Proceedings of the Korean Nuclear Society Autumn Meeting
Yongpyong, Korea, 2003

Evaluation of Fault Coverage for Digitalized System in Nuclear Power Plants Using VHDL

Suk Joon Kim, Jun Suk Lee and Poong Hyun Seong

Dept. of Nuclear & Quantum Engineering
Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon 305-701, Republic of Korea
E-mail : SukJoonKim@kaist.ac.kr

Abstract

Fault coverage of digital systems is found to be one of the most important factors in the safety analysis of nuclear power plants. Several axiomatic models for the estimation of fault coverage of digital systems have been proposed, but to apply those axiomatic models to real digital systems, parameters that the axiomatic models require should be approximated using analytic methods, empirical methods or expert opinions. In this paper, we apply the fault injection method to VHDL computer simulation model of a real digital system which provides the protection function to nuclear power plants, for the approximation of fault detection coverage of the digital system. As a result, the fault detection coverage of the digital system could be obtained.

1. Introduction

The modern technologies that are based on both of digital hardware and advanced software algorithms are being rapidly developed and widely used. Due to the progress of I&C technologies for process engineering such as computer technology, control engineering, data processing and transmission technology, and software technology, the modern digital technology is expected to significantly improve the performance and the safety of nuclear power plants. However, probabilistic safety assessment (PSA) using conventional techniques cannot adequately evaluate all features of digital systems. Kang and Sung found fault coverage, common cause failures, and software reliability to be the three most critical factors in the safety assessment of digital systems [1]. Among them, this research focuses on the fault coverage of a real digital system in nuclear power plants.

Fault coverage is defined as the conditional probability that a system recovers given that a fault has occurred in the system.

C = Pr { *fault processed correctly* | *fault existence* } (1)

There exist several axiomatic models for the estimation of fault coverage of digital systems and experimental methods for the approximation of the parameters that is required in the axiomatic models. Fig. 1 shows an axiomatic model proposed by Dugan and Trivedi [2]. Our purpose of this work is to estimate the fault detection coverage of a real digital system (C_{ed} in Fig. 1).

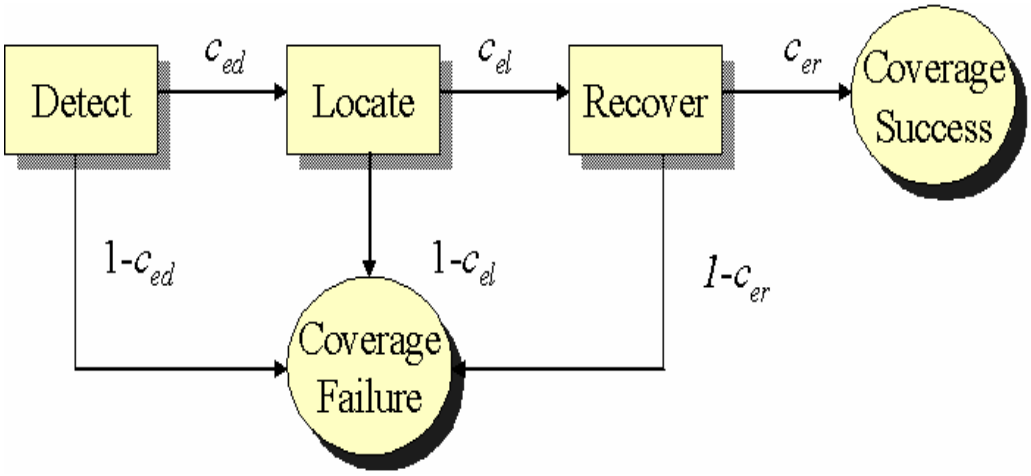

Fig. 1. Permanent effective error axiomatic model

We applied simulated fault injection into VHDL (VHSIC Hardware Description Language) model of the target system. We have chosen this technique due fundamentally to its wide use and acceptance in the fault tolerance community.

Several efforts have been made to develop techniques for coverage estimation as injecting faults into a system prototype. Most of the developed techniques can be implemented within three main categories [3]:

(1) Physical fault injection : It is accomplished at physical level, disturbing the hardware with parameters of the environment (heavy ions radiation, electromagnetic interference, etc.) or modifying the value of the pins of the integrated circuits.
(2) Software implemented fault injection (SWIFI) : The objective of this technique, also called Fault Emulation, consists of reproducing at information level the errors that would have been produced upon occurring faults in the hardware. It is based on different practical types of injection, such as the modification of the memory data, or the mutation of the application software or the lowest service layers (at operating system level, for example).
(3) Simulated fault injection : In this technique, the system under test is simulated in other computer system. The faults are induced altering the logical values during the simulation.

In present work, we intend to perform the evaluation of fault coverage for digitalized system in nuclear power plants (NPPs). Among digital systems in NPPs, we estimate fault

detection coverage of local coincidence logic (LCL) that is a part of digital plant protection system (DPPS) in NPPs. Fig. 2 shows LCL in DPPS.

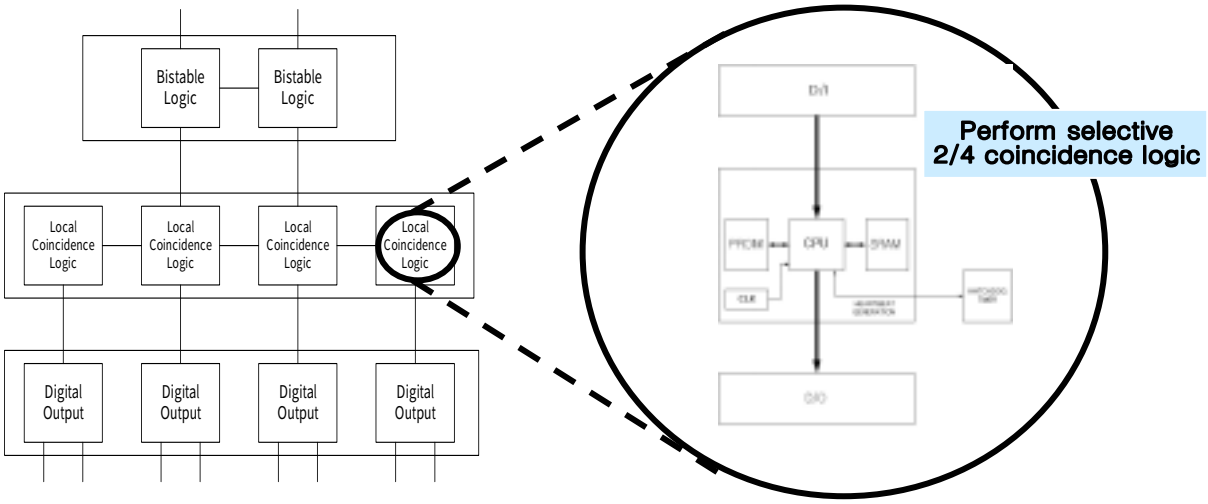

Fig. 2 Local coincidence logic in DPPS

This paper outlines how the simulated fault injection technique is implemented using VHDL, and evaluate fault detection coverage of LCL.

2. Simulated Fault Injection

The VHSIC Hardware Description Language (VHDL) is an industry standard language used to describe hardware from the abstract to the varying level, i.e., the gate, register, or chip level. However, at each abstraction level a design can be defined in either of two domains [4]:

(1) Structural Domain – a domain in which a component is described in terms of an interconnection of more primitive components.
(2) Behavioral Domain – a domain in which a component is described by defining its input/output response by means of a procedure.

In this paper, we perform simulated fault injection in behavioral domain. Fig. 3 shows the fault model for behavioral description. Micro-operation faults perturb individual micro-operations. Control faults perturb the control points that switch between micro-operation sequences. The faults are described in VHDL.

However, we ignore micro-operation faults, because micro-operation faults is performed by changing logical and arithmetic micro-operations. In other words, there is little relation between micro-operation faults and hardware failure.

The control faults change the order of execution or inhibit execution. There are four main techniques. Table 1 shows these techniques and example.

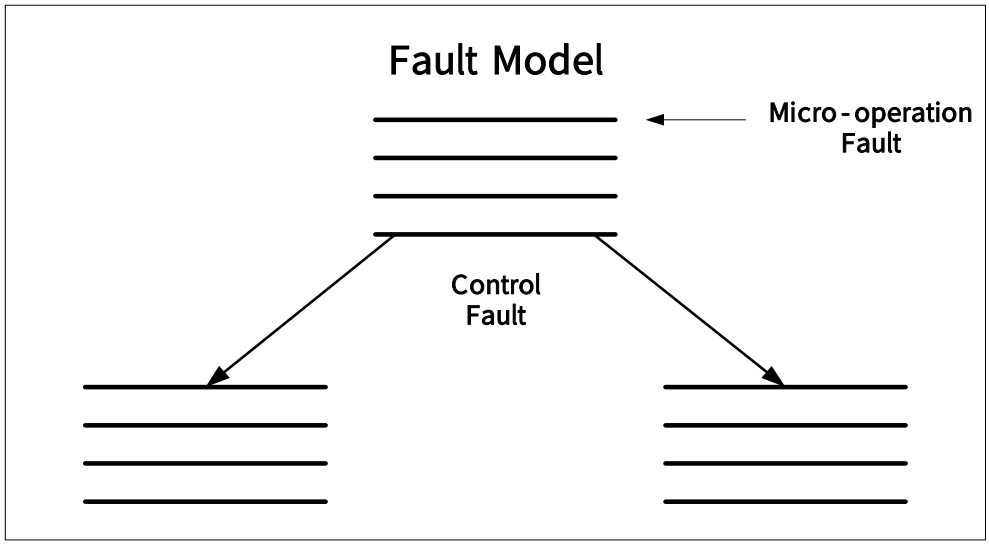

Fig. 3 Behavioral fault model. (ⓒ 1988 IEEE)

Control Fault	Example
1. IF-stuck THEN, stuck ELSE	In the following example, even though x="0101", the second clause would not be performed. Here a clause under the case statement is assumed to be dead. case x(0 to 1) is when "0000" = > A < = B; when "0101" = > A < = not B; -- dead clause when "0111" = > A < = '1'; when "1101" = > A < = '0' … end case;
2. CASE – dead clause	
3. DEAD process fault	
4. Assignment fault	
5. Etc	

Table 1. Control Fault in Behavioral model

The simulated fault injection experiments are broken into two categories; that is, 2-out-of-4 local coincidence logic and LCL hardware. The following two sections detail the experimental setup for the simulated fault injection experiments.

2.1 Simulated fault injection into 2- out- of- 4 LCL

The target system of this work is DPPS which is the plant protection system in Ulchin nuclear power plant unit 5&6 in South Korea. Among various components in the DPPS, we

apply the simulated fault injection to LCL. LCL can determine whether the corresponding plant trip parameter is in the trip state or not as receiving signals from four bistable logics. In this work, fault injection into the 2-out-of-4 coincidence logic was performed to obtain the fault detection coverage of LCL.

For simulated fault injection to VHDL model, the VHDL model for LCL must be constructed. Fig. 4 shows the 2-out-of-4 logic in the LCL. The 2-out-of-4 logic is modeled using VHDL and fault injection was performed into the model.

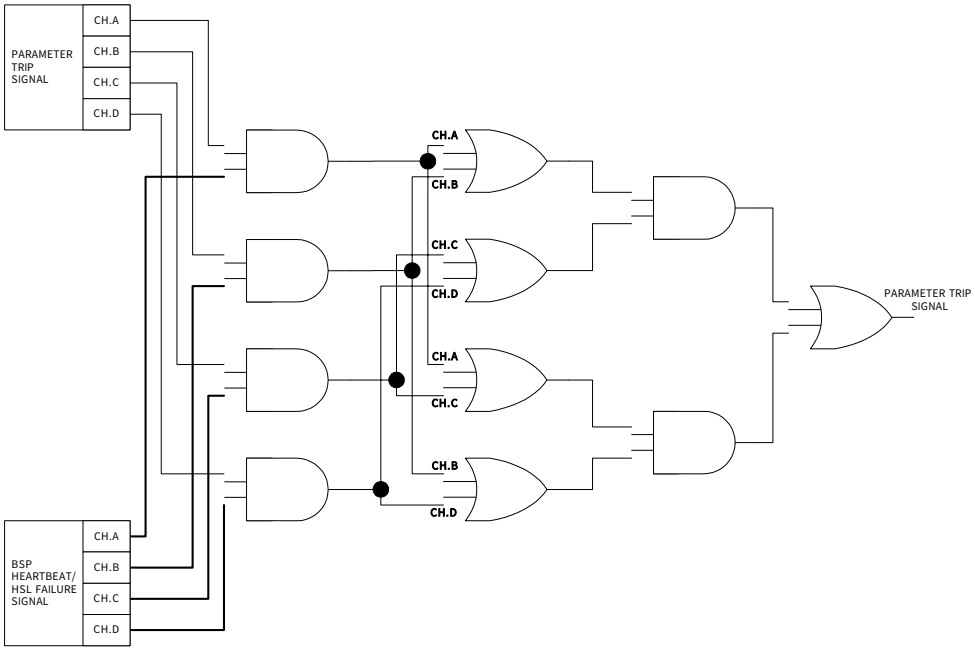

Fig. 4 Local coincidence 2-out-of-4 logic

Fig. 5 shows the simulated fault injection process that is used in this work.

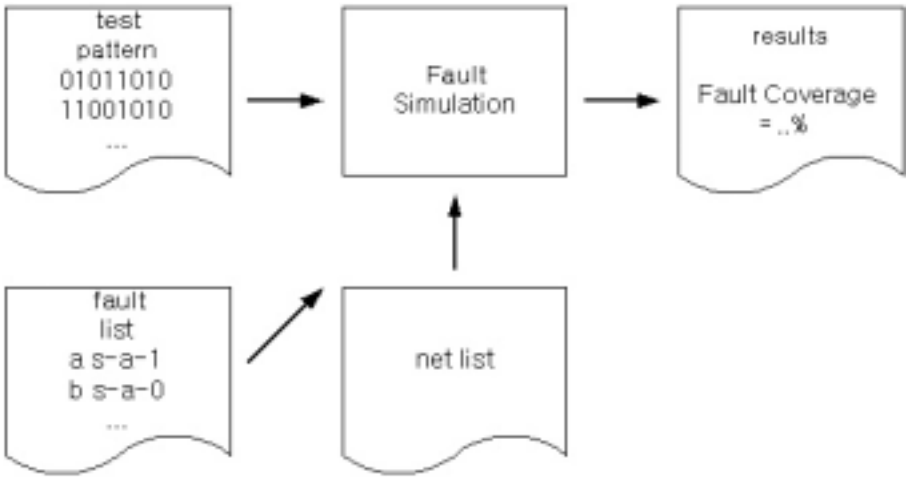

Fig. 5 Fault simulation process [5]

First, we have to prepare test patterns and fault list. In the case of Fig. 4, we need $2^8 = 256$

test patterns (because there are 8 input signals and each signal has two states, 0 and 1) and a fault list with 22 fault cases (because there are 11 gates and each gate has two types of fault, stuck-at-0 fault and stuck-at-1 fault). One thing to note is that we use the single failure assumption, i.e. the failure of only one gate is assumed, because the probability that more than two gates fail simultaneously is so small compared to the probability of the failure of only one gate. As a result, we have to perform 5632 simulated fault injection experiments.

2.2 Simulated fault injection into LCL hardware

The example system in our study is LCL that is composed of Intel 8051 microprocessor instead of Motorola MC68360. Fig. 5 shows the block diagram of the system. It consists of :

(1) 8-bit microcomputer
(2) RAM
(3) ROM
(4) Input/Output port
(5) Etc.

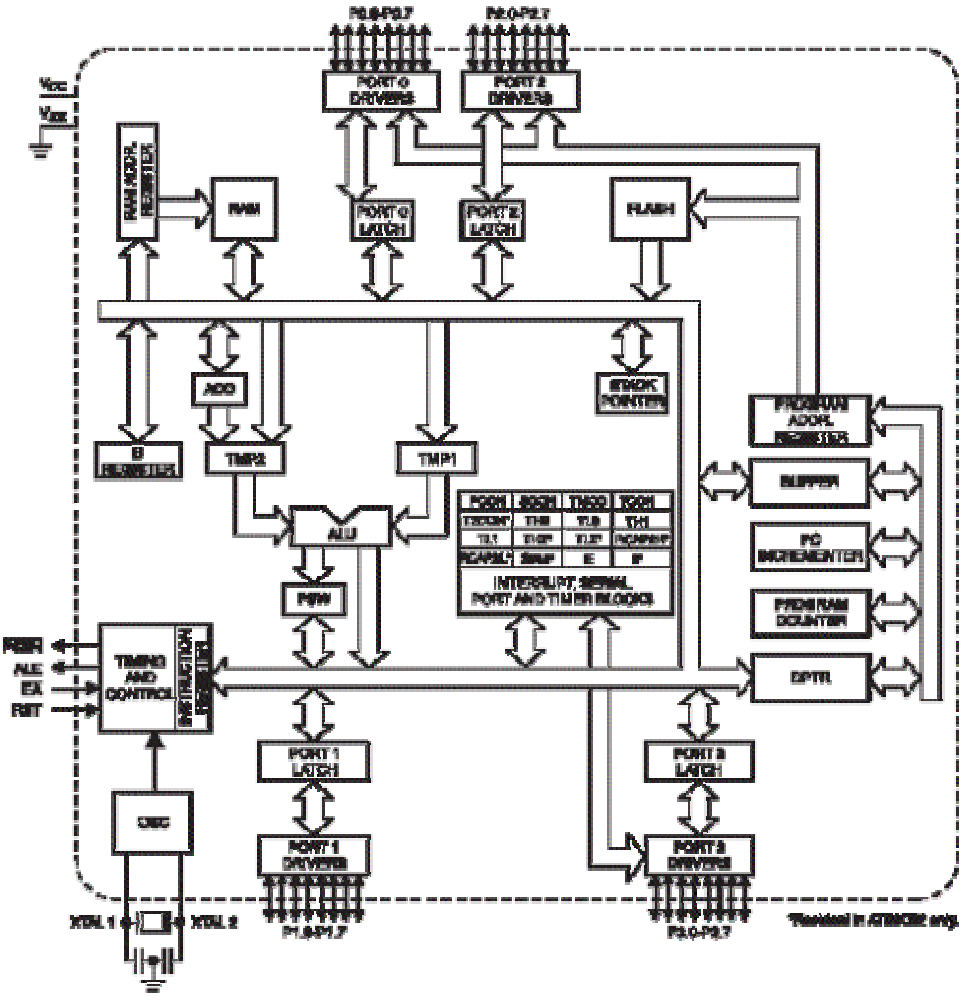

Fig. 5 Micro-processor architecture block diagram [6]

The focus of our simulation experiment is to evaluate the active testing results of permanent and transient fault. The experiment on permanent fault is performed by changing VHDL description statements. If the changed statements are activated, the results of 2-out-of-4 coincidence logic program in ROM will be failure or masked as correct results.

We need compare two parts, fault free state and injected fault state, for determining whether the results of 256 cases are correct or not.

Secondly, the experiment on transient fault is performed by adding fault generation statements. Fig. 6 shows our methods.

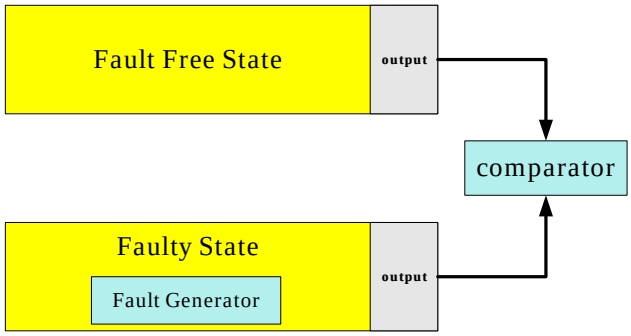

Fig. 6 Schematic diagram for describing transient fault

We performed this experiment using Visual C++, because transient faults that is expressed by using random clause for fault generation can not be described by VHDL. Fig 7 shows fault generator for transient fault generation using random clause.

```
if (( loop%x)==0)
{
        srand(unsigned)time(NULL));
        int ab = rand()%256;
        RAM[ACC] = ab;
}
```

Fig. 7 Transient Fault Generator

Here ‘x’ means that is optional transient fault occurrence probability.

The result of this simulated fault injection experiments are compared with the results of fault free experiments. With one-by-one comparison between the result of the fault free experiments and fault injection experiments, the fault detection coverage of the 2-out-of-4 logic in LCL can be estimated.

3. Conclusion

In this work, we estimated the fault detection coverage of the 2-out-of-4 logic and hardware of the LCL in DPPS, which is a real digital system that provides the protection

function to nuclear power plants. We performed the fault injection experiments on VHDL and Visual C++ computer simulation models to estimate the fault coverage. By comparing the results of fault-free experiments and fault injection experiments, we could estimate the fault detection coverage of the 2-out-of-4 logic in LCL and LCL hardware.

The experiments performed in this work are restricted to the estimation of the fault detection coverage of a small part in DPPS. The simulated fault injection experiments will be expanded to the estimation of the fault coverage of the whole system, DPPS. We expect that the final result of the estimation of the fault coverage will be contributed to the assessment of safety of digital systems.

4. References

[1] Hyun Gook Kang and Taeyong Sung, an analysis of safety-critical digital systems for risk-informed design, Reliability Engineering and System Safety, vol.78,no.3, pp.307-314,2002

[2] Joanne B. Dugan and Kishor S. Trivedi, Coverage Modeling for Dependability Analysis of Fault-tolerant Systems, IEEE Transaction on computers, vol.38,no.6,pp.775-787,June,1989.

[3] Gil D, Martinez R, Busquets JV, et al., Fault Injection into VHDL Models : Experimental Validation of a Fault Tolerant Microcomputer System, LECT NOTES COMPUT SC 1667: 191-208, 1999.

[4] James R. Armstrong, Fong-Shek Lam, Paul C. Ward, Test Generation and Fault simulation for behavioral Models (in Schoen, Joel M., Performance and fault modeling with VHDL), Englewood cliffs, N.J : Prentice Hall, c1992, pp 240-301

[5] Ramin Roosta, Course Material, Chapter 11. Fault Simulation, 2003 spring, pp 7-11.

[6] http://www.8051.co.kr/html/development/cpudata09.php