


<b>Technique</b>	Apply techniques such as Built in Test (BIT), strategic placing of sensors, centralized architecture, and fault isolation and recovery to optimize system availability.
 <h2 style="margin: 0;">Fault-Detection, Fault-Isolation and Recovery (FDIR) Techniques</h2> <p style="margin: 0;"><i>Utilize FDIR Design Techniques to provide for Safe and Maintainable On-Orbit Systems</i></p>	
<b>Benefits</b>	The main goal of fault detection and isolation is to effectively detect faults and accurately isolate them to a failed component in the shortest time possible. This capability leads to reduction in diagnostic time or downtime in general and, therefore, increased system availability. A good inherent diagnostic of a system also enhances the crewmembers' confidence in operating the system, the main driver of mission success. Effective FDIR can keep a difficult to maintain system up and running where normal methods would lead to system downtime. FDIR is especially beneficial to an on-orbit system where maintenance may be impossible.
<b>Key Words</b>	Fault Detection, Fault Isolation, Recovery, FDIR
<b>Application Experience</b>	International Space Station Program
<b>Technical Rationale</b>	Operating in such a critical environment as outer space, astronauts' lives and mission success are dependent on the integrity of a system. Since time and resources are limited, the sooner failures can be accurately detected and a failed system repaired and recovered, the more likely crew survival rate and mission success are to be improved.
<b>Contact Center</b>	<b>Johnson Space Center (JSC)</b>

## ***Fault-Detection, Fault-Isolation, and Recovery (FDIR) Techniques Technique DFE-7***

The growth of electronic technology challenges the use of electronic systems in several respects. One of these is the complexity of testing the systems to determine functional status and to permit efficient fault detection and fault isolation. The term “diagnostic capabilities” refers to the abilities of a system to detect a failure and to isolate it to a failed maintainable unit. In the past, diagnostics were considered only as a design afterthought and, as a result, many programs are faced with higher mean time to repair (MTTR) and higher work-hour and false alarm rates. This reduces system availability and operational readiness while increasing life cycle costs. Diagnostics are a significant key to achieving system performance and cost effectiveness goals.

In such a critical system as the International Space Station, on which human life is dependent, a system recovery concept is also an important aspect that needs to be considered early in the system’s design phase. This technique consists of sections on fault-detection, fault-isolation, and recovery techniques. Since they are all related under the integrated diagnostics concept, techniques of one section may be referenced in other sections.

### ***Fault-Detection Techniques***

A system fault can be detected manually or automatically, depending on operating modes and how quickly the system needs to be restored. For a system that requires human interfaces, system failures can be detected quickly by human visual and/or auditory

senses. If, for example, a light is switched on and there is no illumination, one can visually detect that there is a problem with either the light switch, light bulb, power source, or circuitry. The obvious advantages of manual fault detection are that it incurs no costs associated with complex system designs.

Another common methodology, built-in testing (BIT), is employed to detect and isolate faults without using external test equipment. BIT ranges in complexity from a lamp that lights when equipment fails, to a resident computer that generates test signals and evaluates system responses. BIT can be continuously operated, interleaved with other operations, or initiated on command. During power-on self-testing, for example, the system runs a self-diagnostic test after the power is applied and includes hardware sensors and software error-correcting codes. Its particular mechanization and utilization in a system are, of course, determined by the designer.

BIT often means additional hardware above that required for the primary function. Reliability and cost are affected and trade-offs leading to a balanced solution must be made. BIT protective circuitry, moreover, should be designed to be fail-safe. This means that failure in the BIT circuitry should not affect system performance. Whenever feasible, the BIT input and output should be sufficiently isolated from the normal channels so that any failure in the BIT will not cause impairment of the function being tested. Also, it should be recognized that BIT can fail, and additional measures should be taken to avoid utilization of possibly erroneous BIT output in recovery measures.

In addition to BIT circuitry which actuates visual status indicators, BIT features may also

include test points and self-test meters. The goal of BIT design is to decrease MTTR by steering a technician to the faulty component as quickly as possible. BIT designers attempt to attain this goal through various means, including the use of innovative circuitry and rearrangement of circuits to perform dual functions with a single circuit, if possible; e.g., driving a visual indicator and tying into various AND gates with a single driver.

The BIT designer also standardizes BIT circuitry as much as possible, thus driving down the cost of implementing BIT.

Other important general considerations in designing hardware BIT are:

a. The reliability of the BIT hardware should exceed that of the hardware being tested. If this is not the case, the probability of failure of the BIT may be almost as great as the probability of failure of the unit being tested.

b. The BIT should be kept simple but effective in meeting operational needs.

c. The type of circuitry used for BIT should be, if feasible, of the same type used in the normal system to minimize the number of different types of components used in any particular system.

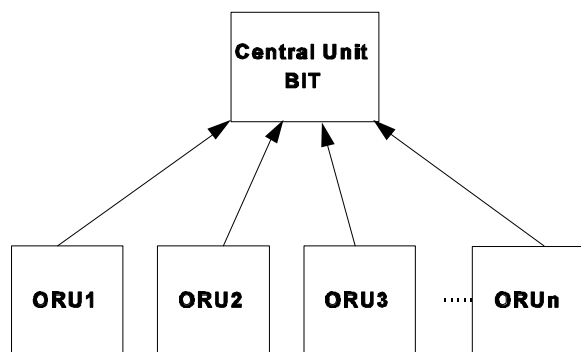
As a part of the BIT design process, the overall system architecture must also be considered for the most effective implementation. Generally, there are two common approaches: centralization and decentralization.

Centralization is regarded as a highly integrated approach in which a centralized unit acts as a "watch dog" in detecting and

reporting system out-of-tolerance conditions. The centralized unit determines if a failure actually occurred based on the data and information queried from the lower level, and annunciates or reports faults (see figure 1).

The type of information acquired by the central unit is an example of passive BIT. Passive BIT monitors system performance on line without the use of a test pattern generator; therefore, it may not be able to completely monitor the system.

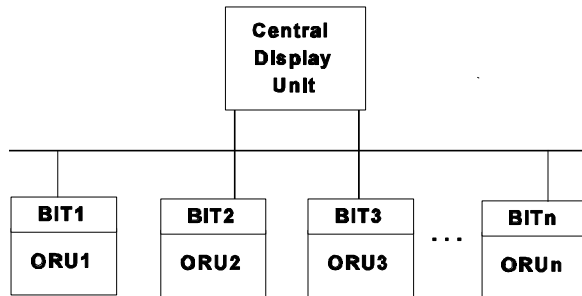
Active BIT, a more comprehensive type of testing, can also be used. In active BIT, a test pattern is written to a unit and compared to an expected pattern. The system operation must be interrupted for this type of test if the module is operating. Not all modules are operated continuously, however, and a computer-controlled BIT system can take advantage of times when a module is not needed to run a test sequence. This is referred to as interleaving BIT, which can be a powerful means for maintaining confidence in a system without disrupting its mission to run tests.



**Figure 1. Centralized Architecture**

On the other hand, decentralized architecture places detection capabilities at the maintainable

unit level. Each maintainable unit has the means to detect all the identified failures within the unit. Once a failure occurs, the unit reports to a higher level for record and fault annunciation (see figure 2). Both passive and active BIT can be used in this case.



**Figure 2. Decentralization**

With decentralized BIT, each module will have its own circuitry to monitor such out-of-tolerance conditions as voltage, current, or parity word at a particular node or memory location. Failure condition of a module can be notified to a central display unit via a data distribution network such as a 1553 bus network. One advantage of decentralized architecture is that a subsystem or a module can be taken off line for a comprehensive test using active BIT (generated test pattern). In general, the decentralized control must have the following characteristics:

- a. Self-test capability.
- b. Isolation from other system data.
- c. Some type of synchronization with system functions.

A voting scheme is another effective technique to detect failure. With this technique, on-line

data are processed by three or more redundant computers. A failure is declared when one unit's output is different from the other two or three units. Decentralized architecture is commonly used in this application. The inertial measurement system on the Orbiter exemplifies this type of architecture, in that three redundant inertial measurement units process real-time data and compare the outputs for majority agreement. This method requires additional resources but is highly effective in fault detection.

### ***Fault-Isolation Technique***

Once a failure is detected, the next step is to locate the cause of that failure. The complexity of a system, quick-turnaround time demand, repair location, and human skill level are some important factors which must be considered in planning the fault isolation strategy of a system.

Faults can be isolated manually by visually inspecting for burned-out components or by using an external test set for system diagnostics. For a more complex system and time-critical mission, the faults can also be isolated automatically. BIT, as mentioned in the fault-detection section, contains an inherent fault-isolation mechanism, since signals generated by a failed module can be identified by the control unit, thereby allowing a test pattern to be injected to that particular unit to confirm its failure.

Sometimes BIT can isolate failures down to a certain system level or to a region of a system that has components connected in a series. In this situation, it is rather difficult to determine exactly which component has actually failed. In contrast, voting scheme will accurately pinpoint the failed unit, because the redundant connections of the system dictate the ease of

fault isolation. However, in situations where the remaining two units indicate a fault has occurred, a failed unit will not be easily identified because there is no "majority" reference data for comparison.

A more recent approach to isolating a failure in the integrated circuit (IC) industry is called the boundary scan. The IC is divided into regions which are accessible via scan operations. A boundary scan path consists of a series of boundary scan cells (BSC's), one BSC per IC function pin. The BSC's are interconnected with the host IC's test data input pin and test data output pin, for serial access.

During normal IC operation, input and output signals pass through each BSC without interference. When the boundary test mode is selected, however, the test stimulus is applied through a series of BSC's, and test results are captured at the end of the scan path. This technique overcomes the test access problems that can cause difficulty in fault isolation. The unit level tests can also be combined for a system-level verification.

### ***Recovery Technique***

In order for a system to recover manually or automatically from a failure, modes of operation which depend on types of failures must be defined and planned ahead. During the design phase, the system's critical functions, levels of redundancy, and functional paths are usually identified so its recovery functions can be realized. In general, there are three categories of recovery: (a) 100 percent functional recovery using redundant system components, (b) functional recovery using an alternative path, and (c) degraded functional recovery.

For category (a), the system is designed so that when a component fails, its failure is reported and the component's redundant or backup unit can be turned on manually or automatically. In such systems as satellites, interplanetary probes, and the Space Station, which require autonomous operations, automatic recovery is likely to be provided.

Resources are almost always limited in any situation; therefore, instead of having a redundant string or unit, an alternative path, category (b), may be taken to recover the lost function. The alternative path usually does not achieve the full capabilities of the original function because of limiting space and design constraints, including costs.

For reasons stated above, many redundant critical functions of the Space Station have been designed using the method of functional recovery. If, for example, the cooling loop of the thermal control system failed and was unable to cool the electronics equipment mounted on the cold plates, cool air from the environmental control and life support system may be redirected in order to keep the equipment from overheating. The equipment may also be required to operate at a minimal level to lessen the heat generated.

As a worst case, redundant strings are out or not available; in which case, operating a system at some minimal capacity must be considered to protect crews or vehicles. In these cases, the critical functions of a system must be looked at to decide which of its components or units may be turned off without losing the ability to control the spacecraft until repairs are made.

If, for example, some of the solar array panels were damaged, insufficient electrical power would be generated to support all the Space

Station functions. In this case, power allotted to less critical functions would have to be curtailed or eliminated and even critical systems may have to be operated at some compromised level.

### ***Summary***

FDIR is becoming an increasingly important factor in designing today's complicated systems and in today's competitive edges for operating an efficient plant or space system with minimal downtime. In any business, downtime or delays may cost millions of dollars a year in addition to operating costs, simply because FDIR was a design afterthought

By implementing FDIR's design features, one can be assured that the final product will be a safe, efficient, and maintainable system.

### ***References***

1. Air Force Design Handbook 1-9, "Maintainability (for Ground Electronic Systems)," Second Edition, Revision 7, February 25, 1988, United States Air Force Aeronautical Systems Division.
2. Architecture Design Document (ADD) D684-10504, "Failure Detection, Isolation, and Recovery (FDIR)," February 22, 1994, International Space Station Alpha Program, National Aeronautics and Space Administration.
3. Anthony Coppola, "A Design Guide for Built-In Test (BIT)," April 1979, Rome Air Development Center, Air Force Systems Command.