Abstract

This paper proposes a new approach to computing the reliability of a system. Worst pattern analysis finds the component misbehavior pattern that produces the greatest system unreliability. The motivation is that it is difficult and expensive to obtain information about the behavior of faulty components. It is also difficult and expensive to obtain information about a system's response to faulty components. For these reasons, this approach is based on minimum information about faulty components, and minimum requirements for system response. There are two minimum requirements. First, quantitative reliability requires that the failure rate of the components be known. Second, long term reliability requires that the system recognize when a redundant unit is misbehaving. In the worst pattern approach all other information is replaced by a computational effort. The procedure is illustrated by an example.

Introduction

We propose a new method of establishing the reliability of redundant/reconfigurable systems. This approach is based on using a minimum of data about device failures and system response to faulty components. This section discusses the basis of this approach, the approach itself, the relation of this approach to future systems, and a survey of the literature that places this approach in context.

Basis of Approach

This approach is based on the architecture of redundant and reconfigurable digital systems. Such systems have a number of major components whose faulty behavior can be observed (by comparison since it's a redundant system). Components known to be faulty can be replaced during maintenance. These components are built of devices whose failure rate is known. (If their failure rate is not known, there cannot be a quantitative reliability assessment.) Their behavior when faulty, however, may not be known. In addition, since these devices are embedded in the system's major components, the manifestation of faults can be a complicated process. This is displayed in figure 1.

![Figure 1. Structure of a Major Component in a System](image)

If an embedded device becomes faulty, the best behavior is that no incorrect result ever appears on the major component's output registers (or output pins). The next best behavior is that incorrect results immediately and constantly appear on the component's output registers. In the latter case the component can be replaced at maintenance before other components become faulty.

The worst pattern is that misbehavior occurs infrequently enough that the fault is classified as transient and faulty components accumulate in the system, but frequently enough that there is a good probability of multiple component misbehavior once a number of faulty components have accumulated.

In this scenario, faults are benign when they first occur because their effects must propagate to the output registers. Let $\beta$ be the benign to active rate. Figure 2 shows the probability of failure plotted against $\beta$. 

U.S. Government work not protected by U.S. Copyright 12.D.4-1
Worst Pattern Analysis

Straightforward worst pattern analysis has three requirements—a field data requirement, a design requirement, and a modeling requirement. The field requirement is that the failure rate of the components is known. This is essential for any quantitative assessment of reliability. The design requirement is that the system recognizes when a major component is misbehaving. This can be established by a combination of testing and arguments from design. This is a stringent requirement, but not as difficult as providing diagnostics for embedded devices. The modeling requirement is a description (model) for the system's detection and handling of component misbehavior. This description includes a model of the algorithms for deciding whether a component has a permanent or transient fault.

It looks possible, although it has not yet been done, to combine worst pattern analysis with other approaches to reliability assessment. We'll consider two possibilities. First, the design requirement that the system recognize component misbehavior may be only partially established. In this case we can adopt methods used for imperfect diagnostics (or lack of coverage) where some fraction of the misbehavior that occurs is not recognized or not handled correctly. Relevant papers on determining and modeling a lack of coverage are listed in the literature survey. Second, it's possible that a significant fraction of device misbehavior and error propagation is understood and can be modeled. There are a large number of papers on fault injection (see the literature survey). In this case, a less conservative reliability estimate can be obtained by performing fault injections experiments for a fraction of the faults and handling the rest by a worst pattern analysis.

Future Systems

For the future we expect a proliferation of devices and the integration of more functions on a chip. Worst pattern analysis goes well with both these predictions. If there are a large number of devices, it becomes difficult to determine the faulty behavior of all (or a significant fraction) of them. More complicated devices will have more complicated faulty behavior. For both cases, a method that requires a minimum of information is appealing.

In addition, for the future, we expect an increase in computational capacity. This favors a method that depends on computational effort instead of experimental effort.

Literature Survey

In general, reliability assessment is regarded as a fruitful area for probability modeling and new computational approaches [1]. For the reliability analysis of reconfigurable systems, there appears to be two major strands. One strand assumes that faults manifest themselves quickly, and the characteristics of fault propagation and manifestations are included in a reliability model. The resulting class of reliability models generates several computational problems. One is numerical stiffness, the combination of slow fault arrival with fast and complicated fault recovery. The stiffness problem is usually handled by model approximation [2, 3]. Another computational problem is the generation and analysis of the extremely large models for redundant/reconfigurable systems. This has prompted advances in computational methods [4]. This last paper is encouraging for the computational burden encountered when doing worst pattern analysis because computational problems become research topics for numerical analysts and computer scientists. As part of this approach where faults manifest themselves quickly, there are efforts to model fault propagation and effect [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Some papers consider one of the motivations for the worst pattern approach, the problem of authenticity—do
laboratory fault injections mimic actual fault occurrences [10, 22, 25].

The other major strand is coverage. Coverage can be defined as the probability of a good outcome given a fault occurrence. The meaning of “good outcome” depends on the circumstances, but there are two common interpretations. One is the diagnostic problem—some faults when injected remain undetected (at least for the duration of the experiment). Another (more comprehensive) interpretation is the correct handling of a faulty component—its detection, identification, and removal. Studies of the coverage factor are in [5, 7, 15, 18, 19, 20, 21, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 44]. There are efforts to incorporate coverage in a reliability model [5, 43, 44].

Worst pattern analysis includes the pattern where faults manifest themselves quickly and the pattern where faults remain undetected for an arbitrary period. It also includes all the intermediate patterns. Hence, worst pattern analysis includes both of the strands above.

Worst pattern analysis changes the nature of fault injection. It is no longer necessary to study the propagation of errors from a faulty embedded device to the interfaces of a major component where they can be observed by the system. It is only necessary to establish that the system recognizes errors at the interfaces of its major components.

**Description of Example**

The sections below describe the system architecture, the fault handling strategy, the Markov model for the system and faults, and the system parameters. Finally, the numerical range for the misbehavior model is derived from the system parameters and the operating time.

**Architecture**

The system is a reconfigurable threeplex plus a cold spare. When an original member of the threeplex is declared (permanently) faulty, it is replaced by the spare. When another component is declared faulty, the system becomes a duplex.

There is system failure whenever half (or more) of the working components are actively faulty.

**Fault Behavior and Handling Models**

The single fault model is in figure 3. The mnemonics are g for a good component, b for a faulty but benign component, and a for an actively faulty component.

![Figure 3. Single Fault Model](image)

In state bgg1, the fault has arrived. The benign to active rate is $\beta$, and the active to benign rate is $\alpha$. When the fault is in the active mode, the system decides it is permanent and removes it at rate $\epsilon$. The fault can become benign before the system removes it in which case the system decides it is a transient at rate $\theta$. If it becomes active again before the system decides it is a transient, the system removes it.

The double fault model is in figure 4 with the same mnemonics and rates. State F is the failure state.

![Figure 4. Double Fault Model](image)
In state \textit{bbg1}, both faults have just arrived or have been quiescent for a sufficient period to have been declared transients. In state \textit{abg}, one of them has become active. If the other fault becomes active there is system failure. The active fault is removed at rate $\varepsilon$. In \textit{bbg2} the first fault has become benign. If it becomes active before being declared transient, it is removed. In \textit{bbg2} the other fault (in a different component) can become active, causing a transition to state \textit{bag}. In state \textit{bag}, the system is still tracking the first fault (now benign), and if it remains quiescent the system declares it a transient and the system goes to state \textit{abg}. From state \textit{bag} the system can fail or remove the active fault (at the indicated rates). If the second fault becoming benign is the first event that occurs, then the system goes to state \textit{bbg3} where the system is still tracking both faults (since both faults were recently active). If either fault becomes active in state \textit{bbg3}, it is removed. The faults in state \textit{bbg3} are declared transient (one at a time) if they remain quiescent for a sufficient period.

The triple fault model is similar.

\textbf{Markov Model}

The Markov model uses the three requirements for doing worst case analysis. First, the failure rate for all the devices is known. Each major component (a module of the threeplex) has a total failure rate of $\lambda$. Second, the model includes the feature that the system recognizes when a major component is misbehaving. The inclusion of this element in the model is subtle. Essentially, the transition represented by the rate $\varepsilon$ is a complete density function—if a component misbehaves long enough, it will be removed. Third, there is a description (model) for the system's detection and handling of component misbehavior. That is, we have the transitions represented by the rates $\varepsilon$ and $\theta$, and numerical values for these rates.

The model is presented in five panels (see Figures 5-9). The failure state F is state number 36.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure5.png}
\caption{First Part of Model for Threeplex with Cold Spare}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure6.png}
\caption{Second Part of Model for Threeplex with Cold Spare}
\end{figure}

12.D.4-4
**System Parameters**

The system parameters are:

- failure rate $\lambda = 1e^{-4}$/hour
- fault removal rate $\varepsilon = 1e+3$/hour
- quiescent rate $\theta = 5e+2$/hour

The operating time is $T = 2$ hours. The control cycle is 50 milliseconds (20/second).

**Fault Behavior Parameters**

The range of values for the on-and-off cycle for faults is based on the operating time, the control cycle time, and the results of preliminary numerical studies.

Since the control program operates at twenty cycles per second, the smallest average time for $\beta$ (the benign to active rate) is $1/40$ seconds which is equivalent to a rate of 144,000 per hour. Rounding up gives a maximum value of $1e+6$/hour. Since the operating time is two hours, it appears reasonable to
choose 100 hours as the largest average time for \( \beta \).
This gives a rate of \( 1 \times 10^{-2} \) hour.

Because an incorrect result remains at the
output registers for at least one control cycle, the
smallest average time for \( \alpha \) (the active to benign
rate) is \( 1/20 \) seconds which is equivalent to a rate of
72,000 per hour. Rounding up gives a value of
\( 1 \times 10^5 \) /hour. Based on preliminary numerical results,
a maximum rate of \( 1 \times 10^6 \) /hour was chosen. Since
the operating time is two hours, it appears
reasonable to choose 100 hours as the largest
average time for \( \alpha \). This gives a rate of \( 1 \times 10^{-2} \) /hour.

Hence, for this system, both \( \alpha \) and \( \beta \) range
from \( 1 \times 10^{-1} \) per hour to \( 1 \times 10^6 \) per hour.

**Single Operating Period Results**

This section finds the worst fault behavior
pattern for the system during a single operating
period of two hours. It has been located by two
different methods. The first method was by
exhaustive computation over the range of values for
\( \alpha \) and \( \beta \). A coarse 9x9 grid was used with adjacent
points representing an order of magnitude
difference in \( \alpha \) or \( \beta \). The second method was a
search algorithm that examined adjacent points and
chose the max for the next starting point. Both
methods returned the same result. The exhaustive
computation results are displayed in figure 10.

The max probability of failure for a single
operating period is

\[
P(F) = 4.0776 \times 10^{-10}
\]

for the parameter values

\[
\alpha = 1 \times 10^4 \quad \text{and} \quad \beta = 10.
\]

**System Lifetime Results**

Since faults can stay benign for a considerable
period of time and they are undetectable when
benign, there is a possibility of imperfect
diagnostics for the maintenance checks between
operating periods. The previous section, with its
analysis of a single operating period, assumed that
maintenance (between operating periods) detects all
faults, even if they were benign during the
operating period. This section assumes the
opposite--faults are only detected during the
operating period. Between operating periods, all
the components detected as faulty are replaced with
good components. Let \( sf(k) \) be the probability of being in state \( k \) after an operating period, and let \( si(k) \) be the initial value for the next operating period. The assignment statements that give the replacement of components declared faulty are:

\[
\begin{align*}
si(1) &= sf(1) + sf(5) + sf(9); \\
si(2) &= sf(2) + sf(6) + sf(10); \\
si(3) &= sf(3) + sf(7); \\
si(4) &= sf(4) + sf(8); \\
si(5) &= si(6) = ... = si(23) = 0; \\
si(24) &= sf(24) + sf(12); \\
si(25) &= sf(25) + sf(13); \\
si(26) &= sf(26) + sf(14); \\
si(27) &= sf(27) + sf(15); \\
si(28) &= sf(28) + sf(15); \\
si(29) &= sf(29) + sf(17); \\
si(30) &= sf(30) + sf(18); \\
si(31) &= sf(31) + sf(19); \\
si(32) &= sf(32) + sf(20); \\
si(33) &= sf(33) + sf(21); \\
si(34) &= sf(34) + sf(22); \\
si(35) &= sf(35) + sf(23); \\
si(36) &= sf(36);
\end{align*}
\]

The computations assume the system has a lifetime of 20,000 two-hour operating periods. This is about six operations per day for ten years. The results are in figure 11. The max value is \( P(F) = 4.338 \times 10^{-4} \) for \( \alpha = 1 \times 10^6 \) and \( \beta = 1 \times 10^{-1} \).

Figure 11. Results for System Lifetime of 20000 Operating Periods

**Summary**

This paper considers one of the outstanding problems in establishing the fault tolerance of a redundant and reconfigurable system—the lack of information about the behavior of faulty devices and the manifestation of faulty behavior in a complex system. The solution offered is to find the misbehavior pattern that produces the largest probability of system failure. The largest probability appears when misbehavior occurs infrequently enough that the fault is classified as transient and faulty components accumulate in the system, but frequently enough that there is a good probability of multiple component misbehavior once a number of faulty components have accumulated. Since any other misbehavior pattern (including the actual misbehavior pattern) produces a smaller probability of failure, this maximum value can be used as metric for system reliability. This approach has three requirements. The field data requirement is that the failure rate of the components is known. The design requirement is that the system recognizes when a major component is misbehaving. The modeling requirement is a description (model) for the system’s detection and handling of component misbehavior. The technique is applied to an example to get both single operating period and lifetime results.
References


