Design and Implementation of FPGA Based Bipolar Stepper Motor Controller for Linear Slide Application

Lam Chee Yuen and Phaklen Ehkan
Embedded, Network and Advanced Computing (ENAC), School of Computer and Communication Engineering, Universiti Malaysia Perlis (UniMAP), Pauh Putra Campus, 02600 Arau, Perlis, Malaysia.
cheeyuen_lam@hotmail.com

***Abstract*—Stepper motor is a common linear actuator in automation. This motor is used in the design of one axis automated cutter motion control with a linear slide. The cutter required a high precision motion and location control to avoid miss-cut condition. With the advantage of parallelism of FPGA, a precise stepper motor control signal is generated to drive the stepper motor. Concurrent logic circuit in FPGA calculated the distance and direction of motion in synchronize mode. Trapezoidal velocity profile and motion control are implemented using finite state machine (FSM) in FPGA. The distance resolution per step achieved in this study is 15.88µm for 8-microstep configuration. The FPGA stepper motor controller consumes only 1 % logic source on Altera DE2 FPGA board.**

***Index Terms*—FPGA; Stepper Motor; Parallelism; Velocity Profile.**

I. Introduction

Stepper motor has been used in automation design for long time due to its precise step rotation motion. Therefore, stepper motor is commonly employed in the development of high precision automation such as printers, pick-and-place robot, robotic arm, CNC machine, and CD-ROM. Stepper motors are well received in automation industry because of its well-known precision performance. The rotor of stepper motor is rotate according to the numbers of pulse signal supplied by the digital controller. Besides that, the speed of the motor can be easily controlled. Without additional electronic circuit, the rotor rotation speed can be easily manipulated by adjusting the digital pulse frequency.

In the development and implementation of an automated cutter system, linear slide is used to guide the cutter head motion in one axis. The cutter head required high precision motion and location control to avoid miss-cut occurrence. The digital controller is the main key component to control the motion smoothness, speed and location accuracy for the linear slide. Field programmable gate array (FPGA) is used to develop the stepper motor controller in this study. Parallelism and precise clocking are the main advantages of FPGA used in this study. This paper covers from the stages of design, development and details implementation of the work. Experimental results are also presented in Section V.

II. Literature Review

Literature survey is conducted on the stepper motor and its controller. Several studies which are related to this work had been found. These include a design of stepper motor controller, speed profile, signal analysis and its application.

A. Hybrid Bipolar Stepper Motor

There are three types of stepper motor available in the market which are unipolar stepper, bipolar stepper and variable reluctance stepper [1][2]. Bipolar stepper motor is used in this study. Bipolar stepper motor is capable to produce more torque than unipolar stepper motor. However, it consumes more power than unipolar stepper motor and hence, required additional circuitry to provide and regulate the electric current flow.

B. Stepper Motor Controller

Several works of designing stepper motor controller were reported by using FPGA. These controllers are used to control multiple stepper motor simultaneously [3][4]. As an expansion of simultaneous multi control of stepper motor, this technique has implemented in robot motion control [5]. By using the parallelism advantage of FPGA, parallel stepper motor control signal is generated by the controller. Besides designing controller that interfaced with stepper driver, a microstep stepper motor controller is also being directly designed using FPGA [6]. A wireless FPGA stepper motor controller is developed to eliminate the wiring restriction [7]. The controller is utilizing Bluetooth technology for wireless communication.

A variable microstepping rate is used to investigate the stepper motor positioning accuracy [8]. In design stage of the controller, the frequency response of the stepper motor is investigated [9]. This is to examine the signal noise when the stepper pulse frequency is increased accordingly. After designed a stepper motor controller, the controlling scheme needed to be benchmarked. The benchmarking method is suggested by Jabeen et al.[10].

C. Speed Profile

Steppers motors are dissimilar with DC motors. By changing the applied voltage, it is unable to change the rotation speed of the motors but changing the frequency of the pulse instead of voltage does [11]. Without a proper speed profile, the stepper motors may misstep or slip [12]. There are two common types of speed profile applied to stepper motors which are trapezoidal and triangular profiles. Both profiles are implementing linear acceleration and deceleration. Speed profile helps to optimize the stepper motor to achieve maximum rotation speed [13]. Speed profile is an important factor contributes to CNC machine tools accuracy [14].

III. Controller Design

The digital architecture of the controller is designed on a platform of Altera Quartus II software using Very High speed Hardware Description Language (VHDL). At top level, this controller takes number of pulse from the origin point as destination location and generating pulse signal and direction signal as output signal. A 10 kHz clock signal is fed to drive the logic circuit in the controller.

Figure 1: Controller input and output

A. Architecture

This section discusses the architecture of the controller. This controller is mainly constructed by two major components; concurrent logic circuit and a finite state machine (FSM). The concurrent logic circuit is relatively simple. There are two main registers are implemented in the concurrent logic circuit called destination register and current location register. When the intended location of the cutter head is calculated, the location is passed to the destination register and then a high pulse in 'Data_Rdy' signal latches the destination into the register.

The 'current_location_reg' is used to keep track the instantaneous location of the cutter head in terms of number of pulse away from origin point. When the cutter head move further away from the origin point, the value in the 'current_location_reg' will be incremented. If the cutter head move otherwise, the value will be decremented. The stepper pulse generated by the FSM is used as clocking signal to directly drive the 'current_location_reg'. The direction signal is used to multiplex the increment or decrement value of the register.

There are two subtractors implemented in this architecture which are used to calculate the instantaneous distance between the cutter head current location with the destination location. Two subtractors are used to avoid signed integer to occur during calculation of the distance. This technique has limited the calculated distance to produce in absolute magnitude value. The calculated distance and direction is then passed to the FSM to perform motion control.

There are also two limit switch signals fed to the FSM. These signals are the liner slide left and right boundary signal and implemented for linear slide motion protection purpose as 'End of rail' signal. Besides that, left limit signal is also used as origin location signal for initialization purpose.

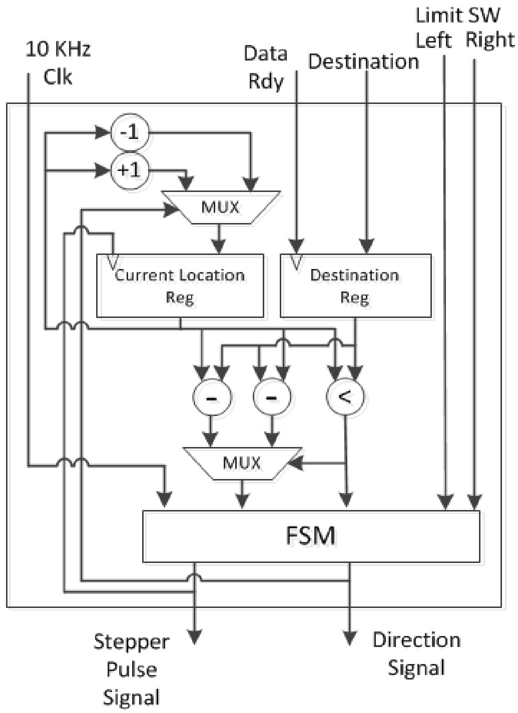

Figure 2: Stepper motor controller architecture

B. Finite State Machine (FSM)

The FSM designed in this study is the main motion control core. This FSM is responsible to generate direction signal, stepper pulse signal and control the timing sequence of the signal. This FSM has total of six states starting from Initialize, Standby, Accelerate, Cruise, Decelerate and Stop.

At the initialization of the state machine, it is defined to start at Initialize state. At this state, the FSM set the direction signal to '0', and drive the cutter head to the origin point. In other word, the FSM is driving the cutter head to left until touches the left limit switch triggered signal '1' from the left limit switch. Then, the FSM comes into Stop state, 'current_location_reg' is reset, and the FSM drive the cutter head to the right and away from the origin point to a predefined location. When the cutter head reaches the predefine location, it decelerate and stop. At this moment, the FSM enters 'Standby' state. At 'Standby' state, the FSM is waiting for incoming destination request. No pulse signal is generated at this state.

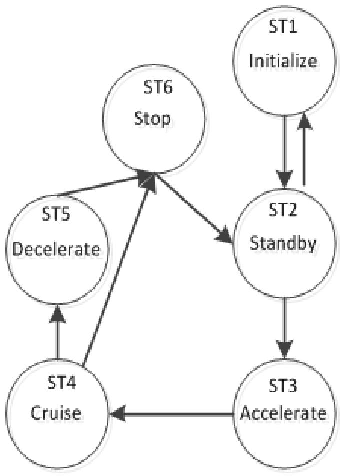

Figure 3: Stepper motor controller FSM

When a destination request is received, FSM read the destination direction and start to accelerate the stepper motor and move the cutter head. The motion profile of the FSM is explained in following section. The FSM continuously monitor the distance to the destination during ‘Accelerate’, ‘Cruise’ and ‘Decelerate’ states. Once the distance to the destination is equal to 5 pulse cycle distance or less, the FSM will enters ‘Decelerate’ state and ready for stop. The FSM controls the stepper motor to fully stop its rotation when distance value equal to zero. As a protection measure, the FSM also continuously monitoring the left and right limit switch signal to prevent end of slide reached. Either left or right limit switch is touch, the FSM stop the pulse signal immediately and back to ‘Standby’ state. After a delay time is passed, the FSM return to ‘Initialize’ state to recalibrate its location.

C. *Velocity Profile*

The FSM in this study has implemented with stepper motor velocity profile. Velocity profile is crucial for smooth motion control. Every mechanical motion is impossible to move at high speed instantly or stop at high speed instantly. The motion must start at low speed then accelerate up to desire velocity and begin to decelerate into a stop.

The velocity profile implemented in this study is trapezoidal profile. This profile has a linear accelerate phase, constant cruising phase and a linear decelerate phase.

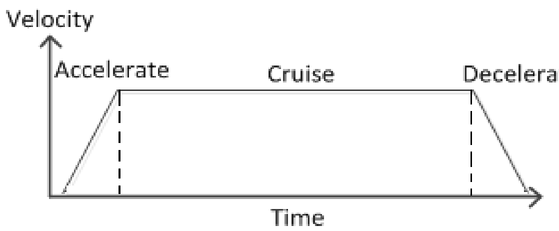

Figure 4: Stepper motor controller FSM

The starting pulse frequency is begun at 100 Hz which is equal to 10ms interval. Since this controller is designed to accelerate in 45° in velocity slope, the inter pulse delay time required to be constantly decreased. Since this controller architecture is driven by a 10 kHz clock signal. A counter is used to count the acceleration pulse delay. When acceleration, the counter constantly reduces its delay value until cruise velocity achieved.

$$delay\_value_t = delay\_value_{t-1} - 2 \quad (1)$$

$$delay\_value_t = delay\_value_{t-1} + 10 \quad (2)$$

The counter initially set to count 100 from 10 kHz clock to obtain 100 Hz pulse. When accelerate, its count value is decreased by 2 for every pulse generated which is shown in Eq. (1). During ‘Cruise’ state, the delay value is maintaining the same. When decelerate, the delay value is incremented by 10 to reduce decelerate period until delay value of 44 and enters full stop condition.

IV. Interfacing Circuit

In order to ensure stepper motion smoothness, a stepper motor driver module Polulu-A4988 is used in this work. This module has maximum of 5.5 V for logic control voltage, and maximum of 35 V for motor driving voltage and it is capable to supply maximum 2 Ampere of current per coil to the motor. An advantage of this A4988 module is that it has implemented photocoupler transistor for circuit protection from the motor.

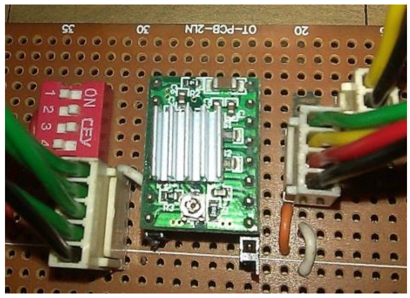

Figure 5: Pololu A4988 stepper driver module on PCB

The A4988 module is configured to 8 microstep setting according to the combination table supplied in the datasheet. The logic control voltage is supplied with 3.3 V from FPGA board, and the ground is wired back to the FPGA. For the motor driven side, the power is supplied directly from the power adapter, with 12 V. The ‘enable’ pin is let floating. FPGA controller is driving ‘Direction’ and ‘Pulse’ pin only. For the stepper motor side, four pin ‘A+’, ‘A-‘, ‘B+’, ‘B-‘ are wired to the coil of the stepper motor. The voltage trimmer mounted at the center of the module is used to adjust electric current that flows to the coil. The electric current flow to the coil is double of the voltage value measured between the trimmer and ground. The stepper motor used in this work is from Wantai Motor (China) model 57BYGH603 with step angle of 1.8°.

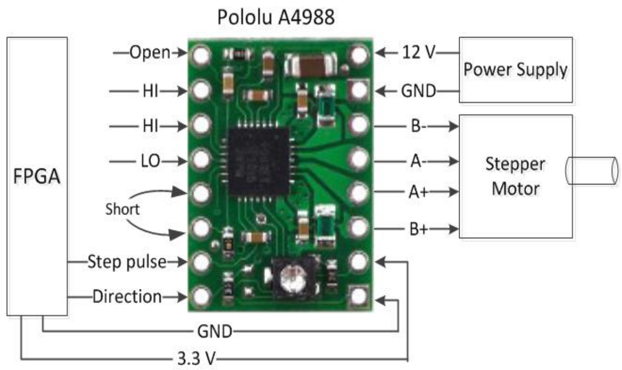

Figure 6: System wiring diagram

V. Results and Discussion

A. *Simulation Result*

The digital architecture of the stepper motor controller is going through simulation process before implemented into physical experiment. The simulation is to inspect the functionality of the stepper motor controller with a given input value. It is noticeable that the stepper pulse signal is performing acceleration stage where the pulse delay is continuously reducing. A stepper pulse signal is entered ‘Cruise’ state then decelerates and stops.

From the simulation waveform shown in Figure 7, the desire location is at 328 pulses away from the origin point. Once the location is latched by the ‘cal_in’ signal, the stepper controller begins to accelerate and move toward destination. The controller is then accurately stops at location 328 pulses away from origin.

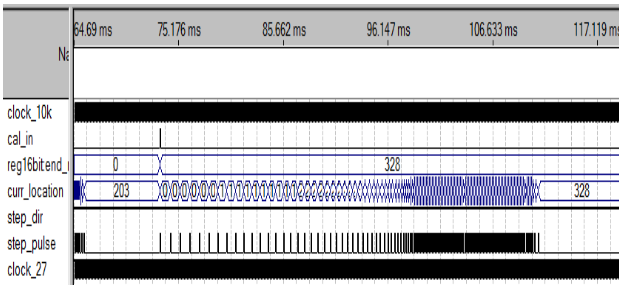

Figure 7: Simulation waveform

B. *Implementation and Test Run*

After simulation verification completed, the controller is integrated with the DE2 FPGA board to perform hardware test run. The experiment setup is shown in Figure 8. Since this study is only a prototype testing, the stepper motor controller is configured to run at slow speed. The cruising speed of the cutter head is designed at 2.7 cm/s with cruising pulse frequency of 1666.66 Hz. The displacement resolution of the cutter head motion achieved in this study is 15.88µm per step.

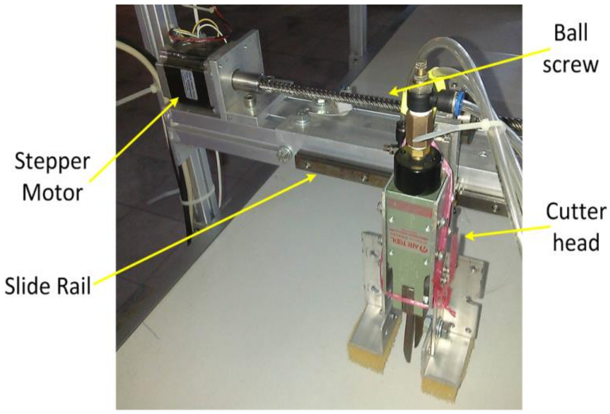

Figure 8: Linear slide and cutter head couple with stepper motor

C. *Hardware Resources Utilization*

The FPGA hardware resources utilized by stepper motor controller are listed in Table 1. The controller consumes only 1% of the hardware logic resources of the Altera FPGA chip EP2C35F672C6. Although the FSM consists of 6 states and implemented using 1-hot method by Altera Quartus II software, the arithmetic operation for each state for conditioning and decision making embedded each state of the FSM uses an abundance of logic resources. On the other hand, with only 1% hardware utilization merit, parallel implementation of multiple controller modules can be achieved and did not affecting much on the FPGA resources.

Table 1
Hardware Resources Utilization

Component	Utilized Logic Element (LE)	Total Utilization (%)
Concurrent Logic Circuit	165	0.496
FSM	170	0.511
Total	335	1.000

VI. CONCLUSION

A FPGA based stepper motor controller is designed and tested. This hardware controller is capable to deliver a precise stepper rotation motion. The designed architecture uses only 1 % of the hardware resources available in the Altera Cyclone II EP2C35F672C6 FPGA chip. The controller capable to control stepper motor rotation with implemented trapezoidal velocity profile. The displacement resolution for the cutter head motion achieved in this work is 15.88µm.

REFERENCES

[1] R. Condit, Dr. D.W. Jones, “Stepping motors fundamentals (application note no. AN907)”, Microchip Technology Inc., 2004. Retrieved 11 Nov 2015. URL: http://www.microchip.com/ wwwAppNotes/AppNotes.aspx? appnote=en012151

[2] Liptak, Bela G. (2005). Instrument Engineers' Handbook: Process Control and Optimization. CRC Press. p. 2464. ISBN 978-0-8493-1081-2.

[3] U. Meshram, P. Bande, P. A. Dwaramwar and R. R. Harkare, "Robot arm controller using FPGA," 2009 International Multimedia, Signal Processing and Communication Technologies, Aligarh, 2009, pp. 8-11.

[4] J. Zhu and G. Liao, "Design of multi-axis motion control system for stepping motor," 2010 WASE International Conference on Information Engineering, Beidaihe, Hebei, 2010, pp. 23-26.

[5] Hagiwara, H., K. Asami, et al. FPGA implementation of image processing for real-time robot vision system. Convergence and Hybrid Information Technology: 5th International Conference, ICHIT 2011, Daejeon, Korea, September 22-24, 2011. Proceedings. Berlin, Heidelberg, Springer Berlin Heidelberg: 134-141.

[6] Ngoc Quy Le and Jae Wook Jeon, "An open-loop stepper motor driver based on FPGA," 2007 International Conference on Control, Automation and Systems, Seoul, 2007, pp. 1322-1326.

[7] N. K. Thulasiraman, H. A. F. Mohamed and Y. S. Cheng, "A reconfigurable wireless stepper motor controller based on FPGA implementation," 2010 IEEE Symposium on Industrial Electronics and Applications (ISIEA), Penang, 2010, pp. 858-590.

[8] D. R. Gaan, M. Kumar and C. G. Majumder, "Variable rate based microstepping of stepper motor using MATLAB GUI," 2017 Indian Control Conference (ICC), Guwahati, 2017, pp. 385-390.

[9] S. Dorin-Mirel, M. Ion and O. Mihai, "The analysis of different frequency for a stepper motor open loop operation," 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, 2016, pp. 1-7.

[10] S. Jabeen, S. K. Srinivasan, S. Shuja and M. A. L. Dubasi, "A formal verification methodology for FPGA-based stepper motor control," in *IEEE Embedded Systems Letters*, vol. 7, no. 3, pp. 85-88, Sept. 2015.

[11] Jose I. Quinones, “Applying acceleration and deceleration profiles to bipolar stepper motors,” *Analog Applications Journal*, Texas Instrument Incorporated, 3Q 2012, pp. 24.

[12] You, B. and L. Pei, A novel stepper speed control algorithm based on FPGA. Advances in Computer Science and Education Applications: International Conference, CSE 2011, Qingdao, China, July 9-10, 2011. Proceedings, Part II. Berlin, Heidelberg, Springer Berlin Heidelberg: 205-209.

[13] P. Ranade, “Linear motor control without the math,” *EETimes.com.* Retrived 12 Nov 2015.

[14] Qian Feng and Lei Wang, "FPGA-based acceleration and deceleration control for CNC machine tools," Proceedings 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC), Shengyang, 2013, pp. 210-214.