



## EFFICIENT FPGA IMPLEMENTATION OF 2<sup>ND</sup> ORDER DIGITAL CONTROLLERS USING MATLAB/SIMULINK

Vikas Gupta<sup>1</sup>, K. Khare<sup>2</sup> and R. P. Singh<sup>2</sup>

<sup>1</sup>Department of Electronics and Telecommunication, Vidyavardhani's College of Engineering and Technology, Vasai, University of Mumbai, India

<sup>2</sup>Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal, India  
E-Mail: [profvikasgupta@rediffmail.com](mailto:profvikasgupta@rediffmail.com)

### ABSTRACT

This paper explains a method for the design and implementation of digital controller based on Field Programmable Gate Array (FPGA) device. It is more compact, power efficient and provides high speed capabilities as compared to software based PID controllers. The proposed method is based on implementation of Digital controller as digital filters using DSP architectures. The PID controller is designed using MATLAB and Simulink to generate a set of coefficients associated with the desired controller characteristics. The controller coefficients are then included in VHDL that implements the PID controller on to FPGA. MATLAB program is used to design PID controller to calculate and plot the time response of the control system. The synthesis report concludes the resource utilization of selected FPGA.

**Keywords:** digital PID controller, FPGA, digital filter architecture, MATLAB, Simulink, Xilinx ISE, Spartan3.

### 1. INTRODUCTION

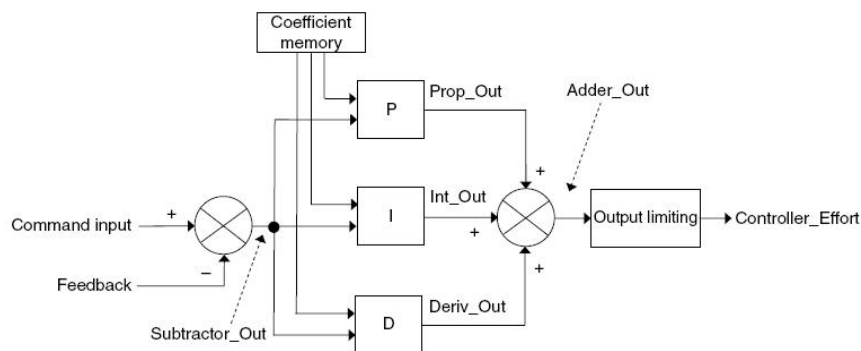
A control system is compared of two subsystems, a plant and a controller. The plant is an entity controlled by the controller. The controller can be either analog or digital. Generally, an implementation of digital PID controller includes the use of microprocessors or microcontrollers. The memory holds the application program while the processor fetches, decodes, and executes the program instructions. This method has a disadvantage in speed of operations because the operations depend on software which has a sequence of instructions and commands which needs many machine cycles to execute. Therefore, FPGA-based digital PID controller is proposed because the operations on FPGA are hardware compatible operations. However, FPGA-based digital PID controller still needs multipliers for computation. These

multipliers will decrease the speed of processing time because the multiplying stage is a consumption process which introduces propagation delay and uses large part of silicon area. These multiplications are change to Distributed Arithmetic (DA) architecture which was first proposed by Peled and Liu in 1974 [1].

Today's high-speed and high-density FPGA's provide practical design alternatives to ASIC and microprocessor-based implementations.

### 2. DIGITAL PID CONTROLLER

One of the most powerful but complex controller mode operations combines the proportional, integral, and derivative modes with a control loop feedback mechanism widely used in industrial control system.



**Figure-1.** Digital PID controller [2].

The proportional, integral, and derivative control actions can be brought together to create a PID controller. Figure-1 shows an example of how this can be created. As the design increases in complexity, the need for more additions or subtractions and multiplications or divisions increases. This highlights the need to develop an architecture that uses hardware proficiently and can

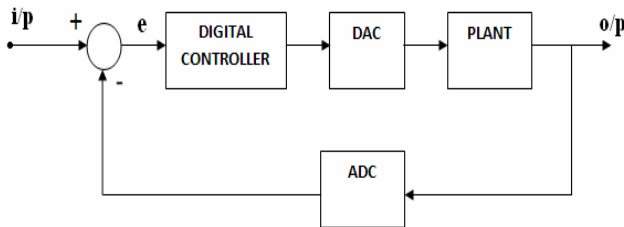
operate within the time constraints of the design. The arithmetic operations which are to be undertaken can be designed to be either separate actions (each action requiring its own dedicated hardware) or can be shared i.e., each addition, subtraction, multiplication, or division has a single common block, as is typical in the architecture of an arithmetic and logic unit, (ALU). Thus, in order to



design digital PID controller we use the above architecture as the base which will improve hardware efficiency [3].

**3. PRINCIPLE OF OPERATION**

Consider the block diagram of digital control system as shown in Figure-2.



**Figure-2.** Block diagram of digital control system.

The analytical equation is:

$$P = K_p e + K_I \int e dt + K_D (de/dt) + P_I(0) \tag{1}$$

Where,

$K_p$  = proportional gain

$K_D$  = derivative gain

$e$  = error in % of full scale range

$K_I$  = integral gain

$P_I(0)$  = value of integral term at  $t=0$

Taking Laplace transform of equation (1) will result in,

$$P(S) = K_p E(S) + \frac{K_I}{S} E(S) + K_D S E(S) \tag{2}$$

Also the transfer function of PID controller is

$$D(S) = K_p + \frac{K_I}{S} + K_D S \tag{3}$$

Where,

$D(S)$  is transfer of PID controller.

Transforming equation (3) into digital domain gives the transfer function of digital PID controller.

$$D(z) = K_p + K_I \frac{T(z+1)}{2(z-1)} + \frac{K_D(z-1)}{Tz} \tag{4}$$

Equation (4) can be realized to direct form structure as:

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{5}$$

Normally for digital controller  $b_2 = 0$  and  $b_1 = -1$

Therefore, coefficients  $a_0, a_1$  and  $a_2$  can be given as:

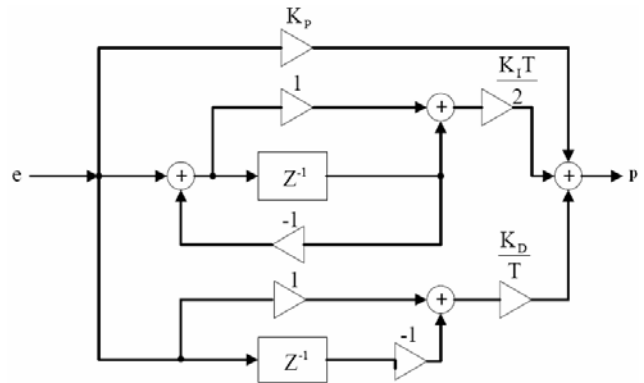
$$a_0 = K_p + K_I \frac{T}{2} + \frac{K_D}{T}$$

$$a_1 = K_p + K_I \frac{T}{2} - 2 \frac{K_D}{T}$$

$$a_2 = \frac{K_D}{T}$$

Where,

$K_p, K_I$  and  $K_D$  are proportional, integral and derivative parameters, respectively of digital PID controller and  $T$  is sampling period. Figure-3 shows the direct form I structure of digital PID controller corresponding to equation (5).



**Figure-3.** Direct Form I structure of digital PID controller.

**4. FPGA IMPLEMENTATION**

System Generator implements the design by considering the correct hardware platform and also takes care of the synchronization and interfacing problems [3].

A separate test bench application for hardware (FPGA) verification is also not required. The co-simulation block can be used with the same Simulink test bench apparatuses that were used to test the original System Generator model. Along- disadvantages also, that are associated with the presented co simulation methodology/tools using automatic bit stream generation. With every release of System Generator, the top level output files change.

Figure-4 illustrates the design flow adopted for the design of given PID controller.

The Open Loop transfer function of a second order antenna control system is taken as  $H(s)$  is given by  $H(s)=0.2083/(s^2+1.71s)$

The PID parameters for this required output speed are shown in Figure-5(a).

The system generator is a Xilins tool box available with MATLAB.

The blocks used for designs are as shown in the Figure-5(c).

Then the PID controller is implemented in Xilinx Sys Gen and Simulink as shown in Figure-5(c).

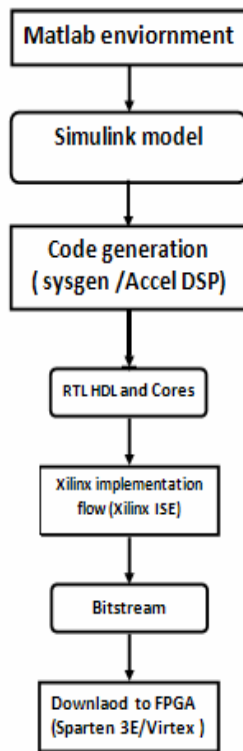


Figure-4. Design methodology [3].

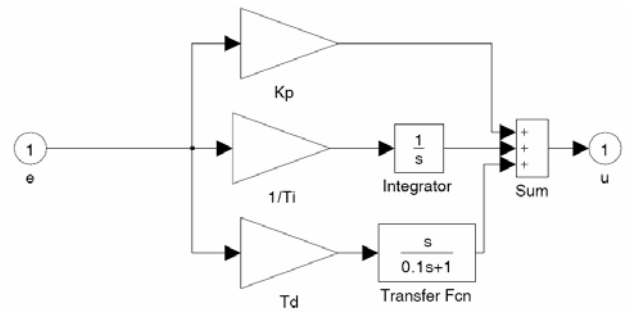


Figure-5(a). Simulink implementation of PID controller.

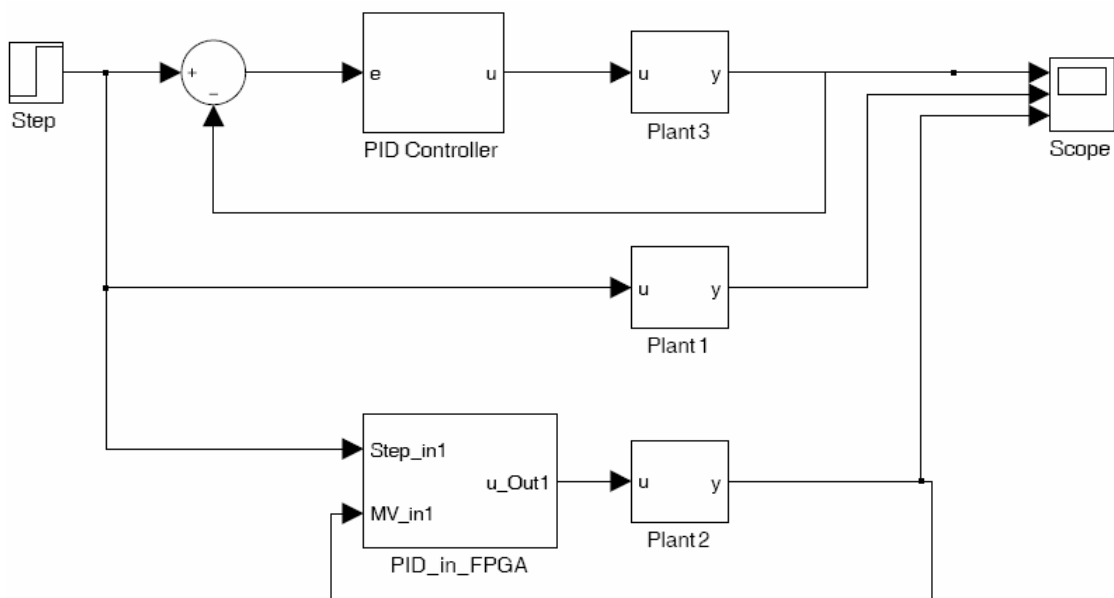


Figure-5(b). Simulink implementation of controlled system.

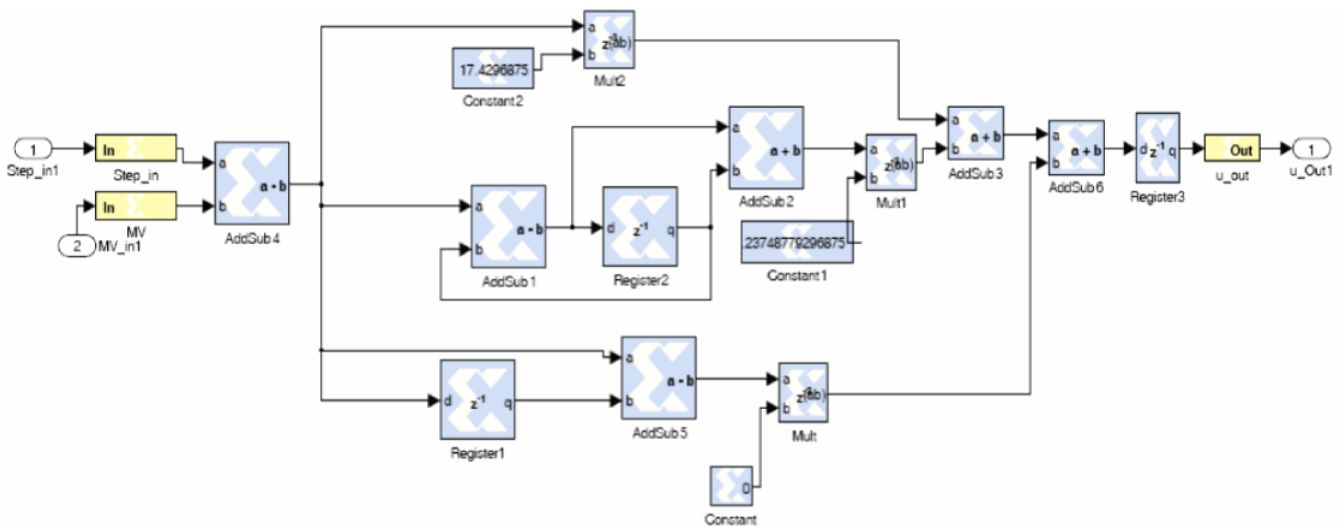
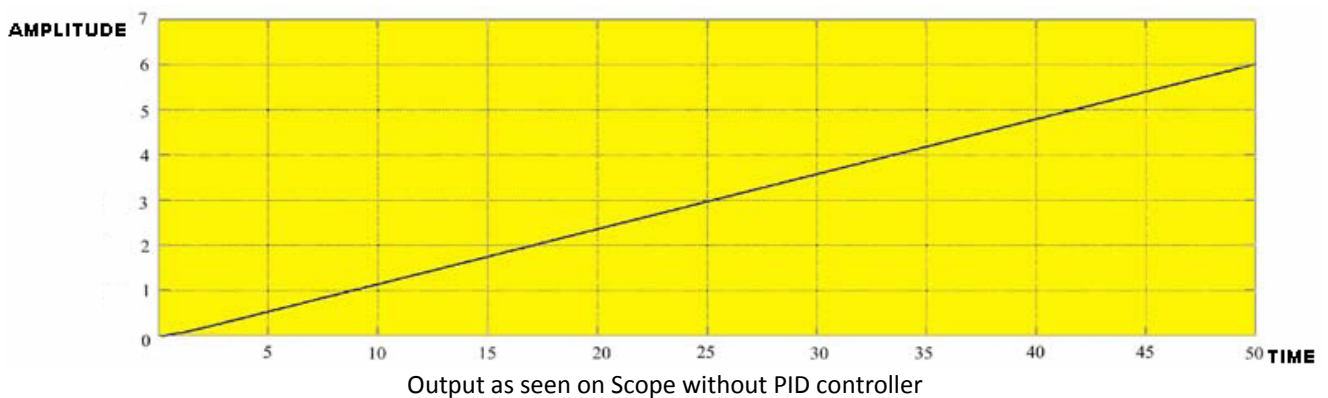
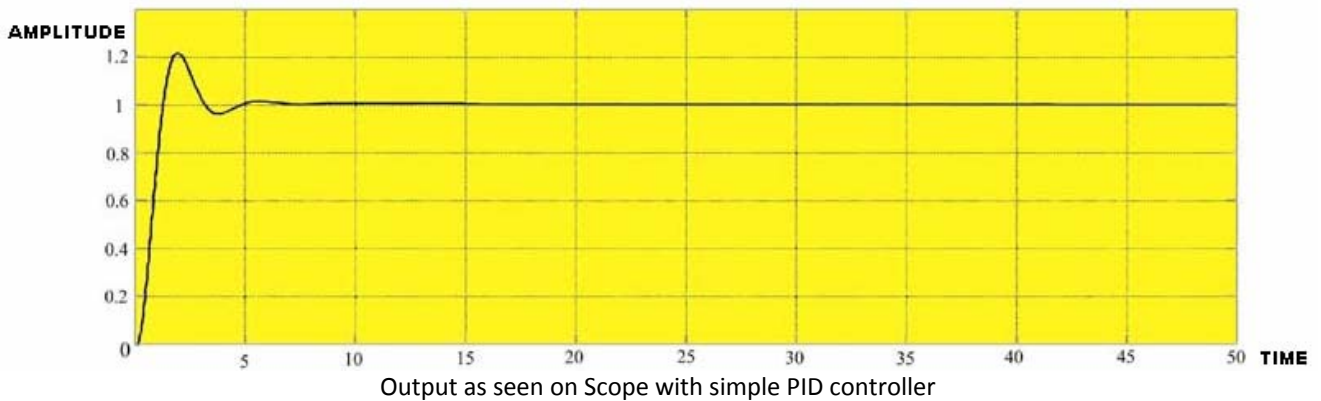


Figure-5(c). SysGen implementation of PID controller.

5. RESULTS

Figure-6 shows the simulation results of FPGA based digital PID controller with multiplier. Figure-6 gives

the comparison between the outputs of simple PID controller, FPGA based digital PID controller. It also gives the synthesis report of FPGA based digital PID controller.





Output as seen on Scope with FPGA based digital PID

**Figure-6.** Output comparison of plant controlled variable.

The synthesis using Xilinx ISE tools resulted in the following synthesis report for a Spartan3 as target FPGA. Here we can see that the overshoot (1.21) in the output of the plant with simple PID controller is higher than the overshoot (1.05) in output of plant with the FPGA based digital PID controller. This could be a very big

advantage. Though the rise time in the output with simple PID controller is better than the rise time with DF-I PID controller but this can be easily improved by increasing the value of  $K_i$ .

The device utilization summary is shown in Table-1.

**Table-1.** Direct Form-I multiplier based digital PID controller.

<b>Device Utilization Summary</b>			
<b>Logic utilization</b>	<b>Used</b>	<b>Available</b>	<b>Utilization</b>
Number of slice flip flops	240	7,168	3%
Number of 4 input LUTs	236	7,168	3%
<b>Logic distribution</b>			
Number of occupied slices	218	3,584	6%
Number of slices containing only related logic	218	218	100%
Number of slices containing unrelated logic	0	218	0%
<b>Total Number of 4 input LUTs</b>	<b>278</b>	<b>7,168</b>	<b>3%</b>
Number used as logic	236		
Number used as a route-thru	42		
Number of bonded IOBs	49	141	34%
Number of MULT18X18s	3	16	18%
Number of BUFGMUXs	1	8	12%
Number of RPM macros	6		

## 6. CONCLUSIONS

Implementing the digital PID controller on FPGA gives better rise time as well as settling time as seen in the results. In designing and implementing the digital PID controller one major thing which affects the performance of controller and its effects on plant is the effective hardware utilization. Also implementing PID controller on FPGA features speed, accuracy, power, compactness, and

cost improvement. In future work we plan to investigate implementation of multiplierless digital PID controller using distributed arithmetic architecture. The advantages are high processing speed, reduced power consumption and hardware compatibility for implementing on FPGA.

**REFERENCES**

- [1] A. peled and B. Liu. 1974. A new hardware realization of digital filters. IEEE Trans. ASSP. ASSP-22: 456-462.
- [2] Ian Grout. Digital systems design with FPGAs and CPLDs. 2008. Elsevier Newnes Press Publication. pp. 661-674.
- [3] Franklin G.F., J.D. Powell and M.L., Workman. 1990. Digital Control of Dynamic System. MA: Addison-Wesley Publishing Company.