# 1

# Introduction

A control computer is one of the key equipment in a spacecraft control system. Its reliability is critical to the operations of the spacecraft. Furthermore, the success of a space mission hinges on failure-free operation of the control computer. In a mission flight, a spacecraft's long-term operation in a hostile space environment without maintenance requires a highly reliable control computer, which usually employs multiple fault-tolerance techniques in the design phase. With focus on the spacecraft control computer's characteristics and reliability requirements, this chapter provides an overview of fundamental fault-tolerance concepts and principles, analyzes the space environment, emphasizes the importance of fault-tolerance techniques in the spacecraft control computer, and summarizes the current status and future development direction of fault-tolerance technology.

## 1.1 Fundamental Concepts and Principles of Fault-tolerance Techniques

Fault-tolerance technology is an important approach to guarantee the dependability of a spacecraft control computer. It improves system reliability through implementation of multiple redundancies. This section briefly introduces its fundamental concepts and principles.

### 1.1.1 Fundamental Concepts

"Fault-tolerance" refers to "a system's ability to function properly in the event of one or more component faults," which means the failure of a component or a subsystem should *not* result in failure of the system. The essential idea is to achieve a highly reliable system using components that may have only standard reliability [1]. A fault-tolerant computer system is defined as a system that is designed to continue fulfilling assigned tasks even in the event of hardware

COPYRIGHTED MATERIAL

*Fault-Tolerance Techniques for Spacecraft Control Computers*, First Edition.
Mengfei Yang, Gengxin Hua, Yanjun Feng and Jian Gong.
© 2017 National Defense Industry Press. All rights reserved.
Published 2017 by John Wiley & Sons Singapore Pte. Ltd.

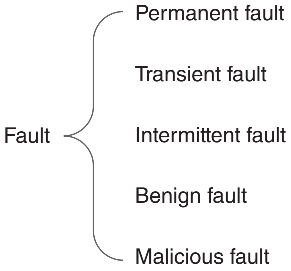


**Figure 1.1** Fault categorization.

faults and/or software errors. The techniques used to design and analyze fault-tolerant computer systems are called fault-tolerance techniques. The combination of theories and research related to fault-tolerant computer techniques is termed fault-tolerant computing [2–4].

System reliability assurance depends on the implementation of fault-tolerance technology. Before the discussion of fault-tolerance, it is necessary to clarify the following concepts [4,5]:

1) Fault: a physical defect in hardware, imperfection in design and manufacturing, or bugs in software.
2) Error: information inaccuracy or incorrect status resulting from a fault.
3) Failure: a system's inability to provide the target service.

A fault can either be explicit or implicit. An error is a consequence and manifestation of a fault. A failure is defined as a system's inability to function. A system error may or may *not* result in system failure – that is, a system with a fault or error may still be able to complete its inherent function, which serves as the foundation of fault-tolerance theory. Because there are no clearly defined boundaries, concepts **1**, **2**, and **3** above are usually collectively known as "fault" (failure).

Faults can be divided into five categories on the basis of their pattern of manifestation, as shown in Figure 1.1.

"Permanent fault" can be interpreted as permanent component failure. "Transient fault" refers to the component's failure at a certain time. "Intermittent fault" refers to recurring component failure – sometimes a failure occurs, sometimes it does *not*. When there is no fault, the system operates properly; when there is a fault, the component fails. A "benign fault" only results in the failure of a component, which is relatively easy to handle. A "malicious fault" causes the failed component to appear normal, or transmit inaccurate values to different receivers as a result of malfunction – hence, it is more hostile.

Currently, the following three fault-tolerant strategies are utilized [4–6]:

1) Fault masking. This strategy prevents faults from entering the system through redundancy design, so that faults are transparent to the system, having no influence. It is mainly applied in systems that require high reliability and real-time performance. The major methods include memory

correction code and majority voting. This type of method is also called static redundancy.

2) Reconfiguration. This strategy recovers system operation through fault removal. It includes the following steps:
   - Fault detection – fault determination, which is a necessary condition for system recovery;
   - Fault location – used to determine the position of the fault;
   - Fault isolation – used to isolate the fault to prevent its propagation to other parts of the system;
   - Fault recovery – used to recover system operation through reconfiguration.

   This method is also defined as dynamic redundancy.
3) Integration of fault masking and reconfiguration. This integration realizes system fault-tolerance through the combination of static redundancy and dynamic redundancy, also called hybrid redundancy.

In addition to strategies **1**, **2**, and **3** above, analysis shows that, in certain scenarios, it is possible to achieve fault-tolerance through degraded redundancy. Since degraded redundancy reduces or incompletely implements system function, this book will not provide further discussion on it.

The key to fault tolerance is redundancy – no redundancy, no fault-tolerance. Computer system fault-tolerance consists of two types of redundancies: time redundancy and space redundancy. In time redundancy, the computation and transmission of data are repeated, and the result is compared to a stored copy of the previous result. In space redundancy, additional resources, such as components, functions or data items, are provided for a fault-free operation.

Redundancy necessitates additional resources for fault-tolerance. The redundancies in the above two categories can be further divided into four types of redundancies: hardware redundancy, software redundancy, information redundancy, and time redundancy. In general, hardware failure is solved with hardware redundancy, information redundancy, and time redundancy, while software failure is solved with software redundancy and time redundancy.

1) Hardware redundancy: In this type of redundancy, the effect of a fault is obviated through extra hardware resources (e.g., using two CPUs to achieve the same function). In this scenario, the failure of one CPU can be detected through comparison of the two results. If there are triple CPUs, masking of one CPU's failure is achieved through majority voting – a typical static redundancy strategy. It is possible to set up a dynamic fault-tolerant system through multiple hardware redundancies, such that backup components replace the ones that fail. Hybrid redundancy incorporates static and dynamic redundancy. Hardware redundancy, which ranges from simple backup to complex fault tolerance structures, is the most widely used and basic redundancy method, and is related to the other three because they all need extra resources.

2) Software redundancy: In this type of redundancy, faults are detected and fault tolerance achieved by using extra software. Using the rationale that different people will *not* make the same mistake, fault tolerance is achieved by developing different versions of the same software using different teams, to avoid the same errors being induced by certain inputs.
3) Information redundancy: This type of redundancy achieves fault-tolerance through extra information (e.g., error correcting code is a typical information redundancy method). Information redundancy needs the support of hardware redundancy to complete error detection and correction.
4) Time redundancy: In this type of redundancy, fault detection and fault-tolerance are achieved over time – for example, a user may repetitively execute certain program on certain hardware, or adopt a two-out-of-three strategy with the result for an important program.

Because of the extra resources involved, redundancy inevitably affects system performance, size, weight, function, and reliability. In the design phase of a computer system with high-reliability requirement, it is necessary to balance all application requirements to select the appropriate redundancy method and fault tolerance structure. In order to reflect all aspects of a fault-tolerant computer system's implementation and research, this book covers system architecture, fault detection, bus, software, FPGA, and fault injection, and introduces intelligence fault tolerance technology.

### 1.1.2 Reliability Principles

#### 1.1.2.1 Reliability Metrics

Qualitative and quantitative analysis and estimation are essential in the design of fault-tolerant computer systems. The major features involved are reliability, availability, maintainability, safety, performability, and testability, with each feature having its own qualitative and quantitative specifications [4,5,7].

1) **Reliability and its measurement ($R(t)$)**

   Reliability is the ability of a system to function under stated time and conditions. Assume that the system is operating normally at $t_0$. The conditional probability that the system is operating normally at $[t_0, t]$ is defined as the system's reliability degree at time $t$, denoted as $R(t)$. Further, the conditional probability of the system operating abnormally at $[t_0, t]$ is defined as the system's unreliability degree at time $t$, denoted as $F(t)$. Reliability and unreliability have the following relationship:

   $$R(t) + F(t) = 1$$

   The failure probability density function can be calculated according to the system's unreliability, i.e., $f(t) = \frac{dF(t)}{dt}$.

2) **Availability and its measurement (*A*(*t*))**
   Availability is the proportion of time a system is in a functioning condition. The normal operation of the system at time $t$ is defined as the system's availability degree at $t$, denoted as $A(t)$. This is also termed transient availability, with mathematical expectation called steady-state availability.
3) **Maintainability and its measurement (*M*(*t*))**
   Maintainability is the ability of a system to recover its required function when the system operates under specified conditions, and is repaired following specified procedures and methods. Assume that the system failed at $t_0$; the probability of the system being successfully repaired within $[t_0, t]$ is defined as its maintainability degree, denoted $M(t)$.
4) **Safety and its measurement (*S*(*t*))**
   Safety is the nature of a system *not* to endanger personnel and equipment. Assume that the system is operating normally at $t_0$. The probability $R(t)$ that the system is operating normally at $[t_0, t]$, plus the conditional probability of the system being in failsafe mode, is defined as the safety degree of the system at $[t_0, t]$, denoted $S(t)$. Failsafe mode is a mode in which the system stops functioning without jeopardizing human life. Therefore, high reliability results in high safety, but high safety does *not* necessarily result in high reliability.
5) **Performability and its measurement (*P*(*L, t*))**
   Performability is the ability of a system to maintain part of its function and gracefully degrade when failure occurs. The probability of the system operating at performance of level $L$ or above is defined as the system performability degree at time $t$, denoted $P(L, t)$. Reliability requires that all functions be properly operational, whereas performability requires only that a portion of the functions be properly operational.
6) **Testability and its measurement**
   Testability is the extent of how difficult a system can be tested, detected, and fault-located – that is, how difficult and complex the test can be. There is currently no universal definition of testability degree; consequently, it is usually measured with test cost.

In summary, fault tolerance is a system's ability to complete its targeted function in the presence of a fault. Fault tolerance techniques are the major methods employed to achieve system reliability. Of the six features described above, reliability and availability are the most important. Therefore, we focus on these two features in the ensuing discussion.

Because $R(t)$ is the probability of the system being in continuous operation within $[t_0, t]$, system mean time to failure (MTTF) and mean time between failure (MTBF) are closely related. MTTF is the time that has elapsed before system failure. MTTF is the mathematical expectation of unrepairable system operating time before failure, i.e.,

$$MTTF = \int_0^\infty t \cdot f(t)dt = -\int_0^\infty t \cdot dR(t) = -[tR(t)]_0^\infty + \int_0^\infty R(t)dt = \int_0^\infty R(t)dt$$

When the system lifetime follows an exponential distribution, that is, $f(t)$ is a constant $\lambda$, i.e. $R(t) = e^{-\lambda t}$, then:

$$MTTF = \int_0^\infty e^{-\lambda t} dt = \frac{1}{\lambda}$$

For a repairable product, MTBF is the mean time between two consecutive failures. Let MTTR represent system recovery time, the difference between MTTF and MTBF is specified by the following equation:

$$MTBF = MTTF + MTTR$$

Availability $A(t)$ is the proportion of time during which the system is available within $[t_0, t]$ (proportion of normal operation time vs. total operation time). It can be calculated using MTBF, MTTF, and MTTR – that is:

$$A(t) = MTTF/MTBF = MTTF/(MTTF + MTTR)$$

The definition of MTTF and MTBF verifies that reliability and availability are *not* positively correlated – that is, high availability does *not* necessarily result in high reliability. For example, given a system that fails once per hour, its recovery time is 1 second and its MTBF is 1 hour, which is quite low, but its availability is $A = 3599/3600 = 0.99972$, which is very high.

### 1.1.2.2 Reliability Model

A reliability model is widely used at the design phase of a computer system to calculate, analyze, and compare its reliability. As described in the following section, the reliability model includes serial connection, parallel connection, and multiple modular redundancy.

1) **Serial connection model**

   A serial connected system is a system in which the failure of one unit will cause the failure of the entire system. Its reliability model is a serial model, shown in Figure 1.2, which is the most widely used model:

   In a serial connected system, if every unit follows the exponential distribution, then the mathematical model of the serial model is:

$$R(t) = \prod_{i=1}^{n} R_i(t) = \prod_{i=1}^{n} e^{-\int_0^t \lambda_i(t)dt} \tag{1-1}$$

   where:
   $R(t)$ is the reliability of the system;
   $R_i(t)$ is the reliability of each unit;
   $\lambda_i(t)$ is each unit's probability of failure;
   $n$ is the total number of units.

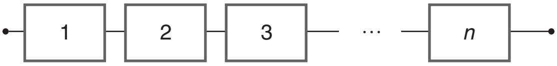


**Figure 1.2** Serial connection model.

The lifetime of the system follows an exponential distribution if the lifetime of each unit follows exponential distribution. The failure probability of the system $\lambda$ is the summation of the failure probability of each unit $\lambda_i$, as shown in the following equation:

$$\lambda = -\frac{\ln(R(t))}{t} = -\sum_{i=1}^{n}\frac{\ln(R_i(t))}{t} = \sum_{i=1}^{n}\lambda_i$$

The MTBF is:

$$MTBF = \frac{1}{\lambda} = \frac{1}{\sum_{i=1}^{n}\lambda_i}$$

Equation (1-1) shows that the reliability of a system is the product of each of its unit's reliability. The more units there are, the lower the system reliability. From a design point of view, in order to improve the reliability of a serial connected system, the following measures may be taken:

- minimize the number of units in the serial connection;
- improve the reliability of each unit, reduce its probability of failure $\lambda_i(t)$;
- reduce operation time $t$.

2) **Parallel connection model**

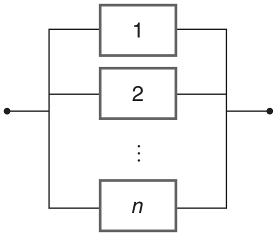
A parallel connected system is a system in which only the failure of all units will cause the failure of the system. Its reliability model is a parallel model, shown in Figure 1.3, which is the simplest and most widely used model with backup:

The mathematical model of the parallel model is:

$$R(t) = 1 - \prod_{i=l}^{n}(1 - R_i(t)) \tag{1-2}$$

where:
$R(t)$ is the reliability of the system;
$R_i(t)$is the reliability of each unit; and
$n$ is the total number of units.

**Figure 1.3** Parallel connection model.



For the usual two-unit parallel system, if the lifetime of each unit follows an exponential distribution, then:

$$R(t) = e^{-\lambda_1 t} + e^{-\lambda_2 t} + e^{-(\lambda_1 + -\lambda_2)t}$$

$$\lambda(t) = \frac{\lambda_1 e^{-\lambda_1 t} + \lambda_2 e^{\lambda_2 t} - (\lambda_1 + \lambda_2) e^{-(\lambda_1 + \lambda_2)t}}{e^{-\lambda_1 t} + e^{\lambda_2 t} - e^{-(\lambda_1 + \lambda_2)t}}$$

Equation (1-2) shows that although the unit probability of failure $\lambda_1 \lambda_2$ is a constant, the parallel system probability of failure $\lambda$ is *not*. For a system with $n$ parallel connected identical units, if the lifetime of each unit follows an exponential distribution, the system reliability is given by:

$$R_i(t) = 1 - \left(1 - e^{-\lambda t}\right)^n$$

Compared to units with no backup, the reliability of the system is significantly improved. However, the amount of improvement decreases as more parallel units are incorporated into the system.

3) **Multiple Modular redundancy (*r/n*(*G*)) model**

Consisting of $n$ units and a voting machine, a system that operates normally when the voting machine operates normally and the number of normal units is no less than $r(1 \le r \le n)$ is defined as an $r/n(G)$ voting system. Its reliability model is defined as the $r/n(G)$ model, shown in Figure 1.4, which is a type of backup model.

The mathematical form of the $r/n(G)$ model is:

$$R(t) = R_m \sum_{i=r}^{n} C_n^i R(t)^i (1 - R(t))^{n-i}$$

where:
$R(t)$ is the reliability of the system;
$R(t)^i$ is the reliability of each of the system's units (identical for each unit); and
$R_m$ is the reliability of the voting machine.

If the reliability of each unit is a function of time and its lifetime failure probability $\lambda$ follows an exponential distribution, then the reliability of the $r/n(G)$ system is:

$$R(t) = R_m \sum_{i=r}^{n} C_n^i e^{-i\lambda t} \left(1 - e^{-\lambda t}\right)^{n-1}$$

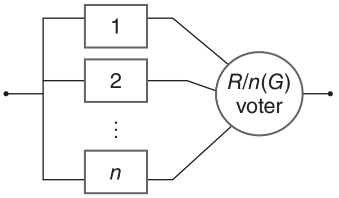


**Figure 1.4** The *r*/*n*(*G*) model.

Assuming $n$ is an odd number, an $r/n(G)$ system that operates normally when the number of normal units is no less than $k+1$ is defined as a majority voting system. A majority voting system is a special case of an $r/n(G)$ system. A two-out-of-three system is the most common majority voting system.

When the reliability of the voting machine is one, and the failure probability of each unit is a constant $\lambda$, the mathematical form of the majority voting model is:

$$R(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}$$

If $r=1$, $r/n(G)$ is a parallel connected system, and the system reliability is:

$$R(t) = 1 - (1 - R(t))^n.$$

If $r=n$, $r/n(G)$ is a serial connected system, and the system reliability is:

$$R(t) = R(t)^n.$$

## 1.2 The Space Environment and Its Hazards for the Spacecraft Control Computer

A spacecraft control computer is constantly exposed to a complex environment that features factors such as zero gravity, vacuum, extreme temperature, and space radiation, in addition to maintenance difficulties. The complexity of the operating environment challenges the computer and often results in system faults during orbit missions [8]. It is particularly important to implement fault tolerance techniques in a spacecraft control computer.

### 1.2.1 Introduction to Space Environment

#### 1.2.1.1 Solar Radiation

Solar radiation is the most active and important aspect of space radiation. Long-term observation shows that solar activity can be categorized into two types, based on the energy levels of the particles and magnetic flux released: slow type and eruptive type. Each type has its own radiation effect on a spacecraft.

In slow type activity, the corona ejects solar winds comprising electrons and protons as major components, accounting for more than 95% of the total ejected mass, with speeds of up to 900 km/s. Helium ions account for 4.8%, and other particles account for an even smaller percentage [9]. In a solar wind, the flux of a low-energy particle is high, and that of a high-energy particle is low. In the quiet period of solar minima, particles at 1 AU (150 000 000 km) consist of low-energy solar wind and a few galactic cosmic rays (GCRs).

Eruptive solar activity includes coronal mass ejection (CME) and solar flares, which can also be called a solar particle event (SPE), a solar proton event, or a relativity proton event. During an eruptive solar activity, streams of charged particles and high-energy radiation are released into space. The speed of the high-energy particles exceeds 2000 km/s. In the most severe five minutes of the eruption, most particles at 1 AU are high-energy particles, with a flux level higher than that of the solar quiet period by several orders of magnitude.

In the 11-year solar cycle, the probability of CME and flare is low during solar minima and high during solar maxima. Compared with the static slow activity, eruptive solar activity is a low probability event with a very short duration and very low energy, but very high power. As the flux level of eruptive solar activity is higher than that of slow solar activity by several orders of magnitude, it has a severely destructive effect on space electronics and astronauts and, hence, is the focus of space radiation research.

With ion emission, the above two types of solar activities emit interplanetary magnetic fields as well. The magnetic field intensity of eruptive solar activity is extremely high and interacts with the magnetic field of the earth, thereby negatively affecting low orbit satellites and the earth environment.

#### 1.2.1.2 Galactic Cosmic Rays (GCRs)

GCRs are from the outer solar system and feature very low ion densities, extremely high-energy levels, and isotropic emissions. They comprise 83% protons, 13% ammonium ions, 3% electrons, and 1% other high-energy level particles. The total energy and flux of a GCR are extremely low. During solar maxima, the GCR flux increases slightly. Conversely, during solar minima, the GCR flux decreases slightly.

#### 1.2.1.3 Van Allen Radiation Belt

The interplanetary magnetic field emitted from solar activity interacts with the magnetic field of the earth, and deforms the earth's magnetic sphere in such a manner that the sun side is compressed and the other side is extended. This effect redirects the charged ions emitted towards the earth to leave the earth along a magnetotail direction, so that nature of the earth may evolve. The shape of the earth's magnetic layer resembles that of a comet's tail, as shown in Figure 1.5.

Those ions that cross the magnetopause and arrive in the vicinity around the earth are captured by the earth's magnetic field. These captured ions form an inner/outer belt around the earth, with the south and north poles as its axis. The capture zone was first discovered by Van Allen and, hence, can also be called the Van Allen radiation belt. The inner belt is situated in a shell space above the meridian plane within latitude range ± 40° (the shell space is above the equator at an altitude in the range 1.2 L to 2.5 L, where L is the radius

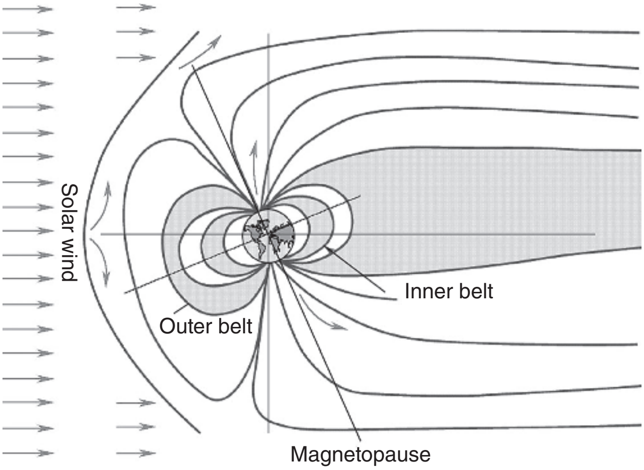


**Figure 1.5** Magnetic layer of the earth and radiation.

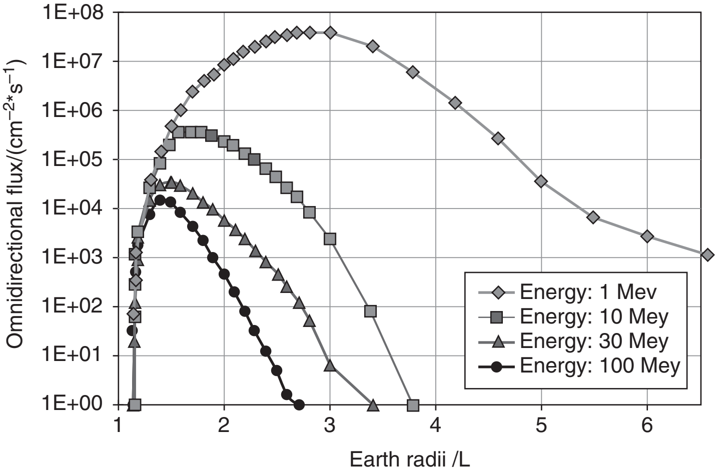


**Figure 1.6** Energy spectrum of protons.

of the earth, L ≈ 6361 km, L = 1 means on the surface of the earth). Protons and electrons constitute most of the inner belt. The outer belt is situated in a shell space above the meridian plane within the latitude range ± 55° to ± 70° (the shell space is above the equator at an altitude in the range 2.8 L to 12 L). The relationship of flux and the position of the protons and electrons inside the Van Allen radiation belt are shown in Figures 1.6 and 1.7.

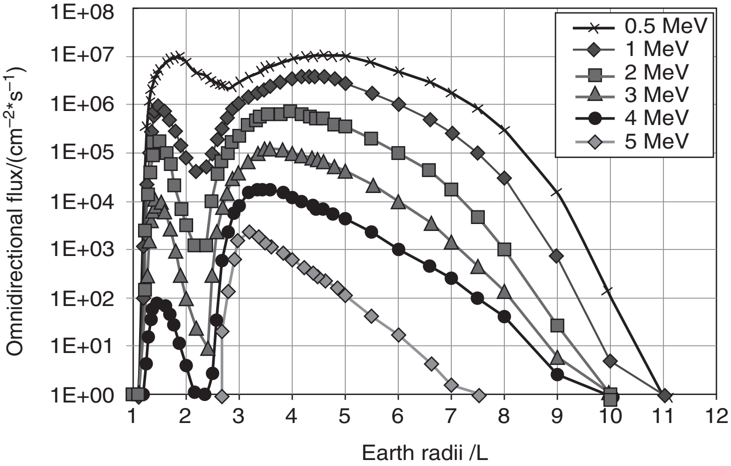


**Figure 1.7** Electron distribution above the equator.

Because of the inhomogeneity of the earth's magnetic field, there are high-energy level protons at an altitude of 200 km above the South Atlantic negative abnormal zone. In addition, the accumulation of magnetic force lines at the polar zone leads to an increase in the high-energy level particle flux in those areas [10].

The components and distribution of high-energy level particles within the Van Allen radiation belt are stable when there is no eruptive solar activity. However, when there is eruptive solar activity, or the planetary magnetic field disturbs the earth's magnetic field, the high-energy level particles' flux and spectrum increase significantly, and the Van Allen radiation belt moves closer to the earth. As a result, geo-satellite electrical facilities (even ground facilities) will fail.

#### 1.2.1.4 Secondary Radiation

When original high-energy level particles penetrate a spacecraft's material, a nuclear reaction is produced, which in turn excites secondary particles and rays, including the strong penetrative types, such as bremsstrahlung and neutrons.

#### 1.2.1.5 Space Surface Charging and Internal Charging

Surface charging results from plasma and the photoelectric effect. Because the mass of electrons in plasma is significantly lower than that of other particles, the speed of the electrons is correspondingly significantly higher than that of other particles. When a satellite is immersed in the uncharged plasma, first a large number of electrons and a small number of other particles are deposited onto the surface of the satellite, to form electron collection current $I_e$ and ion collection current $I_i$; the surface of the material produces secondary electronic radiation and ions, which form surface-deposited ion radiation current $I_{si}$, leaving electronic radiation current $I_{se}$; and the impact of incident electrons on the surface of the material produces reflected electrons to form reflected electron current $I_b$. If the material is situated in a lit region, the surface emits photons, which forms photon photoelectric current $I_p$. Hence, the total current on the surface of the material is given by $I_t = I_e - (I_i + I_{si} + I_{se} + I_b + I_p)$.

At the beginning of charging, as a result of the high speed of the electrons, electron collection current constitutes most of the total current and creates a negative potential that is continuously reduced, until the summation of the repulsive force to electrons and the attractive force to ions result in the total current being zero. The effect is a negative potential with respect to plasma – that is, absolute surface charging. Surface potential is related to the energy level and density of plasma. Research shows that when immersed in 100 eV and 300 eV plasma, the respective potentials of the satellite surface are −270 V and −830 V. Because of the thermal plasma environment in high orbit, the large number of deposited electrons in polar orbit and the cold plasma environment in non-polar-orbit, the negative surface potential of satellite in high orbit and polar orbit is more severe than that of those in non-polar low-earth orbit.

At the lighted surface, the continuous light irradiation results in departure of electrons from the satellite surface, owing to the photoelectric effect. The lighted surface gradually takes on a positive potential of around several volts to dozens of volts, while the unlighted surface maintains a relatively high negative potential. The potential difference between the two surfaces is defined as relative surface charging, which is the major factor contributing to damage to a satellite when it enters or leaves the earth's shadow.

Internal charging is induced by electrons with energy levels higher than 50 keV residing within poor or isolated conductors after penetration of the spacecraft's skin. Because the flux of high-energy level electrons in high orbit and polar orbit is relatively large, satellites in these orbits experience more severe internal charging problems than in others. In addition, during CME and solar flare periods, the flux of high-energy level electrons surges and lasts a long time. This also leads to serious internal charging problems.

#### 1.2.1.6 Summary of Radiation Environment

The natural space radiation environment is composed of multiple particles with continuous energy levels and flux. These particles contribute to stable factors such as solar winds, the Van Allen radiation belt and GCRs, and eruptive factors such as solar flares and CME. Wilson *et al.* provided a rough illustration of the space environment particles and their energy spectra, as depicted in Figure 1.8 [11].

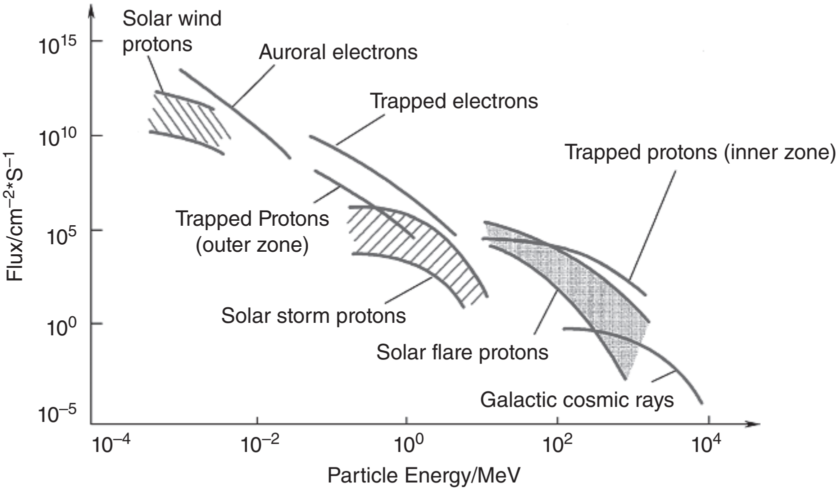


**Figure 1.8** Space radiation environment.

#### 1.2.1.7 Other Space Environments

In addition to space radiation, a spacecraft is subjected to other special space environments and their corresponding negative effects. These include the following:

1) Vacuum environment: when a satellite is in orbit, its electrical equipment is in a very low pressure environment, namely a “vacuum” environment. During launch and the return phase, the electrical equipment is in a gradually changing pressure environment.
2) Thermal environment: the vacuum environment a satellite is in and the huge temperature difference between the satellite’s lighted and unlighted sides invalidate the normal convection thermal control method. This is a new challenge to thermal control. The current practice is to perform thermal control with thermal conduction and radiation.
3) Atomic oxygen, space debris, and so on.

### 1.2.2 Analysis of Damage Caused by the Space Environment

The above space environment will cause permanent, transient, and intermittent electrical equipment and computer failures. The damage caused can be categorized into total ionizing dose (TID), single event effect (SEE), internal/surface charging damage, displacement damage (DD), etc.

#### 1.2.2.1 Total Ionization Dose (TID)

When high-energy level particles penetrate into metal oxide semiconductor (MOS) or bipolar devices and ionize oxides ($SiO_2$), electron and hole pairs that possess composite and drifting types of movements are produced. Without an external electric field, composite movement is dominant; with an external electric field, electrons and holes move in opposite directions along the direction of the field. The high mobility of electrons results in them quickly leaving the oxides, and holes start to accumulate. This process is defined as gate-oxide hole capture. The higher the electric field and electron mobility, the higher the capture ratio. This explains why TID damage is more severe than damage for uncharged components.

An electric field also forms from surface capture at the Si-$SiO_2$ interface. For a negative MOS (NMOS) with an N channel, when $V_g > 0\,V$, surface capture results in negative charge accumulation; for a positive MOS (PMOS), when $V_g < 0\,V$, surface capture results in positive charge accumulation.

The extra electric field produced by gate-oxide and surface capture parasitizes the function area of the components, and leads to drifting of the threshhold voltage $V_{th}$ and transmission delay $T_{pd}$, increase in the static current $I_{cc}$, and attenuation of the transistor's amplification coefficient. Components will fail after these damages exceed a certain limit.

At the outset, the major effect of environmental radiation is gate-oxide capture. However, over time, surface capture becomes more dominant. Therefore, the $V_{th}$ of PMOS is monotone drifting, while the $V_{th}$ of NMOS shows a "rebound phenomenon," which negatively drifts at the beginning, then changes to positive drifting.

Gate-oxide capture anneals at normal temperature (approximately 20–23.5 °C) and achieves accelerated annealing at high temperatures (e.g., 100 °C). Further, it is recoverable damage. Surface capture accumulates charge slowly and steadily. It anneals at normal temperature, but does *not* anneal at high temperatures. Under extreme conditions, high temperatures will even intensify the effect of surface capture and TID, which is difficult to recover from, or is unrecoverable [12].

### 1.2.2.2 Single Event Effect (SEE)

The causal chain of SEE damage is as follows:

Plasma track resulting from high-energy level particle ⇒ charge's movement within the track ⇒ activation of parasitized components or weak components ⇒ all kinds of damage. Based on the effect of SEE, it can be categorized into single event latch-up (SEL), single event upset (SEU), single event burnout (SEB), and so on.

#### *1.2.2.2.1 Single Event Latch-up (SEL)*

SEL is caused externally by current resulting from the potential difference within the "transient plasma needle" before its disappearance. The track of transient plasma is produced by the transient ionization of high-energy level particles entering the Si and $SiO_2$ areas [13].

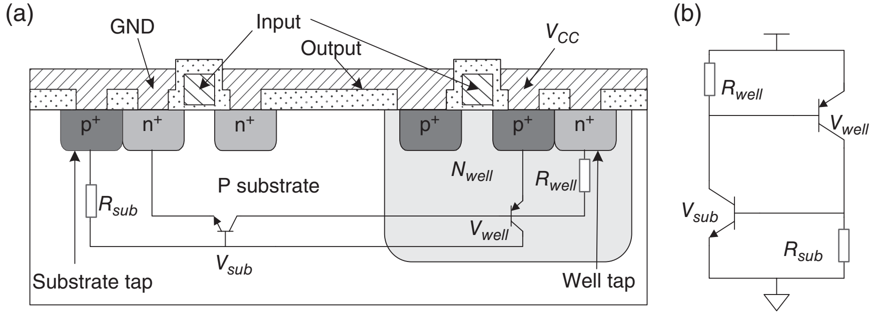


**Figure 1.9** PNPN component: (a) NOT gate. (b) Equivalent circuit.

SEL is caused internally by the parasitic PNPN structure [14]. The PNPN component parasitizes the circuit of the complementary MOS (CMOS), as the NOT gate shown in Figure 1.9(a). The $n^+$ of the NMOS on the P substrate, $p^-$ of the P substrate, and $n^+$ of the N well contact pad form a parallel parasitic NPN transistor, $V_{sub}$. The $p^+$ of the PMOS on the N well, the $n^-$ of the N well, and the $p^+$ of the P substrate contact pad form a vertical parasitic PNP transistor, $V_{well}$. Figure 1.9(b) is the equivalent parasitic PNPN circuit, in which $R_{well}$ and $R_{sub}$ are the parasitic resistances of the well and substrate contacts, respectively.

Under normal conditions, the collector junction of $V_{sub}$ and $V_{well}$ are zero offset, the emitter junction is positively biased, and the PNPN component is in the cut-off state. The impedance between $V_{cc}$ and GND is high. When high-energy level particles enter between the well and substrate, current is produced in this area because of the potential difference. Consequently, the well and substrate are turned on transiently. The result is a voltage drop on $R_{well}$ and $R_{sub}$ which, in turn, makes the emitter junction of $V_{sub}$ positively biased. If the positive bias value is sufficiently large, it turns on $V_{sub}$, and $V_{sub}$ turns on $V_{well}$. Consequently, the PNPN component is in a positive feedback state and $V_{cc}$ and GND are in a low impedance state, with a large current that will fuse metal wire and permanently damage the component if no current limiting measures are taken.

The parasitic PNPN structure is a unique damage mode of a CMOS circuit. There are three requirements for the occurrence of latchup:

1) The loop gain $\beta$ must be greater than one.
2) In order to start the latchup positive feedback, there must be proper excitation to provide necessary bias and starting current.
3) The electrical power supply must provide enough current to maintain latchup positive feedback.

In a CMOS chip, there are many parasitic resistors and bipolar transistors that may become involved in latchup positive feedback. The consequence is a parasitic circuit that is more complex than that shown in Figure 1.9. Latchup current differs as the number of parasitic circuits varies.

Single event snapback (SES) damage may occur in the OFF state of an NMOS transistor. The equivalent circuit is the horizontal parasitic NPN bipolar transistor, and its parasitic resistor on the base junction [15]. When heavy ions hit the source, as a result of the potential difference between the source and the substrate: current flows and a potential difference exists across the parasitic resistor to meet the requirements for amplification of the excitation of the parasitic bipolar junction transistor (BJT); the emitter junction is positively biased; and the bias of the collector junction is the reverse. Furthermore, the potential difference between the source and the substrate provides the necessary BJT turn-on current and turn-on state maintaining current through the parasitic resistor. Long-term BJT turn-on current results in permanent thermal damage to the NMOS transistor.

#### *1.2.2.2.2 Single Event Upset, Turbulence, and Failure Interruption (SEU, SET, Single Event Functional Interrupt (SEFI))*

The rationale for SEU can be described as follows: If the movement of the charge resulting from the potential difference within the vicinity of the plasma track is sufficiently large, the relevant unit's logic state will change (i.e., logic state upset will occur). When only a single digit is upset in a byte, the event is termed SEU, and the upset of multiple digits is termed multiple bits upset (MBU). As illustrated in the SRAM shown in Figure 1.10, when heavy ions turn on the NMOS transistor in the lower left corner through bombardment, the

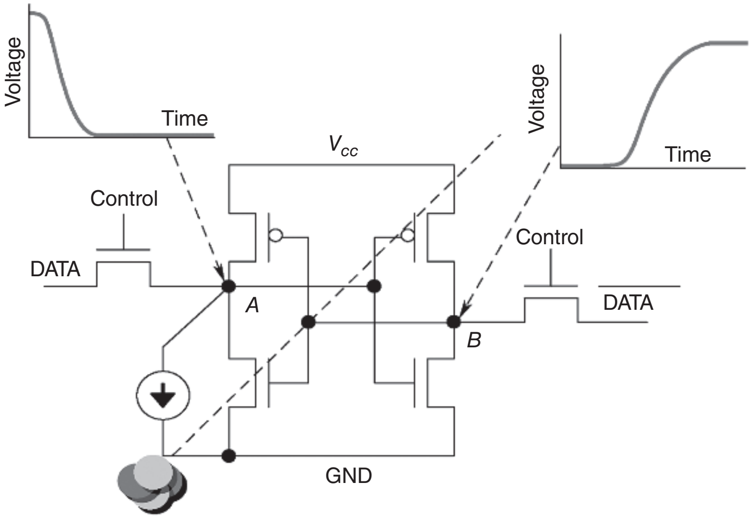


**Figure 1.10** Physical essence of SEU damage.

logic state of point A will change from one to zero because it is grounded. This, in turn, will lead to a change in point B's logic state from zero to one. In addition to the SRAM, time sequential components, such as flip-flops and latches, will also experience SEU failure. In fact, because of the complexity of the component's structure, the actual fault mechanism is more complex than that depicted in Figure 1.10.

When SEU occurs in a control or configuration unit (e.g., the working mode selector of a network protocol chip, address register of a CPU, or configuration unit of an FPGA), component failure can result (i.e., SEFI) [16].

When heavy ions bombard the sensors of analog devices, the analog signal will experience transient turbulence, which can lead to incorrect circuit action if the turbulence amplitude exceeds a certain level, namely, a single event transient effect (SET). It is obvious that SET and SEU share the same intrinsic physical mechanism [17].

#### *1.2.2.2.3 Damage to Power Component*

The common technique used to increase the current capacity of the power type metal oxide semiconductor field-effect transistor (MOSFET) is to connect thousands of MOS transistors or strip shaped transistors in parallel, with a large contact area between the source and the drain. Figure 1.11 shows the topology of a MOS transistor. There is a vertical parasitic NPN BJT between

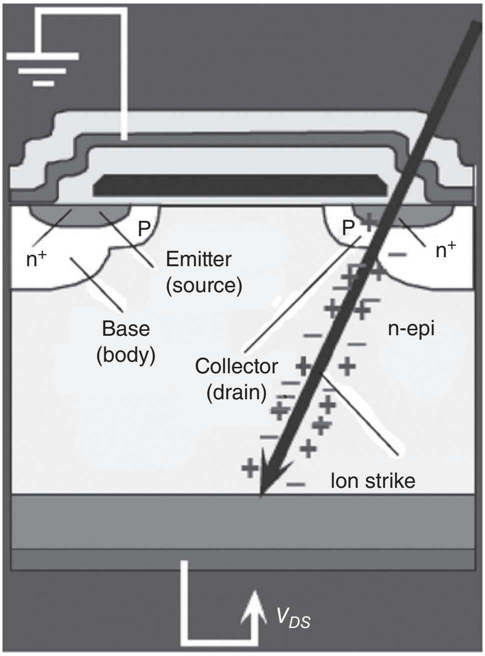


**Figure 1.11** MOSFET parasitic BJT structure that leads to SEB.

the source and the substrate below it. The emitter is an $n^+$ source, the base is a $p$ well, and the collector is an extended $n$ substrate.

When a MOSFET is turned off, the source has a high voltage. The penetration of high-energy level particles through this high voltage source zone (emitter of the parasitic BJT) results in the activation of a vertical parasitic BJT. The high voltage of the MOSFET produces a high current between the parasitic BJT's emitter and collector that breaks down and melts the high voltage $n^+$ – that is, SEB occurs [18]. SEB will NOT occur when the power MOSFET is turned on and there is a discharge path between the source and the drain. The structure of the power BJT is the same as that of the parasitic BJT; consequently, the risk of SEB in power type high voltage BJTs is high [19]. The damage mechanism shows that high voltage is a necessary condition for SEB. Power MOSFETs, BJTs, and logic MOS transistors operating at low voltages will *not* experience SEB failure. This is an effective measure to protect against SEB.

In addition, when a power MOS transistor is turned off, a single event gate rupture (SEGR) may occur [20]. As shown in Figure 1.12(a), bombardment by heavy ions to the lower part of the gate will produce high density plasma in the vicinity of the ion's track within the substrate. The electric field results in the electron-hole pair drifting in the opposite direction to accumulate charge, as shown in Figure 1.12(b). The gate-oxide functions as a capacitor with a large amount of charge. When the voltage between the two ends of the capacitor is sufficiently high, it will break down the gate-oxide structure and result in unrecoverable damage.

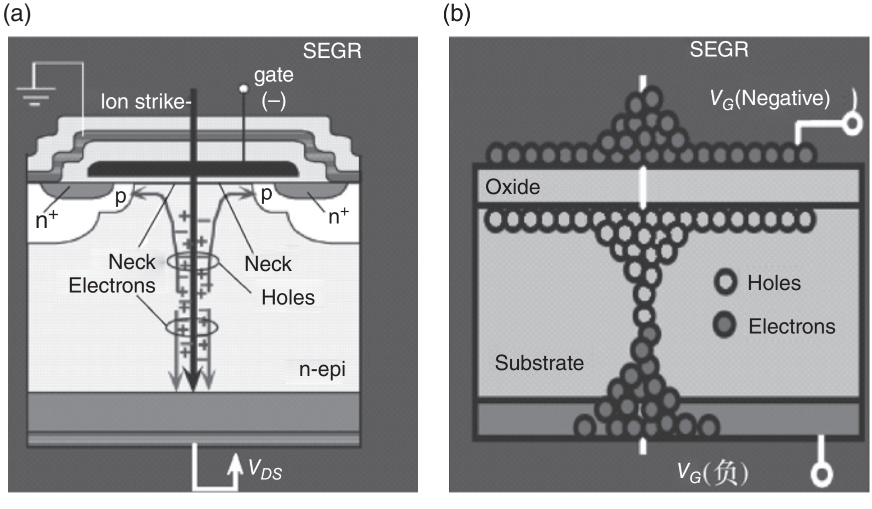


**Figure 1.12** Physical mechanism of SEGR damage.

##### *1.2.2.2.4 Micro-dose Effects*

With advances in the manufacturing process, the feature size of a single transistor is comparable to the ionization track of a single high-energy level heavy particle. The dose deposited by a single high-energy level particle's impact on an IC is large enough to cause a single MOS transistor to fail in a fixed state, which results in it *not* responding to offset changes (i.e., stuck bit failure) [21].

#### 1.2.2.3 Internal/surface Charging Damage Effect

The casual chain of charged damage is as follows: radiation environment ⇒ internal/surface charged ⇒ ESD ⇒ all kinds of damage. Absolute surface charging does *not* cause damage, whereas relative surface charging produces ESD to form intense current and electromagnetic interference (EMI). This can result in analog components producing false signals, digital signal state upsets, and eventually latchup in CMOS component.

#### 1.2.2.4 Displacement Damage Effect

When incoming high-energy level particles hit electrical components and materials, their arrival at the nucleus of the crystal lattice will result in Kulun elastic collisions. The nucleus of the lattice acquires mechanical energy during the collision, and leaves its normal position to form lattice defects in the component's functional area. If the energy of the incoming particle is sufficiently large, it will displace the atoms within the lattice to form a defect cluster. Lattice defect degrades component feature specifications (transistor amplification coefficient, conversion efficiency of solar cells, CCDs, and other photoelectric devices), and results in defects accumulating until the component loses its functionality. Therefore, displacement damage is the result of accumulative effect [22].

The manifestation of displacement damage is similar to that of TID damage, but the nature of the two differ:

1) Different damage mechanisms; TID leads to component damage through loss of energy induced by ionization, whereas displacement damage leads to component damage through loss of energy induced by the transfer of mechanical energy.
2) The target of displacement damage includes bipolar components, solar cells, and photoelectric devices (e.g. CCDs), whereas the target of TID damage is more extensive.
3) TID is relevant to the charging state of the component, whereas displacement damage is irrelevant to the charging state of the component.

#### 1.2.2.5 Other Damage Effect

Vacuum and vacuum discharge: Vacuum discharge occurs in the $10^3$–$10^{-1}$ Pa low vacuum range. Low vacuum discharge mainly affects electrical instruments operating in the ascending or return phase, such as the guidance, navigation, and control (GNC) computer. When the vacuum level reaches $1 \times 10^{-2}$ Pa or higher, it produces micro-discharge, corona discharge, which produces a damage effect that is significant to a satellite's power supply system.

A space computer system will have to contend with the space environment described above. Thus, in order to mitigate the effects of the space environment effect, improve system reliability, and ensure mission success, the implementation of fault tolerance techniques in the spacecraft control computer is essential.

## 1.3 Development Status and Prospects of Fault Tolerance Techniques

Advancements in computer technology have resulted in expansions in a variety of fields in which it is applied. Computer technology is used extensively in fields such as aeronautics, astronautics, defense, finance, and industrial control. People are relying ever more increasingly on computers (e.g., in space applications). Owing to the catastrophic losses that would result from computer failure and the irreparable nature of most systems, computer systems are now expected to possess the ability to perform major functions with high reliability when component failure occurs or, at least, not to result in serious consequences. This demand underlies the emergence and advancement of computer fault tolerance techniques.

Computer fault tolerance techniques began with Von Neumann's proposal that high reliability could be achieved through the use of redundant low reliability components. First generation computers (1946–1957) were composed of electron tubes, relays, and delay line memory, whose high failure ratio and susceptibility to transient faults resulted in limited MTTF (several minutes to 2–3 hours). It was necessary to adopt fault detection and recovery techniques; for example, at that time, IBM 650 and UNIVAC all implemented parity checks to improve reliability.

Implementation of fault tolerance techniques in the space industry began in the 1960s [23–25]. In the late 1960s, the Jet Propulsion Laboratory (JPL) in America put the fault-tolerant computer STAR, designed for the Apollo program, into operation, and employed a fault detection module to identify faults. Transient fault was masked through result comparison and rollback. Tolerance to permanent fault was achieved through replacement of permanently failed units. STAR proved to be a milestone in the development of fault-tolerant computer techniques [26].

Another typical fault-tolerant computer was placed in the JPL "Voyager" deep space explorer. This adopted a dual hot backup dynamic fault tolerance strategy through mutual monitoring with information exchange between the two computers. If one failed, the other one would take over automatically or be remotely controlled. “Voyager” operated in space for 35 years. The dynamic redundancy fault-tolerant computer architecture was extensively used in subsequent models.

A fault-tolerant computer was also employed on space shuttles in the 1980s [27]. This computer consisted of four homogeneous machines and a heterogeneous machine. Each homogeneous machine consisted of a CPU for mission calculation and an IOP for input and output. During critical mission phases, all four homogeneous machines operated simultaneously, to mask any isolated fault that may occur in one machine by employing the voting mechanism. Only the astronaut could downgrade the quadri-modular redundancy to triple modular redundancy (TMR). Furthermore, if a determinate fault occurred in the TMR, the astronaut could downgrade from TMR to dual hot backup. Finally, if there was a fault in the dual hot backup, it could be identified if it was a common software bug, and the astronaut could switch to the heterogeneous machine. The fault-tolerant computer successfully fulfilled its flight mission during the 20 years of space shuttle operation time.

In China, research on space fault tolerance techniques began in the 1970s. The fault tolerance architectures used for the control computer can be categorized into fault masking type, fault detection type, and system reconfiguration type [28,29]. In a spacecraft computer system, the most commonly used is the fault masking type, which is defined as static redundancy. It applies redundant resources (e.g., hardware, software, time, and information) to mask the effect of a fault, so that system operation is ensured.

In the 1980s, the Beijing Institute of Control Engineering (BICE) applied dual machine fault tolerance architecture to the onboard computer in a spacecraft control system for the first time. Satellite programming and attitude control was achieved using an Intel 3000I2L CPU and small-scale CMOS devices. The satellite was launched in 1987, and China’s first onboard computer successfully fulfilled its satellite control mission. Since then, the computer system applied in the control system has varied in accordance with the mission of the spacecraft.

The recoverable spacecraft control computer, designed in the mid-1980s, with maiden flight launch in 1992, is a typical fault masking, detection, and system reconfiguration fault tolerance model. The computer’s adoption of a triple redundancy/single redundancy (TMR/S) structure to perform real-time fault detection and disposition significantly improved system reliability. Its implementation of a feedback test method improved the self-diagnostic capability of the system. The structure was a special case of N-modular redundancy, which applied three identical machines running identical software. The three outputs resulting from the same input was transmitted to a voting machine, whose output was based on the mechanism of majority voting. If one of the machines failed, the correct output of the other two could be used to mask the failure. If a machine failed permanently, it would be cut off to avoid additional machine failure induced false vote or incapability to vote. A post-cut-off system could be reconfigured as a single-machine system; hence, it was defined as a triple/single module fault-tolerant system.

The advantage of the system lay in the fact that its excellent masking effect could eliminate the effect of transient fault on the system and maintain continuous control when a single machine failed permanently, plus its strong self-diagnosis and self-switching ability. The TMR/S structure was applicable to both short period control and real-time control. In addition, this structure was also applied in the GNC system of the Shenzhou manned spacecraft to perform guidance, navigation, and control functions; it fulfilled the mission.

As the demand for spacecraft and satellites with a lifetime greater than two years increased after the 1980s, onboard control computers began to adopt a cold backup, cold-hot backup module reconfiguration type of fault tolerance structure, which integrates fault detection and diagnosis, static redundancy design, system reconfiguration, and operation recovery techniques. This structure is also defined as a dynamic redundancy structure, with major features such as fault detection, fault location, failed module cut off, redundant resource start up, and system operation maintenance. The core technology is control process continuity maintenance during the computer system reconfiguration period. This is necessary to apply measures to the overall system to minimize the effect of "interruption" to the system.

A typical fault tolerance structure includes multiple machine cold backups, module-level cold backups, and cold-hot backups. These technologies have been applied in resource satellites, communication satellites, remote sensing satellites, and navigation satellites; for example, a resource satellite's control computer has multiple cold backups, plus emergency modules to solve system lifespan and reliability problems. The satellite itself also has a reconfigurable module, hardware monitored and switched control computer, which consists of two identical machines – one in operation and the other standing by without being powered on. If the CPU board, I/O board, or memory board of the operating computer fails, or part of those fails, the microcontroller-centered hardware would replace the failed one by starting another cold-backup identical module to ensure system operation. A single machine in cold-backup state could improve overall system reliability and simultaneously reduce the TID effect on computers.

Another example is a communication satellite's integration of static and dynamic redundancy, dual comparison backup, and multiple working modes, including full single machine, reconfigurable single machine, and single machine cold backup. Dual hot backup mode is applicable to SEE, because the comparison between the two outputs can eliminate transient disturbance effects. Cold backup can minimize the effect of TID on computers; hence, it is applicable to long-term operation requirements. Module-level reconfiguration can improve the reliability of the overall system. With the development of IC technology, multiple machine cold backup, combined with the emergency module method, is used extensively in spacecraft control computers, such as computers on resource, communication, remote sensing, and navigation satellites.

Throughout the development of fault tolerance techniques over the past several decades, spacecraft fault-tolerant computers have succeeded prominently in space applications. The development of fault-tolerant computers is greatly promoted by the leading of space applications and driving of techniques' improvements. As higher requirements resulted from the growth of China's space industry, the research on fault-tolerant control computers has become more focused on the following areas [30]:

1) Highly dependable spacecraft fault-tolerant computer architecture. With higher requirements for computer systems in space missions, newer and higher fault-tolerant computer and architecture techniques should be studied.
   - Because performance improvements require a parallel structure, it is necessary to study how to utilize resources in a parallel structure to meet the performance requirements while attaining reliable fault-tolerance, especially in the aspects of task scheduling and reconfiguration techniques.
   - The increased complexity of system functionality requires a distributed system architecture, a centralized system architecture, or a combination of these two, which makes it necessary to study new fault tolerance interconnection techniques, including wireless connection.
   - The extension of adaptability to failure mode resulting from dependability improvements makes it necessary to study a computer fault tolerance structure that can tolerate malicious faults (Byzantine failure mode).
2) Fault tolerance technique for system on chip (SoC). SoC (including SoC implemented with SRAM-based FPGA) is now being used extensively in spacecraft control computers, resulting from the spacecraft's requirement for more powerful functions, low power consumption, small size, and light weight. In these applications, the new challenges from space environment make it necessary to study the following:
   - Failure mode, fault detection, fault identification, recovery, and reconfiguration technology of SoC.
   - Radiation hardening techniques.
   - A reliable system design technique based on intellectual property (IP).
3) Intelligence fault tolerance technique. Further development of space exploration requires that the onboard computer possesses self-failure processing and recovery abilities. As a result, it is necessary to study intelligent techniques:
   - Hardware evolutionary techniques that are able to solve problems in practical applications.
   - Realization of artificial immunity techniques in fault-tolerant computers.

4) Construction and verification techniques for highly dependable software. Because the guarantee of high dependable software is an important aspect of spacecraft fault tolerance techniques, it is necessary to study the following:
   - How to improve the reliability of current software fault tolerance techniques in applications.
   - Software dependability assurance systems.
   - Space software formal verification methods.
   - Integrated environments for dependable software verification.
5) Fault tolerance verification technique. Aiming at new fault models and new fault tolerance structures, new fault tolerance verification techniques are necessary, including the following:
   - Simulation techniques for fault-tolerant computer verification environments.
   - New fault injection techniques (including no-probes techniques and no-stubs techniques).
   - Automatic test-case generation techniques and fault tolerance performance estimation techniques.

## References

1 Von Neumann J (1956). Probalilistic logics and the synthesis of reliable organisms from unreliable components. In: Shannon C E and McCarthy J (eds). *Automata Studies*. Princeton: Princeton University Press.

2 Avizienis A (1997). Toward systematic design of fault-tolerant systems [J]. *IEEE Computer Magazine* **41**(4), 51–58.

3 White R V, Miles F M (1996). *Principles of fault-tolerance* [C]. IEEE proceedings of the eleventh annual applied power electronics conference and exposition, March 3–7 1996, 1, 18–25.

4 Hu Mou. (1995). *Computer fault-tolerance technology* [M]. Beijing: China Railway Publishing House.

5 Yuan Youguang, Chen Yinong. (1992). *Fault-tolerance and fault avoidance techniques and their applications* [M]. Beijing: Science Press.

6 Yang Shiyuan. (1989). *Fault diagnosis and reliability design of digital system* [M]. Beijing: Tsinghua University Press.

7 Zeng Shengkui, Zhao Yandi, Zhang Jianguo, Kang Rui, Shi Junyou. (2001). *System reliability design and analysis* [M]. Beijing: Beihang University Press.

8 Leach K R Bedingfield, Alexander M. (1996). *Spacecraft system failures and anomalies attributed to the natural space environment* [R]. NASA-RP-1390, NASA.

9 Russell C. (2000). The solar wind interaction with the earth's magnetosphere: a Tutorial [J]. *IEEE Transactions on Plasma Sciences* **28**(6), 1818–1830.

10 Mullen E, Ginet G, Gussenhoven M, *et al.* (1998). SEE relative probability maps for space operations [J]. *IEEE Transactions on Nuclear Science* **45**(6), 2954–2963.

11 Wilson J, Townsend L, Schimmerling W, *et al.* (1991). *Transport methods and interactions for space radiations.* Reference Publication-1257, Dec. 1991.

12 Shaneyfelt M, Schwank J, Fleetwood D, *et al.* (2004). Annealing behavior of linear bipolar devices with enhanced low-dose-rate sensitivity [J]. *IEEE Transactions on Nuclear Science* **51**(6), 3172–3177.

13 Dussault H, Howard J, Block R, *et al.* (1995). High-energy heavy-ion-induced charge transport across multiple junctions [J]. *IEEE Transactions on Nuclear Science* **42**(6), 1780–1788.

14 Troutman R R. (1986). *Latch-up in CMOS technology* [M]. Boston: Kluwer Academic Publishers.

15 Koga R, Kolasinski W. (1989). Heavy-Ion-Induced Snapback in CMOS Devices [J]. *IEEE Transactions on Nuclear Science* **36**(6), 2367–2374.

16 Koga R, Penzin S, Crawford K, *et al.* (1997). *Single event functional interrupt (SEFI) sensitivity in microcircuits* [A]. In: Proc. 4th Radiation and Effects Components and Systems (RADECS), Cannes, France, Sep. 1997, pp. 311–318.

17 Adell P, Schrimpf R, Barnaby H, *et al.* (2000). Analysis of single-event transients in analog circuits [J]. *IEEE Transactions on Nuclear Science* **47**(6), 2616–2623.

18 Adolphsen J, Barth J, Gee G. (1996). First observation of proton induced power MOSFET burnout in space: the CRUX experiment on APEX [J]. *IEEE Transactions on Nuclear Science* **43**(4), 2921–2926.

19 Titus J, Johnson G, Schrimpf R, *et al.* (1991). Single event burnout of power bipolar junction transistors [J]. *IEEE Transactions on Nuclear Science* **38**(6), 1315–1322.

20 Allenspach M, Brews J, Mouret I, *et al.* (1994). Evaluation of SEGR threshold in power MOSFETs [J]. *IEEE Transactions on Nuclear Science* **41**(6), 2160–2166.

21 Oldham T, Bennett K, Beaucour J, *et al.* (1993). Total dose failures in advanced electronics from single Ions [J]. *IEEE Transactions on Nuclear Science* **40**(6), 1820–1830.

22 Srour J, Marshall C, Marshall P. (2003). Review of displacement damage effects in silicon devices [J]. *IEEE Transactions on Nuclear Science* **50**(3), 653–670.

23 Wensley J H, *et al.* (1978). SIFT: Design and analysis of a fault tolerant computer for aircraft control [C]. *Proceedings of the IEEE* **66**, 1240–1254.

24 Jenkins D R (1996). *Space Shuttle, the history of developing the national space transportation system* [M]. Walsworth Publishing Company.

25 Urban G *et al.* (1998). A survivable avionics system for space application. *FTCS-28* 372–381.

**26** Avizienis A, Gilley G C, Mathur F P, Rennels D A, Rohr J A, Rubin D K. (1971). The STAR (self-testing and repairing) computer: an investigation of the theory and practice of fault-tolerant computer design. *IEEE Transactions on Computers* **20**(11), 1312–1321.

**27** Hanaway J F, Moorehead R W (1989). *Space shuttle avionics system*. NASA.

**28** Yang Mengfei, Guo Shuling, Sun Zengqi. (2005). On-board computer technology for spacecraft control applications. *Aerospace Control* **23**(2), 69–73.

**29** Feng Yanjun, Hua Gengxin, Liu Shufen. (2007). Review of research on anti-radiation of aerospace electronics [J]. *Journal of Astronauticas* **28**(5),1071–1080.

**30** Yang Mengfei, Hua Gengxin. (2005). *Current status and future development trends of aerospace computer technology* [C]. The First Annual Conference of Chinese Society of Astronautics.