The failure rates of the individual components are:

$$\lambda_i = 10^{-6} \text{ failures per hour}$$
$$\lambda_e = 3.5 \times 10^{-4} \text{ failures per hour}$$
$$\lambda_d = 5 \times 10^{-5} \text{ failures per hour}$$
$$\lambda_m = 10^{-6} \text{ failures per hour}$$
$$\lambda_f = 10^{-6} \text{ failures per hour}$$

Using the given failure rate data results in a reliability function for the system of

$$R(t)_{\text{system}} = e^{-4.56\times 10^{-4}t}$$

A plot of the reliability function versus time is shown in Fig. 4.32. Figure 4.32 shows the probability that the control system will *not* fail as a function of time. For example, if no maintenance is performed on the system during one month of continuous operation, the probability that a failure will not occur within that month is approximately 0.72. Stated differently, the user has a probability of 0.28 of the system failing within a month of continuous operation.

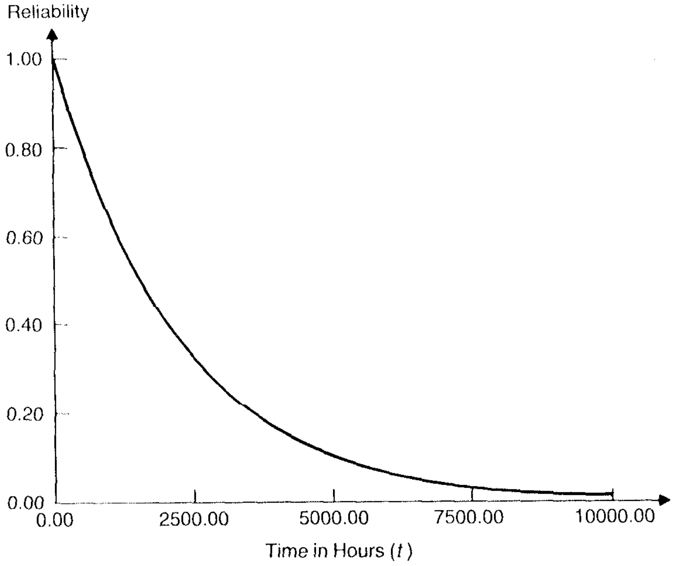


Fig. 4.32 Reliability versus time for the nonredundant system ($R(t) = e^{-4.56\times 10^{-4}t}$).

The times shown in Fig. 4.32 are actual operation times. It is assumed that the system has a zero probability of failing while it is not operating. If we assume that the system is used approximately four hours per day, the reliability figures indicate that the user will have about 382 days of use before the probability of failure rises above 0.5.

Another way of looking at the reliability of the system is through the mean time to failure (MTTF). Recall that the MTTF is the expected time that elapses before the first failure occurs, and it is defined in terms of the reliability function as

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

where $R(t)$ is the reliability function. For a simple system with an exponential reliability function, the MTTF can be written in terms of the failure rate as

$$\text{MTTF} = \frac{1}{\lambda}$$

For the system under consideration with the failure rate of $\lambda_{\text{system}} = 4.56 \times 10^{-4}$, the MTTF is approximately 2192 hours. In other words, the user can expect, on the average, to operate the system for only 2192 hours before it fails while in use.

The second crucial issue of the analysis is availability. The availability, as mentioned previously, is the percentage of time that the system is available to provide its services. Availability varies as a function of time, but, as we have seen earlier, a steady-state value can be computed using the failure rate and the repair rate of the system. The steady-state availability is defined in terms of the mean time to failure (MTTF) and the mean time to repair (MTTR). As defined previously, the MTTF is the average time that elapses before the first failure occurs. The MTTR is the average time that elapses before the average failure can be repaired and the system made operational once again. The steady-state availability A_{ss} can be written as

$$A_{\text{ss}} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}}$$

The MTTR and the MTTF are related to the repair rate and the failure rate, respectively, as

$$\text{MTTF} = \frac{1}{\lambda}$$

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

Therefore, A_{ss} can be written as

$$A_{ss} - \frac{1}{1 + \frac{\lambda}{\mu}}$$

The steady-state availability is an estimate of the probability that the system is going to operate correctly when requested to do so. Using the system failure rate of 4.56 × 10^{-4}, Fig. 4.33 shows the steady-state availability of the system as a function of the MTTR. If one week is required, on the average, for repair then the steady-state availability is slightly higher than 0.92.

Unfortunately, the designers of the system in the present example have little control over the repair rate. The repair rate depends on the capabilities of the user to diagnose the problem, or the proximity of a facility where qualified personnel are employed to aid individuals in repairs. If the user is forced to get the manufacturer to repair the system, the repair period could easily extend beyond a week, particularly if the controller or other component must be mailed back to the factory for diagnosis and repair. To over-

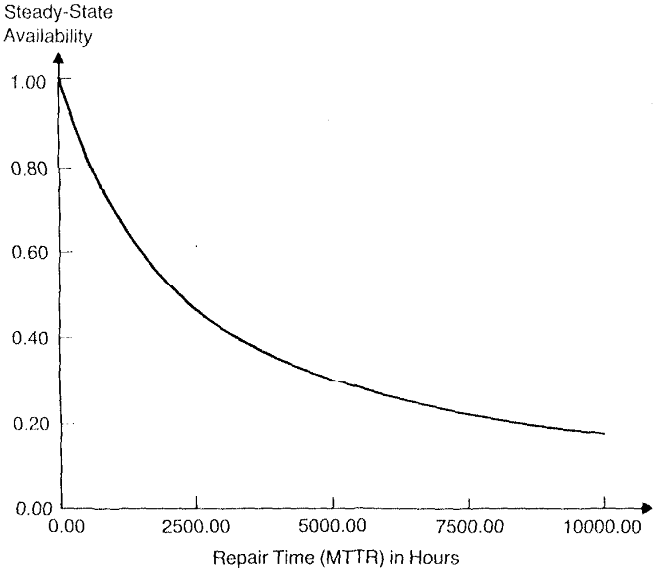


Fig. 4.33 *Steady-state availability versus the repair time (MTTR) in hours for* the nonredundant system ($\lambda = 4.56 \times 10^{-4}$ failures/hour).

come such problems, many users try to accumulate spare parts of their own such that the system can be operated while the repairs are being made. This is an expensive technique that could possibly be overcome through the development of designs that possess lower system failure rates or ones that use built-in redundancy to accommodate faults until the faulty components can be replaced. Figure 4.34 shows the improvement in availability as the failure rate is improved for a fixed repair rate of one repair per week.

Now we wish to examine the improvements that can be obtained through the use of redundancy to achieve some level of fault detection or fault tolerance. Fault tolerance can improve the reliability by preventing faults from resulting in a failure of the system. For example, if a power supply fluctuation occurs, the controller can be designed to automatically reset, prevent erroneous values from being sent to the motor drive circuits, and recover from the condition. In addition, conditions such as a run-away processor can be detected and a recovery implemented before catastrophic effects result.

Availability can be improved by using redundancy to keep the system functioning while repairs are made. For example, duplicate processors can

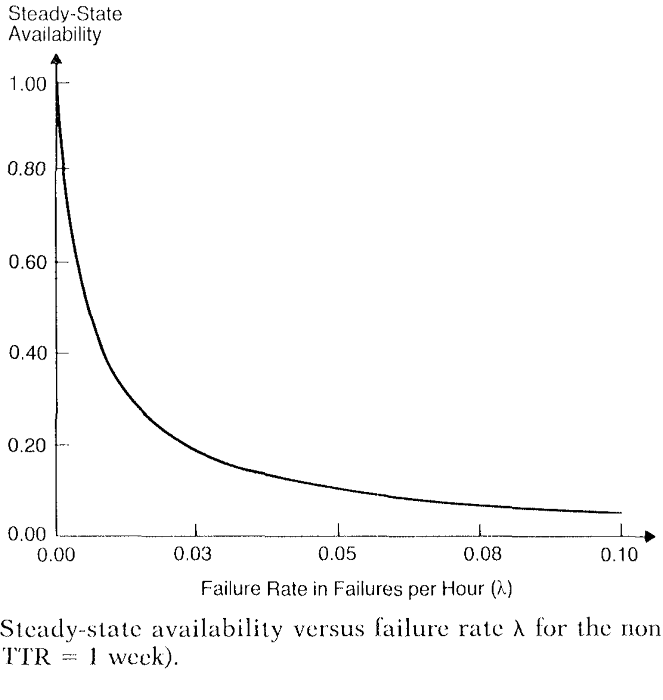


Fig. 4.34 Steady-state availability versus failure rate λ for the nonredundan system (MTTR = 1 week).

be incorporated into the controller; when the failure of one processor occurs, the second processor assumes the control functions until the first processor is repaired or replaced. Now, instead of the system being inoperative while being repaired, the redundant unit can perform the necessary computations and keep the system functioning properly. The necessary reconfiguration can be performed automatically to prevent the user from being required to take any action.

Safety can be improved through the incorporation of features that provide for the fail-safe operation of the system. For example, if a fault occurs, a mechanism must be provided to detect the existence of the fault and to prevent the fault from resulting in an undesired response from the control system.

Because of several practical constraints, it is not feasible to introduce redundancy into the system at any point other than the electronic controller. The motors, motor drives, and the sensors are too big and expensive to replicate. The manufacturer of the system would like to determine the advantage of incorporating redundancy into only the electronic controller.

The architecture of the candidate fault-tolerant controller is shown in Fig. 4.35. The system is completely dual redundant and uses an architecture similar to that of the Tandem NonStop computer system [Katzman 1982]. The processors both perform the same computations, and each is capable of functioning as the controller for the system. Various fault detection features are incorporated into the system to provide fault coverage and reconfiguration capability. At this point in the analysis, we will assume that the redundant system implements a reconfigurable duplication approach to fault detection and reconfiguration. In other words, the processors compare their results as well as use internal diagnostics to perform fault detection. Either processor can perform the functions of the system as long as the faulty processor is detected and removed from the system.

The improvements that can be obtained in the reliability and availability of the system can be assessed by analyzing the redundant controller. The reliability block diagram of the redundant system is shown in Fig. 4.36. By comparing Fig. 4.36 with the reliability block diagram of the nonredundant system shown in Fig. 4.31, we can see that the only difference lies in the controller electronics. As discussed earlier, the controller electronics are a weak link in the system, so improvements in the overall reliability and availability should be obtained by improving this weak link.

The reliability of the redundant system can be written by analyzing the combination of the series and parallel systems contained in the reliability block diagram of Fig. 4.36. The reliability of the system of Fig. 4.36 can be written as

$$R_{\text{system}}(t) = R_i(t)R_p(t)R_d(t)R_m(t)R_f(t)$$

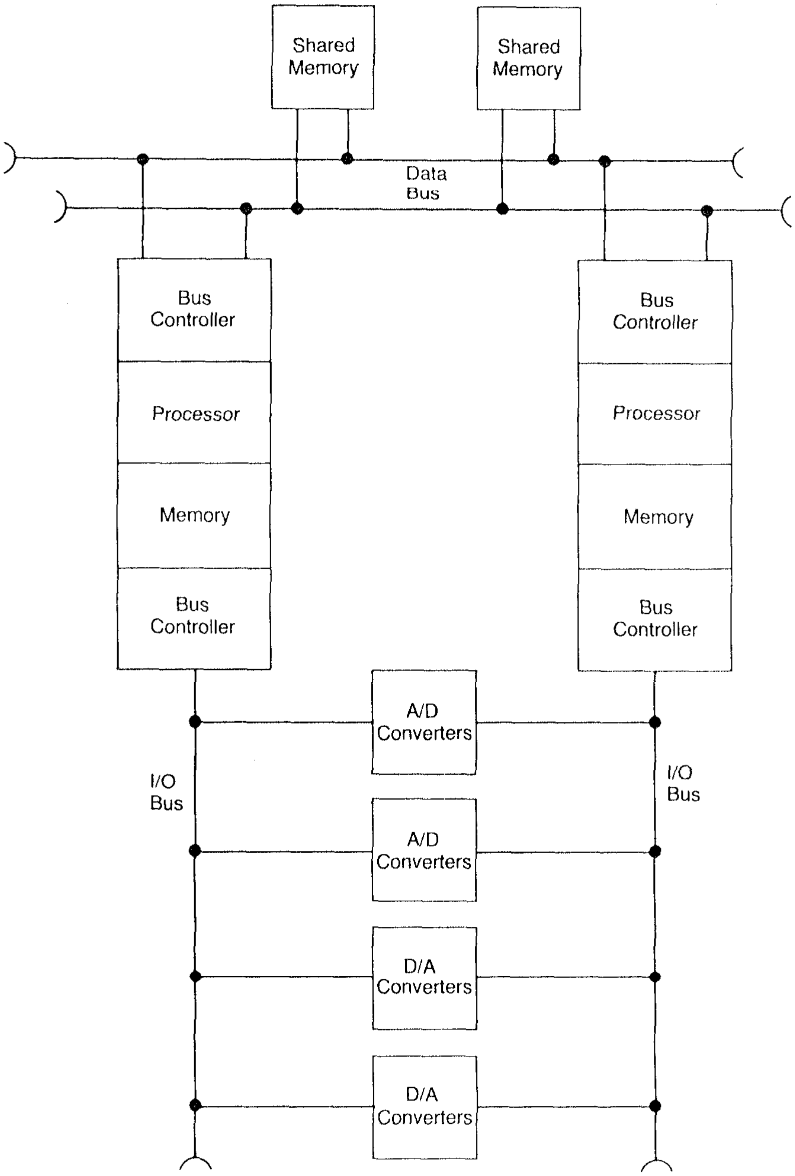


Fig. 4.35 Dual redundant architecture for the electronic controller.

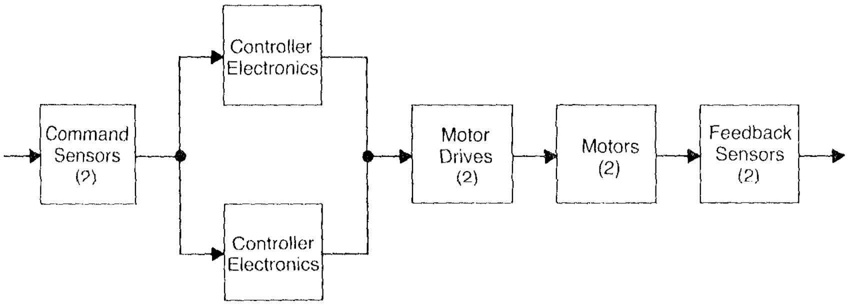


Fig. 4.36 Reliability block diagram of the control system with duplicated controller electronics.

where $R_i(t)$ is the reliability of the series combination of the two input sensors, $R_p(t)$ is the reliability of the parallel combination of the controller electronics, $R_d(t)$ is the reliability of the series combination of the two motor drives, $R_m(t)$ is the reliability of the series combination of the two motors, and $R_f(t)$ is the reliability of the series combination of the two feedback sensors.

The most interesting aspect of the parallel combination of the controller electronics is the fault coverage. We know that the fault coverage plays an important role in the reliability of the parallel system and the overall reliability of the control system. Using the techniques of this chapter, the reliability $R_p(t)$ can be written as

$$R_p(t) = R_e(t)R_e(t) + 2CR_e(t)(1 - R_e(t))$$

vhere $R_e(t)$ is the reliability of the nonredundant controller electronics.

Since we are assuming that each element of the control system obeys the exponential failure law, the reliability of the complete system can be written as

$$R_{\text{system}}(t) = e^{-\lambda_1 t}[e^{-2\lambda_e t} + 2Ce^{-\lambda_e t}(1 - e^{-\lambda_e t})]$$

where

$$\lambda_1 = 2\lambda_i + 2\lambda_d + 2\lambda_m + 2\lambda_f$$

and λ_e is the failure rate of the nonredundant electronics of the controller. The term C is the fault coverage. If $C = 1$, the system can recover from all faults that can possibly occur. As we have seen earlier in this chapter, the coverage factor can have a substantial impact on the reliability of the system.

Assume that a practical coverage factor is approximately 0.95. For a fault coverage of 0.95, Figure 4.37 compares the reliability of the redundant and the nonredundant systems. For short periods of time, the reliability improvements achieved through the use of redundancy are not overwhelming. However, as time increases, therefore increasing the probability of a failure occurring, the reliability improvements become very significant. For example, at 500 hours, the reliability of the redundant system is 0.9184, whereas that of the nonredundant system has fallen below 0.8. As time further increases, the reliability improvements diminish because the redundancy increases the amount of hardware that can fail.

An interesting way to see clearly the improvements obtained in the redundant system is to look at the time that can elapse before each system's reliability falls below a certain value. In other words, we look at the mission time of the system. For example, the reliability of the nonredundant system falls below 0.9 at approximately 230 hours. The reliability of the redundant system, however, does not fall below 0.9 until about 600 hours. Therefore, the redundancy has increased the time that the reliability remains above 0.9 by a factor of 2.6.

The availability is also improved by the incorporation of redundancy and fault detection techniques into the design of the controller. To deter-

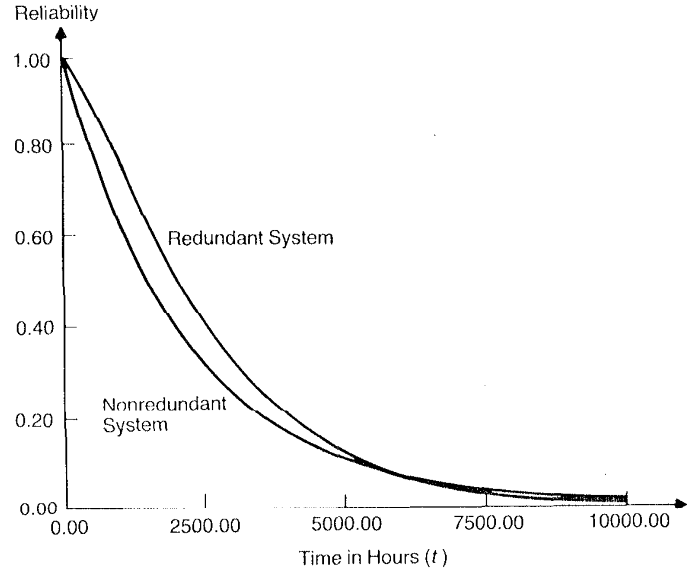


Fig. 4.37 Reliability of the redundant and nonredundant systems versus time.

mine the availability of the redundant system, we first need to compute the MTTF of the redundant system. The MTTF of the redundant system can be calculated by integrating the reliability function of the redundant system. In other words, we have

$$\mathrm{MTTF}_{\mathrm{redundant}} = \int_0^{\infty} [e^{-\lambda_1 t}[e^{-2\lambda_e t} + 2Ce^{-\lambda_e t}(1 - e^{-\lambda_e t})]]\,dt$$

Evaluating the integral for a coverage factor of 0.95, we find that the MTTF of the redundant system is approximately 3113 hours. Recall that the MTTF of the nonredundant system was approximately 2192 hours, so the redundancy has improved the MTTF by a factor of approximately 1.42. For a coverage factor of 0.95, Fig. 4.38 compares the availability of the redundant and the nonredundant systems.

Now we want to examine some of the specifics of implementing the redundant controller. We have provided some convincing information concerning the benefits of redundancy, but we have not completed the decisions concerning the specifics of the redundant implementation.

Suppose the dual-redundant system can be designed to function in one of two ways. The first is the reconfigurable duplication scheme that we have

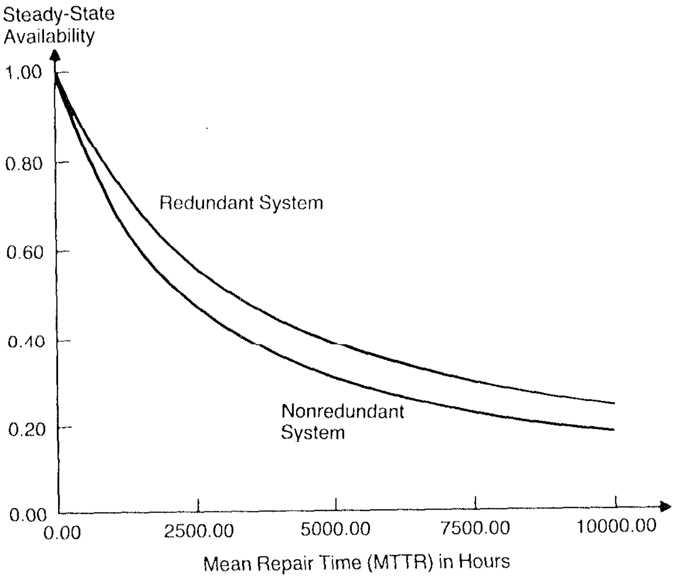


Fig. 4.38 Comparison of steady-state availability as a function of the MTTR.

just considered. In reconfigurable duplication, both processors operate in synchrony while performing the same computations. The results produced by the two processors are compared and any discrepancy is indicative of a fault in some component of the dual-redundant system. Each processor also runs self-diagnostic routines that attempt to locate the source of a fault. If the faulty processor is located, the system is reconfigured to construct a completely functional system using the fault-free processor.

The concept of reconfigurable duplication has been frequently used in industrial applications. Examples include the Bell Electronic Switching [Toy 1978] System (ESS), the COMTRAC railroad traffic control computer [Ihara et al. 1978], the Agusta A129 flight control computer system [Johnson and Julich 1985], and the AXE telephone switching control unit [Ossfeldt and Jonsson 1980]. Reconfigurable duplication is popular because it requires relatively little redundancy and is capable of significantly improving the reliability of a system.

The second possible mode of operation is the use of a standby spare. In standby sparing, one of the two processors is selected as the online unit that performs all of the computations for the system as long as it remains fault free. Concurrent fault detection routines are executed in the online processor to detect faults. If, and only if, a fault is detected in the online unit, the spare processing unit is brought on-line to assume the functions of the failed processor. In many cases, the spare processor might even remain unpowered until a fault is detected in the online unit.

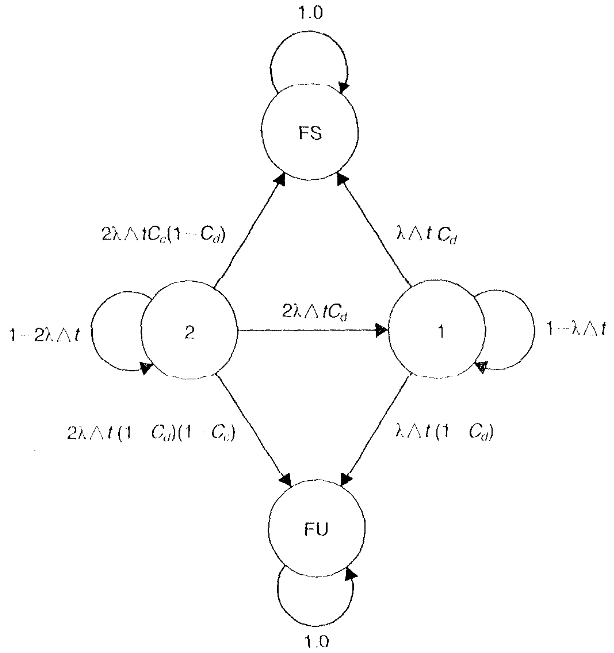
Many space applications have selected the standby sparing approach because of the potential to reduce overall power consumption by keeping the spare unpowered until needed. Examples include the Self-Test And Repair (STAR) processor [Avizienis et al. 1971] and the United Data System (UDS) architecture [Rennels 1978].

During the early design stages of the fault-tolerant controller, certain tradeoffs must be performed on the type of redundancy that will be employed. Therefore, both the reconfigurable duplication and the standby sparing concepts need to be analyzed. The subsequent material is a description of that analysis.

The reliability analysis of the two candidate approaches will be conducted using Markov models. The Markov models allow the flexibility to consider various factors including the repair process. More importantly, the Markov model is easy to construct and solve. One of the factors that is considerably important is the fault coverage.

In the reconfigurable duplication technique, there are actually two fault coverages to consider. The first is the probability that the comparison process will detect the existence of a fault. The second is the probability that the faulty processor can be located and the necessary reconfiguration performed. In the standby sparing approach, only one coverage factor is involved: the probability that the fault will be detected and the spare will successfully replace the online unit.

The Markov model of the reconfigurable duplication system is shown in Fig. 4.39. The model contains four states that represent all configurations of the system. State 2 represents the case where both of the processing modules are operating correctly. Sometimes called the perfect state, this is the state in which the system initially begins operation. State 1 represents the system in which one of the two processing modules has failed and the problem has been successfully detected and appropriately handled. While in state 1, the system has one faulty processor and one fault-free processor, and the fault-free processor is performing the functions of the system. State *FS* is the fail-safe state. While in the fail-safe state, the system has ceased to



State	Definition
2	Both modules are fault free.
1	One module fault free and one faulty (detected).
FS	System has failed in a safe manner.
FU	System has failed in an unsafe manner.

Fig. 4.39 Four-state Markov model of reconfigurable duplication.

perform its functions, but it has done so in an entirely safe manner. In other words, the system has failed, but the problem has been detected and handled in a safe and effective manner. The final state, state *FU*, is the failed unsafe state. When in state *FU*, the system has failed and the problem has not been handled in a manner that guarantees the safe operation of the system. When the system is in state *FU*, unexpected and perhaps unsafe operations can result.

The transitions illustrated on the diagram of Fig. 4.39 can be explained as follows. The transition from state 2 to state 1 can occur if and only if one of the originally fault-free processors fails and the failure is detected by the self-diagnostics. If the failure is not detected by the self-diagnostics, the system cannot remain operational regardless of whether the comparison process detects the problem. If the comparison process detects the fault and the self-diagnostics do not, the system transitions from state 2 to state *FS* since the system will be safe but will not be able to continue to operate. The transition from state 2 to state *FU* will occur if neither the comparison process nor the self-diagnostics detects the fault. In that event, the system will have failed in an unsafe manner. The transitions out of state 1 are the result of the last fault-free processor becoming faulty. At this point, the self-diagnostics are the only mechanism available for the fault detection. If the self-diagnostics detect the problem, the system transitions to state *FS*. If, however, the problem is not detected, the system transitions to state *FU*.

The equations for the Markov model of the reconfigurable duplication system can be written in matrix form as

$$\mathbf{P}_{rd}(t + \Delta t) = \mathbf{T}_{rd}\mathbf{P}_{rd}(t)$$

where each element of $\mathbf{P}_{rd}(t)$is the probability of being in the corresponding state at the time t, each element of $\mathbf{P}_{rd}(t + \Delta t)$ is the probability of being in the corresponding state at the time $t + \Delta t$, and $\mathbf{T}_{rd}$ is the state transition matrix. Each of these quantities can be written as

$$\mathbf{P}_{rd}(t + \Delta t) = \begin{bmatrix} p_2(t + \Delta t) \\ p_1(t + \Delta t) \\ p_{FS}(t + \Delta t) \\ p_{FU}(t + \Delta t) \end{bmatrix}$$

$$\mathbf{T}_{rd} = \begin{bmatrix} 1 - 2\lambda\Delta t & 0 & 0 & 0 \\ 2\lambda\Delta t C_d & 1 - \lambda\Delta t & 0 & 0 \\ 2\lambda\Delta t C_c(1 - C_d) & \lambda\Delta t C_d & 1 & 0 \\ 2\lambda\Delta t(1 - C_d)(1 - C_c) & \lambda\Delta t(1 - C_d) & 0 & 1 \end{bmatrix}$$

$$\mathbf{P}_{rd}(t) = \begin{bmatrix} p_2(t) \\ p_1(t) \\ p_{FS}(t) \\ p_{FU}(t) \end{bmatrix}$$

The solution to the matrix equation of the Markov model can be obtained by selecting values for the initial probability vector **P**(0) and the Δt time increment. The value for **P**(Δt) can be computed by multiplying **P**(0) by the transition matrix. Next, **P**($2\,\Delta t$) can be obtained by multiplying **P**(Δt) by the transition matrix. In general, the solution to the equations of the Markov model is given by

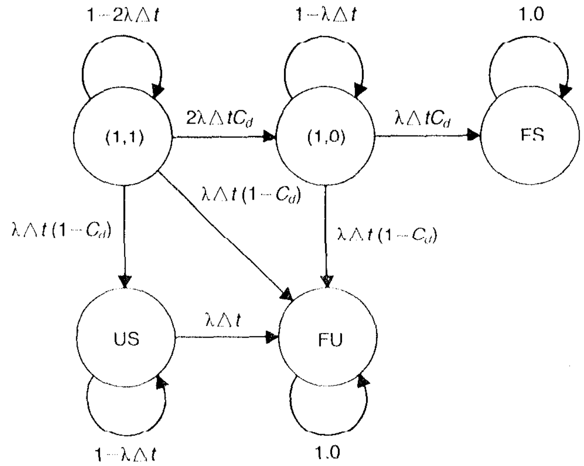
$$\mathbf{P}(n\,\Delta t) = \mathbf{T}^n\mathbf{P}(0)$$

The reliability, at some time t, of the reconfigurable duplication system is the probability that the system operates correctly from the initial time until time t. In reference to the Markov model, the reliability of the reconfigurable duplication system is the probability that the system will be in either state 2 or state 1, which are the only two states in which the system is operating correctly. Therefore, the reliability of the reconfigurable duplication system can be written as

$$R_{rd}(t) = p_2(t) + p_1(t)$$

where $R_{rd}(t)$ is the reliability of the reconfigurable duplication system at the time t.

The Markov model of the standby spare system is constructed in a manner similar to that of the reconfigurable duplication system. The resulting model is shown in Fig. 4.40. The Markov model of the standby spare system



State	Definition
(1,1)	Both online and spare modules are fault free.
(1,0)	Spare module has successfully replaced online module.
US	Undetected failure in spare module.
FS	System has failed in a safe manner.
FU	System has failed in an unsafe manner.

Fig. 4.40 Five-state Markov model of the standby spare system.

contains a total of five states. State (1, 1) represents the condition where the system has both the online processor and the spare processor functioning in a fault-free manner. State (1,0) corresponds to the existence of one of two conditions: (1) the online processor has failed and the spare has been successfully brought on-line or (2) the spare has failed, the failure has been detected, and the spare has been taken out of service. State (*FS*) occurs when both the online processor and the spare processor have failed and both failures have been detected and appropriately handled. While in state (*FS*), the system cannot function, but the system is safe. State (*FU*) represents the condition where the system has failed in an unsafe manner. The system can enter state (*FU*) in two ways: through the undetected failure of the online unit or through the undetected failure of the spare processor and the subsequent use of that spare when the online unit fails. In both cases, the system is operating with processors that possess undetected failures. The final state is state (*US*). While in state (*US*), the system continues to perform its functions because the online unit has not failed; however, the spare has failed in a manner that is undetectable. The undetectable failure of the spare results in a faulty spare being substituted for a faulty online unit, in the event that the online unit fails and the failure of the online unit is detected.

The equations for the Markov model of the standby spare system can be written in matrix form as

$$\mathbf{P}_{ss}(t + \Delta t) = \mathbf{T}_{ss}\,\mathbf{P}_{ss}(t)$$

where

$$\mathbf{P}_{ss}(t) = \begin{bmatrix} p_{(1,1)}(t) \\ p_{(1,0)}(t) \\ p_{(FS)}(t) \\ p_{(FU)}(t) \\ p_{(US)}(t) \end{bmatrix}$$

$$\mathbf{T}_{ss} = \begin{bmatrix} 1 - 2\lambda\,\Delta t & 0 & 0 & 0 & 0 \\ 2\lambda\,\Delta t C_d & 1 - \lambda\,\Delta t & 0 & 0 & 0 \\ 0 & \lambda\,\Delta t C_d & 1 & 0 & 0 \\ \lambda\,\Delta t(1 - C_d) & \lambda\,\Delta t(1 - C_d) & 0 & 1 & \lambda\,\Delta t \\ \lambda\,\Delta t(1 - C_d) & 0 & 0 & 0 & 1 - \lambda\,\Delta t \end{bmatrix}$$

$$\mathbf{P}_{ss}(t + \Delta t) = \begin{bmatrix} p_{(1,1)}(t + \Delta t) \\ p_{(1,0)}(t + \Delta t) \\ p_{(FS)}(t + \Delta t) \\ p_{(FU)}(t + \Delta t) \\ p_{(US)}(t + \Delta t) \end{bmatrix}$$

The standby sparing approach will be completely operational as long as the system is in one of three states: state (1, 1), state (1,0), or state (*US*). Therefore, the reliability of the standby sparing system can be written as

$$R_{ss}(t) = p_{(1,1)}(t) + p_{(1,0)}(t) + p_{(US)}(t)$$

As is evident from the Markov models of the two candidate systems, the reliability depends on several key factors, including the failure rates, coverage factors, and time.

For comparison purposes, assume that the failure rate λ is 3.5×10^{-4}, the coverage provided by the comparison process C_c is perfect, and the self-diagnostics provide a coverage factor C_d of 0.95. Using the Markov models of the two systems allows the reliability of each system to be computed as a function of time. Figure 4.41 compares the reliability of the reconfigurable duplication system and the standby sparing system for the given failure rate and coverage factors. At a reasonably high fault coverage value, the reliability of the two approaches is very similar. The value of the fault coverage factor, however, does have a significant impact on the reliability of the system.

Figure 4.42 shows the reliability of the reconfigurable duplication concept and the standby sparing system as functions of the fault coverage factor C_d. For a given module reliability, the standby sparing approach is capable of achieving a reliability that exceeds that of reconfigurable duplication. The single exception to this fact occurs when the coverage is perfect,

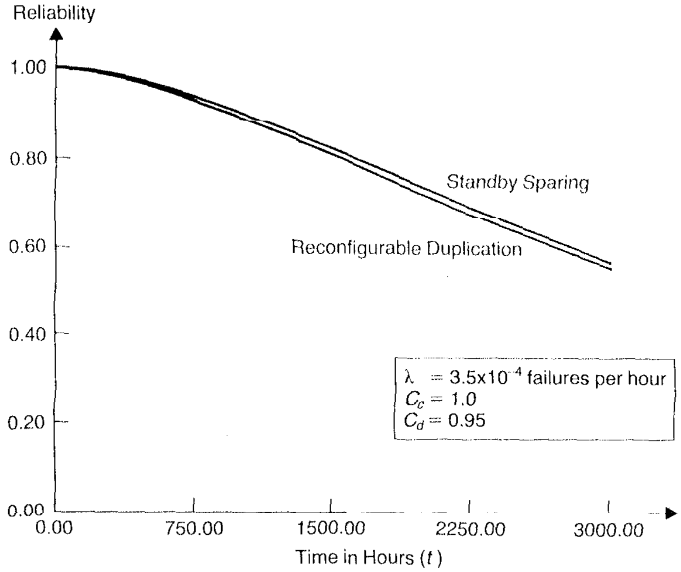


Fig. 4.41 Reliability of standby sparing and reconfigurable duplication versus time.

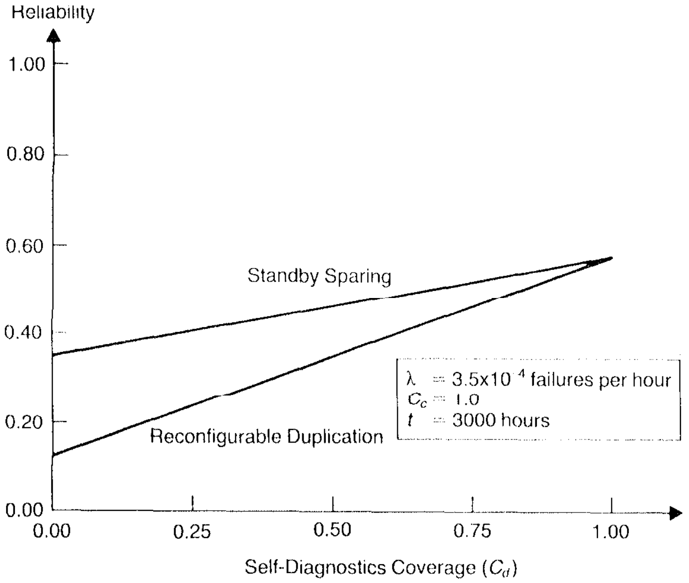


Fig. 4.42 Reliability of standby sparing and reconfigurable duplication versus fault coverage.

in which case the reliability of the two approaches is equivalent. This is as expected because with perfect fault coverage, both the reconfigurable duplication and the standby sparing concepts are equivalent to a perfect parallel system. At the other extreme where the fault coverage factor is zero, the reliability of the standby sparing system will be simply the reliability of the online module. The reliability of the reconfigurable duplication system, however, will be the probability that both modules are fault free because the comparison process can only detect the existence of a fault and cannot determine the source of the fault. If the self-diagnostics have zero fault coverage, any fault will result in the system becoming inoperable.

If one simply conducted a reliability analysis, the conclusion would be that the use of standby sparing is preferred because of the improved reliability that can be obtained. For a given fault coverage factor, the standby sparing approach is capable of achieving a reliability that is at least as high as that of the reconfigurable duplication system. In many applications, however, the reliability is not the only consideration. The reliability of the system simply accounts for the probability that the system will perform its functions correctly. In many cases, the designer is also interested in the probability that if the system does not remain operational it will at least fail in a manner that is safe. For example, the pilot of an airplane can fly the aircraft even if the auto-pilot fails, provided that it fails in such a way that it does not affect the remainder of the system. Likewise, in this control application, the user can survive even if the controller fails, as long as the failure does not result in the system performing some undesired function. This is the concept of *fail-safe operation*. A system fails safely when the failure does not produce an incorrect action from the system; the system may not perform its functions, but it at least does not perform the wrong functions.

Recall that we defined safety as the probability that a system will either perform its functions correctly or will discontinue its functions in a manner that is completely safe. In other words, the system either operates correctly or fails safely.

Referring to the Markov model of the reconfigurable duplication system shown in Fig. 4.39, the system will be safe as long as it is in one of three states: state 2, state 1, or state FS. Therefore, the safety of the system can be written as

$$S_{rd}(t) = p_2(t) + p_1(t) + p_{FS}(t)$$

where $S_{rd}(t)$ is the safety of the reconfigurable duplication system.

The safety of the standby sparing system can be written in a similar manner. Referring to the Markov model of the standby sparing approach, shown in Fig. 4.40, the system will be safe as long as it is in one of four states: state (1, 1), state (1,0), state (FS), or state (US). The safety can be written as

$$S_{ss}(t) = p_{(1,1)}(t) + p_{(1,0)}(t) + p_{(FS)}(t) + p_{(US)}(t)$$

It is interesting to compare the reliability and the safety of a particular system. Figures 4.43 and 4.44 show the reliability and safety of both candidate architectures as functions of time. As can be seen, the safety of both systems is considerably higher than the reliability.

To allow the safety feature to be further investigated, the failure rate, time, and the fault coverage of the comparison process were held constant and the fault coverage of the self-diagnostics was allowed to vary. The failure rate is $\lambda = 3.5 \times 10^{-4}$ failures per hour, the comparison fault coverage is $C_c = 1.0$, and the time was selected as 3000 hours. The 3000 hour time period corresponds to approximately eight hours of use per day for one year. For these values, Figs. 4.45 and 4.46 show how the reliability and safety of the two systems vary as the capability of the self-diagnostics varies. As expected, the safety and the reliability of the standby sparing approach are equal when no coverage is provided by the self-diagnostics. The standby sparing system depends on the self-diagnostics to achieve any level of safety.

The safety of the reconfigurable duplication system has several interesting features. As the coverage of the self-diagnostics increases from zero until approximately .5, the safety of the system actually decreases. This can be explained as follows. When the coverage of the self-diagnostics is zero,

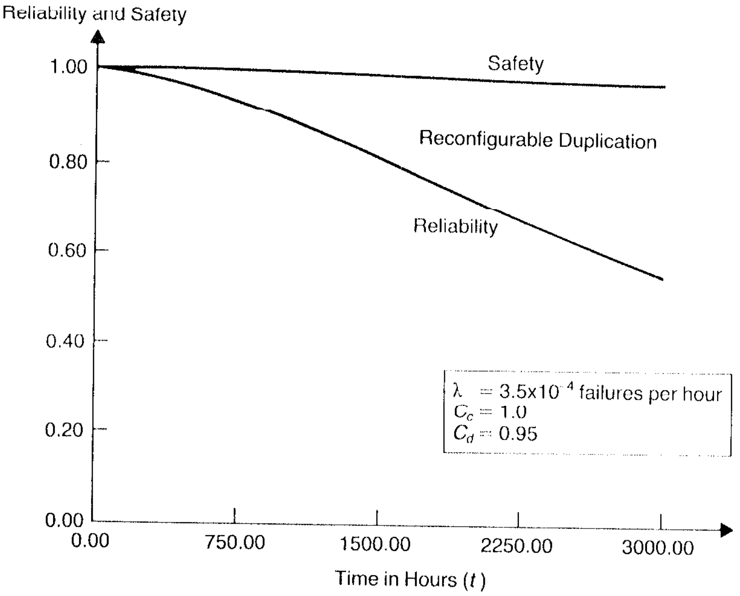


Fig. 4.43 Reliability and safety of reconfigurable duplication versus time.

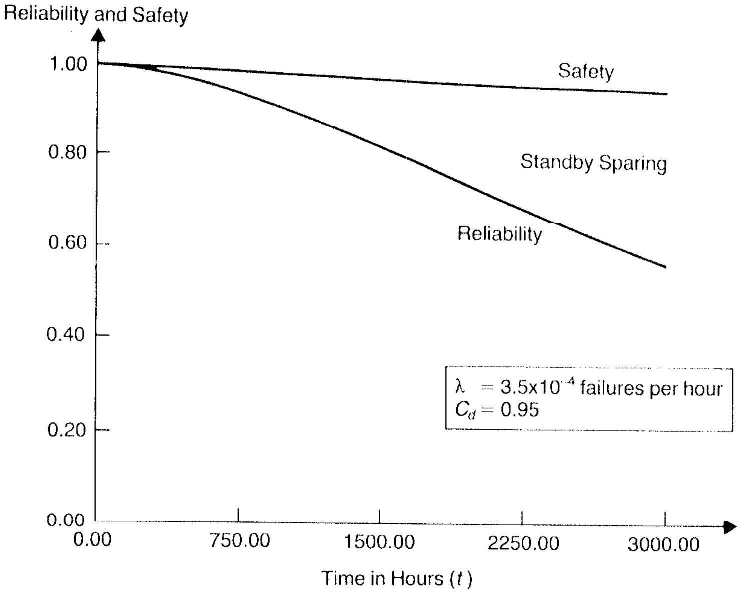


Fig. 4.44 Reliability and safety of standby sparing versus time.

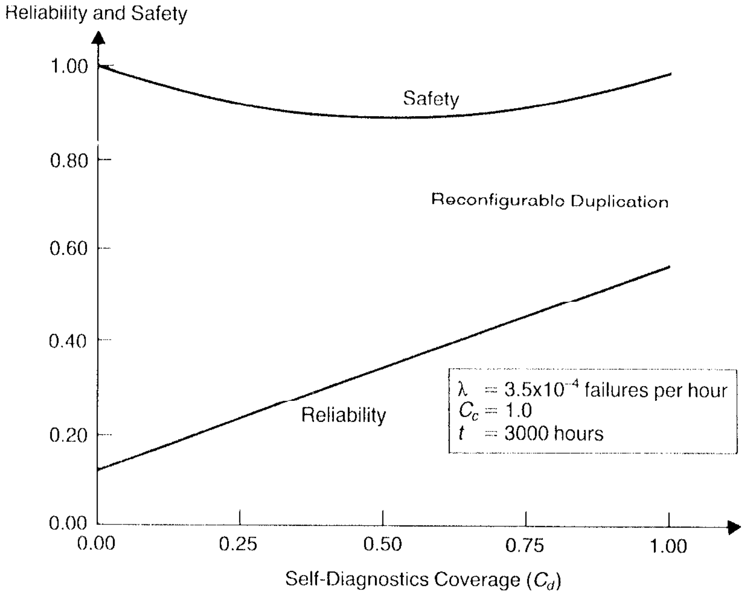


Fig. 4.45 Reliability and safety of reconfigurable duplication versus fault coverage.

the system will never reconfigure because the fault can never be located. The comparison process, in this example, is perfect, therefore, the fault will always be detected and the safety of the system maintained. As the self-diagnostics improve, some of the faults will be located and the system will be reconfigured. Once reconfigured, however, the system must depend on the self-diagnostics to maintain safety. But, when the self-diagnostics are very poor (less than .5 coverage), the safety of the system is compromised.

The coverage of the comparison process has a tremendous impact on the safety of the reconfigurable duplication system. Figure 4.47 shows the variation in the safety of the system as a function of both the coverage of the comparison process and the coverage of the self-diagnostics.

The final comparison is the safety of the reconfigurable duplication system and the standby sparing system. The use of comparison for fault detection cannot be perfect in most applications because of phenomena such as multiple faults and the manner in which the comparison is performed. Seldom is a direct bit-by-bit comparison between two units pratical. The units may receive their data through different analog-to-digital converters or from different sensors in a redundant system. Even though the sensors may be functionally equivalent, the outputs may not agree exactly. Small differ-

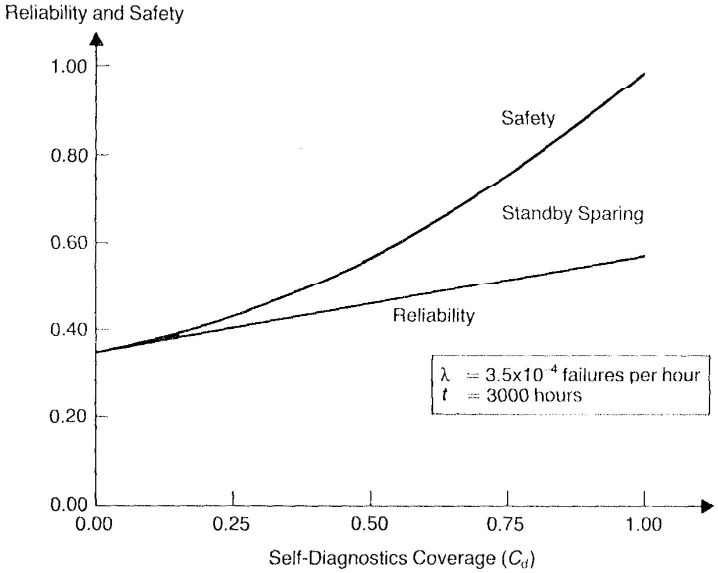


Fig. 4.46 Reliability and safety of standby sparing versus fault coverage.

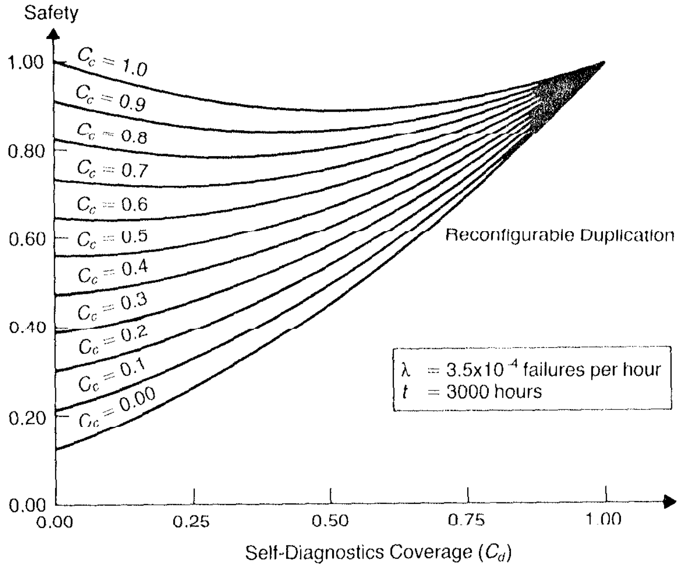


Fig. 4.47 Safety of reconfigurable duplication versus fault coverage for various comparison coverage values.

ences such as these can propagate through calculations to produce larger differences at the outputs of a process. Therefore, the comparison process may be performed by ignoring several of the least significant bits or by computing a difference between the two values and performing a threshold operation on that difference. If the difference exceeds the threshold, a miscompare is indicated. The result is something less than perfect coverage. Practice has shown that the comparison process can achieve a coverage of 0.95 or greater [Johnson and Julich 1985]. Figure 4.48 compares the safety of the reconfigurable duplication system and the standby sparing system for a comparison coverage of 0.95 and for various values of the self-diagnostics coverage factor. As can be seen, the reconfigurable duplication system achieves a much higher safety than the standby sparing approach.

Even though the standby sparing system achieves a higher reliability, for a given self-diagnostics coverage factor, than the reconfigurable duplication concept, the latter is preferable in many applications because of the improved safety. The importance of the system's safety has made the reliability somewhat misleading in that the most reliable system is not necessarily the most desirable one in applications mandating high safety.

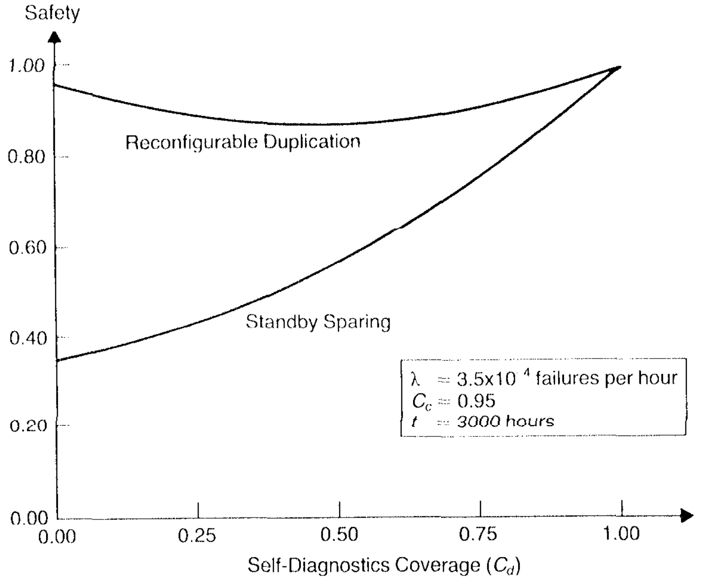


Fig. 4.48 Comparison of standby sparing and reconfigurable duplication safety versus fault coverage.

Summary

This chapter has presented the various analysis techniques that are available to evaluate fault-tolerant systems. Several methods for determining the reliability, availability, and safety of systems have been developed and illustrated, including both combinatorial modeling and Markov modeling approaches. Perhaps one of the most important parts of this chapter has been the example tradeoff analysis. During that analysis, several interesting comparisons were made on the reliability and safety of a practical fault-tolerant computing architecture for a digital control application. The analysis illustrated the importance of considering several factors during the decision making process. With the tools presented in the first four chapters of this book, we are now ready to practice the process of designing fault-tolerant systems. Chapter 5 begins to explore example designs as well as the mechanisms of the design process.

The following list provides a summary of the key concepts presented in this chapter.

Bathtub Curve—the variation of the failure rate with time for electronic components. The failure rate is assumed to be constant during the useful life of the component.

Combinatorial Models—a method of developing an analytical expression for a system's reliability.

Exponential Failure Law—a relationship whereby reliability varies exponentially with time.

Failure Rate, λ—the expected number of failures per unit time.

Fault Containment Coverage—the probability of fault containment given that a fault has occurred.

Fault Coverage—a generic term used for the probability of fault recovery given that a fault has occurred.

Fault Detection Coverage—the probability of fault detection given that a fault has occurred.

Fault Location Coverage—the probability of fault location given that a fault has occurred.

Fault Recovery Coverage—the probability of fault recovery given that a fault has occurred.

Flexibility—the ability of a system to adapt to change.

M-of-*N* System—a system in which *M* out of *N* components must operate correctly for the system to operate correctly.

Markov Model—a method of modeling reliability in terms of system state and state transition.

Mean Time Between Failure (MTBF)—the average time between consecutive failures. The MTBF is the sum of the MTTF and the MTTR.

Mean Time To Failure (MTTF)—the average time that a system will operate before the first failure is encountered.

Mean Time To Repair (MTTR)—the average time required to repair a system.

MIL-HDBK-217—a United States Department of Defense standard that defines a method of calculating failure rates.

Mission Time, MT[r]—the time at which the reliability falls below the value of r.

Mission Time Improvement—the ratio of the mission times of two systems being compared.

Parallel System—a system in which only one of N components must operate correctly for the system to operate correctly.

Redundancy Ratio—the ratio of the amount of a resource (hardware, software, time, or information) used in a redundant system to the amount of the same resource used in a nonredundant system.

Reliability Block Diagram—a graphical method of depicting the elements that must operate correctly for the system to operate correctly.

Repair Rate, μ—the expected number of repairs per unit time.

Series System—a system in which all components must operate correctly for the system to operate correctly.

State Transition—the process of transitioning from one system state to another.

Steady-state Availability—the limiting value of the availability function as time approaches infinity.

System State—a description of the combination of operational and failed modules in a system.

Technology Dependence—the dependence of a system on technological progress.

Testability—the ability to verify that a system is operating correctly.

Transparency to User—the impact of fault tolerance on the user of a system.

Weibull Distribution—a technique used to represent time-varying failure rate functions.

References

1. Avizienis, A., G.C. Gilley, F.P. Mathur, D.A. Rennels, J.A. Rohr, and D.K. Rubin. "The STAR (Self-Testing And Repairing) computer: An investigation of the theory and practice of fault tolerant computer design," *IEEE Transactions on Computers*, Vol. C-20, No. 11, November 1971, pp. 1394–1403.
2. Bouricius, W.G., W.C. Carter, and P.R. Schneider. "Reliability modeling techniques for self-repairing computer systems," *Proceedings of the 24th ACM Annual Conference*, 1969, pp. 295–309.
3. Ihara, H., K. Fukuoka, Y. Kubo, and S. Yokota. "Fault tolerant computer system with three symmetric computers," *Proceedings of the IEEE*, Vol. 66, No. 10, October 1978, pp. 1160–1177.
4. Johnson, B.W., and P.M. Julich. "Fault tolerant computer system for the A129 helicopter," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-21, No. 2, March 1985, pp. 220–229.
5. Johnson, B.W., and J.H. Aylor. "Reliability and safety analysis of a fault-tolerant controller," *IEEE Transactions on Reliability*, Vol. R-35, No. 4, October 1986, pp. 355–362.
6. Katzman, J.A. "A fault-tolerant computing system," in *The Theory and Practice of Reliable System Design* by D.P. Siewiorek and R.S. Swarz, Digital Press, Bedford, Mass. 1982.
7. Nelson, V.P., and B.D. Carroll. *Tutorial: Fault-Tolerant Computing*, IEEE Computer Society Press, Washington, D.C., 1986.
8. Ossfeldt, B.E., and I. Jonsson. "Recovery and diagnostics in the central control of the AXE switching system," *IEEE Transactions on Computers*, Vol. C-29, No. 6, June 1980, pp. 482–491.
9. Rennels, D.A. "Architectures for fault tolerant spacecraft computers," *Proceedings of the IEEE*, Vol. 66, No 10, October 1978, pp. 1255–1268.
10. Shooman, M.L. *Probabilistic Reliability: An Engineering Approach*, McGraw-Hill, New York, 1968.
11. Siewiorek, D.P., and R.S. Swarz, *The Theory and Practice of Reliable System Design*, Digital Press, Bedford, Mass. 1982.
12. Toy, W.N. "Fault tolerant design of local ESS processors," *Proceedings of the IEEE*, Vol. 66, No. 10, October 1978, pp. 1126–1145.
13. Trivedi, K.S. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, Englewood Cliffs, N.J., 1982.
14. United States Department of Defense, *Military Standardization Handbook: Reliability Prediction of Electronic Equipment*, MIL-HDBK-217, 1965.
15. United States Department of Defense, *Military Standardization Handbook: Reliability Prediction of Electronic Equipment*, MIL-HDBK-217B, 1974.
16. United States Department of Defense, *Military Standardization Handbook: Reliability Prediction of Electronic Equipment*, MIL-HDBK-217C, 1979.

Additional Reading

For the reader interested in pursuing the topics of this chapter in more detail, the following selection of additional reading is provided. This material covers articles and texts that are specifically devoted to the analysis of digital systems.

Beaudry, M.D. "Performance-related reliability measures for computing systems," *IEEE Transactions on Computers*, Vol. C-27, No. 6, June 1978.

Borgerson, B.R., and R.F. Freitas. "A reliability model for gracefully degrading and standby sparing systems," *IEEE Transactions on Computers*, Vol. C-24, No. 5, May 1975.

Costes, A.C., C. Landrault, and J.C. Laprie. "Reliability and availability models for maintained systems featuring hardware failures and design faults," *IEEE Transactions on Computers*, Vol. C-27, No. 6, June 1978.

Mathur, F.P. "On reliability modeling and analysis of ultra-reliable fault-tolerant digital systems," *IEEE Transactions on Computers*, Vol. C-20, No. 11, November 1971.

Masreliez, C.J., and B.E. Bjurman. "Fault tolerant system reliability modeling/analysis," *Journal of Aircraft*, Vol. 14, No. 8, August 1977.

Meyer, J.F. "On evaluating the performability of degradable computing systems," *IEEE Transactions on Computers*, Vol. C-29, No. 8, August 1980.

Meyer, J.F., D.C. Furchtgott, L.T. Wu. "Performability evaluation of the SIFT computer," *IEEE Transactions on Computers*, Vol. C-29, No. 8, August 1980.

Molloy, M.K. "Performance analysis using stochastic petri nets," *IEEE Transactions on Computers*, Vol. C-31, No. 9, September 1982.

Ng, Y.W., and A.A. Avizienis. "A unified reliability model for fault tolerant computers," *IEEE Transactions on Computers*, Vol. C-29, No. 11, November 1980.

Pedar, A., and V.V. S. Sarma. "Phased-mission analysis for evaluating the effectiveness of aerospace computing systems," *IEEE Transactions on Reliability*, Vol. R-30, No. 5, December 1981.

Siewiorek, D.P., V. Kini, H. Mashburn, S.R. McConnel, and M.M. Tsao. "A case study of C.mmp, Cm*, and C.vmp: Part II- predicting and calibrating reliability of multiprocessor systems," *Proceedings of the IEEE*, Vol. 66, No. 10, October 1978.

Takahashi, K. "Reliability and availability of redundant satellite orbit systems," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-18, No. 3, May 1982.

Varshney, P.K. "On analytical modeling of intermittent faults in digital systems," *IEEE Transactions on Computers*, Vol. C-28, No. 10, October 1979.

Wakerly, J.F. "Microcomputer reliability improvement using triple-modular redundancy," *Proceedings of the IEEE*, Vol. 64, No. 6, June 1976.