

Reliability Estimation of Nuclear Digital I&C System using Software Functional Block Diagram and Control Flow

Jong Gyun Choi, Hyun Gook Kang, Tae Yong Seong and Poong Hyun Seong

Department of Nuclear Engineering

Korea Advanced Institute of Science and Technology

373-1 Kusong-dong, Yusong-gu, Taejon, Korea 305-701

1. Introduction

The use of digital systems in nuclear instrument and control system (I&C) prevails because of their increased capability and superior performance compared with the analog systems. However, it is very difficult to evaluate the reliability of digital systems because they include the complex fault processing mechanisms at various levels of the systems. Software is another obstacle in reliability assessment of the systems that requires ultra-high reliability. In addition, the reliability of digital systems has to be assessed considering software, hardware and SW/HW interactions because the software consideration cannot be fully understood apart from hardware considerations and vice versa.

In the hierarchical functional view of a digital system shown in Figure 1, the software system is designed to accomplish functions that the digital system is required to perform. The software system is composed of software modules. The software modules perform their allotted tasks through the combination of instruction sets provided by the microprocessor. The parts of hardware components such as microprocessors and memories are used for processing of one instruction. That is, in order that the digital system completes its required function, the software determines the correct sequence in which the hardware resources should be used.

The failure of system, thus, occurs when the software cannot arrange the sequence of use of the hardware resources correctly or when the one or more of used hardware resources have the faults though the software has determined the correct sequences of use of hardware resources.

In this work, a technique using software functional diagram and control flow is introduced to estimate the reliability of digital system. When the system reliability is estimated, the software should not be considered separately from the hardware because they are related interactively. We propose the reliability model that consider the interaction between hardware and software.

2. Nuclear I&C System

The nuclear I&C system performs its intended function periodically and repeatedly. Additionally, The function does not require complex algorithm as operating system does. Therefore, the Functional Block Diagram technique (FBD) is generally

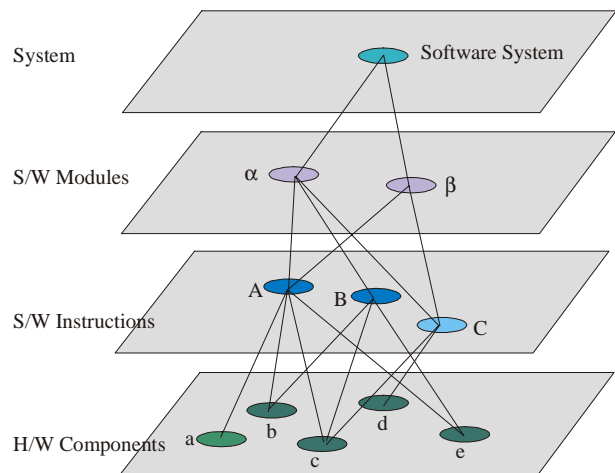


Figure 1. Hierarchical Functional View of Digital System

used for designing and coding the application software in nuclear I&C system. In this technique, the software is designed with only limited number of functional blocks (modules) such as adder, subtractor, comparator and multiplier. Then, they are directly compiled into executable code without the need for the further transformation by designers and programmers.

The software designed through this procedure can have fault sources as follows:

- The faults in requirement specification itself
- The faults occurred during transforming the requirement specification into FBD
- The faults in each functional blocks

The hardware at operational stage has three types of faults as follows:

- The transient faults
- The intermittent faults
- The permanent faults

3. Reliability Model

3.1. Software has no faults and there is no fault tolerance mechanism

It is assumed that the system failure does not occur when the system is idle. In addition, when the failure rates of hardware components are constant and software has no faults, the failure rate of one instruction can be obtained by multiplying sum of failure rates of hardware components that are used by the instruction to the clock time necessary for the instruction processing as follows:

$$\lambda_{ins}^j = m_{ins}^j \sum_i \lambda_{hw}^i, \quad (1)$$

where m_{ins}^j is the clock time necessary for processing of j th instruction. The failure rate of one software module is as follows:

$$\lambda_{mod}^k = \sum_{i=1}^{n_{ins}^k} p_i n_{ins}^{k,i} \lambda_{ins}^i, \quad (2)$$

where p_i is the software branch probability and n_{ins}^k is the total number of instruction i used in k th software module. The p_i is determined by software operational profile. The failure rate of system is as follows:

$$\lambda_{sys} = \sum_{k=1}^n \lambda_{mod}^k. \quad (3)$$

3.2. Software has faults and there is no fault tolerance mechanism at board level

Since the software is not always perfect, the failure occurred by software should also be considered and included in model of the system reliability. Equation (3) can be extended easily to include the software as follows:

$$\lambda_{sys} = \sum_{k=1}^n \lambda_{mod}^k + \lambda_{sf}, \quad (4)$$

where λ_{sf} is the software failure rate. This term is obtained through software testing by the importance sampling based on software operation profile[1].

3.3. Software has faults and there is fault tolerance mechanism at board level

The various fault tolerance methods can be applied to any layer of the application software and hardware. The typical hardware fault tolerance methods are error detecting and correction codes for memories, parity bits for data buses, self-checking circuits, and watchdog timer[2]. These techniques detect and recover the faults in system. Therefore, these techniques affect the reliability of system. When the hardware fault tolerance methods is considered, Equation (4) can be extended as follows:

$$\lambda_{sys} = (1-C) \sum_{k=1}^n \lambda_{mod}^k + \lambda_{sf},$$

where C is coverage factor determined by

$$C = \frac{\text{the number of faults covered by fault tolerance methods}}{\text{the number of faults occurred in system}}.$$

It is difficult to derive mathematical (analytical) form of C . Generally, the value of C is obtained by fault injection experiments[3,4].

4. Further Study

Since this approach was focused on the reliability assessment of nuclear I&C system, the basic assumption of reliability model was that the modeled system performed its intended function periodically and repeatedly like typical protection system does. Thus, it is difficult that reliability model proposed in this work is applied to the systems such as information system and alarm system. The reliability model should be generalized in order that it can be applied to general case.

NOMENCLATURE

λ_{hw}^i : The failure rate of i th hardware components (#/clock time).

t_c : The clock determined by microprocessor.

m_{ins}^j : The clock time necessary for processing of j th instruction.

λ_{ins}^j : The failure rate of j th instruction.

λ_{mod}^k : The failure rate of k th software module.

λ_{sys} : The failure rate of system

References

- [1] D. Tang and H. Hecht, "An Approach to Measuring and Assessing Dependability for Critical Software systems", ISSRE, November 1997.
- [2] V. P. Nelson, "Fault-Tolerant Computing: Fundamental Concepts", IEEE Computers.
- [3] J. Karlson, P. Folkesson, J. Arlat, Y. Crouzet, G. Leber, J. Reisinger, "Application of Three Physical Fault Injection Techniques to the experimental Assessment of the MARS Architecture", Preprints of Fifth International working Conference on Dependable Computing for Critical Applications, Urbana-Champaign, Illinois, USA, pp. 150-161, September 1995.
- [4] A.C. Brombacher and I.W.R.J. van Beurden, "RIFIT: analyzing hardware and software in safeguarding systems", Reliability Engineering and System Safety, pp. 149-156, 1999.