International Journal of Recent Trends in Engineering, Vol 2, No. 6, November 2009

Efficient FPGA Design and Implementation of Digital PID Controllers in Simulink®

Prof. Vikas Gupta[1] Dr Kavita Khare [2], Dr R. P. Singh[3]

[1]VCET, Vasai ,INDIA, E-mail: profvikasgupta@rediffmail.com
[2 and 3] MANIT, Bhopal, INDIA, E-mail:kavitakhare1@yahoo.co.in

Abstract: This paper explains a method for the design and implementation of multiplierless digital PID 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 Distributed Arithmetic (DA) architecture. The PID controller is designed using MATLAB and Simulink to generate a set of coefficients associated with the desired controller characteristics. The architecture was implemented using SysGen to give flexibility and compatibility with the Simulink design of a controller. 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.

Keywords: **Multiplierless architecture, Distributed Arithmetic algorithm, Digital PID controller, FPGA, Matlab, Simulink, Xilinx SysGeneartor**

I. INTRODUCTION

A control system consists 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. Due to this the multiplierless PID controller can be realized [2]. The implementation involves forming look up tables for above DA based method and interface with the ROM in SysGen architecture. [1]

More flexible and compatible Sys Gen System Architecture Description for the given control applications achieved. Finally, system functionality Verification using FPGA and comparisons of implementation was done.

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

II. DIGITAL PID CONTROLLERS

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.

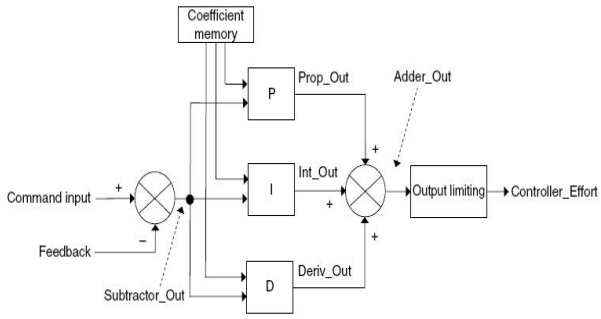

Fig.1 Digital PID controller [11]

The proportional, integral, and derivative control actions can be brought together to create a PID controller. Fig.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].

III. PRINCIPLE OF OPERATION

Consider the block diagram of digital control system as shown below.

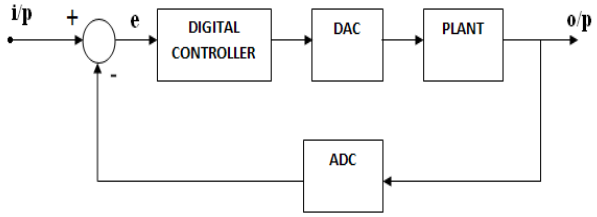

Fig.2 Block diagram of Digital Control System

The analytical equation is:

$P = K_p e + K_I \int e dt + K_D (de/dt) + P_I(0)$ (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,

$$P(S) = KpE(S) + \frac{K_I}{S} E(S) + K_D S\, E(S) \quad (2)$$

Also the transfer function of PID controller is

$$D(S) = Kp + \frac{K_I}{S} + K_D\, S \quad (3)$$

where,

D(S) is transfer of PID controller.

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

$$D(z) = Kp + K_I \frac{T}{2}\frac{(z+1)}{(z-1)} + \frac{K_D}{T}\frac{(z-1)}{z} \quad (4)$$

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

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

IV.DISTRIBUTED ARITHMETIC ARCHITECTURE

Distributed Arithmetic (DA) is an efficient technique for calculation of sum of products, vector dot product, inner product, and multiply and accumulate (MAC). Distributed arithmetic is an important algorithm for DSP applications. Distributed arithmetic is a bit level rearrangement of a multiply accumulate to hide the multiplications. It is a powerful technique for reducing the size of a hardware that is well suited to FPGA designs. It can be implemented on LUT "Look-up-table". DA architecture is used because of its extreme computational efficiency.

From equation (5),

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

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

Therefore, the transfer function becomes:

$$D(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - z^{-1}} \quad (6)$$

and the coefficients a_0, a_1 and a_2 can be given as

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

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

$$a_2 = \frac{K_D}{T} \quad \ldots\ldots\ldots\ldots\ldots \quad (7)$$

where K_P, K_I and K_D are proportional, integral and derivative parameters respectively of digital PID controller and T is sampling period.

Taking Inverse z- transform of equation (6) and deriving it into difference equation it will give:

$m(n) = a_0 e(n) + a_1 e(n-1) + a_2 e(n-2) + m(n-1)$..... (8)

All the possible values of m(n) are stored in ROM. There will be 16 different possible values of m(n) which is illustrated in the following table.

Table 1.Evaluation of ROM values

Address				Values in ROM
0	0	0	0	0
0	0	0	1	1
0	0	1	0	a_2
0	0	1	1	$a_2 + 1$
0	1	0	0	a_1
0	1	0	1	$a_1 + 1$
0	1	1	0	$a_1 + a_2$
0	1	1	1	$a_1 + a_2 + 1$
1	0	0	0	a_0
1	0	0	1	$a_0 + 1$
1	0	1	0	$a_0 + a_2$
1	0	1	1	$a_0 + a_2 + 1$
1	1	0	0	$a_0 + a_1$
1	1	0	1	$a_0 + a_1 + 1$
1	1	1	0	$a_0 + a_1 + a_2$
1	1	1	1	$a_0 + a_1 + a_2 + 1$

The size of ROM is very important for high speed implementation as well as area efficiency. ROM size grows exponentially with each added input address line. The proposed multiplierless based architecture is obtained using DA.[9]

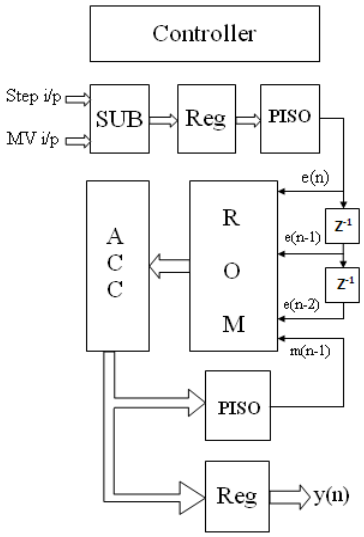

Fig.3. Architecture based on DA

IV.FPGA IMPLEMENTATION

System Generator implements the design by considering the appropriate hardware platform and also takes care of the synchronization and interfacing problems. Outputs from the digital controller are functions of current and past input samples, as well as past output samples - this can be implemented by storing relevant values of input and output in registers. The output can then be formed by a weighted sum of these stored values.[5]

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 apparatus 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. [3]

V. Design Example

The implementation of the controller in Simulink is as shown in figure 4(a).The step response of the plat can be observed with the scope block.

The multiplierless architecture is implemented.

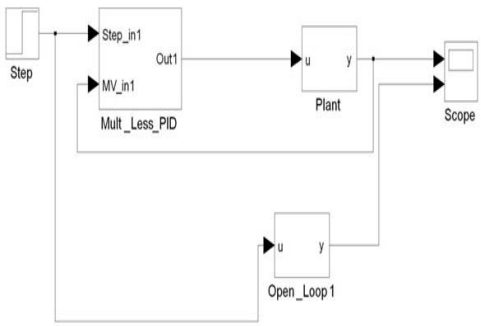

Fig. 4(a) Simulink block diagram for the controller with plant

The control unit for the architecture is implemented The look up table constants is interfaced in the ROM of the implementation.

The sampling time T_s=10kHz is selected for obtaining the desired step response.

After simulation we synthesize the design by using Xilinx ISE tool to obtain various results of Resource utilization, RTL_ level synthesis, maximum clock frequency etc.

The resource estimator in Sys gen is used to estimate the resource utilization before going for further synthesis.[7]

VI.RESULTS

Here in fig.5 are the simulation results of FPGA based multiplierless PID controller. Figure 5 gives the comparison between the step response of plant without PID controller and FPGA based Multiplierless PID controller. Both the responses are simulated on Scope block of Simulink.

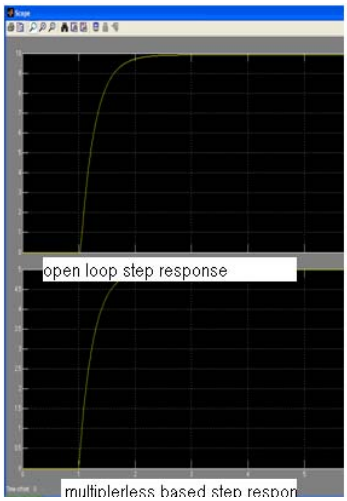

Fig. 5 Results of step response

The table below gives the synthesis report of FPGA based multiplierless PID controller. It provides the information regarding the number of slices, Flip-Flops, Input Output blocks required. The results show that the resource utilization is very less about 1% of the total slice flip flops which suggest the design is efficient.

The step response of figure 5 shows that the response remain almost same when simulated in Sys Gen.

Table2 . Estimator Report

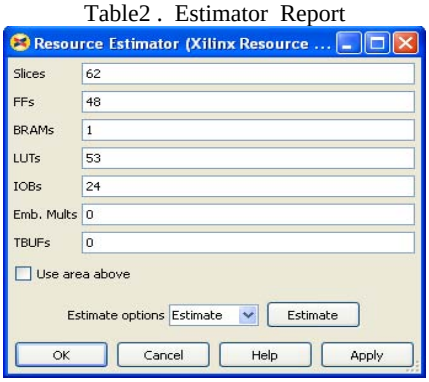

Table:3Device utilization summary

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	61	7,168	1%
Number of 4 input LUTs	59	7,168	1%
Number of Slices	48	3,584	1%
Number of Bonded IOBs	25	141	17%
Number of RAMB16s	1	16	6%
Number of BUFGMUXs	1	8	12%

The frequency of operation is 155.352MHz and the minimum period required is 6.437ns. Maximum path delay from/to any node: 3.958ns

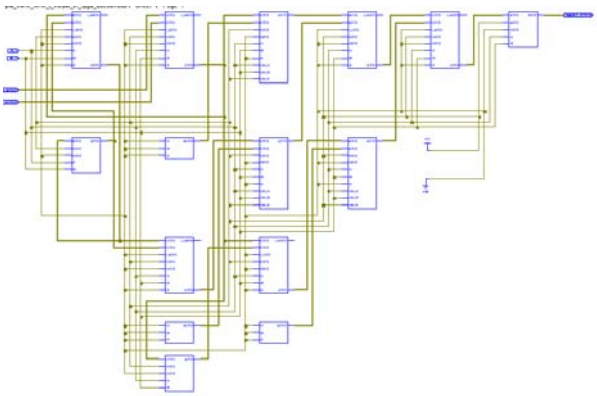

Fig 6(a) RTL_level synthesis of a simulation with Multipliers(DF_1 realization)

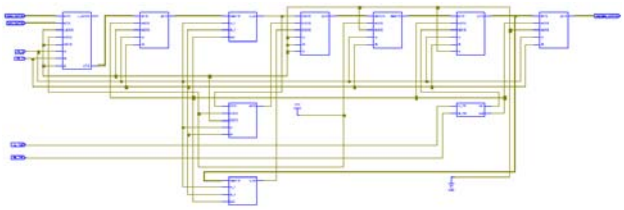

Fig 6(b) Muntiplierless RTL_ synthesis showing saving in resources compare to multiplier based implementation fig 6 (a)

VII.CONCLUSION

Implementing the multiplierless 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, many PID controllers can be implemented on a single FPGA chip including coefficient calculation and auto-tuning. Due to the use of Distributed Arithmetic algorithm, the number of PID controllers on single FPGA chip can be increased immensely. The advantages are high processing speed, reduced power consumption and hardware compatibility for implementing on FPGA.

REFERENCES

[1] Anthony Cataldo, “Low-priced FPGA options set to expand” Electronic Engineering Times Journal, N 1361, PP 38-45, USA 2005.

[2] A peled and B. Liu. “A new hardware realization of digital filters.”IEEE Trans. ASSP, vol. ASSP-22, pp.456-462, Dec 1974.

[3] Franklin, G.F., J.D. Powell and M.L., Workman 1990.”Digital Control of Dynamic Systems : Addison-Wesley Publishing Company.

[4] Joao Lima, Ricardo Menotti, Joao M. P. Cardoso, and Eduardo Marques ‘A Methodology to Design FPGA-based PID Controllers ‘. 8th Oct IEEE International Conference 2006.

[5] National Instruments: http://www.ni.com. FPGA based control: Millions of transistors at your command, 2004.

[6] Charles H Roth Jr. Digital System Design Using VHDL. Brooks/Cole, 1998.

[7] Diligent, Inc., “Diligent Spartan-3 System Board”, June,2004. BOARD-brochure.pdf

[8] FPGA-Based Multiplierless Digital PID Controller Using Distributed Arithmetic by Sorawat Chivapreech , Surapan Yimman, Chusit Pradabpet and Kobchai Dejha

[9] L. Samet, N. Masmoudi, M.W. Kharrat, and L. Kamoun, "A Digital PID Controller for Real Time and Multi Loop Control: a comparative study," in IEEE Int'l Conference on Electronics, Circuits and Systems (ICECS'98), Vol.1, Sep. 7-10, 1998, pp. 291-296.

[10] Kariyappa B. S., Hariprasad S. A., and R. Nagaraj ‘Position Control of an AC Servo Motor Using VHDL & FPGA’ Proceedings of World Academy of Science, Engineering and Technology ,Volume 37, January 2009 ,pp 1342-1345.

[11] Ian Grout ‘Digital Systems Design with FPGAs and CPLDs’, Elsevier (Newnes press) publications,2008, chapter-10 ,pp 661-700.