The Fault-Tolerant Design and Fault Injection Test for Embedded Software*

Wang ping[1,2]

1,Electrical and Information Engineering College
Xihua University
Chengdu, Sichuan Province, 610039, china

2,The State Key Laboratory of High-end Server & Storage Technology
Jinan, Shandong Province, 250101, china

ping_wang@126.com

***Abstract* - The paper introduces the fault-tolerant technique of chuangxin-1 micro-satellite on-board computer. A fault injection test system is built to verify the fault-tolerant function. The test system is made up of monitor computer, Trace32 ICE, monitor instrument for output, and fault injection instrument. The test case is employed to verify the behavior of fault-tolerance and judge the validity of fault-tolerant design of hardware and software, and it includes typically test case name, test content, instrument and device, test method, verification method, expectation result, actual result, etc. The result shows that the fault injection test system can verify the fault-tolerant design well.**

Index Terms* - *fault tolerant; fault injection; test case; reliability.

I. INTRODUCTION

Chuangxin-1 micro-satellite is the first satellite less than 100kg of china; a centralized dual-computer system is employed to manage the satellite. On-Board Computer (OBC) software is one of the most important parts of satellite. Chuang Xin-1 OBC software manages the application process and running mode of system. The main functions include dual-computer management, mode management, running mode switch, task management, monitor and control process, process switch, process restart, indirect telecommand, GPS receiver manage, communication, attitude control, power control, thermal control, telemetry, etc. To improve the reliability of computer system, we apply some fault-tolerant techniques to software.

According to the functions of satellite, the main functions of OBC software include attitude control, On-Board Data Handling, communication, thermal control, power control, GPS receiver control, orbit decide and prediction, indirect telecommand, telemetry, satellite clock check, device driver, interrupt management, memory management, task management, ground test support, inject program management, etc. Being the most important software in OBC, On-Board Data Handling (OBDH) software manages each of application process and software system running mode.

II. EMBEDDED ON-BOARD COMPUTER SOFTWARE AND FAULT-TOLERANT DESIGN

To protect the normal running when the switch between dual computer is done, the host and standby computer are designed have same OBDH software, OBDH software can run as host state or standby state according to the state of computer. When a computer runs as host state, the OBDH software of the computer can identifies the state by the data of hardware port, and then runs as host software and controls the running of satellite, otherwise, the OBDH software runs as standby software mode. OBDH is required to identify the mode of its computer hardware and switch between host and standby states correctly. The main functions of OBDH software include running mode management of system, manage process running, internal memory management, external memory management, dual-computer switch management, OBC hardware parameter telemetry, software running parameter telemetry, refresh internal memory timely for rectify one error and find two errors, Single Event Upset (SEU) count, and FIFO management, etc.

OBDH software is the core of OBC software, each of process need OBDH software to manage. OBDH software judge the computer running as host or standby computer by A/D acquisition port at first, then decide the software running as host state or standby state. For the control can be smoothly transferred to standby computer in case the host computer is fault, the host would send some important data of its hardware and software to standby computer by dual-computer FIFO communication channel. The standby computer gets these data from FIFO communication port and save them to its memory periodically.

The switch between host and standby OBC can be done by direct telecommand from ground control station or hardware watch-dog find computer running abnormally. The two switch ways aren't performed by software, so OBDH software should identify the computer switch and change software running mode accordingly. When OBC runs in host computer, once OBDH software detected computer switch, computer will be warm booted immediately, then the software is changed to run as standby state. On the contrary, when

* This work is partially supported by the State Key Laboratory of High-end Server & Storage Technology(Grant No. 2009HSSA03), and Scientific Research Fund of Sichuan Provincial Education Department (Grant No. 09ZZ029), and the research fund of Key Lab of Signal and Information Processing of Xihua University.

OBC runs in standby computer, if only OBDH software detect computer switch, computer will be warm booted and the software is changed to run as host state.

The main functions of OBDH software of host computer is as follow: start computer, manage clock, process dual-computer switch, manage running mode, start application process, manage ground test process, form and send FIFO frame, receive and process FIFO frame from standby OBC, count and process SEU. The main functions of OBDH software of standby computer is as follow: start computer, manage clock, manage ground test process, form and send FIFO frame, receive and process FIFO frame from host OBC, etc.

OBDH software decides the running mode of satellite computer by hardware port state, and manages corresponding application process. Different software will be running in each mode.

1. Ground test mode. Ground test software and all kinds of running mode as follow, and it is the mode which satellite work in ground test and simulate flying.

2. Tower mode. All application software and ground test software will be run. The mode will end when the plug between rocket and ground is separated, and then the software change to launch mode.

3. Launch mode. The satellite is in rocket during this course, so the OBDH software and telemetry software will be run, the mode will be switch to capture attitude mode when the separated signal of the plug between satellite and rocket is detected.

4. Capture attitude mode. The application software is run during this course as follow: OBDH software, attitude software, communication software, telemetry software, telecommand software, GPS and orbit software, power control software. The mode will be switch to normal work mode when the flag of gravity gradient boom deployment is detected.

5. Normal work mode. The running software includes OBDH software, attitude software, communication software, telemetry software, telecommand software, GPS and orbit software, power control software, thermal control software.

Application process manages module start process according to system mode. To avoid software trap to infinite loop for some temporal errors, we designed software watch-dog. Application process manage module control these watch-dog too. Software watch-dog is a counter, every process have a watch-dog. OBDH software add 1 to counter and judge the value of counter when it start a process, if the value is over threshold, the OBDH software will deem the process is abnormal and reboot computer immediately. Accordingly, the application process must clear the counter to zero.

III. THE FAULT INJECTION TEST FOR FAULT-TOLERANT SYSTEM

There are some kinds of method to verify the reliability of fault-tolerant computer system software, such as proving or analytical modeling method, field failure data analysis method and fault injection method.

Analytical modeling method is applied to the stage of design or discuss scheme. At this point, the system under study is only a series of high-level abstractions; implementation details have yet to be determined. Thus the system is simulated on the basis of simplified assumptions. The use of dynamic test data, which is relatively crude, to improve a carefully formulated detailed analysis of a structure may appear to be inappropriate. However, if analytical predictions do not match observed behavior, some action should be taken.

Field failures data analysis method assesses the reliability of system by analyze the real-time error data which acquired from the fault field. Because the occurrence of errors and failures are randomly, it will take a lot of time to collect the failure or error data. Field conditions can vary widely, thus leading to doubt on the statistical validity of the result.

Compared to other approaches, fault injection is particularly attractive. Injecting faults into an operational system can provide information about the failure process. To do fault injection, faults are injected either at the hardware level or at the software level and the effects are monitored. The system used for evaluation can be either a prototype or a fully operational system. By speeding up the occurrence of errors and failures, fault injection is a method for verifying the validity of fault-tolerant design with respect to their own specific inputs: the faults that they are intended to tolerate. In fault injection, we inject faults into the system to identify dependability bottlenecks, study system behavior in the presence of faults, determine the coverage of error detection and recovery mechanisms, and evaluate the validity of fault tolerance mechanisms and performance loss.

Fault injection tests fault detection, fault isolation, and reconfiguration and recovery capabilities. There are three kinds of methods for injecting fault: software fault injection, hardware fault injection and radiation fault injection.

Radiation fault injection verifies the fault-tolerant capability of computer system in radiation environment. In radiation fault injection, the hardware is placed in the real radiation environment, and the errors are generated by the heavy ion in radiation to change the state of memory unit.

Hardware fault injection uses additional hardware to introduce faults into the target system's hardware. Depending on the faults and their locations, hardware-implemented fault injection methods fall into two categories: hardware fault injection with contact, and hardware fault injection without contact.

Software fault-injection techniques are applied widely because they don't require additional hardware. Furthermore, it can be used to verify application software and operating systems, which is difficult to do with hardware fault injection. There are two kinds of software fault injection methods: during compile-time or during runtime.

For injecting faults at compile-time, the program instruction must be modified before the program image is loaded and executed. This method injects errors into the source code or assembly code of the target program to emulate

the effect of hardware, software, and transient faults. The modified code alters the target program instructions, causing fault injection. Injection generates an erroneous software image, and when the system executes the fault image, it activates the fault. This method requires the modification of the program that will evaluate fault effect, and it requires no additional software during runtime. In addition, it causes no perturbation to the target system during execution. Because the fault effect is hard-coded, we can use it to emulate permanent faults.

For injecting faults runtime, a mechanism is needed to trigger fault injection. The triggering mechanisms include: time-out, exception or trap, and code insertion. Time-out employs a timer expires at a predetermined time, triggering injection. The time-out event generates an interrupt to invoke fault injection. For exception or trap, a hardware exception or a software trap transfer control to the fault injector. Exception or trap can inject the fault whenever certain events or conditions occur. In code insertion technique, instructions are added to the target program that allows fault injection to occur before particular instructions.

IV. FAULT INJECTION FOR EMBEDDED ON-BOARD DATA HANDLING SOFTWARE

The main functions of on-board data handling software include dual-computer switch, FIFO data management, system running mode management, software running state management, internal and external memory management, etc. Most of the functions are designed for fault-tolerant, then they must be verified, and we designed fault injection test system to check the fault-tolerant behavior of software to verify the validity of fault-tolerance design.

The fault injection test system is made up of 4 parts which are shown in figure, and they are monitor computer, Trace32 ICE, monitor instrument for output, and fault injection instrument. The OBDH software do many job for satellite management, the verification of all fault-tolerant design should be involved. The OBDH software don't only control the running of software, but also control the redundant hardware component, and the control instruction must be verified, then monitor instrument for output is employed to watch the control output.

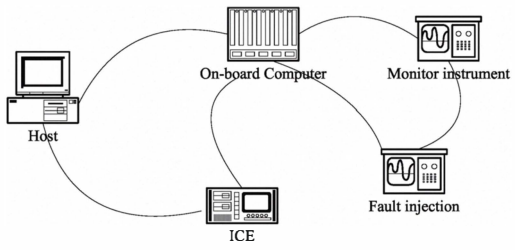

Fig.1 The fault injection test system

The OBDH software run in host computer, the TRACE32 ICE simulate the running of CPU, and then we can debug the software, such as setting breakpoint, tracing program, setting memory value, modifying parameter, etc. With the test, the function of software can be verified, such as path coverage of software, boundary-scan register, etc. the OBC connect the fault injection instrument with cable, and the fault injection instrument can set all kinds of normal or abnormal running state. There are several monitor instrument for output, they are connected with the control output port of OBC and connected with the fault injection instrument for monitoring their output separately.

Before the test of verification, we must know what we want when we do something, and then we must prepare test cases for all kinds of fault preplan. To assure the validity of fault injection test, test cases must cover all kinds of fault and the all probable output of on-Board computer. Each test case should include the content of test, the related instrument of test, test method, how to set the related data of hardware and software, expect result, and how to watch test result, etc. Next we will illustrate a test case which verifies the fault-tolerance of power control.

According to the analysis of power supply fault-tolerance, the OBDH software verdicts whether power is fault or not is based on two rules, one is the monitor parameter of power, another is parameter of payload. To avoid error instruction when the measurement circuit of parameter is fault, only the two parameters are not correct at same time, the OBDH software will send the power switch instruction. According to the function, we design a test case as follow:

Table I
EXAMPLE OF TEST CASE

Test case name	Switch control for power supply fault.
Content	Test the fault-tolerance behavior when the measurement circuit of power is fault
Instrument and device	PC, Trace32 ICE, On-board computer, power fault injection instrument, etc.
Method	Change the output value of power fault injection instrument, make the monitor parameter of power and parameter of payload overflow the normal value, run the power control software in trace32 ICE, watch the output of power control port and the power simulator.
Verificationmethod	Watching the chart of oscillograph and display of power simulator
Expectation	Control pulse is detected on oscillograph, and power supply is switched to standby
Fact	(fill the blank cell according to test)

According to the requirement of fault-tolerance, we should design the test cases which handle with all kinds of faults. We must test the behavior of software according to the test case, and write down the test result, then verdict whether or not the fault-tolerant design meets the requirement of function. If the fact don't meet the expectation, we must analyze the reason and revise the fault-tolerant software to meet the expectation.

ACKNOWLEDGMENT

This project is supported by the State Key Laboratory of High-end Server & Storage Technology of china (Grant No. 2009HSSA03), and Scientific Research Fund of Sichuan Provincial Education Department of china (Grant No.

09ZZ029), and the research fund of Key Lab of Signal and Information Processing of Xihua University.

REFERENCES

[1] Mei-Chen Hsueh, Timothy K. Tsai, Ravishankar K. Iyer, "Fault Injection Techniques and Tools," Computer, vol. 30, no. 4, pp. 75-82, Apr. 1997

[2] Daniel J. Sorin, Fault Tolerant Computer Architecture, Morgan & Claypool Publishers, 2009.

[3] JV Carreira, D Costa, JG Silva, "Fault injection spot-checks computer system dependability", IEEE Spectrum, Vol. 36, no. 8, pp. 50-55.1999

[4] J. Arlat et al, "Fault Injection and Dependability Evaluation of Fault Tolerant Systems", IEEE Transactions on Computers, vol. 42, no. 8, pp.919-923, Aug. 1993.

[5] Jean Arlat, Peter Folkesson, and Gu¨ nther H. Leber, "Comparison of Physical and Software-Implemented Fault Injection Techniques, IEEE Transaction on computers", Vol.52, No.9, pp.1115-1132, 2003

[6] Wang, Ping; Sun, Ning, et al. "Fault-tolerant design of software/hardware for on-board-computer control system of micro-satellite", Journal of Astronautics, Vol.27, No.3, pp.412-415, May, 2006

[7] Ping Wang, "The design of highly reliable on-board data handling software for micro-satellite", Second International Conference on Space Information Technology, 67951Q, Nov. 2007