Design Project Proposal

Development of a Training System for Defense against Common Mode Failure

Alix Martin

Don King II

Ravindranath Jaglal

Proposal Review Panel Representative:

_____

Name Signature Date

Senior Design I Instructor:

_____

Name Signature Date

December 01, 2010

From: CMF Team – Alix Martin, Don King II, and Ravindranath Jaglal

To: Whom it may concern

Subject: Development of a Training System for Defense against Common Mode Failure – Proposal

Within this document holds the device that the CMF team proposes to develop for senior design 2010/2011. The device that is being proposed is a training kit to assist teachers in educating engineering students in the area of Common Mode Failure. The concept of the training kit that is being proposed is currently not on the market. After extensive research and consulting our Advisor Dr. Charles Kim the CMF team made a design for a training that is contained within the following pages of this document. The document also contains brief reason why this project was proposed, and why the Common Mode Failure training kit is needed.

Table of Contents

Introduction

Electrical systems, in networks for monitoring and controlling today's various systems, are largely based on computer based control instruments. This has created a new set of concerns, since the failure of a single component, such as a processor or a network card, could disable major functions of the system and cripple the whole system. The benefits of component duplication can be defeated by common-cause or common-mode failures. Common-cause failures can occur owing to common external or internal influences. External causes may involve operational, environmental, or human factors. Common mode failure is a phenomenon that occurs when events are not statistically independent. That is, one event causes multiple systems to fail. e.g. Power Spikes, Water, Magnetism, Temperature Variation, and Software Errors. To protect against common design errors, components with a different internal design (but performing the same function) may be used. This approach is called "design diversity". If design diversity is properly implemented, it can significantly increase the reliability and safety of many systems.

An example of this situation is Three Mile Island melt down that took place in Dauphin County, Pennsylvania near Harrisburg on Wednesday, March 28, 1979. The plant was owned and operated by General Public Utilities and the Metropolitan Edison Co. It was the most significant accident in American history in commercial nuclear power generating industry, resulting in the release of radioactive gases. The accident began with failures in the non-nuclear secondary system, followed by a stuck-open pilot-operated relief valve (PORV) in the primary system, which allowed large amounts of nuclear reactor coolant to escape. The mechanical failures were compounded by the initial failure of plant operators to recognize the situation as a loss of coolant accident due to inadequate training and human factors.

The Nuclear Energy Agency (NEA) is currently controlling the international common-cause data exchange (ICDE) project which commenced in August of 1994. The NEA ICDE project collects data on many components which are: centrifugal pumps, diesel generators, motor-operated valves, safety and relief valves, check valves, batteries, reactor protection system components, circuit breakers, and heat exchangers.[1]

So the training kit simulation would be using only a couple of these parameters to demonstrate Common Mode Failure. So the kit would represent one of the devices that maintain one of the systems in nuclear power plant.[1]

[1] Improvement in material since Senior Design Presentation on 12/01/2010

Problem Definition

Reliability and safety of a system are the most important characteristics when it comes to designing something where humans are involved. The purpose of this project is to develop a kit that would assist in the education of students in the area of design diversity, and common mode failure. It would show how the failure is caused in the network, and how to protect against it. Several different types of architecture, and different algorithms that perform the same function will be used, e.g. FPGA (Field-Programmable Gate Array), PIC (Programmable Interface Controller), PLD (Programmable Logic Device), and Microcontrollers. The circuit design would be programmed in C, and then a converter will be use to obtain the assembly code for each architecture. Also, the C code would be written using different algorithms to highlight CMF in software. The training system would use a GUI to control which architecture is used.

Current Status of Art

A training kit for teaching common mode failure is currently not on the market. There is no single course in any university that educate students in common mode failure in both hardware and software applications. A lot of research is being done on CMF in software therefore there exist graduate level courses that lecture on CMF in software applications. There are many institutions that lecture on CMF in hardware using FPGAs, PICs, and PLDs.

Design Requirements[2]

- The kit would only contain four different types of computer systems.
- All the systems contained in the kit will only be programmed through the C language.
- The entire kit should cost less than $250.00.
- The power supply for the kit should not be more than 9V.
- The kit should be no larger than 8 in x 8 in.
- The weight of the kit should not be more than 3lb.

General Solution

Our project will consist of a training kit made to show how common mode failure works. This training kit will connect to the computer and be monitored by a GUI. The GUI will take in a C program, load that program on to each architecture, and then monitor the architectures progress.

[2] Improvement in material since Senior Design Presentation on 12/01/2010

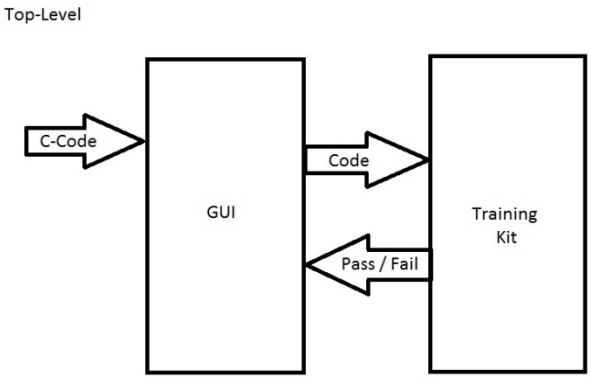

Figure 1.) Top Level

Our training kit will consist of constructing a metal box that will house the four architectures we will be using. Each architecture will connect independently to a computer through a USB connection. By using four different architectures running the same program there is less of a chance of the entire system failing. The four architectures to be used will be a FPGA (Field-Programmable Gate Array), a PIC (Programmable Interface Controller), a PLD (Programmable Logic Device) and an 8-bit Motorola Microcontroller.

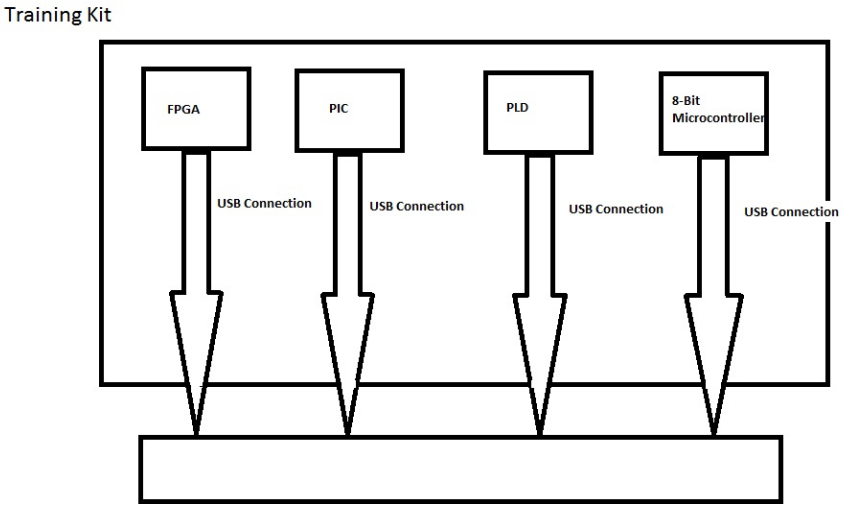

Figure 2.) Training Kit

The GUI is the monitoring system that shows whether the architectures pass or fail. Each architecture will have their own space that shows whether it passed the testing situations. Under the architecture monitoring there will be a space for the user to enter their C program that will be downloaded to each architecture. To download the program there will be a button called “RUN.”

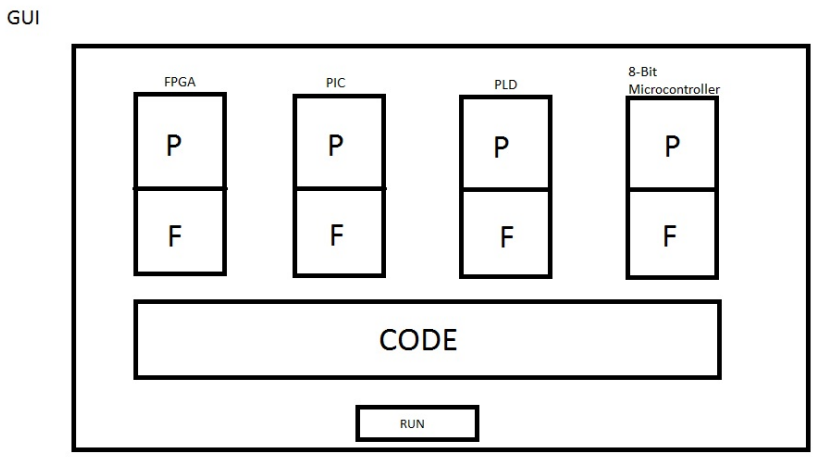

Figure 3.) Graphiccal User Interface

Though the GUI only takes in C programs not all of the architectures being used can be programmed in C. The FPGA board in programmed using VHDL not C. To overcome this problem the C program would have to be converted into VHDL using the Impulse CoDevelopment Converter. The other architechtures can be programmed using C.

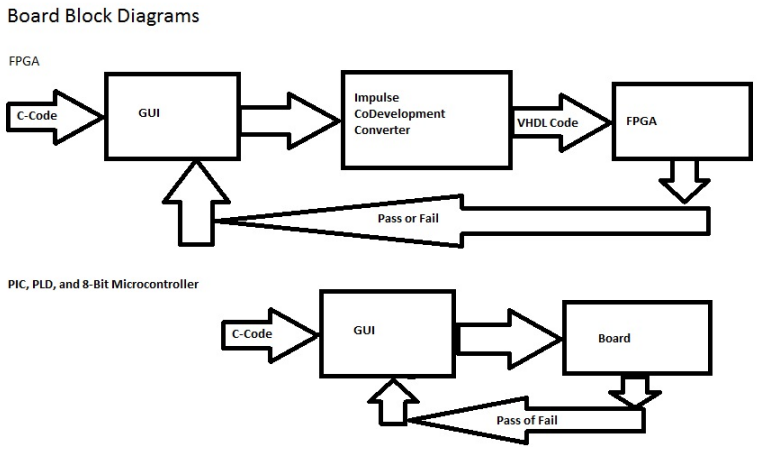

Figure 4.) Block Diagrams

Scenarios

The conditions to be monitored are pressure, temperature, water level, and thermal power. The input logic would work as follows: 0 is normal, 1 is abnormal. The representations of the conditions are shown below.

P = Pressure, T = Temperature, W.L. = Water level, Power = Thermal Power

When the LED’s are 0 there is no action being taken, but when the LED’s are 1 it means the power plant would be performing a certain action. The relation between reactions and LED’s are listed below.

LED1 = Inject More Coolant (0 means everything is ok, 1 means it's activated)

LED2 = Containment Spray System (Spray cold water)

LED3 = Insert Control Rods into Reactor

LED4 = Plant Shutdown

Truth Table Conditions and Responses

Pressure	Temperature	Water Level	Power	LED1	LED2	LED3	LED4
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	0
0	0	1	1	0	0	0	1
0	1	0	0	0	1	0	0
0	1	0	1	0	0	1	0
0	1	1	0	0	0	1	0

0	1	1	1	0	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

Generating Common Mode Failure

Common mode failure would be introduced to system while the system is performing the test scenarios. The first test would be with software, and then followed by hardware.

So to test for common mode failure in software the team would induce several algorithms to the system to perform the same function, and then record the results.

The test for common failure in hardware would be to induce to system to magnetism, temperature changes, power fluctuations, and moisture. These different parameters would test the kit as if it is performing in real life situations. The results of the test will be recorded and tabulated to analyze how the system responded to common mode failure.

Task and Deliverables

Task

- To collect information on methods to test for common mode failure.
- To develop a friendly user interface so every major can use the system.

- To design a kit to house the different computer architectures.
- To design the software application.
- To design a power supply for the kit to power all four architectures.

Table Showing Task Assignments

Task	Assignment
Information Collection	Alix Martin
Develop User Interface	Don King II
Housing Kit	Alix Martin
Software Application	Don King II, Ravindranath Jaglal
Power Supply	Alix Martin, Ravindranath Jaglal

Deliverables

The team would design and produce a kit that would aid an instructor on lecturing in the field of common mode failure. The kit would so also contain test scenarios so students can learn about common mode failure very quickly.

Project Management

Timelines and Milestones

Objective	Due Date
Proposal of Design	12/01/10
Acquire Computer Architectures	01/19/11
Design of Power Supply	02/02/11
Housing for Architectures	02/02/11
Design of Interface	02/02/11
CMF Application	02/23/11
Prototype Completion	03/09/11
Final Testing	03/16/11

Resources and Budget

Resource	Cost
FPGA	$120.00
PIC	$50.00
8 Bit Microcontroller	$35.00
Housing Kit	$20.00
Power Supply	$10.00
Miscellaneous	$50.00
Total	$285.00

Conclusion

In conclusion, on a small scale this is a training kit made to teach students about common mode failure. Students will be able to test the kit in different situations and monitor whether the program still runs under certain conditions. On a larger scale this system could be implemented in a power plant. This system could lower the cause of system failure due to regular or unconventional situations. As stated before we will be using C programming language to program all of the architectures. Also we will be using the Impulse CoDevelopment Converter to convert the C program into VHDL for the FPGA. The system will be monitored by a GUI of our creation that will show whether if each architecture passed our failed during our testing scenarios. The project should cost no more the $250 including software and materials.

References

Nuclear Energy Agency: http://www.oecd-nea.org/jointproj/icde.html

Three Mile Island: http://www.threemileisland.org/