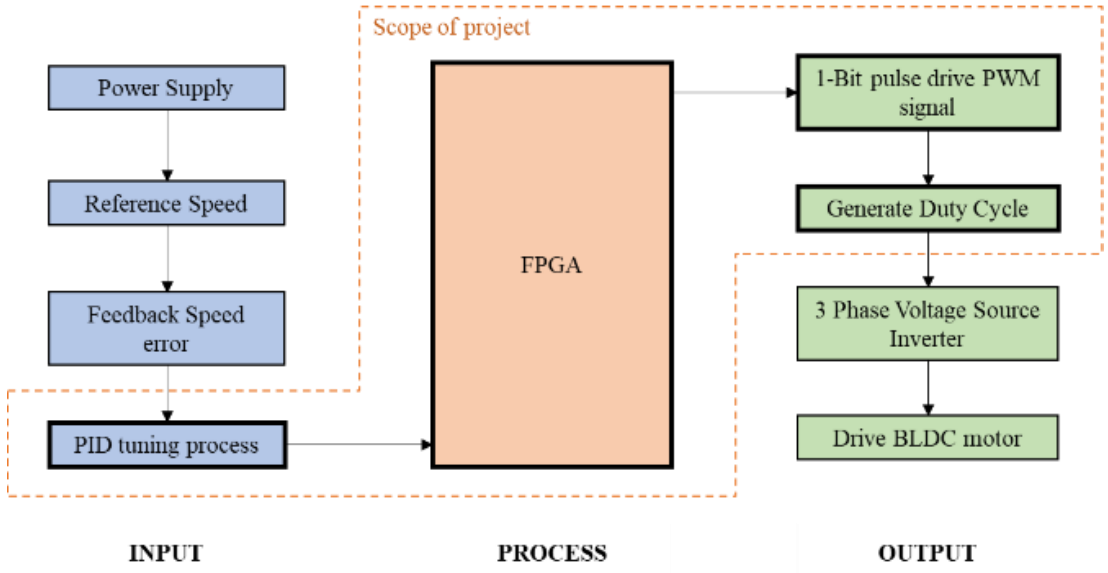

Fig. 1 *Block Diagram*

The simulation model used 100V DC voltage as a supply into the system, along with a reference speed that was set to 500 rpm. The speed setting in the simulation model is to reflect the operational requirement. Many CNC machines, especially those used in small-scale operations, have limits on how fast they can spin and how quickly they can move the cutting tool. The selected speed must fall within these limits [10]. Proportional Integral Derivative (PID) is used to achieve the desired performance by adjusting the parameters to optimize system performance from the feedback speed. Several sophisticated control algorithms were implemented throughout the process in the FPGA environment. First, the output from the PID controller is implemented in the FPGA interface to generate a PWM signal. Besides, Hall effect signals, which represent the instantaneous state of the magnetic field, generated from the BLDC motor are then channeled to the decoder by providing the rotor's speed. Hence, the decoder design will interpret the signals using decoder logic to determine which of the six electrical sectors the rotor is located in. Then, the identified rotor's position in the decoder logic will generate a voltage waveform and an electromotive force (EMF) for motor phases. To maintain the desired speed, Emf signals are compared to zero, which is the process that happens in Gates. Next, the Emf signal, which is used to determine the position of the motor and the duty cycle that indicates the desired speed, will be directed to the inverter. The FPGA

Evolution in Electrical and Electronic Engineering Vol. 5 No. 1 (2024) p. 115-122

controller will instruct these gates to regulate the transistors in accordance with the rotor position, as shown in the closed-loop control system [11].

2.2 Flowchart of Design and Simulation of Motor Control on Simulink

Fig. 2 shows the flowchart of the Simulink/MATLAB simulation for speed and precision of motor position. The model block was designed by following the proper operation of the BLDC motor.

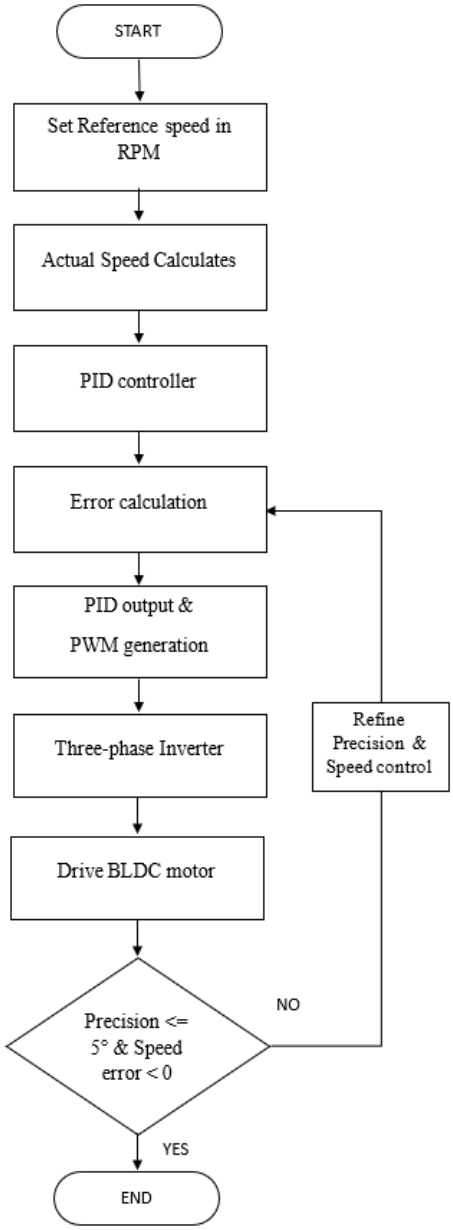

Fig. 2 *Design and simulation of motor control*

2.3 Simulink Design Model Simulation of Motor Control

Fig. 3 shows the diagram for motor control simulation using MATLAB Simulink, which includes the connection of DC Voltage source, Universal Bridge (inverter), Permanent Magnet Synchronous Machine in Trapezoidal Back EMF waveform setting, PID controller, Reference speed, PWM generator, Gates and Decoder.

2.4 Vivado Simulation and Implementation Flowchart

Fig. 4 shows the Vivado implementation and simulation step flow. The model block was designed starting from the modeling block model in Simulink until the performance evaluation on Vivado implementation.

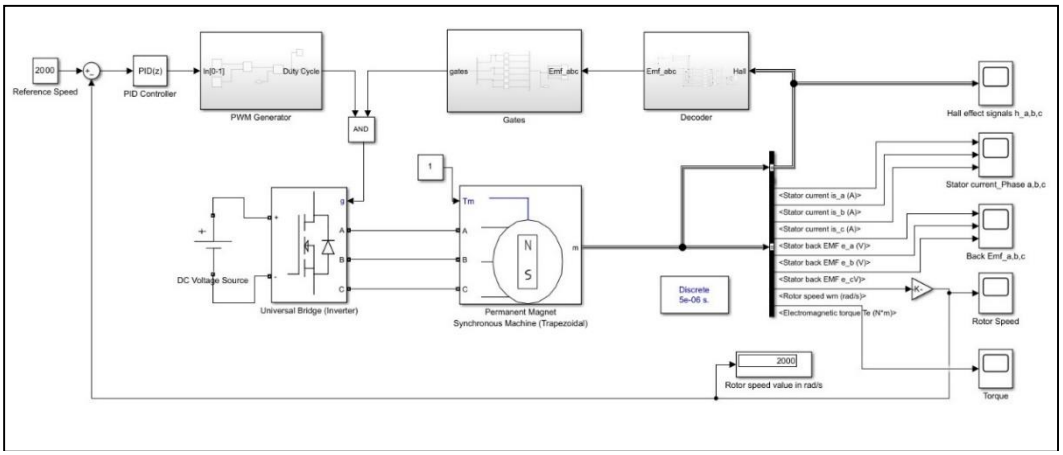

Fig. 3 *BLDC motor control modeling block*

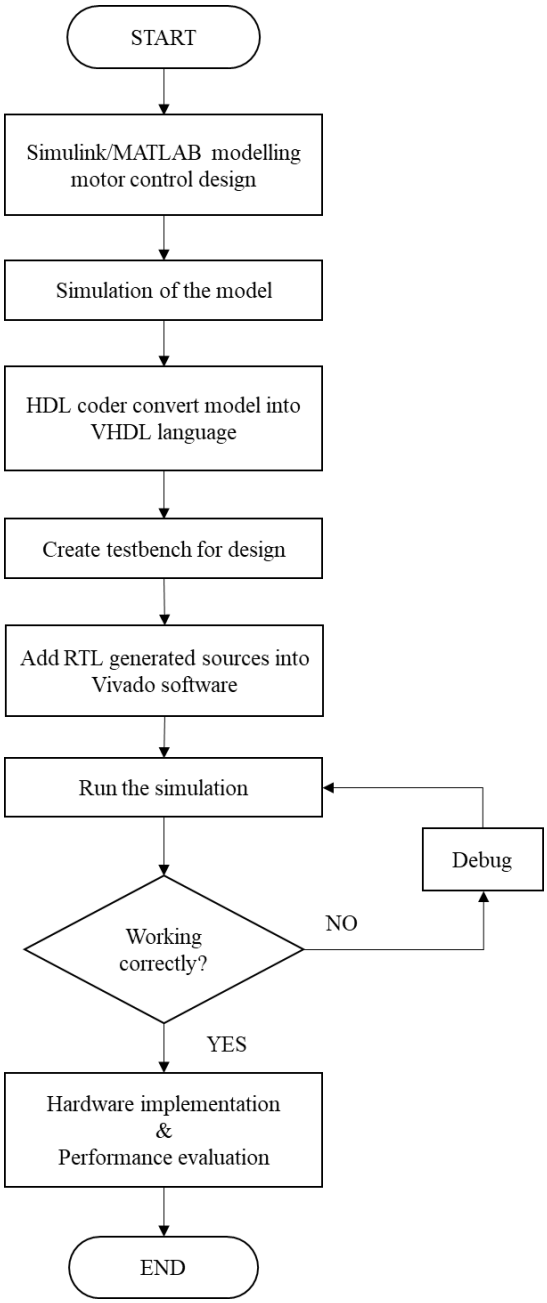

Fig. 4 *Vivado simulation and implementation*

Evolution in Electrical and Electronic Engineering Vol. 5 No. 1 (2024) p. 115-122

2.5 Vivado Simulation and Implementation Flowchart

Fig. 5 shows the diagram of BLDC motor control for generating VHDL code. The conversion of this diagram used the 'HDL coder' application provided in MATLAB Simulink. Referring to Fig. 5, some model blocks were removed and replaced by ports. These unsupported blocks are categorized as power electronics devices from the SimPowerSystem Library, which are not supported by HDL coders and are incapable of being built in FPGA. Therefore, input ports for the Universal Bridge (Inverter), Hall Effect signals, and Rotor Speed output ports are replaced to model their behaviors.

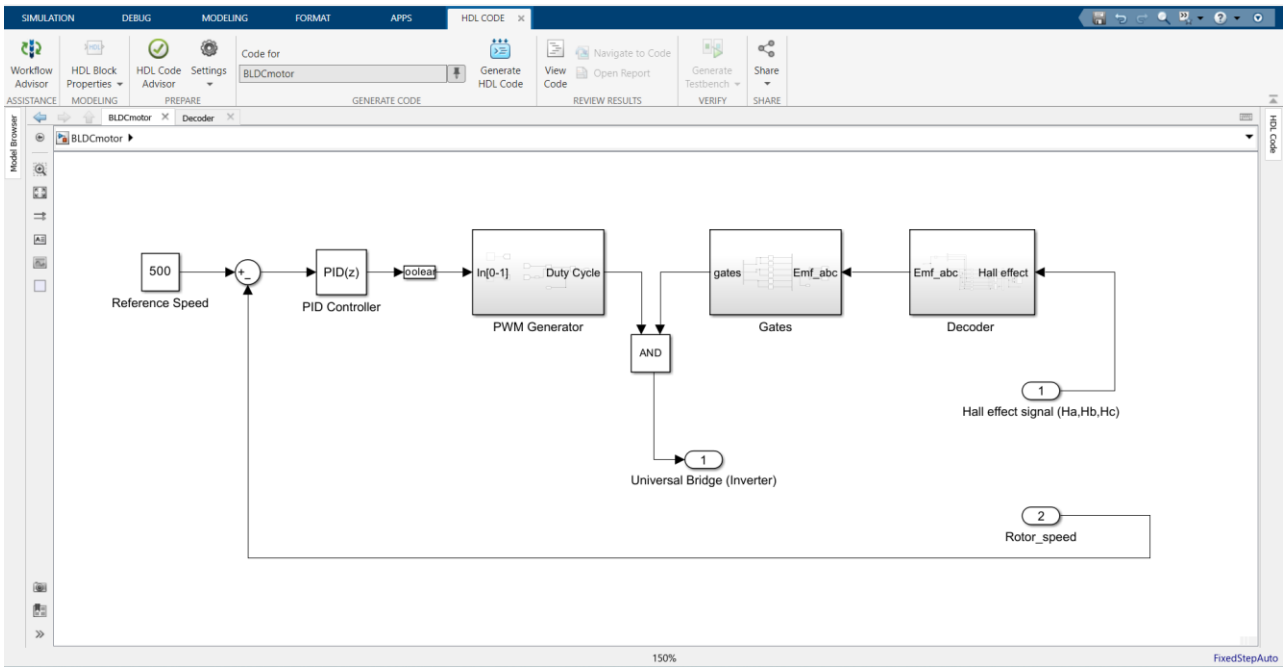

Fig. 5 *VHDL code generation block diagram of BLDC motor control*

2.6 HDL Coder Generation Optimization

Fig. 6 shows several optimization options that are provided in Simulink HDL code. Two options were used for this work optimization: pipeline and resource sharing. Both optimizations can significantly minimize the chip area process of various operations simultaneously and reduce the operation's latency.

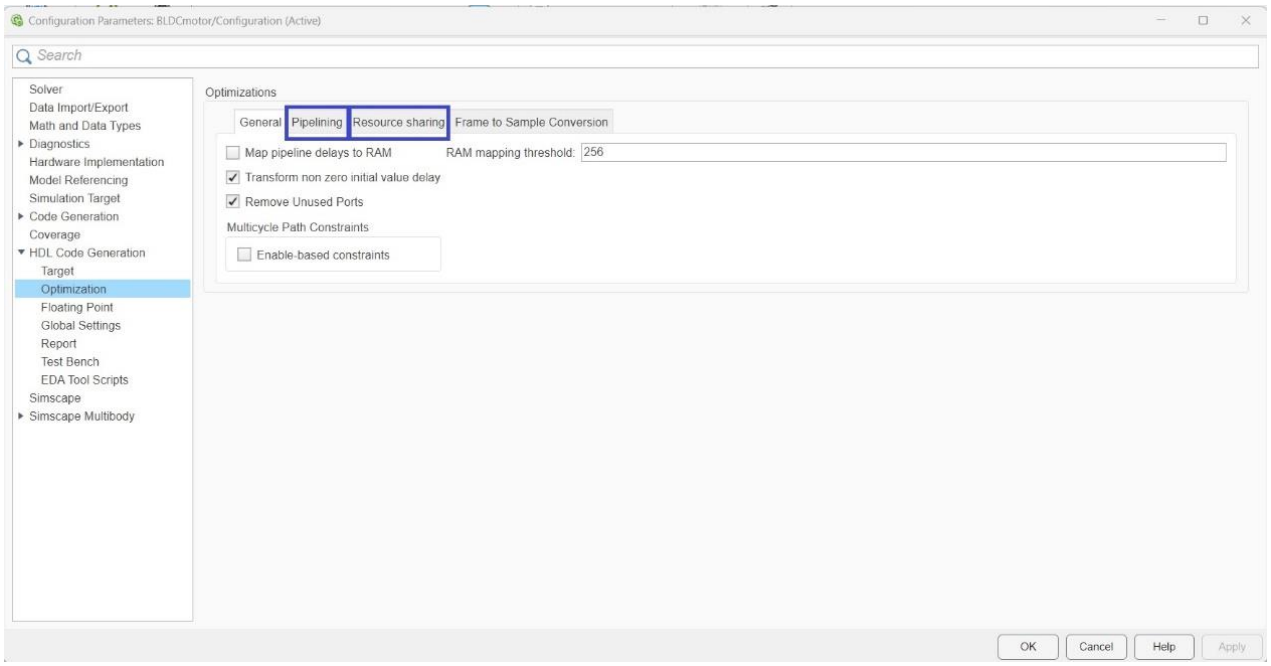

Fig. 6 *Optimization options in Simulink MATLAB HDL code generation*

3. Result and Discussion

3.1 Comparison Modeling Simulation and Hardware Architecture Simulation

The results in Fig. 7 and Fig. 8 show the comparison between simulation on Simulink and Vivado. Fig. 7 shows three signals representing the Rotor Speed, the Hall Effect Signal, and the Universal Bridge Inverter. At the same time, Fig. 8 illustrates the Vivado waveform results of the generated Simulink HDL coder that consists of input data of 64-bit “Rotor_Speed," 3-bit "Hall_Effect_Signal_Ha_Hb_Hc” and output of 6-bits “Universal_Bridge_Inverter.

Cursor 1 in Fig.7 illustrates the motor's initial condition, accelerating before reaching a constant speed, as indicated by Cursor 2. The waveform exhibits result that closely resemble those obtained in the Simulink environment as presented in Vivado implementation in Fig. 8. Besides, Cursor 2 in Fig. 7 is positioned during the achievement of a constant speed, where the Hall Effect Signal is at binary 110 in 12 microseconds. This indicates that by following the attainment of constant speed, the duty cycle of the PWM remains consistently generated until the completion of the process, which can also be seen in the Vivado simulation result in Fig. 8.

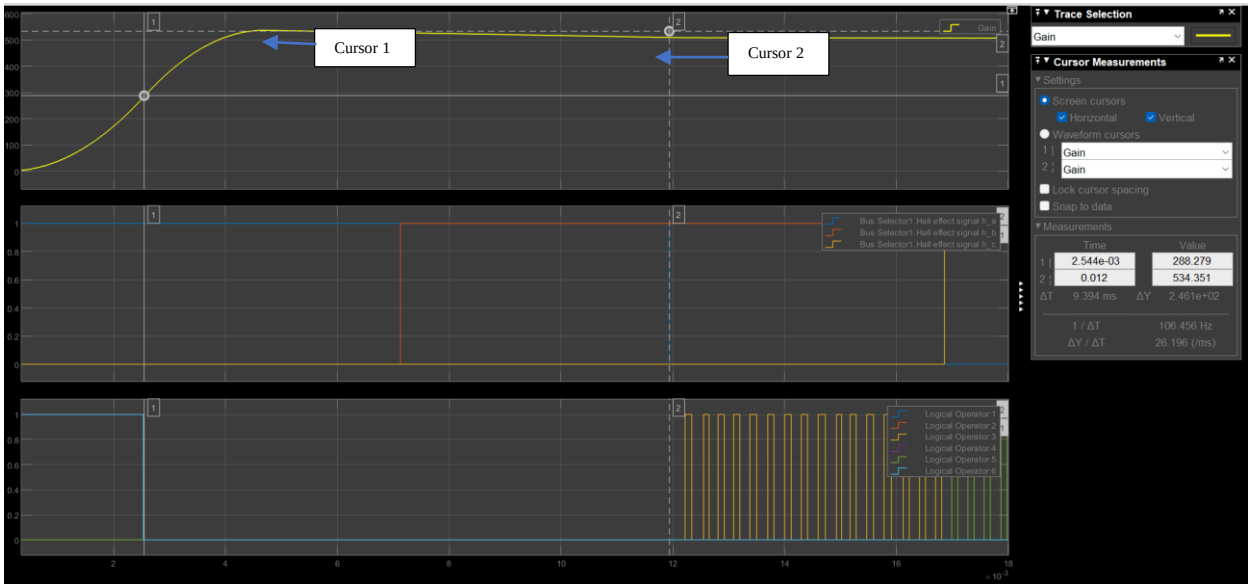

Fig. 7 *Simulink Model Simulation Result*

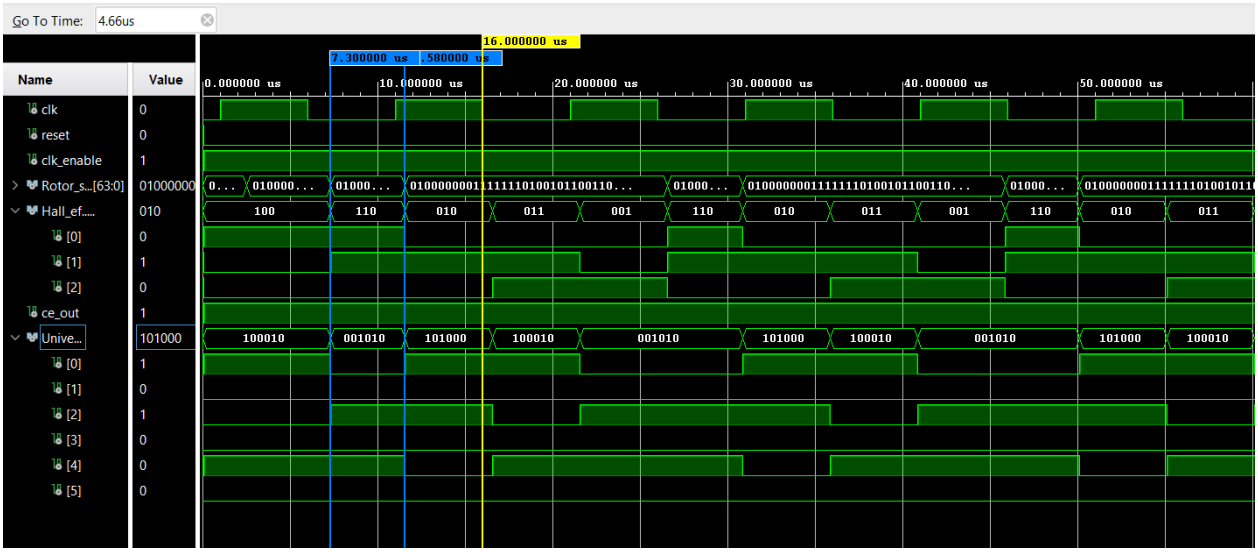

Fig. 8 *Waveform Vivado Simulation Result*

Both simulation outcomes validate the effectiveness of utilizing Simulink MATLAB for circuit diagram design and HDL code generation. The code generated through this process is seamlessly applicable in Vivado implementation. This approach demonstrates ease and convenience and facilitates efficient FPGA hardware implementation and debugging.

Evolution in Electrical and Electronic Engineering Vol. 5 No. 1 (2024) p. 115-122

3.2 Design Optimization of Hardware Implementation

Next are the design optimization results of hardware implementation regarding power consumption, utilization, maximum clock speed, and timing summary. Table 1 shows the pipelining and resource-sharing optimization results on the VHDL code after the implementation. Based on this simulation testing results on design control for FPGA in Table 1, it is proven that enabling both optimizations can improve the timing of the design on the target FPGA by optimizing the critical path. Next, the design consumes more power as more logic and connections in the hardware architecture are needed to satisfy fast-timing requirements. Worst Negative Slack (WNS) of Pipelining or Resources Sharing is 95% better than Baseline, and both optimizations are 96% better than Baseline. Therefore, by enabling optimization, the amount of time a signal fails to meet its required timing constraints on the critical is faster than no optimization.

Table 1 *Performance of under-analysis elements for pipelining and resources sharing optimization*

Performance under analysis	Results
Power consumption	0.325 W
Maximum clock speed	100 MHz
LUT utilization	13.79%
Worst Negative Slack (WNS)	1.632 ns

4. Conclusion

In summary, this work aimed to enhance the control of the BLDC motor system used in welding CNC machines by implementing FPGA-based control systems. By utilizing PID and PWM controllers together with modeling Simulink MATLAB and Vivado verification, significant progress was achieved in improving the accuracy and efficiency of motor control. The HDL code generated from Simulink MATLAB demonstrated effectiveness when implemented on the FPGA, successfully verifying the system's performance. The main findings include the effectiveness of BLDC motor control systems modeling, hardware architecture verification, and optimization impacts on the FPGA-selected device. Applying the trapezoidal control approach in conjunction with PID and PWM controllers demonstrated encouraging outcomes in terms of power consumption, clock speed, and HDL code use. Besides, implementing Pipeline and resource-sharing techniques contributed to improved power efficiency and faster timing within the FPGA. Some recommendations for future works on this work are as follows.

1) Improved Angular Position Accuracy: Future research should focus on integrating additional components, such as encoders, to achieve higher accuracy in determining angular position. This enhancement would further enhance the precision and reliability of motor rotation, ensuring better overall performance.
2) Hardware Implementation: While the work successfully validated the BLDC motor control system through simulation, future efforts should prioritize hardware implementation. Real-world scenarios demand accurate simulations of speed, angular position, and latency, which can be achieved through hardware implementation. This step would provide a more comprehensive understanding of the system's performance and validate its suitability for practical applications.
3) Enhanced Testbench Functionality: Incorporating the capability to read text values in the testbench for FPGA environments would enable more thorough testing approaches. This enhancement would facilitate the simulation of input data with varying parameters, leading to more accurate and precise output data and timing analysis. Researchers can ensure robust validation of FPGA-based control systems by improving testbench functionality

Acknowledgement

The authors would like to thank the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

The author confirms sole responsibility for the following: study conception and design, data collection, analysis and interpretation of results, and manuscript preparation.

References

[1] Chen, W. C., Cheng, M. Y., Kuo, T. H., & Li, C. L. (2011). Development of Intellectual Property for Brushless DC Motor Drives. *Symposium on Industrial Electronics and Applications* (pp. 499–504). https://doi.org/10.1109/ISIEA.2011.6108762

[2] Sathyan, A., Milivojevic, N., Lee, Y. J., Krishnamurthy, M., & Emadi, A. (2009). An FPGA-based novel digital PWM control scheme for BLDC motor drives. *IEEE Transactions on Industrial Electronics*, 56(8), 3040–3049. https://doi.org/10.1109/TIE.2009.2022067

[3] Hidayat, N., Samman, F. A., & Sadjad, R. S. (2022). FPGA Based Controller of BLDC Motor Using Trapezoid Control. *14th International Conference on Information Technology and Electrical Engineering (ICITEE)* (pp. 58–63). https://doi.org/10.1109/ICITEE56407.2022.9954075

[4] Varghese, L., & Kuncheria, J. T. (2014). Modelling and Design of Cost Efficient Novel Digital Controller for Brushless DC Motor Drive. *Annual International Conference on Emerging Research Areas: Magnetics, Machines and Drives (AICERA/iCMMD)* (pp. 1–5). https://doi.org/10.1109/AICERA.2014.6908229

[5] Tsai, M.-F., Quy, T. P., Wu, B.-F., & Tseng, C.-S. (2011). Model construction and verification of a BLDC motor using MATLAB/SIMULINK and FPGA control. *6th IEEE Conference on Industrial Electronics and Applications* (pp. 1797-1802). Beijing, China. https://doi.org/10.1109/ICIEA.2011.5975884

[6] Blignault, C. (2002). Design, Development and Analysis of The Friction Stir Welding Process (Doctoral dissertation, PORT ELIZABETH TECHNIKON).

[7] Wang, X., Zhang, D., & Zhang, Z. (2023). A review of dynamics design methods for high-speed and high-precision CNC machine tool feed systems (pp. 1–30). arXiv.org. https://arxiv.org/abs/2307.03440

[8] Tufekci, M. B., Onal, B., Dere, H., & Ugurdag, H. F. (2020). Efficient FPGA Implementation of Field Oriented Control for 3-Phase Machine Drives. *IEEE East-West Design & Test Symposium (EWDTS)* (pp. 1-5). Varna, Bulgaria. https://doi.org/10.1109/EWDTS50664.2020.9224884

[9] Arulmozhiyal, R. (2012). Design and Implementation of Fuzzy PID controller for BLDC motor using FPGA. IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES) (pp. 1-6). Bengaluru, India. https://doi.org/10.1109/PEDES.2012.6484251

[10] Watts, L. (n.d.). CNC Feeds & speeds formula (how to calculate feed rate). CNCSourced. Retrieved February 2, 2024, from https://www.cncsourced.com/cnc-machining/cnc-feed-rate-and-speed-feeds/

[11] Gamazo-Real, J. C., Vázquez-Sánchez, E., & Gomez-Gil, J. (2010). Position and speed control of brushless DC motors using sensorless techniques and application trends. *Sensors (Basel)*, 10, 6901–6946. https://doi.org/10.3390/s100706901