Onboard Software Maintenance Design and Implementation for Networking Satellites

Wei Wu(✉), Liang Qiao, Hongcheng Yan, Cuilian Wang, Yong Xu, and Xiaorui Yang

Beijing Institute of Spacecraft System Engineering, Beijing, China
18610104602@163.com

Abstract. A design method of onboard software automatic update for networking satellites is proposed. Using the inter-satellite link, the hand-in-hand satellites with each other transmit the version numbers of the software and FPGA that can be updated and the applicable satellites to each other. When the version numbes of a software item between hand-in-hand Sat-A and Sat-B is inconsistent (Sat-a is higher than Sat-B), Sat-B can automatically initiate the request to update. Then Sat-A can automatically read out the software data from the Nandflash memory with splitting and re-framing and send it to Sat-B, according to the dedicated parameters of Sat-B and the inter-satellite link data transmission format. The updated software data is received and verified by Sat-B, and then stored in Nandflash memory and program store Norflash or EEPROM memory. After the update is completed, Sat-B will transmit the latest software version information to other hand-in-hand satellites, and support the software update Correspondingly. If this scheme can complete the software update of one of the group networking satellites on the ground, it can automatically upgrade the satellites software of the whole network, reduce the ground operation, and greatly improve the efficiency of onboard software maintenance. The method has been demonstrated with two space network routers for Compass Navigation Satellites in laboratory. A Virtex-5 FPGA configuration data in one router is updated by terrestrial injection and the corresponding Virtex-5 FPGA configuration data in the other router is updated automatically with Inter-Satellite Link.

Keywords: Networking Satellites · Software Maintenance · Automatic Updating

1 Introduction

With the rapid development of space industry, the types and functions of satellites are also expanding, while the development cycle of satellites is constantly shortening. In order to adapt to the expansion of the complexity of satellite functions and the compression of the development cycle, the satellite functions trends to be more and more defined and implemented by software. Therefore there are more and more embedded software and FPGA items in a satellite (hereinafter referred to as software without distinction). At present, a large satellite has hundreds of software items, and even one electronic unit with complex functions has more than a dozen.

R. Li et al. (Eds.): MOBILWARE 2022, LNICST 507, pp. 106–116, 2023.
https://doi.org/10.1007/978-3-031-34497-8_9

Function based-on-software brings great flexibility to the realization of satellite functions [1]. An important feature is that some core function can be maintained onboard, that is, after the satellite is launched into orbit, the satellite functions can also be modified by changing the software code. Onboard software maintenance mainly has the following three design necessities or requirements: First, correcting software errors. Due to the shortening of satellite development cycle and the enlargement of functional complexity, many software functions are not fully tested on the ground, and cannot be debugged completely. The second is to adapt to the space operation environment. Due to the radiation characteristics of space environment, the software program carrier may have errors caused by memory bit reversal. In order to correct these errors, the software needs to be maintained onboard. Third, software functions upgrading. With the advancement of satellite in orbit operation, it may be necessary to add some new functions or upgrade the original functions, which also requires the software to have the function of onboard maintenance [2–4].

Onboard software maintenance needs to be considered in the system design. The software items that need to be maintained need to be supported in the hardware design. For example, the software program memory needs to be designed into EEPROM or flash that can be rewritten many times. At the same time, the satellite OBDH software needs to be able to support the data receiving and distribution processing of satellite software program data.

At present, the design of onboard software maintenance is mostly for a single satellite. The ground TT&C system uploads the data of a software program into the OBC frame-by-frame according to the format of special data frame. The program data are analyzed and distributed by OBC and finally received and stored by an electronic unit where the software item is located. These operations are executed and written into the memory with one-by-one instruction. The data go through multiple steps, and the slowest step will become the bottleneck of the maintenance system, resulting in low efficiency and high complexity of onboard maintenance operations.

As many satellites add inter satellite links for networking operation, the efficiency will not meet the requirements of simultaneous maintenance of many satellites while the above method is used to maintain the satellite software program one by one.

2 Single Satellite Onboard Software Maintenance

The design of single satellite onboard software maintenance is relatively mature. The operation mainly uploads the software data from the ground TT&C station frame by frame into the memory of the target software program. Figure 1 shows the maintenance path diagram of an electronic unit on a satellite, which is a common software onboard maintenance scheme of the current satellite. The electronic unit has an SRAM-based FPGA that needs to be maintained. The ground system needs to cut the configuration data of the FPGA into data blocks continuously and organize them into telecontrol frames. The ground TT&C system uploads these frames into the satellite one by one through the TT&C channel. The frames will reach the CMD receiving block of the OBC through transponder, and parsed and distributed to the maintenance-needed electronic unit through the corresponding interface (1553B bus interface in the example). After received by electronic unit, the maintenance data frame will be written into program memory by the FPGA control block, which is generally realized by a highly reliable anti-fuse FPGA or a special AISC.

In the single satellite maintenance scheme [2], the telecontrol data frames are directly written into the program memory frame by frame, without corresponding collection, storage, processing and distribution. For example, the OBC only takes instruction decoding and forwarding. The correctness of the configuration item data is mainly guaranteed by the target electronic unit itself, while the transponder and OBC only guarantees the correctness of channel decoding and the verification of the instruction data frame, respectively. In this mode, the efficiency of onboard software maintenance is determined by the slowest of the three links: first, the rate of the TT&C channel (① in Fig. 1); The second is the bus data transmission rate from the OBC to the electronic unit (② in Fig. 1); The third is the rate at which the FPGA control block writes data to the program memory (③ in Fig. 1). For example, if the rate of uploading from TT&C channel is 100kbps, the transfer rate allocated by the bus is only 10kbps, and the write rate of program memory is 50kbps, then the effective rate of software maintenance can only be 10kbps. Therefore, the efficiency of onboard software maintenance is greatly reduced, and the cost of updating operation is increased accordingly. This problem will become more prominent as the satellite is configured with a higher-speed TT&C channel.

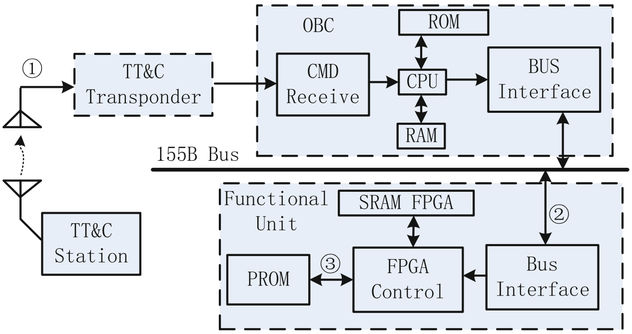

Fig. 1. Onboard software maintenance diagram for single satellite

Table 1. Configuration data size of common SRAM-based FPGAs

NO	*FPGA*	*Configuration data size (bit)*
1	XC2V3000(V2)	10,494,368
2	XC4VSX55(V4)	22,744,832
3	XC5VFX130T(V5)	49,234,944
4	XC7K325T(K7)	91,548,896
5	XC7VX690T(V7)	229,878,496

In order to solve the problems of low efficiency, the OBC can be configured with a large RAM to cache the maintenance data frames quickly uploaded from the ground into RAM, and then distribute them according to the corresponding rate [5]. With the increasing scale of FPGAs used on satellites, the amount of configuration data of some FPGAs has reached 30 Mega-bytes (see Table 1), and the SRAM configured by the OBC has been unable to meet the cache requirements of maintenance data frames, so it is necessary to deploy large-capacity SDRAM memory.

3 Networking Satellite Onboard Software Maintenance

3.1 Characteristics of Networking Satellites and Software Items

All satellites operating in a network have exactly the same or similar functions, such as one certain generation of GPS navigation satellites. Each satellite has the same software list, and the functions of the given software programs are also the same. For example, inter-satellite link data processing FPGAs, which are allocated on each satellite, are given the same functions.

Are all the software program data with the same functions on networking satellites consistent? It is needed to be divided into two cases: embedded CPU software and FPGA.

For embedded CPU software program with identical functions, the program data may also be inconsistent, mainly manifested in the inconsistency of some parameters, such as SCID. Storing these parameters as important data in the nonvolatile memory can achieve the consistency of program data. When the software is initialized and running, it loads the corresponding parameters to complete the configuration of the software function. Some embedded CPU software with different functions can be processed with different branches, and the control of which still uses state parameters. For example, the different functions of the two satellites are A and B, which are controlled by the state parameter S. If $S = 0$, execute the A function, otherwise $S = 1$, execute the B function, and the S parameter needs to be saved as an important parameter.

The configuration data of FPGA with the same function is almost consistent in networking satellites, and the configuration of some different parameters can be set by CPU software. Some satellites within networking constellation have inconsistent functions. For example, some satellites are configured with two inter satellite links, while others are configured with five inter satellite links, so the functions of inter satellite link data processing FPGA are different. Under this situation, different status satellites can be identified by making different settings on the peripheral pins of FPGA. For example, pulling down a pin (logic level is 0) and pulling up (logic level is 1) indicate the satellite configured with two and five inter satellite links respectively.

Therefore, the software program data of networking satellites with the same function can be the same, which makes it possible to realize the all networking satellites onboard software maintenance with one-time data uploads by TT&C systems. When the program data of one satellite are successfully updated in orbit, there is no need to upload data into other satellites. Sending the corresponding "copy" and "paste" commands to the satellites that have completed the update can achieve all the networking satellites software maintenance. This can greatly reduce the workload of maintenance operation.

3.2 System Design of Networking Satellites Onboard Software Maintenance

The network topology of satellites is usually divided into hybrid network and single network. The satellites of a single network are all located at the same orbital altitude. The satellites in the hybrid network are at different orbital altitudes, and interconnected by inter satellite links in the same and different orbits. For example, the BeiDou-3 Navigation Satellite System includes satellites with GEO–MEO orbital altitudes.

The maintenance design of networking satellites needs to select one satellite as the software maintenance server, which receives the uploading data from TT&C system to complete the software maintenance function of the satellite, and supports the sending of program data to other satellites through the inter satellite link.

For hybrid network satellites, the observation and control of higher orbit satellites has a long visible time, which is suitable as software maintenance server. For example, for the satellite network containing GEO orbiting satellites, GEO satellite is always visible from ground, and there are also visible arcs between GEO orbiting satellites and LEO satellites. Therefore, selecting one or more GEO as software maintenance servers can realize the software updates of the whole networking satellites.

For a single network, the design and hardware configuration of each satellite are the same, and the visibility from ground is basically the same. The visibility between satellites is related to the network topology.

Generally, it is impossible to maintain the software of the whole networking satellite with one satellite as the software maintenance server. Therefore, in a single network satellite system, each satellite can be used as a software maintenance server. After each satellite completes the reception and procession of the maintenance data, it can realize the distribution and transmission of the program data to the adjacent chain satellites.

4 Design Technologies of Software Maintenance Server

The software maintenance server is the core to realize the software updating of all the networking satellites. It has the functions of receiving, verification, storage and distribution of the uploaded data from TT&C system.

4.1 Receive and Verification of Reconstructed Data Annotation

Software maintenance server function is generally located in OBDH subsystem. Before sending the reconstructed data frame, one maintenance start instruction, which includes the software program identification, the number of maintenance data frames, is sent prior to data frames themselves. While receiving the instruction, the OBDH subsystem is ready to start receiving the maintenance data frames, which are uploaded from the ground TT&C system. The data frame format is generally virtual channel data unit (VCDU) [6], whose format is shown in Fig. 2.

The VCDU adopts the standard CCSDS format. The version number is a fixed value. The spacecraft identifier(SCID) is a unique identifier of the satellite. The virtual channel identifier (VCID) will be allocated according to the data type, and the maintenance data frame will be given a specially defined identifier. The symbol field and insertion field are fixed values, and each spacecraft will be designed and determined according to requirements. The data field contains a header and a segment of program data. The header contains the sequence information of frames and the characteristic information of the segmented program data. One software program data need to be divided into several same size data segments, which are placed in the VCDU data zone.

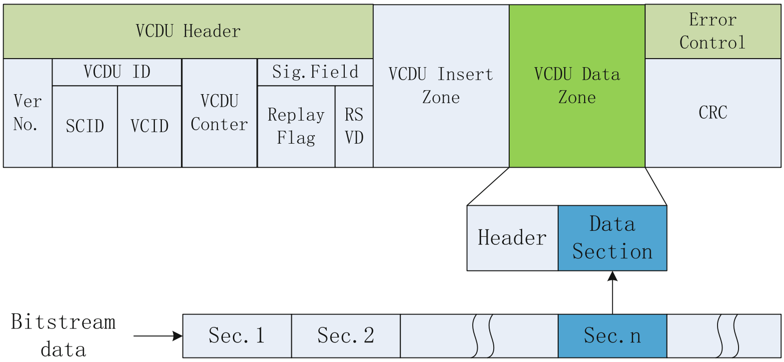

Fig. 2. VCDU data frame

After receiving the VCDU frame, the OBDH subsystem completes the CRC verification of each frame of data and temporarily stores each frame of data in RAM. When there is CRC error or frame loss according to the frame continuity interpretation, the corresponding frame will be re-uploaded from the ground. Then all frames of data are received correctly. The continuity of frames can be judged according to the VCDU counter or the frame sequence information in the header of data zone.

A program data verification instruction, which contains the verification code of the software data is sent from ground while all maintenance data frames are received correctly and completely. Then, the OBDH subsystem verifies the program data. If the verification code is consistent with the verification code in the verification instruction, it indicates that the program data uploading is correct, otherwise it needs to be uploaded again on the ground.

4.2 Maintenance Data Processing and Storage

As a software maintenance server, all uploaded maintenance data need to be stored for subsequent writing into memory or distribution to other satellites. Program data can be stored in RAM or flash. RAM is not suitable for long-term storage of maintenance data due to data loss of power failure. After receiving maintenance data, the OBDH subsystem can temporarily cache it in RAM. Flash or EEPROM is suitable for long-term data storage, because data in which can be saved even when it is powered down. Therefore, the maintenance data needs to be stored in high-capacity flash or EEPROM after receiving and checking.

According to the way of storing data, it can be divided into three ways. The first is to store the whole VCDU frame, the second is to store the VCDU data containing the header in Fig. 2 (striping the VCDU Head, VCDU Insert Zone and CRC), and the third is to store only program data (striping Header of data zone after the second way). The first way needs the most storage space. Its advantage is that it includes CRC verification, which can be used as checking code. In addition, if the configuration items of the local satellite are maintained, the format of the data frame sent and output can be VCDU format, which can reduce the re-framing expense. The second way is to store less data. At the same time, the header also contains the important information of the frame data, which can also be distributed as the data frame format maintained by the satellite software. The third way is to store the least amount of data, which only contain the binary program data. When distributing maintenance data, the software needs to re-frame. The first method can be adopted if the OBDH process capacity is limited, while the third method can be adopted if storage resources are limited. If both resources are not limited, the third method is recommended for the following reason: the data frames need to be reorganized for both ways when they are distributed to other satellites through inter satellite link. In addition, when data are stored in flash or EEPROM, it is generally necessary to encode the data to realize error correction and detection. Therefore, the CRC retained in the first method is of little significance.

4.3 Maintenance Data Distribution

When receiving the ground command to start the maintenance of a software program on the local satellite, the OBDH subsystem reads the maintenance data from flash and completes the verification, and then re-frame the data according to the local data transmission protocol to distribute the data to the destination electronic unit, which completes the receiving, storage and processing of the maintenance data. The process of software maintenance of local satellite is shown in Fig. 3.

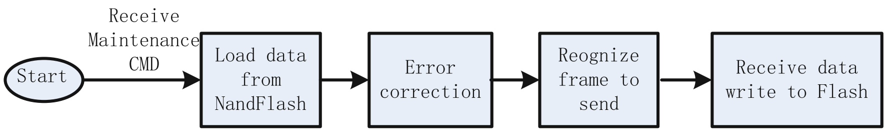

Fig. 3. Software maintenance data distribution process for local satellite

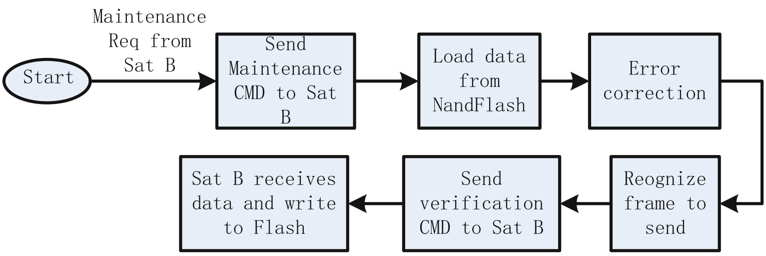

Fig. 4. Software maintenance data distribution process for slave satellite

When the local satellite receives the maintenance instructions sent by the ground to other satellites, the local satellite needs to send the following commands autonomously through the inter satellite link: a). Send the maintenance start instruction so that the destination satellite begins to prepare to receive maintenance data; b). Read out the maintenance data from flash and complete the verification, reorganize the frame according to the data frame format of the inter satellite link and send it to the destination satellite frame by frame. The destination satellite will return some telemetry information of the received maintenance frame. The local satellite can reissue the lost frame and wrong frame information according to the telemetry judgment until all data frames are correctly sent to the destination satellite; c). Send the maintenance data verification command. After the verification is correct, the destination satellite writes the data into flash to complete the receiving process of maintenance data. The whole distribution process is shown in Fig. 4.

If data frame format also adopts AOS for inter satellite link, the frame format is also mainly VCDU, which may have the following differences from the VCDU frame format uploaded to the local satellite from the ground: a). Frame length of VCDU; b). Spacecraft identifier in VCDU leader; c). VCDU insertion field; d). CRC check.

4.4 Design of Automatic Updating of Networking Satellite Software

According to the data distribution design in above section, it is necessary to send control instructions from the ground and maintain the software program one by one. The advantage of this updating method is high reliability. The designer confirms the necessity and feasibility of software updating. However, the workload of manual updating will be unbearable when the scale of networking satellites increases to hundreds.

At present, most of the ground consumer electronic product supports automatic software update. When it is detected that its software version is lower than the software version of the server, it can send an update application to the server and then update the software to the latest version with the user authorization. The software server of ground consumer electronic products has strong performance and can support all users' software update applications. The performance of the software maintenance server of the networking satellites is not much different from that of other satellites, and it is difficult to support the same automatic update method as the ground system.

In order to reduce the link length of software maintenance data transmission, it is more reasonable to use the hand-in-hand software automatic update method, which means that the data updated by the satellite software come from the adjacent satellites with established connections. All adjacent satellites regularly send version information of all their software to each other. When the software version is consistent, it does not need to be updated; otherwise, satellites with older software versions can send software update requests to "hand in hand" satellites. The specific operation process is as follows: a). The ground injects the software update program as well as version information and the applicable satellite lists into the software maintenance server Sat-A through the satellite ground link. b). Sat-A sends the software version information to "hand in hand" Sat-B through the inter satellite link. When Sat-B finds that the software version is higher than the current version itself and the applicable satellite list includes Sat-B, it sends the software update request to Sat-A. c). After receiving the software update request, Sat-A will send software update data to Sat-B according to the method in above section. d). While updating all data and runs successfully, Sat-B will send the updated software version information to the other satellites. When the version of other satellites software "hand in hand" with Sat-B is lower than this version, the same method to send a software update request to Sat-B can be achieved.

In the above method of automatic software update, there are two problems to be solved: a). When a satellite has multiple "hand in hand" satellites with software versions higher than its own, which one should it send a software update request? b). In the process of software update, the link time cannot meet the completion of data update. Should it wait until the next link building or switch to obtain software update data from other satellites?

For the first problem, in the case of non automatic, the ground selects a link to complete the software update at one time according to the satellite link conditions (such as link availability time and link rate). Under automatic conditions, after knowing all the "hand in hand" satellites currently available for software update, the target satellite needs to select a "hand in hand" satellite with the highest bandwidth for software update according to the link building planning table of the whole networking satellite. The bandwidth here is the link available time multiplied by the link rate.

For the second problem, Sat-A sends updated data to Sat-B. After finishing a part of updating, the link is interrupted. If waiting for the next link between Sat-B and Sat-A, it may take.

a long time, while the implementation is relatively simple. If there is a "hand-in-hand" Sat-C that can also send updated data, Sat-B can send a data software update request to Sat-C. At this time, the request information also includes the data package information that needs to be updated. The data package that has been received through Sat-A does not need to be re-transmitted. This implementation method is more complex. In the latter way, after rebuilding the available link between Sat-B and Sat-A, it is necessary to inform Sat-A that it is no longer necessary to send software update packets, unless B sends a software update request to Sat-A again.

5 Verification of Onboard Software Maintenance

Two space routers [7] were used for relevant maintenance tests in the laboratory. These two routers are used for one GEO and one MEO orbiting satellite respectively. The GEO satellite contains five high-speed inter satellite link ports, and the MEO satellite contains two ports. The high-speed forwarding blocks of the two routers are implemented with Virtex FPGA. The situation of the two FPGAs conforms to that described in section III and can be distinguished by setting of hardware pins. Pulling up the external pin of the FPGA of GEO satellite indicates that the FPGA has five high-speed inter satellite link ports, and pulling down the corresponding external pin of the FPGA of MEO satellite indicates that the FPGA has two ports.

The basic structure of the router for GEO satellites is shown in the Fig. 5. The difference between routers used for MEO satellites is that the ST pin of FPGA is pulled down to the ground through resistance, and there are two external high-speed inter satellite link interfaces. The on-board router includes CPU system, FPGA system and solid-state storage system. The CPU system includes processor, program memory PROM and data memory SDRAM, and the interface with FPGA system and storage system is

IO Bus. The FPGA system includes Virtex FPGA, controlling ASIC circuit and FPGA program memory Norflash. The CPU system can receive maintenance data from TT&C Circuits and write it into the program memory Norflash of FPGA through the controlling ASIC. The router can also receive maintenance data through the high-speed inter satellite link interface, forward it to the on-board CPU, and then write it into the program memory Norflash. Solid state storage system includes storage management circuit and solid state memory Nandflash (Fig. 6).

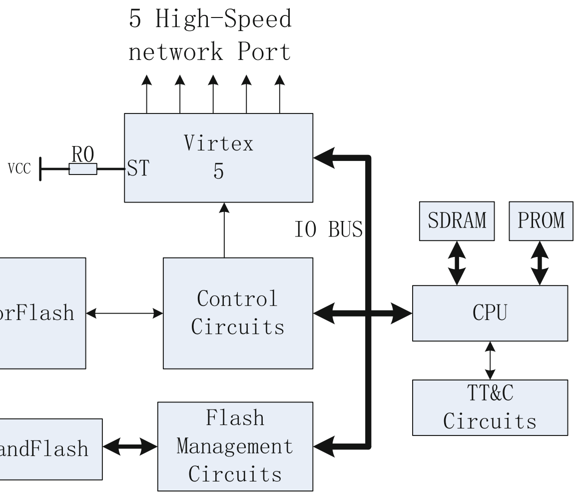

Fig. 5. Logical composition of GEO satellite on-board router

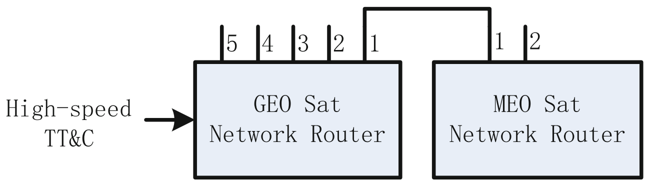

Fig. 6. Software maintenance test of networking satellites

According to one link state of the satellite in orbit, GEO satellite high-speed port one is connected with MEO satellite high-speed port one by cable, indicating that the two satellites are linked in orbit. In the initial state, the Virtex FPGA version number of both routers is 0x00, and then the program of GEO satellite router Virtex FPGA is updated to 0x01 version through the satellite ground high-speed TT&C system. At the same time, the configuration data are stored in the solid-state memory Nandflash. The version information is transmitted to the MEO satellite space router through the inter satellite link. When the MEO system finds that the program version of Virtex FPGA is lower than the FPGA program version of the GEO satellite, it immediately initiates an update application. After GEO orbiting satellite receives the request, its CPU software reads out the configuration data of version 0x01 in the solid-state memory into SDRAM and completes the verification. Then it is re-split and packaged to form maintenance data frame to send to MEO satellite, according to the inter satellite link data frame format between the two satellites. The MEO satellite stores all the configuration data in SDRAM when it receives the maintenance data frames through the inter satellite port one. After the data frames are collected completely, all the configuration data are written into the program memory Norflash of FPGA. The whole maintenance process takes about half an hour without manual intervention.

6 Conclusion

This paper describes an efficient software maintenance technology for networking satellites by using inter satellite links. Each satellite sends the version information of its maintainable software periodically. When a satellite finds that the version of a software program is lower than the software version of other satellites, it will send an application for software update, so that the software can also be updated to the latest version. After the software version of a satellite is updated on the ground, the software of the whole network can be updated automatically, which greatly reduces the time cost of ground manual maintenance operations of networking satellites. The scheme has been verified by ground tests and will be further tested and verified in orbit.

Acknowledgments. This research is sponsored by “National Key R&D Program of China” with its research number: 2022YFB2902700.

References

1. Zan, P., Cao, Y., Zhang C.: High functional density avionics system for satellites. Chin. Space Sci. Technol. **40**(1) (2020)
2. Pang, B., Hao, W., Zhang, W.: Scheme of SRAM-FPGA On-orbit reconfiguration. Spacecraft Eng. **26**(5), 51–56 (2017)
3. Wang,W., Yan, X., Xiu, Z.: Design method of online reconstruction based on FPGA. Comput. Meas. Control **28**(12) (2020)
4. Yuan, S., Qu, Z., Shao, Y.: A low cost and high reliability FPGA on orbit reconfigurable load management scheme. Space Electron. Technol. **28**(12) (2017)
5. Xiong,H., Yan, G., Li, G.: Design of on-orbit software reconfiguration system of small satellite based on minimum system. J. Telemetry Tracking Command **41**(3) (2020)
6. Zeng, L., Yan, C.: Design and hardware implementation of virtual channel link control unit and VCDU multiplexing unit. Chin. Space Sci. Technol. **27**(2), 17 (2020)
7. Wu, W., Zhou, D., Yan, H.: Desing and implementation of high-speed space router. In: Six National Space Data System Conference (2019)