# SELF ERROR DETECTING PLC ON FPGA

**Harshala Patil( 1) ,** ***AssistantProf*** **Shweta Ashtekar(2),*AssistantProf* Amruta Chintawar(3)**
Electronics Department , Mumbai University
RAIT College of Engineering(1)(2)(3) Nerul, Navi Mumbai,Maharashtra,India
harshalapatil19@yahoo.com, shweta_sa06@yahoo.com, amruta_chintawar239@yahoo.com

**Abstract - Today's automotive industry are demanding for flexible and capable of producing multiple vehicles in multiple variations on a single line. Such automation also requires guaranteed output with safety critical consideration. With the increase in flexible manufacturing comes more parts, more variation and programs, and more interfaces to robots and other devices. To maintain continual operations without reprogramming all of the equipment or reconfiguring entire control architecture to ensure output and productivity remain high. This paper discusses about the implementing offline failure detecting method of any Programmable Logic Controllers (PLCs) and the test result. It further discuss about self-error detection PLC on FPGA .**

***Keywords— Ladder Diagram, Logic Analyser, PLC, State Machine, VHDL.***

## I. INTRODUCTION

Today many of the industries uses devices to control and automation PLCs that exist in complex automation solution e.g. automation industry, semiconductor industry, food and medicine industry, etc. In the automobile industry in each production line there are various parts that will be assembled in the car considering various parameters like detecting welds in the given job, cutting position, angle and position of automobile chasses, nuts positions, etc. performing all these activities the production line should be flexible so that if there is any change in the parts we should be able to perform it. So while increasing in the production there was some random failure problem in the PLCs. The PLC failure reasons was unknown, and as the system get failed there is a high risk for the industry, there workers and financial loss. To detect the fault in the system we have developed an offline testing unit which consists of an emulator of various inputs and those inputs are feed to the PLC[1]. The emulator provides these inputs to PLC at various speed and there parameters are monitored using logic analyzer.

## II. SIGNIFICANCE OF OFFLINE PLC TEST UNIT

A PLC is a microprocessor-based controller with multiple inputs and outputs. It uses a programmable memory to store instructions and carry out functions to control machines and processes [2]. It is programmed with help of ladder diagram which has many rungs containing electro-mechanical inputs and outputs. It provides stability, accuracy and human safety, so it act as the central controlling unit in most of the automation industry. It decides the controlling action that has to be executed while running the complex system. The inputs and outputs of the PLC are provided the protection to isolate the microprocessor from getting it damage from electrical fluctuation. Field Programmable Gate Arrays (FPGAs) are integrated circuits that enable designers to program customized digital logic in the field. FPGAs are given this name because they are structured very much like a gate array ASIC. As we all know, like an ASIC, the FPGA consists of a regular array of logic, an architecture that lends itself to very complex designs. FPGAs are pre-fabricated silicon devices that can be electrically program to become almost any kind of digital circuit or system. They provide a number of convincing merits over fixed-function Application Specific Integrated Circuit (ASIC) technologies such as standard cells[3]. The important thing to note about the FPGA architecture is its regular, ASIC-like structure. This regular structure makes FPGAs useful for all kinds of logic designs. Many newer FPGA architectures are incorporating complex devices inside their FPGAs. Reprogrammable FPGAs are capable of dynamically changing their logic and interconnect structure to adapt to changing system requirements. This offers a new computing paradigm, which blurs the traditional lines between hardware and software[4]. The automation in the automobile industry are present in many ways. The production line are having various functionality in the task like assembly of various parts, positioning and inclination of the part, etc. While performing all this activities at the same time the production rate also have to be very high because of the market needs. To maintain the good production rate the field of automation plays a very important role. As the production rate is increased the speed of operation and their controlling action also increases. This impacts on the speed of the operation of controlling unit, but because of the increased speed the performance also changes. In our application the controlling action of PLC didn't gave the desired performance. The performance variation was very random.

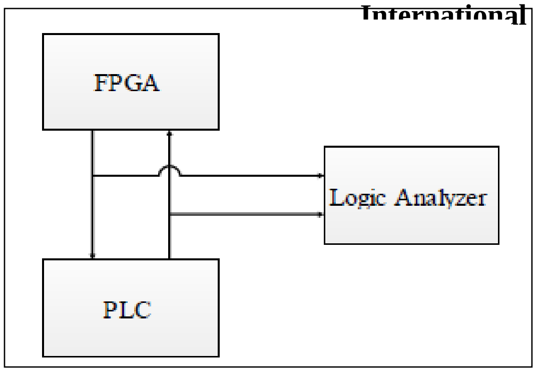

Figure 1. System Block Diagram

This was causing increase in the risk for the industry as well as their workers. This was a growing problem for that industry, as they don't want to take more risk toward their worker. While studying there system we have developed this offline testing unit that was able to find the exact problems. This unit helped them to find the problem and later the errors was removed which ultimately reduce the risks[5].

## III. SYSTEM HARDWARE DESIGN

Figure 1 shows the system block diagram in which it is depicting the important blocks of the system as FPGA, Logic Analyzer and PLC. The FPGA is used as an emulator which generate different test inputs for the offline PLC. PLC will generate the output as per the emulated input signals. The operation of PLC is being monitored on logic analyzer. The speed of emulated signal can be easily controlled by the user using FPGA with the help of slider switches present on it.

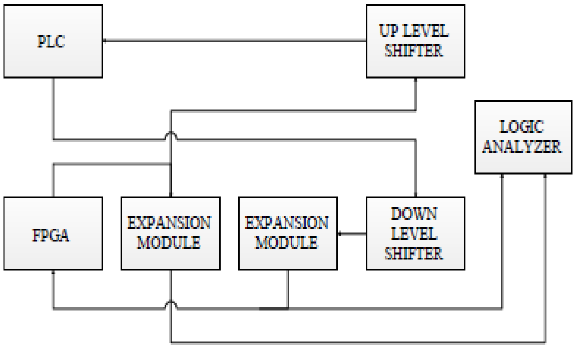

Figure 2. System Architecture

Figure 2 shows the architecture diagram of the system.
The input and output signals generated by the FPGA are TTL level and for PLC it is 24V. As these two voltage logics are not compatible with each other, we need to introduce up and down voltage level sifter. FPGA is having all possible test sequences and it is generated at specific production rate. This production rate is being varied variable slider switch combinations. Now these input sequences are fed to the PLC to generate respective outputs according to the ladder diagram. These outputs are observed on logic analyzer as well as it is fed t FPGA again to display on LEDs present on the FPGA board so that the user can see the changes in the output sequences[7]. Figure 3 shows the actual hardware set-up.

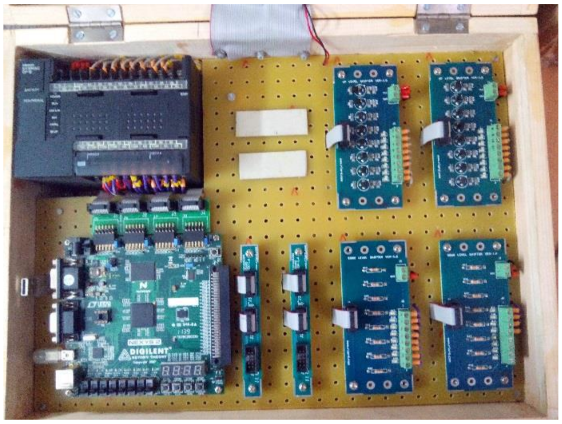

Figure 3. Hardware Setup

## IV. SYSTEM SOFTWARE DESIGN

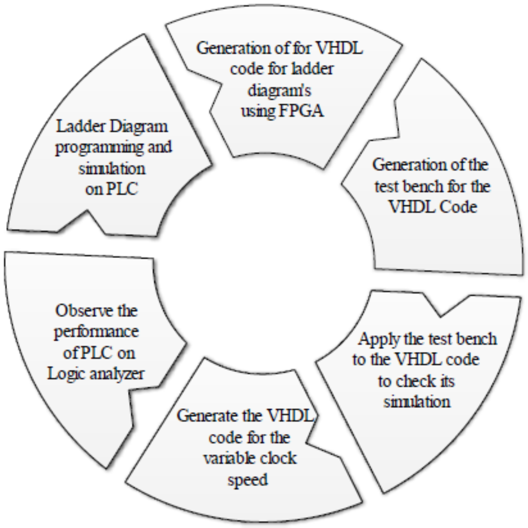
Generation of for VHDL code for ladder diagram's using FPGA
Generation of the test bench for the VHDL Code
Apply the test bench to the VHDL code to check its simulation
Generate the VHDL code for the variable clock speed
Observe the performance of PLC on Logic analyzer
Ladder Diagram programming and simulation on PLC

Figure 4. Methodology

Figure 4 show the implemented methodology for the offline PLC test unit. First we have studied PLC ladder diagram programming, its implementation and simulation on PLC. We have implemented ladder diagram for a small application and its performance analysis on it. Then we have created VHDL code for the inputs of PLC at variable operating speed. These emulated signals were generated by the help of FPGA. Before actually feeding these signals to PLC we have actually simulated these programs using VHDL test bench on ISim simulator. After providing these input sequence to PLC we have monitored and analyses the output sequences using logic analyzer[8].

Figure 5 shows the flowchart of the software program developed to generate the test sequences at variable clock speed measured in Hz.

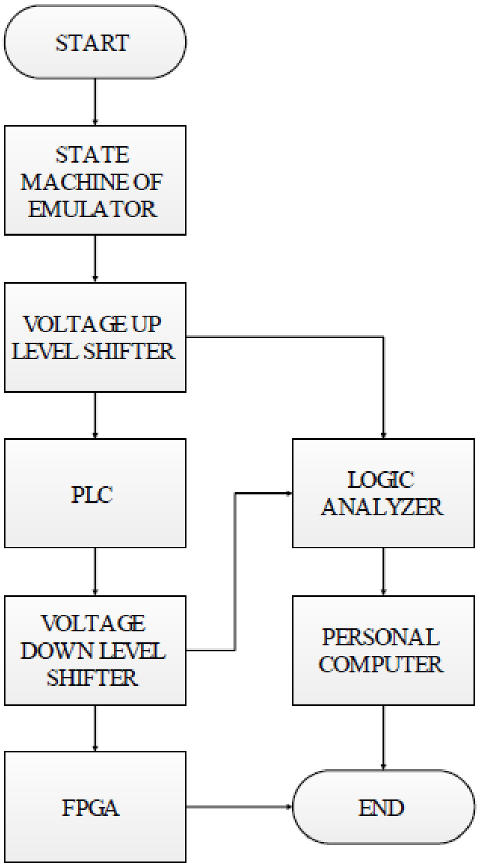

Figure 5. Flow Chart

## V. CASE STUDIES

For the confirmation of our process and to check whether we are going in correct direction along with the automation process in production line we discussed above we took three more case studies as follows:

1. Car parking
2. Digital lock
3. Vending Machine

Systems hardware and software design is same as discussed in above two sections. We use Omron CP1E-N30DT1-D PLC and Spartan 3E module for the experiment. Complete actual process is emulated using the experimental setup. Test cases were generated using FPGA module. The FPGA module provides the necessary stimulus to the PLC. We observe the complete process using a logic analyzer. The stimulus and response waveforms those were observed too.

### *A. Car parking*

This case includes the automatrion of two parking spaces .Automation is done by following description**.**

1. When cars are parked at both spaces 1 and 2 then o/p parking full.
2. When cars are not parked at both or any one space between 1 and 2 then o/p space available for parking .
3. When car is at entry gate and space is available for parking then entry gate is opened.
4. When car is at exit gate then exit gate is opened **.**

### *B. Digital lock*

Description
1. After start is pressed enter the password .
2. After start enter digit 1.
3. If digit1 matched enter digit 2.
4. If both digits matched then password is right else wrong password.
5. When password is matched and enter key pressed then lock get opened.

### *C. Vending machine*

Description
1. When coin is inserted and diameter and weight is matched then coin is accepted else rejected .
2. After acceptance of coin soft drink filling starts.
3. When quantity is finished or enough button pressed then beep sound and filling stops.

## VI. RESULT& ANALYSIS

Figure 6, 7 and 8 shows the output observed on logic analyser. Where X axis is the time in milliseconds. Y axis has various digital channel numbers. D0, D1 and D2 are emulated test pattern for PLC generated by the FPGA. D3 is variable clock set by the user using slider switch. D4-D8 are variable output sequences generated by the PLC.

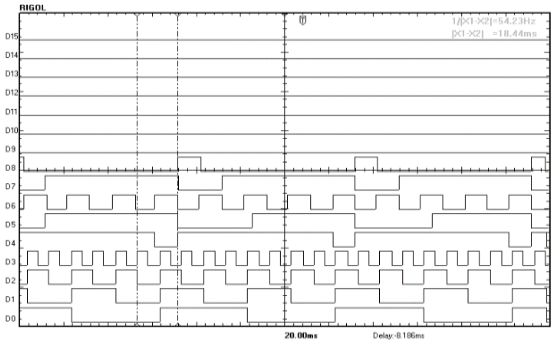

Figure 6. Output at 100 Hz

From the figure 6 and 7 we have observed that PLC has not generated the output at the same instant of input applied due to which there are variable delays at variable output clock speed which also affects the speed of operation of production line.

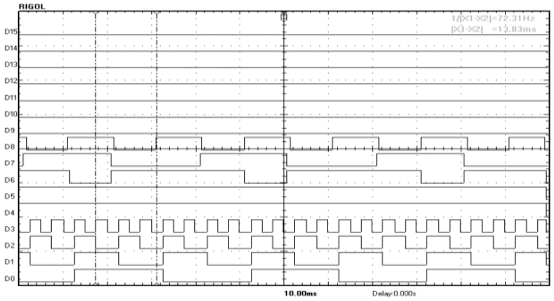

Figure 7. Output at 200Hz

From figure 8 we can see that, at 300 Hz or more than that of the clock PLC fails to generate the output though FPGA generates input sequences continuously and there is no output change by PLC i.e. D4-D8 are freeze. This causes the control failure of production line.

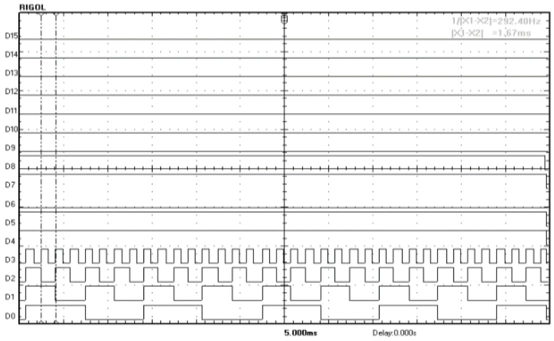

Figure 8. Output at 200Hz

Similarly the results for different case studies was observed on logic analyser.

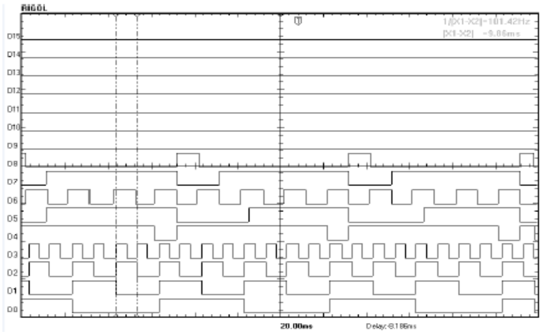

Figure 9. Car parking output at 100Hz

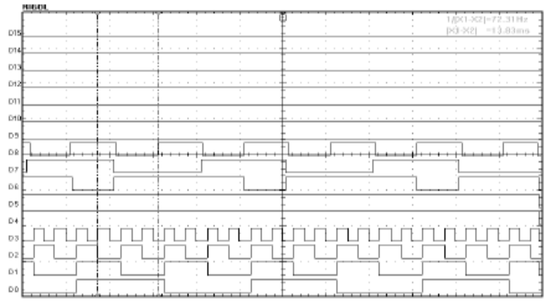

Figure 10. Car parking output at 200Hz

Figure 9, 10 , and 11 shows the results observed on logic analyser for car parking. We can also see the delay that PLC generates and further increasing the frequency or speed of input PLC fails to respond. Such failure or delays are harzardous in real time system.

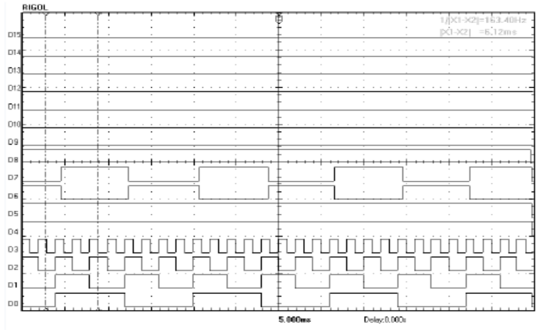

Figure 11. Car parking output at 300Hz

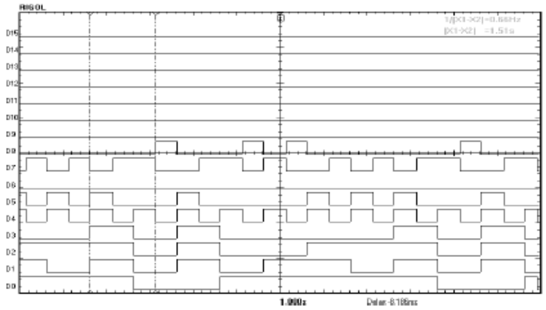

Figure 12. Digital lock output at 100Hz

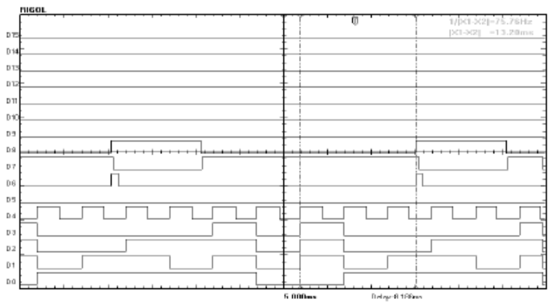

Figure 13.digital lock output at 200Hz

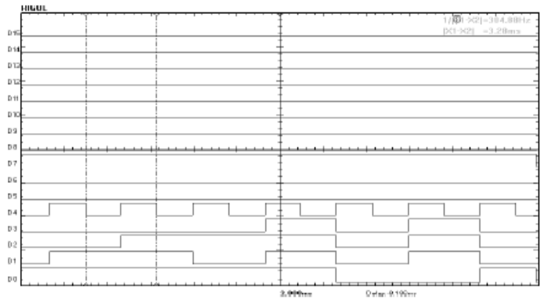

Figure 14. Digital lock output at 300Hz

Figure 12, 13, 14 shows digital lock ouput on logic analyser. Here PLC froze at 300Hz.

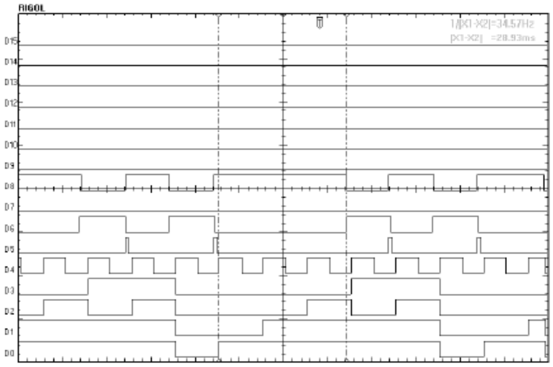

Figure 15. Vending machine output at 100Hz

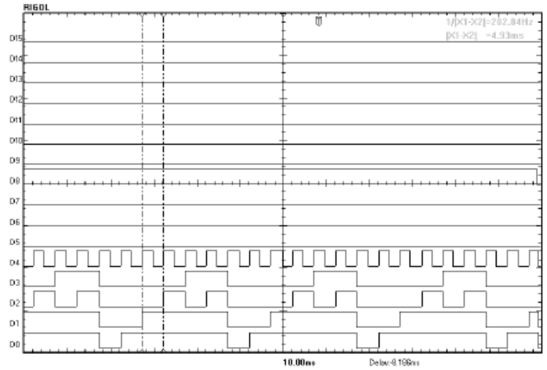

Figure 16. Vending machine output at 200Hz

Figure 15 and 16 shows vending machine output on logic analyser. For this case PLC did not respond for 200Hz itself.

With the help of offline PLC test unit we have detected that after a specific production rate the PLC was not able to control the different devices. Due to the delays generated by the PLC in the controlling action, the production line in the automation processes was not efficiently performing its task. Such kind of errors can be easily analysed or monitored by our system that help plant engineer to minimize the errors generated in the actual production line.

## VII. CONCLUSION

From this case study we can see there are certain limitations in PLC while operating at high production rate due to which random errors are generated during production and there is a high risk for industry, the workers and financial losses. These errors are generated because of the internal architecture of PLC, as it is microprocessor based system the execution of rungs are done in a sequential manner and it fails to perform parallel process.

## Acknowledgements

The successful completion of any master thesis requires a lot of guidance and assistance .I take this opportunity to express my deep sense of gratitude and sincere thanks to my project guide Prof Shweta Ashtekar and Co guide Amruta Chintawar for spending their valuable time and guidance given to me for successful completion of my project work.

## REFERENCES

[1] Kevin Collins," PLC Programming for Industrial Automation ", Sixth edition, Triangle Research International, 2006.
[2] Frank D.Petruzella", Programmable Logic Controllers", Fourth edition, McGraw-Hill Science/Engineering/Math, 2010
[3] W.Bolton," Programmable Logic Controllers ", Second edition, Newnes, 2009
[4] Douglas Perry,” VHDL: Programming by example”, Fourth edition, McGraw Hill, 2002.
[5] Park Chang Mok, Bajimaya Sachin Man, Park Sang C., Wang Gi Nam, Kwak Jong Geun, Kwan Hee Han, Chang Minho, “Development of Virtual Simulator for Visual Validation of PLC Program”, International Conference on Computational Intelligence for Modelling Control and Automation,2006 IEEE.
[6] ThapaDevinder, Park Chang Mok, DangolSuraj, and Wang Gi-Nam, “III-Phase Verification and Validation of IEC Standard Programmable Logic Controller”, International Conference on Computational Intelligence for Modelling

Control and Automation, Web Technologies and Internet Commerce, 2006 IEEE.

[7] Cheon S.W., Lee J.S., Won K.C.K, Kim D.H., “The Software Verification arid Validation Process for a PLC-based Engineered Safety Features-Component Control System in Nuclear Power Plants”, The 30th Annual Conference of the IEEE IES, Busan, Korea, Nov. 2004, pp.827-831

[8] Gourcuff Vincent, De Smet Olivier and Faure Jean-Marc, “Efficient representation for formal verification of PLC programs”, Proceedings of the 8th International Workshop on Discrete Event Systems Ann Arbor, Michigan, USA, July 10-12, 2006 IEEE, pp. 182-187

[9] Kajihara Shigeru, Ono Masahide, Houzouji Harumi, Tamishi Hajime, Takayanag Yoichi, “Development and Products of the Object-Oriented Engineering Tool for the Integrated Controller based on IEC 61131-3”, SICE Annual Conference in Sapporo, Hokkaido Institute of Technology, Japan, August 2004, pp. 1952-1956.