The Design on SEU-Tolerant Information Processing System of the On-Board-Computer

Huang Ying[1,2], Zhang Chun-yuan[1], Liu Dong[1], Li Yi[1], and Weng Sheng-xin[2]

[1] Department of Computer Science, National University of Defense and Technology, Changsha, 410073 Hunan Province, P.R. China
[2] Department of Computer Management Center, Navy General Hospital, Beijing 100037, P.R. China
{cyzhang,yinghuangying}@nudt.edu.cn

Abstract. For SEU(Single-Event-Upsets) of space radiation environment, a multi-level fault-tolerant mechanism based-on FPGA, which has greatly improved the system's ability of resisting SEU was presented. The design of three-level fault-tolerant included the dual fault-tolerant system based-on FPGA, the module-level triple modular redundancy, and the chip-level SEU-tolerant FPGA. Finally, the evaluation for the SEU reliability of the OBC(on-board-computer) was mentioned.

Keywords: Single-Event-Upsets, Field Programmable Gate Array, Dual Fault-Tolerant, Triple Module Redundancy, Cost-Off-The-Shelf.

1 Introduction

Nowadays, the R&D(research and development) and application of the COTS-based (Cost-Off-The-Shelf) space computer system are the keystone of many universities and institutes all around the world. Moreover, as a combined product of spatial flight application and computer, one of the targets of designing COTS-based computer subsystem is to achieve low-cost, light-weight and short-development-cycle, and it is supported by using commercial device and development kits[1].

Due to the high energy particles and diversified external interferences in the spatial environment, the safty of flight, at a large extent, relies on the design of computer system reliability. In order to avoid spatial fault, the selection of devices should be seriously considered. Normally, we adopt industrial-class and commercial-class chips in the spatial COTS-based computer. However, from the view of system-design level, the most important thing is to introduce the fault-tolerant technique. Combined with the system architecture and needs of the design, we put forward a design scheme of COTS-based multi-level fault-tolerant architecture, and finally used an analytical model to validate the reliability of the design.

2 Architecture

The COTS-based space computer system architecture is shown in figure 1. The COTS-based chips of the system like SRAM, Flash, CPU and so on, are all prone to suffer from SEU, and consequently result in bad working status.

M. Xu et al. (Eds.): APPT 2007, LNCS 4847, pp. 30–39, 2007.

SEU is caused by electrification particles casting into the sensitive fields of integrate circuit[2]. It usually introduces many important errors such as CPU's inner register content changed or bit flipped. These errors may directly result in wrong computing results, wrong sequence of program executing, and even breakdown of the system. In order to adapt to spatial radiation environment and improve system's reliability of SEU-mitigation, according to the rules of design on COTS-based devices, we put forward a multi-level SEU-mitigation design scheme based on FPGA(Field Programmable Gate Array). There were FPGA-based dual fault-tolerant system at the system level, FPGA-based SEU-mitigation design of RAM and Flash at the module level and SEU-mitigation design on SRAM-based FPGA at the chip level. As a result, the system's reliability to mitigate SEU has been improved by this multi-level-tolerant mechanism.

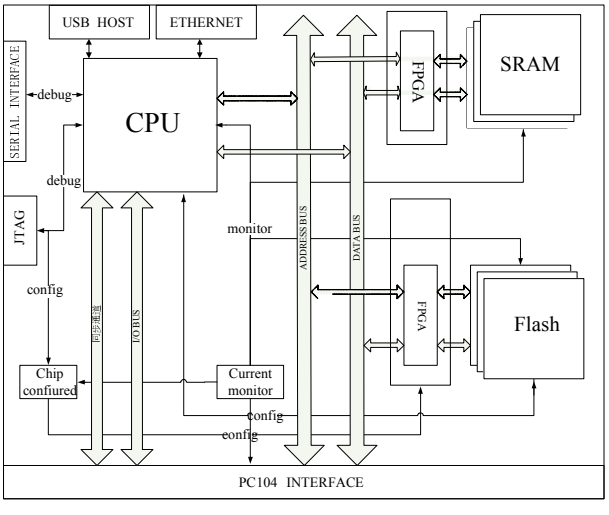

Fig. 1. The two same single boards were connected by the interface PC104 constituted of the dual fault-tolerant subsystem of the OBC

3 Design of Multi-level Fault-Tolerant System Architecture

3.1 System-Level Design of SEU Mitigation

This section introduces a kind of FPGA-based warm-back-up dual fault-tolerant system scheme. By efficiently integrated advantages and characteristics of FPGA chips, we not only implemented the mechanism of judging the fault computer of dual fault-tolerant system and resume by itself, but also arbitrated diversified data signals by MS(master symbol), so that the system would be immune to the influence of single-computer faults and hold communicating with external data.

The architecture of the adopted system-level dual fault-tolerant mechanism based on FPGA, is shown in Fig.2. The watchdog timer circuit(WDT0,WDT1) and the arbitrator(ARBITER) , which were both the key modules of the dual system,

respectively took charge of supervising its running states and arbitrating signals of the dual system like communication and control. In order to implement the system-level dual warm-back-up fault-tolerant function, the two CPU were linked by the middle interface module(the arbitrator). At the same time, FPGA was also responsible for providing the interface (arbitrated by FPGA) communicated with the external. It was supposed that the mechanism of CPU1 and WDT1 were respectively the same as CPU0 and WDT0. There are symbol explained as follows.

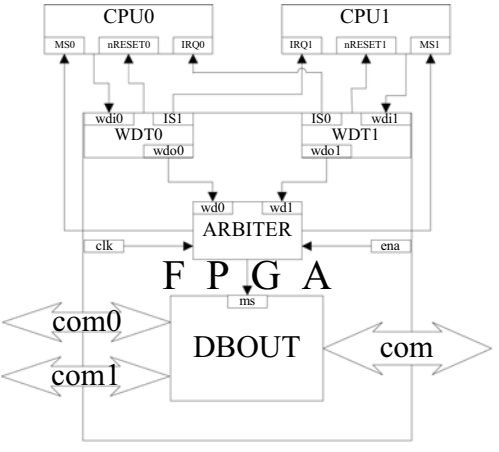

Fig. 2. The architecture of dual fault-tolerant system based on FPGA

A. CPU0 saw after 4 kinds of signals

①CPU0 periodically sent impulse signal wdi0 to WDT0, denoting that CPU0 works normally. ②CPU0 received nRESET from WDT0, in order to reset CPU0 and resume from fault state. ③CPU0 received IS0 from WDT1 to take over the system when CPU1 had been broken. ④CPU0 received MS0 simultaneously from arbitrator, in order to judge the respective current state of CPU0 and CPU1.

B. WDT0 took charge into inspecting CPU0

After the system had been electrified, CPU0 periodically sent WDT0 an impulse signal wdi0, which reset the counter of WDT0 to 0. When the next clk0(WDT0's inner clock) came, the counter of WDT0 recounted from 0. If WDT0 counted over some value(CPU0 had not sent wdi0 during a scheduled period), CPU0 was considered broken, and WDT0 would act as follows: ①WDT0 sent IS1 to CPU1, required it taking over CPU0 and kept system go on. ②WDT0 sent CPU0 nRESET0 to reset it. ③WDT0 sent low current-flow of wdo0 to the port wd0 of the ARBITER module of FPGA, in order to notify the arbitrator that CPU0 had been broken and should power on CPU1 to take the place of CPU0.

C. ARBITER was the core of the realization of dual warm-back-up mechanism

It judged the value of ms(master symbol) through changes of wd0 and wd1, and export the value of ms to DBOUT. When the system was powered on, the enable

signal ena set ms=0, wd0=0, and wd1=0, export ms=0, denoting that CPU0 was the host. After a period of time, if CPU1 had been broken(wd1=1), ms maintained value 0. If CPU0 had been broken, wd0 from WDT0 was set to 1, and simultaneously ms was set to 1. However, after CPU0 resumed from fault state, wd0 would be set to 0, and the value of ms would not be changed immediately. By the time CPU1 had been broken(it means that the value of wd1 had been changed to 1), the value of ms would be changed. In this way, it could be avoid to frequently switch the host and the guest from the dual system, by too frequently changed the value of ms. CPU could judge the host or guest from the dual system through the value of correspond master symbol MS.

D. The effect of DBOUT module

By judging the value of ms, the system port com was switched to communicate with correspond channels of each CPU. When system was powered on and initialized, com was switched into com0, and then CPU0, as the host, was responsible for communicate with external data out of system. If CPU0 appeared fault(namely ms=1), com was switched into com1, and CPU1 took instead of CPU0. The way to switch system communication channels by judging the value of ms, assured system was able to resume itself from fault state and continued to communicate with external data despite that a computer was broken in the dual system.

3.2 The Module-Level Mitigation Design on SEU

As memory device, SRAM and Flash are both prone to suffer from SEU. Usually there are two methods to mitigate SEU on memory in the module-level, EDAC(Erro Detection And Correction) and TMR(Triple Module Redundancy). With different coding measures, the capacity of fault tolerance are different. EDAC is a software fault-tolerant technique. But usually its ability to correct errors is confined to the cost and performance. And it costs much more additional delay time to check and correct errors, since comparatively opacity to the processor. Especially for high-speed device like SRAM, performance is influenced, and the complexity of design and difficulty of realization are increased as well. In this paper, we have adopted and implemented FPGA-based TMR technique to mitigate SEU on SRAM and Flash, due to the rich storage resource and based on the hypothesis that the possibility of SEU is 0 in the same bit of two memory cell.

3.2.1 SRAM-Mitigation Design

To improve the reliability at the maximum extent, and to reduce memory cells' upsets caused by high energy particles, we used three identified memory cells arbitrated by a controller to form a TMR fault-tolerant mechanism. The cell could be arbitrated out by TMR arbitrator, when it met SEU, and the system continued working freely. For the adequate memory resources, the upset cells could be scrubbed into right value instead of wrong[3]. Fig.3 shows the logic circuit of TMR.

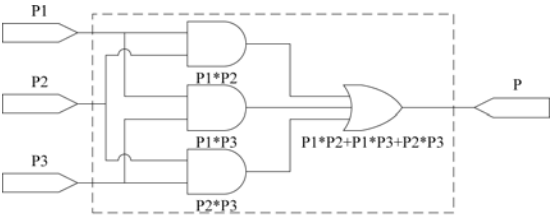

Fig. 3. The logic circuit of TMR

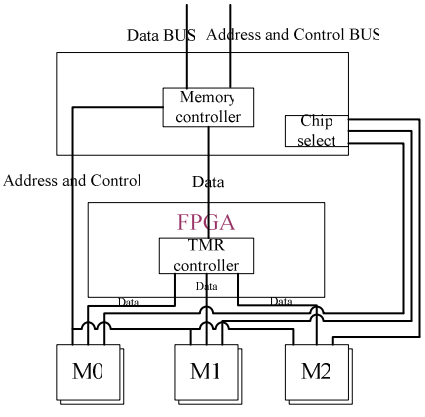

Fig. 4. The function of the arbitrating controller of TMR

Fig. 4. illustrates the function of the FPGA-based TMR arbitrator controller.

As Shown in the Fig.4, triple voting is typically used for functional blocks with high clock rates of at least 66 MHz between a processor and banks of volatile memory. To maximize the processor performance, this mechanism must allow for zero wait-state operation such that the fetch and write performance is not degraded for additional delays in implementing this voting methodology. From the processor perspective, triple voted memory is therefore achieved to perform the voting function within the same CPU read-memory request transaction; it is completely transparent to the processor. The general procedure to operate a triple voting memory mechanism is shown in Fig.5 [4] .

A typical triple voted controller was implemented in a radiation-tolerant FPGA and included the memory data integrity logic and circuitry, which were needed to enhance system reliability in space environment applications.

3.2.2 The SEU-Mitigation Design on Flash

Flash was used to reserve programs and data of the system while working. In order to assure correctness of the data, FPGA-based TMR technique was used to mitigate SEU. Data can directly be read out and written back by Flash(without memory controller), so it could be easily immune to SEU as long as TMR arbitrating mechanism. One of the advantages was transparent for the processor to directly access and execute instructions from Flash, without the processor judging whether it was correct.

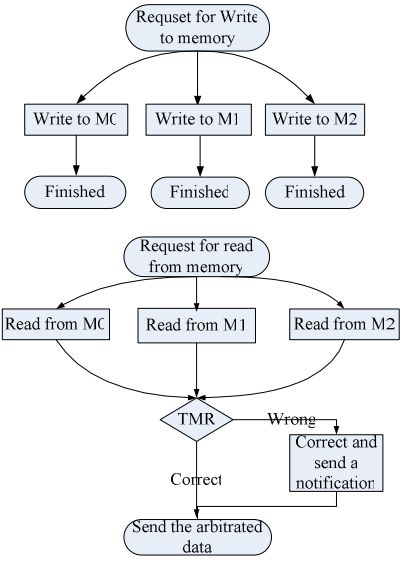

Fig. 5. The mechanism of memory controller of TMR

3.3 The Chip-Level Mitigation Design on SEU

The arbitrator of dual system and TMR arbitrating controller, on the above, were both implemented on a basis of a SEU-hardened FPGA, including memory data integrated logic, so as to improve the reliability of spatial application.

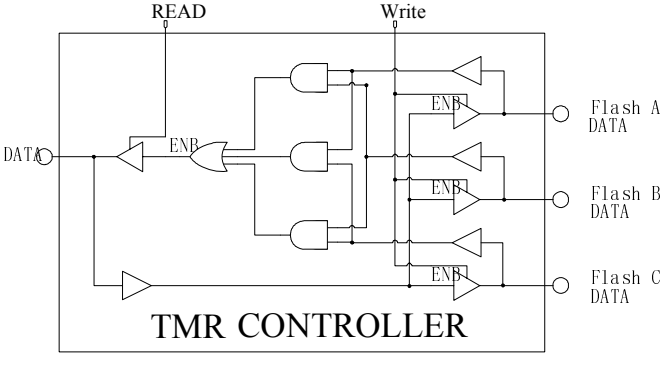

Fig. 6. The logic circuit of TMR of FLASH

Considering overseas import limit and the cost of SEU-hard product, we used SRAM-based FPGA based on COTS. There were several advantages, low-price, programmable, and widely-used, but it was easily incurred SEU. So, as bottleneck of system's reliability, SEU-mitigation design on FPGA was a sticking point of the system design.

3.3.1 The Inner Fault-Tolerant Design on FPGA

A. The redundant design on I/O pins

The signals of I/O pins were all triple-voted in the inner logic cells of FPGA. Then data signals and control signals could bear the effect of SEU while were being processed via inner logic of FPGA. As Shown in Fig.7, there was a SEU-tolerant design principle chart.

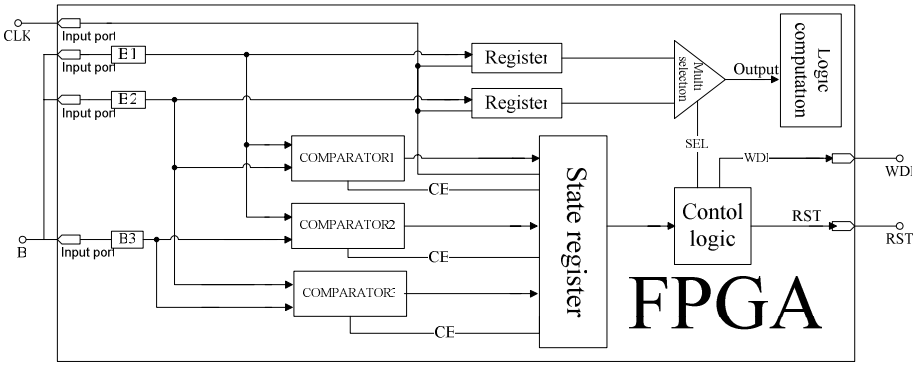

Fig. 7. The inner SEU-tolerant redundant architecture of FPGA

B. The dual-module redundant design on inner logic module

We have adopted the dual-module redundant design on the inner logic module M of FPGA. The thought was to integrate the advantages of high-speed hardware-based and economy time-based redundancy, which saved nearly 1/3 redundant hardware resource compared with traditional method TMR. The Scheme is shown in the fig.8 as follows: at the time of t, the input signal Input was exported to R(t) through the module M; then Input came by encoding and decoding to the redundant signal R(t+d). if the result compared was 1, which means the two signals reached the comparator C at the same time, R would be output, else returned a recalculation flag to reset the signal R from Input to M, and export the result to the signal Output.

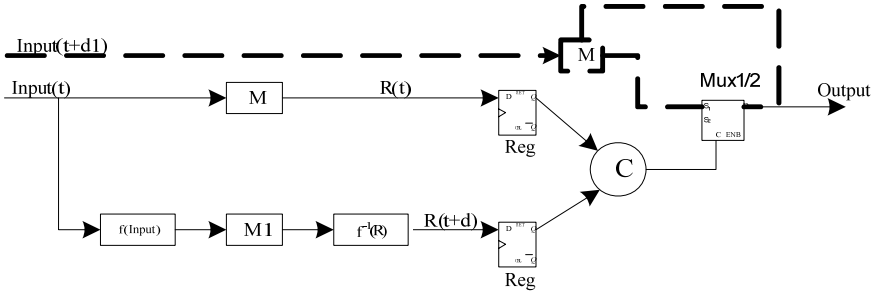

Fig. 8. The dual module redundancy design of the inner logic module of FPGA

C. The outer reconfigurable fault-tolerant design on FPGA

As the key to the implementation of inner hardware logic redundancy technique, the arbitrator was the bottleneck of the system reliability. So we have setup a watchdog (seen in the Fig.7) in the inner controller of FPGA. Once the watchdog sending an overtime signal, the logic result of the FPGA was wrong, and then it would resume from fault state through the online reconfiguration operation. However, in the course of deigning on computer system of spacecraft, we must implement its ability of auto-fault-repair. Thereby according to the characteristics of configuration of SRAM-based FPGA, we have designed an auto-reconfiguration circuit with watchdog. The typical configuration-circuit signal elements chart of SRAM-based FPGA was shown in the Fig.9, and the auto-repair function was triggered by the signal CONF_DONE.

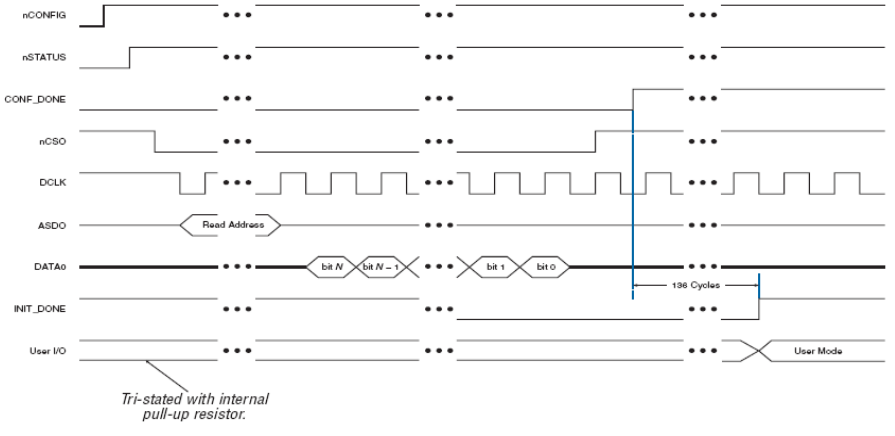

Fig. 9. The typical configuration-circuit signal elements chart of SRAM-based FPGA. When watchdog had alarmed, the signal CONF_DONE would be set to low level to reset the system.

The signal RST, the timeout alarm signal of watchdog, was connected with CONF_DONE. After the FPGA had been powered on and been configured, CONF_DONE became high level from low. Once the watchdog alarmed, RST would set CONF_DONE to low level, then the circuit of FPGA would restarted, and the inner logic circuits would be downloaded and auto-reconfigured from storage.

4 Analysis of the Reliability

The entire spacecraft computer subsystem was nonobjectively seen as a mixed combination system with serial and parallel connection. Seen in the Fig.10, the dual system came differently by fault-tolerant reinforcing to be the dual fault-tolerant system with parallel connection.

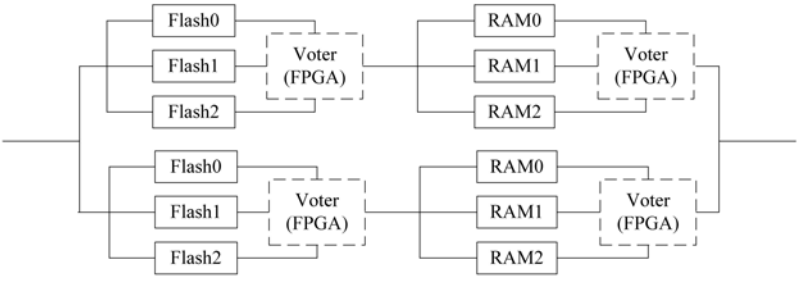

Fig. 10. The model on the reliability of SEU of OBC

4.1 The Reliability of a Single Board

Shown in the Fig.10, the single board was also the mixed combination model. The SEU-tolerant reliability of its modules (concluding Flash and RAM) was represented by Rm (Rm=R1R2), thereinto R1 denoted the reliability of Flash (or RAM) and R2 denoted the arbitrator of FPGA. λ denoted the SEU-invalidation effects, Rf and Rr respectly denoted the reliability of triple-module-redundant system of Flash and RAM, and the reliability of CPU in a single board was denoted by Rcpu. In the thesis, we have supposed that there were not more than one-bit SEU-upset in synchronization, and the invalidation effects of each module in the parallel connection modules was equal.

From the foregone conditions, results were obtained as follows.

$$R_{f10}=R_{f11}=R_{f12}=R_2=e^{-\lambda t}. \tag{1}$$

$$R_1=(e^{-\lambda t})^3+3(1-e^{-\lambda t})(e^{-\lambda t})^2=3e^{-2\lambda t}(1-e^{-\lambda t}). \tag{2}$$

$$R_m=R_1R_2=3e^{-3\lambda t}(1-e^{-\lambda t}). \tag{3}$$

$$R_f=R_r=R_m. \tag{4}$$

So the reliability of a single board could be denoted by

$$R_{cpu}=R_m{}^2=9e^{-6\lambda t}(1-e^{-\lambda t})^2. \tag{5}$$

4.2 The Reliability of the Dual System

From the Fig.10, the host and the spare made up of a mixed combination model through the judge controller of FPGA. The SEU-tolerant reliability of this system was denoted by Rd (Rd=Rcpu2). The equations were obtained as follows, according to the same hypothesis and known conditions presented before.

$$R_{cpu0}=R_{cpu1}=R_fR_r=R_m{}^2. \tag{6}$$

So the reliability of entire system could be denoted by

$$R=1-(1-R_m{}^2)^2=1-(1-9e^{-6\lambda t}(1-e^{-\lambda t})^2)^2. \tag{7}$$

Table 1. The SEU-tolerant reliability of OBC

Time(year)	R_m	R
1	0.9802	0.9996
2	0.8976	0.9895
3	0.7712	0.8600

In case of the disable probability of a single device was 10-7, and then the SEU-tolerant reliability of spacecraft computer system in apiece time extent could be shown in the table 1.

According to the reference[5], the disabled probability of computer subsystem should be less than 0.05 in two years. The result of above shows that its reliability satisfied the request.

5 Conclusion

With consideration of the characteristics of COTS-based devices and radiation effects in outer space, this thesis illustrated the design on SEU-tolerant architecture of COTS-based OBC. The fault-tolerant technology of its subsystem must give attention to the volume & power efficiency of spacecraft and meet the needs of appropriate redundancy and adequate reliability and the commercial request for low cost and high performance of COTS. So we analyzed function and logic structure of the OBC firstly. Secondly, we put forward an embedded design scheme with SEU-tolerant architecture. Finally, its reliability has been validated in theory. In the future, we will carry out the system to be validated in the spatial environment.

References

1. Du, W., Tan, W.: Research on Realizationof ASIC in Chinese Spacecraft. Chinese Space Science and Technology 5 (2002)
2. Reorda, S., Paccagnella, A.: Analyzing SEU Effects in SRAM-based FPGAs. IOLTS2003: IEEE International On-Line Testing Symposium , 119–123 (2003)
3. Pratt, B., Johnson, E., Wirthlin, M., Caffrey, M., Morgan, K., Graham, P.: Improving FPGA Design Robustness with Partial TMR. In: MAPLD 2005, Washington, D.C. (September 2005)
4. Lai, A.: Mitigation techniques for electronics in Single Event Upset environments[EB/OL]. US: Opensystems Publishing, Military Embedded Systems, [2006-6-27]. (2006), http://www.mil-embeded.com/articles/authors/lai
5. Xiang, L., Qu, G.: The Fault Tolerant System Design of Housekeeping Computer for small satellite. Aerospace Control 2, 92–96 (2005)