Digital Implementation of PID Controller for Temperature Control

PrachiRusia

Abstract—In this paper, design scheme for implementation of digital PID controller using FPGA based microcontroller core is presented for temperature control system. The differential equation includes different controller gains, required to be tuned for desired output response. Usually PID control is much used for second or higher order systems. On the other hand, PI and PD controllers are preferred for first order systems. Here controller is implemented to fulfill the demand of achieving and maintaining the temperature of an Al plate to the desired set point temperature with control precision better than 0.1□C, acceptable rise time and settling time. In implementation load used is a first order system as it has single heat transfer element. For FPGA implementation of PID, VHDL is used for code development. Microcontroller core is used for configuring and interfacing to peripherals using C. Libero IDE is used to develop, simulate, synthesis and place & route the design.

Index Terms—PID, FPGA, ADC, AMBA Bus, APB.

1 Introduction

All industrial processes require control of the output to achieve the desired result. Which include altitude control of an aircraft, speed control of an elevator, voltage control of a power supply and temperature control in a chemical plant etc.

The proportional-integral-derivative (PID) controller is the most used controller in dynamic systems [1]. There may exist one PID control closed loop for each specific control [2]. The implementation of PID controller has evolved from early mechanical and pneumatic designs, to analog circuits using transistors, andlately to micro-processor and digital systems. The emergence of low cost, powerful micro-processors and digital signal processors (DSPs) made it possible for the digital control to become a competitive choice. Moreover, FPGAs are capable to execute concurrent operations, allowing parallel architectural design of digital controllers [3].

2 Pid Controller

Most practical feedback loops are based on PID control or some minor variations of it. Virtually all feedback controllers determine their output by observing the error between the set point and a measurement of the process variable. Errors occur when an operator changes the set point intentionally or when a disturbance or a load on the process changes the process variable accidentally. The controller's mission is to eliminate the error automatically. A feedback controller is designed to generate an output that causes some corrective effort to be applied to a process so as to drive a measurable process variable towards a desired set point. The controller uses an actuator to affect the process and a sensor to measure the results.

There are two types of PID Controller-
1.Analog PID Controllers
2. Digital PID Controllers

2.1 Analog PID Controllers-

Analog controllers are more accurate as they are able to sense continuous and small changes. However, in analog domain to change the parameters of PID controller, circuit components (capacitors, resistors etc.) need to be changed. The

Theoretical model of the Analog PID controller is-

$$Y(t)=K\left[e(t)+\frac{1}{T_i}*\int_0^{\tau}e(\tau)d\tau+T_d*\frac{de(t)}{dt}\right]$$

(1)

Another form of PID in parallel form is modeled as:

$$Y(t)=K_p*e(t)+K_i*\int_0^{\tau}e(t)dt+K_d*\frac{de(t)}{dt}$$

(2)

Where, $Y(t)$ is the controller variable,
$e(t)$is the error signal between reference value and process variable,

P-term is proportional gain,
I-term is integral gain, and
D-term is derivative gain.

Rewriting above equation in S domain-

$$Y(s)=K_p*e(s)+K_i*\frac{e(s)}{s}+K_d*e(s)*s$$

(3)

$$\frac{Y(s)}{e(s)}=K_p+\frac{K_i}{s}+K_d*s$$

(4)

We can convert the parameters from one form to another by-

$$K_p=K$$

$$K_i=\frac{K}{T_i}$$

$$K_d=K*T_d$$

The proportional part acts on the present value of the error, the integral represent average of past errors and the derivative can be interpreted as a prediction of future errors based on linear extrapolation.

2.2 Digital PID Controllers-

They can be implemented using fewer components and provide faster response using Micro-controllers, FPGAs and Microcontroller based FPGAs [2].
For digital implementation, we are interested in Z-Transform of above equation-

$$Y(z)=\left[K_p+K_i*\frac{1}{\left(1-z^{-1}\right)}+K_d*\left(1-z^{-1}\right)\right]*E(z) \tag{5}$$

Rearranging above equation gives,

$$Y(z)=\left[\left\{K1+K2*z^{-1}+K3*z^{-2}\right\}/\left(1-z^{-1}\right)\right]*E(z) \tag{6}$$

Also equation can be written as,

$$Y(z)-z^{-1}*Y(z)=\left[K1+K2*z^{-1}+K3*z^{-2}\right]*E(z) \tag{7}$$

Then converted back to difference equation as-

$$Y(n)=Y(n-1)+K1*e(n)+K2*e(n-1)+K3*e(n-2) \tag{8}$$

Another approach is Rewriting the equation in discrete domain for nth sample -

$$Y(n)=K_p*e(n)+K_i\sum_{i=0}^{n}e(i)+K_d*\left[e(n)-e(n-1)\right] \tag{9}$$

In other form,

$$Y(n)=K_p*e(n)+K_i*e(n)+K_i\sum_{i=0}^{n-1}e(i)+K_d*\left[e(n)-e(n-1)\right]$$

For (n-1)th sample the equation is –

$$Y(n-1)=K_p*e(n-1)+K_i\sum_{i=0}^{n-1}e(i)+K_d*\left[e(n-1)-e(n-2)\right]$$

Subtracting the above two equations, we get-

$$Y(n)=Y(n-1)+K1*e(n)+K2*e(n-1)+K3*e(n-2) \tag{10}$$

Where,

$$K1=K_p+K_i+K_d$$
$$K2=-K_p-2K_d$$
$$K3=K_d$$

This equation is Discrete PID equation, which can be realized in VHDL. Digital domain design provides flexibility in implementing and testing as compared to analog domain.

3 Tuning Of Controller

Tuning a temperature controller involves setting the P, I & D values to get the best possible control for a particular process. There are several different ways to tune a PID Controller for system optimization. One approach is to choose some controller parameters, to observe the behaviour of the system, and to modify the parameters until desired behaviour is obtained. Another approach is to first develop mathematical model that describes the behaviour of the process. The controller parameters are then determined using some method for control design. The tuning rules of PID controllers can be divided in to the following groups:

- Tuning rules that give a specified closed-loop response.
- Tuning rules based on minimizing an appropriate performance criterion.
- Tuning rules based on a measured step response.

In this, Ziegler-Nichols tuning method is used where the controller coefficients are derived from MATLAB-Simulink simulation [5].

4 System Block Diagram

The system blocks consists of FPGA based Microcontroller core and other logics implementing PID controller & PWM generation, control element for heater , Al plate as load, thermistor as temperature sensing element, foil heaters and ADC for Digitizing sensor data. Figure shows the block diagram of the system

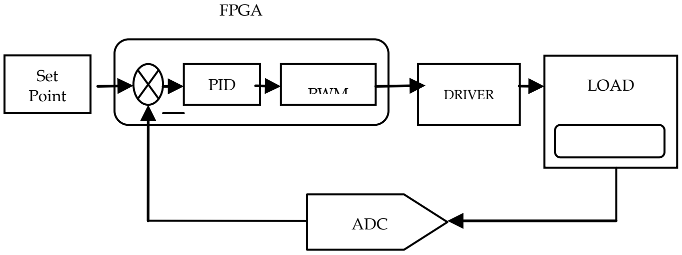

Fig.1. FPGA Based System Block Diagram

This closed loop control system is realized using Microsemi ProASIC3E-A3PE3000 FPGA for controller implementation. A low speed, multichannel 24-bit digitizer (AD7714) is used for digitizing temperature data sensed using thermistor mounted on load. Load is simulated as an Al plate. A 5W foil heater is used as heating element, which is controlled through an NPN transistor (max. power = 800mW).

5 Design Approach

This integrated system for monitoring temperature can be developed using different approaches. This may include implementing the whole system using only microcontroller that will provide sequential execution of the statements in the code and hence is not very appropriate for system requiring concurrent execution of statements. Next is using only FPGAs, providing concurrent execution but has a disadvantage of not having any memory for storing temporary data in the system. The third approach use FPGA based microcontroller core satisfying the need for concurrent execution of the code statements along with integrated program memory.

FPGA based Microcontroller -core approach: In this, FPGA implements the microcontroller core and associatedperipherals like program memory, GPIOs, watchdogs etc. which then interfaces with external sensor and digitizer. The complete system interfaces with temperature sensor (thermistor) giving analog output. The FPGA based Microcontroller core is interfaced with peripherals like UART core, GPIO core, SRAM etc… through AMBA peripheral bus included in the system design. The developed system would interface with the control system, which will achieve and maintain the temperature of the load.

Block Schematic of the System-

The implementation broadly consists of the Microcontroller core and its peripherals forming the subsystem. The top level includes the PID Module, the program memory, counters for generating clock, and peripheral support blocks.

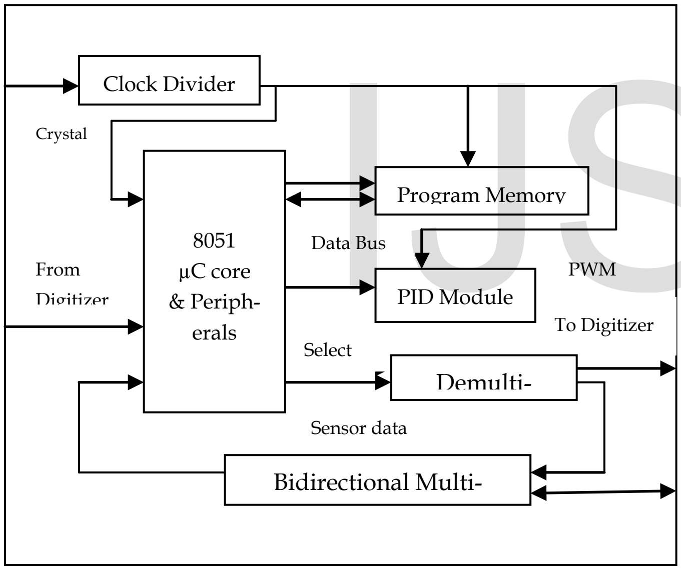

Fig.2. Block Schematic of System

The Program memory- contains the instructions to be executed by the microcontroller core. This interfaces to the microcontroller by a 16 bit Address bus and 8 bit data bus.

Counter (Clock Divider) - The operating frequency is selected at 3 MHz. This is selected to meet the requirement of digitizer (AD7714) which requires this frequency for its operation.

Bidirectional Multiplexors-The temperature sensor has a bidirectional data interface.

PID Module- This is the PID Module designed using VHDL which takes input clock from clock divider circuit and ADC data input from GPIOs. The desired Set point of the controller is defined in the logic code. This module produces PWM Signal resultant to the controller variable.

6 Experimental Results

6.1 RTL Simulation Results-

For Digital implementation, the differential equation is implemented in VHDL Code along with the Test bench for giving the input clock and reset to the system. The desired set point temperature is set based on estimated digital count at ADC output. Estimation of controller variable is carried out and PWM is generated for switching purpose. PWM generation is required when the set point with desired tolerance is achieved.

Initially the PWM duty cycle is 100% till the error is outside tolerance band. As the error reduces to the desired band, corresponding PWM duty cycle depending on the present controller variable is generated. As the set point is achieved PWM duty reduces to 0% achieving control precision.

Simulation is carried out considering initial temperature of 25°C and Set-point set to 28°C. In simulation cycle temperature was forced to increase linearly in the code. The output of controller was observed to show a stable response and generating PWM signal in the desired band. The below shown simulation result shows the clock, reset, gain parameters, set point of the system, controller variable and PWM signal.

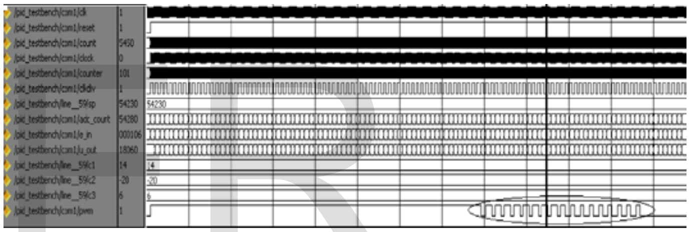
/pid_testbench/cm1/clk 1
/pid_testbench/cm1/reset 1
/pid_testbench/cm1/count 5450
/pid_testbench/cm1/clock 0
/pid_testbench/cm1/counter 101
/pid_testbench/cm1/clkdiv 1
/pid_testbench/line__59/sp 54230
/pid_testbench/cm1/adc_count 54280
/pid_testbench/cm1/e_in 000106
/pid_testbench/cm1/u_out 18060
/pid_testbench/line__59/c1 14
/pid_testbench/line__59/c2 -20
/pid_testbench/line__59/c3 6
/pid_testbench/cm1/pwm 1
54230
14
-20
6

Fig.3. Pre-Synthesis simulation (PWM generation)

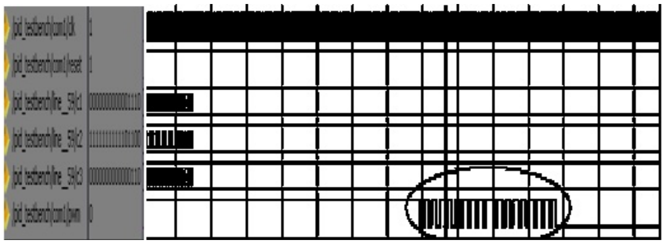
/pid_testbench/cm1/clk 1
/pid_testbench/cm1/reset 1
/pid_testbench/line__59/c1
/pid_testbench/line__59/c2
/pid_testbench/line__59/c3
/pid_testbench/cm1/pwm 0

Fig.4. Post-Layout simulation (PWM generation)

The result of RTL Pre-synthesis and Post-layout simulation shows that the output response of the controller is stable and the controller gains can be used for attaining and controlling the temperature.

5.1 Hardware Test Setup Results

After completing FPGA design for digital PID Controller next is system testing and verification for desired output response with specified heater power rating of 591mW on test setup. The results for control test are shown in the table 1 and the achieved control precision is shown in table 2.

Control Precision Temp. (°C)		Achieved Temp. (°C)	Precision Achieved (°C)
Min. Temp.	Max. Temp.		
28.122	28.172	28.144	±0.025

The desired setting precision for response of the system was 28 ± 0.5°C and achieved is 28± 0.2°C.

The control precision for the response of the system is specified as the precision achieved by the output response after settling time. The aimed control precision for this test set up was ± 0.1 °C and achieved is of the order of ±0.02 °C as seen from the observed results for the real hardware setup.

7 Analysis Of Setting Tolerance

In implementation the setting tolerance may vary in accordance with the uncertainties of different blocks used in the system design. These uncertainties are required to be attributed for defining the tolerance in setting precision of the output response.

The origins of the system uncertainty are as follows-

- Parameters of system model that are approximately known.
- Parameters that vary due to changes in the operating conditions.
- Measurement devices often have imperfections.

Due to these uncertainties,the design aims to achieve the set point within this tolerance of ± 0.5 °C.

8 Concluson

In this paper, implementation of digital PID controller was carried out on FPGA. This design was further tested and modified for simulated load (Al plate with foil heater) in which temperature response was regulated by the proposed PD controller. In hardware implementation it was noticed that developed controller was effective in decreasing overshoot and had faster time response. The implementation results show that the desired response of the system was achieved and maintained.

9 Acknowledgments

The authors are thankful to the experimental labs of SFED/SEG/SEDA-SAC-ISRO, Indian Space Research Organization, Ahmedabad for providing the computing facility, tools for simulations and hardware setup along with the FPGA board for performing the tests during system development.

10 References

	Target	Achieved
Setting temp. (°C)	28±0.5	28.144
Control temperature precision (°C)	±0.100	±0.025

[1] “ PID Controllers: Theory, Design and Tuning”, K.J. Astrom and T. H. Hagglund in Instrument Society of America, Research Triangle Park, NC, USA, 2nd, 1995.

[2] “A Methodology to Design FPGA-based PID Controllers” João Lima, Ricardo Menotti, João M. P. Cardoso, and Eduardo Marques, IEEE International Conference on Systems, Man and Cybernetics, October8-11, 2006, Taiwan.

[3] “Presentation of an efficient design methodology for FPGA implementation of control systems: Application to the design of an antiwind-up PI Controller”, L. Charaabi, E. Monmasson and Slama-Belkhodja in Proc. IEEE Ind. Electron. Soc. Annual Conf., Nov-2002, Vol.3, pp.1942-1947.

[4] “Efficient Dynamic System Implementation Of FPGA Based PID Control Algorithm For Temperature Control System”, M. A. Majed, Prof. C.S. Khandelwal, IJEET, Vol. 3, No. 2, pp.306-312, July-Sep 2012.

[5] “Efficient FPGA Design and Implementation ofDigital PID Controllers in Simulink”,Prof.Vikas Gupta, Dr.KavitaKhare and Dr. R. P. Singh, International Journal of Recent Trends in Engineering, Vol. 2, No. 6, November 2009.

.