Design and Simulation of PID Controller using FPGA

Ankur Dave
PG Student
Department of Electronics & Communication Engineering
Atmiya Institute of Technology & Science Rajkot, India

V. S. Vora
Assistant Professor
Department of Electronics & Communication Engineering
Atmiya Institute of Technology & Science Rajkot, India

Abstract

Proportional Integral Derivative (PID) controller is the most preferable controller in industries that does not require precise analytical model of the system to be controlled. Implementation of PID controllers has gone through several stages of evolution from the early mechanical and pneumatic designs to the microprocessor based systems. Recently “Field Programmable Gate Array” has become an alternative solution for the realization of Digital PID controllers. To overcome the hardware complexity by the use of more processors for multi-channel, we are using single PID controller for multi-channel. With all any of all threes gains, desired performance can be achieved with this control by carefully choosing gains. PID controller deals with three control actions. They are P (proportional) control, I (integral control) and D (derivative control).

Keywords: PID Control, Field programmable gate array (FPGA), Digital signal processing (DSP)

I. INTRODUCTION

Due to its simple arithmetic and robust properties, analog Proportional Integral Derivative (PID) controllers are applied in many areas. But nowadays it cannot keep pace with faster, cheaper, more stable and more flexible requirements dictated by modern applications. In recent years, digital controllers implemented with Digital Signal Processor (DSP) or Field Programmable Gate Array (FPGA) have been conceived and designed to improve performance [1]. Implemented with these fixed point devices, floating point arithmetic is usually converted to fixed point format. This paper presents a PID core suitable to be introduced in such a system. The simulation of the VHDL core is performed around the Libero System Generator. The “System Generator” is a collection of Simulink block sets that permit interaction between hardware and modelled systems. In this paper we consider discrete time PID controller and is implemented in a dedicated FPGA with PWM modulator. The simulation of the VHDL core is performed around the Libero System Generator the “System Generator” is a collection of Simulink block sets that permit interaction between hardware and modelled systems. In this paper we considered digital PID controller and it is implemented on given FPGA board with PWM modulator. This paper presented in which a modular FPGA-based design is applied on temperature control system. The same design approach can be extended to other embedded design controller using FPGA board [2].The whole system is implemented by some functions which is required for this modular design. The whole system can be passed on three phases of control systems: (1) Software modelling/simulation in an environment such as Matlab/Simulink; (2) Hardware implementation; (3) Co-simulation of the whole system including hardware and software.

PID controller has all the necessary dynamics: fast reaction on change of the controller input (D mode), increase in control signal to lead error towards zero (I mode) and suitable action inside control error area to eliminate oscillations (P mode) [3]. Derivative mode improves stability of the system and enables increase in gain K and decrease in integral time constant Ti, which increases speed of the controller response. Microprocessors, Microcontrollers and Digital Signal Processors (DSPs) can no longer keep pace with the new generation of applications that requires more flexible [4]. The PID is used in the field of servo motor control, temperature control system, robotics, and signal processing and power electronics. This paper is as follows: In the next section related work of PID controller is discussed. Subsequent section discusses about PID controller. At last simulation and FPGA implementation results are discussed.

II. EXISTING WORK

This paper is presented an efficient design of Proportional-Integral Derivative controller implemented on Field Programmable Gate Array (FPGA) technology. Here we have done design, analysis and implementation of PID controller using FPGA [5]. The research presented in this article applies the newest Field-Programmable-Gate-Arrays to implement motor controller devices in accordance with the actual core based design [6]. The flexibility of the System-on-a-Programmable-Chips in motor multi-axis control systems enables the processing of the most intensive computation operations by hardware (PID IP cores) and the trajectory computation by software in the same device. We discussed here, modular design of embedded feedback controllers using field-programmable gate array (FPGA) technology is studied. The implementation of PID controllers using microprocessors and Digital Signal Processor (DSP) chips is old and well known, whereas very little work can be found in the literature on how to implement PID controllers using FPGA.

Implementations using a single architecture, shared to perform different PID controllers, have also been considered. Since with FPGAs we are not constrained to a specific number of bits to represent data such is the case when using e.g. microcontrollers, the implementation of PID controllers using FPGAs may permit to use in embedded systems more efficient, robust and stable controllers and auto-tuning schemes. Although the advantage to have high-levels of freedom for bit-width representation when using FPGAs, it imposes several design problems due to the large design space exploration, difficulties to convert floating- to fixed-point numbers and difficulties to design reconfigurable hardware, which requires hardware expertise.

III. OVERVIEW OF COMPLETE SYSTEM

It is the most complex mode of controller mode combinations. It results in better control than one or two control modes. But it is difficult to achieve its advantage because of the difficulty of selecting the proper tuning parameters. They are used in process industry to control slow variables such as temperature.

$$c(t) = k[e(t) + 1/Ti \int e(t)\, dt + Td\, d/dt\, e(t)] \qquad (1)$$

Another form of PID in parallel form is modeled as:

$$c(t) = [kp * e(t) + ki * \int e(t)\, dt + kd * d/dt\, e(t)] \qquad (2)$$

Where, c(t) is the controller variable, e(t) is the error signal between reference value and process variable, Kp -term is proportional gain, Ki-term is integral gain, and Kd-term is derivative gain. So, conversion can be easily done with Kp = K, Ki = K/Ti and Kd = KTd. As PID contains all three control parameters, its use and tuning requires great care to deal with various parameters like stability, response time, maximum overshoot etc. There are several methods for tuning a PID loop. The most effective methods generally involve the development of some form of process model, and then choosing P, I, and D based on the dynamic model parameters. Manual tuning methods can be relatively inefficient, particularly if the loops have response times on the order of minutes or longer.

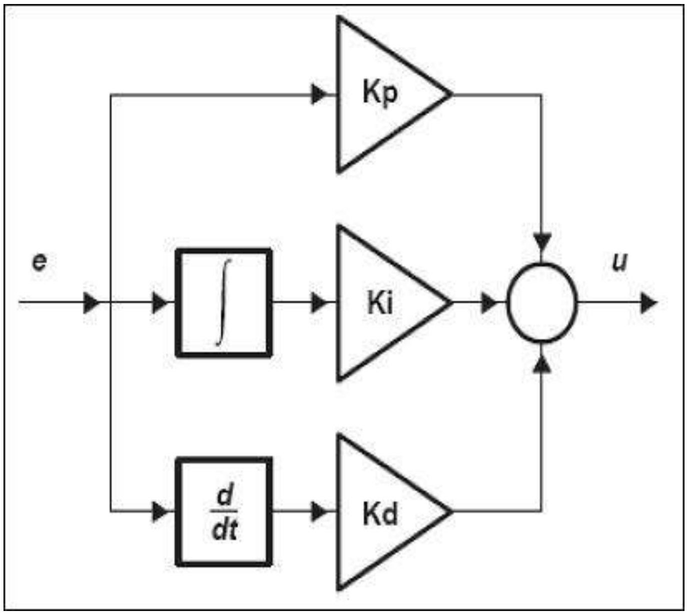


Fig. 1: Block Diagram of PID Controller

This parallel form is shown in Figure 1, where the parameters are treated as simple gains, is generalized form of PID controller. All parameters have the least physical interpretation of the PID controller [8]. In this paper, we focused when the error is more it can differentiate and produce the constant output, when signal is low when compared to reference signal it can integrate it. The derivative action adds phase lead, which improves stability and increases system bandwidth. Figure 2 shows that flow chart of PID control in cooler drive electronics. Initially error is zero then error signal e(t) has considered value is 100.

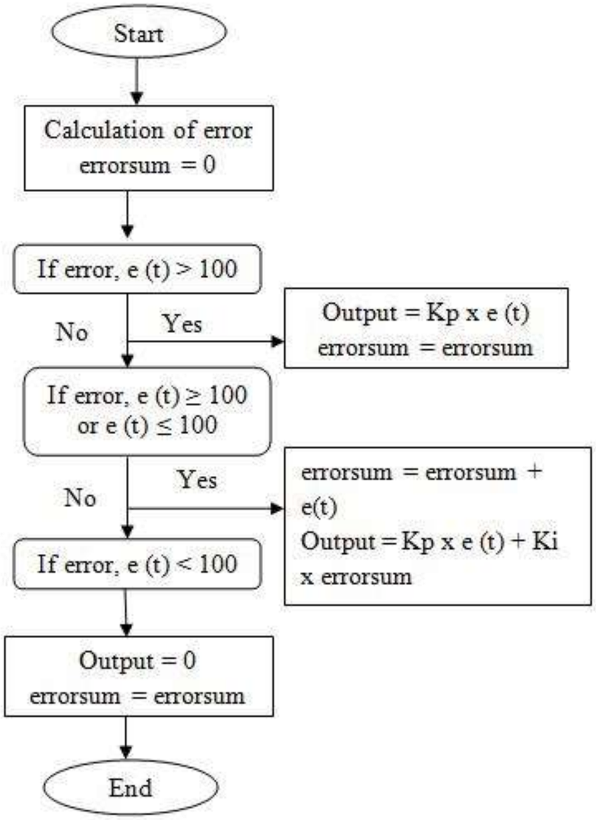


Fig. 2: Flowchart for Implementation of Digital PID Control.

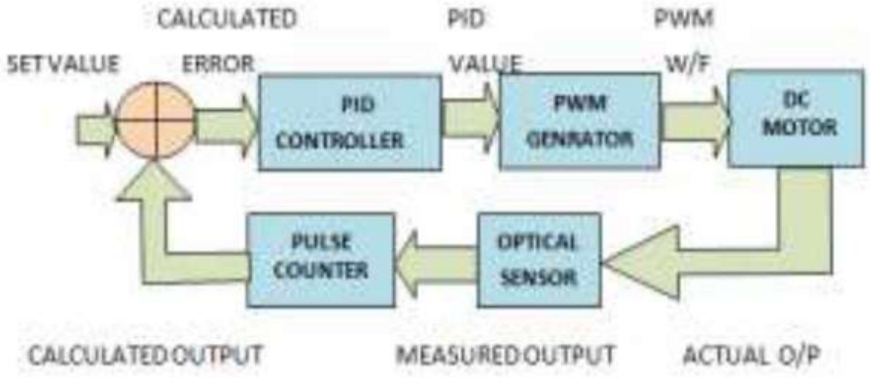


Fig. 3: Block diagram of the complete system

The set speed is assigned to switches according to the requirement and the capture control switch is enabled. Once this is done the generated set value and previous calculated value of speed will be read and sent to the PID controller as an error value. To calculate the current speed optical sensor and pulse counter module is used in the feedback system. The PID controller module will calculate the equivalent PID value and send to the PWM generator module. Figure 3 shows the overall view of the system along with the implemented modules on FPGA. As the set speed is varied, the PWM waveform also varies. The scheme proposed in is based on a distributed arithmetic algorithm where a Look-Up-table (LUT) mechanism inside the FPGA is utilized. The contribution focused on power and area issues while FPGA interfacing is totally unaddressed. In our work we introduce a simple method for implementing PID controllers together with many related constructing modules. Some other contributions focused on proposing algorithms for tuning the coefficients of PID controllers using FPGAs while the controller itself is still implemented in software. These contributions are considered complementary to our work as they provide tools for building adaptive PID applications.

IV. RESULTS

System block diagram is shown in the Fig. 4. The entire system (CDE) is composed of Sensor Processing Circuit, Digital Control Block and Power Amplifier Card, which are already available as hardware since the beginning of the project. As project objectives mainly focus on digital control block, relevant hardware components like FPGA Xc2V3000 (Virtex) and A3PE3000 (PROASIC) along with ADC128s102 (in SPC) and cryo cooler are considered.

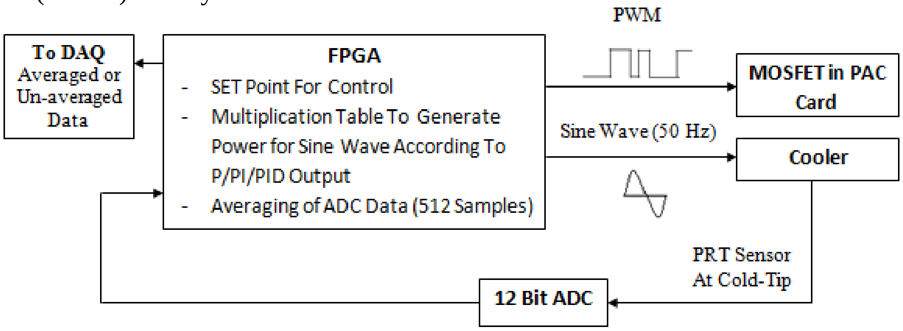
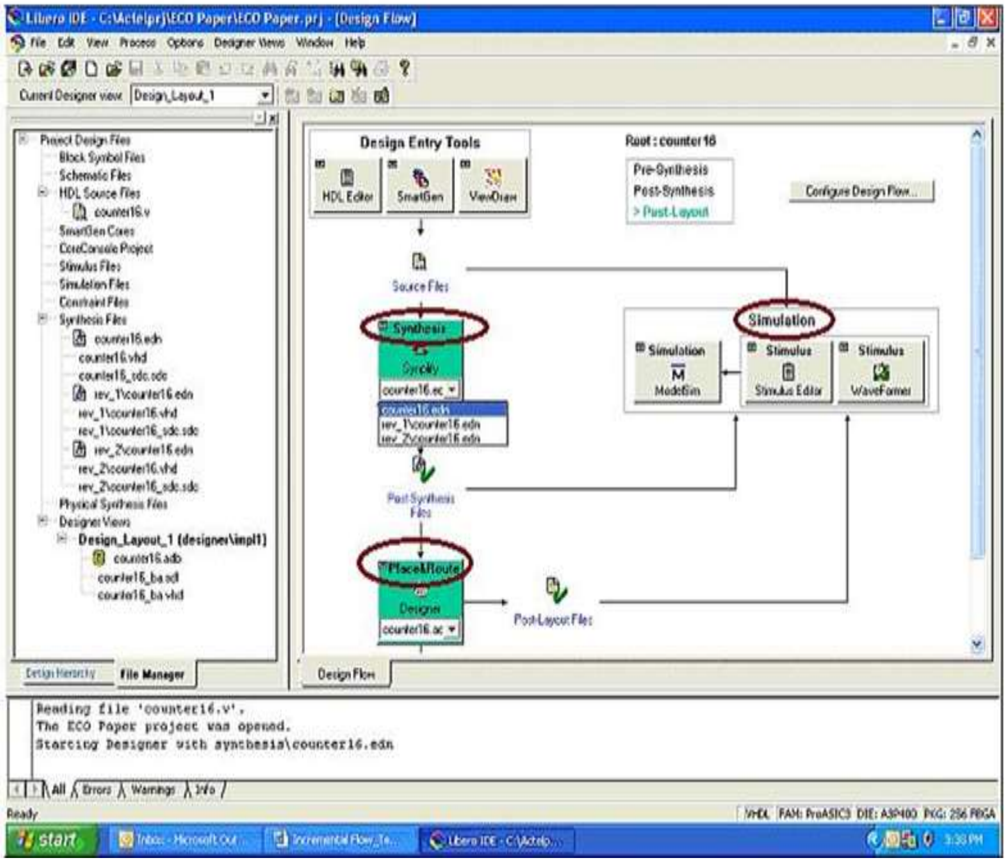


Fig. 4: Block-Diagram for Closed-Loop Implementation

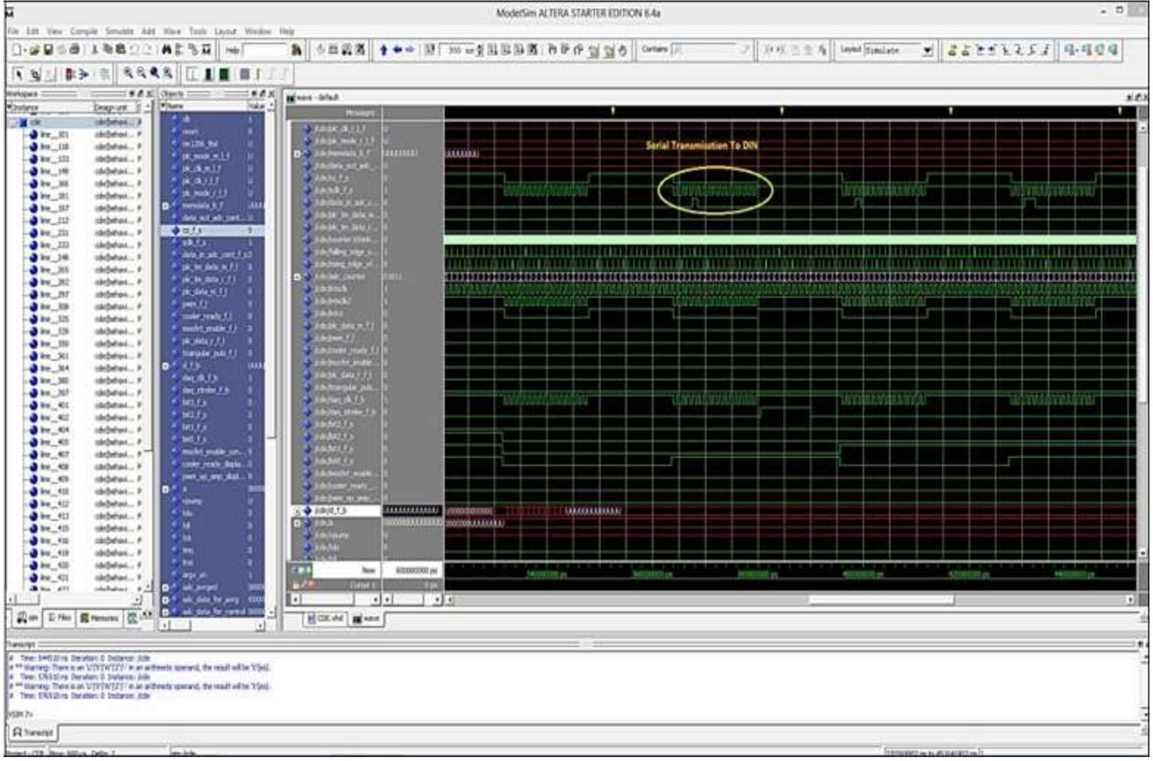
Crystal of 100 MHz supplies the clock to FPGA mounted on the digital control card. FPGA generates sine wave of 50 Hz in order to drive cooler. It also generates PWM wave to enable the MOSFET in PAC card. Alternatively, it can provide triangular wave along with sine wave to class-D amplifier in PAC card for PWM generation. Presently, direct digital PWM wave is provided to PAC card. According to power supplied by sine wave to cooler, it cools down and this can be sensed by the PRT sensor mounted at cold-tip. In feedback path, it is provided to ADC, which generates corresponding digital count, which in turn, is taken as the channel 0 input of the FPGA. For in house data processing, these digital counts are provided to the data acquisition card and data acquisition is done in LabVIEW. ProASIC3E is the third-generation family of Actel flash FPGAs which offers high performance and density features. Nonvolatile flash technology gives ProASIC3Edevices the advantage of being a secure, low-power, single-chip solution that is live at power-up. ProASIC3E is reprogrammable and offers time-to-market benefits at an ASIC-level unit cost. These features enable designers to create high-density systems using existing ASIC or FPGA design flows and tools.



```
Reading file 'counter16.v'.
The ECO Paper project was opened.
Starting Designer with synthesis\counter16.edn
```

Fig. 5: Preview of Libero IDE Screen

FPGA implementation for the XC2V3000 (Virtex) can be done through Xilinx ISE. Similarly, project development and implementation of PROASIC A3PE3000 can be done through Actel Libero. Simulation of these projects can be done through ModelSim by Altera. Actel Libero has been used to develop the project for PROASIC A3PE3000 when the new card has been available. Figure 4 shows preview of Libero IDE screen. It shows that same process steps can be followed in Libero i.e., code development through HDL editor, pre-synthesis simulation, synthesis through Synplify, Post-synthesis simulation in ModelSim simulator, Place and Route and Configuring the device. ModelSim simulator by Altera can be used for simulation along with project development in Xilinx or Actel. ModelSim can also be used to develop standalone project and to simulate it. Figure 5 shows preview of ModelSim window, which is showing simulation of the multiplexing of 8 ADC input channels.


ModelSim ALTERA STARTER EDITION 6.4a
File Edit View Compile Simulate Add Wave Tools Layout Window Help
Workspace
Objects
wave - default
Serial Transmission To DIN
Now
Cursor 1
Transcript
sim
Files
Memories

```
# Time: [illegible] ns Iteration: 0 Instance: /cdc
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
# Time: [illegible] ns Iteration: 0 Instance: /cdc
# ** Warning: There is an 'U'|'X'|'W'|'Z'|'-' in an arithmetic operand, the result will be 'X'(es).
# Time: [illegible] ns Iteration: 0 Instance: /cdc

VSIM 7>
```

Fig. 6: Preview of ModelSim for Simulation of Multiplexing Channels.

V. CONCLUSION

In this paper, a novel PID based controller was presented, for FPGA implementation. The test results showed that with PID controller added, the steady-state error is eliminated and the desired output speed is obtained. The implementation of controller has reduced the total hardware complexity and cost. According to the experiment done it is observed that, in the simulation, when the set speed is changed, the motor speed locks to the set speed, when the current error becomes zero. In brief, the role of FPGA, in measurement and control point of view, is to acquire the data from sensor through analog to digital converter. Although in most industries, PID controllers are the oldest they represent the most used controllers are currently in use and attempts are made in replacing the analog controllers into digital controllers.

REFERENCES

[1] Sreenivasappa BV, Udaykumar RY, Analysis and Implementation of Discrete Time PID Controllers using FPGA, ISSN 0974-2190, 2(1): (2010),71–82p.
[2] Rene Cumplido, Simon Jones, Roger M. Goodall, et al. A High Performance Processor for Embedded Real Time Control, IEEE T Control Syst Technol. May 2005; 13(3): 485–492p.
[3] João Lima, Ricardo Menotti, João MP, Cardoso, and Eduardo Marques, A Methodology to Design FPGA-based PID Controllers, IEEE International Conference on Systems, Man and Cybernetics, October8-11, 2006, Taiwan.
[4] Vikas Gupta, Kavita Khare, Singh RP, Efficient FPGA Design and Implementation of Digital PID Controllers in Simulink, Int J Recent Trends Eng. November 2009; 2(6): 147–150p.
[5] Majed MA, Khandelwal CS, Efficient Dynamic System Implementation of FPGA Based PID Control Algorithm for Temperature Control System, IJEET. July-Sep 2012; 3(2): 306–312p.
[6] Narmadha. G and Deivasigamani. S," An Efficient Multipath delay commutator architecture", International journal of computer application, July2014, vol.98, pp.21-23.
[7] Charaabi L, Monmasson E, Slama-Belkhodja, Presentation of an Efficient Design Methodology for FPGA Implementation of Control Systems: Application to the Design of an Antiwind-up PI Controller, Proc. IEEE Ind.Electron. Soc. Annual Conf., Nov-2002; 3:1942–1947p.
[8] Astrom KJ, Hagglund TH, PIDControllers: Theory, Design and Tuning, 2nd ed. Instrument Society of America, Research Triangle Park, NC, USA, 1995.