Digital Pipelined PID Controller

Nitin sinha[1], Korrapati Eswari Pujitha[2], Vijaykumar Patale[3]

[1]*M.Tech Embedded System SENSE Department, VIT University, nitin03ece@outlook.com*
[2]*M.Tech Embedded System SENSE Department, VIT University, eswaripujitha@gmail.com*
[3]*M.Tech Embedded System SENSE Department, VIT University, vijaypatale@gmail.com*

Abstract- A Proportional-Integral-Derivative controller which is a closed loop controlling system prolifically used in industrial automation process. In the conventional format, PID controller is implemented either in a discrete analog system or by some virtual software means. In this paper pivotal emphasis has been laid over designing a PID controller on the Field Programmable Gate Array (FPGAs) platform. Moreover for utilizing better performance effect of the system, the concept of pipelining is incorporated to facilitate high speed processing.

Keyword— FPGA, PID, Discretization, Backward Difference Method, Pipelining.

I. INTRODUCTION

The ambit of FPGA application is inflating day by day, which is due to its dynamic behaviour which comprises flexibility and high computational speed. Hence these attributes of FPGA leads to acclimating PID controller into this platform. According to the recent survey it has been estimated that PID controller which occupies the controlling mechanism of almost 90-95% of the industrial process. Also, a pivotal nature of this controller is that it doesn't require the exact and relevant model of the process or plant that is being controlled. Hence for this reason, PID controller is put into large utility which is further used in plethora number of application. Therefore this paper inhabits the PID controller mechanism on the Xilinx FPGA Platform. To apropos PID Controller in FPGA requires its discretization. The Discretization process is carried out using backward difference method. So the resultant obtained PID Controller's equation after the application of discretization can be easily implemented in FPGA. Furthermore, a real time transfer function is utilized in this system and on the basis of which the proportional, integral and derivative constant is calculated. The calculated Constant is accosted in the final implementation of the PID Controller. Moreover, this paper also presents pipelining of the whole process to further provide high speed and less time consumption. The simulation and result of the inclining process is also discussed in detail in the subsequent sequence.

II. PID CONTROLLER AND ITS DISCRETIZATION

As widely known, the PID controller, a closed loop controller exploited in industrial systems to control the process or plant efficiently, minimizes the error value by tuning the process input involving the difference of the output value of the process and the desired set point value. PID Controller which is, also named as Three-Term-Controller in which the Proportional, Integral and Derivative entity is represented as P, I and D respectively. Considering in time, P is dependent on the present value whereas I depend on the value of the past while D regulates on the future value.

Conventional form of PID controller is given by,

$u(t) = kp\ [e(t) + 1/Ti \int e(t)\ dt + Td\ d/dt\ e(t)]$ ....(1)

Where,

Kp is the constant of proportional entity or proportionality gain, Ti is the Time Integral constant, Td is the constant of derivative time, whereas, e(t) is defined as error signal and u(t) is computed as the output of the controller.

The most orthodox form of PID Controller, shown in Fig.1,

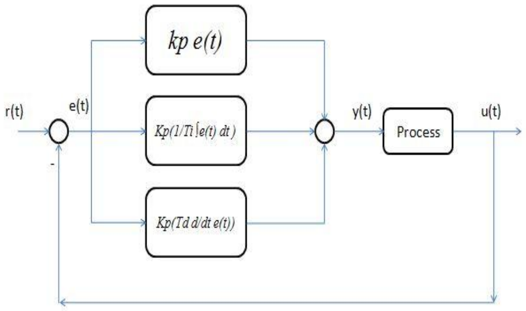


Fig.1 The Basic PID Controller

A. *Discretization of the Controller*

Generally the equation of PID controller is reduced to compatible format in order to implement it in the FPGA. To discretize the PID general equation, Backward Difference Method is accosted.

The obtained equation after backward difference method of discretization was,

$U(t) = u(t-1) + K1*e(t) - K2*e(t-1) + K3*e(t-2)$ ....(2)

Where,

K1=Proportional constant(K_p) + integral constant(K_i) + derivative Constant(K_d)
*K2=Proportional constant(K_p) + 2*derivative Constant(K_d) K3=Derivative Constant(K_d)*

III. TRANSFER FUNCTION

A real time example of stirred tank heater is selected, as shown in Fig.2,

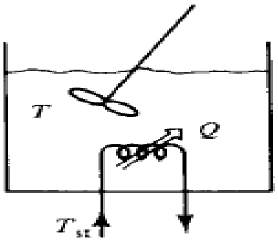


Fig.2 The Stirred Tank Heater

The whole system yields,

$$V\rho C_p\left(\frac{dT}{dt}\right) = Q = UA_t(\,T_{st} - T) \quad ....(3)$$

Where,

V = Volume of liquid in the tank.
$V\rho Cp$ = Liquid density and heat capacity.
U = Overall heat transfer coefficient between steam and liquid.
At = Total heat area transfer.
T_{st} = Temperature of the saturated heat.

After taking the Laplace transform of the system equation with consideration of steady state the transfer function first order equation obtained was,

$$G(s) = \frac{T'(s)}{T_{st}'(s)} = \frac{1}{\frac{V\rho C_p}{UA_t}s+1} \quad ....(4)$$

Since the first order process's transfer function is derived as follows,

$$G(s) = \frac{y(s)}{f(s)} = \frac{K}{\tau_p s + 1} \quad ....(5)$$

Where, τ_p is defined as the time constant of the process while the K is referred as steady state gain of the process.

Hence, the obtained value was,

$K = 1$ (Steady State Gain)
$\tau_p = \frac{V\rho C_p}{UA_t}$ (Time constant of the process)

This K value using Ziegler-Nichols frequency domain (ZNFD) method is cranked up putting T_i *and* T_d to null till the system starts oscillating. The point from which K_p, K_i and K_d value is easily calculated using Ziegler-Nichols table.

A. PID Constant Calculation

By the standard value of those parameters Steady state gain, K and time constant, T_p, was calculated and thereby using Ziegler-Nichols Frequency Domain method, real time value of K_p, K_i and K_d was obtained,

K_p =0.6 (Proportional constant)
K_i =171.4285 (Integral Constant)
K_d =0.000525 (Derivative Constant)

Using these constant values K1, K2 and K3 were calculated and utilized in the implementation of PID Controller.

IV. PIPELINING OF PID CONTROLLER

Pipelining of PID Controller remains the soul constituent of this paper. Since from the beginning of implementation of PID in digital controller which propagated through umpteenth phases of evolution. So at every stage an increasing demands of less computational time also incremented substantially. Hence the concept of pipelining and parallel processing was brought into the frame work. Fig 2 represents the schematic representation of equation (2).

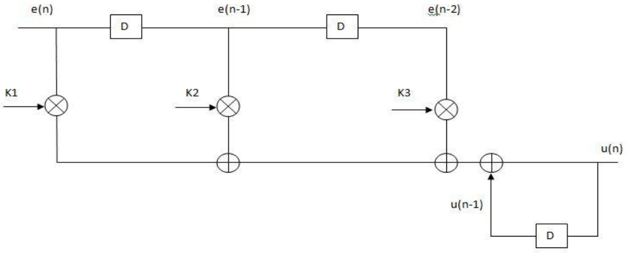


Fig.3Discretized Schematic of PID Controller

As in this paper, pipelining of PID controller reduced the computational time by some measurable value as compared to normal implementation of PID controller. Pipelined implementation involves introduction of latches, hence reducing critical timing from *Tm+2Ta* to *Tm+Ta*. Latches introduction creates delay element, so the equation (2) becomes,

$U(t)= u(t-1)+K1*e(t-1)-K2*e(t-2)+K3*e(t-3)$ ………(6)

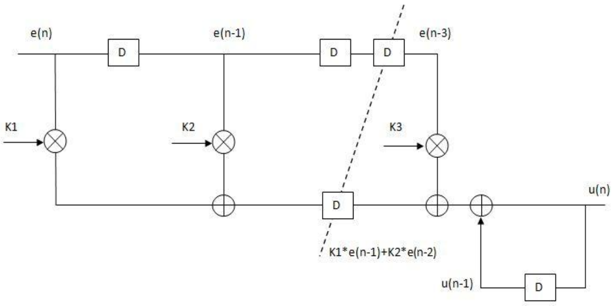


Fig.4 PID Controller Pipelined by the introduction of latches

Fig.4 represents pipelined PID controller by slicing the latch element. During the implementation the utilized number of parameters of FPGA including Slices, Flip flops, number of LUTs, Input Output Blocks, etc, were decreased reasonably while pipelining the same process. Table. I shown below represent the design summary of normal PID Controller. And Table. IIIII represents the pipelined PID Controller.

TABLE IV NORMAL PID CONTROLLER

Normal PID Controller			
Logic Utilization	**Used**	**Available**	**Utilization**
Slices	17	2448	0%
Flip-Flop	18	4896	0%
4 input LUTs	21	4896	0%
Bonded IOBs	18	108	16%
MULT18X18SIOs	02	12	16%

GCLKS	01	24	4%

TABLE VI
PIPELINED PID CONTROLLER

Pipelined PID Controller			
Logic Utilization	**Used**	**Available**	**Utilization**
Slices	3	2448	0%
Flip-Flop	6	4896	0%
4 input LUTs	3	4896	0%
Bonded IOBs	15	108	13%
GCLKS	1	24	4%

V. SIMULATION AND OUTPUT

A. Implementation using Xilinx FPGA

After the application of the transfer function of the process into the system, Implementation was carried out in Xilinx ISE 12.2 using Verilog language. The set up included the Spartan 3E family, XC3S250E Device, Package TQ144 and with the grading speed of -5. Since the implementation of Digital PID Controller was carried out in Register transfer level (RTL), the obtained RTL of the implemented work is shown in Fig.5.

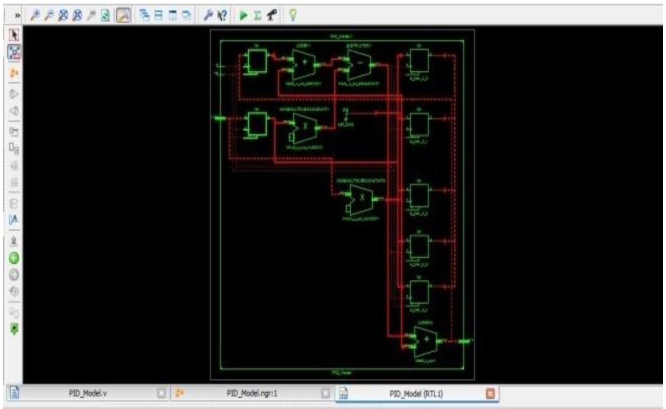
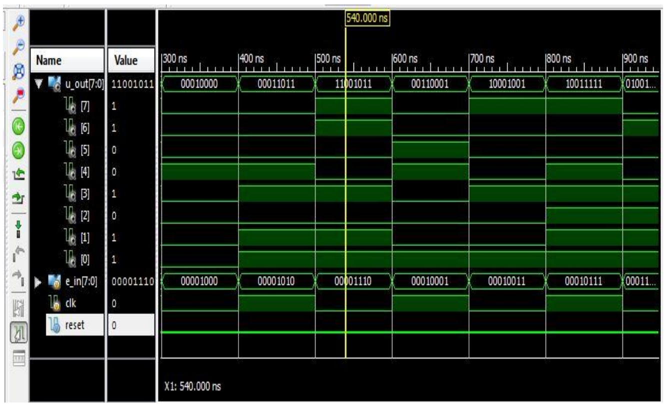


Fig.5 RTL schematic of PID controller

B. Implementation Result and Simulation

Since the efficient implementation using Verilog code was devised and the flow and behaviour of the code was verified using ISim (M.63c) Simulator. The simulator Result is shown in Fig.6



Name	Value
u_out[7:0]	11001011
[7]	1
[6]	1
[5]	0
[4]	0
[3]	1
[2]	0
[1]	1
[0]	1
e_in[7:0]	00001110
clk	0
reset	0

Fig.6 Result obtained after the PID Controller's simulation.

VI. CONCLUSION

FPGA premises the basic platform for implementing proposed Digital Pipelined PID Controller, in this paper, giving the categorically optimized result of implementation in terms of various parameters, which is then compared to the normal PID Controller, such as multipliers, Flip-Flops, slices, LUTs, GCLKS, etc. Further, the whole process can also be more optimized by adopting the low power strategy. Hence morphing the overall system into speed and power effectiveness.

REFERENCES

[1] Ramesh Nema, Rajeev Thakur and Ruchi Gupta, *Design and implementation of FPGA based on PID Controller with motor and sensor*, international journal of emerging science and engineering, Jan 2013.

[2] Sreenivasappa B.V. and Dr. Udaykumar R.V, *Analysis and Implementation of discrete time PID controller Using FPGA*, international journal of emerging science and engineering (IJESE), Nov 2012.

[3] Dr. Varodom Toochinda, *Digital PID controllers*, RDiPT, Kasetsart University, June 2011.

[4] Finn Haugen, *Discretization of simulator, filter, and PID Controller*, TechTeach, 10'May 2012.

[5] Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*, Second Edition.

[6] Mohammad Shahrokhi and Alireza Zomorrodi *Comparison of PID Controller Tuning Methods*, Sharif University of Technology.

[7] George Stephanopoulos, *Chemical Process Control*,

[8] Michael D. Ciletti, *Advanced Digital Design With the Verilog HDL.*

[9] Keshab K. Parhi, *VLSI digital signal processing*

[10] Akshay Sharma, Katherine Compton, Carl Ebeling and Scott Hauck, *Exploration of Pipelined FPGA Interconnect Structures*, Techeitech, 10'May 2012.