Soft Core Processor for Electrostatic Precipitator Controller

[1]Dr. R. Venkata Siva Reddy, [2]Dr. Bharathi S. H
[1] Professor, Department of E&CE, REVA ITM, Bangalore, [2] Professor, Department of E&CE, REVA ITM Bangalore
rvsiva_reddy@ieee.org, bharathi_hiremutt@rediffmail.com

Abstract— Reconfigurable processor architectures are increasingly becoming popular for many Industrial applications. Microprocessor obsolescence is the major concern for many companies. Reconfigurable logic can provide a feasible solution to this problem. By using soft core microprocessor embedded within a programmable logic device, we can own the processor core for use in any future devices for Industrial control application. This work presents soft-core processor design based on 8 bit microprocessor, address generator, decoder, ROM to house the firmware, RAM and Input/Outputs all on a single FPGA. The soft-core processor designed is more efficient in terms of processing speed by over 30 times over the prevailing 8 bit microprocessors. In this paper we have designed, developed and implemented architectures and novel algorithms for 8 bit processor using RTL complaint Verilog code. The system has been synthesized at 118 MHz using Xilinx target FPGA Vertex board and has a gate count of 85,000.The RTL design is truly platform and technology independent.

***Index Terms*- Field Programmable Gate Array, Soft-core processor, Verilog and Resistor Transfer Logic, Intellectual Property(IP) Electrostatic precipitator(ESP).**

I. INTRODUCTION

Now days to make the difference on the market [1], new industrial control processors have to be highly performing, very flexible and reliable. Microprocessor obsolescence is a major concern for many companies. To cope up with all these challenges, the designers can rely on more and more on mature digital electronic technologies which come along with friendly software development tools. To implement efficient real time industrial control systems, designers have the choice between two main families of digital device technologies. The first families based on pure software platform. The associated devices are microcontrollers and DSP controllers. These components integrate a performing microprocessor core along with several peripherals which are necessary to control the targeted system in real time and to communicate with the industrial environment. The main advantage of this approach is the maturity of these technologies, the quality of the associated development tools as well as their low price. The main limitations are the difficulty to take advantageof thepotential parallelism offered by the control algorithm to be implemented and as a consequence, the limitation of the performance in terms of the throughputs and achievable bandwidths.

The emergent trend is to move from custom made microprocessors to soft core processors embedded with in FPGAs. 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), some of the advantages of designing with soft IP cores include Higher level of design reuse, reduced obsolescence risk, simplified design update or change, increased design implementation options through design modularization. All of the benefits and characteristics of soft IP cores are realized by soft processor cores implemented within FPGA components. In this work soft core processor for ESP controller application has been designed and implemented.

Electrostatic precipitator controllers are used in fly ash disposal in a thermal power plant. Several tons of fly ash are generated, disposal of which is quite cumbersome [2]. For example, a 210 MW thermal power plant generates about 4000 tons of ash every day. If released in the air, the entire township will be

covered by ash. Water stream cannot directly wash the ash away since passage will get clogged in a short time. The solution is to apply a high DC voltage in the order of 80 KV in the ESP, a large chamber with electrodes all over, where the fly ash is blown in from a boiler. Ash gets attracted to negative electrodes and hence gets tamed. Activating special hammers frees the ash, which is promptly washed away by a water stream at the bottom of the electrostatic precipitator and finally disposed of in huge ash ponds situated about 5 miles away from the power house. The special hammers need to be activated in a specific sequence in order to dislodge the ash from the electrode. These hammers are, however, controlled by another controller called Rapper controller.

All the previous application approaches for ESP controller use 8085 microprocessor. Very few ESP Controllers have been reported in the literature. In the work of Ref. [3], the Electrostatic Precipitator Management System is built around three physically and functionally autonomous microprocessor based units mounted together with associated accessory boards inside a panel and interconnected by a communication channel. I. Stevanović et al. and Zi-Sheng Zhang et al. have developed Control Systems for Electrostatic Dust Separator in Thermal Power Plants [4, 5]. Power industries such as ABB, Flakt, BHEL, etc. have designed ESP controllers using 8085 processor. However, system cost is quite high and their processing powers are also low.

If the ESP Controllers are designed using FPGA or ASIC, the processing throughput can be enhanced by over 30 times and also bring down the cost by at least 50%. Number of processors can be integrated in to a single FPGA with the intent of improving the throughput and reducing the system cost, we propose here the design of an Intellectual Property (IP) Core for the ESP controller system so that it may be implemented on an FPGA or as an ASIC. A. Moore, Scott [6] and Aviral Mittal [7] have developed IP cores for 8080 CPU (Verilog) and 8085 (VHDL) respectively. However, they are not RTL compliant. The IP Core for the ESP processor described in this paper has achieved significant speed of operation (over 30 folds) compared to 8085 based systems used presently in the thermal power industries. Our design isfully RTL complaint and processor design is the customized solution for ESP controllers.

The paper is organized as follows. Section 2 describes the ESP controller system architecture as implemented in this work. Section 3 presents sample Algorithmic State Machine (ASM) charts for execution of instructions. It also presents the group of instructions implemented together with a comparison of processing speed with that of the existing 8085 based systems. The simulation results and the Place & Route results using Xilinx tool are presented in Section 4. The last section presents the conclusions derived as well as the scope for future work.

II. ARCHITECTURE OF THE PROPOSED ESP CONTROLLER SYSTEM

Electrostatic precipitator controller system may be realized using a single FPGA or an ASIC. The proposed architecture for realizing the ESP Controller on an FPGA or as an ASIC is presented in Fig. 1. As mentioned in the introduction, the solution for ash disposal is to apply a high DC voltage in the order of 80 KV in the ESP. The DC high voltage is generated by firing a couple of thyristors configured as full-wave rectifiers. The firing circuits for these thyristors are based on pulse transformers housed in a control panel in which the ESP controller unit is mounted. A small capacity transformer (30 VA), together with opto-isolated transistors identify the zero crossover and AC positive/negative swings in order to generate triggers for firing the thyristors at appropriate time.The transformer is also used to supply power to the ESP controller unit.

The positive and negative AC power swings are converted as pulses marked ‘AC +ve’ and ‘AC –ve’ in the Input/Output (I/O) card and are fed as inputs to the FPGA/ASIC processor. RESET* derived from the system reset and the watch dog timer in the I/O card is connected to an analog input. TRIGGER is an output pulse generated by the FPGA/ASIC once every scan time of the ESP controller, which triggers the watch dog timer. In the rare event of the controller loosing control owing to severe noise conditions etc., the trigger pulse will not be

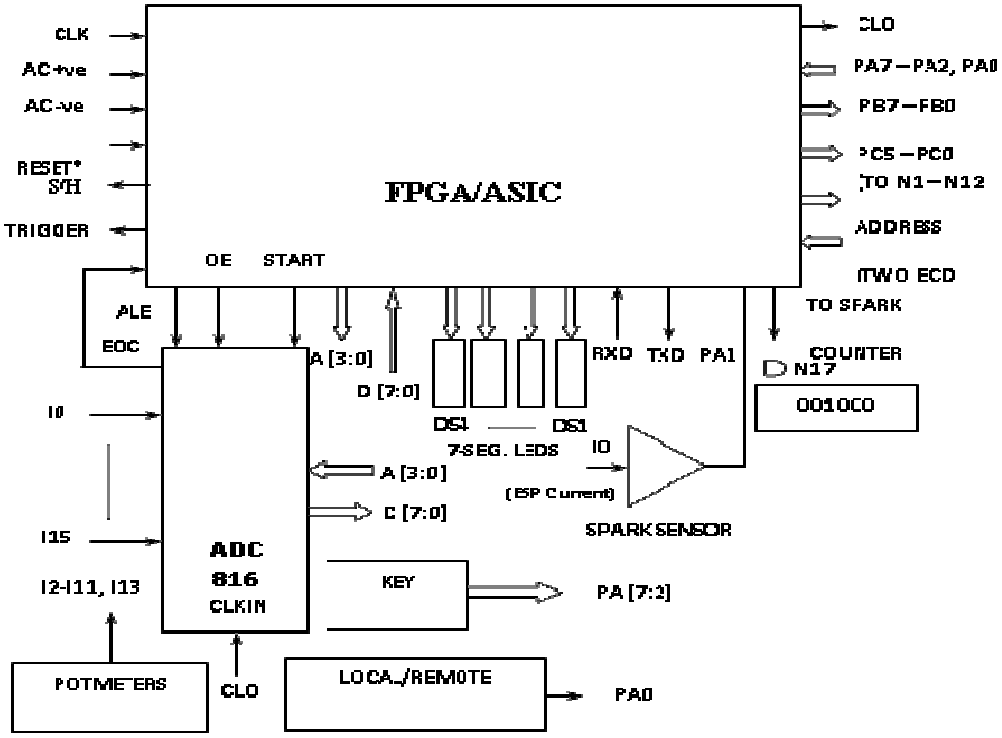

Figure 1. Architecture of the Proposed FPGA/ASIC based Electrostatic Precipitator Controller System

generated. This in turn would reset the system and recover the normal system operation again, thus preventing system crash. Self-recovery is one of the most important characteristics in embedded systems. S/H output signal is asserted whenever the sample and hold is to be processed. I1 senses the integrated ESP high voltage, whereas I14 senses the dynamically changing high voltage to measure the peak or valley of the ESP high voltage. The sampled output of I14 is fed to I15 input of an ADC 816 mounted on the FPGA/ASIC board. I0 input of the ADC is the ESP current measured by the I/O card. I2 to I11 and I13 are connected to potmeters P1 to P11 respectively. The FPGA/ASIC generates a four bit address: A [3:0] for the ADC to select one of the sixteen analog channels I0 to I15 at a time. This address can be registered by applying 'ALE' signal. The analog to digital conversion can be initiated by asserting the 'START' signal. Once the conversion is complete, the ADC will assert 'EOC' signal. Subsequently, the FPGA/ASIC reads the converted digital channel information via D [7:0] with 'OE' asserted. The FPGA/ASIC also generates a low frequency clock 'CLO' for the ADC operation.

A SPARK SENSOR circuit derived from the ESP current and comprising an analog comparator and a register indicates when the sparking takes place in the electrostatic precipitator through PA1 port.

Six key pads are connected to input ports PA [7:2] and the LOCAL/REMOTE switch to PA0. DS4 to DS1 are seven segment LEDs and are connected to output ports of the FPGA/ASIC. TXD and RXD are transmit and receive serial data signals respectively connected to a serial network after conditioning the signals using CNY17-2 opto- isolators. PB7 to PB0 are field inputs derived from the I/O card. PC5 to PC0 are outputs from the FPGA/ASIC to drive four relays in the I/O card and the thyristors firing card. N1 to N12 are discrete LEDs located on

the front fascia of the ESP Controller system and are driven by the output ports. Another digital output advances a non-resettable six digit electromagnetic counter once every time a spark is sensed. Its status is indicated by N17 LED. Potmeters P1 to P11 are used to set various engineering parameters, thus providing non-volatile settings. Each ESP controller unit has a unique identityby setting an 'ADDRESS' using two BCD switches. The address range is 00 to 99.

III. DEVELOPMENT OF ALGORITHM FOR ESPCONTROLLER PROCESSOR

The proposed ESP processor emulates the presently existing 8085 based systems. The system is primarily a Finite State Machine (FSM). The process Algorithm may, therefore, be easily designed using Algorithmic State Machine charts [8]. The ASM chart for the proposed system is presented in Fig. 2. It shows the algorithm of a couple of instructions as examples although most of the instructions of 8085 processor have been realized in this work.

Upon switching on the system, a power on reset signal is issued taking the processor system to 'Initialize' state "0" as shown in the ASM chart. In this state, the processor initializes various control signals such as the read, write, address latch enable and IO/Memory signals as well as the program counter and the address bus of the system. The read(rd_n), write (wr_n) and IO/Memory (iom_n) signals are active low. The program counter (pc) and the address (addr) bus are of width 16 bits. A low "iom_n" signal indicates that the processor has selected the memory for access; otherwise, it has selected Input/Output of the system. The "ale" signal, as the name implies, is used to separate the time multiplexed 8-bits Address/Data bus named "ad [7:0]". The state of the FSM changes from one to another at every rising edge of the system clock.

With the arrival of the clock, the FSM advances to the next state named "Assert ALE", where the "ale" signal is set. The address bus derives the concatenated values of the MSB 8 bits of the program counter and the Address/Data bus. In the next state marked "Opcode Fetch", the user firmware stored in ROM is fetched. First, the Op code of the instruction is fetched, followed by other bytes in subsequent states, depending upon which instruction is being processed. In this state, the "ale" signal is withdrawn and the read signal is asserted, thus reading the "opcode" from the User memory. This appears on the Address/Data bus. With the arrival of the third clock, the FSM goes to the state marked "Execute Instruction", where the actual processing of the instruction commences. To start with, the read signal is withdrawn and the "opcode" is decoded. Depending upon the instruction, the FSM is taken to a group of other states in order to complete the processing of the instruction. For examples, the processing of a couple of instructions, namely, MOV A, B and XRA M is shown in the chart.

If the "opcode" read is 78 H, then it is the first one; else if it is AE H for the second instruction mentioned above. In the first case, the accumulator "A" gets the contents of the "B" register, the program counter is incremented by 1 and the control rolls back to state "1" with the arrival of the next clock, ready to process the next instruction. Thus for this instruction, the processing time is 4 clock cycles.

For the second instruction, the "opcode" read is AE H, in which case the MSB bus address gets the value from H register and branches to the next state "500". In this state, the LSB bus address is derived from L register and ALE signal is activated once again in order to send the memory address over the address bus. In the next state "501", the ALE pulse is withdrawn and read pulse is issued in order to fetch the contents of the memory addressed by HL register pair. The next state is the executing phase of the actual instruction XRA M. In this state, the XOR of A and M is evaluated and read pulse is withdrawn. CY, AC flags are cleared and Z, S and P flags are set or cleared accordingly. Finally, the program counter is incremented by 1 and the control loops back to state "1" to continue processing the next instruction.

Most of the 8085 instructions have been coded. Table 1 presents some of the instructions realized along with their processing times. The table also compares the processing speed of the proposed FPGA implementation of ESP Controller System with the existing ESP Systems. Most of the instructions maintain the

same processing clock cycles in the two cases. The proposed implementation scores over the existing systems for instructions such as SPHL, DAD, JNZ, HLT and PCHL, whereas it is the other way round for stack operation instructions. As can be seen from the table, the proposed FPGA implementation is faster than the existing systems by over 30 times. However, the memory utilization is exactly the same in both the cases.

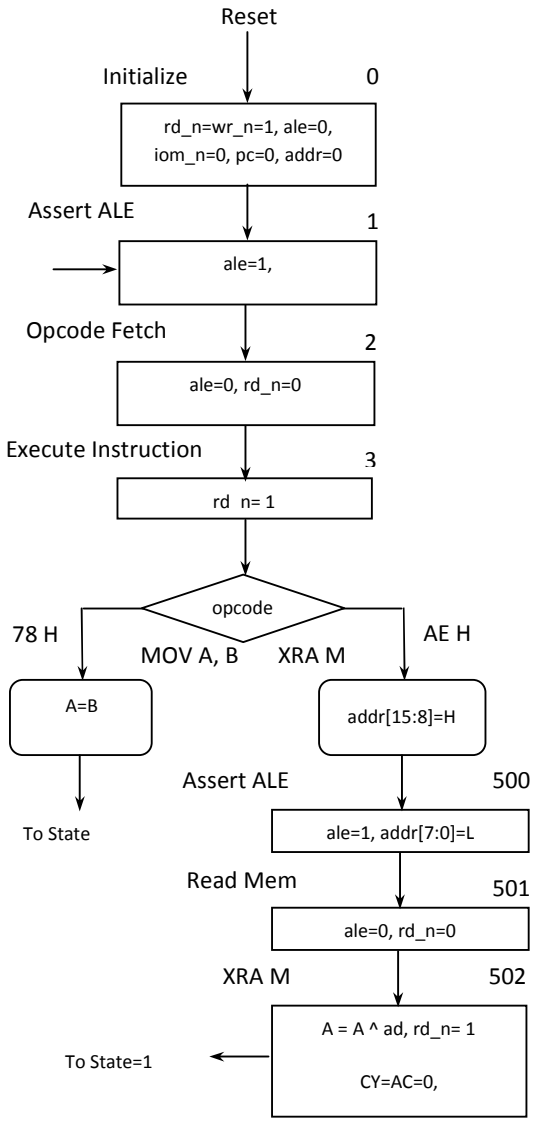

Figure 2 Sample ASM Chart forProcessing Instructions

1V. Simulation and Place & Route Results for ESP Control System

The development of hardware algorithm for the ESP Control System was presented in the previous section. The partial list of instructions implemented was also presented in Table 1. The complete processing system was coded in RTL compliant Verilog, a popular industry-standard hardware design language. The simulation was run on Modelsim and the results for a sample group of instructions presented in the following are shown in Fig. 3(a) &3(b). As seen in the waveforms, the instruction MVI A, AA H located in 1600 (pc value) commences at 21355 ns and the last instruction JMP 0 completes at 22085 ns, thus processing the group of instructions in 730 ns or 73 clock cycles since the “clk” runs at 100 MHz. The processing time of individual instructions observed is also tabulated in the group of instructions. These values are in agreement with the processing time presented in Table 1. “cs0” to “cs1” select ROM, RAM and UART. Carefully studying the waveforms, we see that all the instructions are working correctly. User Program with all the instructions developed have been written and found to be working satisfactorily.

Sample User Instructions in ROM

Address	Contents	Instruction	Processing Time in clockcycles
1600:	8'h3E	MVI A, AAH	7
1601:	8'hAA		
1602:	8'h21	LXI H, 641H	10
1603:	8'h41		
1604:	8'h06		
1605:	8'hAE	XRA M	7
1606:	8'hCA	JZ 1709	10/4
1607:	8'hAD		
1608:	8'h06		
1709:	8'h37	STC	4
1710:	8'h3F	CMC	4
1711:	8'h1E	MVI E, 55H	7
1712:	8'h55		
1713:	8'hBB	CMP E	4
1714:	8'hDA	JC 1800	10/4
1715:	8'h08		
1716:	8'h07		
1800:	8'hC3	JMP 0	10
1801:	8'h0		

1802: 8'h0

Total processing time in clock cycles: 73

Place & Route had been run and its results are as follows:

The design was mapped on to a target FPGA 2VP30FF896-6. Using Xilinx ISE 8.2 tool,

Table 1 Comparison of Processing Speed of the Proposed Implementation of ESP Control System with Existing ESP Systems

Sample Instructions	**Proposed FPGA Implementation of ESPC System**		**Existing ESPC Systems**	
	No. of clock cycles	**Processing speed in nanoseconds working@ 100MHz**	**No. of clock cycles**	**Execution speed in nanoseconds**
MOV reg, reg; XCHG / SPHL	4	40	6-Apr	1302/1953
MVI reg, data; ANI/CPI data	7	70	7	2279
LXI rp, 16 bit data	10	100	10	3255
LDAX rp; SUI data;	7	70	7	2279
ADD/ADC/SBB reg / DAD rp	4	40	4/10	1302/3255
CMA; STC; CMC; DAA	4	40	4	1302
INR/DCR reg / INX/DCX rp	4	40	4/6	1302/1953
ANA reg; RLC/RAL; RAR/RRC	4	40	4	1302
MOV reg, M; XRA M;	7	70	7	2279
CMP reg	4	40	4	1302
JMP addr	10	100	10	3255
JNZ addr	10/4	100/40	10/7	3255/2279
CALL addr	20	200	18	5859
RET	12	120	10	3255
PUSH	14	140	12	3906
POP	12	120	10	3255
EI; DI; NOP / HLT / PCHL	4	40	4/5/6	1302/1628/1953

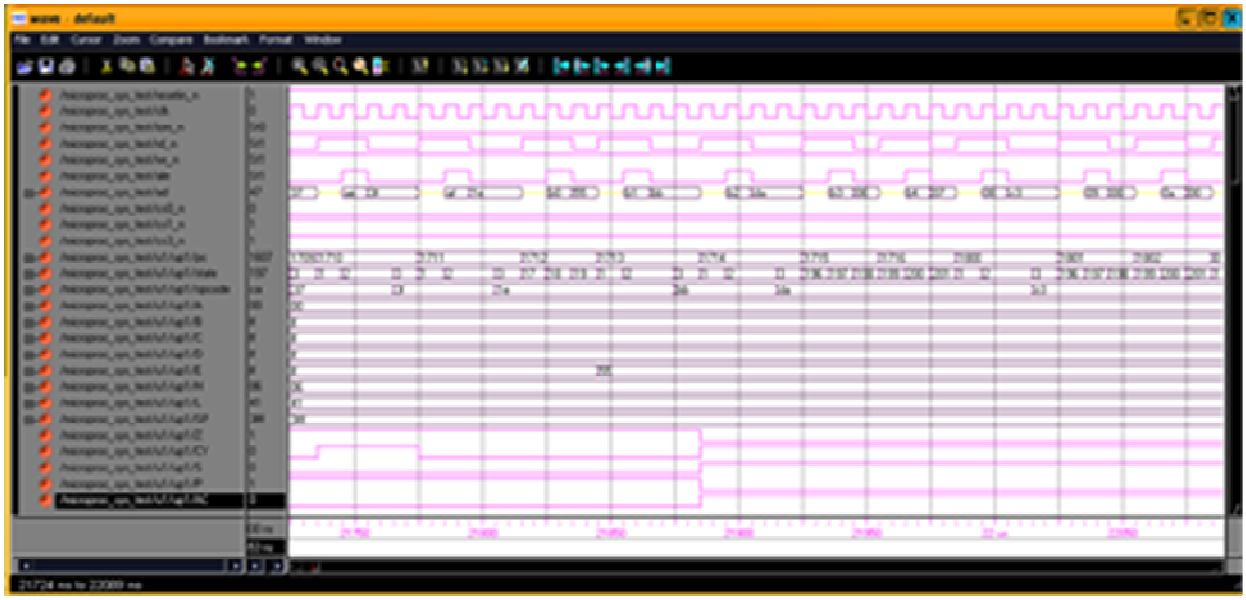

Figure 3(a) Simulation Results of the Proposed FPGA/ASIC Based Electrostatic Precipitator ControllerSystem

Figure 3(b) Simulation Results of the Proposed FPGA/ASIC Based Electrostatic Precipitator Controller System

Place & Route Results

Selected Device: 2vp30ff896-6

No. of occupied Slices: 5,151 out of 13,696

Total No. of 4 input LUTs: 9,549 out of 27,392 34%

Total equivalent gate count for design: 85,364

Additional JTAG gate count for IOBs: 2,208

Timing Summary:

Maximum Frequency: 118.273 MHz

From the P & R report, we see that the ESP Control processor has a gate count of about 85,364 and is capable of running at 118 MHz.Therefore we are justified in running the simulation at 100 MHz. The RTL schematic of ESP Controller is shown in Fig. 4. The schematic depicts the microprocessor module as specified in the design.

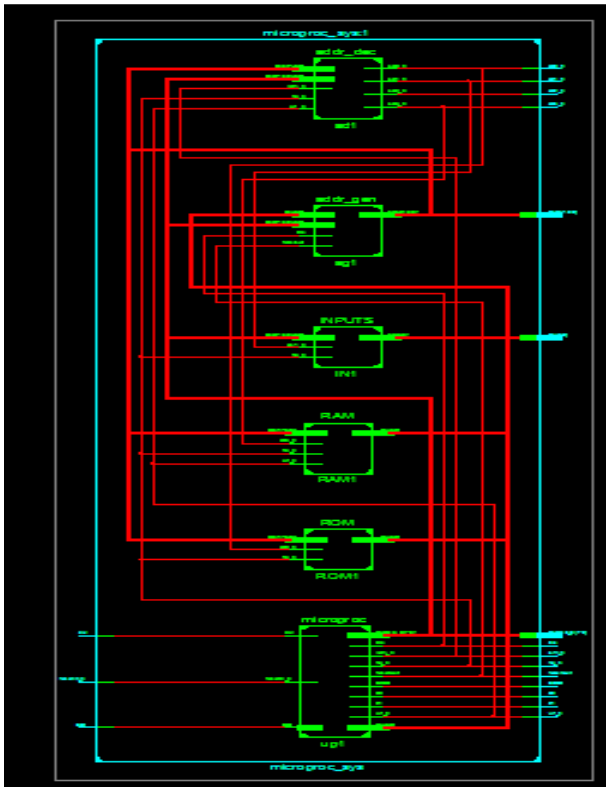

Figure 4 RTL schematic of ESP Controller

V. CONCLUSION

The IP core design of ESP Control system was presented for FPGA implementation. The design conforms to RTL coding guidelines and is independent of vendor device, be it FPGA or ASIC, as well as technology. The design occupies about 85,000 gates and can operate at a Maximum frequency of 118 MHz.The IP core has potential business value in ESP controllers to validate the design, a complete Verilog description along with FPGA RTL schematic has been presented. Though many soft core processors are available in market the one designed has full flexibility for industrial controls.

References

[1] Eric Monmasson, et al.,"FPGAs in Industrial Control Applications" , IEEE Transactions on Industrial Informatics, Vol.7, N0. 2, May 2011.

[2] S. Ramachandran, "Digital VLSI Systems Design", Springer, Netherlands, 2007, www.springer.com.

[3] N. V. P. R. Durga Prasad, T. Lakshminarayana, et al.,"Automatic Control and Management of Electrostatic Precipitator", IEEE Transactions on Industry Applications, pp. 561-567, Vol. 35, No. 3, May/June, 1999.

[4] I. Stevanovic. R. Prole. S. Dobricic. D. Jevtic. D. Amautovic. S. Vukosavic, "Development Power Supply and Control of Electrostatic Dust Separatior in Thermal Power Plants." Power Plants 2006 Serbia.

[5] Zi-Sheng Zhang, Qing Li, Zhi-Qiang Liu and Qiang Zhao Yu-Rong, "Control System of High Voltage Electrostatic Precipitator Based on Field Bus for Coal ash", Machine Learning and Cybernetics, International Conference on Publication, pp. 983-985, Aug. 2006, China.

[6] A. Moore, Scott, "8080 compatible CPU inVerilog", http://opencores.org/project,cpu8080.

[7] Aviral Mittal, Philips Semiconductors, Southampton http://www.vlsiip.com/vhdl/8085.vhd

[8] M. Morris Mano, "Digital Design", Prentice Hall, NJ, 2002.

[9] Jason G. Tong, Ian D. L. Anderson and Mohammed A. S. KhalidUniversity of Windsor " Soft Core Processors for Embedded Systems" – Department of Electrical and Computer Engineering Research Centre for Integrated Microsystems Windsor, Ontario, Canada.

[10] [16] Zi-Sheng Zhang, Qing Li, Zhi-Qiang Liu and Qiang Zhao Yu-Rong, "Control System of High Voltage Electrostatic Precipitator Based on Field Bus for Coal ash", Machine Learning and Cybernetics, International Conference on Publication, pp. 983-985, Aug. 2006, China.

[11]M.G.Kumar, S.Sekar, R.Sivasubramanian, " Environmentally Acceptable Emission from Power Plant Through improved Precipitator controls", Proc. 1st International Conference on Green Power- The need for the 21st century.

[12] Electrostatic Precipitator: Technical data: Supplied by Bharat Heavy Electricals Ltd. For Project.

[13] Robert R.Cryanck: "Improving Precipitator Performance and Saving Energy with intermittent Energization- A case study" 78th Annual Meeting of air pollution control Association, Detroit, Michigan (June, 1985)

[14] F. Zhengwei, J.E Carletta, and R.J. Veillette, " A methodology for FPGA- based control implementation," IEEE Trans. Control Syst. Technol., vol.13, no.6, pp.997-987, Nov.2005

[15] Bradbur, K.Francis, S.Porle. Electrostatic Precipitators for Industrial Applications, Published byRheva. Brussels, Belgium, 2005