Implementation Of Brushless DC Motor Using FPGA Interface

Mohammed Zoheb, Vijeyta Sharma M, Mrs. ShaliniVashishtha (Sr. lecturer, A.I.T),
Mr. MohammedShahid (Sc. 'E', A.D.E)
Department of Electronics and Communication Engineering, Atria Institute of Technology, Bangalore-24
Aeronautical Development Establishment(DRDO), MST Department,CV Raman Nagar,Bangalore

Abstract

This paper presents implementation of brushless DC (BLDC) motor using Xilinx FPGAto drive a six switch 3 phase inverter using PWM control technique. The main reason we choose brushless DC over brushed DC is because BLDC motor is a digitally driven device. Hence it is more suitable in today's digitally driven industry. The other advantages of the BLDC motor drives are high efficiency, low maintenance and long life, low noise, control simplicity, low weight and compact construction. Simulation of the implemented BLDC motor is carried out using FPGA SPARTAN-3A trainer kit from Xilinx with the help of VHDL and Verilog programming algorithm of digital PWM Generator topology.

***Index terms*---**Brushless DC (BLDC) motor, Xilinx, Field-Programmable gate arrays (FPGAs), Pulse width modulation (PWM), Inverter.

1. Introduction

Many industrial, commercial, and domestic applications require variable speed motor drives. Traditionally, DC machines have dominated this application area. The main disadvantages are its commutators and brushlessbecause; DC motors operate by virtue of an electro-mechanical commutator, which is a major source of unreliability. In practice, DC motors require frequent maintenance to ensure trouble free commutation. Furthermore, the sparking associated with the action of a commutator makes a DC motor drive unsuitable for use in certain industrial locations,since the sparks can cause radio interference or ignite flammable gases. In addition, the relative orientation of brushes and commutator bars are fixed and cannot be changed during operation.

The requirement from industry for reliable, low-maintenance and controlled drives has therefore led to considerable work in recent years on various "brushless" drive systems. This controlled drive achieves the desired performance by the combination of a machine and power electronics. Brushless DC motors (BLDC's) have enjoyed an increasing usage in disk drives, laser printers and other equipment where a uniform speed is required. The primary reasons for their popularity are the absence of commutators and brushes as found in ordinary DC motors and their ability to respond to digital input pulses.

The absence of commutators and brushes offer several advantages. Since a BLDC motor has no mechanical contact with a moving member (rotor), there is no debris generation, a critical aspect in disk drives that demand a particle-free environment. It is very reliable and maintenance free, usually for the lifetime of a machine.

The speed of the motor is directly proportional to the applied voltage. By varying the average voltage across the windings, the speed can be altered. This is achieved by altering the duty cycle of the base PWM signal.

The use of PWM in power electronics to control high energy with maximum efficiency & power saving is not new but, interesting is to generate PWM signals using HDL and implementing it in FPGA.

The paper presents the simulation of the speed control of BLDC motor, which can be done using the software XILINX with the help of VHDL programming. An FPGA based speed controller is designed for closed loop operation of the BLDC motor so that the motor runs much close to the reference speed.

FPGAs are increasingly being used in motor control applications due to their robustness and customizability. Microcontrollers have typically been used to implement motor controls, with computation algorithms executed by software. Some of the challenges in this implementation are response time, a fixed number of PWM channels, limited communication interfaces and pre-determined analog triggering. The solution is to use an FPGA.

2.Driver strategies

BLDC motors have two types of driving strategies namely, uni-polar and bipolar drives. A simple three phase uni-polar operated motor uses optical sensors (phototransistors) as position detectors. Three phototransistors are placed on the end-plate at 120 degree intervals,and are exposed to light in sequence through a revolving shutter coupled to the motor shaft.

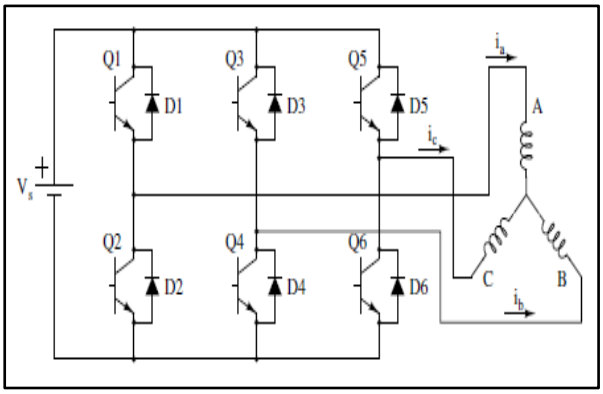


Figure 1. Driver Stage

When a three-phase (brushless) motor is driven by a three-phase bridge circuit, comprised of either MOSFETs or IGBTs as shown in *Figure.1*, the efficiency, which is the ratio of the mechanical output power to the electrical input power, is the highest, since in this drive an alternating current flows through each winding as an ac motor. This drive is often referred to as 'bipolar drive'. Here, 'bipolar' means that a winding is alternatively energized in the south and north poles. The bipolar driving strategy includes sensored and sensor-less techniques.

Position encoders and back emf are used in sensor-less techniques. Whereas, hall sensors, optical sensors are used in sensored technique.

The back electromotive force (EMF) can have a trapezoidal or sinusoidal waveform. Typical back EMF waveforms for a three-phase BLDC motor with trapezoidal flux distribution are shown in *Figure. 2*. As it can be seen, the back EMF induced per phase of the motor winding is constant for 120° and changes linearly with rotor angle before and after the constant part. In order to obtain constant output power, current is driven through the motor winding during the flat Portion of the back EMF waveform.

In this paper we will consider that, the rotor information is being collected using hall sensors.The Hall position sensors are placed 120 degrees apart and they sense the actual rotor position. Unlike a brushed DC motor, the commutation of a BLDC motor is controlled electronically. To rotate the BLDC rotor, the stator windings should be energized in a sequence. It is important to know the rotor position in order to follow the proper energizing sequence.

For the present system, rotor position is sensed using Hall Effect sensors, *Table 1*. Can be obtained with values ranging from one to six. Each code value represents a sector in which the rotor is presently located. Each code value therefore gives us information on which windings need to be excited to turn the rotor. State '0' and '7' are considered invalid states for Hall Effect sensors.

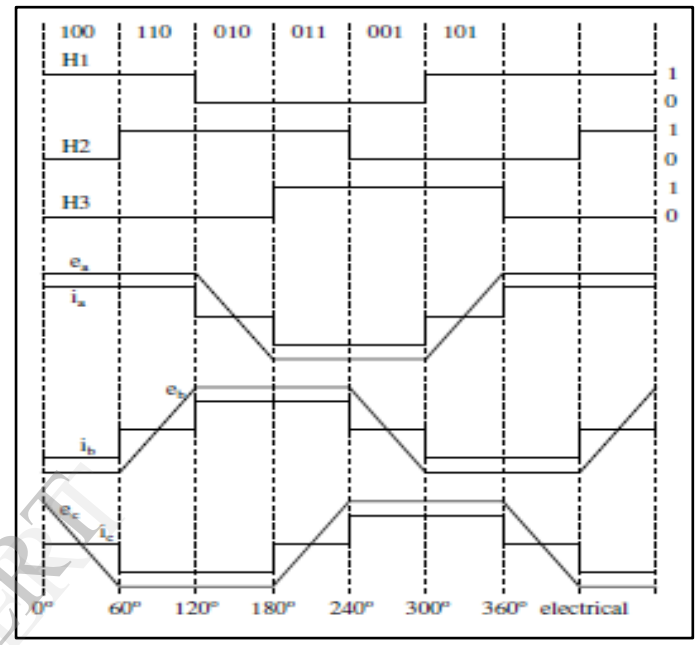


Figure 2. Back EMF and phase current variation with rotor electrical

Hall sensor A	Hall sensor B	Hall Sensor C	Phase A	Phase B	Phase C
1	0	0	$-V_{DCB}$	$+V_{DCB}$	NC
1	0	1	NC	$+V_{DCB}$	$-V_{DCB}$
0	0	1	$+V_{DCB}$	NC	$-V_{DCB}$
0	1	1	$+V_{DCB}$	$-V_{DCB}$	NC
0	1	0	NC	$-V_{DCB}$	$+V_{DCB}$
1	1	0	$-V_{DCB}$	NC	$+V_{DCB}$

TABLE 1. Hall Sensor States

The electrical-to-mechanical ratio is based on the pole pairs inside the motor. Each state corresponds to the actual rotor position inside the motor. This determines the required direction of voltage vector

based on the direction in which the rotor needs to be moved.

A timing diagram showing the relationship between the sensor outputs and the required motor drive voltages is shown in *Figure.3*.

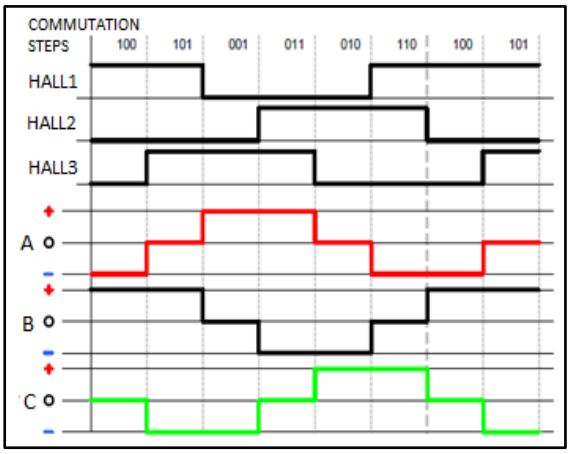
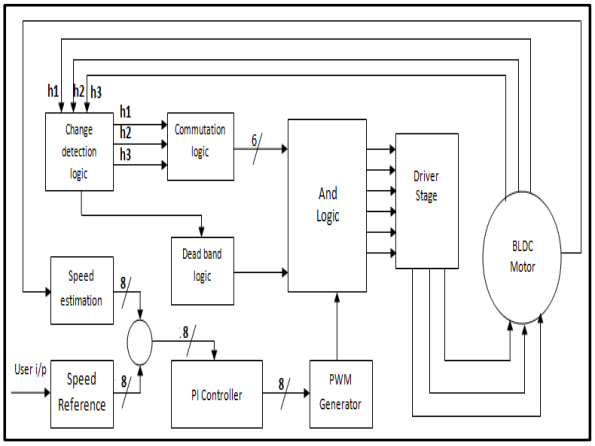


Figure3. Voltage Waveform

3. BLDC speed controller operation

A schematic diagram of the BLDC motor speed controller is shown in *Figure 4*. The controller is totally digital, and is interfaced to the motor through a driver circuit (Inverter Bridge) and by three Hall Effect sensors.

A discrete proportional-integral (PI) type speed regulator is used.The figure presents the block diagram of the controller. Each part of the controller is explained.We use a synchronous design, which is driven by a 50MHz clock signal. The logic blocks are implemented by VHDL programming using Xilinx



System Generator for FPGA and hence integrated with the rest of the design.

Figure4. BLDC Motor with Controller

A. Change Detection Logic

The hall sensor states acquired from the BLDC motor is first saved into three D type flip-flops atevery clock signal rising edge. Another set of FFs are used to accumulate the previous inputs obtained from the hall sensors, which is the next state of the hall sensor. This hall sensor state is then fed to the commutation logic block. The new inputs are then compared with the previous inputs and hence a change in the inputs is represented by a logic '1'. If no change is seen between the new input and the previous input, then a logic '0' is obtained. This one cycle pulse is used by the dead band generation logic to start the dead time measurement.

B. Commutation Logic

To make the motor turn the coils are energized in a predefined sequence, making the motor rotate in one direction, say clockwise. Running the sequence in reverse order the motor will rotate in the opposite direction. One should understand that the sequence defines the path of the current flow in the coils and thereby the magnetic field produced by the individual coils. By varying the current flow in the coils and thereby the polarity of the magnetic fields at the right moment – and in the right sequence – the motor rotates. Alternation of the current flow through the coils to make the rotor rotate is referred to as commutation.Based on the commutation sequence presented in the *Table 1,Figure 5*. The commutation logic is designed along with the Hall sensor readings (A, B, C). The commutation logic derives 6 signals, which represents the state of one of the 6 transistors in the driver stage(inverter) and hence, the hall sensor which is switched on for the appropriate rotor position.

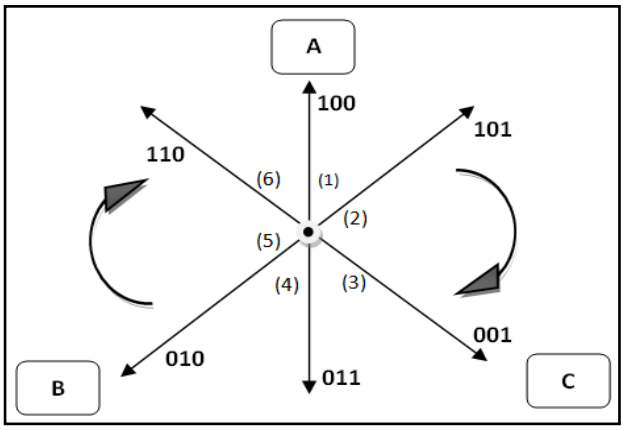


Figure 5. Clockwise commutation

C. Dead Band Generation Logic

As discussed in the previous section, the commutation from one energizing state to another should be deferred by a "dead time" adequate to cover the transistor switch on or switch off time. For this setup, a dead time of at least 0.5µs is required. The dead time generation logic consists of an 8 bit counter, a comparator and a D type flip flop (FF), with synchronous reset. Both counter and the FF will be reset when there is a change in the Hall sensor states, i.e. when the change detection logic produces a logic '1' which is the dead-band logic's input. From this event, the counter begins to count. The counter outputs are given to the comparator which then sets the FF. The output of the FF is therefore 0 only for a small amount of time which is equal to the time taken by the change detector to change from one state to another, given by the counter.

D. PWM Generation and AND Logic

The PWM generation logic consists of a 8 bit counter, a zero detector, a D type FF and a SR FF. The counter is driven by the same 10MHz clock signal.PWM signals are generated from the Spartan-3 processor by writing VHDL program to control the driver. The switching signal parameters are namely, switching frequency and the duty ratio. The numbers of pulses are easily controlled via VHDL programming language. Each output of the commutation logic block, dead-band logic and the PWM generator are given to the AND logic, the result of which is then fed to a six transistor driver circuit (inverter).

E. PI Controller

The regulation of speed is done with the PI controller. The error difference between the actual speed and reference speed is calculated at every PWM cycle and is given as an input to the PI controller.

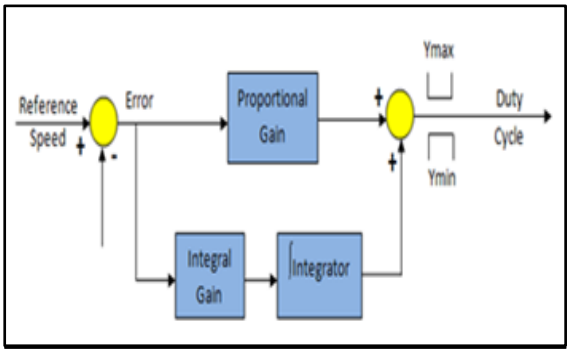


Figure 6. PI Controller

The duty cycle output from the PI controller is given in continuous time domain as:

$$Dutycycle = KP * error + KI * \int error\ dt$$

Where,

KP: Proportional Gain.
KI: Integral Gain.
error: Difference in Reference inspeed with Actual speed.
DutyCycle: Controller Duty cycle Output.

In discrete time domain the same PI controller is represented by the following equations:

$$yn(k+1) = yn(k) + KI * e(k)$$

$$Yn(k+1) = yn(k+1) + KP * e(k)$$

Where,

KP: Proportional Gain
KI: Integral Gain
e(k): Difference in Reference in speed with Actual speed
Yn (k+1): Current computed duty cycle
yn (k+1): Current integrated error term
yn (k): Previously integrated error term

4. Results

The proposed architecture operation is verified using Xilinx SPARTAN-3 board with a FPGA processor, to generate the PWM signal and commutation logic. Hence the simulated results are seen .Here, the results are shown for each of the module explained earlier.

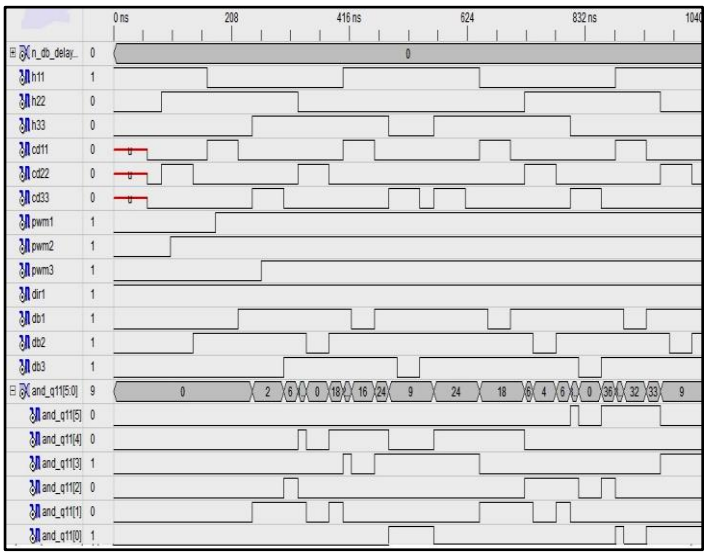


Figure 7. Simulation Result

The onboard running clock of 10MHz was used and hence the resulting PWM signal was generated using a8-bit down-counter. The VHDL program modeling used for the architecture has been synthesized using XILINX v7.1i software. The PWM signals are successfully used to drive a six switch MOSFET

driver stage, which acts as a DAC, needed to energize the windings of a three phase BLDC motor.
Therefore successful results are obtained as the BLDC motor rotates and hence speed of the motor can be varied by varying the duty-cycle of the PWM signal.

5.Conclusion

This paper explains a method to implement BLDC motor using Spartan 3 FPGA kit. This paperdescribes the analysis work that was carried out ona hall-sensored brushless d. c motor. The aim ofthe analysis is to model a brushless DC motor driveoperating in closed-loop modes using a six switchinverter. A generalized approach and computersimulation method for comprehensive analysis ofa brushless dc motor drive has been presented. Theanalysis takes into account all possible modes ofoperation of the inverter including commutation generation of PWM and PI control. The FPGA kithas been mainly used as a controller of the brushlessDC motor, whereas, the inverter is mainly used foramplification and conversion. All the waveformshave been validated and verified using Xilinx 7.1vVHDL code.

6. Acknowledgement

It would not have been possible to write this paper without the kind support of benevolent support around me to whom we express our sincere gratitude. We owe our sincere thankfulness to our guides who made us believe in ourselves and guide us through the whole process of the study. This study would not have been possible without the guidance, help and critical thoughts of our mentors and guidesMrsShaliniVashishtha and Mr. Mohammed Shahid(Sc.'E',MST). Their advice and support at all stages of our study and implementation has been invaluable on both academics and personal level for which we are extremely grateful. This work was carried out at Aeronautical Development Establishment (A.D.E), Defence and Research Establishment as our final year project work for the year 2012-2013.

7. References

[1]. Electrical Technology Vol. 2 by B.L Tharaja and A.K Tharaja.
[2]. K. Alfoni Jose and S.M Padmaja "Implementation of BLDC Motor" International Journal of Applied Sciences, Engineering and Technology Vol. 01, No.01, August 2012.
[3]. Amtel "Sensor-based control of three phase Brushless DC motor" application notes.
[4]. Robert I. Lorincz, Mihai E. Basch, Ivan Bogdanov, "Hardware Implementation of BLDC Motor and Control System Diagnosis" International Journal of Circuits, Systems and Signal Processing".
[5]. StefánBaldursson "BLDC Motor Modeling and Control– A Matlab/Simulink Implementation".
[6]. Microsemi "Speed Control of Brushless DC Motors-Sinusoidal Commutation with Hall Sensors" user guide.
[7]. Peterchev, J. Xiao,S. Sanders, Architecture and IC implementation of digital VRM controller, IEEE Transaction on Power Electonics 18 (2003) 356-364.
[8]. Patella, A. Prodic, A. Zirger,D. Maksimovic. High Frequency digital PWM controller IC for DC-DC converters, IEEE Transaction on Power Electronics 18 (2003) 438-446.
[9]. Rashid M., "Power electronics: circuits, devices and application" Prentice Hall India, 2004
[10]. JayaramBhasker, "VHDL-Primer", 3rd Edition Pearson Prentice Hall, 1999.
[11]. MATLAB 2011 user guide.
[12]. Leonard N. Elevich, "3-Phase BLDC MotorControl with Hall sensorsUsing 56800/E Digital SignalControllers"Freescale Semiconductor Application Note, Rev. 2.0, 11/2005.
[13]. StefánBaldursson, "BLDC Motor modelling and Control – A Matlab/SimulinkImplementation", International Master's program in Electric Power Engineering, May 2005.
[14]. EftichiosKoutroulis, ApostolosDollas, Kostas Kalaitzakis, "High-frequency pulse modulation implementation using FPGA and CPLD IC's", Elsevier journal of system architecture, October 2005.
[15]. Nazeih M. Botros, "HDL Programming, VHDL and Verilog", 2008.
[16]. D P Kothari , I J Nagrath, "Electric Machines", Edition 3, 2006.