ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

PLC AUTOMATION USING VHDL PROGRAMMING

[1]NIDHI GOPAL, [2]NAVEEN UPADHYAY, [3]KOUSHIK CHAKRABORTY

[1]M. Tech Scholar, Department of ECE, Jayoti Vidyapeeth Women's University, Rajasthan, India Email- nid0203@gmail.com

[2]Assistant Professor, Department of ECE, Jayoti Vidyapeeth Women's University, Rajasthan, India, Email- naveen@jvwu.ac.in

[3]Head of Department, Department of ECE, Jayoti Vidyapeeth Women's University, Rajasthan, India Email- koushik@jvwu.ac.in

ABSTRACT

CNC Machines in heavy power plants have an important part to play. CNC stands for "Computer Numerical Control." A CNC machine directs a cutting tool, which may range from a small scalpel to a woodcutting blade or router, based on directions that the operator has entered into a computer that controls the movement of the head holding the cutter. A Programmable Logic Controller (PLC) is a microcontroller-based, general-purpose electronic device to control the operation of a machine or process. Contrary to conventional microcontroller based systems, PLCs are not programmed by the device manufacturer but by the machine builder or the end user. Study of CNC Machines, PLC Programming and its practical implementation using VHDL Language has been designed, simulated and discussed in this report.

Study of CNC Machines, PLC Programming and its practical implementation using VHDL Language has been designed, simulated and discussed in this research report. Challenges & design aspects are defined and discussed also, application areas and future of VHDL Based PLC Programming have been proposed and studied under this research and a detailed report has been prepared to summarize the study and research conducted on the above mentioned topic.

INDEX TERMS: PLC Programming, CNC Machines, VHDL Language

1. INTRODUCTION TO PLC

A Programmable Logic Controller is a solid state control system that

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

continuously monitors the status of devices connected as inputs. Based upon a user written program, stored in memory, it controls the status of devices connected as outputs. Basically, PLC implements **logic control functions** by means of a **program.** PLC Programming is generally manipulated by programmable software languages.

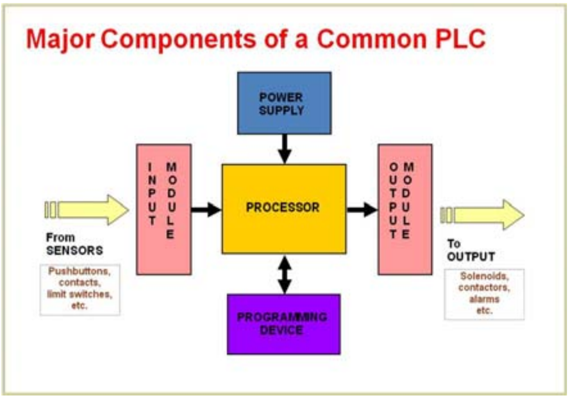
Major Components of a Common PLC
POWER SUPPLY
INPUT MODULE
PROCESSOR
OUTPUT MODULE
PROGRAMMING DEVICE
From SENSORS
Pushbuttons, contacts, limit switches, etc.
To OUTPUT
Solenoids, contactors, alarms etc.

Figure 1: Block Diagram

A PLC matches the NC to the machine. PLCs were basically introduced as replacement for hard wired relay control panels. They were developed to be reprogrammed without hardware changes when requirements were altered and thus are reusable. PLCs are now available with increased functions, more memory and large input/output capabilities. Figure 2 gives the generalized PLC block diagram.

In the CPU, all the decisions are made relative to controlling a machine or a process. The CPU receives input data, performs logical decisions based upon stored programs and drives the outputs. Connections to a computer for hierarchical control are done via the CPU.

The I/O structure of the PLCs is one of their major strengths. The inputs can be push buttons, limit switches, relay contacts, analog sensor, selector switches, proximity switches, float switches, etc. The outputs can be motor starters, solenoid valves, position valves, relay coils, indicator lights, LED displays, etc.

2. LADDER LOGIC PROGRAMMING

Ladder logic was originally a written method to document the design and construction of relay racks as used in manufacturing and process control. Each device in the relay rack would be represented by a symbol on the ladder diagram with connections between those devices shown.

Ladder logic has evolved into a programming language that represents a program by a graphical diagram based on the circuit diagrams of relay logic hardware. Ladder logic is used to develop software for programmable logic controllers (PLCs) used in industrial control applications. The name is based on the observation that programs in this

ISSN: 2348-4098
VOLUME 02 ISSUE 06 JULY 2014

language resemble ladders, with two vertical rails and a series of horizontal rungs between them. While ladder diagrams were once the only available notation for recording programmable controller programs, today other forms are standardized in IEC 61131-3.

3. WHAT IS CNC?

Numerical control (NC) is a method employed for controlling the motions of a machine tool slide and its auxiliary functions with input in the form of numerical data. A Computer Numerical Control (CNC) is a microprocessor-based system to store and process the data for the control of slide motions and auxiliary functions of the machine tools. The CNC system is the heart and brain of a CNC machine which enables the operation of various machine members such as slides, spindles, etc. as per the sequence programmed into it, depending on the machining operations.

The main advantage of a CNC system lies in the fact that the skills of the operator hitherto required in the operation of a conventional machine is removed and the part production is made automatic.

The CNC systems are constructed with a NC unit integrated with a programmable logic controller (PLC) and sometimes with an additional external PLC (non-integrated). The NC controls the spindle movement and the speeds and feeds in machining. It calculates the traversing path of the axes as defined by the inputs. The PLC controls the peripheral actuating elements of the machine such as solenoids, relay coils, etc. Working together, the NC and PLC enable the machine tool to operate automatically. Positioning and part accuracy depend on the CNC system's computer control algorithms, the system resolution and the basic mechanical machine accuracy. Control algorithm may cause errors while computing, which will reflect during contouring, but they are very negligible. Though this does not cause point to point positioning error, but when mechanical machine inaccuracy is present, it will result in poorer part accuracy.

Figure 2: A PLC operated Machine

We can translate PLC programs into VHDL programs. The method only deals with simple logic functions such as AND, OR, NOT, and flip-flops.

The core of implementing PLC based on FPGA is how to convert IEC61131-3 standard language into hardware description language. In the past, converted SFC to Verilog language also the research and conversion to the framework of instruction table translation into HDL has been implemented. But the documents don't implement the conversion of the LD that is widely used, and also have indirect and hidden problems.

This paper gives a strong idea how ladder diagram and its VHDL Implementations are related.

4. STUDY CARRIED OUT FOR CONVERTING LADDER DIAGRAM INTO SYNTHESIZABLE VHDL

4.1DEVELOPMENT OF IDE

Figure below shows the IDE structure for IEC61131-3 control specification to HDL synthesis and mapping of this on target FPGA. Block A in this figure accepts the input control specifications as per IEC-61131-3 in the form of ladder diagram while Block B which is the heart of IDE synthesizes this input specification into VHDL. Block C here represents any suitable FPGA platform with respective support tools to synthesize HDL into corresponding hardware blocks and map on selected FPGA device. Such environment is expected to help in implementing existing as well as new control applications with minimum efforts.

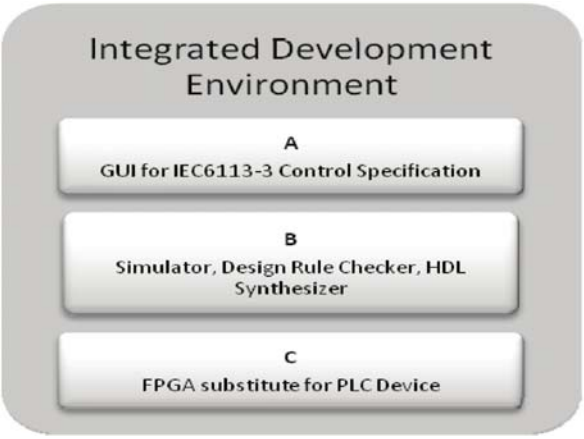
Integrated Development Environment
A
GUI for IEC6113-3 Control Specification
B
Simulator, Design Rule Checker, HDL Synthesizer
C
FPGA substitute for PLC Device

Figure 3: Proposed Development Environment for high speed PLC based Applications

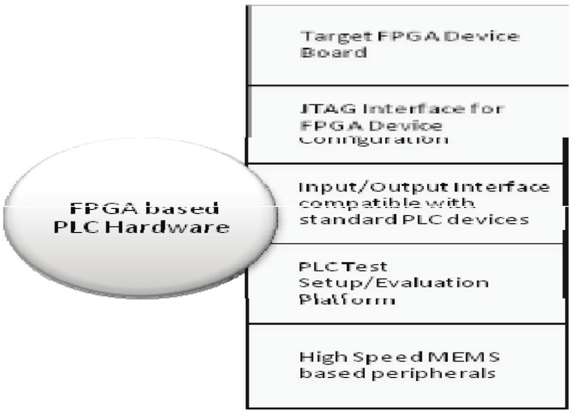
FPGA based PLC Hardware
Target FPGA Device Board
JTAG Interface for FPGA Device Configuration
Input/Output Interface compatible with standard PLC devices
PLC Test Setup/Evaluation Platform
High Speed MEMS based peripherals

Figure 4: FPGA Based PLC Hardware Platform

4.2 SOFTWARE DEVELOPMENT FLOW

The input to the system is invariably

based upon IEC 61131-3 standard e.g. ladder diagram which is a well accepted control specification standard by industry.

Figure 5 shows overview of the software development flow. The entire software development process is divided into four stages. These are as follows:

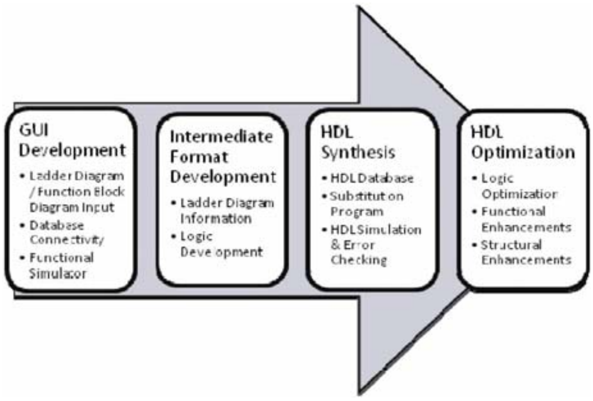
GUI Development
• Ladder Diagram /Function Block Diagram Input
• Database Connectivity
• Functional Simulator
Intermediate Format Development
• Ladder Diagram Information
• Logic Development
HDL Synthesis
• HDL Database
• Substitution Program
• HDL Simulation & Error Checking
HDL Optimization
• Logic Optimization
• Functional Enhancements
• Structural Enhancements

Figure 5: Software Development Flow: Overview

The software thus developed is used as an interface for the user who can create, edit and modify ladder diagram which is automatically converted further into a synthesizable HDL file that represents equivalent hardware model for the defined control specification.

4.3 DETAILS OF SOFTWARE DEVELOPMENT

Considering various features and facilities available in existing PLC development software, the IDE (developed) was expected to have following features:

A. GUI Development to accept ladder diagram entry compliant to IEC 61131-3.

B. Intermediate data base development to maintain and manage the ladder diagram entry with suitable database technology.

C. HDL synthesis and optimization to use this database to create the synthesizable VHDL file to realize digital logic on FPGA for expected control.

D. RTL synthesis and mapping on target FPGA to provide smooth integration to FPGA development environment.

This synthesis algorithm developed to carry out this task is capable of:

(a) Identifying combinational and sequential blocks from control specification,

(b) Synchronizing inputs,

(c) identifying concurrent blocks & their interdependency for sharing common resources

(d) detecting mismatch between simulation and synthesis and

(e) Migrating digital design techniques to RTL synthesis.

The details of the four steps mentioned above as a part of software development are explained below.

A. GUI DEVELOPMENT

The primary objective of developing a GUI was to enable even a non-technical person to create ladder diagram representing required control specifications. At this stage, the IDE developed by the authors supports all ladder diagram components and 99 ladder rungs to represent sufficiently complex discrete state control logic. Figure shows snapshot of the developed GUI. A small part of the complex ladder diagram has been shown as an illustration. This development process passed through following phases before achieving the final results.

1. In phase-I, a GUI prototype of 5 X 8 rung matrix was developed. According to IEC 61131-3, the important constituent elements of a ladder diagram are: Input Relay, Output Relay, Memory, Holding Bits, Timer, Counter, Horizontal connections and Vertical connections. All these were taken into consideration in the first phase of GUI development.

2. In phase-II, database connectivity was created for the input of the ladder diagram with full reproducibility. This, of course, was subjected to thorough error checking in terms of ladder rules, design rules and connectivity rules.

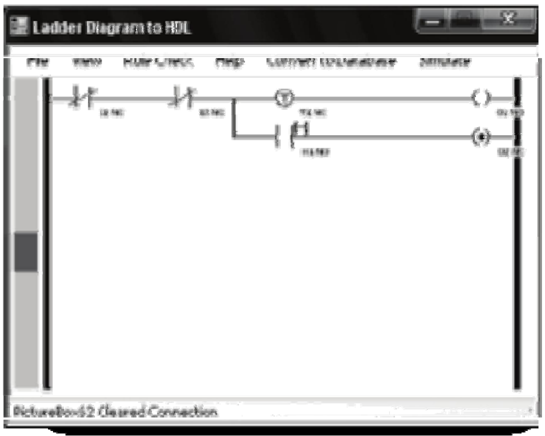

Figure 6: Ladder to HDL Converter: GUI

3. In phase-III, a simulator program was developed to verify expected functionality of ladder diagram entered by the user. Simulator interprets the ladder diagram in terms of its equivalent Boolean expressions to accept user stimulus for updating results on the screen.

This helps user to verify the control logic before implanting on final hardware. Functionality is verified in terms of logic as well as recurrence & behavioral discrepancy. Figure 5 shows the snapshot of the simulator. User can click on the respective input to change its value and observe the effects on the screen.

B. INTERMEDIATE DATABASE DEVELOPMENT

Objective of this stage was to generate optimum possible database to store the ladder diagram information (MS Access) which subsequently was used to generate VHDL file. The module reads and substitutes the ladder diagram information so as to generate a VHDL file with expected coding style. This intermediate database file is designed to capture all required aspects of the control specification and correctly represent the ladder logic.

C. HDL SYNTHESIS AND OPTIMIZATION

Discussed above, this intermediate file captures all required control features from the ladder diagram including number of variables, logic equations and sequence data input in coded form. The pre-generated synthesizable VHDL snippets are available in the form of another database at the backend for direct synthesis. A generator program is used to fetch the keywords and pre-coded HDL snippets from library database to substitute in output file, forming a valid synthesizable VHDL code.

D. RTL SYNTHESIS AND MAPPING CONTROL LOGIC ON TARGET FPGA

The optimized and synthesizable VHDL output obtained from the developed synthesis tool can be ported into FPGA development environment for RTL synthesis and mapping of ladder control logic on target FPGA hardware through any freely available FPGA development environment. As synthesizable VHDL is IEEE and IEC standard, it is well accepted by all such development environments. Xilinx ISE Design Suite was used by in the experimentation for implementation purpose and for creation of the bit files which were used to configure FPGA.

5. CONCLUSION

As a future scope of this, it is proposed to develop PLC validation platform (electromechanical / hydro-pneumatic test set-up to establish bench-mark applications) for PLC performance evaluation. Such test results would be helpful in comparing performance of PLC designed and developed through this IDE with other PLCs available commercially in the market. Presently it is necessary for plant engineer to have working knowledge of FPGA design environment for implementing PLC on target board. However it is proposed to reduce this burden by writing script file in IDE development environment that can automatically invoke the FPGA development environment on a single button click through GUI. Further, it is proposed to add function block entry, expand existing version with increased row and column for ladder diagram editor to support more flexibility in terms of size and complexity. Ongoing research on identifying effective output VHDL coding style and post RTL synthesis verification is also expected to add further strength to this IDE.

Also, PLC, based on FPGA, can greatly improve the speed of logic control. In this report, the conversion from IEC61131-3 standard language to hardware description language was researched and software converting the ladder diagram into VHDL program was developed. The software implemented the generation of ladder diagram structure, Boolean equivalence and VHDL program. The control of tool magazine was realized by PLC based on FPGA. Research shows that PLC, based on FPGA, plays an important role in promoting the development of high-speed PLC control.

REFERENCES

1. **Wikipedia : Google**

2. **Research papers:**

(a) Designing an efficient Programmable Logic Controller using Programmable System On Chip By Raja Narayanasamy, Product Apps Manager Sr, Cypress Semiconductor Corp. – Published in EE Times Design.

https://www.element14.com/community/servlet/JiveServlet/previewBody/28092-102-1-77293/Low%20Cost%20PLC%20-%20article.pdf

(b) Research on FPGA-based Programmable Logic Controllers'Technology - Published in: TELKOMNIKA, Vol. 11, No. 12, December 2013, pp. 7655~7663 e-ISSN: 2087-278X.

http://www.iaesjournal.com/online/index.php/TELKOMNIKA/article/viewFile/3701/pdf

(c) A Study on a Hardware Translation Tool for PLC Programs

Published in: IEEE: http://www.hindawi.com/journals/ame/2014/127618/

(d) Novel Integrated Development Environment for Implementing PLC on FPGA by Converting Ladder Diagram to Synthesizable VHDL Code Published in : IEEE.

http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5707833&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D5707833

3. VHDL Books

4. Past presentations of the topic

5. Reading Material and Documents provided by External Guide

6. Google Books

ACKNOWLEDGEMENT

The authors would like to thanks to Dr. Shobha Lal, Dean, Faculty of Engineering, Jayoti Vidyapeeth Women's University, Jaipur and Mr. Koushik Chakraborty, Head of Department, Jayoti Vidyapeeth Women's University, Jaipur and faculties of ECE Department for his continuous support and encouragement.

BIOGRAPHIES

Nidhi Gopal did her B. Tech in ECE from Jayoti Vidyapeeth Women's University, Jaipur. She has recently completed her M. Tech dissertation work. Her area of interests are Image processing, VLSI etc.

Naveen Upadhyay, B.E. and M. Tech in ECE. Presently he is working as assistant professor in JVWU, Jaipur. He is a member of IEEE Communication Society. His area of interest are antenna, mobile communication etc.

Koushik Chakraborty, presently working as Head of Department in JVWU, Jaipur. His area of interest are antenna, mobile communication etc.