A Space Network Oriented Spacecraft Software High-Reliability Maintenance Method

Liang Qiao[1(✉)], Ying Wu[1], Xiaoyi Zhou[2], and Lijun Yang[1]

[1] Beijing Institute of Spacecraft System Engineering, China Academy of Space Technology, Beijing 100094, China
owenqiao@126.com

[2] School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Abstract. In order to improve the capability of on-demand customization and agile reconfiguration of space network, and to improve the scalability and reliability of space network, this paper proposed a method of spacecraft software in-orbit high-reliability maintenance method that is suitable for space network. This method is based on the electrical erasable memory of the onboard computer, uses various methods to improve the reliability, writes the reconfiguration data into the memory, and realizes the non-volatile reconfiguration of onboard software based on multiple granularities such as parameters, modules, processes or complete software. In addition, this method uses the hardware EDAC function to detect and correct errors in the memory during operation, which further improves the robustness. The in-orbit application and a large number of tests show that the method is suitable for on-demand customization of space network and can realize in-orbit reconfiguration of onboard software with high reliability.

Keywords: Space network · Spacecraft software · Software high-reliability maintenance

1 Introduction

Currently, there are more than 200 in-orbit satellites in China, most of which are designed for functional solidification. Also, these satellites are heterogeneous and difficult to connect to the Internet, resulting in the delay of information transmission, the difficulty of cooperation among satellites, the high complexity of ground operation and maintenance and the low efficiency of satellite application [1]. However, current satellite users have increasing demands for agile reconfiguration and on-demand customization of the space network. The existing space network does not have the overall design of software architecture based on the software definition, and it is difficult to make the dynamic deployment and in-orbit reconfiguration for several types of inter-satellite links and multiple network nodes, which makes the space network unable to exert the full effectiveness of the existing satellite system.

The spacecraft data management software acts as the brain of all satellite information on the satellite. It is not only responsible for the transmission, processing and autonomous

Q. Wu et al. (Eds.): WiSATS 2020, LNICST 358, pp. 401–409, 2021.
https://doi.org/10.1007/978-3-030-69072-4_32

task management within the satellite, between the satellites and the ground stations, or among the satellites through inter-satellite links, but also an important implementation carrier of space network routing, network protocols and antennas pointing calculating. The PROM (Programmable Read-Only Memory) chips, that are stable but can be programmed only once, are generally used for programming software on traditional satellites. The only way to reconfigure software is to remove the chip and reprogram the new software. Using programming-once PROM chips have better resistance to spatial radiation, but make software debugging difficult. Even worse, it is impossible to change the space network protocols after the launches of the satellites, and this programming method is not conductive to agile reconfiguration and on-demand customization of the topology and the protocols of space network. Another way to reconfigure spacecraft software is to change the data in RAM (Random Access Memory) directly as the software changes a little. But RAM is a kind of volatile memory. In the space environment, high energy particles often cause the computer to restart [2], resulting in the loss of software updates, which means disconnection from the network and loss of communication for space network node satellites. Therefore, in practical usage, this method also has many problems.

In traditional spacecraft computer design, in order to improve the reliability of satellite service, software and hardware are with integrated design, but integrated design also poses the problems of pattern rigidity, which has an essential counterpart and a deep diction with the future demand for flexibility and plasticity. In order to improve the flexibility and maintainability of the spacecraft software, some satellites, especially those connected to the space network or carrying the network protocol function, adopt non-volatile memory, such as flash [3] and EEPROM (Electrically Erasable Programmable Read-only Memory) [4], to program software with space network management module. Since updates to non-volatile memory will not be lost after a power failure, all or part of the new software version can be written to memory through telecommand channel. In this way, the non-volatile reconfiguration of onboard software can be performed based on multiple granularities such as parameters, modules, processes or complete software, which can achieve downward compatibility and upward evolution of space network. In academia, there are several discussions on the method of in-orbit maintenance of spacecraft software. Wang Zhanqiang et al. proposed a method on software on-orbit reconfiguration of space-borne processing equipment [5], which can adapt to the software in-orbit reconfiguration tasks related to software designed radio technology, but this method needs to be improved in reliability; Guo Zongzhi et al. proposed a method on spacecraft software refactoring scheme based on dynamic loading mechanism of module [6]. However, this method can only be used with VxWorks operating system, which lacks of universality.

In this paper, we propose a method of maintaining in-orbit spacecraft software orienting the agile reconfiguration of space network with high reliability. First of all, this paper introduces the basic process of software in-orbit maintenance based on non-volatile memory, which is suitable for agile reconfiguration of space network. Then, this paper introduces several key technologies to ensure the reliability of software in-orbit maintenance. Finally, the effectiveness of this method is illustrated by introducing the test results.

2 Basic Process of Software High-Reliability Maintenance for Space Network Agile Reconfiguration

Compared to common embedded software, there are three main differences in the in-orbit maintenance process of spacecraft software. Firstly, the TT&C ground station cannot connect the satellites at all time [7]. For instance, LEO satellites can connect with a ground station for a few minutes per circle. And it takes a long time to send and execute the reconfiguration telecommands of the entire space network protocol, so the whole reconfiguration process may be divided into several times. Secondly, the telecommand channel is a wireless channel, which has a relatively high error rate. Therefore, to improve the reliability of the reconfiguration process, there should be more reliable design. Thirdly, high energy particles in the universe can be projected into the sensitive area of semiconductor devices, resulting in Single Event Effect [8]. Program data stored in the memory may have a single-bit error or a double-bit error. Therefore, it is necessary to detect and repair errors continuously while the program running.

Taking into account the needs for maintenance of spacecraft software with space network function and the different characteristics from common embedded software, as shown in Fig. 1, the authors designs the maintenance process of in-orbit software suitable for agile reconfiguration of space network.

After starting the spacecraft onboard computer, the program data in the EEPROM is moved to the RAM and run in RAM. If the program has to be modified in orbit due to the updates of software parameters or the space network protocols, the data in RAM can be modified directly, but this method will cause data loss in case of power failure. If a permanent update is needed, it can be written with whole or part of the new version of the in-orbit software in EEPROM, by this way, the non-volatile reconfiguration of onboard software can be performed based on multiple granularities such as parameters, modules, processes or complete software. In order to resolve the high dynamic performance of space network and the relatively high bit error of the uplink channel, and to achieve the high reliability of the software maintenance process, the uplink telecommand instructions from the ground are temporarily stored in RAM. After receiving all instructions, the running software compare the check code attached to the instructions with the check code calculated from the data temporarily stored in RAM by same algorithm. If the verification is correct, the reconfiguration data is written to EEPROM.

To improve the reliability of the in-orbit software, to improve the reliability of its communication inter-satellite link and to reduce the impact of Single Event Effect, the EDAC circuit can be used for self-checking of EEPROM in the idle time of software running. In addition, another common method is Triple Modular Redundancy (TMR), that is to store the same data in three pieces of memory. When loading data, program is processed by a majority-voting system to produce a single output., but this method needs much more storage resources.

After the satellite was launched, when there are agile reconfiguration and customization requirements of the space network topology or protocols, the in-orbit maintenance method in this section can be used to reconfigure the spacecraft onboard software performing space network functions and detect and correct errors in memory.

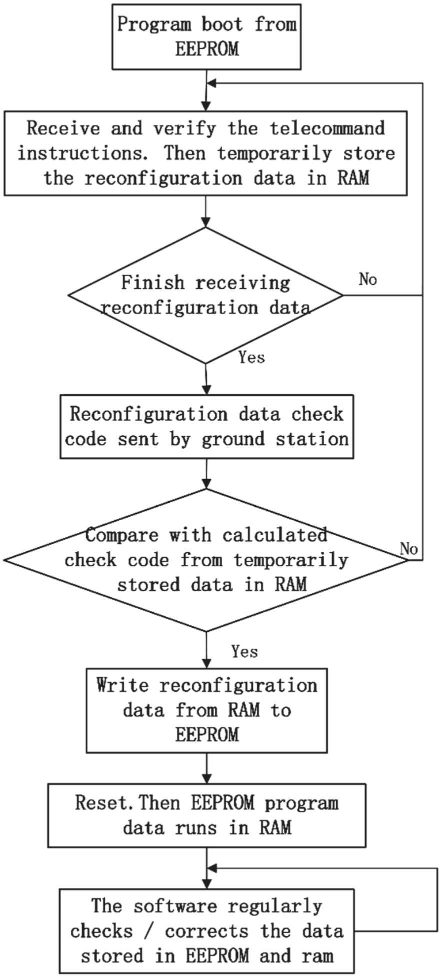

Fig. 1. The flow chart of in-orbit software high-reliability maintenance

3 Key Technologies of Spacecraft Software In-Orbit Maintenance

In this section, some key technologies of spacecraft software in-orbit maintenance adapted for agile reconfiguration of space network are discussed.

3.1 High Reliability EEPROM Data Maintenance

A single high energy particle in the universe can be projected into the sensitive area of semiconductor devices, causing Single Event Effect. Under this circumstance, program data stored in the EEPROM may have a single-bit error or a double-bit error, which could cause the software crash in severe cases. This is a fatal problem for spacecraft software that performs space network functions and could interrupt communications

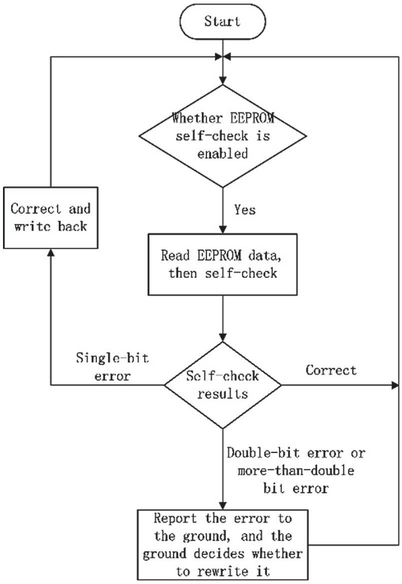

Fig. 2. The flow chart of EEPROM self-check and auto-correc

among satellites. As a result, the redundant design for EEPROM is very important for the reliability of the space network (Fig. 2).

It should be noted that when writing EEPROM, the tunnel oxide layer would withstand high voltage and high electric field, so the lifetime of the EEPROM is limited. To extend the lifetime of EEPROM and to prevent in-orbit failure of EEPROM, the frequency of self-detection of EEPROM should not be too high.

3.2 Reliability Design of In-Orbit Reconfiguration Telecommand Instructions

The TT&C ground station cannot maintain communication links with satellites at all time, and it also takes a long time to reconfigure the entire space network parameters and protocols, so the entire program may have to be divided into hundreds of instructions and be sent separately. Moreover, the data uplink channel is a wireless channel, which has a relatively high error rate. Therefore, to improve the reliability of the reconfiguration process, there should be more reliable design. In this section, the fault-tolerant design of reconfiguration instructions is of great importance to ensure the integrity and efficiency of reconfigured data and the implementation of space network functions.

The reliable design of the software in-orbit reconfiguration instructions mainly includes the following:

(1) Verification of the instruction format. Software checks the instruction fields according to pre-determined format including the instruction verification code format comparison, optional range of each field and the validity of the memory addresses to reconfigure, etc. The software will refuse to execute the telecommand instructions that unable to pass the format verification. Verification of the instruction format could prevent the instructions from appearing error codes in instruction uplink.

(2) Comparison of the telecommand instruction code. For small-scale and key changes, the onboard computer can be configured to transmit instruction code data back to the ground station for comparison and verification after receiving the instructions. Once the correctness of the instructions is confirmed, the ground station sends the confirm instruction and the original instructions can take effect.

(3) Continuity check of the serial number of instruction packets. To write a large amount of data to EEPROM, it is necessary to divide the large instructions into several packets. And the onboard software will receive these packets in sequence. In case of packets loss or packet serial number error, it will alarm the ground station by telemetry channel to ensure the integrity of instructions.

(4) Temporary storage in RAM. In order to further improve the reliability of the reconfiguration instructions, the author also designed a method to store temporarily the reconfiguration data in RAM. The reconfiguration data is temporarily stored in RAM. After all of the reconfigured data have been received, the ground station sends the verification code for the whole data and compares it to the verification code calculated from the temporary stored data in RAM. If the two parts of verification code are matched, the reconfiguration data will be written to EEPROM from RAM.

The reliability design of reconfiguration instructions can ensure the accuracy and completeness of data and ensure the correct operations of the space network after reconfiguration.

4 Engineering Design and Test Verification

At present, there are several satellites in orbit equipped with computer with non-volatile memory, which can realize space network oriented software in-orbit reconfiguration, and realize the on-demand customization of space network protocols. The typical computer architecture is shown as Fig. 3. Compared with the traditional method, that is programming program in PROM, the use of non-volatile memory, although losing the reliability within the tolerable range, improves software flexibility and achieve a balance between reliability and flexibility, which is very important for the rapid evolution of the current space network development. The CPU board of a high orbit satellite platform computer takes CPU as the core, and constitutes a complete computer system with other modules including memory, 1553B bus controller, high stability clock interface, etc. In the process of software running, the non-volatile reconfiguration of onboard software can be performed based on multiple granularities such as parameters, modules, processes or complete software. In the process of reconfiguration, the original program can still continue to run.

The authors constructed the test environment, and verified the in-orbit software reconfiguration method introduced in this paper. As shown in the Fig. 4, the test environment is composed of the main control computer, test client, LAN network, cables and test software to form a high-performance test system. The main control computer controls each test client, and all test data and test results can be displayed and processed on the main control computer.

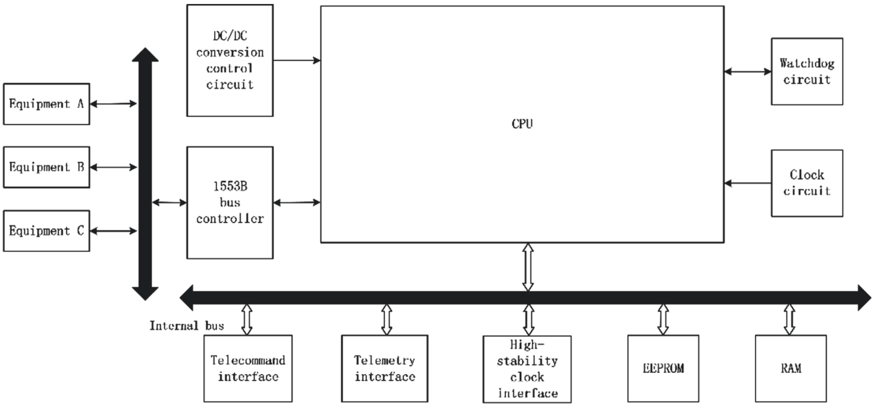

Fig. 3. A typical spacecraft computer architecture

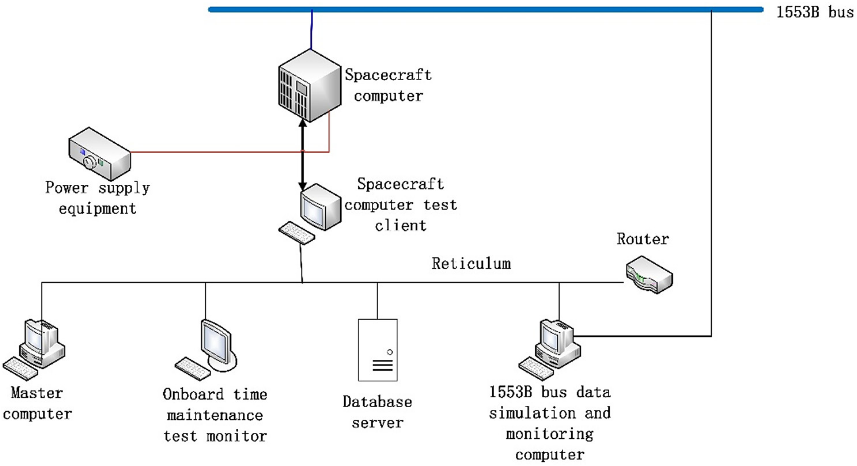

Fig. 4. Spacecraft software high-reliability maintenance test environment

4.1 High Reliability EEPROM Data Maintenance Test

Single-bit Error Test. Firstly, we wrote the data at an address within EEPROM address space as 0x55555555. With calculation, we got the EDAC check code as 0xed, then used the emulator to change the original data to 0x55555554. Next, we turned on the EEPROM self-check function. After a period of time, we read the data at that address, and found that the data became 0x55555555, which showed that in the self-check, the single-bit error of EEPROM data has been corrected automatically.

Double-bit Error Test. Firstly, we wrote the data at an address within EEPROM address space as 0x55555555. With calculation, we got the EDAC check code as 0xed, then used the emulator to change the original data to 0x55555556. Next, we turned on the EEPROM self-check function, after a period of time, the telemetry showed that that EEPROM address had an uncorrectable error, indicating that in the self-check, the double-bit error of EEPROM data has been detected.

4.2 Software Reconfiguration Test

At first, we compiled the whole program containing the new space network protocols into executable file, and divided the binary code into several reconfiguration instruction packets according to the telecommand instruction formats. Then the packets were injected into the spacecraft computer system through the ground test software in sequence. We got some information from telemetry frames that the format of each instruction packet has been checked to be correct, the check code is also correct, and the order and quantity of received instruction packets are checked to be correct. After the restart of the computer, the newly program was started by EEPROM normally, and the new space network protocols ran correctly, which proves the correctness of the software reconfiguration process.

The correctness of space network oriented spacecraft software high-reliability maintenance method is verified by high reliability EEPROM data maintenance test and software reconfiguration test.

5 Conclusion

In this paper, a space network oriented spacecraft software high-reliability maintenance method is proposed. Firstly, the limitation and shortage of using one-time programmable PROM to program the spacecraft software are analyzed. Then, based on the characteristics of EEPROM, the basic process of high-reliability maintenance for onboard software is introduced, and then several key technologies of in-orbit software reconfiguration and data maintenance are introduced in details. In the end, this paper describes the engineering design and analyzes the experimental verification of the in-orbit software high-reliability maintenance method that suitable for agile reconfiguration and on-demand customization of space network. The test results show that the method can realize in-orbit reconfiguration and data maintenance of spacecraft software, meet the design requirements, and improve the in-orbit maintainability and expansibility of space network.

References

1. Chien, W.C., Lai, C.F., Hossain, M.S., Muhammad, G.: Heterogeneous space and terrestrial integrated networks for IoT: architecture and challenges. IEEE Netw. **33**(1), 15–21 (2019)
2. Peng, C., et al.: Low-energy proton-induced single event effect in NAND flash memories. Nucl. Inst. Methods Phys. Res. **969**, 164064 (2020)
3. Choi, S., Oh, Y., Song, Y.-H.: A novel three-dimensional NAND flash structure for improving the erase performance. IEICE Electron. Express **16**(3), 20181016–20181016 (2019)
4. Chenguang, W., Wensheng, S.: Design and implementation of multi-state reboot recording system for embedded devices based on linux. Softw. Guide **18**(05), 125–129 (2019)
5. Zhanqiang, W., Shenghua, Z.: Research on software on-orbit reconfiguration of space-borne processing equipment. Space Electron. Technol. **1**, 7–13 (2013)

6. Zongzhi, G., Bin, L., Yulong, Z., Xiaoli, T., Chong, N.: Research on spacecraft software refactoring scheme based on dynamic loading mechanism of module. Comput. Meas. Contr. **19**, 126–9 (2018)
7. Qiao, L., Yan, H., Zhang, Y., Zhang, R., Jia, W.: Multilayer satellite network topology design technology based on incomplete IGSO/MEO constellation. In: Yu, Q. (ed.) SINC 2019. CCIS, vol. 1169, pp. 28–38. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-3442-3_3
8. Aerospace Research - Avionics; Findings from Institute of Space Systems in Avionics Reported (Heavy Ion Induced Single Event Effects Characterization on an RF-Agile Transceiver for Flexible Multi-Band Radio Systems in NewSpace Avionics). Defense & Aerospace Week (2020)