FPGA based RISC Processor with Inbuilt Auto tuned PID Controller for Liquid Level Control System

S. V. Kulkarni [1*], P. P. Shah [1], M. K. Bhanarkar [2]
[1] Department of Electronics, Devchand College, Arjunnagar, MS-India
[2] Department of Electronics, Shivaji University, Kolhapur, MS-India

Abstract:- In the industrial applications for the real time signal processing, nowadays most of the automatic control system are presents and most of the industries focused on a closed loop controller like PID (Proportional-Integral-Derivative) controller. Because it is common controller to find the oriented solutions based on Reduced Instructions Set Computer (RISC). This work deals with the hardware/software co-design of a PID coprocessor, all embedded on a system-on-chip device. The performances reached by a platform composed of an 8-bit MCU and a dynamically reconfigurable FPGA allow scheduling the PID algorithm as a set of tasks executed by both devices concurrently. Moreover, thanks to the flexible hardware characteristics, some modules synthesized into the FPGA are reconfigured at run-time while the rest keeps on active. This cost-effective approach, encouraged by its parallelism, is an alternative to commercial –both general-purpose and specific-purpose–processors in whatever made-to-measure engineering application. This paper deals with implementation of RISC processor having inbuilt auto tuned PID controller for Liquid level control system.

Keywords: - RISC Processor, Auto tuned PID Controller, Liquid level control, Simulink

I INTRODUCTION:-

FPGA- Field Programmable Gate Array; plays a major role for customizing the processor and reconfigures the many of the electronics. The Hardware Description Languages (HDLs) increase the range of options available to FPGA designers by enabling hardware implementation with the flexibility that language based design provides HDLs allow designers to implement flexible intellectual Property (IP), often referred to as IP cores. IP is the implementation of reusable components, which describe and implement hardware functionality [1].

Fied Programmable Gate Arrays (FPGA) is growing fast with cost reduction compareto ASIC design.This researchconcerned with the d esign and implementation of a lowcost8-bit Reduced Instruction Set Computer (RISC) processor on aFPGA.It provide the benefits of custom VLSI design while avoiding the initial cost, time delay and inherent risk of a conventional masked gate array [2]. They are customized by loading configuration data into the internal memory cell. RAM based FPGA's can be infinitely reprogrammed in-circuit in only a fraction of seconds. Design revisions even for fielded products can be implemented quickly and precisely [3].

RISC stands for Reduced Instructions Set Computer (RISC). It is a processor which uses only simple instructions, which performs low level operation in a single clock cycle. It has a very high performance capacity and capable of executing those instructions in a single microprocessor cycle. It has advantage of having pipelining therefore multiple instructions can execute in a single clock cycle. This results in high speed instruction execution. It also has Load/Store architecture where memory is accessed through particular instructions. It also has a simple Arithmetic Logic Unit (ALU) for basic operations with simple and uniform instruction set. This work presents the design of reconfigurable 8 bit RISC processor. The fig. No. 1 and 2 shows Block diagram of 8 bit RISC processor and RTL view of 8 bit RISC processor. The basic modules of this processor are programmed by using Verilog Hardware Description Language (VHDL), it is then verified the simulation result using XILLINX ISE 14.2 tool.[4]

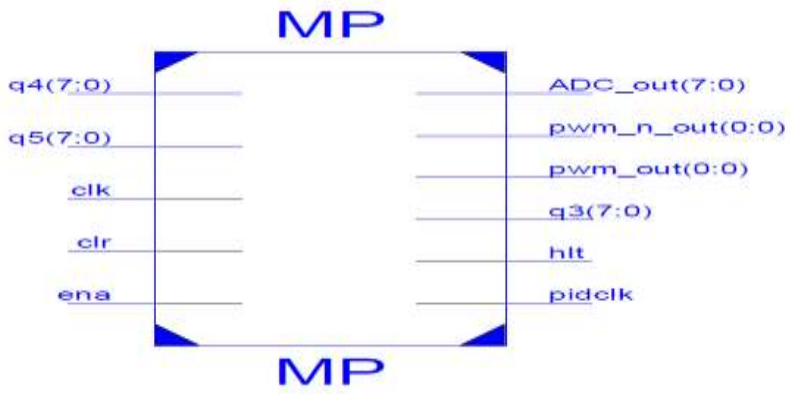

Fig 1:- Block diagram of RISC processor

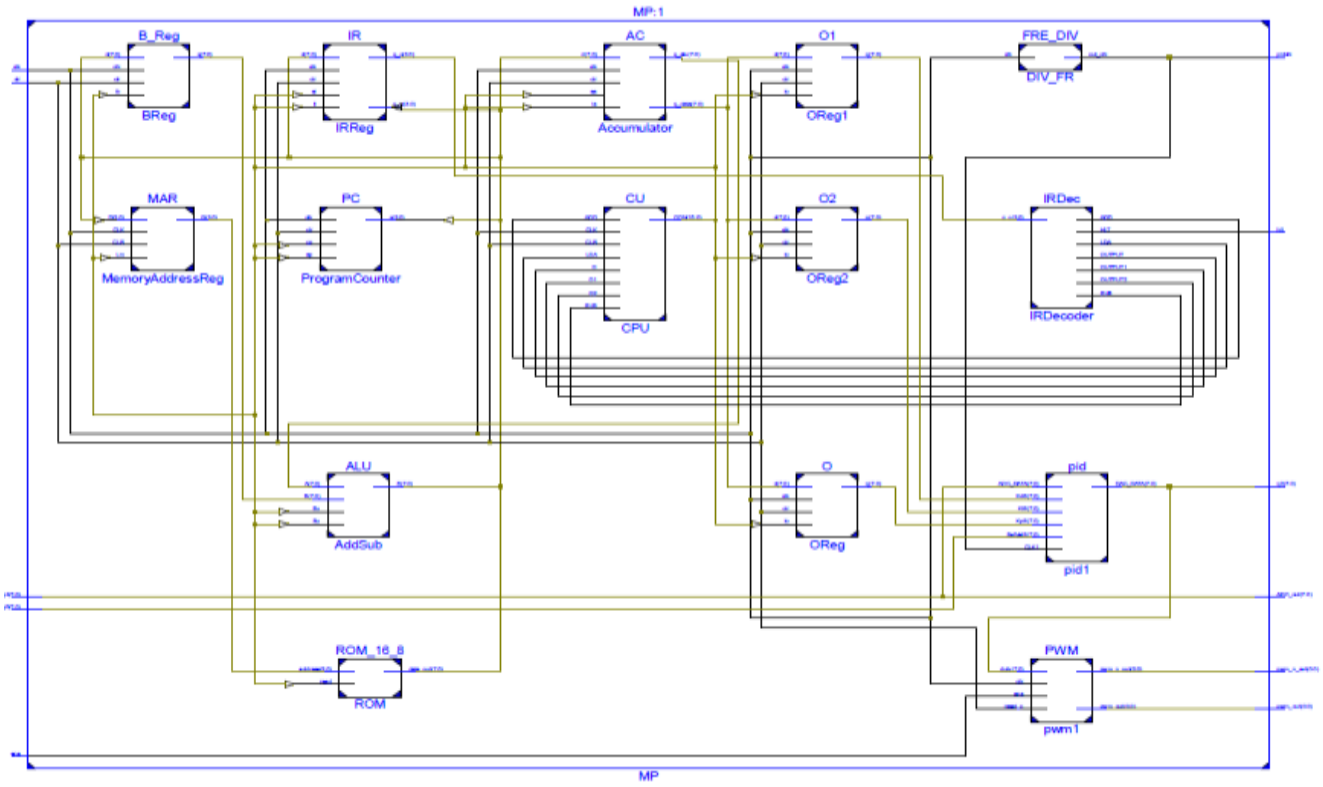

Fig 2:- RTL schematic of implemented RISC processor.

The fig. no. 2 shows the RTL view 8 bit RISC processor. It contains Central Processing Unit (CPU), Read Only Memory (ROM), Arithmetic and Logical Unit (ALU), Accumulator (A Register), B Register, Instruction Register (IR), Instruction Register (IR) Decoder, Memory Address Register (MAR), OUT Register, Frequency Divider and Supporting Peripherals for RISC processor like Proportional-Integral-Derivative (PID) controller, Counter, Pulse Width Modulation (PWM), and Universal Asynchronous Transfer Receiver (UART).[5]

II LIQUID LEVEL CONTROL SYSTEM

The PID controller looks for compensating the error between the desired and the effective value of the output signal. A closed loop is inserted for this in which the plant is self-fed by an input consisting of the sum of three terms: a proportional factor responsible for adjusting the control signal in the same percentage than the instantaneous error, a derivative factor that contributes proportionally to the error rate of change, and an integral term with the role of eliminating the steady state error by means of integrating the instantaneous error along the time. The three terms help to cancel any deviation or disturbance present in the system and maintain therefore the equilibrium, achieving an input tracking with a dynamic response in accordance with the fixed characteristics of the whole plant-controller.

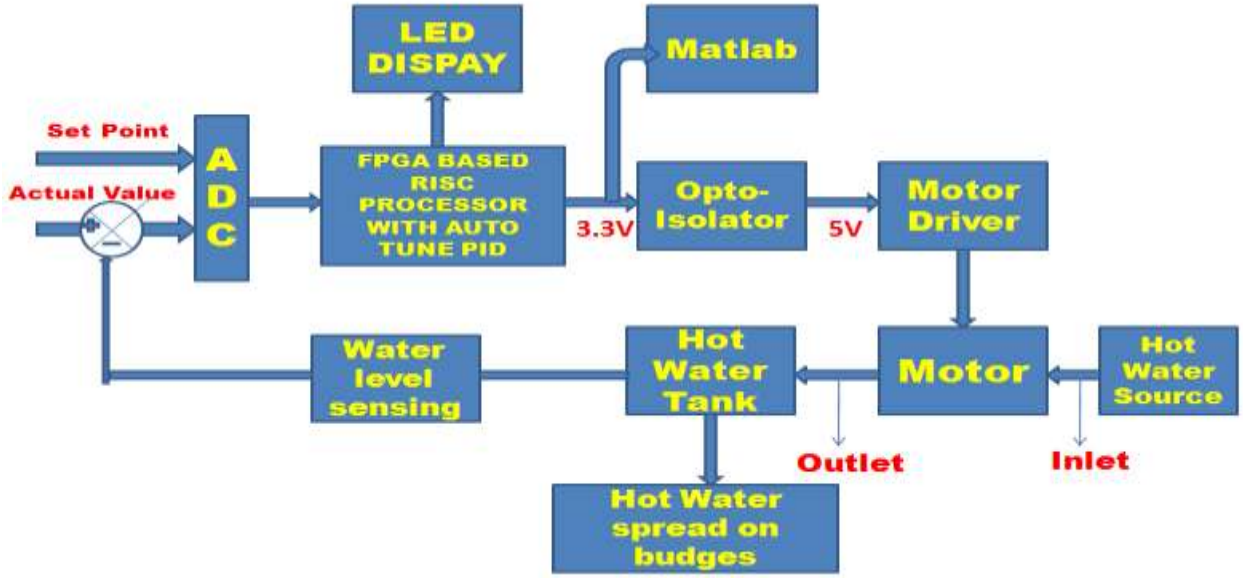

Fig. 3:- Block Diagram of Liquid level control system using Auto tune PID.

III HARDWARE IMPLEMENTATION:-

- FPGA based RISC processor with auto tune PID
- Opto-Isolator
- Motor Driver

- AC Induction Motor
- Water Tank
- Water Level Sensor

For the operation of liquid level control, the variable frequencies are Generated by the FPGA based RISC processor by using PWM application. Those variable frequencies are applied to motor through motor driver (L239D). As this is closed loop system, the frequency changes as error between actual value and set value is increased or decreased.

The above system consists of two potentiometers, one for the set point and another is for to adjusting the actual value coming from feedback. The difference between these two values are given to the Analog to Digital Converter (ADC), because the FPGA requires the digital input values but set value and actual value are the analog signals therefore, to convert analog values to digital values the PIC 18F4520 is used. It receives analog signals from the potentiometers and converted those values in to digital and gave to the FPGA. The FPGA based RISC processor contains the auto tune PID controller which is used for this process control application. For the implementation of RISC processor with auto tune PID the Numato FPGA sparten6 board (XC6SLX9) is used. By using Xilinx 14.2 web pack and i-sim simulator all system implementation is done. In this RISC processor, by setting the PID values for the single time, the PID controller controls the whole system for any change in the difference between set value and actual value. The FPGA based RISC processor receives the digital inputs from the ADC and based on those values the auto tune PID controller present in the RISC processor controls the system. The output of the PID controller is displayed on the matlab. By designing the simulink in the matlab, the current output of auto tune PID controller is observed in the matlab and at a same time the output of PID controller given to the motor driver circuit through opto-isolator. The optical isolator provides isolation between the outputs of RISC processor which is 3.3V and input of motor driver IC which is 5V.

As difference between set point (set water level) and actual value (output of the level sensor) varies the PID controller varies the speed of the motor. As speed of the motor varies the motor feeds more water from the hot water source to hot water tank. From this tank hot water spread on the budges. If the load of the system increases that is more hot water spread on the budges. That's resulting in the change in water tank level. That level sense by the level sensor and the output of the sensor is the actual value of the system. That generates more difference in the set point and actual value. As the difference increases auto tune PID controller varies the speed of the motor that results in more water feeds in the hot water tank from the hot water source. Therefore the level of the hot water tank remains constant.

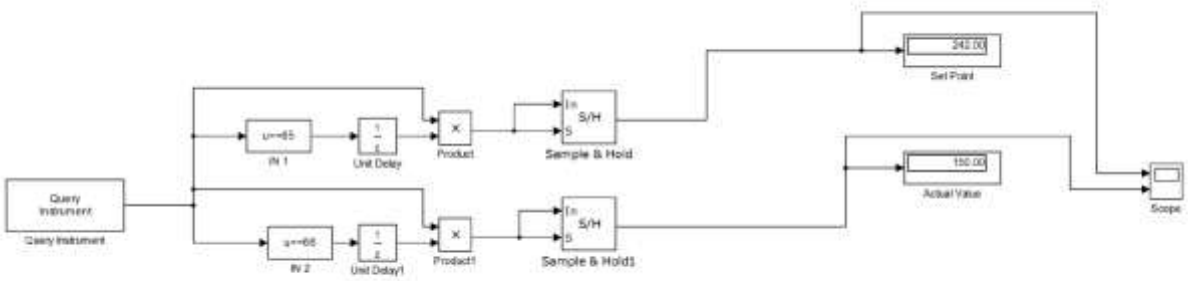

Fig. 4:- Simulink design for Liquid level control system using Matlab.

The above simulink designed in the Matlab 7.9 to observe the real time output of auto tune PID controller. In this the query instrument reads the real time output value of PID. As shown in fig. 9 the two displays are used to display real time set point and actual value and the block scope displays the waveform of the real time PID controller's output.

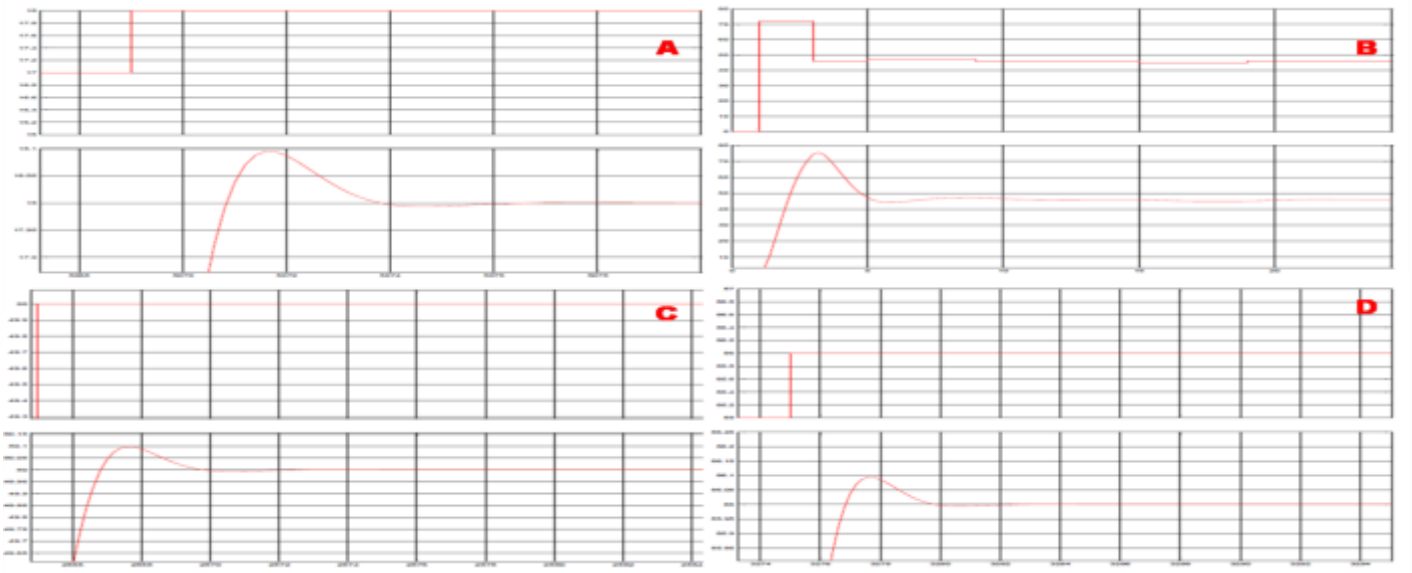

Fig. 5:- Real time output Wave forms with different set points and actual values in Matlab.

Sr. No.	Figure	Set Point	Actual Value	Rise Time (Sec)	Peak Time (Sec)	Settling Time (Sec)	Peak Overshoot %
1	A	33	15	1	2.6	4.5	2
2	B	45	00	1	2.2	4.5	2
3	C	50	00	2	4	4	5
4	D	170	84	1	3	4	2

Table 1:- Real time Output of Auto tuned PID controller with various set points and actual values in Matlab

IV CONCLUSION

The present paper deals with the design of FPGA based 8 bit RISC processor with PID controller for Liquid Level Control System. The basic modules of this processor are programmed by using Verilog Hardware Description Language (VHDL), it is then verified the simulation result using XILLINX ISE 12.4 tool. The present 8 bit RISC processor performs the controlling action of drum liquid level control for sugar cane industry. It concludes from the table no 1 that, the settling time increases or decrease as set point is changed.

V REFERANCES

[1] Application note for FarhatMasood, RISC and CISC Computer Architecture, available at http://arxiv.org/ftp/arxiv/papers/1101/1101.5364.pdf, retrieved from: 11/07/2012

[2] RC Cofer and Ben Harding, "FPGA Soft Processor Design Considerations", Available at http://www.eetimes.com/General/PrintView/4014791**,** online published, date 10/12/2005.

[3] W.M Medany, K.Kooheji,"Design and Implementation of a 32-bit RISC Processor", Available at http://www.ursi.org/proceedings/procGA08/papers/DP1p2.pdf, retrieved from:24/08/2012

[4]. Kulkarni, S. V.. Nadaf A. I., Shah P. P., Bhanarkar M. K., 2016, Design of FPGA based 8 bit RISC Processor, International journal of VLSI design and embedded system, 7, 1691-1698.

[5]. Kulkarni, S. V., Shah P. P., Bhanarkar M. K., 2017,Design of FPGA based 8 bit RISC Processor with peripherals, International journal of Development and research, 7(7), 13531-13535.

[6]. Uma, R. 2012. Design and Performance analysis of 8 bit RISC processor using Xilinx Tool., International Journal Of Engineering Research and Applications, 2(2), 053-058.

[7] Luker, Jarrod D., Prasad, Vinod B. 2001 "RISC system design in a FPGA" MWSCAS 2001, vol2, pp. 532-536,.