Journal of the Korean Nuclear Society
Volume 26, Number 4, December 1994

Safety Computer System, CPCS Design in Nuclear Power Plant

Se-Do Sohn, Young Suh, Byung-Heon Kang,
Ji-Tae Shin, and Chong Son Chun
Korea Atomic Energy Research Institute
(Received April 18, 1994)

안전등급 컴퓨터, 노심보호계산기계통 설계

손세도 · 서 영 · 강병헌 · 신지태 · 전종선
한국원자력연구소
(1994. 4. 18 접수)

Abstract

The design of safety computer system is described along with the case of software design and testing in the Core Protection Calculator System (CPCS). The application of computer system in safety system requires not only hardware qualification but thorough testing on software to verify its correctness and completeness. The testing on software for CPCS is performed by comparing the outputs of two versions of code. One is implemented in assembly language and the other is in Fortran. The testing is performed in sequencial and overlapping manner. Phase I test verifies that each software module is implemented correctly by executing every branch. Phase II test verifies that the integrated software is complete, meeting its requirements specification and also the integrated system meets its requirement and timing constraints. Through these testing, the Yonggwang Nuclear Power Plant Units (YGN) 3 and 4 CPCS software is verified to be correct and complete, and the integrated system is designed as in its requirements specification.

요 약

안전등급 컴퓨터계통 설계에 대해서 노심보호계산기계통 소프트웨어 설계와 검증시험을 중심으로 살펴보았다. 안전계통에 컴퓨터계통을 사용하면, 하드웨어뿐만 아니라 소프트웨어에 대한 철저한 검증시험이 요구된다. 노심보호계산기계통에 대한 검증시험은 두개의 소프트웨어 버전을 개발하여, 수행결과를 서로 비교함으로써 수행된다. 하나는 어셈블리언어로, 또 다른 하나는 포트란으로 구현된다. 검증시험은 순차적이고 중첩되게 수행된다. 일차 시험은 모든 브랜치 루틴을 수행하여 각 소프트웨어 모듈이 바르게 구현되었음을 확인하는 작업이다. 이차 시험은 통합된 소프트웨어가 모든 요건을 만족하는가 그리고 시간요건을 만족하는 지를 검증한다. 이러한 철저한 검증시험을 통해서 영광 3, 4호기 노심보호계산기계통이 올바르고 완벽하게 구현되어 요건서를 만족함이 확인되었다.

1. Introduction

The safety ststem in nuclear power plant is to generate the trip signal for reactor and/or the actuating signal for engineered safety features. The conventional safety system was implemented using analog circuits including filters, amplifiers and comparators. The processed field signals are compared against the preset setpoints. If it exceeds the setpoint, the trip signal is generated. All the calculational details of safety analysis and analysis uncertainties and the measurement channel uncertainties including environmental effects of design basis events are included in this setpoint. The inclusion of all this complex effects in this simple setpoint drives the value into very conservative direction resulting in the loss of margin in operation. The adoption of computer system makes the claculation of design parameters directly from the field inputs, which reduces the calculational uncertainties. Also the measurement channel uncertainties can be accessed in terms of overall uncertainty toward final design values. The computer system can provide the output in accurate and more user friendly way to the operator. The Core Protection Calculator System (CPCS) in Yonggwang Units 3 and 4 is a safety computer system generating trip signals based on the calculation of Depature from Nucleate Boiling Ratio (DNBR) and Local Power Density (LPD).

In this paper, the design of the safety computer system is described along with CPCS.

2. Design of the Computer System

The design of safety system should meet the design criteria specified in Regulatory Guides or endorsed IEEE standards. Thus the criteria for safety systems such as independence, redundancy, seismic qualification, and environmental qualification are to be enforced. The safety system is constructed as multiple independent channels, each channel with its own components including sensors. In case of computer system, the design should meet additional requirements specified in standards for computer software. This includes additional design document for software and its verification and validation. The software design process can be devided into three major steps of requirements specification, design and implementation, and testing. The requirements specification specifies system functions, inputs, outputs, timing constraints, and system interfaces. The software design and implementation includes data design, architectureal design, and procedural design; implementation which translates the detail representation of software into a programming language realization and finally into machine code. The testing of the software verifies the software is error free and meets all the specification requirements [1]. The relationship among the requirements specification and the various testing steps are depicted in Figure 1.

When a computer system involves with data acquisition, transmission, or interaction with environment at precise time, it can be classified as real time system. The primary requirement in building a real time system is to guarantee a consistent response that satisfies the system's timing requirements. The second requirement is to ensure an acceptable result that meets functional requirements. The third requirement is reasonably fast performance. The real time systems can be classified as a hard real time system or a soft real time system, based on its effect on people in case of failing to meet its time constraint.

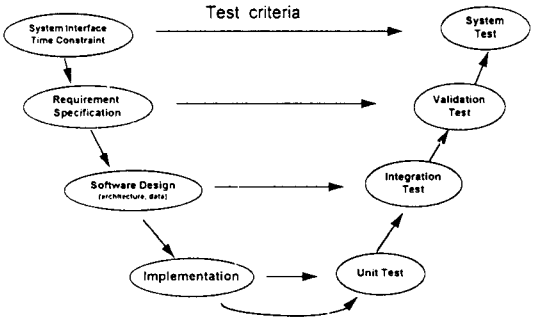

Fig. 1. Design Flow and Test Criteria

Also it can be dassified as a static or dynamic real time system. In a static system, the application has predefined requirements and usually perform same amount of computation at fixed intervals. In a dynamic system, requirements and resource availability vary during system operation. In case of static system, the resourece requirements are acquired analytically or experimentally during design. In a dynamic system, the resource requirements are predicted, tasks are scheduled dynamically, and the guarantee is issued for a task dynamically[2]. Real time systems should be designed to meet the requirements of determinism, responsiveness, user controllability, reliability, and fail-safe operation. Determinism is the tendency of a system to perform an operation in a well defined, or determined time period[3]. Other requirements are manifest by itself.

3. CPCS Software Design

The CPCS is a safety system calculating the DNBR and LPD based on reactor conditions of Reactor coolant pump speed, cold leg and hot leg temperatures, Reactor Coolant System(RCS) pressure, excore neutron flux measuremcnts, and Control Element Assembly(CEA) position signals. The calculations of DNBR and LPD are performed in the following programs: FLOW, UPDATE, POWER, and STATIC. The FLOW calculates the flow rate in the core from pump speed, hot leg, and cold leg temperatures. The UPDATE calculates updated DNBR and LPD based on current read pump speed, pressure, excore neutron flux, CEA position, and temperature input. The POWER calculates the power distribution using the excore reading, CEA positions. The STATIC performs the detailed hot channel DNBR calculation. The execution periods for each program are 50 msec, 100 msec, 1 sec, and 2 sec. respectively[4]. The data flow diagram at system level is shown in Figure 2, in which the shaded task represents the periodic task.

The safety system in nuclear power plant is required to be built wiht redundancy and independence. As such the requirements of real time system reliability and fail-safe are incorporated in independent four channel configuration of CPCS. The determinism of the system is most important requirement in safety system application. Thus it is recommended not to use "interrupt" feature in safety computer system. But most of the commercial operating systems use the interrupt feature to enhance the performance of computer system. Thus the Executive which provides all necessary operating system functions is developed using minimum interrupts. The selection of hardware is based on the following facts: Hardware Knowledge, Performance, and Cost. The selection of Concurrent Computer Corp.(CCC) model 3205 computer for CPCS was made based on the hardware knowledge. The CCC 3205 computer has the following characteristics: 8 sets of general purpose register to reduce time in context switch during interrupt handling, Memory Address Translator providing the memory segmentation and system integrity, and Single bit memory error correction[5]. The CCC 3205 computer adopts its own proprietary bus system ans such specific interface card should be built.

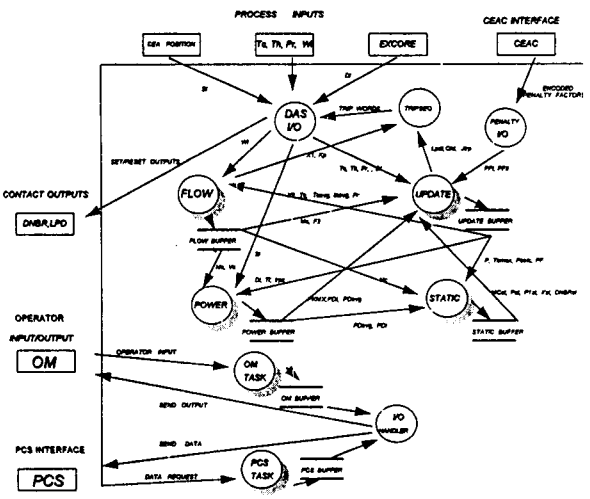

Fig. 2. Data Flow Diagram of CPCS

The Executive is a priority driven real time operating system. It can handle upto 16 independent

tasks. The CCC 3205 computer provides the Test and Set instruction for resolving conflicts among contending processes by making it possible for only one process to receive a permit to enter its critical section[6]. But the Executive uses the pessimistic method of disabling the interrupt when a process enters into a critical section The interface with external devices or with operator is handled by conventional polling method. The Executive can be devided into the following units: Task Scheduler, Interrupt Handler, I/O Handler, Operator's Module Handler, Data Link Handler[7].

4. Testing of CPCS

The testing is performed starting from small modules toward entire integrated system as such the error is detected in orderly manner. Generally the testing is devided into unit testing, integration testing, validation testing, and system testing[1]. The relationship between these test steps and the design process is shown in Figure 1. For the system validation testing, the standard requires that the testing be performed by the person who have not participated in the software design process[8]. In the CPCS, the software is implemented using the Assembly language and the core image is loaded into four independent channels. Also the Fortran code is developed based on the same functional requirements, and the software verification and validation (V and V) is performed by comparing the results of two versions of code as shown in Figure 3. Also for testing, a single channel (meaning one channel among its four channels) is built to verify the integrated system. The results of assembly code execution is compared against the results of Fortran code. The V and V is performed in sequential order of Phase I and Phase II. Phase I is verification test performed for each module and integrated software. Phase II is performed on integrated software and on integrated system (hardware and software). The purpose of Phase I test is to verify the implementation of the CPCS software, i.e. the translation of the system functional requirements into modules of machine executable code and the integration of the code modules into a real time software. In Phase I testing, first the module is tested and then testing on task unit is performed. For each module and task program testing, the test cases are generated manually to execute every branch instructions of the module. The test cases are run on the single channel and recorded in tape. The same test cases are executed using the Fortran code. The results of the assembly language code and those of Fortran code are compared using comparison utility program[9].

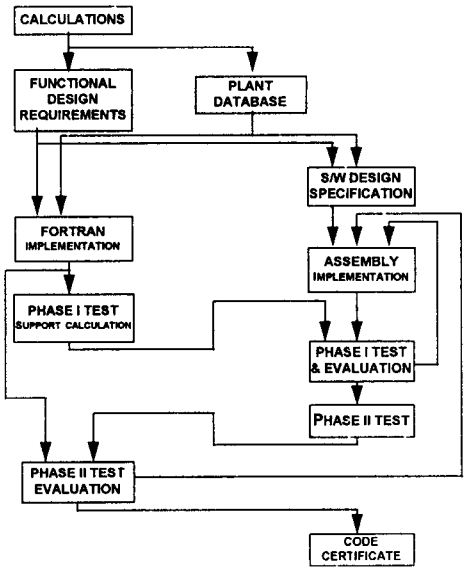

Fig. 3. Software Design Process in CPCS

After completion of Phase I test, the Phase II test is commenced. The integrated software is verified against large sets of static inputs and dynamic inputs. The integrated system including hardware is verified against live input variables by measuring the actual response time. The Phase II test is devided into following tests: Input Sweep Test, Dynamic Software Verification Test (DSVT), and Live Input Single Par-

ameter (LISP) Test. The purpose of input sweep test is to verify system initialization capability and system processing uncertainty. A large number of input sets are selected within the input range of each input variables. The cases are run on single channel, and the same cases are run using Fortran code. The differences of the calculated results (DNBR and LPD) are processed statistically to determine the processor uncertainty. Also the integrated software is verified to initialize for all the input sets. The DSVT is to verify that the dynamic response of the integrated system software is consistent with that predicted by design analyses. The representative transient cases are predefined to execute every functional algorithms. The integrated software on single channel is executed using these transient cases. The single channel results are compared to the results of the Fortran calculation. The LISP test is a real time exercise of the integrated system with transient input values generated from external source and read through the input hardware. The LISP test is to verify that the dynamic response of the integrated system (software and hardware) is consistent with that predicted by design analysis. Also it evaluates the integrated hardware and software system during operational modes approximating plant conditions. Also tested are the system restart capability and its interface with operator. The input voltage and frequency signals are applied to the system, and one input signal is changed at a specified rate. The contact output status change is monitored, and the status change time after transient initiation is measured. These times are compared to the expected results predicted using Fortran code [10]. In each process of Phase I and Phase II test, if any discrepancy is found, the software or hardware is investigated for cause. And the tests are repeated until the tests are successfully finished.

5. Conclusion

The application of computer system in safety system requires thorough testing on software. The determinism of hard real time requires the interrupt usage minimized, thus the executive is also to be developed as such. The testing is performed using two versions of software. One is implemented in fortran and the other is implemented in assembly language. The testing is performed in sequential and overlapping manner devided in phase I and phase II test. Phase I test is a unit testing verifying each module is correct, Phase II test is integrated software testing and includes system testing. In these overlapping and thorough testing, the software and its integrated system is verified for its intended usage. In YGN 3 and 4 units, the CPCS testing showed that the integrated system performs as specified in its requirements.

References

1. R.S. Pressman, Software Engineering, A Practitioner's Approach (1992)
2. S. Natarajan et al., "Issues in Building Dynamic Real-Time Systems", IEEE Software Sep. pp. 16−21 (1992).
3. K.D. Morgan, "The RTOS Difference", BYTE Aug. pp. 161−172 (1992).
4. ACC-CE, "Functional Design Requirements for A Core Protection Calculator" (CE-NPSD-335-P).
5. ConCurrent Computer Corp. "3205 Computer Description"
6. ConCurrent Computer Corp. "3280 Instruction Sets"
7. ABB-CE, "3205 CPC/CEAC Executive System Software Description" (00000-ICE-3094).
8. ANSI/IEEE-ANS-7.4.3.2-1982, "Application Criteria for Programmable Digital Computer Systems in Safety Systems of Nuclear Power Generating Stations".
9. ABB-CE, "YGN 3, 4 CPCS Phase I Design Qualification Test Report" (10287-ICE-3717).
10. ABB-CE, "Phase II Test Report YGN-3/4" (10287-ICE-3791).