Asian Journal of Convergence in Technology Volume1, Issue 2
Issn No.:2350-1146, I.F-1.7

Architectural Design Of Instruction List Processor For Process Control

Dheeraj Umakant Kore
Department Of Electronics & Telecommunication
Trinity College Of Engineering & Research
University Of Pune, India
kore.dheeraj09@gmail.com

Rahul A. Pagare
Department Of Electronics & Telecommunication
Trinity College Of Engineering & Research
University Of Pune, India.
pagarera@gmail.com

***Abstract*— In this paper an architecture of Instruction List (IL) processor is designed for process control according to the norm of IEC 61131-3. Increasing complexity in process control and safety critical application require fast output response. In order to improve execution speed of process control operation controller generates fast output response. The PLC (Programmable Logic Controller) is device that automates the process control operation. The designed processor exhibits desired performance much higher than current commercial PLCs. The architecture is to be implemented on FPGA platform for verification purpose. The proposed architecture is to be specified fully in VHDL and is designed to emulate the functionality of IL Processor in terms of instruction set fetching, decoding, executing, operand addressing and bus operation. To validate the advance of the proposed architecture, two ladder programs are compiled to the instruction set of proposed IL processor as well as in IL programming language.**

Keywords*— *Architecture, Instruction List(IL),VHDL,FPGA, IEC 61131-3 , PLC, FPGA

I INTRODUCTION

In early 80's hardware is described in schematic diagrams as circuit become more and more complex it become more and more impracticable which gives rise to requirement of text based description that enables few lines codes. PLC is a new servomechanism, which combines the technique of automation and communication, and regards the microprocessor as the core. Because of the high dependability , better anti-jamming ability, hard real-time, volatile control program, PLC is widely applied in the field of industry, such as steel, oil, chemical industry, electric power, building materials, machine manufacture, auto, traffic and so on. In China, most industrial PLC adopts general microprocessor or consists of bool cooperating processor. Pipelining is used to improve the overall CPI (Clock Cycles per Instruction) is proposed in implementation of 16 bit RISC processor on FPGA [1].The general processor mostly deals with the data format like byte or word, but mostly PLC instructions are bool instructions, which occupy 69% frequency of PLC instructions. [2] So the general processors hardly satisfy PLC's practical application. Understanding these facts, it is proposed to develop IEC61131-3 compliant instruction list processor using Xilinx design environment for better response during safety critical operations [3]. It is also proposed to compare the results obtained with conventional processor as well as PLC available in the open market.

II. IL PROCESSOR INSTRUCTION SET

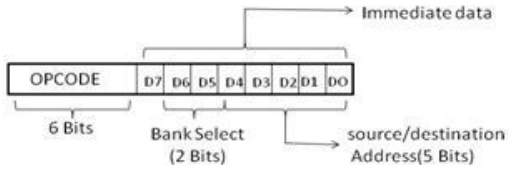
To reduce complexity for the users of PLCs, the International Electrotechnical Commission (IEC) elaborated the standard IEC 61131. Part 3 of this standard was published in 1992[4]. They have now become a standard that is accepted and appreciated around the world, as it reduces training efforts and fosters harmonization. As a starting point for the IL processor design, the IEC 61131-3 IL programming language is used, which is made of the following instructions (fig.1).

Operator	Modifiers	Data Types	Description
LD	N	many	set current result to value
ST	N	many	store current result to location
S, R		BOOL	set or reset a value (latches or flip-flops)
AND, &	N, (	BOOL	boolean and
OR	N, (	BOOL	boolean or
XOR	N, (	BOOL	boolean exclusive or
ADD	(	many	mathematical add
SUB	(	many	mathematical subtraction
MUL	(	many	mathematical multiplication
DIV	(	many	mathematical division
GT	(	many	comparison greater than >
GE	(	many	comparison greater than or equal >=
EQ	(	many	comparison equals =
NE	(	many	comparison not equal <>
LE	(	many	comparison less than or equals <=
LT	(	many	comparison less than <
JMP	C, N	LABEL	jump to LABEL
CAL	C, N	NAME	call subroutine NAME
RET	C, N		return from subroutine call
)			get value from stack

Fig. 1 Instruction set according to the norm IEC-61131-3

In the proposed design the instructions MUL, DIV, were not implemented, which makes reference to the multiplication and division operations. Next step is to define the PLC application specific instructions format and instruction for proposed IL processor. The main idea behind this is to enhance performance by exploiting a prior knowledge about the

processor's target application and can improve the speed of executing most PLC instructions by its instruction set and architecture [5]. The PLC application specific instructions are the core of executing PLC instructions. Instructions which will be used by the proposed IL processor to carry out the execution of the instructions stipulated in the norm is shown below (Figure 2a, 2b).



Field	Description
OPCODE	6 bits primary operation code
D0-D7	5 bits specifies source/destination register or all 8 bits for immediate data

Fig. 2a. Instruction format

Instructions	Description
LD	Loads operand(Bit)
LD	Loads operand(Byte)
LDI	Loads Immidiate data(Bit)
LDI	Loads Immidiate data(Byte)
ST	Store operand in Acc(Bit)
ST	Store operand in Acc(Byte)
NOT	Complement operand(Bit)
NOT	Complement operand(Byte)
ADD	Bit Addition
ADD	Byte Addition
SUB	Bit Subtraction
SUB	Byte Subtraction
AND	Boolean AND(Bit)
AND	Boolean AND(Byte)
OR	Boolean OR(Bit)
OR	Boolean OR(Byte)
XOR	Boolean XOR(Bit)
XOR	Boolean XOR(Bit)
SGT	Skip if Test>
SGT	Skip if Test>
SGE	Skip if Test>=
SGE	Skip if Test>=
SEQ	Skip if Test=
SEQ	Skip if Test=
SLE	Skip if Test<=
SLE	Skip if Test<=
SLT	Skip if Test<
SLT	Skip if Test<
JMP	Jump to lable
JMP	Jump to Immidiate address
CUP	count up
CDN	count down
TON	Timer ON
TOFF	Timer OFF
RES	Reset
END	End of Program
SKIP	Skip instructions

Fig.2b Instruction set of IL processor

III. IL PROCESSOR ARCHITECTURE

After the accomplishment of the instructions set, we proceeded with the design of the data path in such away the IL processor was able to execute each one of the machine codes.

Load instruction always loads the data from any location to accumulator. Store instruction stores data from accumulator to any location. The logical, comparative and arithmetic instructions have in common that one of the operands is always the content of the accumulator. Proposed IL processor architecture adopts direct accessing mode, which is suited for PLC's frequent data accessing which is having 16 I/O's, consists of a control and a data processing parts. The control part is composed of program counter, program memory and instruction decoder. The Arithmetical-Logic unit (ALU) is the core of data processing part. Further parts are input multiplexer; accumulator registers. The program counter generates addresses for program memory. The programs counter increments by 1 after instruction execution. The program memory is a 256x14b ROM type memory, which is addressed by 8 bit address from program counter. Instruction word length is 14 bits. According to instruction the instruction decoder generates control signals. The instruction is composed of following bits: 6 MSB for opcode and remaining 8 bits for immediate data or address purpose.

A. Instruction Fetch Entity

The function of the instruction fetch unit is to obtain an instruction from the instruction memory using the current value of the PC and increment the PC value for the next instruction. This design uses an 14-bit program memory width.. The instruction fetch component contains the following logic elements that are implemented in VHDL: 8-bit program counter (PC) register, an adder to increment the PC by one or two, the instruction memory, a multiplexor, and an AND gate used to select the value of the next PC.

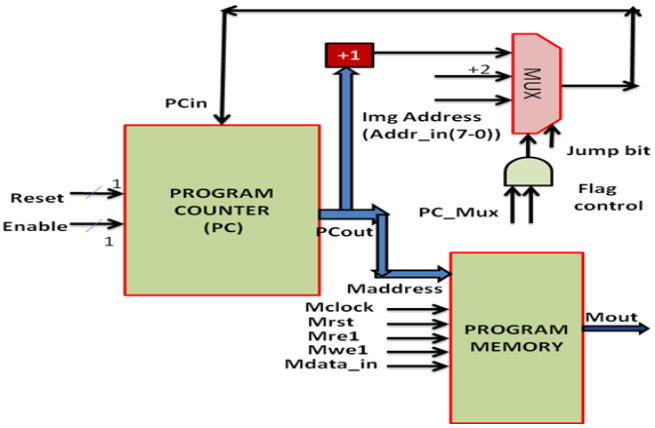


Fig. 3a IL Processor Instruction Fetch Entity

B. Instruction Decode Entity

The main function of the instruction decode unit is to use the 14-bit instruction provided from the previous instruction fetch unit to instruction register . However with our design most significant 6 bit specifies opcode and remaining 8 bits defines source/destination address or immediate data/address. The logic elements to be implemented in VHDL include several multiplexors and the registers.

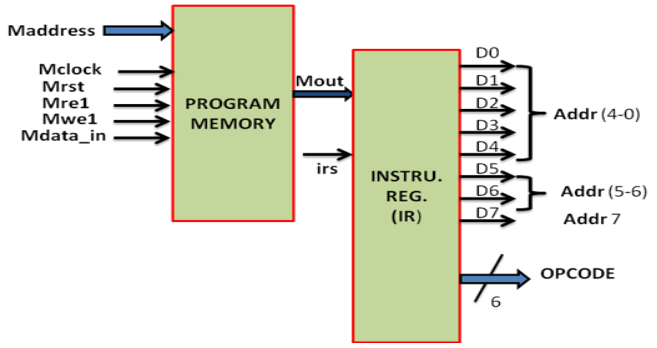


Fig. 3b IL Processor Instruction Decode Entity

C. Execution Entity

The execution unit of proposed processor contains the arithmetic logic unit (ALU) which performs the operation determined by the Alu_opc ode signal generated by control unit module. The logic elements to be implemented in VHDL include a multiplexor, Acc, adder, and the ALU.

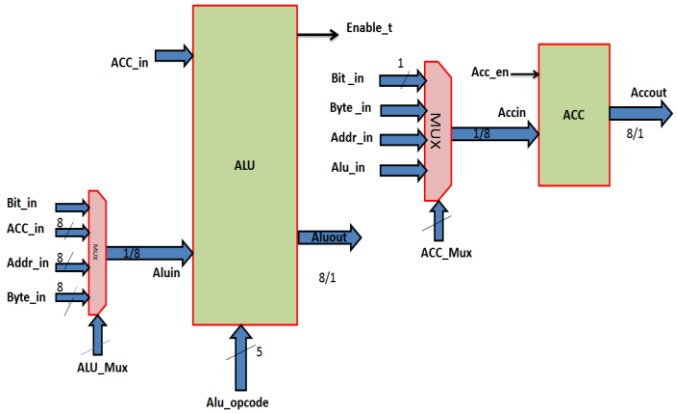


Fig. 3c IL Processor Execution Entity

D. Control Entity

The control unit is a finite state machine (FSM) that governs each one of the operating modes of the microprocessor; for this reason it is considered the most important block for a micro processor. This microprocessor does not use pipeline for execution. The control unit of proposed process or examines the instruction opcode bits [14-9] and decodes the instruction to generate control signals to drive the different modules of the architecture that is to be used to control the flow of instruction execution . This unit works on opcode, cpu clock and cpu reset signal.

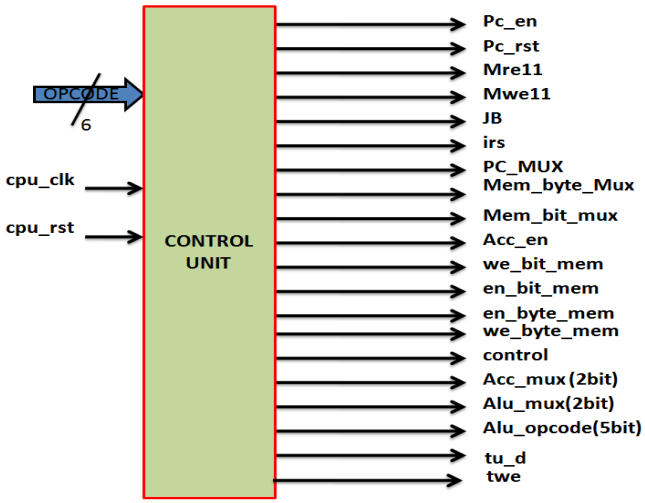


Fig. 3d IL Processor Instruction Control Entity

E. Data Memory Entity

The proposed processor has b it as well as byte memory. Bit data related to 16input and out module of PLC is directly connected to bit memories particular location to avoid extra time required for accessing data form module for execution. There are four memory banks available for temporary storage of data.

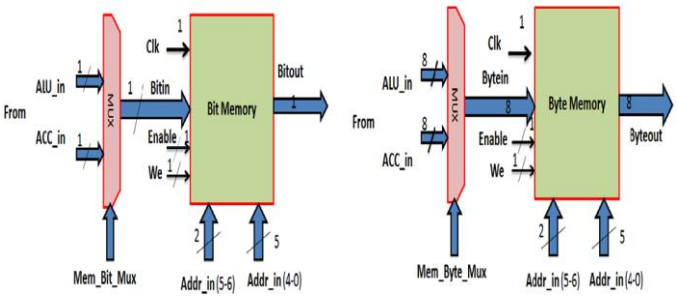


Fig. 3e IL Processor Data Memory Entity

IV. SIMULATION AND SYNTHESIS RESULTS ON FPGA

A completely different architecture was introduced in the mid-1980's that uses RAM-based lookup tables instead of AND-OR gates to implement combinational logic. These devices are called field programmable gate arrays (FPGAs). The device consists of an array of configurable logic blocks (CLBs) surrounded by an array of I/O blocks. The Spartan-3E from Xilinx also contains some blocks of RAM, 18 x 18 multipliers, as well as Digital Clock Manager (DCM) blocks. These DCMs are used to eliminate clock distribution delay and can also increase or decrease the frequency of the clock. Each CLB in the Spartan-3E FPGA contains four slices, each of which contains two 16 x 1 RAM look-up tables (LUTs), which can implement any combinational logic function of four

variables[6]. In addition to two look-up tables, each slice contains two D flip-flops which act as storage devices for bits.

The synthesis of IL processor is d one by using Xilinx ISE webpack. Results related to th at are shown in Fig.4, 5.

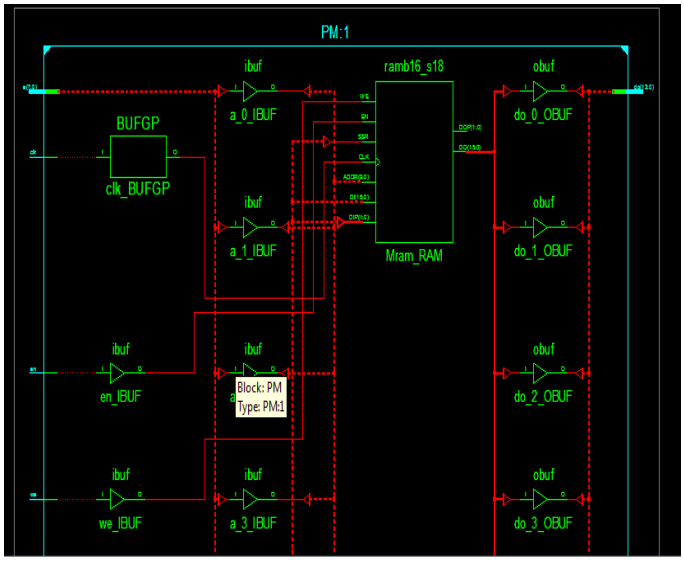


Fig. 4 Synthesis result

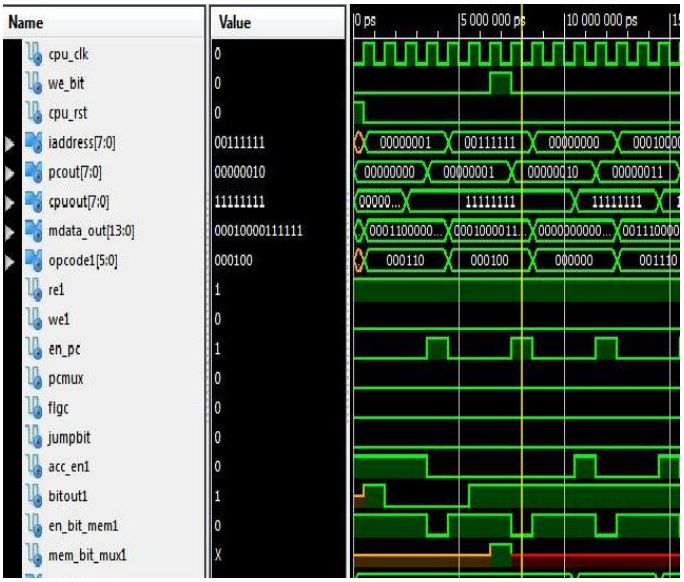


Fig.5. Simulation result

V. CONCLUSION

The number of clock cycles the IL processor takes to execute each one of the instructions is improved as compare to results obtained in the paper "Design and Implementation of an Embedded Microprocessor Compatible With IL Language in Accordance to the Norm IEC 61131-3" by Snaider Carrillo L., Agenor Polo Z., Mario Esmeral P.[7]shown in Fig.5. As number of clock cycles required to execute instruction is less, time required to execute whole program will be less. Therefore execution speed of proposed IL processor fast, which is required for high speed application. Dense architecture is proposed for accomplishment of IL processor on FPGA. Various building blocks, their bringing together and key architectural differences are discussed. It is expected that results obtained will persuade use of similar processor on FPGA for process control and automation.

VI. ACKNOWLEDGMENT

This paper has been conducted under the guidance of our tutor Prof. R.A. Pagare and Prof. V.N.Patil. Also authors would like to thank the department of E&TC Trinity College Of Engineering & Research Pune and RLARD Pune.

REFERENCES

[1] Pravin S. Mane, Indra Gupta, M. K. Vasantha, "Implementation of RISC Processor on FPGA" Computer Science & EngineeringD epartment, Mody Institute Of Technology & Science, Lakshmangarh-332311."Electrical Engineering Department, Indian Institute ofTechnology Roorkee, Roorkee-247667, 1-4244-0726-5/06/$20.00 '2006 IEEE.

[2] Gab SeonRho, Kyeonog-hoon Koo,Naehyuc Chang, Jaehyun Park, Yeong-gi Kim and Wook Hyun Kwon. "Implementation of a RISC microprocessor for programmable logic controllers", Elsevier Science B.V, Microprocessors and Microsystems Volume 19 Number 10 ,December . 1995

[3] Mrs. Shilpa Rudrawar and Prof. Manish M. Patil, "Design of instruction list (IL) processor for process control' In Proceedings of International Journal of Electronics and Computer Science Engineering V1N3-1734-1742,India 2012

[4] IEC 61131 Ageneral overview & emerginh trends IEEE INDUSTRIAL ELECTRONICS MAGAZINE 1932-4529/09/$26.00&2009 IEEE

[5] Shuting-zeng , Zhijia-yang, "High performance architecture design of PLC dedicated processor" Industrial informatics laboratoryShenyang institute of automation, Graduate School of the Chinese Academy of Sciences Shenyang, China 2010 IEEE.

[6] Spartan-3 Generation FPGA User Guide UG331 (v1.8) June 13, 2011

[7] Snaider Carrillo L., Agenor Polo Z. , Mario Esmeral P."Design and Implementation of an Embedded Microprocessor Compatible WithIL Language in Accordance to the Norm IEC 61131-3", Proceedings of the 2005 International Conference on Reconfigurable Computing and FPGAs (ReConFig 2005) 0-7695-2456-7/05-$20.00 © 2005 IEEE.