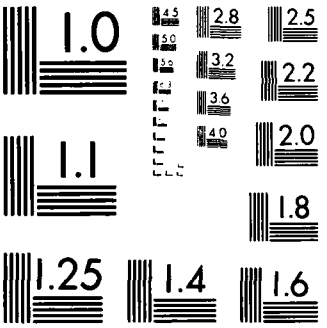
AD-A170 189 PERIODICALLY SELF RESTORING REDUNDANT SYSTEMS FOR VLSI 1/1
BASED HIGHLY RELIA (U) VIRGINIA POLYTECHNIC INST AND
STATE UNIV BLACKSBURG DEPT OF E A D SINGH ET AL.
UNCLASSIFIED 1984 ARO-18803.13-EL DAA229-82-K-0102 F/G 9/2 NL



UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

(2)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ARO 18803.13-EL	2. GOVT ACCESSION NO. N/A	3. RECIPIENT'S CATALOG NUMBER N/A
4. TITLE (and Subtitle) Periodically Self Restoring Redundant Systems for VLSI Based Highly Reliable Design		5. TYPE OF REPORT & PERIOD COVERED Reprint 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Adit D. Singh Dr. F. Gail Gray		8. CONTRACT OR GRANT NUMBER(s) DAAG 29-82-K-0102
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Electrical Engineering Virginia Polytechnic Institute & State University Blacksburg, VA 24061		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office Post Office Box 12211 Research Triangle Park, NC 27709		12. REPORT DATE 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reliability Modeling
Redundant Systems
Transient Faults
Temporary Faults
Soft Faults
Self-Restoration
VLSI

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Error masking by majority voting remains an important fault tolerance technique for realizing highly reliable computer systems for critical control applications. However, VLSI technology has imposed a relatively high cost on hardware voter circuits because of their high interconnect complexity. In this paper we present and analyze a new design for redundant microcomputer systems which appears well suited for implementation from VLSI modules. In the proposed design, redundant computing units (CU's) making up the system communicate with each other periodically to restore units that may have been disabled by

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
86 7 23 080

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Date Entered)

transient faults. The system is also protected against permanent failures. To evaluate the proposed approach a reliability model for triple redundant periodically self restoring systems is developed in this paper. The model accounts for both permanent and transient faults during system operation, as well as the possibility of undetected failures in the redundant units at the start of the mission.

PERIODICALLY SELF RESTORING REDUNDANT SYSTEMS FOR VLSI BASED HIGHLY RELIABLE DESIGN

Adit D. Singh
Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, Massachusetts 01003

F. Gail Gray
Department of Electrical Engineering
Virginia Tech
Blacksburg, VA 24061

ABSTRACT

Error masking by majority voting remains an important fault tolerance technique for realizing highly reliable computer systems for critical control applications. However, VLSI technology has imposed a relatively high cost on hardware voter circuits because of their high interconnect complexity. In this paper we present and analyze a new design for redundant microcomputer systems which appears well suited for implementation from VLSI modules. In the proposed design, redundant computing units (CU's) making up the system communicate with each other periodically to restore units that may have been disabled by transient faults. The system is also protected against permanent failures. To evaluate the proposed approach a reliability model for triple redundant periodically self restoring systems is developed in this paper. The model accounts for both permanent and transient faults during system operation, as well as the possibility of undetected failures in the redundant units at the start of the mission.

1. INTRODUCTION

Modular redundancy schemes have been widely used for implementing hardware fault tolerance in computer systems designed for critical real time control applications. In the classical "static" voted redundancy schemes, TMR (triple modular redundancy [1]) and its generalization NMR [2], failures in individual modules are masked by voter circuits so that a subsystem stays operational as long as a majority of its redundant modules continue to operate correctly. Systems employing the "dynamic" standby sparing scheme [3] have the built in capability of automatically detecting module failures, and replacing the failed module from a pool of redundant spares. For the same level of redundancy, a standby sparing system is generally better for long missions because it can usually stay operational down to the last correctly operating module. A NMR system operates correctly only as long as a majority of the redundant modules are failure free. However, for applications requiring highly reliable operation for short periods, the NMR system is often better because a majority of the redundant modules are unlikely to fail over such a short interval. The standby sparing system is more susceptible to malfunction in its generally complex fault detection and reconfiguration circuitry, which can lead to system failure well before all the spares are exhausted. Several 'hybrid' redundancy schemes have been proposed [4], [5], [6] to combine the advantages of NMR and standby sparing. Although some of the early space computers used basic TMR [7], state-of-the-art ultra-reliable systems such as the SIFT [8] and the FTMP [9] employ the hybrid approach by voting only among the redundant modules that are believed to be operating correctly; failed modules are discarded from the vote. Nevertheless, even in these systems, simple error masking by majority voting remains a primary fault tolerance technique for achieving highly reliable operation.

In implementing a TMR (or NMR) system, the system designer must decide on how the voting is carried out (by hardware or by software), and where the voters are placed (what signals or data are to be voted on). These decisions have a very significant impact on the resulting system reliability. In systems implemented out of VLSI building blocks, reliability depends more on the chip count and the interconnect complexity than on the circuit complexity per se. Unfortunately, voting requires a large number of interconnections. This results in high modular complexity because pinout limitations restrict the number of voter circuits that can be placed on a VLSI chip. Because dedicated computer systems deployed in control applications are often implemented from relatively few VLSI modules, the injudicious use of a large number of hardware voters can actually reduce overall systems reliability by significantly increasing the modular and interconnect complexity of the system [12].

The placement of the voters in a redundant system also greatly effects its capability of handling transient failures. The complex VLSI modules used in implementing present-day systems are generally complex sequential circuits containing memory. A transient fault in such a circuit can alter its state and result in con-

This work was supported in part by ARO grant DAAG29-82-k-102

tinued erroneous operation until the circuit is correctly resynchronized. Unless the voters help resynchronize such a disabled module, it has the same effect on the system as a module with a permanent failure. Thus a build up of disabled modules in a subsystem, beyond its fault tolerance capability, can cause system failure even though hardware resources exist in the system for continued operation. This can be a major reliability degrading factor because emperical data has shown that transient failures are far more frequent than permanent faults [10].

Recent NMR designs have taken two approaches on the question of voter placement. One approach, used in the space shuttle computers [11], is to vote only on the outputs obtained from a pool of independently operating redundant computer systems. Since the redundant units here do not interact to effect transient upset recovery, this approach is only suited to relatively short missions and requires a high level of redundancy for acceptable reliability. The other approach suggested by Wakerly [12] and employed in the C.vmp [13] is to use additional voters on the signal lines connecting the processors and memories as shown in Figure 1. (The C.vmp was actually implemented with a single bidirectional voter, but it takes triplicated voters to provide full single-point failure protection). In such a design the normal flow

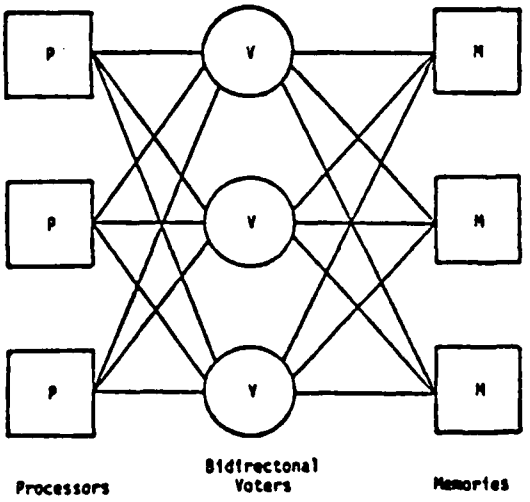


Figure 1: C.vmp type architecture. (Only a single voter was actually implemented)

of data between the processors and memories resynchronizes any processor that experiences a transient upset. However, note the high interconnect complexity created by the voters because of the large number of interconnection lines that run between the processor and memory in a computer system. Address, data, as well as timing and control signals must all be voted upon and correctly synchronized. The reliability of C.vmp type architectures has been analyzed by Wakerly [12] with respect to permanent faults. The analysis indicates an improvement in overall system reliability only if the modular complexity of the voters is low as compared to that for the rest of the system. While this is true of the C.vmp because it has been implemented out of MSI and LSI components, it doesnot generally to hold for systems implemented in VLSI technology. Therefore, newer designs such as the SIFT and the FTMP basically employ the first approach and provide massive redundancy. They do not attempt to restore units upset by transient faults, but instead, discard units making repeated errors from the vote and assume that such units have permanently failed. However, some transient upsets may still be corrected in these systems because partial results are voted on among the redundant units (explicitly by the software in SIFT, by the hardware during transfers between module triads in FTMP) and the majority opinion used by all processors for further computation.

In this paper we present and analyze a new approach wherein a set of redundant computer systems stay in sychronization by periodically communicating and restoring each other to the majority consensus state. Because this consensus voting is done in software, such as scheme provides protection from transient upsets without increasing the hardware complexity of the system. We shall show that such periodically self restoring redundant (PSRR) systems offer a very attractive approach for implementing TMR, NMR, and hybrid redundancy systems in a VLSI environment. In Section III we present a reliability model for triple redundant PSRR systems, which accounts for the effects of both transient and permanent faults. Making reasonable assumptions, closed form expressions for system reliability and mean time to failure are derived. These results should prove quite useful to the designer of PSRR systems because triple redundant systems can often provide the needed reliability for many applications. More elaborate models for general N redundant systems have been developed and can be found in [15].

It should be noted that in addition to recovery from transient upsets, there are other important issues that must be addressed in the implementation of NMR systems. These include the design of reliable clocking circuits to provide synchronized clock signals to the redundant modules. This clock system must itself be redundant and protected against single-point failures. Issues relating to data I/O from the system and its interactive consistency are also of critical importance. These problems are common to all redundant designs and have been addressed in the literature [16], [17], [18]. Here we shall assume that a reliable fault tolerant clock can be designed, and that each I/O interface has voters that monitor the output signals from the redundant processors and simultaneously provide input signals to all the redundant units.

2. PSRR SYSTEM DESCRIPTION

A triple redundant PSRR system employes three computing units (CU's) operating redundantly in synchronization. Each CU has all the computational capabilities of the desired fault tolerant system and, in general, is made up of processors, memories and I/O units. System input is provided simultaneously to all three CU's. System output is taken to be the majority of the three outputs available from the CU's. The voting is carried out on each output word.

A CU is said to be operational if its computations are error free. Since for a computing system correct operation only requires the correct execution of a set of programs and not necessarily the correct functioning of all hardware components, the proposed triple redundant system will be considered operational as long as the correct output can be recognized from among the three outputs available from the CU's. Clearly this requires that at least two CU's be operational.

In general, failure in a computing system may occur for diverse reasons. These include improper design, software failures, component failures, etc. In this paper we shall confine ourselves to system failure resulting from the failure of hardware components. We shall assume that the system has been properly designed and that the software is error free.

The failure of a hardware component may be either permanent or transient (intermittent) in nature. In the proposed redundant system made up of three CU's operating in synchronization, a CU may fail and fall out of step due to errors caused by both transient and permanent component failures. This is because a transient failure can upset the state of the CU and result in continued erroneous computations until the system is restored. A CU that has failed in this manner (due to a transient) will be considered to have temporarily failed. A CU with a permanent failure in a hardware component is said to have permanently failed. While permanently failed CU's obviously cannot be resynchronized with the rest of the system, if the CU failure is temporary (due to a transient), it can be brought back into synchronization with the rest. In the proposed system this is done by the three CU's communicating with each other periodically to restore failed CU's. To facilitate this communication, each CU is directly linked to both the other two CU's, as shown in Figure 2. The restoration program is initiated by non maskable interrupt from an external fault tolerant clocking circuit (such as the one described in [16]) and is executed by each CU out of read only memory (ROM). This insures that a transient failure that may have corrupted the memory in a CU does not prevent it from participating in the restoration process. During the restoration interval the three CU's vote in their entire memory contents and also on their processor states, replacing disagreeing words

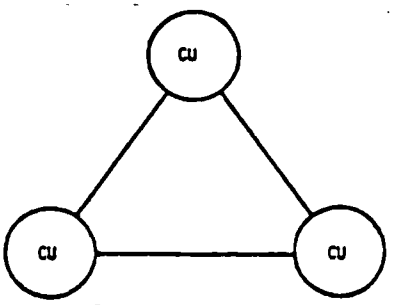


Figure 2: Triple redundant PSRR system

with the majority opinion. Thus, if over the restoration interval the system stays operational, that is two or more of the CU's operate correctly, then a temporarily failed CU will be restored provided it does not experience additional failures during this period.

Operation of the redundant system is broken up into computing intervals, when the system is performing useful computation, the restoration intervals, when temporarily failed processors are being restored (Figure 3). The length of the restoration interval is determined by the time required to execute the restoration program and is a function of the memory size of each CU. As discussed below, it

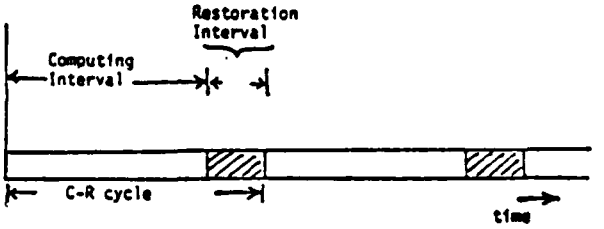


Figure 3: PSRR system operation

is estimated that the restoration interval will be quite small, typically less than a second for most systems. The length of the computation interval is a design option and determines the computation-restoration (C-R) cycle time. Shorter C-R cycle times imply more frequent system restoration. This increases system reliability by reducing the probability of system failure due to an accumulation of temporarily failed CU's.

At first it may appear that a word by word vote on the entire memory contents is not very attractive, particularly for real time control applications, because the system is not available for control purposes during this time. This draw back may indeed disqualify PSRR systems for some applications. Note however that such dedicated systems often do not require large memories. For example the main memory on the SIFT and FTMP is only 32K words. Further, because the control programs once developed do not have to be changed, they can be stored in ROM which does not have to be voted on. Thus

the read/write (RAM) memory size in such systems can be reduced to only that required to store and manipulate data and program control. Assuming an average time of ten microseconds to vote on one memory location, a system can vote on 100K memory words in one second. Thus the restoration interval for most PSRR systems will last only a fraction of a second. Even if restoration is carried once every few minutes which, as we shall see from the reliability model in the next section will usually be often enough, the performance degradation due to computation time lost during restoration is negligible. In comparison, the extra delays introduced by the voter in the C.vmp result in a performance degradation of about 15 percent [13] as compared to a non redundant system.

The triple redundant PSRR system described above can be generalized to a N redundant system in a straight forward way. The latter will comprise of N fully connected CU's with the periodic restoration carried out by a majority vote on the N system states. In practice, the reliability of N redundant PSRR systems can be still further improved by modifying the restoration process so as to discard CU's that are repeatedly in a minority (presumed to have permanently failed) from the vote. Such a "self purging" N redundant PSRR system would incorporate the advantages of dynamic redundancy into the PSRR concept. Robust distributed restoration algorithms for such self purging systems are currently under investigation.

3. RELIABILITY MODEL

The reliability R(t) of a fault tolerant system to time t is defined to be the probability that the system stays operational up to time t, given that it was completely free of failures at time t = 0.

For the purpose of this analysis, we shall assume that the restoration interval is negligibly short and that no failure takes place during the restoration process. This assumption is justified later in this section. The assumption implies that at the instant just after restoration, if the system is operational, it does not contain a temporarily failed CU. At any such instant, therefore, the system must be in one of the following three states

State 1: All CU's operational

State 2: Exactly two CU's operational, the third having permanently failed.

State 3: Failed system due to more than one failed CU. (This may result from any combination of temporary and permanent CU failures.)

Using these states we can model the operation of the proposed redundant system as a three state discrete parameter Markov chain [19]. The state transition probabilities over one C-R cycle time form the one step matrix of transition probabilities,

$$T = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{bmatrix}$$

It can be readily seen that some of the elements of T are trivial. $p_{21} = 0$ because once the system has a CU with a permanent failure (state 2), it can never go to a state with all operational CU's (state 1). Also $p_{31} = 0$, $p_{32} = 0$ and $p_{33} = 1$, because if the system fails (state 3) it can never recover to an operational state.

Making these entries in T we find that the matrix of transition probabilities is upper triangular.

$$T = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ 0 & p_{22} & p_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

The five remaining unknown elements in T depend on the CU failure probabilities and the C-R cycle time for the system. From these parameters, the remaining elements can be evaluated as follows,

Let p_t by the probability of a temporary CU failure occurring during a C-R cycle. Let $q_t = (1 - p_t)$ be the probability that such a failure does not occur. If a constant transient failure rate λ_t is assumed, then $p_t = (1 - e^{-\lambda_t t_0})$, where t_0 is the C-R cycle time. We shall assume that p_t is the same whether or not the CU is operational.

Similarly let p_p be the probability that a permanent CU failure occurs over a C-R cycle, $q_p = (1 - p_p)$ is the probability that such a failure does not occur. If λ_p is a constant permanent failure rate over a C-R cycle, then $p_p = (1 - e^{\lambda_p t_0})$. It is assumed that the occurrences of temporary and permanent failures are mutually independent.

In the matrix of transition probabilities T, p_{11} is the probability that a failure free system will still be failure free after one C-R cycle. Clearly this requires that no permanent failure and at most one temporary failure occurs over this interval. (Since we always consider the state of the system at the instant just after restoration, a single temporary failure will always be restored). Therefore

$$p_{11} = q_p^3 \times (q_t^3 + 3q_t p_t)$$

By similar reasoning

$$p_{12} = 3q_p^2 p_p \times (q_t^3 + q_t^2 p_t)$$

Also since $p_{11} + p_{12} + p_{13} = 1$

$$p_{13} = 1 - p_{11} - p_{12}$$

p_{22} is the probability that a state with one permanently failed CU is retained over a C-R cycle. This requires that no additional failures take place in the two CU's. Therefore

$$p_{22} = q_p^2 \times q_t^2$$

Again since $p_{22} + p_{23} = 1$

$$p_{23} = 1 - p_{22}$$

Once the one step transition probability matrix T is obtained, using the theory of Markov Chains the transition probabilities over n C-R cycles can be obtained by evaluating $(T)^n$.

The reliability of the systems R(t) to n C-R cycle times is the probability that a failure free system is still operational after n C-R cycles. Since state 1 represents a failure free system and state 3 a failed system, the (1,3), entry in $(T)^n$, $p_{13}^{(n)}$, gives the probability that a system that was initially failure free, fails over n C-R cycles. $(1 - p_{13}^{(n)})$ is the probability that the system is still operational. Therefore

$$R(nt_0) = 1 - p_{13}^{(n)}$$

We next obtain $p_{13}^{(n)}$ and hence $R(nt_0)$ in closed form in terms of the one step state transition probabilities. This can be done by taking advantage of the fact that since T is upper triangular, $(T)^n$ is also upper triangular.

$$(T)^n = (T)^{n-1} \times T$$

Writing $p_{12}^{(k)}$ as $(1 - p_{11}^{(k)} - p_{13}^{(k)})$, the above equation can be written as

$$\begin{bmatrix} p_{11}^{(n)} & (1-p_{11}^{(n)} - p_{13}^{(n)}) & p_{13}^{(n)} \\ 0 & p_{22}^{(n)} & p_{23}^{(n)} \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} p_{11}^{(n-1)} & (1-p_{11}^{(n-1)} - p_{13}^{(n-1)}) & p_{13}^{(n-1)} \\ 0 & p_{22}^{(n-1)} & p_{23}^{(n-1)} \\ 0 & 0 & 1 \end{bmatrix}$$

$$\times \begin{bmatrix} p_{11} & (1-p_{11}-p_{13}) & p_{13} \\ 0 & p_{22} & p_{23} \\ 0 & 0 & 1 \end{bmatrix}$$

Obtaining expressions for $p_{12}^{(n)}$ from both sides of this identity we get

$$1-p_{11}^{(n)}-p_{13}^{(n)} = p_{11}^{(n-1)}(1-p_{11}-p_{13}) + p_{22}(1-p_{11}^{(n-1)}-p_{13}^{(n-1)})$$

Noting that for a triangular matrix $p_{11}^{(k)}=(p_{11})^k$ and $p_{22}^{(k)}=(p_{22})^k$ We get

$$1-(p_{11})^n-p_{13}^{(n)} = (p_{11})^{n-1}(1-p_{11}-p_{13}) + p_{22}\{(1-(p_{11})^{n-1}-p_{13}^{(n-1)}\}$$

Rearranging terms gives the recurrence relation

$$p_{13}^{(n)}-p_{22}p_{13}^{(n-1)}=1-p_{22}-(p_{11})^{n-1}(1-p_{13}-p_{22})$$

This equation can be solved using known methods such as the one described in Liu [20]. The general solution to the recurrence relation is

$$p_{13}^{(n)}=\frac{-p_{22}}{p_{22}-p_{11}}(p_{22})^n+\frac{(p_{23}-p_{12})}{p_{22}-p_{11}}(p_{11})^n + 1$$

Thus the reliability of the periodically self restoring systems

$$R(nt_0) = 1 - p_{13}^{(n)}$$

is given by

$$R(nt_0) = \frac{p_{12}}{p_{22}-p_{11}}(p_{22})^n - \frac{(p_{23}-p_{13})}{p_{22}-p_{11}}(p_{11})^n$$

Note that this reliability expression is completely general and does not assume any specific failure distribution (such as the constant failure rate, exponential distribution), for the CU's. System reliability to estimated based on the temporary and permanent failure probabilities over one C-R cycle, and these can have any general time dependence. However, for simplicity, we do use constant failure rates to obtain the reliability plots in Figures 4 and 5. In these figures the transient failure rate λ_t is taken to be 0.01 per hour (an average of one failure every 100 hours) and the permanent failure rate λ_p to be 0.001 per hour. These values are consistent with observations [10] that permanent faults cause only a small fraction of system failures.

Figure 4 shows plots of reliability versus time for different C-R cycle times, t_0. Figure 5 shows the same plots for short mission times

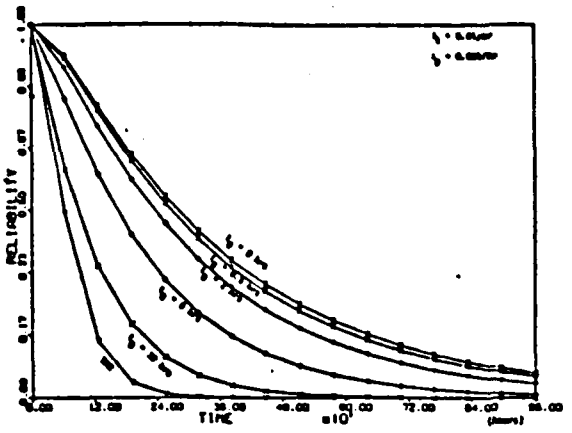


Figure 4: Reliability plots for the N = 3 systems

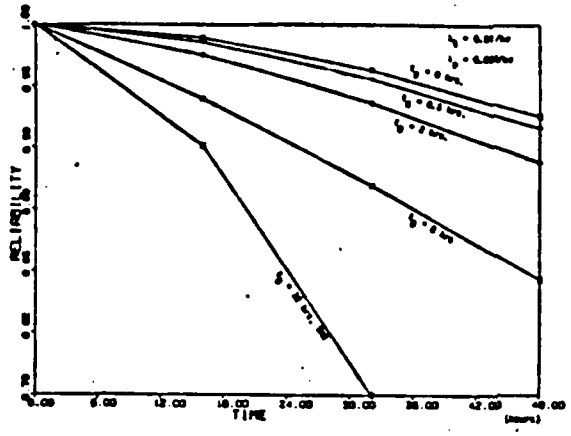


Figure 5: Reliability Plots for the N = 3 PSRR system for short missions

important in ultra-reliable design. As expected the plots indicate that reliability improves with more frequent restoration i.e., as the C-R cycle time is decreased. Also included in the figures are reliability plots for a unrestored TMR design with no provisions for restoring temporarily failed CU's ($t_0 = \infty$), and a hypothetical system that recovers instantaneously from temporary failures in any one CU as long as the other two CU's are operational ($t_0 = 0$). These two plots provide bounds on the reliability of the proposed scheme. If the C-R cycle time is made very large, in the limit infinite, restoration does not take place in the proposed scheme before the system fails, and its reliability is reduced to that of the unrestored TMR design, which has no provision for restoring failed CU's. On the other hand as the C-R cycle time is reduced and approaches zero, recovery from temporary failures is almost instantaneous and the proposed scheme approaches the hypothetical system described above. In practice this upper limit on reliability can never, of course, be reached since it requires that both the computational interval and the restoration interval be of zero duration.

Notice in Figures 4 and 5 the substantial improvement in reliability offered by PSRR systems as compared to unrestored TMR. This is particularly true of PSRR systems with short C-R cycle time. For example, over a 16-hour mission time, Figure 5 indicates that the unrestored TMR system is eight times more likely to fail than a PSRR system with $t_0 = 0.5$ hours. (Recall that the probability of system failure is [1-reliability].) As discussed in the previous section, such values of C-R cycle time t_0 are quite practical for PSRR systems.

4. MEAN TIME TO FAILURE CALCULATIONS

Another parameter of interest in fault tolerant systems is the mean time to failure (MTTF). In this section we derive the MTTF for triplicated PSRR systems.

Let μ_{ij} be the average number of C-R cycles taken by a system starting in state i to go to state j for the first time. Then for the three state triplicated PSRR System under discussion, it takes, on the average μ_{13} C-R cycles for a system with all CU's operational to fail. The system MTTF is, therefore, given by $\mu_{13} \times t_0$.

To evaluate μ_{13}, consider a system one C-R cycle after starting in state 1. At this point in time the system may be in any one of the three system states 1,2, or 3, with probability p_{11}, p_{12} and p_{13} respectively. Therefore, from this instant the average number of C-R cycles to system failure is given by $p_{11}\mu_{13} + p_{12}\mu_{23} + p_{13}\mu_{33}$. Since the system was in state 1 one C-R cycle before the instant under consideration,

$$\mu_{13} = 1 + p_{11}\mu_{13} + p_{12}\mu_{23} + p_{13}\mu_{33}.$$

It can be similarly seen that

$$\mu_{23} = 1 + p_{21}\mu_{13} + p_{22}\mu_{23} + p_{23}\mu_{33}.$$

Noting that $\mu_{33} = 0$ (the average number of C-R cycles to go from state 3 to itself for the first time is 0) and $p_{21} = 0$, the above two equations reduce to

$$\mu_{13} = 1 + p_{11}\mu_{13} + p_{12}\mu_{23}$$

$$\mu_{23} = 1 + p_{22}\mu_{23}$$

Solving simultaneously we get

$$\mu_{23} = \frac{1}{1-p_{22}}$$

$$\mu_{13} = \frac{p_{12} + p_{23}}{(1-p_{11})(1-p_{22})}$$

Therefore, the system MTTF is given by

$$MTTF = \frac{(p_{12} + p_{23})t_0}{(1-p_{11})(1-p_{22})}$$

It should be noted that the above expression for MTTF is valid only for $\mu_{13} \geq 2$ This is because in the derivation of μ_{13}, it is implicitly assumed that it always takes more than one C-R cycle for the system to fail, an assumption that is not valid unless $t_0 \ll \frac{1}{\lambda_t}, \frac{1}{\lambda_p}$. This inequality is satisfied for the range of values of t_0 in Figure 6, which displays a plot of MTTF versus t_0 for $\lambda_t = 0.01$ per hour and $\lambda_p = 0.001$ per hour.

We have seen in Figure 4 that the reliability of a PSRR system always improves with more frequent restoration. Therefore, the system MTTF must also increase with reduced t_0. While this is not immediately apparent from the above expression for MTTF because the state transition probabilities also depend on t_0, it is confirmed by the plot in Figure 6. Again as expected, the longest MTTF is obtained when t_0

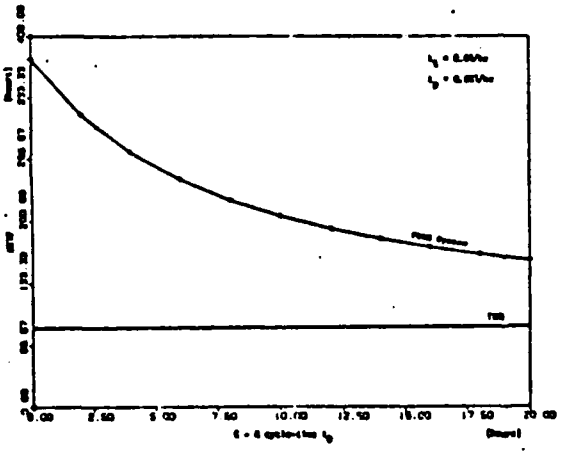


Figure 6: MTTF versus C-R cycle time t_0 for the N = 3 PSRR system

approaches zero. At t_0 grows large, the system MTTF asymtotically approaches that for a unrestored TMR system with no provisions for restoring temporarily failed CU's.

Notice in Figure 6 the very substantial improvement in MTTF achieved by PSRR systems as compared with unrestored TMR. For short C-R cycle times of up to 2 hours, which are expected for such systems, the MTTF for PSRR systems is over four times that for unrestored TMR systems.

5. INITIAL FAILURE PROBABILITIES

The reliability of most current fault tolerant systems is defined and evaluated assuming that it is known with certainty that all redundant subsystems are fault free at the start of the mission. For practical systems, this is usually not be a valid assumption. Because of the difficulty of completely testing complex systems, testing procedures usually established to a high probability (fault coverage) that the system is fault free. Unfortunately even a small possibility of a faulty subsystem at the start of the mission can very significantly effect system reliability. Our modelling approach for PSRR systems offers the important advantage that the possibility of one or more CU failures before the start of the mission can be incorporated into the reliability model. We show how this is done in this section.

Let p_{st} be the probability that a CU in the system has failed temporarily before the start of the mission. Let p_{sp} be the probability that a CU in the system has failed permanently before the start of the mission. Then the system must in in one of the following four states at the start of the mission:

a) All CU's operational
b) Two CU's operational, one temporarily failed
c) Two CU's operational, one permanently failed
d) Failed system with two or more failed CU's.

If we assume that system operation always beings with a restoration interval (which as before is assumed to be negligibly short in duration with no possibility of additional failure during this interval), then this instantaneous restoration will always take a system initially in state (b) to state (a). Therefore for all practical purposes states (a) and (b) can be grouped together to form a single state corresponding to state 1 in the Markov model of the previous sections. State (c) corresponds to state 2 and the failed system state (d) to state 3.

The probability that the system is in state 1 at the start of the mission is, therefore, given by

$$p_{I1} = (1 - p_{sp})^3 \times \{(1 - p_{st})^3 + 3p_{st}(1-p_{st})^2\}$$

The probability that the system is initially in state 2 is

$$p_{I2} = 3p_{sp}(1 - p_{sp})^2 \times (1 - p_{st})^2.$$

And the probability that the system has failed before the start of the mission (is in state 3) is given by

$$p_{I3} = 1 - p_{I1} - p_{I2}.$$

The probability that a system with these initial state probabilities fails before the completion of n C-R cycles is

$$P_{I1} P_{13}^{(n)} + P_{I2} P_{23}^{(n)} + P_{I3}$$

Therefore the system reliability to n C-R cycles, with initial failure probabilities considered, is given by

$$R'(nt_0) = 1 - P_{I1}P_{13} - P_{I2}P_{23} - P_{I3}$$

In section III, it is shown that

$$P_{13}^{(n)} = 1 + \frac{(1 - P_{13} - P_{22})}{(P_{22} - P_{11})} (P_{11})^n - \frac{P_{12}}{(P_{22} - P_{11})} (P_{22})^n$$

Also because T is 3x3 and upper triangular

$$P_{23}^{(n)} = 1 - P_{22}^{(n)} = 1 - (P_{22})^n$$

Substituting for $P_{13}^{(n)}$ and $P_{23}^{(n)}$ and simplifying we get

$$R'(nt_0) = 1 - P_{I1} - P_{I2} - P_{I3} - \frac{(P_{23} - P_{13})}{(P_{22} - P_{11})} (P_{11})^n P_{I1} + \frac{P_{12}}{P_{22} - P_{11}} (P_{22})^n P_{I1} + (P_{22})^n P_{I2}$$

The affect of possible temporary and permanent initial CU failures on system reliability is illustrated by the plots in Figure 7, which were obtained using the above expression. To isolate the contributions of each of the two types of initial failures and to also observe their combined affect, four reliability plots are displayed. These are for a system with $\lambda_t = 0.01$ per hour, $\lambda_p = 0.001$ per hour, $t_0 = 0.1$ hours and (i) no possibility of initial CU failures; $p_{st} = 0$, $p_{sp} = 0$, (ii) the possibility of only temporary initial CU failures; $p_{st} = 0.05$, $p_{sp} = 0$, (iii) the possibility of only permanent initial CU failures; $p_{st} = 0$ $p_{sp} = 0.05$, and (iv) the possibility of both temporary and permanent initial CU failures; $p_{st} = 0.05$, $p_{sp} = 0.05$.

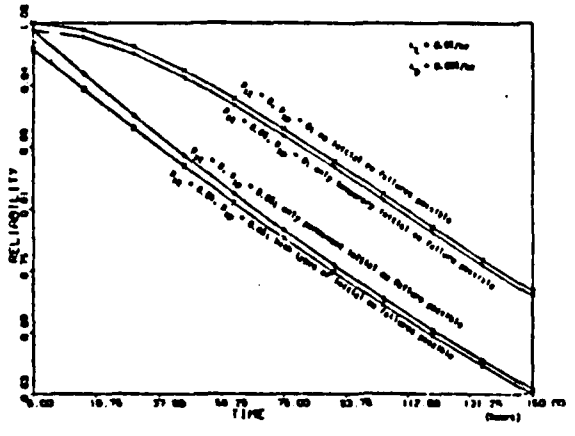


Figure 7: Reliability plots displaying the effect of initial CU failure probabilities.

The plots in Figure 7 show that both temporary and permanent initial CU failures reduce system reliability at the start of a mission. However, the possibility of a temporary initial CU failure does not have any further impact on system reliability over the rest of the mission. This is because any such failure would be restored during the first restoration interval. On the other hand a permanent initial CU failure will exist in the system for all time and continue to degrade system reliability.

Notice from the plots that for equally probably temporary and permanent initial CU failures, the permanent failures degrade the system more severely. Note also that the possibility of initial CU failures has a more significant impact on systems that call for highly reliable operations over short periods. For example, let us compare the increase in system failure probability due to a 0.05 probability of both temporary and permanent initial CU failures over that for an initial failure free system. For a mission time of 96 hours, the probability of system failure is not significantly increased if only temporary initial CU failures are allowed, and is increased by a factor of 1.5 if only permanent initial CU failures are allowed. On the other hand, for a shorter 16 hours mission time, the increase in system failure probability is much more significant. The 0.05 probability of temporary initial CU failures alone doubles the probability of system failure, while the possibility of permanent initial CU failures increases system failure probability by a factor of 7. The initial CU failure probabilities impact even more significantly on more reliable operation with shorter mission times.

System MTTF can also be derived to take into account the possibility of initial CU failures. Recall that μ_{k3} is the average number of C-R cycles that it takes for a system in state k to go to state 3 for the first time. At the start of the mission of system is in state 1, 2 or 3 with probability P_{I1}, P_{I2} and P_{I3} respectively. Therefore, the average number of C-R cycles required for the system to go to state 3 is given by $N'_{steps} = P_{I1} \mu_{13} + P_{I2} \mu_{23} + P_{I3} \mu_{33}$ Noting that $\mu_{33} = 0$ and from section IV,

$$\mu_{13} = \frac{p_{12} + p_{23}}{(1-p_{11})(1-p_{22})} \quad \text{and} \quad \mu_{23} = \frac{1}{1-p_{22}},$$

$$N'_{steps} = \frac{p_{I1}(p_{12} + p_{23})}{(1-p_{11})(1-p_{22})} + \frac{p_{I2}}{(1-p_{22})}$$

The system MTTF with initial failure probabilities considered MTTF' = $N'_{steps} \times t_0$ which is therefore given by

$$MTTF' = \frac{p_{I1}(p_{12} + p_{23})t_0}{(1-p_{11})(1-p_{22})} + \frac{p_{I2}}{(1-p_{22})}$$

Figure 8 shows a plot of MTTF' versus t_0 for a system with $\lambda_t = 0.01$ and $\lambda_p = 0.001$ and the same four sets of initial CU failure probabilities plotted in Figure 7. Note that for equally probably temporary and permanent initial CU failures, the permanent failures more significantly affect system MTTF. For the range of values of t_0 plotted, a 0.05 probability of temporary initial CU failure results in only about a 1% reduction in system MTTF, while the same probability of permanent initial CU failure results in about a 12% reduction in system MTTF. This is to be expected based on the preceding discussion which concluded that the possibility of permanent initial CU failures has a more significant impact on system reliability.

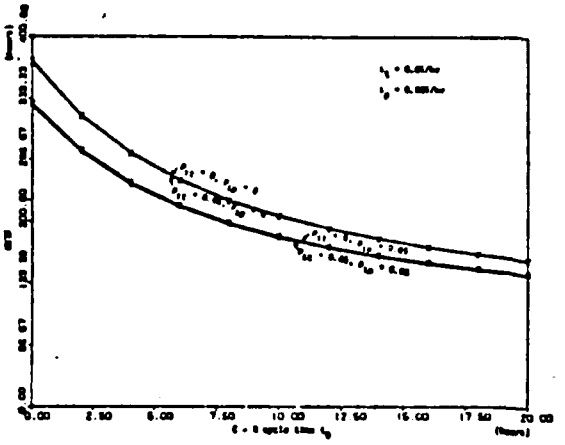


Figure 8: MTTF plots displaying the effect of initial CU failure probabilities

For the plots in Figures 7 and 8 the probability of temporary and permanent initial CU failures was taken to be 0.05. While this value may be quite pessimistic for practical fault tolerant systems, it was chosen to highlight the affects of the possibility of initial CU failures on the system. From the form of the reliability and MTTF expressions it is clear that the observed trends also hold for more realistic (smaller) initial CU failure probabilities. It appears then that the possibility of initial CU failures, particularly permanent initial CU failures, can significantly degrade the reliability of PSRR systems designed for highly reliable operation over short mission times. Their affect on system MTTF is less significant.

6. DISCUSSION

In order to maximize reliability, a fault tolerant system is usually designed with the following objectives. (We assume here that the software is error free.)

1. The hardware should be free of design errors and should meet the operational specifications of the desired system.
2. The component failure rate for the hardware should be as low as possible.
3. The fault tolerance mechanisms built into the system should protect it against the widest possible range of failures, particularly from failure modes that are most likely to occur.

It is also usually desirable to have a good quantitative reliability model for the system so that the risk of failure can be realistically considered when deploying the system.

Due to the difficulty in completely testing and validating complex hardware, objective (1) is usually best achieved by implementing the system out of proven off the shelf building blocks, with minimal specialized low level design. Such an approach has the significant additional advantage of low hardware and design costs. PSRR systems clearly meet this objective. Except for the fault tolerant clocking circuit (which all hardware redundancy designs require) no additional specialized hardware is needed. The vote is conducted by the processors themselves, with communication between the CU's taking place through conventional I/O ports. On going research indicates that robust distributed restoration algorithms can be developed for general N redundant PSRR system that will purge out permanently failed CU's from the vote and thereby also provide the reliability advantages of dynamic redundancy. In contrast, redundant systems employing concurrent error masking require voting circuits. The design of the voters for C.vmp type systems is non-trivial [17] because timing and control signals between the processors and memories must also be voted on and correctly synchronized. Dynamic redundancy designs such as the SIFT and FTMP require even more extensive specialized design to implement the fault detection and reconfiguration mechanisms.

The objective (2) of low failure rates in the hardware is also met by employing proven components. Further, in an VLSI technology, failure rates are largely a function of the number of IC's in the design and their interconnection complexity. To prevent a single fault from causing system failure, it is usually not desirable to implement highly relia

able redundant systems on a single chip. It seems optimal from a reliability standpoint to implement a N-redundant system using N IC's such that no two copies of a circuit are on the same chip. Figure 2 indicates that PSRR systems are well suited for such an implementation, with each CU (which is basically a conventional microcomputer) being realized on a different chip. A three chip implementation of the C.vmp architecture in Figure 1 would require more complex IC's and significantly greater interconnect complexity, probably resulting in higher failure rates.

Next, let us consider objective (3) which addresses the types of failure that are handled by the fault tolerance mechanisms. While also providing protection against permanent faults, PSRR systems are paticularly effective in providing recovery from upsets caused by transient faults. This is significant because a large majority of computer system "crashes" are believed to be caused by transients. Systems that do not provide recovery from such upsets are very inefficient in their use of redundant hardware. Of course, PSRR systems, like other redundant systems, remain vulnerable to simultaneous upsets in multiple units. To provide protection against such failures, we are currently studying time redundancy techniques. It appears that these techniques can be easily integrated into the PSRR approach of periodically voting on redundant system states.

A major concern in the implementation of ultra reliable systems is hardware design faults. Because such faults may often only manifest themselves under unusual combinations of system state, inputs and environmental conditions, they are difficult to test for and can rarely be completely eliminated in todays complex systems. Further, the unreliability introduced in a system by design faults is difficult to bound. For ultra-reliable designs, this can render reliability estimation meaningless. Unfortunately, many of the redundant designs presently deployed do not provide any protection from design faults because the resulting failures are likely to simultaneously effect all the redundant units. The PSRR design offers some improvement in this regard because it can be implemented almost entirely from proven high volume off the shelf hardware. This will allow a system designer to obtain the redundant copies of the hardware from as diverse manufacturing sources as possible. Such hardware is less likely to have common model failures and will thus provide some protection from design faults.

Finally, as we have shown in this paper for triple redundant systems, PSRR systems can be nicely modeled for reliability estimation. This should allow reliability-redundancy (cost) trade-offs to be computed while PSRR systems are being designed and facilitate the systematic design of such systems to desired reliability specifications.

7. CONCLUDING REMARKS

The PSRR design presented in this paper offers a viable and attractive approach for implementing fault tolerance in VLSI based microcomputer systems. The fault-tolerance capability of such a system depends on the level of redundancy employed. In the proposed design, protection against permanent failures is provided by error masking, and the purging of permanently failed modules. Erronous signals from modules upset by transient failures are also masked out and the modules periodically restored to the correct operational state so as to allow the system to withstand further failures.

PSRR systems can be largely implemented from off the shelf components and require minimal specialized design. They are particularly well suited for implementation from VLSI modules. Thus, they offer low costs and ease of design. The reliability models for such systems presented in this paper and in [15] should allow the systematic design of such systems to desired reliability specifications.

REFERENCES

1. A. Avizienis, "Design of Fault Tolerant Computers", Proc. AFIPS 1976 FJCC, Vol. 31, pp. 733-743, 1967.

2. F. P. Mathur and P. T. DeSousa, "Reliability Models of NMR Systems", IEEE Transactions on Reliability, Vol. R-24, pp. 108-113, 1975.

3. R. A. Short, "The attainment of reliable digital systems through the use of redundancy--A survey", IEEE Computer Group News, Vol. 2, pp. 2-17, March 1968.

4. F. P. Mathur, and A. Avizienis, "Reliability Analysis and Architecture of a Hybrid-Redundant Digital System", AFIPS Conf. Proc., Vol. 36, pp. 375-383, May 1970.

5. P. T. deSousa and F. P. Mathur, "Sift-Our Modular Redundancy", IEEE Transactions on Computers, Vol. C-27, No. 7, pp. 624-627, July 1978.

6. J. Losq, "Highly Efficient Redundancy Scheme: Self-Purging Redundancy", IEEE Transactions on Computers, Vol. C-25, pp. 569-578, June 1976.

7. A. E. Cooper and W. T. Chow, "Development of on-board Space Computer Systems", IBM Journal of Research Development, Vol. 20, No. 1, pp. 5-19, January 1976.

8. J. H. Wensley, et. al., "SIFT: Design and Analysis of a Fault-Tolerant Computer for Aircraft Control", Proceedings of the IEEE, Vol. 66, No. 10, Oct. 1978, pp. 1240-1255.

9. A. L. Hopkins, T. Basil Smith and J. H. Lala, "FTMP - A Highly Reliable Fault Tolerant Multiprocessor for Aircraft", Proceedings of the IEEE, Vol. 66, No. 10, October 1978.

10. R. S. McConnel, D. P. Siewiorek and M. M. Tsao, "The Measurement and Analysis of Transient Errors in Digital Computer Systems", Proc. Ninth Annual International Symposium on Fault Tolerant Computing, IEEE, New York, June 1979, pp. 67-70.

11. J. R. Sklaroff, "Redundancy Management Technique for Space Shuttle Computers", IBM Journal of Research and Development, Vol. 20, No. 1, pp. 20-28, January 1976.

12. J. F. Wakerly, "Microcomputer Reliability Improvement Using Triple Modular Redundancy", Proceedings of the IEEE, Vol. 64, No. 6, June 1976.

13. D. P. Siewiorek, V. Kini, H. Mashburn, S. R. McConnel and M. Tsao, "A Case Study of C. mmp, Cm*, and C. vmp: Part I--Experiences with Fault Tolerance in Multiprocessor Systems", Proceedings of the IEEE, Vol. 66, No. 10, pp. 1178-1199, October 1978.

14. J. J. Stiffler, et al., "CARE III Final Report Phase I", Vol. 1 and 2, NASA CR-159123, 1979.

15. A. D. Singh "The Design of Periodically Self Restoring Redundant Systems" Ph.D. Dissertation, Virginia Polytechnic Institute and State University, December 1982.

16. D. Davies and J. F. Wakerly: "Synchronization and matching in redundant systems", IEEE Transactions on Computers, Vol. C-27, No. 6, June 1978.

17. S. R. McConnel, and D. P. Siewiorek, "Synchronization and Voting", IEEE Transactions on Computers, Vol. C-30, No. 3, pp. 161-164, February 1981.

18. F. G. Frison, and J. H. Wensley, "Interactive Consistency and Its Impact of the Design of TMR Systems", Proceedings 12th International Symposium on Fault Tolerant Computing, pp. 228-233, June 1982.

19. D. L. Isaacson and R. W. Madwin, Markov Chains--Theory and Applications, John Wiley & Sons, New York, 1976.

20. C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill, New York, 1968.

END

DTIC

8-86