ISA Transactions 54 (2015) 156–168

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Programmable logic controller performance enhancement by field programmable gate array based design

Dhruv Patel [a,*], Jignesh Bhatt [b], Sanjay Trivedi [c]

[a] Department of Instrumentation and Control Engineering , Sardar Vallabhbhai Patel Institute of Technology (SVIT), Vasad 388306, Gujarat, India
[b] Department of Instrumentation and Control Engineering, Faculty of Technology, Dharmsinh Desai University, Nadiad 387001, Gujarat, India
[c] Space Application Center (SAC), Indian Space Research Organization (ISRO), Ahmedabad 380015, Gujarat, India

ARTICLE INFO

Article history:
Received 20 August 2013
Received in revised form
27 August 2014
Accepted 30 August 2014
Available online 14 October 2014
This paper was recommended for publication by Prof. A.B. Rad.

Keywords:
Automation
Embedded System
FPGA
GUI development
Instrumentation
PLC

ABSTRACT

PLC, the core element of modern automation systems, due to serial execution, exhibits limitations like slow speed and poor scan time. Improved PLC design using FPGA has been proposed based on parallel execution mechanism for enhancement of performance and flexibility. Modelsim as simulation platform and VHDL used to translate, integrate and implement the logic circuit in FPGA. Xilinx's Spartan kit for implementation-testing and VB has been used for GUI development. Salient merits of the design include cost-effectiveness, miniaturization, user-friendliness, simplicity, along with lower power consumption, smaller scan time and higher speed. Various functionalities and applications like typical PLC and industrial alarm annunciator have been developed and successfully tested. Results of simulation, design and implementation have been reported.

1. Introduction

PLCs have been quite widely utilized for automatic control of manufacturing units, especially using binary, sequential, logic, analog as well as hybrid control algorithms [1]. PLC has been a user-configurable, microprocessor based specialized computational device, to handle different types of control functions and levels of complexities [2]. PLC can monitor and control the process using its output devices that can either go on/off (also known as discrete or digital outputs) or vary proportionately (also known as analog outputs). Conventional PLCs could be quite complex and are usually built around 486 and/or Pentium based processors with plenty of analog and digital I/Os. Early PLCs were designed to replace relay logic systems. These PLCs were programmed using "Ladder Logic" or "Ladder Diagram" (LD) that strongly resembles a schematic of relay logic. Modern PLCs could be programmed in a variety of ways, from ladder logic to more traditional programming languages such as BASIC, C, etc. In industrial applications, LD is one of the most popular programming method for customized PLC configuration development. Traditionally, a typical PLC includes a microprocessor and, LD is sequentially executed inside the microprocessor in a cyclic scan period. Hence, PLC performance is limited by the cyclic scan period, which depends upon program length and processing speed of the microprocessor. PLCs are also referred to as Programmable Controllers or Sequential Processors as devices as they handle step by step execution of sequence of operations for commercial and industrial applications. Their superior reliability and robustness along with greater flexibility have made them particularly attractive to designers of industrial automation systems. However, many a times, it has been observed that conventional PLC designs are not powerful enough to cover the performance needs of certain real-time applications. In such situations, possibility of use of low-cost reconfigurable hardware components such as an FPGA has been explored [3].

The main focus of the work presented has two objectives: (a) To analyze the LD program and organize it with sequential and parallel structure and (b) To implement the sequential and parallel structure of the LD program with HDL inside FPGA.

Abbreviations: ALU, Arithmetic and Logic Unit; ASLC, Application Specific Logic Controller; CPN, Coloured Petri Net; CPU, Central Processing Unit; FBD, Functional Block Diagram; FPGA, Field Programmable Gate Array; FLC, Fuzzy Logic Controller; FSM, Finite State Machine; GUI, Graphical User Interface; HDL, Hardware Description Language; I/O, Input/Output; IL, Instruction List; LD, Ladder Diagram; MEMS, Micro-Electro-Mechanical Systems; PLC, Programmable Logic Controller; SFC, Sequential Function Chart; RLC, Reconfigurable Logic Controller; RPS, Reactor Protection System; VB, Visual Basic; VHDL, Very High-speed integrated circuit Hardware Description Language

* Corresponding author.
E-mail addresses: dmpatelic@gmail.com (D. Patel), jigneshgbhatt@gmail.com (J. Bhatt), sanjay@sac.isro.gov.in (S. Trivedi).

http://dx.doi.org/10.1016/j.isatra.2014.08.019
0019-0578/

The remaining paper has been organized as follows. Section 2 includes FPGA under Section 2.1, while the FPGA based PLC design approach, proposed design architecture and GUI development have been included under Section 2.2. Section 3 includes results of literature survey to summarize and contrast the works carried out in the same area. Under Section 4 for validation and testing of the proposed design approach, two applications – (i) Typical PLC Ladder Logic Functionality Application and (ii) Industrial Alarm Annunciator Application, have been presented along with brief summary and information regarding simulation and test results. Synthesis report for the proposed design marks the end of Section 4 with details of device utilization. Finally, the paper ends with conclusions and list of references utilized.

2. Materials and methods

2.1. Materials

2.1.1. FPGA

An FPGA consists of an array of uncommitted elements that could be interconnected in a general way. Like typical ladder logic, the interconnections between elements are user programmable. FPGAs were introduced in 1985 by Xilinx Company. Since then, different FPGAs have been developed by different manufacturers – such as Actel, Altera, Plessey, Plus, Advanced Micro Devices, Quick Logic, Algotronix, Concurrent Logic, Crosspoint Solutions, etc. [4]. FPGA being a semiconductor device, could be configured by the customer or designer post-manufacturing as well, hence popularly known as “re-confogurable hardware” as well as “field programmable devices”. FPGAs are programmed using a logic circuit diagram or a source code in an HDL to specify how the chip would work. FPGAs contain programmable logic components called “logic blocks” and an hierarchy of reconfigurable interconnections that allow the blocks to be “wired together”, somewhat like one-chip programmable breadboard. Logic blocks could be configured to perform complex combinational functions, or merely simple logic gates like AND, XOR, etc. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory. FPGAs have been quite popular today for their parallel execution mechanism and reconfigurable hardware structure [5]. Hence, we propose the approach of utilization of FPGAs instead of conventional μp and μc for PLC design on account of its potential for significant performance improvements due to FPGA's parallel instruction execution capabilities.

2.2. Methods

2.2.1. FPGA based PLC design approach [7]

Our suggested approach presents design using which a general purpose Micro-PLC could be realized using an FPGA implementation. Once FPGA is properly configured, along with suitable human interface or GUI, the design demonstrates capability to function as Micro-PLC with satisfactory performance and reasonable flexibility. Ladder program could be input in Micro-PLC using its programming mode through dedicated ladder programming software. Ladder program debugging could be carried out by the same software itself. Thus, our suggested approach necessitates no special training requirements for the plant engineers, as it employs pre-configured FPGA as conventional Micro-PLC.

The pilot model of the proposed design has been developed and presented in [6] as well as discussed at length in [7]. The design methodology of proposed design has been shown in Fig. 1. By using serial communication, the hex codes of 52-bit instructions data have been transmitted using GUI to an FPGA. This 52-bit data consist coded information regarding selection of particular rung, components and component inputs. For a particular rung, such transmitted hex codes have been stored in a text file first and received inside FPGA by using UART decoding logic later. After receiving 52-bit instruction codes separately for each rung, the corresponding logic has been dumped into RAM block of FPGA. The depth × width dimensions of RAM block are 32 bits × 52 bits respectively. Out of 52 bits, 4 bits are for selection of component operation and 48 bits are for selection of component inputs.

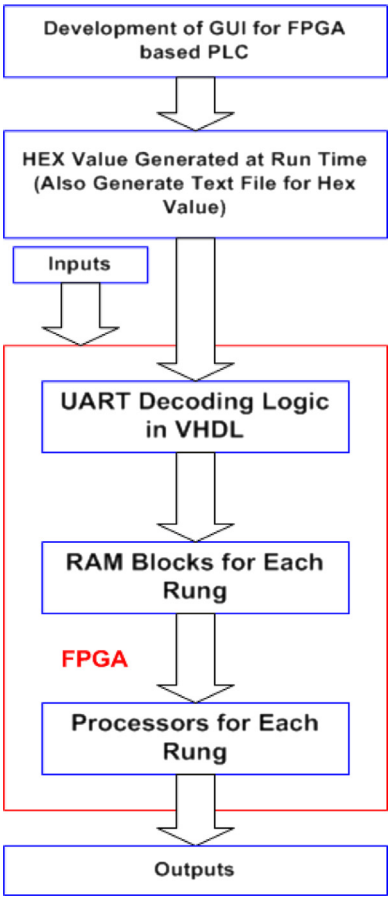

Fig. 1. Design methodology for proposed FPGA based PLC design [7].

2.2.2. Proposed design architecture [7]

The architecture of the proposed design has been shown in Fig. 2. The proposed design supports 4 rungs with 4 inputs and one output corresponding to each rung. In the design, total 12 components such as AND, OR, Addition, Subtraction, Move, Shift Left, Shift Right, Rotate Left, Rotate Right, Compare Equal, Compare Greater and Compare Less functions have been used. Each rung can contain maximum 16 components, which could be selected from combination of the 12 components mentioned above along with 4 registers. Each component input has capability of handling four external inputs and/or outputs. AND and OR type components can handle maximum six inputs, which can also be outputs from other rungs. For example, if AND or OR components have used three inputs, then remaining three input components shall have default logic (i.e. remaining three input components have all 1's bits). ANDing and ORing components require total 52 bits for operation. The Addition, Subtraction, Move, Compare Equal, Compare Greater, Compare Less components require only 12 bits for operation, hence 40 bits remains unutilized. Among them, the Most Significant Bit (MSB) from the 1st one to 4th bits have been utilized for selection of components; 5th to 8th bits have been used as addresses of source register and last four bits have been used for