Journal of VLSI Design Tools and Technology
Volume 2, Issue (1-3), Compilation 2012

An FPGA-based Controller Design for Servo Actuator Using Xilinx System Generator and HDL Cosimulator

T. Ananthan[1], **M. V. Vaidyan**[2], **M. V. Varghese**[3]*
[1]Research scholar, Electrical and Electronics Engineering, NIT-Calicut, Kerala, India
[2]Professor, Electrical and Electronics Engineering, NIT-Calicut, Kerala, India
[3]Trainee, Kerala State Electricity Board, Calicut, Kerala, India

ABSTRACT

This paper aims at designing a Field programmable gate array (FPGA)-based proportional, integral derivative (PID) controller in a servo-actuated control system. In the conventional method of implementing control systems in FPGA, there is no interaction between the controller and the system except for the final stage of implementing control system in the hardware. The proposed methods provide approaches for analyzing the performance of the controller and system in MATLAB simulink environment prior to the implementation in FPGA. This results in a faster and efficient hardware implementation of digital control system. Two methods developed and used in designing the controller and corresponding simulation results are presented. In addition, a comparative evaluation of the two methods is also discussed.

***Keywords**: FPGA design, PID controller, servo actuator, HDL cosimulation*

****Author for Correspondence*** Email: ananthan_pee10@nitc.ac.in, Tel: + 91-8089976325

1. INTRODUCTION

Proportional, Integral Derivative (PID) controllers have been widely used in all industries due to their simple architecture. PID controllers have gone through several stages of evolution.

Recently FPGAs have become an alternative solution for the realization of digital control systems, which were previously dominated by general purpose microprocessor and microcontroller systems [1–3]. The advancements in very large scale integration (VLSI) and electronic design automation (EDA) have created an opportunity for control system design engineers to develop complex and compact controllers. The design can be verified quickly without committing to the hardware using hardware/software simulation platforms to evaluate its performance with high confidence.

The conventional implementation of control system design in FPGA consists of the following stages:

(i) Simulation of system model in simulink
(ii) Controller design in VHDL or Verilog
(iii) Implementation in hardware and system performance analysis.

The demerits in the above approach are:

1. Need to write extra HDL (hardware descriptive language) code (test bench) to generate the stimuli for simulation of control algorithm. This is difficult for more complex systems.

2. During the simulation, the controller has no interaction with system model to be controlled, except for the final stage, when the control system is implemented in the hardware.
3. Any fault can be spotted only after the implementation in FPGA, which results in loss of time and cost.

To overcome these demerits, a design environment is presented in this paper in which the FPGA controller can interact with system model to evaluate the performance of the control system design before real-time implementation.

Two approaches are described in this paper for the design of FPGA-based PID controller [4, 5] for a servo actuator system. In the first approach, Xilinx system generator/Matlab simulink is used and in the second approach HDL cosimulation with ModelSim HDL simulator is used.

The organization of this paper is as follows. In section 2, the overview of VLSI design flow and tools used is discussed. Section 3 presents the PID control algorithm.

The simulation results of FPGA-based PID controller for servo actuator using Xilinx system generator is presented in section 4. In section 5, the simulation result for servo actuator using Modelsim cosimulation is presented. Comparison of the two methods discussed in section 6 and section 7 gives the conclusions and future work.

2. VLSI DESIGN FLOW AND TOOLS

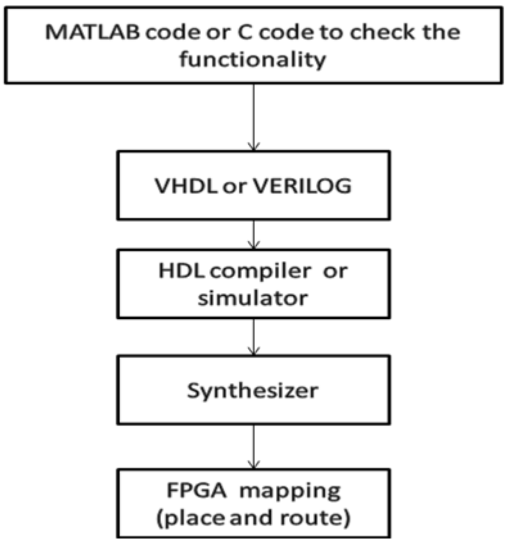

MATLAB code or C code to check the functionality
VHDL or VERILOG
HDL compiler or simulator
Synthesizer
FPGA mapping (place and route)

Fig. 1: *VLSI Design Flow.*

Figure 1 shows the VLSI design flow. Firstly, any developed algorithm has to be checked for its functionality using Matlab or C code. Then the designer has to write HDL code using VHDL or Verilog and complied for its correctness. In this stage, various HDL simulators are used like modelsim, active HDL, etc. But Model sim is commonly used in all VLSI industries. One of the popular synthesis tool is Synplify pro by Synopsis for all FPGA makes. Finally, place and route is a vendor-based tool. Xilinx and Altera are the leaders in FPGA products and they are producing FPGA in various series depending upon the gate density and speed requirement. In addition, Quartus II is a software tool produced by Altera for analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis,

and examine RTL diagrams. Likewise Xilinx ISE from Xilinx

3. PID CONTROL ALGORITHM

The ideal continuous time PID controller [6] can be expressed as:

$$u(t) = K_p\left[e(t) + \frac{1}{T_i}\int_0^t e(t)dt + \frac{de(t)}{dt}\right] \qquad (1)$$

where $u(t)$ is the control input, K_p is the proportional gain, T_i is the integral time, T_d is the derivative time and e is the error between the reference value and the actual output. The S -domain expression of the corresponding PID controller can be given as:

$$U(s) = \left[K_p + \frac{K_i}{s} + K_d s\right]E(s) \qquad (2)$$

where K_p, K_i and K_d are the proportional, integral and the derivative gains of the controller respectively. For the digital PID control with sampling time T_s, the following digital PID control algorithm can be obtained by replacing the derivative term and the integral term with a backward difference function and a sum using rectangular integration respectively. The difference equation is given by:

$$u(n) = K_p\left\{e(n) + \frac{T_s}{T_i}\sum_{j=0}^{n} e(j) + \frac{T_d}{T_s}[e(n) - e(n-1)]\right\} \qquad (3)$$

where indices *n* and *j* refer to the time instant. The digital PID block is further simplified as: $u(n) = K_p.e(n) + K_i\sum_{j=0}^{n} e(j) + K_d[e(n) - e(n-1)]$ (4)

where $K_i = K_p\frac{T_s}{T_i}$ is the digital integral coefficient, $K_d = K_p\frac{T_d}{T_s}$ is the digital derivative coefficient.

4. PID CONTROLLER USING XILINX SYSTEM GENERATOR FOR SERVO ACTUATOR

Xilinx system generator: It is a DSP design tool from Xilinx that enables the use of the mathworks model-based design environment simulink for FPGA design. Previous Xilinx FPGAs or register transfer level (RTL) design methodologies are not required when using system generator. Design is captured in the DSP-friendly simulink modeling environment using Xilinx-specific block sets as shown in Figure 2.

Servo actuator: Servo actuators [7] are used in many control system applications as a final decision element. Here a DC servo actuator is considered whose control system can be described by a third-order differential equation:

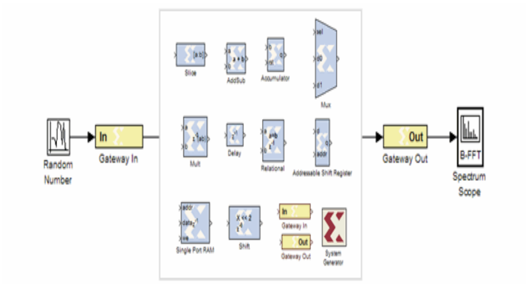


Fig .2: *Simulink Modeling Environment Using Xilinx DSP Block Sets* [8].

$$\frac{d^3y}{dt^3} + a_2\frac{d^2y}{dt^2} + a_1\frac{dy}{dt} + a_o y = b_0 u(t) \tag{5}$$

After deriving with parameter values, the transfer function of the system is:

$$G(s) = \frac{1}{s^3 + 5s^2 + 8s + 4} \tag{6}$$

An FPGA PID controller was designed for the servo actuator using system generator DSP block set and simulated in Matlab simulink environment.

The gain values were tuned using Zeigler and Nichols method. The block diagram of Matlab simulink – system generator and simulation results – is shown in Figure 3 and Figure 4 respectively. XILINX ISE 12.2 and MATLAB 7.9(R2009b) are used for simulationn.

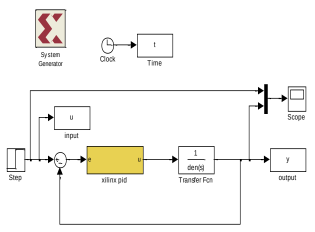
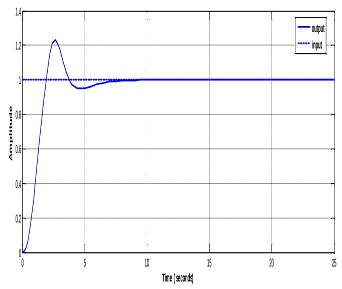


Fig. 3: *Xilinx PID Controller and Simulink Block Diagram*



.

Fig. 4: *Simulation Result – Unit Step Response of Servo Actuator Using Xilinx PID Controller.*

5. PID CONTROLLER USING MATLAB SIMULINK-MODELSIM COSIMULATION.

5.1. Model Sim

Modelsim is a verification and simulation tool for VHDL and Verilog from Mentor Graphics Company. Here Modelsim SE 6.3f is used for simulation.

5.2. EDA Simulator Link

The Matlab simulink consists of electronic design automation (EDA) simulator link. This EDA simulator link software consists of Matlab functions that establish communication links between the HDL simulator, Matlab and library of simulink blocks that include HDL simulator designs in simulink models for cosimulation. By using EDA simulator link, it is possible to verify a Verilog or VHDL design against the simulink model using cosimulation with an HDL simulator Modelsim.

Once the HDL simulator is linked with the simulink, the HDL simulator functions as a client as shown in Figure 5.

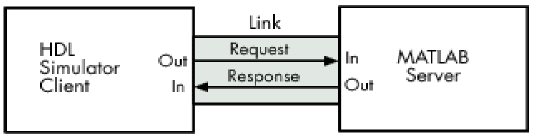


Fig. 5: *Linking with Matlab and HDL Simulator* [9].

In this scenario, a Matlab server function waits for service requests that it receives from an HDL simulator session. After receiving a request, the server establishes a communication link and invokes a specified Matlab function that computes data for the HDL module (coded in Verilog or VHDL) that is under simulation in the HDL simulator.

A servo actuator system is designed here and Figure 6 shows the simulink diagram with Modelsim HDL simulator.

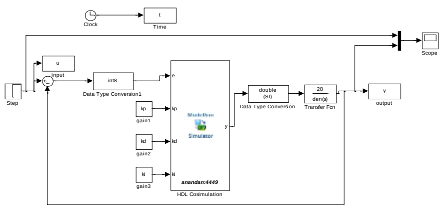


Fig. 6: *Matlab Simulink and Modelsim Cosimulation Environment.*

The simulation result shown in Figure 7 shows that in the response there is a small delay, but this delay can be ignored because a timescale setting was done between Matlab simulation time and HDL simulation time, in which one second in Matlab simulation corresponds to one millisecond in HDL simulation. Here, FPGA PID controller was designed using Verilog.

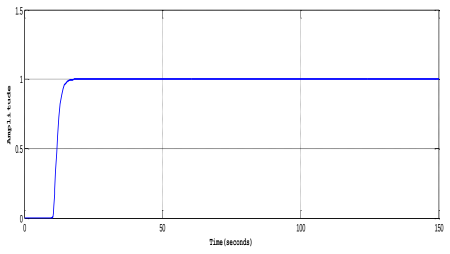


Fig. 7: *Simulation Result – Unit Step Response of Servo Actuator Using HDL Cosimulation. Delay Can Be Ignored Because of Time Scale Settings.*

6. COMPARATIVE EVALUATION OF TWO METHODS

In the first method using Xilinx system generator, a designer can verify the hardware output in MATLAB simulink environment and target a Xilinx FPGA. Prior knowledge of Xilinx FPGA and RTL design methodologies is not required for the designer.

It is a vendor-based tool and is applicable only to Xilinx FPGA but not for all makes. In the second method, the Modelsim HDL cosimulation also works in MATLAB simulink environment but the design engineer should have the knowledge of HDL to design in VHDL or Verilog and FPGA design methodologies. This design is not vendor-based. Hence, a designer has the freedom to target any FPGA make like Xilinx, Altera, etc. For any developed algorithm to be implemented in FPGA, the designer has to develop the corresponding FPGA-based architecture.

The design process in developing the FPGA-based architecture can be made easy using dedicated block sets proposed by FPGA manufacturers like Xilinx system generator. However, FPGA implementation of complex algorithms, using block sets proposed by FPGA manufacturers, may lead to unoptimized architecture that may be inadequate to the available resources [2]. Hence a designer has to develop and code the FPGA-based architecture.

7. CONCLUSIONS

This paper presents the FPGA-based PID controller design for a servo actuated control system. Two simulation platforms were proposed to verify the controller and system performance. In these methods the FPGA controller and system model can be interacted using Xilinx system generator and Modelsim HDL cosimulation. This idea enables a control system designer to make a complete analysis of the FPGA controller and system performance prior to hardware implementation. The comparison of two methods gives a clear idea about the selection of design methods. A future study is to develop an efficient FPGA architecture for complex control algorithms and online applications with fast execution time.

REFERENCES

1. Y. F. Chan, M. Moallem and W. Wang. *IEEE transactions on Industrial Electronics.* August 2007. 54(4). 1898–1905p.
2. Eric Manmasson, Lahoucine Idkhajine. *IEEE Transactions on Industrial Information.* May 2011. 7(3). 224 p.
3. FPGA-based controllers – Different Perspectives of Power Electronics and Drive Applications. *IEEE Industrial Electronics Magazine.* Mar 2011.
4. J. Lima, R. Menotti, J. M. P. Cardoso, et al. *Proceedings of IEEE International Conference on System, Management, and Cybernetics*, Taipei, Taiwan. Oct. 2006. 2577–2583p.
5. Feng Li, Tingeun Wei, Ran Zheng, et al. *International Conference on Computer, Mechtronics, Control and Electronic Engineering.* 2010. 273–376p.
6. Chander, S, Agarwal, P., Gupta,I. *Power Electronics, Drives and Energy Systems (PEDES) & 2010 Power India Joint International Conference.* 1–6p.
7. Eronini Umez. *System Dynamics and Control.* Thomson Books. 2002.
8. *System Generator for DSP – Getting Started Guide, Xilinx. 2010.*
9. *www.mathworks.com*