EE3025 : IDP - Production line control

Akshita Ramya
ES16BTECH11012

Krupakar Reddy
ES16BTECH11008

Abstract

In this project titled Production Line Control we are implementing a proof of concept of an actual production line in a factory. We integrate various units found in the factory like a test shop to test the working and components of the model. A rectifier shop is placed to correct any errors if found. Finally, a random number generator is used to provide the product with a unique product code.

1 Introduction

In this project, we are implementing a proof of concept production line control system. In this, we include typical components like a test shop which tests the performance of the components and gives us an error report. This error report is then sent to the rectifier shop for rectification/replacement of faulty components. We also have a 4-bit random number generator to give our final product a unique code. The initial value for the random number is seeded at the time of the product and parts code being fed. This can be understood as a time stamp for the production of the product. The final 8-bit code gives us information about the product specifications. Bit 3-0 give us the unique random number code and bits 7-4 give us the information of the components in the product.

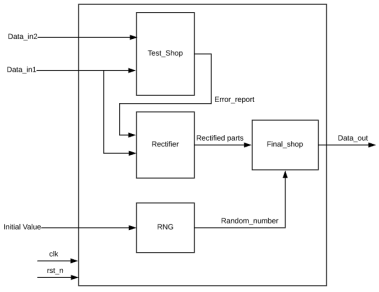

Figure 1: Overall Structure

2 Implementation

All the software and/or packages used for this project are open source. Detailed instructions for installations are given in the README file on GitHub. The source code and compiling instructions can be found here
`https://github.com/akramya/IDP_EE3025`

The final product code is a 8-bit binary number with the first 4 bits giving information about the the parts in the model and the and last 4 bits being the random 4-bit number generated by the random number generated . The implementation is divided into four sections.

2.1 Test Shop

Here the unit takes in two inputs and we implement a XOR function between them to give us the output. The idea of writing a unit which basically implements a xor function rather than use the built in function is to ensure better handling of the error cases. As you can see we have put in a If else if else block. While the if and else if parts handle the usual cases of the data inputs the else part handles the case where if the system is not able to determine if the inputs are not either 1 or 0 . this ensures better exception handling of the system being used.

2.2 Rectifier Unit

The rectifier unit acts as the basic unit of the rectification shop. In the rectifier unit we take in inputs from the error report produced by the test shop and based upon that we correct product code which signifies corrections being made in the components which are found to be faulty .

In the code we see that if the error input bit is 1 then the component works fine and need not be replaced. If the error bit is 0 then the component is faulty and needs to be replaced. This is signi-

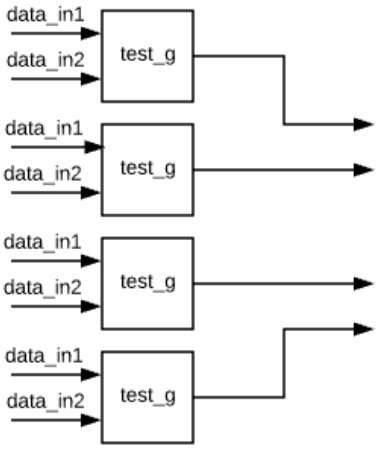

Figure 2: Test Shop

fied by the the negation of the product being performed.

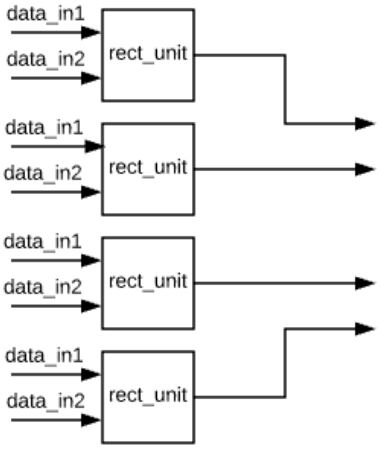

Figure 3: Rectifier Shop

2.3 Random Number Generator

To give the product a unique 8 bit signature we use a 4 bit random number generator using a linear feedback shift register(LFSR). In this we use 4 D flip flops whose outputs are looped back to give us a 4-bit random number generator. Depending upon the flip flops used one can create a random number comprising any number of bits.

In the code for D flip flop we can see that we have taken q as output which gets looped back. One can also take q to be the output . This gives a bit-wise negation of the random number generated in the first case.

In the final shop part we integrate the random number generated and the rectified product code to give us the final output.

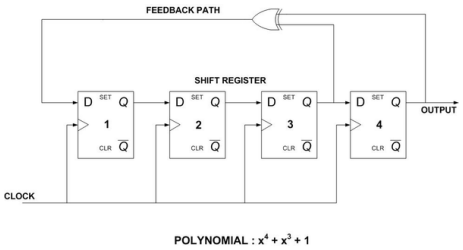

Figure 4: Random Number Generator

3 Hardware

We have used the following components :

- ICO board
- Raspberry PI3
- SD card - 32GB flashed with raspian OS
- Arduino UNO - 2
- LCD display - 16x2
- Jumper cables - male-to-male -20

4 Acknowledgments

We express our most sincere gratitude to Dr. GVV Sharma and Theresh Babu Benguluri for their continuous guidance and support throughout the course. We also thank the Mr.Thirumurugan, EE lab in-charge, for helping us procure the necessary components.

5 References

https://www.arduino.cc/en/Tutorial/
https://en.wikipedia.org/wiki/Pseudorandom_number_generator