Analysis of a Multi-Layer Fault-Tolerant COTS Architecture for Deep Space Missions

Savio N. Chau
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Savio.chau@jpl.nasa.gov

Leon Alkalai
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
Leon.alkalai@jpl.nasa.gov

Ann T. Tai
IA Tech, Inc.
10501 Kinnard Avenue
Los Angeles, CA 90024
A.t.tai@ieee.org

Abstract

Fault-tolerant systems are traditionally divided into fault containment regions and custom logic is added to ensure the effects of a fault within a containment region would not propagate to the other regions. This technique may not be applicable in a commercial-off-the-shelf (COTS) based system. While COTS technology is attractive due to its low cost, they are not developed with the same level of rigorous fault tolerance in mind. Furthermore, COTS suppliers usually have no interest to add any overhead or sacrifice performance to implement fault-tolerance for a narrow market of high reliability applications. To overcome this shortcoming, Jet Propulsion Laboratory (JPL) has developed a multi-layer fault protection methodology to achieve high reliability in COTS-based avionics systems. This methodology has been applied to the bus architecture that uses the COTS bus interface standards IEEE 1394 and I^2C. This paper first gives an overview of the multi-layer fault-protection design methodology for COTS-based mission-critical systems. Then the effectiveness of the methodology is analyzed in terms of coverage and cost. The results are compared to the traditional custom designed system.

1. Introduction

In recent years, commercial-off-the-shelf (COTS) products have found many applications in space exploration. The attractiveness of COTS is that low cost hardware and software products are widely available in the commercial market. By using COTS through out the system, the development and recurring costs of the system can be significantly reduced. As an example, the Cassini mission at JPL developed a set of ASICs that has a total of about 100 kgates. The design of the ASICs was totally in-house and it took 7.5 workyears to develop. On the contrary, JPL's X2000 program has developed two ASICs with a total gate count of 700 kgates, of which 400 kgates were logic circuits and 300 kgates were memory cells. Most parts of the ASICs were designed with COTS intellectual properties (IPs) and it took only 4 workyears, or 1 workyear per 100 kgates. Furthermore, the X2000 ASICs dedicate a much less percent of circuitry for custom-designed fault tolerance than the Cassini ASICs. Hence the cost benefit of using COTS technology is obvious.

On the other hand, in most cases, COTS are not suitable for highly reliable applications such as long-life deep-space missions. There are two reasons. First, the suppliers of COTS products have no interest to change their design, add any overhead, or sacrifice their performance for a narrow market of high reliability applications. Second, any modification will render the COTS incompatible with commercial test equipment or software, and therefore diminish the economic benefits of COTS drastically. Therefore, the challenge is how to deliver a low-cost, highly reliable and long-term survivable system based on COTS technologies that are not developed with high-reliability in mind.

2. A Multi-Level Fault Protection Methodology for COTS-Based Systems

To compensate for COTS technologies' weakness in fault tolerance, the X2000 has employed a multi-level fault protection methodology to achieve high reliability [1][2]. The methodology is applied to the X2000 bus architecture by using four levels of fault protection mechanisms. The mechanisms are depicted in Figure 1 and the resulting bus architecture is shown in Figure 2. The overhead for implementing the methodology is included in the 400 kgate logic mentioned above. These four levels of fault protection mechanisms are described as follows.

Level 1: Native Fault Protection – most of COTS bus standards have some limited fault detection capabilities. These capabilities should be exploited as the first line of defense.

Level 2: Enhanced Fault Protection – addition layer of hardware or software can be used to enhance the fault

detection, isolation, and recovery capabilities of the native fault containment region. This layer contains a small amount of custom logic. Examples are watchdog timer or additional layer of error checking in the protocol. These fault tolerance mechanisms are designed in such a way that they do not affect the basic COTS functions.

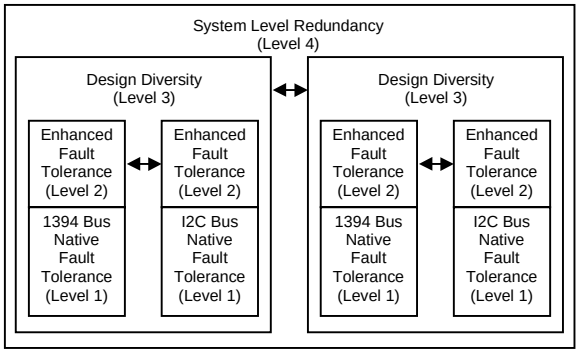

Figure 1: X2000 Multi-Level Fault Protection Methodology

Level 3: Fault Protection by Component Level Design Diversity – many COTS have fundamental fault tolerance weakness that cannot simply be removed by enhancing the native fault protection mechanisms. These weakness usually are related to single points of failures. One example is the tree topology of the IEEE 1394 bus [3][4][7]. When the bus is partitioned by a failed node, no watchdog timer or extra layer of protocol can reconnect the bus. In order to compensate for such fundamental weaknesses, different types of buses may be used to complement the IEEE 1394 bus. Specifically, the I^2C bus, which has a multi-drop bus topology [5][6], is used to assist the IEEE 1394 fault isolation and recovery in X2000 [1][2]. The coordination between these two bus would require some custom software.

Level 4: Fault Protection by System Level Redundancy – the Level 3 fault containment regions will be replicated for system level fault containment. To further enhance the effectiveness of the system level redundancy, diversity is also employed in this level. For example, the X2000 implements the redundant IEEE 1394 buses with different topologies, such that any branch node in primary bus set is a leaf node in the backup bus set and vice versa [1][2]. In other words, there is no node that is a branch node for both buses. Hence, a failed node can only partition the bus in which it is a branch node. The redundant fault containment regions can be either in ready or dormant states, depending on the recovery time and other system requirments. If they are in ready state, voting or comparison of outputs among the regions will provide one more level of fault detection. In either case, the redundant regions are necessary resources for the fault recovery process.

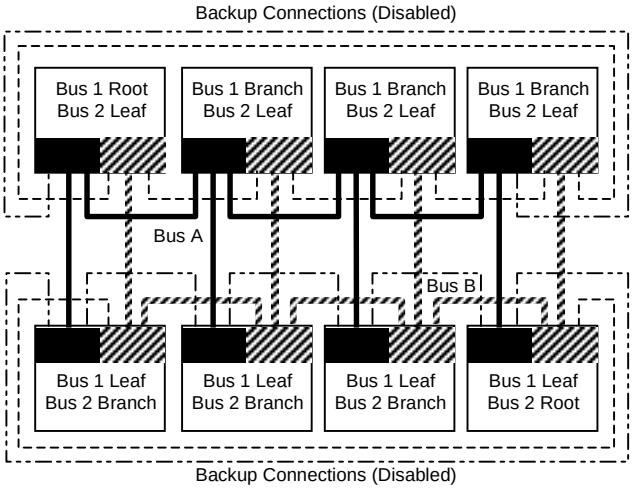

Figure 2: Stack-tree Topology of IEEE 1394 Bus

3. Comparison with the Single Level Custom Designed Fault Protection Approach

JPL has a long history of successfully applying fault protection techniques in space exploration. One of the most important techniques used in the design of space vehicle fault protection is fault containment [8][9]. In traditional designs, a spacecraft is divided into fault containment regions. Rigorous design effort is used to ensure no effects of a fault within a containment region will propagate to the other regions. The philosophy is trying to contain the faults with a single layer of containment regions, and the fault protection design is largely custom.

Since this approach has been so successful, an obvious question is whether the multi-level fault can achieve the same level of fault protection, given that the COTS layer inherently has lower fault coverage than the traditional custom-designed approach. In addition, since the multi-level approach requires additional layers of designs, would the overall cost be actually lower than the traditional single level approach?

In the following sections, these questions will be examined in an analytical way. To facilitate the analysis, the questions will be re-phrased as follows. First, if the COTS-based multi-level system were to have the same overall coverage as the custom-designed single level system, what would be the minimum coverage of the COTS layer required? Second, with that required minimum coverage, how would be the design cost of a COTS-based multi-level system in comparison with a custom-designed single level system?

4. Coverage Analysis of the Multi-Level Fault Protection Methodology

In order to simplify the analysis, the multi-level fault protection methodology shown in Figure 1 is represented with a fault propagation model depicted in Figure 3. The arrows indicate possible paths of fault propagation and the circular ends of the arrows signify the possible origins of faults. For comparison, the fault propagation model of the custom-designed single level fault tolerance is also illustrated in Figure 3 with the same graphical notations. The following analysis is conducted based on these models.

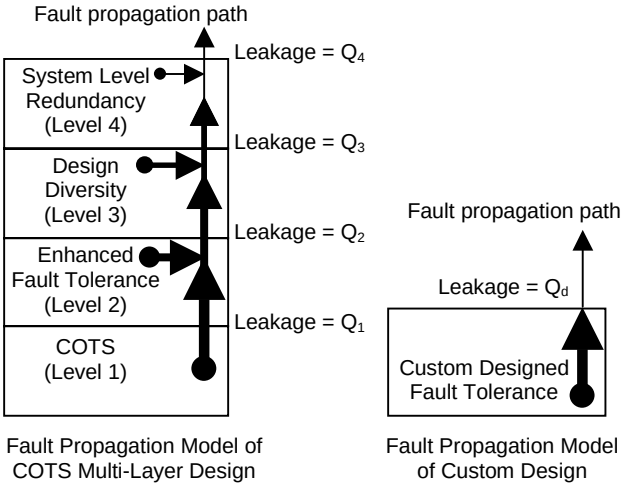

Figure 3: Comparison of Fault Propagation in Multi-Layer and Custom Design Approaches

In Figure 3, Q denotes the fault leakage of a containment region, which is defined as the probability that a fault occurs in a particular level but not detected or recovered by the containment regions at that level. The subscript denotes the containment region where the leakage occurs. More precisely,

F_i = Fault occured in region i
D_i = Fault detected by region i
R_i = Fault recovered by region I
E_i = Fault propagated out of region i
Q_i = P(Fault propagated out of region i)
= $P(E_i)$

Following is the derivation of the fault propagation probability in terms of coverage and probability of fault occurrence.

4.1. Fault Propagation in Lavel 1 (COTS Level)

$$Q_1 = P(E_1)$$
$$= P(\sim D_1 \mid F_1) * P(F_1) + P(D_1 \wedge \sim R_1 \mid F_1) * P(F_1)$$
$$= [P(\sim D_1 \mid F_1) + P(D_1 \wedge \sim R_1 \mid F_1)] * P(F_1)$$

Let's define *self-coverage* C_1 in Level 1 as follows.

$$C_1 = P(D_1 \wedge R_1 \mid F_1)$$
$$= 1 - [P(\sim D_1 \mid F_1) + P(D_1 \wedge \sim R_1 \mid F_1)]$$

Let's denote $P(F_1) = (1-e^{-\lambda_1 \tau})$, where λ = component failure rate and τ = mission time, then

$$Q_1 = (1-C_1) * (1-e^{-\lambda_1 \tau})$$

4.2. Fault Propagation in Level 2

As it is shown in Figure 3, there are two ways a fault can propagate out of the Level 2: faults propagated from Level 1 or fault originated in Level 2 is not detected or recovered by Level 2. Let's denote

$P(p_2)$ = P(propagated fault is not contained by level 2)
$P(s_2)$ = P(self-generated fault is not contained in level 2)

Therefore, the leakage of Level 2 can be expressed as

$$Q_2 = P(E_2)$$
$$= P(p_2) + P(s_2)$$

Notice that a fault originated in Level 1 cannot be contained by Level 2 when Level 2 either fails to detect or recover from that propagated fault. Therefore,

$$P(p_2) = P(\sim D_2 \mid E_1) * P(E_1) + P(D_2 \wedge \sim R_2 \mid E_1) * P(E_1)$$
$$= [P(\sim D_2 \mid E_1) + P(D_2 \wedge \sim R_2 \mid E_1)] * Q_1$$

Similarly, a fault originated in Level 2 cannot be contained in Level 2 because it fails to detect or recover from its own fault. Therefore,

$$P(s_2) = P(\sim D_2 \mid F_2) * P(F_2) + P(D_2 \wedge \sim R_2 \mid F_2) * P(F_2)$$
$$= [P(\sim D_2 \mid F_2) + P(D_2 \wedge \sim R_2 \mid F_2)] * P(F_2)$$

Let's define *self-coverage* C_2 in Level 2 as

$$C_2 = P(D_2 \wedge R_2 \mid F_2)$$
$$= 1 - [P(\sim D_2 \mid F_2) + P(D_2 \wedge \sim R_2 \mid F_2)]$$

Also, Let's define *propagation-coverage* Γ_2 in Level 2 as

$$\Gamma_2 = P(D_2 \wedge R_2 \mid E_1)$$
$$= 1 - [P(\sim D_2 \mid E_1) + P(D_2 \wedge \sim R_2 \mid E_1)]$$

Let's assume $P(F_2) = (1-e^{-\lambda_2 \tau})$. Then,

$$Q_2 = (1-\Gamma_2) * Q_1 + (1-C_2) * (1-e^{-\lambda_2 \tau})$$
$$= (1-\Gamma_2) * (1-C_1) * (1-e^{-\lambda_1 \tau}) + (1-C_2) * (1-e^{-\lambda_2 \tau})$$

4.3. Fault Propagation in Higher Levels

By the same reasoning as that shown in Level 2,

$$\begin{aligned} Q_3 &= (1-\Gamma_3) * Q_2 + (1-C_3) * (1-e^{-\lambda_3\tau}) \\ &= (1-\Gamma_3) * [(1-\Gamma_2) * (1-C_1) * (1-e^{-\lambda_1\tau}) \\ &\quad + (1-C_2) * (1-e^{-\lambda_2\tau})] + (1-C_3) * (1-e^{-\lambda_3\tau}) \\ &= (1-\Gamma_3) * (1-\Gamma_2) * (1-C_1) * (1-e^{-\lambda_1\tau}) \\ &\quad + (1-\Gamma_3) * (1-C_2) * (1-e^{-\lambda_2\tau}) + (1-C_3) * (1-e^{-\lambda_3\tau}) \end{aligned}$$

$$\begin{aligned} Q_4 &= (1-\Gamma_4) * Q_3 + (1-C_4) * (1-e^{-\lambda_4\tau}) \\ &= (1-\Gamma_4) * [(1-\Gamma_3) * (1-\Gamma_2) * (1-C_1) * (1-e^{-\lambda_1\tau}) \\ &\quad + (1-\Gamma_3) * (1-C_2) * (1-e^{-\lambda_2\tau}) + (1-C_3) * (1-e^{-\lambda_3\tau})] \\ &\quad + (1-C_4) * (1-e^{-\lambda_4\tau}) \\ &= (1-\Gamma_4) * (1-\Gamma_3) * (1-\Gamma_2) * (1-C_1) * (1-e^{-\lambda_1\tau}) \\ &\quad + (1-\Gamma_4) * (1-\Gamma_3) * (1-C_2) * (1-e^{-\lambda_2\tau}) \\ &\quad + (1-\Gamma_4) * (1-C_3) * (1-e^{-\lambda_3\tau}) + (1-C_4) * (1-e^{-\lambda_4\tau}) \end{aligned}$$

It should be noticed that the system will fail when the fault propagates through Level 4. Therefore, Q_4 is in fact the probability of system failure in this paper. However, this result can be generalized as follows.

$$Q_n = \sum_{i=1}^{n} (1-C_i) * (1-e^{-\lambda_i\tau}) \prod_{j=i+1}^{n} (1-\Gamma_j) \qquad \text{(Eq 1)}$$

4.4. Fault Propagation in the Single Level Custom Design

Similarly, the probability of fault propagation of the single level custom design can be derived. Let the probability of fault propagation in a traditional single level custom fault tolerance design

$$Q_d = (1-C_d) * (1-e^{-\lambda_d\tau})$$

4.6. Comparison of Fault Propagation in both Approaches

At this point, we are ready to answer the question of what would be the *self-coverage* C and *propagation-coverage* Γ required in each level for the multi-level fault tolerant system to have a equivalent overall coverage as the custom-designed single level system. For both approaches have the same leakage, $Q_d = Q_n$. In other words,

$$(1-C_d) * (1-e^{-\lambda_d\tau}) = \sum_{i=1}^{n} (1-C_i) * (1-e^{-\lambda_i\tau}) \prod_{j=i+1}^{n} (1-\Gamma_j) \qquad \text{(Eq 2)}$$

To simplify the analysis, let's assume that the *propagation-coverage* of levels 2, 3, and 4 are the same as the *self-coverage* of the COTS layer (i.e., $\Gamma_2 = \Gamma_3 = \ldots = C_1$, see Section 6 for the justification of this assumption). Furthermore, let's assume amount of circuits to implement fault tolerance in levels 2, 3, and 4 are very small in comparison to the COTS layer, so that their failure rate can be negligible. That is, $\lambda_1 >> \lambda_2 = \lambda_3 = \ldots \approx 0$. Then,

$$\begin{aligned} (1-C_d) * (1-e^{-\lambda_d\tau}) &= (1-C_1)^n * (1-e^{-\lambda_1\tau}) \\ (1-C_d) &= (1-C_1)^n * (1-e^{-\lambda_1\tau}) / (1-e^{-\lambda_d\tau}) \\ (1-C_1) &= [(1-C_d) * (1-e^{-\lambda_d\tau}) / (1-e^{-\lambda_1\tau})]^{1/n} \end{aligned}$$

Example 1:

Assuming the component failure rates of the COTS and the custom design are the same (i.e., $\lambda_d = \lambda_1 = \lambda$), and the coverage of the custom design $C_d = 0.999$, then for a 4-level system, the COTS coverage C_1 required to match the custom design is:

$$\begin{aligned} (1-C_1) &= [(1-C_d) * (1-e^{-\lambda\tau}) / (1-e^{-\lambda\tau})]^{1/4} \\ &= (1-C_d)^{1/4} \\ &= (1-0.9999)^{1/4} \\ &= 0.1 \\ C_1 &= 0.9 \end{aligned}$$

Example 2:

Assuming the component failure rate of the custom design is 100 times better than that of the COTS (i.e., $\lambda_d = 0.01\lambda_1$), λ_1 = 1e-4 faults/year, the mission time τ is 1 year, and the coverage of the custom design is $C_d = 0.9999$, then the COTS coverage C_1 required to match the custom design is:

$$\begin{aligned} (1-C_1) &= [(1-C_d) * (1-e^{-0.01\lambda_1\tau}) / (1-e^{-\lambda_1\tau})]^{1/4} \\ &= [(1-0.999900) * (1-0.999999) / (1-0.999900)]^{1/4} \\ &= 0.0316 \\ C_1 &= 0.9684 \end{aligned}$$

Notice that, in general, the lower the coverage, the less development cost. Conversely, the lower component failure rate, the more expensive the components. Therefore, in both examples, it is obvious that the COTS approach is less expensive than the custom design while their overall system coverage are the same (i.e. $(1 - Q_4) = (1 - Q_d)$). A more careful look of the cost analysis is given in the next section.

5. Cost Analysis of the Multi-Level Fault Protection Methodology

In order to perform a cost analysis, it is required to have a cost model as a function of coverage. As a general observation, the higher the total coverage, the more expensive to further improve it (see Section 6 for the justification of this assumption). Based on this assumption, the relationships of total coverage and cost can be modeled as shown in Figure 4.

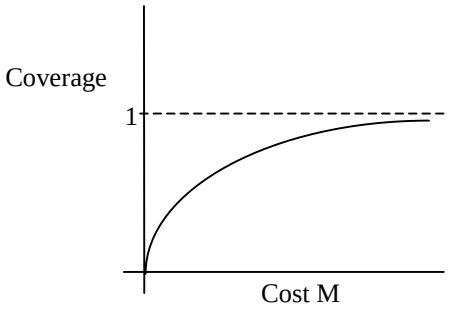

Figure 4: Cost as function of Coverage

For the single level custom-designed system, the cost and coverage can be related by:

$$C_d = 1 - e^{-\alpha M_d}$$
$$M_d = (-1/\alpha_d) \ln (1-C_d)$$
$$M_T = M_b + M_d = M_b + (-1/\alpha_d) \ln (1-C_d)$$

Where C_d is the system coverage, M_b is the cost for the basic function of the system, M_d is the cost for fault tolerance design, M_T is the total cost of the system, and α_d is a constant.

For the multi-level system, the cost of the COTS layer (M_1) includes basic function and any built-in fault protection features. It is set by the supplier and therefore is independent of the coverage. For the rest of the layers, there are two kinds of coverage: the *self-coverage* and the *propagation-coverage*. Hence, the total coverage in a level is:

$$T = C * \rho + \Gamma * (1 - \rho)$$

Where ρ is the fraction of faults covered by the self-coverage. The cost for each level is

$$M_t = M_1 - \sum_{i=2}^{n} (1/\alpha_i) \ln (1-T_i) \qquad \text{(Eq 5)}$$

$$M_i = (-1/\alpha_i) \ln (1-C_i) \qquad i > 2$$

Hence, the total cost of the multi-level system can be written as

$$M_t = M_1 + M_2 + M_3 + M_4 \qquad M_i = \text{cost of level } i$$
$$M_t = M_1 + (-1/\alpha_2) \ln (1-T_2) + (-1/\alpha_3) \ln (1-T_3) + (-1/\alpha_4) \ln (1-T_4)$$

In general, for a n-level system

A ratio between M_t and M_T can be found to compare the costs.

$$\frac{M_t}{M_T} = \frac{M_1 - \sum_{i=2}^{n} (1/\alpha_i) \ln (1-T_i)}{M_b + (-1/\alpha_d) \ln (1-C_d)} \qquad \text{(Eq 6)}$$

For simplicity, let's assume $\alpha_2 = \alpha_3 = \ldots = \alpha_d = \alpha$ and $T_2 = T_3 = \ldots = T$, then

$$\frac{M_t}{M_T} = \frac{M_1 - n(1/\alpha)\ln (1-T)}{M_b + (-1/\alpha) \ln (1-C_d)} \qquad \text{(Eq 7)}$$

In order to achieve any saving, the cost of the COTS layer (M_1) is

$$M_t < \frac{M_1 - n(1/\alpha)\ln (1-T)}{M_b + (-1/\alpha) \ln (1-C_d)} M_T \qquad \text{(Eq 8)}$$

Example 3:

Assuming a custom-designed system with a coverage of 0.9999 would cost $1,000,000 for designing the basic function and $200,000 for including the fault protection desgin. A similar system using COTS system would only cause $100,000 for both the basic functions and built-in fault protection features, but it has a coverage of only 0.9 Three additional levels of fault tolerance designs are added to the COTS system. Each layer also has a total coverage 0.9. The saving by the multi-level design can be calculated as follows. From the custom-designed system:

$$200000 = (-1/\alpha) * \ln(1 - 0.9999)$$
$$\alpha = 4.61 \times 10^{-5}$$

The cost of the multi-level system is

$$M_t = M_1 + 3(-1/\alpha)\ln (1-T) = 100000 + 3 * (-21715) * (-2.3026) = 250000$$

The cost saving is:

$$M_T - M_t = (1000000 + 200000) - 250000 = 950000$$

The leakage of the two systems can also be compared as follows. Let's assume the failure rates of Levels 2, 3,

and 4 are negligible (i.e., $\lambda_2 = \lambda_3 = \lambda_4 = 0$), then the *self-coverage* of these level is 0. Also, let's assume $\lambda_1 = \lambda_d = \lambda$, then the leakage of the multi-level system is

$$Q_4 = (1-0.9) * (1-0.9) * (1-0.9) * (1-0.9) * (1-e^{-\lambda\tau})$$
$$= 0.0001 * (1-e^{-\lambda\tau})$$

Meanwhile, the leakage of the custom-designed system is:

$$Q_d = 0.0001 * (1-e^{-\lambda\tau})$$

Therefore, if the multi-level system has simple logic at the levels above the COTS level, such that their failure rates are negligible, then both systems will have the same leakage while the multi-level system costs $950,000 less then the single layer custom-designed system.

6. Estimation of Self-Coverage and Propagation Coverage

The analytic technique mentioned above is useful only if there is a practical way of obtaining the values of the self-coverage C and propagation-coverage Γ. Since there are infinite number of possible faults in any real system and countless ways that the faults can propagate, it is impossible to obtain the true values of C and Γ. However, Failure Mode Effect and Criticality Analysis (FMECA) can be used to estimate these values. Following is a typical data sheet of FMECA that has been used at JPL [10][11].

Table 1: Typical FMECA Data Sheet

No.	Failure Mode	Possible Cause	Local Effect	System Effect	Prob/ Criticality	Detection Method	Recovery Method
1	Failure A1	Cause A1	Local Effect A1	System Effect A1	Low/6	Detection X1	Recovery X1
2	Failure B1	Cause B1	Local Effect B1	System Effect B1	High/5	Detection X1	Recovery X1
3	Failure C1	Cause C1	Local Effect C1	System Effect C1	Medium/6	Detection Y1	None
4	Failure D1	Cause D1	Local Effect D1	System Effect D1	Low/6	None	None
5	Failure E1	Cause E1	Local Effect E1	System Effect E1	Low/2	Detection Z1	Recovery Z1
6	Failure F1	Cause F1	Local Effect F1	System Effect F1	High/1	Detection X1	Recovery X1

In Table 1, the *Failure Mode* column describes observable behavior of the system or subsystem under failure conditions. Examples are (1) premature operation, (2) failure to operate at a prescribed time, (3) failure to cease operation at a prescribed time, (4) failure during the prescribed operating period. The *Possible Cause* column describes the mechanism that has the highest probability of inducing the failure. The *Local Effect* column describes the effect of the failure mode within the fault containment region. The *System Effect* column describes the effect of the failure mode at the system level or to the mission. The *Criticality and Probability* column describes and ranks the criticality of the function from 1 to 6, with Level 6 indicates complete loss of mission and Level 1 indicates minor or no impact on spacecraft life or performance. The *Detection Method* column identifies the indicators by which a particular failure mode is detected and the Recovery Method column identifies the resources and mechanisms to enable the system returns to operational state.

There are two observations need to be made about the detection and recovery method columns. First, detection or recovery methods might not be found for all failure modes. The fraction of the failure modes that can be detected and recovered is the estimated self-coverage C^*. For instance, the coverage C^* in Table 1 is 2/3. Second, some detection methods can detect several failure modes. In fact, a significant percentage of the failure modes are usually covered by the most obvious fault detection and recovery methods, while the remaining failure modes have to be covered by more sophisticated methods. A few difficult failure modes have to be covered by very specific and expensive methods. This is the reason why the cost can be modeled as a function of coverage as it is shown in Figure 4.

In the single level custom-designed fault tolerance approach, after the non-detectable or non-recoverable failure modes are discovered, the system engineer would tend to eliminate them by adding more fault detection or recovery mechanisms. This is possible in the custom-designed fault tolerance approach because the system engineer has total control of the design.

On the other hand, in a COTS-based design, the system engineer has no leverage to modify the design provided by the COTS supplier. Therefore, non-detectable or non-recoverable failure modes in the COTS cannot be eliminated. The multi-level fault tolerance method can handle such failure modes by using higher-level fault tolerance designs. A separate FMECA table will have to be developed for the each level of fault tolerance design. An example of the FMECA for Level 2 is shown in Table 2.

Notice that there are two types of entries in Table 2, the propagation type and the self-generated type. The first two failure modes, P1 and P2, are propagated from the COTS level (Level 1). One of them can be detected and recovered by the fault tolerance mechanisms in Level 2. The estimated *propagation-coverage* Γ^* is the fraction of

Table 2: Level 2 FMECA Data Sheet

No.	Failure Mode	Possible Cause	Local Effect	System Effect	Prob/ Criticality	Detection Method	Recovery Method
P1	Failure C1	Level 1	Local Effect C2	System Effect C2	Medium/6	Detection C2	Detection C2
P2	Failure D1	Level 1	Local Effect D2	System Effect D2	Low/6	None	None
S1	Failure A2	Cause A2	Local Effect A2	System Effect A2	Medium/6	Detection A2	None
S2	Failure B2	Cause B2	Local Effect B2	System Effect B2	Low/6	Detection B2	Recovery B2
S3	Failure C2	Cause C2	Local Effect C2	System Effect C2	Low/2	Detection C2	Recovery C2
S4	Failure D2	Cause D2	Local Effect D2	System Effect D2	High/1	Detection D2	Recovery D2

propagated failure modes from lower levels that can be detected and recovered by this level. Therefore, Γ_2^* in Table 2 is 0.5. The last four failure modes are originated from this level. The *self-coverage* C_2^* can be estimated in the same way as in Level 1, which in this case is 0.75. The fraction of failure modes covered by the *self-coverage* (ρ) in this case is 2/3, so that the total coverage $T = 1/2 * 1/3 + 3/4 * 2/3 = 2/3$.

After Γ^*, C^*, ρ, and T are estimated for each level, and assume λ and α can be found, then the system coverage and development costs can be estimated as in Equations 1 and 4, respectively. Furthermore, if the fraction of faults covered by the *self-coverage* is the same as that of the *propagation-coverage*, then C and Γ will have same values. This is the simplifying assumption used in the derivation of Equations 3 and 7.

7. Conclusion and Future Work

This paper has described how the multi-level fault tolerant approach can improve the system reliability. An analysis also shows that if the multi-level system has negligible failure rates in the levels above the COTS, then it will have the same leakage as the single layer custom-designed system while the cost is significantly lower. Hence, the multi-layer fault tolerance is a viable approach to enhance the reliability of a COTS system under certain circumstances. These circumstances will be further investigated in the future.

Acknowledgment

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration

Reference:

[1] S. N. Chau, L. Alkalai, A. T. Tai, and J. B. Burt, "The design of a fault-tolerant cots-based bus architecture," in *IEEE Transaction on Reliability*, Dec. 1999.

[2] A. T. Tai, S. N. Chau, and L. Alkalai, "COTS-based fault tolerance in deep space: qualitative and quantitative analyses of a bus network architecture," in *Proceedings of the 4th IEEE High-Assurance System Engineering Symposium*, (Washington, D.C.), Nov. 1999.

[3] IEEE 1394, *Standard for a High Performance Serial Bus*. Institute of Electrical and Electronic Engineers, Jan. 1995.

[4] D. Anderson, *FireWire System Architecture, IEEE 1394*. PC System Architecture Series, MA: Addison Wesley, 1998.

[5] D. Paret and C. Fenger, *The I^2C Bus: From Theory to Practice*. John Wiley, 1997.

[6] Philips Semiconductor, *The I^2C-Bus Specification Version 2.0*, Philips Semiconductor, Dec. 1998.

[7] IEEE P1394A, *Standard for a High Performance Serial Bus (Supplement)*, Draft 2.0. Institute of Electrical and Electronic Engineers, Mar. 1998.

[8] J. Donaldson, "Cassini Orbiter Functional Requirements Book: Command and Data Subsystem," *JPL Document CAS-4-2006*, June 28, 1994.

[9] A. Avizienis and D. Rennels, "The Evolution of Fault Tolerant Computing at the Jet Propulsion Laboratory & at UCLA 1960-86," Computer Science Department Technical Report CSD-870022, University of California, Los Angeles, June 1987.

[10] P. Noone, "CDS Interface FMECA," *JPL Document D-11736*, Jet Propulsion Laboratory, June 1994.

[11] J. Arnett et al, " Jet Propulsion Laboratory Reliability Analyses Handbook," *JPL Document D-5703*, Jet Propulsion Laboratory, July 1990.