Design and Analysis of Fault-Tolerant Digital Systems

Barry W. Johnson

University of Virginia, Charlottesville

Addison-Wesley Publishing Company

Reading, Massachusetts • Menlo Park, California • New York
Don Mills, Ontario • Wokingham, England • Amsterdam • Bonn
Sydney • Singapore • Tokyo • Madrid • San Juan

4

Evaluation Techniques

4.1 Introduction

The techniques presented in the previous chapter form a collection of approaches that can be used to achieve fault tolerance. The specific techniques used in a given application depend on not only the application, but also on the ideas and philosophies of the designers. One team of designers might develop one fault-tolerant design, whereas another team would choose a completely different approach for the same application. It is important to be

able to compare two or more approaches for a particular application. The process of comparison is actually a critical part of the design process because it leads to tradeoffs and modifications of the design. It is through such tradeoffs that the most optimal design is developed.

The methods for evaluating fault-tolerant systems can be divided into two major categories: *quantitative* and *qualitative*. Qualitative measures are typically subjective in nature and describe the benefits of one design over another. Examples include the flexibility of a particular design and the degree to which the fault tolerance techniques are transparent to the user of a system. Quantitative evaluation techniques produce numbers that can be used to compare two or more systems. Examples include specific numbers for the reliability, availability, maintainability, mission life, or fault coverage of a system.

The purpose of this chapter is to examine, in detail, the techniques that are available for evaluating fault-tolerant systems. The final section contains a specific tradeoff example that shows how the evaluation techniques can be used to make design decisions and tradeoffs.

4.2 Quantitative Evaluation Methods

As previously mentioned, the purpose of quantitative evaluation methods is to assign a number to some attribute of a system such that the attribute can be compared among systems. For example, the reliability of one system may be greater than that of another, or the weight and cost of one approach may be less than that of another. Reliability, weight, and cost are three examples of quantitative measures that can be used to evaluate systems. Certainly, the weight and cost of a system are extremely important, and in some cases can be more important than reliability or fault tolerance.

In this chapter, we consider several approaches to quantitative evaluation, including the failure rate, mean time to failure (MTTF), mean time between failure (MTBF), fault coverage, reliability analysis, safety analysis, availability analysis, maintainability analysis, redundancy ratios, and costs. Several techniques for generating the reliability of a system are presented.

4.2.1 Failure Rate and the Reliability Function

Intuitively, the **failure rate** is the expected number of failures of a type of device or system per a given time period [Shooman 1968]. For example, if a computer fails, on the average, once every 2000 hours, the computer has a failure rate of one failure per 2000 hours, or 1/2000 failures/hour. The failure rate is typically denoted as λ. The failure rate is one measure that can be used to compare systems or components. In selecting a computer for a

banking application, one would like to select a computer that fails as infrequently as possible. If redundancy has been incorporated as a means of achieving fault tolerance, the failure rate of the redundant system should be lower than the failure rate of a similar, nonredundant system.

To more clearly understand the mathematical basis for the concept of a failure rate, recall the definition of the reliability function. The reliability $R(t)$ of a component, or a system, is the conditional probability that the component operates correctly throughout the interval $[t_0, t]$ given that it was operating correctly at time t_0. Suppose that we test N identical components by placing all N components in operation at time t_0 and recording the number of failed and working components at time t. Let $N_f(t)$ be the number of components that have failed at time t and $N_o(t)$ be the number of components that are operating correctly at time t. It is assumed that once a component fails it remains failed indefinitely. The *reliability* of the components at time t is given by

$$R(t) = \frac{N_o(t)}{N} = \frac{N_o(t)}{N_o(t) + N_f(t)}$$

which is simply the probability that a component has survived the interval $[t_o, t]$. The probability that a component has not survived the time interval is called the *unreliability* and is given by

$$Q(t) = \frac{N_f(t)}{N} = \frac{N_f(t)}{N_o(t) + N_f(t)}$$

Note that at any time t, $R(t) = 1.0 - Q(t)$ because

$$R(t) + Q(t) = \frac{N_o(t) + N_f(t)}{N_o(t) + N_f(t)} = 1.0$$

If we write the reliability function as

$$R(t) = 1.0 - \frac{N_f(t)}{N}$$

and differentiate $R(t)$ with respect to time, we obtain

$$\frac{dR(t)}{dt} = -\frac{1}{N}\frac{dN_f(t)}{dt}$$

which can be written as

$$\frac{dN_f(t)}{dt} = -N\frac{dR(t)}{dt}$$

The derivative of $N_f(t)$, $dN_f(t)/dt$, is simply the instantaneous rate at which components are failing. At time t, there are still $N_o(t)$ components operational. Dividing $dN_f(t)/dt$ by $N_o(t)$ we obtain

$$z(t) = \frac{1}{N_o(t)} \frac{dN_f(t)}{dt}$$

$z(t)$ is called the *hazard function*, *hazard rate*, or *failure rate function*. The units for the failure rate function are failures per unit of time.

The failure rate function can be expressed in different ways. For example, $z(t)$ can be written strictly in terms of the reliability function $R(t)$ as

$$z(t) = \frac{1}{N_o(t)} \frac{dN_f(t)}{dt} = \frac{1}{N_o(t)} \left[-N \frac{dR(t)}{dt} \right] = -\frac{\dfrac{dR(t)}{dt}}{R(t)}$$

Similarly, $z(t)$ can be written in terms of the unreliability $Q(t)$ as

$$z(t) = -\frac{\dfrac{dR(t)}{dt}}{R(t)} = \frac{\dfrac{dQ(t)}{dt}}{1 - Q(t)}$$

The derivative of the unreliability $dQ(t)/dt$ is called the *failure density function*.

The failure rate function clearly depends on time because the value of $N_o(t)$ and the value of $dN_f(t)/dt$ change as functions of time. However, experience has shown that the failure rate function for electronic components does have a period where the value of $z(t)$ is approximately constant. The commonly accepted relationship between the failure rate function and time for electronic components is called the **bathtub curve** and is illustrated in Fig. 4.1. The bathtub curve assumes that during the early life of systems failures occur frequently due to substandard or weak components. The decreasing part of the bathtub curve is called the *early-life* or *infant mortality* region. At the opposite end of the curve is the wear-out region where systems have been functional for a long period of time and are beginning to experience failures due to the physical wearing of electronic or mechanical components. The increasing part of the bathtub curve is called the *wear-out phase*. During the intermediate region, the failure rate function is assumed to be a constant. The constant portion of the bathtub curve is called the *useful life* phase of the system, and the failure rate function is assumed to have a value of λ during that period. (λ is referred to as the failure rate and is normally expressed in units of failures per hour.)

The period of a constant failure rate is typically the most useful portion of a system's life. During the useful-life phase, the system is providing its most predictable service to its users. We usually attempt to get a system beyond the infant mortality stage by using the concept of *burn-in* to remove weak components. Burn-in implies operating a system, often at an acceler-

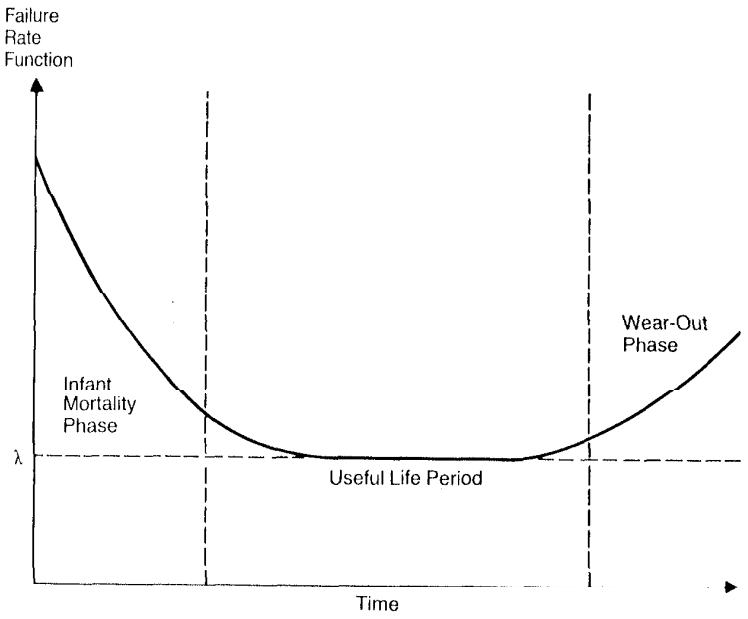

Fig. 4.1 Bathtub curve relationship between the failure rate function and time—the failure rate function is constant during the useful life period.

ated pace, prior to placing the system into service to get the system to the beginning of the useful-life period. In addition, the system is normally replaced before it enters the wear-out phase of its life. Thus, the primary interest is the performance of the system during the useful-life phase.

As noted previously, the failure rate function can be related to the reliability function as

$$z(t) = \frac{1}{N_o(t)} \frac{dN_f(t)}{dt} = -\frac{N}{N_o(t)} \frac{dR(t)}{dt}$$

We know, however, that the quantity $N/N_o(t)$ is the inverse of the reliability function $R(t)$ so we can write

$$z(t) = -\frac{1}{R(t)} \frac{dR(t)}{dt}$$

The result is a differential equation of the form

$$\frac{dR(t)}{dt} = -z(t)R(t)$$

If we assume that the system is in the useful-life stage where the failure rate function has a constant value of λ, the solution to the differential equation is well known to be an exponential function of the parameter λ given by

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

where λ is the constant failure rate. The exponential relationship between the reliability and time is known as the **exponential failure law,** which states that for a constant failure rate function, the reliability varies exponentially as a function of time.

The exponential failure law is extremely valuable for the analysis of electronic components and is by far the most commonly used relationship between reliability and time. Many cases, however, cannot assume that the failure rate function is constant, so the exponential failure law cannot be used; other modeling schemes and representations must be employed. An example of a time-varying failure rate function is found in the analysis of software. Software failures are the result of design faults, and as a software package is used, design faults are discovered and corrected. Consequently, the reliability of software should improve as a function of time, and the failure rate function should decrease.

A common modeling technique used to represent time varying failure rate functions is the **Weibull distribution** [Siewiorek and Swarz 1982]. The failure rate function associated with the Weibull distribution is given by

$$z(t) = \alpha\lambda(\lambda t)^{\alpha-1}$$

where α and λ are constants that control the variation of the failure rate function with time. The failure rate function given by the Weibull distribution is intuitively appealing. For example, if the value of α is 1, $z(t)$ is simply the constant λ. If α is greater than 1, $z(t)$ increases as time increases; if α is less than 1, $z(t)$ decreases as time increases. Consequently, we can envision modeling software using the Weibull distribution with the constant α being less than 1.

The reliability function that results from the Weibull distribution is the solution to the differential equation

$$\frac{dR(t)}{dt} = -z(t)R(t) = -\alpha\lambda(\lambda t)^{\alpha-1}R(t)$$

and is given by

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

The expression for $R(t)$ can be verified by calculating the derivative $dR(t)/dt$. Specifically,

$$\frac{dR(t)}{dt} = -e^{-(\lambda t)^{\alpha}}\alpha(\lambda t)^{\alpha-1}(\lambda) = -\alpha\lambda(\lambda t)^{\alpha-1}e^{-(\lambda t)^{\alpha}} = -z(t)R(t)$$

As can be seen, certain values of α result in a reliability function that increases as time increases. For example, if $\alpha = -1$, the reliability is given by

$$R(t) = e^{-1/\lambda t}$$

which approaches 1 as t approaches infinity and is 0 when t is 0. Also note that for $\alpha = 1$, the reliability function is identical to the exponential failure law.

Although time-varying failure rate functions are important in the analysis of software and other systems, by far the most common analysis is performed assuming a constant failure rate function and the exponential failure law. The remainder of this chapter assumes the exponential failure law.

4.2.2 Failure Rate Calculation

An important aspect in the analysis of systems is the estimation of the failure rate of specific components. The most common technique for estimating the failure rate is the United States Department of Defense (USDOD) MIL-HDBK-217 standard ([USDOD 1965], [USDOD 1974], and [USDOD 1979]). Several versions of the standard have been published, including the original standard, MIL-HDBK-217 [USDOD 1965], as well as several revisions, which include, for example, MIL-HDBK-217C [USDOD 1979]. In each version of the standard, the objective has been to develop a model for the failure rate of electronic components using experimental data obtained by analyzing the failures of actual devices. Here, we only summarize the model and the important parameters that are used in calculating the failure rate.

The MIL-HDBK-217B ([Siewiorek and Swarz 1982] and [USDOD 1974]) model predicts the constant failure rate of an integrated circuit (IC) as

$$\lambda = \pi_L \pi_Q (C_1 \pi_T + C_2 \pi_E) \pi_P \quad \textit{failures per million hours}$$

where π_L is a *learning* factor, π_Q is a *quality* factor, π_T is a *temperature* factor, π_E is an *environmental* factor, π_P is a *pin* factor, and C_1 and C_2 are *complexity* factors.

The learning factor π_L represents the overall maturity of the fabrication process used to produce the IC. Devices produced using a new, and as yet unproven, manufacturing process are assigned a learning factor of 10, while those produced using a proven process are assigned a learning factor of 1. In other words, the learning factor represents the overall confidence in the ability of the fabrication process to produce devices that will fail infrequently.

The quality factor π_Q represents the amount of device screening that occurs. Device screening is simply the testing that a device goes through prior to being sold by a manufacturer. The lowest level of screening implies that no testing is performed. In other words, the manufacturer simply produces

and sells the IC without verifying that it is operational. At higher levels of screening, the manufacturer randomly selects ICs from a manufacturing run and subjects the selected ICs to certain tests. At even higher levels of screening, the manufacturer thoroughly tests each IC produced. In MIL-HDBK-217B, the quality factor varies from 1 to 300, depending on the level of screening.

The four primary screening levels for ICs are Class A, Class B, Class C, and Class D. Classes A and B are the highest screening levels and are used typically in military applications. The quality factor is 1 for Class A and 2 for Class B. Class C components are representative of high-quality commercial components and have a quality factor of 16. Finally, Class D represents a standard, hermetically sealed commercial component and has a quality factor of 150.

The temperature factor π_t is a function of the device technology, operating temperature, device packaging technology, and power dissipation. The specific equations used for the temperature factor are

$$\pi_T = 0.1e^{(-8120\{(1/(T_j+273))-(1/298)\})}$$

for linear circuits and

$$\pi_T = 0.1e^{(-4794\{(1/(T_j+273))-(1/298)\})}$$

for digital bipolar circuits. T_j is the junction temperature and is expressed in degrees Celsius. The second equation given above for π_T is used for transistor-transistor logic (TTL) circuits. As an example, the calculation for a TTL circuit with a junction temperature of 25 degrees Celsius is

$$\pi_T = 0.1e^{(-4794\{(1/(25+273))-(1/298)\})} = 0.1e^{(0.0)} = 0.1$$

The environmental factor π_E is a function of the harshness of the environment. For example, components operated in an air-conditioned computer room have a much lower environmental factor than those operated on a typical factory floor or in an airborne application. Typical values of the environmental factor vary in the MIL-HDBK-217B standard from 0.2 to 10.0. For example, components located in a computer room have an environmental factor of 0.2; components located in an uninhabited airborne environment have an environmental factor of 6.0; components in a launched missile have an environmental factor of 10.0.

The pin factor π_P is a function of the number of pins on the IC package. In MIL-HDBK-217B, the pin factor ranges from 1.0 to 1.2, for large-scale integration (LSI) technology, as the number of pins increases from 1 to greater than 64. LSI logic is normally defined as an IC having between 100 and 1000 logic gates. For LSI devices, the pin factor is 1.0 if the IC has 25 or fewer pins, 1.1 if the IC has between 26 and 64 pins, and 1.2 for an LSI device having more than 64 pins.

The final factors included in the failure rate model are the complexity factors C_1 and C_2. The complexity factors are a function of the number of gates for logic circuits, the number of transistors for linear circuits, and the number of bits for memories. The complexity factors for ICs having between 100 and 1300 gates are

$$C_1 = (0.0187)e^{(0.00471)(N_g)}$$
$$C_2 = (0.013)e^{(0.00423)(N_g)}$$

where N_g is the number of gates on the IC. The complexity factors for logic having fewer than 100 gates are

$$C_1 = (0.00129)N_g^{(0.677)}$$
$$C_2 = (0.00389)N_g^{(0.359)}$$

where N_g is the number of gates on the IC. The complexity factors for linear circuits are

$$C_1 = (0.00056)N_t^{(0.763)}$$
$$C_2 = (0.0026)N_t^{(0.547)}$$

where N_t is the number of transistors on the IC. The complexity factors for read-only memory (ROM) are

$$C_1 = (0.00114)B^{(0.603)}$$
$$C_2 = (0.00032)B^{(0.646)}$$

where B is the total number of bits in the memory. Finally, the complexity factors for random access memory (RAM) are

$$C_1 = (0.00199)B^{(0.603)}$$
$$C_2 = (0.00056)B^{(0.644)}$$

where B is the total number of bits in the memory.

As an example, consider the calculation of the failure rate for a device having 24 pins and 500 logic gates. We will assume a learning factor of 1.0, a quality factor of 16, a temperature factor of 0.35, and an environmental factor of 0.2. The pin factor is 1.0 for devices with 25 or fewer pins. The complexity factors are calculated as

$$C_1 = 0.0187e^{(0.00471)(500)} = 0.19706$$
$$C_2 = 0.013e^{(0.00423)(500)} = 0.10776$$

and the resulting failure rate is

$$\lambda = \pi_L \pi_Q (C_1 \pi_T + C_2 \pi_E)\pi_P =$$
$$(1.0)(16)[(0.19706)(0.35) + (0.10776)(0.2)](1.0) = 1.448$$

which is in failures per million hours.

Table 4.1 shows some additional typical values computed using the MIL-HDBK-217B standard.

4.2.3 Mean Time to Failure

In addition to the failure rate, the **mean time to failure (MTTF)** is a useful parameter to specify the quality of a system. The MTTF is the expected time that a system will operate before the *first* failure occurs. For example, if we have N identical systems placed into operation at time $t = 0$, and we measure the time that each system operates before failing, the average time is the MTTF. If each system i operates for a time t_i before encountering the first failure, the MTTF is given by

$$\text{MTTF} = \frac{\sum_{i=1}^{N} t_i}{N}$$

The MTTF can be calculated by finding the expected value of the time of failure. From probability theory, we know that the expected value of a random variable X is

$$E[X] = \int_{-\infty}^{\infty} xf(x)\,dx$$

TABLE 4.1 Typical failure rates calculated using MIL-HDBK-217B ($\pi_L = 1$, $\pi_Q = 16$, $\pi_T = 0.35$, $\pi_E = 0.2$, $\pi_n = 1$)

Number of logic gates	Failure rate (Failures per million hours)
(a) Logic circuits	
50	0.1527
100	0.2312
200	0.3655
500	1.4483
1000	14.4880
(b) Memories (RAM)	
Number of bits	**Failure rate (Failures per million hours)**
1024 (1K)	0.8837
2048 (2K)	1.3491
8192 (8K)	3.1453
16,384 (16K)	4.8033
32,768 (32K)	7.3362

where $f(x)$ is the *probability density function*. In reliability analysis we are interested in the expected value of the time of failure (MTTF), so

$$\text{MTTF} = \int_0^\infty t f(t)\, dt$$

where $f(t)$ is the *failure density function*, and the integral runs from 0 to ∞ because the failure density function is undefined for times less than 0. We know, however, that the failure density function is

$$f(t) = \frac{dQ(t)}{dt}$$

so, the MTTF can be written as

$$\text{MTTF} = \int_0^\infty t \frac{dQ(t)}{dt}\, dt$$

Using integration by parts and the fact that $dQ(t)/dt = -dR(t)/dt$, we can show that

$$\text{MTTF} = \int_0^\infty t \frac{dQ(t)}{dt}\, dt = -\int_0^\infty t \frac{dR(t)}{dt}\, dt = \left[-tR(t) + \int R(t)\, dt\right]_0^\infty = \int_0^\infty R(t)\, dt$$

The term $-tR(t)$ clearly disappears when $t = 0$; but, it also disappears when $t = \infty$ because $R(\infty) = 0$. Consequently, the MTTF is defined in terms of the reliability function as

$$\text{MTTF} = \int_0^\infty R(t)\, dt$$

which is valid for any reliability function that satisfies $R(\infty) = 0$.

If the reliability function obeys the exponential failure law, the result of calculating the MTTF is given by

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

In other words, the MTTF of a system that obeys the exponential failure law is the inverse of the failure rate of the system. Note that the reliability at a time equal to the MTTF for the exponential failure law is

$$R(\text{MTTF}) = R\left(\frac{1}{\lambda}\right) = e^{-\lambda(1/\lambda)} = e^{-1} = 0.3678$$

In other words, a system obeying the exponential failure law has a probability of 0.3678 of *not* experiencing a failure before a time equal to the MTTF, given that the system was perfect at the beginning of that time period.

Stated differently, a system obeying the exponential failure law has a probability of 0.6322 of failing during a time period equal to the MTTF, given that the system was perfect at the beginning of that time period.

4.2.4 Mean Time to Repair

The **mean time to repair (MTTR)** is simply the average time required to repair a system. The MTTR is extremely difficult to estimate and is often determined experimentally by injecting a set of faults, one at a time, into a system and measuring the time required to repair the system in each case. The measured repair times are averaged to determine an average time to repair. In other words, if the i^{th} of N faults requires a time t_i to repair, the MTTR is estimated as

$$\text{MTTR} = \frac{\sum_{i=1}^{N} t_i}{N}$$

Often the estimate of the MTTR is improved by averaging over several repair personnel to account for the differences in the abilities of these personnel. For example, if the set of N faults is repaired by M personnel, each of the personnel has an average time to repair, say, MTTR_i, which is the MTTR for the i^{th} person. The estimate of the overall MTTR is the average of the individual MTTRs. In other terms,

$$\text{MTTR} = \frac{\sum_{i=1}^{M} \text{MTTR}_i}{M}$$

The MTTR is normally specified in terms of a **repair rate** μ, which is the average number of repairs that occur per time period. The units of the repair rate are normally number of repairs per hour. The MTTR and the repair rate μ are related by

$$\text{MTTR} = \frac{1}{\mu}$$

4.2.5 Mean Time Between Failure

It is very important to understand the difference between the MTTF and the **mean time between failure (MTBF)**. Unfortunately, these two terms are *often used interchangeably*. Although the numerical difference is small in many cases, the conceptual difference is very important. The MTTF is the average time until the *first* failure of a system, whereas the MTBF is the average time *between* failures of a system. As noted in the previous section, we can estimate the MTTF for a system by placing each of a population of N

identical systems into operation at time $t = 0$, measuring the time required for each system to encounter its first failure, and averaging these times over the N systems. The MTBF, however, is calculated by averaging the time between failures, including any time required to repair the system and place it back into an operational status. In other words, each of the N systems is operated for some time T and the number of failures encountered by the i^{th} system is recorded as n_i. The average number of failures is computed as

$$n_{\text{avg}} = \sum_{i=1}^{N} \frac{n_i}{N}$$

Finally, the MTBF is

$$\text{MTBF} = \frac{T}{n_{\text{avg}}}$$

In other words, the MTBF is the total operation time T, divided by the average number of failures experienced during the time T.

If we assume that all repairs to a system make the system perfect once again just as it was when it was new, the relationship between the MTTF and the MTBF is as illustrated in Fig. 4.2. Once successfully placed into operation, a system operates, on the average, a time corresponding to the MTTF before encountering the first failure. The system then requires some time, MTTR, to repair the system and place it back into operation once again. The system then is perfect once again and will operate for a time corresponding to the MTTF before encountering its next failure. The time between the two failures is the sum of the MTTF and the MTTR and is the MTBF. Thus, the difference between the MTTF and the MTBF is the MTTR. Specifically, the MTBF is given by

$$\text{MTBF} = \text{MTTF} + \text{MTTR}$$

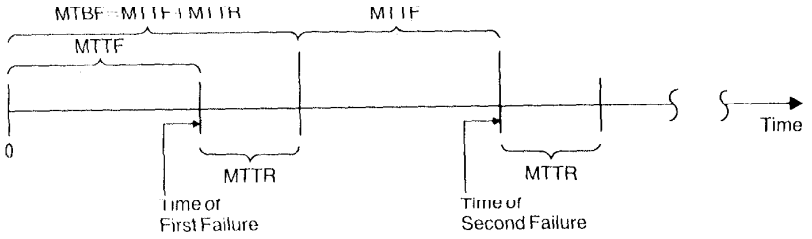

Fig. 4.2 Relationship between the MTBF and the MTTF.

In most practical applications the MTTR is a small fraction of the MTTF, so the *approximation* that the MTBF and MTTF are equal is often quite good. Conceptually, however, it is crucial to understand the difference between the MTBF and the MTTF.

4.2.6. Fault Coverage

An extremely important parameter in the design and analysis of fault-tolerant systems is **fault coverage.** The fault coverage available in a system can have a tremendous impact on the reliability, safety, and other attributes of the system. There are several types of fault coverage, depending on whether the designer is concerned with fault detection, fault location, fault containment, or fault recovery. In addition, there are two primary definitions of fault coverage: one is intuitive, the other is more mathematical.

The intuitive definition is that *coverage* is a measure of a system's ability to perform fault detection, fault location, fault containment, and/or fault recovery. The four primary types of fault coverage are fault detection coverage, fault location coverage, fault containment coverage, and fault recovery coverage. **Fault detection coverage** is a measure of a system's ability to detect faults. For example, a system requirement may be that a certain fraction of all faults be detected; the fault detection coverage is a measure of the system's capability to meet such a requirement. **Fault location coverage** is a measure of a system's ability to locate faults. Once again, it is very common to require a system to locate faults to within easily replaceable modules, and the fault location coverage is a measure of the success with which fault location is performed. **Fault containment coverage** is a measure of a system's ability to contain faults; specifically, the fault containment coverage represents a system's ability to make the *extent* attribute of faults *local* instead of *global*. Finally, **fault recovery coverage** is a measure of a system's ability to recover from faults and maintain an operational status. Clearly, a high fault recovery coverage requires high fault detection, location, and containment coverages.

In the evaluation of fault-tolerant systems, the fault recovery coverage is the most commonly considered, and the general term "fault coverage" is often used to mean fault recovery coverage. In other words, fault coverage is interpreted as a measure of a system's ability to successfully recover after the occurrence of a fault, therefore tolerating the fault. Therefore, when using the term "fault coverage," make sure that the type of coverage—detection, location, containment, or recovery—is understood.

The remainder of this chapter uses the term "fault coverage" to imply fault recovery coverage since fault recovery is the most common form of coverage encountered. In all cases, however, it will be made clear whether detection, location, containment, or recovery coverage is being considered.

Fault coverage is mathematically defined as the conditional probability that, given the existence of a fault, the system recovers [Bouricius, Carter, and Schneider 1969]. In mathematical terms, fault coverage is written as

$$C = P(\text{fault recovery} \mid \text{fault existence})$$

where C is the fault coverage and $P(\text{fault recovery} \mid \text{fault existence})$ is read as the probability of fault recovery *given* the existence of a fault. Recall that fault recovery is the process of maintaining or regaining operational status after a fault occurs.

The fundamental problem with fault coverage is that it is extremely difficult to calculate. Probably the most common approach to estimating fault coverage is to develop a list of all the faults that can occur in a system and to form, from that list, a list of faults that can be detected, a list of faults that can be located, a list of faults that can be contained, and a list of faults from which the system can recover. The fault detection coverage factor, for example, is then computed as simply the fraction of faults that can be detected; that is, the number of faults detected divided by the total number of faults. The remaining fault coverage factors are calculated in a similar manner. As an example, consider the circuit shown in Fig. 4.3 which has fifteen potential sites of stuck-at-1 or stuck-at-0 faults; consequently, there are a total of 30 faults. Table 4.2 shows the input combinations that yield erroneous outputs when certain faults are present, therefore detecting the faults. Note that the circuit performs correctly even if a single stuck-at-0 fault on one of the lines F, G, or M occurs. In other words, a single stuck-at-0 fault on line F, G, or M cannot be detected. As a result, the fault detection coverage for the circuit of Fig. 4.3 is (30–3)/30, or 0.9. In other words, 90% of the stuck-at-1 and stuck-at-0 faults are detected by at least one of the input combinations.

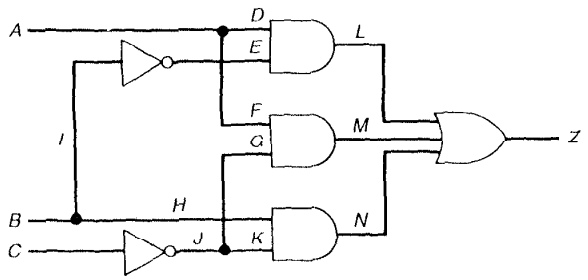

Fig. 4.3 Combinational circuit to illustrate fault detection coverage.

TABLE 4.2 Input patterns capable of detecting faults (test vectors) in the circuit of Fig. 4.3

Fault	Number of test vectors	Test vectors ABC
A_0	2	100, 101
A_1	2	000, 001
B_0	2	010, 111
B_1	2	000, 101
C_0	2	011, 111
C_1	2	010, 110
D_0	1	101
D_1	2	000, 001
E_0	1	101
E_1	1	111
F_0	0	—
F_1	2	000, 101
G_0	0	
G_1	1	111
H_0	1	010
H_1	1	000
I_0	1	111
I_1	1	101
J_0	2	010, 110
J_1	2	011, 111
K_0	1	010
K_1	2	011, 111
L_0	1	101
L_1	4	000, 001, 011, 111
M_0	0	—
M_1	4	000, 001, 011, 111
N_0	1	010
N_1	4	000, 001, 011, 111
Z_0	4	010, 100, 101, 110
Z_1	4	000, 001, 011, 111

Several important points should be made about the estimation of coverage. First, the estimation of fault coverage requires the definition of the types of faults that can occur. Stating that the fault detection coverage is 0.9, for example, is meaningless unless the types of faults considered are identified. For example, the fault detection coverage for the circuit of Fig. 4.3 is 0.9 for all stuck-at-1 and stuck-at-0 faults, but the fault detection coverage may decrease substantially if stuck-open faults are included.

A second important point about the fault coverage is that it is typically assumed to be a constant. It is easy to envision applications in which the

probability of detecting a fault, for example, increases as a function of time, after the occurrence of the fault. However, to simplify the analysis, the various fault coverages are normally assumed to be constants.

4.3 Reliability Modeling

Reliability is perhaps one of the most important attributes of systems. Almost all specifications for systems mandate that certain values for reliability be achieved and in some way proved. We have seen in the previous sections that reliability can be determined experimentally if a set of N systems is operated over a period of time and the number of systems that fail during that time period is recorded. One problem with the experimental approach is the number of systems that would be required to achieve a level of confidence in the experimental results. This is particularly a problem when costs limit the number of systems that can be built. For example, the space shuttle program could not afford to build 1000 of its on-board processing systems such that reliability could be experimentally verified.

A second problem with the experimental approach is the time required to run such experiments. Many systems today are being designed to achieve reliabilities of 0.9_7, or higher, after ten hours of operation. Using the exponential failure law, a reliability of 0.9_7 corresponds to a failure rate of 10^{-8} failures per hour. Therefore, on the average, we would have to wait approximately 100 million hours, or approximately 11,416 years for the first failure to occur. Clearly, we need alternatives to the experimental approach.

The most popular reliability analysis techniques are the analytical approaches. Of the analytical techniques, combinatorial modeling and Markov modeling are the two most commonly used approaches.

4.3.1 Combinatorial Models

Combinatorial models use probabilistic techniques that enumerate the different ways in which a system can remain operational. The probabilities of the events that lead to a system being operational are calculated to form an estimate of the system's reliability.

The reliability of a system is generally derived in terms of the reliabilities of the individual components of the system. The two models of systems that are most common in practice are the series and the parallel. In a **series system,** each element of the system is required to operate correctly for the system to operate correctly. In a **parallel system,** on the other hand, only one of several elements must be operational for the system to perform its functions correctly.

In practice, systems are typically combinations of series and parallel subsystems. Once we have discussed both the series and parallel structures,

we will examine techniques for modeling systems that contain *both* series and parallel subsystems.

Series Systems

The series system is best thought of as a system that contains no redundancy; that is, each element of the system is needed to make the system function correctly. For example, a digital filter that contains a microprocessor, an analog-to-digital converter, and a digital-to-analog converter needs each of these elements to perform the digital filtering function; if any one of the three elements fails, the system fails. One way of representing the series system is through the use of **reliability block diagrams.** The reliability block diagram can be thought of as a flow diagram from the input of the system to the output of the system. Each element of the system is a block in the reliability block diagram and, for the series system, the blocks are placed in series to indicate that a path from the input to the output is broken if one of the elements fails.

For example, the generalized reliability block diagram of a series system that contains N elements is shown in Fig. 4.4. Each of the N elements is required for the system to function correctly. The reliability of the series system can be calculated as the probability that none of the elements will fail. Another way to look at this is that the reliability of the series system is the probability that all of the elements are working properly.

Suppose we let $C_{iw}(t)$ represent the event that component C_i is working properly at time t, $R_i(t)$ is the reliability of component C_i at time t, and $R_{\text{series}}(t)$ is the reliability of the series system. Further suppose that the series system contains N series components as shown in Fig. 4.4. The reliability at any time t is the probability that all N components are working properly. In mathematical terms,

$$R_{\text{series}}(t) = P(C_{1w}(t) \cap C_{2w}(t) \cap \cdots \cap C_{Nw}(t))$$

Assuming that the events, $C_{iw}(t)$, are independent, we have

$$R_{\text{series}}(t) = R_1(t)R_2(t)\cdots R_N(t)$$

or

$$R_{\text{series}}(t) = \prod_{i=1}^{N} R_i(t)$$

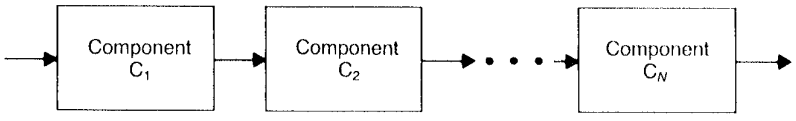

Fig. 4.4 The reliability block diagram of a series system—each element of the system must operate correctly for the system to operate correctly.

An interesting relationship exists in a series system if each individual component satisfies the exponential failure law. Suppose that we have a series system made up of N components, and each component i has a constant failure rate of λ_i. Also assume that each component satisfies the exponential failure law such that the reliability of each component is $R_i(t) = e^{-\lambda_i t}$. The reliability of the series system is given by

$$R_{\text{series}}(t) = e^{-\lambda_1 t} e^{-\lambda_2 t} \cdots e^{-\lambda_N t}$$

or

$$R_{\text{series}}(t) = e^{-\sum_{i=1}^{N} \lambda_i t} = e^{-\lambda_{\text{system}} t}$$

where $\lambda_{\text{system}} = \sum_{i=1}^{N} \lambda_i$ and corresponds to the failure rate of the system. In other words, the failure rate of a series system can be calculated by adding the failure rates of all the components that make up the series system.

As an example of a series system, consider the simple aircraft control system shown in Fig. 4.5. This system contains sensors that are used to measure the roll, pitch, and yaw positions of the aircraft; sensors to measure the crew's desired roll, pitch, and yaw positions; actuators that are used to control the roll, pitch, and yaw; and computers that perform the computations required of the flight control system. The computers receive the sensor values and supply the actuator commands over a serial data bus that connects the sensors, actuators, and the computers. A special high-speed data bus interconnects the computers for the purpose of data transfer among the computers. Each element of the system is required if the system is to perform correctly; there is no redundancy in the system. For example, the failure of any one sensor or computer renders the system unable to operate correctly.

The reliability block diagram of the flight control system is shown in Fig. 4.6. The reliability block diagram illustrates the series nature of the system. For simplicity, assume that all six sensors have the same reliability $R_s(t)$, each of the three actuators has the reliability, $R_{\text{act}}(t)$, and each computer has the reliability $R_c(t)$. Also, let the computer interconnection bus have the reliability $R_{\text{bus1}}(t)$ and the primary control bus have the reliability $R_{\text{bus2}}(t)$. By taking the product of the element reliabilities, we find that the reliability of the system is given by

$$R_{\text{system}}(t) = R_s^6(t) R_{\text{act}}^3(t) R_c^3(t) R_{\text{bus1}}(t) R_{\text{bus2}}(t)$$

Because the failure rates can be added in a series system to obtain the failure rate of the system, we can write

$$\lambda_{\text{system}} = 6\lambda_s + 3\lambda_{\text{act}} + 3\lambda_c + \lambda_{\text{bus1}} + \lambda_{\text{bus2}}$$

where λ_s is the failure rate of one sensor, λ_{act} is the failure rate of one actuator, λ_c is the failure rate of one computer, λ_{bus1} is the failure rate of the com-

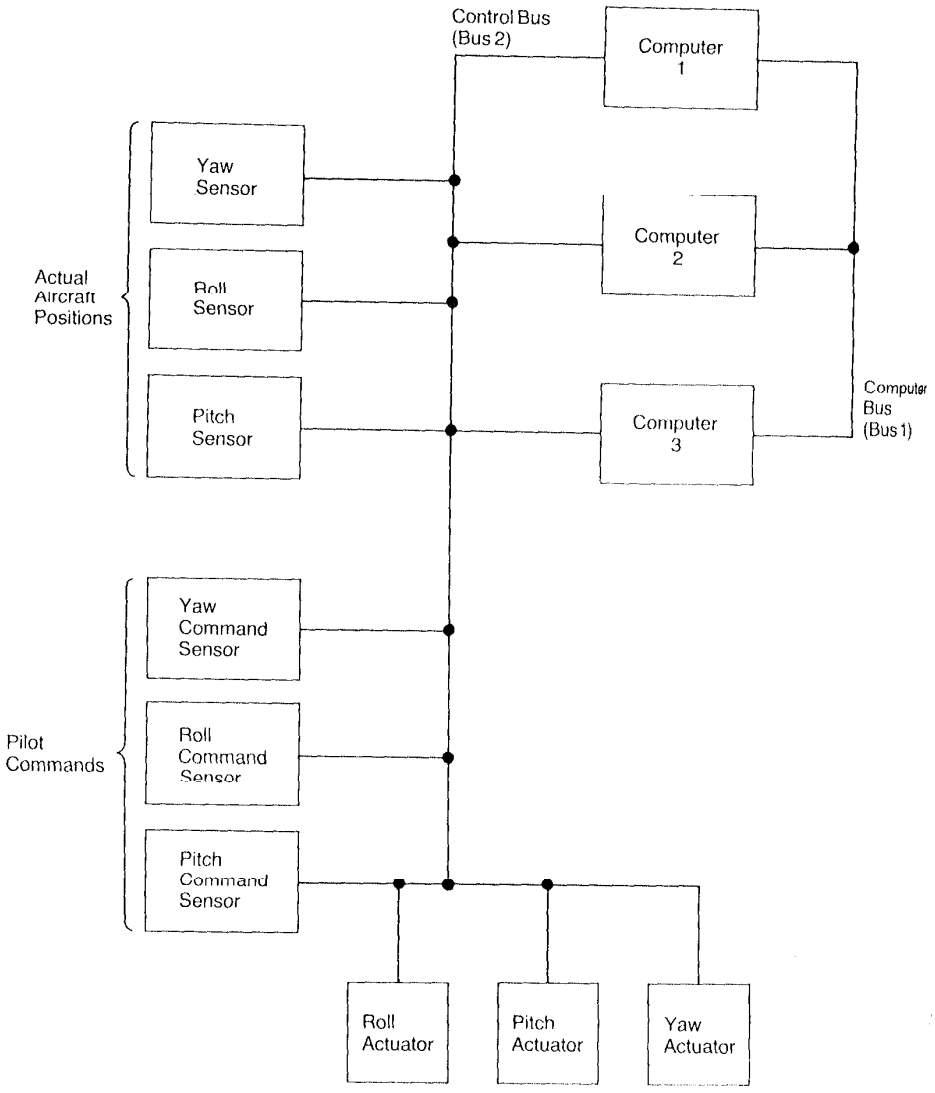

Fig. 4.5 An aircraft control system designed as a series system.

puter interconnection bus, and λ_{bus2} is the failure rate of the primary control bus. λ_{system} is the failure rate of the system. If the failure rates of the system are

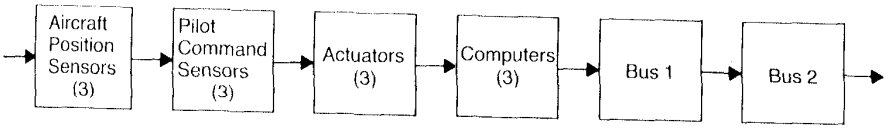

Fig. 4.6 The reliability block diagram of the system in Figure 4.5 illustrates the series nature of the system.

$$\lambda_s = 1 \times 10^{-6} \quad \text{failures per hour}$$
$$\lambda_{act} = 1 \times 10^{-5} \quad \text{failures per hour}$$
$$\lambda_c = 4 \times 10^{-4} \quad \text{failures per hour}$$
$$\lambda_{bus1} = 1 \times 10^{-6} \quad \text{failures per hour}$$
$$\lambda_{bus2} = 2 \times 10^{-6} \quad \text{failures per hour}$$

the system failure rate will be

$$\lambda_{system} = 1.239 \times 10^{-3} \quad \text{failures per hour}$$

The reliability after five hours for this system is approximately 0.995.

Parallel Systems

The distinguishing feature of the basic parallel system is that only one of N identical elements is required for the system to function. For example, many families have two or more cars when one is, in many cases, sufficient to meet the family's needs. The probability of having at least one car working can be determined by modeling the multiple-car family as a parallel system.

The reliability block diagram of the basic parallel system that contains N identical elements is shown in Fig. 4.7. As can be seen, a path exists in the reliability block diagram from input to output as long as one of the N identical elements remains operational. The unreliability of the parallel system can be computed as the probability that all of the N elements fail. Suppose that we let $C_{if}(t)$ represent the event that element i in the parallel system has failed at time t, $Q_{\text{parallel}}(t)$ be the unreliability of the parallel system, and $Q_i(t)$ be the unreliability of the i^{th} element. $Q_{\text{parallel}}(t)$ can be computed as

$$Q_{\text{parallel}}(t) = P(C_{1f}(t) \cap C_{2f}(t) \cap \cdots \cap C_{Nf}(t))$$

or

$$Q_{\text{parallel}}(t) = Q_1(t)Q_2(t)\cdots Q_N(t) = \prod_{i=1}^{N} Q_i(t)$$

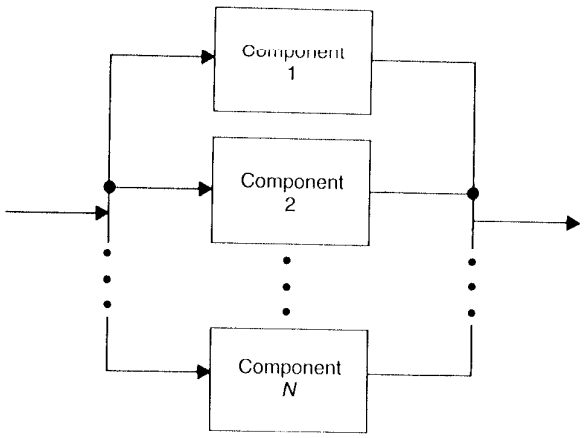

Fig. 4.7 The reliability block diagram of the parallel system—only one of N components must operate correctly for the system to operate correctly.

The reliability of the parallel system can now be computed because we know that the reliability and the unreliability must add to 1.0. Mathematically, we must have $R(t) + Q(t) = 1.0$ for any system. Consequently, we can write

$$R_{\text{parallel}}(t) = 1.0 - Q_{\text{parallel}}(t) = 1.0 - \prod_{i=1}^{N} Q_i(t) = 1.0 - \prod_{i=1}^{N} (1.0 - R_i(t))$$

Note that the equations for the parallel system assume that the failures of the individual elements that make up the parallel system are independent. For random hardware failures, the independence of failures is a good assumption; however, for failures that are the result of items such as external disturbances, the independence assumption is not very good. Therefore, the combinatorial modeling techniques are most often applied to the analysis of random failures in a system's hardware.

To analyze a system that has a parallel structure, consider the system shown in Fig. 4.8. The architecture of the system in Fig. 4.8 is commonly found in aerospace applications. The system consists of two identical computers, two identical interface units, two identical display devices, and two identical communication buses. The system requires that at least one of each unit work properly for the system to perform its functions. Once a particular unit has failed, it is assumed that the second unit of that type automatically assumes the functions of the failed unit.

One important point about the system of Fig. 4.8 is that it has both a series and a parallel structure. It is parallel in the sense that only one of the

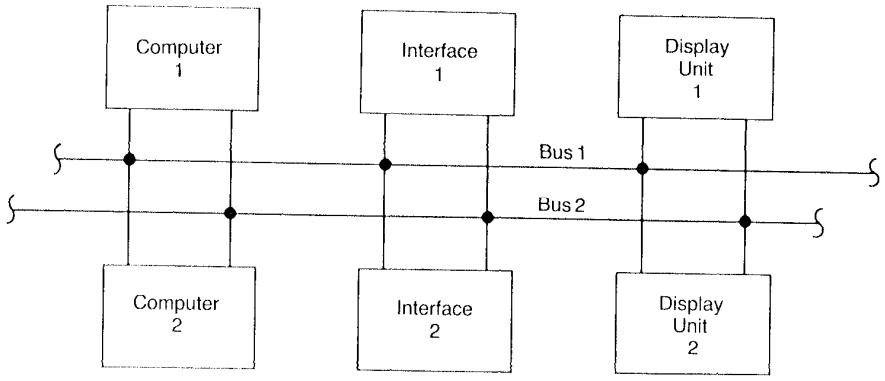

Fig. 4.8 An example computer system having a structure that is a combination of series and parallel.

two computers, for example, must function for the system to function. It is series in the sense that one computer, one interface unit, one display device, *and* one bus must operate for the system to operate. The reliability block diagram of the system of Fig. 4.8 is shown in Fig. 4.9. Note that a path from the input of the diagram to the output exists if and only if enough elements are functioning to allow the system to operate properly.

A reliability block diagram that contains both series and parallel structures can be reduced to a single series diagram by replacing each of the parallel portions of the system with an equivalent, single element that has the same reliability as the parallel structure. For example, we know from the

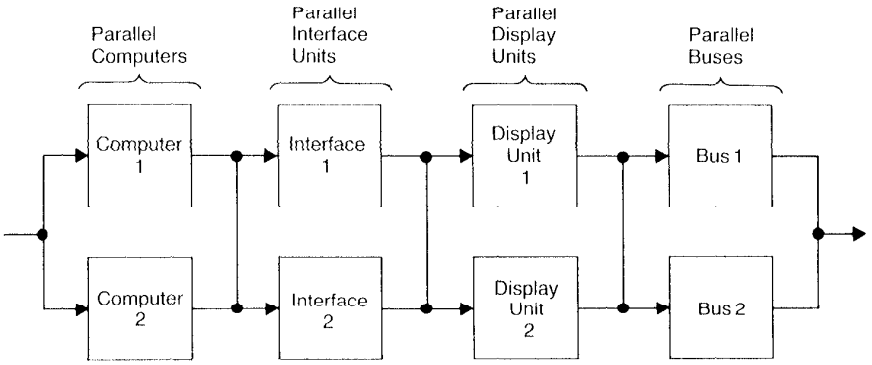

Fig. 4.9 Reliability block diagram of the series/parallel system of Fig. 4.8.

analysis of a parallel system that the parallel organization of the two computers in Fig. 4.9 has a reliability given by

$$1.0 - (1 - R_c(t))^2$$

where $R_c(t)$ is the reliability of one computer. Therefore, the parallel organization of the computers can be replaced by a single element having a reliability of $1.0 - (1 - R_c(t))^2$, as is illustrated in Fig. 4.10. The transformation of a parallel system into an equivalent series system is a very common technique used to reduce reliability block diagrams. The reliability block diagram of Fig. 4.9 can be reduced to a series diagram by applying the reduction concept to each parallel structure. The reduced block diagram is shown in Fig. 4.11 where $R_c(t)$ is the reliability of one computer, $R_{if}(t)$ is the reliability of one interface unit, $R_d(t)$ is the reliability of one display unit, and $R_b(t)$ is the reliability of one bus.

The reliability for the reduced block diagram of Fig. 4.11 can be written as

$$R_{\text{system}}(t) = [1 - (1 - R_c(t))^2][1 - (1 - R_{if}(t))^2][1 - (1 - R_d(t))^2] \cdot [1 - (1 - R_b(t))^2]$$

As an example, the reliability of the system after one hour given $R_c(1) = R_{if}(1) = R_d(1) = R_b(1) = .9$ will be

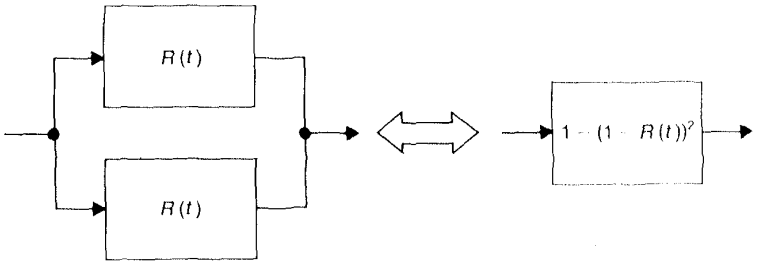

Fig. 4.10 A parallel system can be reduced to a series element with the proper reliability function.

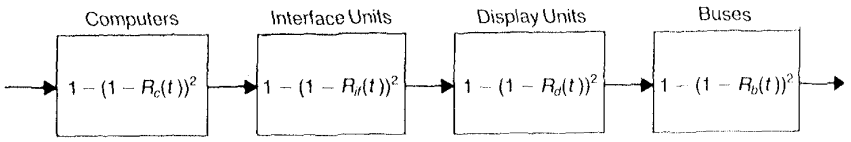

Fig. 4.11 Reduced reliability block diagram for the system of Fig. 4.8.

$R_{system}(1 \text{ hour}) = [1 - (1 - .9)^2][1 - (1 - .9)^2][1 - (1 - .9)^2][1 - (1 - .9)^2] = .96$

Now that we have several tools for investigating the reliability of systems, we can compare the reliability benefits that redundancy can offer. For example, the redundant system just analyzed had a reliability of 0.96 after one hour. The nonredundant system containing one computer, one display device, one bus, and one interface device has a reliability equivalent to the product of the individual element's reliabilities because the nonredundant system is a simple series system. Therefore, the nonredundant system has a reliability of 0.6561 after one hour. As is seen, the incorporation of redundancy has significantly improved the system's reliability.

Note, however, that redundancy does not always improve a system's reliability. Whether or not the reliability is improved depends on the amount of redundancy employed and the reliability of the elements used to construct the system, as well as other factors. For example, if each element of the redundant system in Fig. 4.8 has a reliability of 0.1 at the end of one hour, the redundant system has a reliability of 0.0013 at the end of one hour, and the nonredundant system has a reliability of 0.0001 at the end of one hour. We certainly would hope that elements with a reliability as poor as 0.1 would never be used in a system, but this example does show that the redundancy does not significantly improve the reliability. As we stated in the first chapter, reliability and fault tolerance are not one in the same. As we shall see when we begin to analyze more complex systems, the distinction between fault tolerance and reliability becomes even clearer.

4.3.2 Fault Coverage and Its Impact on Reliability

As defined earlier, fault coverage is a measure of a system's ability to recover from faults. For example, a system with redundant computers that requires reconfiguration before the redundancy can be used depends heavily on good fault coverage. During the analysis of the parallel system, we assumed that the fault coverage was perfect; if we had three computers and needed only one to operate, the reliability was calculated solely as the probability that one of the three computers was operational. Unfortunately, the assumption of perfect fault coverage does not consider that the system may not be able to use the redundancy because it cannot identify that a unit is faulty, remove that faulty unit, and replace it with a fault-free one.

To illustrate the problem, consider a simple parallel system consisting of two identical modules and having the reliability block diagram shown in Fig. 4.12. Assume that module 1 is the primary module and that module 2 is a spare module that is switched on-line in the event of failure of module 1. In other words, the system uses the concept of standby sparing. Under ideal circumstances, the standby sparing system functions correctly as long as one of the two modules functions correctly. In reality, however, the failure

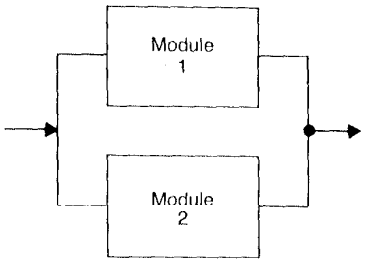

Fig. 4.12 Reliability block diagram of a simple parallel system to illustrate the impact of fault coverage.

of the primary module (module 1, in this case) must be detected and correctly handled before the second module can be used. In other words, the parallel system with two modules functions correctly as long as one of the following two conditions exist:

1. Module 1 is functioning correctly.
2. Module 2 is functioning correctly, module 1 has failed, and the failure was detected and appropriately handled.

The probability that one of these two events will exist can be written in terms of the reliabilities of the modules and the fault coverage as

$$R_{\text{system}}(t) = R_1(t) + (1 - R_1(t))C_1R_2(t)$$

where C_1 is the fault coverage associated with module 1, $R_1(t)$ is the reliability of module 1, and $R_2(t)$ is the reliability of module 2. The reliability equation enumerates all of the working states of the system. If the reliabilities and the coverage factors of the two modules are identical, the reliability expression reduces to

$$R_{\text{system}}(t) = R(t) + R(t)C(1 - R(t))$$

where $R(t)$ is the reliability of one module and C is the fault coverage. Note that if the fault coverage C, is 1.0, the reliability expression reduces to

$$R_{\text{system}}(t) = 2R(t) - R^2(t) = 1 - (1 - R(t))^2$$

which is the reliability of the perfect parallel system. Also note that if the fault coverage is 0.0, the reliability expression reduces to simply the reliability of one module; therefore, the primary module must function correctly for the system to function correctly.

It is interesting to study the impact of the fault coverage in the parallel system with two modules. Figure 4.13 shows the reliability of the system as a function of fault coverage for a module reliability of R. Note that the

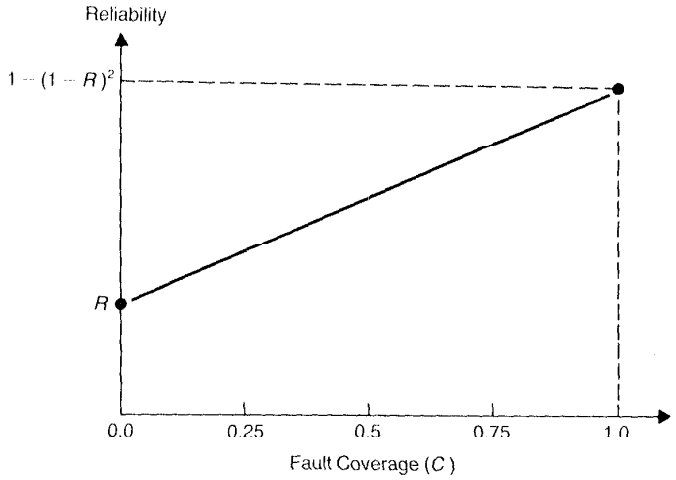

Fig. 4.13 Reliability versus fault coverage for a parallel system with two modules and using standby sparing. Each module has a reliability of R and a fault coverage of C.

reliability of the parallel system with two modules is a linear function of the coverage factor. At a coverage of 1.0, the reliability is $1 - (1 - R)^2$, which is the reliability of the perfect parallel system. At a coverage of 0.0, the reliability is R, which is simply the probability that the primary module will not fail.

One important point about the above analysis is that the failure of the second module is unimportant unless it has replaced the first module. In other words, module 1 is the primary module and as long as it functions correctly, the system functions correctly, even if module 2 fails. In many systems, this may not be true. For example, consider a duplex system that performs comparisons between the two modules as one form of fault detection. Once a fault is detected, the two modules go into more detailed fault analysis routines, often called self-diagnostics, in an attempt to identify which of the two modules is faulty. If the faulty module is identified, the system continues to operate with the one fault-free module, and the comparison mechanism is disabled. If the faulty module cannot be identified, the system discontinues its operation. Therefore, an undetected fault in either of the modules causes the system to fail and must be accounted for in the reliability analysis.

Consider once again the parallel system with two modules as shown in Fig. 4.12, but now perform comparisons between the two modules as one means of fault detection. Assume for now that the comparison is perfect and

detects all faults. Once the comparison process detects a fault, the system implements self-diagnostics to attempt to determine which of the two modules is faulty. If the fault can be located successfully, the fault-free module begins to perform the functions of the system. The system functions correctly as long as both modules work or the fault has been detected and handled correctly. The reliability of the system can be written by enumerating the working states of the system.

In mathematical terms, we have

$$R_{system}(t) = R_1(t)R_2(t) + R_1(t)(1 - R_2(t))C_2 + (1 - R_1(t))C_1R_2(t)$$

where $R_1(t)$ is the reliability of module 1, $R_2(t)$ is the reliability of module 2, C_1 is the fault coverage of the self-diagnostics of module 1, and C_2 is the fault coverage of the self diagnostics of module 2. If the reliabilities and fault coverages of the two modules are identical, the reliability reduces to

$$R_{system}(t) = R^2(t) + 2R(t)C(1 - R(t))$$

For perfect fault coverage, we obtain the same as before; that is, the system has the reliability of the perfect parallel system. If the fault coverage is 0.0, the system has a reliability of $R_{system}(t) = R^2(t)$, which is simply the probability that both modules operate correctly. Figure 4.14 shows the reliability of the system as a function of fault coverage.

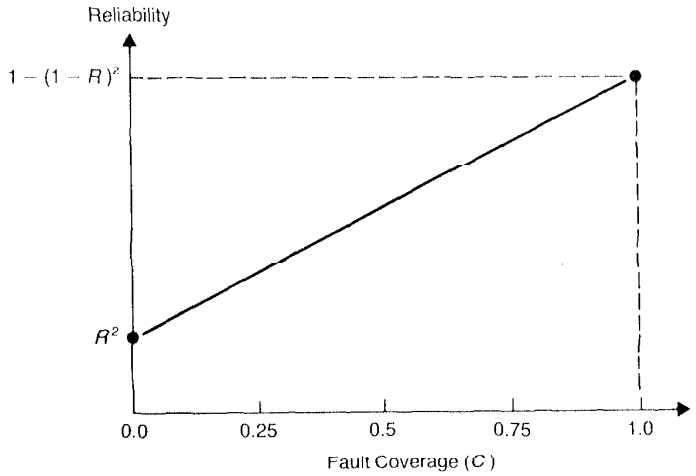

Fig. 4.14 Reliability versus fault coverage for a parallel system with two modules and using comparisons between the two modules. Each module has a reliability of R and a fault coverage of C.

4.3.3 *M*-of-*N* Systems

***M*-of-*N* systems** are a generalization of the ideal parallel system. In the ideal parallel system, only one of N modules is required to work for the system to work. In the M-of-N system, however, M of the total of N identical modules are required to function for the system to function. A good example is the TMR configuration where two of the three modules must work for the majority voting mechanism to function properly. Therefore, the TMR system is a 2-of-3 system.

Consider as an example the TMR system. As seen in the previous sections, we can write the reliability of a system by enumerating all of the possible states in which the system can be functional. Suppose that we have a TMR system with modules 1, 2, and 3 connected in a majority voting arrangement. As long as two of the three modules are functioning correctly, the system will perform correctly. Ignoring the reliability of the voter, the reliability of the TMR system can be written as

$$\begin{aligned} R_{\text{TMR}}(t) = {} & R_1(t)R_2(t)R_3(t) + R_1(t)R_2(t)(1 - R_3(t)) \\ & + R_1(t)(1 - R_2(t))R_3(t) + (1 - R_1(t))R_2(t)R_3(t) \end{aligned}$$

where $R_i(t)$ is the reliability of the i^{th} module. If $R_1(t) = R_2(t) = R_3(t) = R(t)$, the reliability of the TMR system reduces to

$$R_{\text{TMR}}(t) = R^3(t) + 3R^2(t)(1 - R(t)) = 3R^2(t) - 2R^3(t)$$

Now that we have the expression for the reliability of the TMR system, it is interesting to examine the reliability improvements that can be obtained through the use of TMR. Figure 4.15 shows a plot of the reliability of a TMR arrangement as a function of the reliability of the modules that compose the TMR system. In other words, Fig. 4.15 simply shows a plot of the equation $R_{\text{TMR}} = 3R^2 - 2R^3$ versus R. As can be seen, there is a point at which the reliability of the TMR system and the reliability of the single module cross. The crossover point is easily found by setting the reliability of the TMR system equal to the reliability of the single module and solving the resulting quadratic equation.

In mathematical terms, we have

$$R_{\text{TMR}} = 3R^2 - 2R^3 = R$$

or

$$3R - 2R^2 = 1$$

which implies the quadratic equation

$$R^2 - \frac{3}{2}R + 0.5 = 0$$

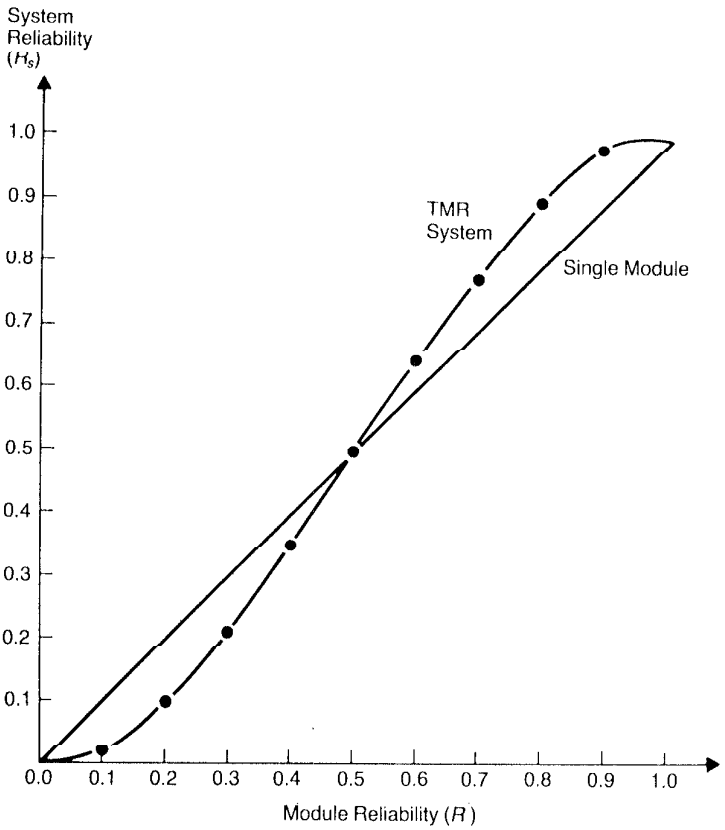

Fig. 4.15 Comparison of the reliability of a TMR system composed of three identical modules with the reliability of a single module.

The two solutions to the quadratic equation are 0.5 and 1.0, which implies that the reliability of the TMR system is equal to that of the corresponding nonredundant system when the reliability of the single module is 0.5 or the module is perfect ($R = 1$).

This further illustrates a point that we made when we defined fault tolerance and reliability. A system can be tolerant of faults and still have a low reliability. For example, a TMR system constructed from modules that have individual reliabilities of 0.5 can tolerate a fault in one of those modules, but the reliability of the TMR system is the same as the reliability of a single module. Conversely, a system can achieve a high reliability without being fault tolerant. Certainly, a system that consists of a perfect module will have the highest possible reliability but will not possess, or need, the at-

tribute of fault tolerance. This is, of course, an unrealistic example, but, in general, as the reliability of the components of a system increases, the reliability of the system also increases. It is possible for the reliability of a nonredundant system to approach that of a redundant system constructed from the same modules. The nonredundant system, however, will not be fault tolerant.

In many cases, we have systems that are of the M-of-N structure but are not TMR; the general NMR system is a good example. In general, if there are N identical modules and M of those are required for the system to function properly, the system can tolerate $N - M$ module failures. The expression for the reliability of an M-of-N system can be written as

$$R_{M\text{-of-}N}(t) = \sum_{i=0}^{N-M} \binom{N}{i} R^{N-i}(t)(1 - R(t))^i$$

where

$$\binom{N}{i} = \frac{N!}{(N-i)!\,i!}$$

For example, the TMR system reliability is given by

$$R_{\text{TMR}}(t) = \sum_{i=0}^{1} \binom{3}{i} R^{3-i}(t)(1 - R(t))^i$$

which reduces to

$$R_{\text{TMR}}(t) = 3R^2(t) - 2R^3(t)$$

which is identical to the expression derived earlier.

4.3.4 Markov Models

The primary difficulty with the combinatorial models is that many complex systems cannot be modeled easily in a combinatorial fashion. The reliability block diagrams can be extremely difficult to construct, and the resulting reliability expressions are often very complex. In addition, the fault coverage that we have seen to be extremely important in the reliability of a system is sometimes difficult to incorporate into the reliability expression in a combinatorial model. Finally, the process of repair that occurs in many systems is very difficult to model in a combinatorial fashion. For these reasons, we often use **Markov models.**

The purpose of the presentation in this text is not to delve into the mathematical details of Markov models but to understand how to use Markov models. For more explicit mathematical details, refer to the references ([Shooman 1968] and [Trivedi 1982]). The discussions here will provide sufficient mathematical background to apply the Markov model but will not pursue various techniques for solving the models.

The two main concepts in the Markov model are the **system state** and the **state transition.** The state of a system represents all that must be known to describe the system at any given instant of time. For reliability models, each state of the Markov model represents a distinct combination of faulty and fault-free modules. For example, suppose we have a TMR system with three identical computers in a majority voting arrangement with a perfect voter. We can define the state of this system as $S = (S_1, S_2, S_3)$ where $S_i = 1$ if module i is fault free and $S_i = 0$ if module i is faulty. The TMR system has eight distinct states in which it can operate: (000), (001), (010), (011), (100), (101), (110), and, (111). Each state represents a unique combination of faulty and fault-free modules within the system. For TMR, we know that at least two of the modules must be fault free for the system to operate correctly. Therefore, the states (000), (001), (010), and (100) represent states in which the system has ceased to function correctly. The remaining states are those in which the system is functioning correctly.

The state transitions govern the changes of state that occur within a system. As time passes and failures and reconfigurations occur, the system goes from one state to another. For example, if the TMR system starts its operation in state (111) and at some time t module 1 fails, the system transitions to state (011). The state transitions are characterized by probabilities such as the probability of failure, fault coverage, and the probability of repair.

As an example of the state transitions that can occur, consider the TMR system. We have already defined the states that can exist in the system; now let us define the transitions that can occur. We construct our transitions using several assumptions. First, we assume that the system does not contain repair. In other words, once a module has failed, it remains failed permanently. Second, we assume that only one failure will occur at a time. In a TMR system, the single failure assumption implies that the system cannot go directly from the state corresponding to all modules operating correctly to a state that corresponds to the system having failed. In other words, no single failure can cause the complete TMR system to fail. Finally, we assume that the system starts in the perfect state (111) where all of the system's modules are operating correctly.

The state diagram that results for the TMR system is shown in Fig. 4.16. As can be seen, the system begins in state (111) and, upon the first module failure, transitions to state (110), (101), or (011), depending on whether module 1, 2, or 3 is the module that fails. Note that the transition exists for the module to remain in a state if a module failure does not occur. The state diagram shown in Fig. 4.16 is analogous to the state diagram of a synchronous digital circuit. When some event, a module failure in the case of the reliability model and the occurrence of a clock signal in the case of a synchronous machine, occurs, the system transitions from one state to another.