Design & Implementation of FPGA Based On PID Controller

Rajesh Nema, Rajeev Thakur, Ruchi Gupta

Abstract- ***Proportional-Integra-Derivative controllers are universal control structure and have widely used in Automation systems, they are usually implemented either in hardware using analog components or in software using Computer-based systems. In this paper, we focused our works designing on building a multi-channel PID controller by Field Programmable Gate Arrays (FPGAs). To overcome the hardware complexity by the use of more processors for multi channel, we are using single PID controller for multi channel .Multi channel can be implemented by the use of FPGA.when the error is more it can differentiate and produce the constant output, when signal is low when compared to reference signal it can integrate it.FPGA can offer parallel processing, more speed.***

Keywords: - FPGA, PID, PWM, SENSOR,

I. INTRODUCTION

The application range of FPGA based designs increases every day. This is mainly due to the flexibility and capability to perform parallel tasks. The industry is adopting massively the core-based design methodology for system integration using FPGAs, which leads to the appearance of the System-on-Programmable-Chip (SoPC) platforms [1]. This paper presents a PID core suitable to be introduced in such a system. Furthermore, the simulation scheme to successfully simulate the hardware and the motor involved is also presented. The simulation of the VHDL core is performed around the Xilinx System Generator [6]. The "System Generator" is a collection of Simulink block sets that permit interaction between hardware and modeled systems. In this paper we consider discrete time PID controller and is implemented in a dedicated FPGA with PWM modulator. In this paper, a case study is presented in

Which a modular FPGA-based design approach is applied to design a temperature control system. The same approach can be extended to design other embedded controllers using FPGA. The complete system is implemented by dividing system functions into reconfigurable modules. In general, embedded control designers need to go through three phases in the design of digital control systems: 1) software modeling/simulation in an environment such as Matlab/Simulink; 2) hardware implementation; and 3) co simulation of the whole system including both hardware and software. The organization of this paper is as follows: In section **II.** Related Work **III.** Different discrete time PID controllers are reviewed. Discuss about the PID controller and PWM modulator explained in section **IV**. The simulation and FPGA implementation results are discussed in section **V**&VI.

Manuscript received on January 22, 2013.

Rajesh Nema, Dept. E&C., R.G.P.V. / NRI institute of science of technology / Bhopal / India.

Rajeev Thakur, Dept. E&C., R.G.P.V. / NRI institute of science of technology / Bhopal / India.

Ruchi Gupta, Dept. E&C., R.G.P.V. / NRI institute of science of technology / Bhopal / India.

II. RELATED WORK

1. In this paper, an efficient design scheme for implementation of the Proportional-Integral Derivative (*PID*) controller using Field Programmable Gate Array (FPGA) technology is presented.
2. This paper presents a series of high speed and low-power finite-word-length PID controllers based on a new recursive multiplication algorithm.
3. In this paper analysis and implementation of Proportional Integral Derivative (PID) controller using Field Programmable Gate Array (FPGA) is presented.
4. The research presented in this article applies the newest Field-Programmable-Gate-Arrays to implement motor controller devices in accordance with the actual core-based design. The flexibility of the System on-a-Programmable-Chips in motor multi-axis control systems enables the processing of the most intensive computation operations by hardware (PID IP cores) and the trajectory computation by software in the same device.
5. In this paper, modular design of embedded feedback controllers using field-programmable gate array (FPGA) technology is studied. To this end, a novel distributed-arithmetic (DA)-based proportional–integral–derivative (PID) controller algorithm is proposed and integrated into a digital feedback control system.
6. This paper presents a PID controller core described in VHDL suitable to be introduced into a System-on-Programmable Chip design. The flexibility of the System-on-a-Programmable-Chips (SoPCs) in motor multi-axis control systems makes possible the processing of the most intensive computation operations by hardware (PID IP Cores) and the trajectory computation by software.

III. DISCRETIZATION PID CONTROLLER

The general form of PID controller given in most of the text book is the standard form.

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

where *p K* is the Proportional gain, *i T* is the Integral time, *d T* is the derivative time, *e*(*t*) is the error signal and *u*(*t*) is the output of the controller. The ideal parallel form of the PID controller shown in Fig. 1 is represented by a mathematical equation as The whole expression of PID algorithm is:

$u(t) = k_p\,[e(t) + 1/T_i \int e(t)\,dt + T_d\,d/dt\,e(t)]$ (1)

Where k_p is the proportional gain, T_i is the integral time constant, T_d is the differential time constant, u(t) is the control variable and e(t) is the system error.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Equation (1) can be transformed into a discrete-time description

$u(k)= u(k-1)+ k_p x_1(k)+ k_i x_2(k)+ k_d x_3(k)$

$x_1(k)= e(k)-e(k-1)$

$x_2(k)=e(k)$

$x_3(k)=e(k)-2 e(k-1)+ e(k-2)$

Where k_i is the integral coefficient and k_d is the differential coefficient. Hence we get.

$u(k)= u(k-1)+k_0 e(k)+k_1 e(k-1) +k_2 e(k-2)$

$k_0= k_p +k_i +k_d$

$k_1= -k_p-2k_d$

$K_2= k_d$

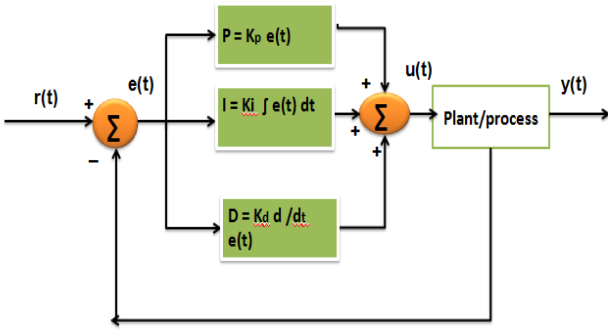


Figure 1: Block diagram of PID controller

This parallel form is shown in Fig 1, where the parameters are treated as simple gains, is the most general and flexible form. However, it is also the form where the parameters have the least physical interpretation and is generally reserved for theoretical treatment of the PID controller. The purpose of integral action is to increase the low frequency gain and thus steady state error reduces. The derivative action adds phase lead, which improves stability and increases system bandwidth. Fallowing [3], from a practical point of view, implementation of (2) has certain limitations. Firstly, actuator saturation can cause integrator windup, leading to sluggish transient response. Secondly, the pure differentiation term amplifies noise, leading to deterioration of the control command. Finally, the differentiation term acts on the error signal, taking the derivative of the command signal as well.

IV. PID CONTROLLER AND PWM MODULATOR

A. PID controller

The PID controller follows the classical structure. It contains two saturation Blocks, one for the integral part and the other for the overall sum (see figure 4). The controller has a pipeline structure of three stages, in other words, it needs three clock cycles to perform all the operations. In order to improve the area and speed, hardware multipliers have been used [9]. These multipliers are included in the Spartan 3 family of Xilinx and subsequent FPGAs. These multipliers have 15 bit input data bus and are signed. This leads to optimum implementation when the fixed point implementation uses less than 15 bit in two's complement.

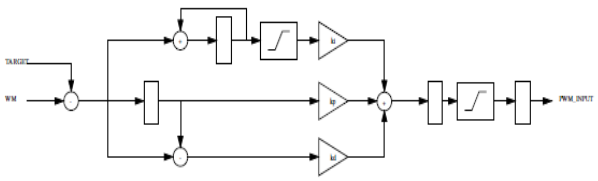


Fig. 4. Block diagram of the PID core.

B. PWM modulator

The PWM modulator admits a two's complement input and transforms it into a PWM signal. The module has two outputs, one the modulated PWM and the other one the sign of the modulation (see figure 5). These two outputs permit direct connection with existing Full-Bridge Motor Drivers [10].

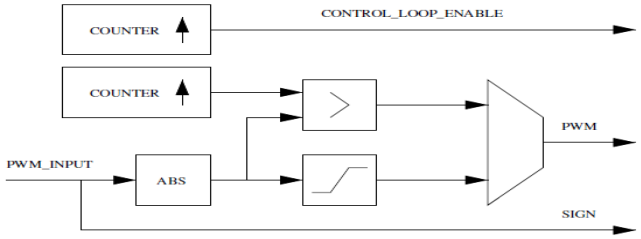


Fig. 5. Block diagram of the PWM modulator.

The PWM module also generates the enable signal for the control loop. This signal makes the PID controller begin a new cycle and calculate a new PWM input value. The PWM has a saturating block; this saturation value is symmetrical for positive and negative values and can be configured. The ratio between PWM cycles and PID cycles and the number of bits of the PWM input can also be configured. The PWM block also has an on/off input, allowing the disconnection of the modulator and the brake of the motor.

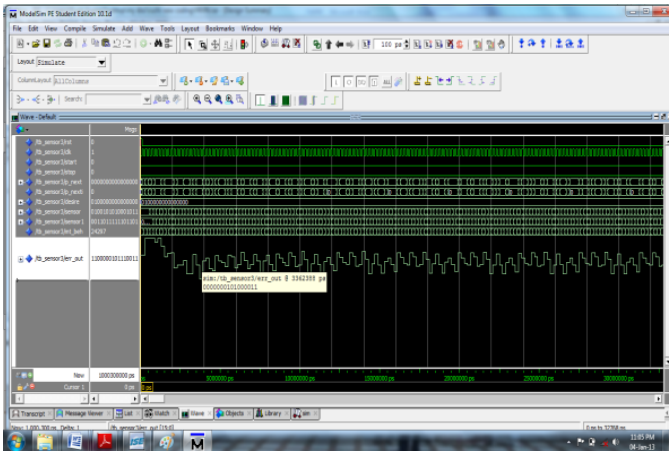
V. PROPOSED METHODOLOGY

We are proposing a sensor by eliminating PWM Modulator and A/D Convertor, which can reduce the power consumption and overall Delay for system. In block diagram shown below shows the comparison between General PID Controller and proposed PID Controller.

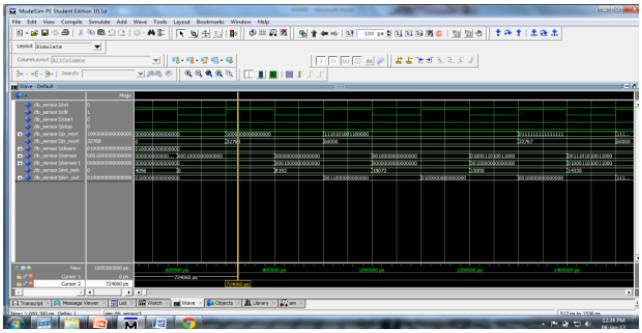
VI. RESULT AND DESCUSSION

PARAMETERS	VALUES
DELAY	13ns
TOTAL POWER	43.27mw
FREQUENCY	50 MHz

Simulation Results



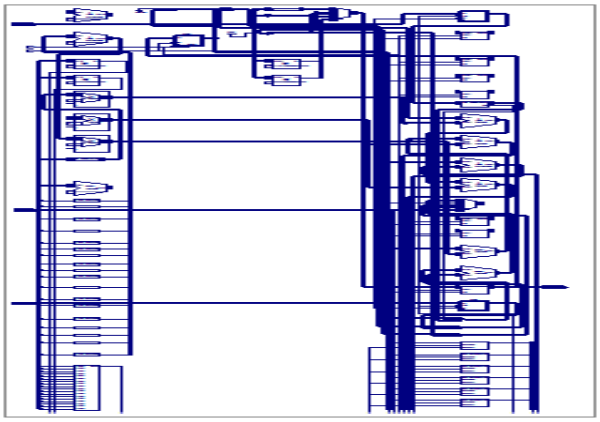
Error output



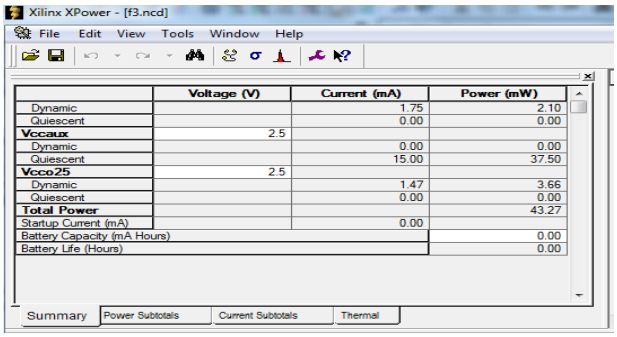
Utilization summary:-

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	165	7,168	2%	
Number of 4 input LUTs	303	7,168	4%	
Logic Distribution				
Number of occupied Slices	223	3,584	6%	
Number of Slices containing only related logic	223	223	100%	
Number of Slices containing unrelated logic	0	223	0%	
Total Number 4 input LUTs	362	7,168	5%	
Number used as logic	303			
Number used as a route thru	59			
Number of bonded IOBs	68	141	48%	
IOB Flip Flops	32			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	5,011			
Additional JTAG gate count for IOBs	3,264			

Synthesis Results

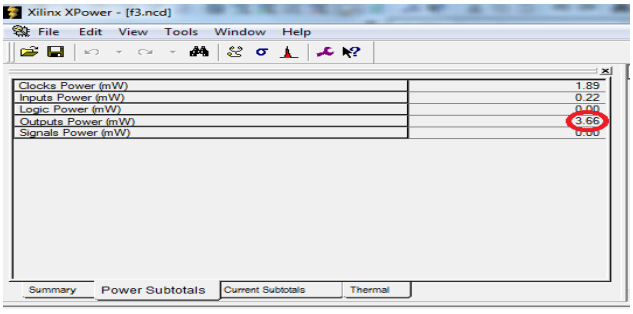


Total Power



	Voltage (V)	Current (mA)	Power (mW)
Dynamic		1.75	2.10
Quiescent		0.00	0.00
Vccaux	2.5		
Dynamic		0.00	0.00
Quiescent		15.00	37.50
Vcco25	2.5		
Dynamic		1.47	3.66
Quiescent		0.00	0.00
Total Power			43.27
Startup Current (mA)		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

Out Put Power



Clocks Power (mW)	1.89
Inputs Power (mW)	0.22
Logic Power (mW)	0.00
Outputs Power (mW)	3.66
Signals Power (mW)	0.00

VII. CONCLUSION

In this paper, a novel PID based controller was presented, for FPGA implementation. Hence the efficient dynamic system implementation is done which uses only 3.66 mw of the power. We conclude this research that we can minimize the cost of overall system by eliminating the PWM Modulator and A/D Convertor which can all give advantages over Power Consumption and Delay. We can achieve high speed as well as better performance and low cost as compared to analog counterpart. Average latency can be minimized about 14ns.

REFERENCES

1. Efficient Dynamic System Implementation Of Fpga Based Pid Control Algorithm For Temperature Control System. Volume 3, Issue 2, July – September (2012), pp. 306-312
2. High-Speed and Low-Power PID Structures for Embedded Applications PATMOS'11. Madrid: Spain (2011)"
3. ISSN 0974-2190 Volume 2, Number 1 (2010), pp. 71--82 Analysis and Implementation of Discrete Time PID Controllers using FPGA
4. FPGA technology for multi-axis control systems Armando Astarloa*, Jesús Lázaro, Unai Bidarte, Jaime Jiménez, Aitzol Zuloaga Accepted 1 September 2008
5. Design and Implementation of Modular FPGA-Based PID Controllers,Yuen Fong Chan, M. Moallem, Member, IEEE, and Wei Wang, Member, IEEE-2007.
6. Simulink/Modelsim Simulable VHDL PID Core for Industrial SoPC Multiaxis Controllers Jes´us L´azaro, Armando Astarloa, Jagoba Arias, Unai Bidarte, Aitzol Zuloaga IEEE-2006.
7. Xilinx Corp. Multipliers.
8. National Semiconductor. LMD18245 3A, 55V DMOS Full- Bridge Motor D