WORD-VOTER: A NEW VOTER DESIGN FOR TRIPLE MODULAR REDUNDANT SYSTEMS

Subhasish Mitra and Edward J. McCluskey
Center for Reliable Computing
Departments of Electrical Engineering and Computer Science
Stanford University, Stanford, California

ABSTRACT

Redundancy techniques are commonly used to design dependable systems to ensure high reliability, availability and data integrity. Triple Modular Redundancy (TMR) is a widely used redundancy technique that masks faults. In a TMR system, we have three implementations of the same logic function and their outputs are voted using a voter circuit. In this paper, we present a new voter design called the Word-Voter that has some distinct advantages over the bit-by-bit voting schemes used in conventional TMR systems. This paper demonstrates the usefulness of the word-voter design in increasing the data integrity (reducing the probability of corrupt outputs) of TMR systems. The area and delay overhead of the word-voter design is compared to that of the bit-by-bit voter. An efficient design of a TMR-Simplex system using the word-voter is also presented.

1. INTRODUCTION

Redundancy techniques such as duplication and Triple Modular Redundancy are commonly used for designing dependable systems to ensure high reliability, availability and data integrity. *Triple Modular Redundancy* (TMR) [Von Neumann 56] is an example of a redundancy scheme that is used for fault-masking. In a TMR system, we use three (same or different) implementations of the same logic function and the outputs of all the implementations are connected to a voter (Fig. 1.1). There are numerous examples of dependable systems using the TMR technique [Siewiorek 92].

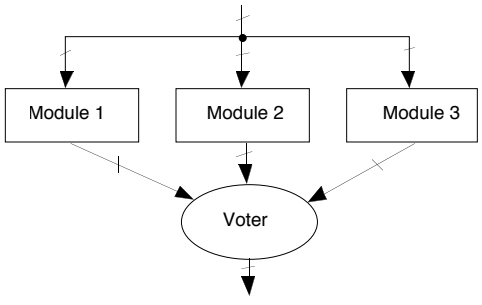

Figure 1.1. Triple Modular Redundancy

For voting on the outputs of the individual modules, majority voting circuits are generally used in TMR systems. Figure 1.2 shows the design of a majority voting circuit. In Fig. 1.2, Z_1^1, Z_1^2 and Z_1^3 are the outputs corresponding to the bit position Z_1 of the three modules of the TMR system. The corresponding voted output bit of the system is Z_1.

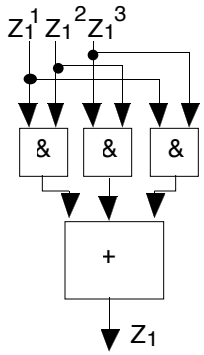

Figure 1.2. A majority voting circuit

In TMR systems, majority voting is normally performed on a bit-by-bit basis [Siewiorek 92]. For a system with n outputs, conventional TMR systems use n single-bit voters. Figure 1.3 shows the implementation of such a TMR system with two outputs.

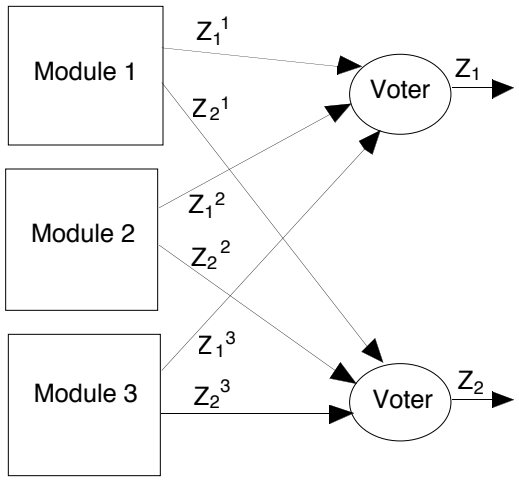

Figure 1.3. A TMR system with bit-wise voting

There is a large body of literature on reliability modeling of TMR systems [Trivedi 82]. For the classical

TMR system shown in Fig. 1.1, the reliability R is given by the following expression.

$$R = R_m^3 + 3R_m^2(1 - R_m)$$

In the above expression, R_m is the reliability of each individual module in the TMR system. The expression follows from the fact that for the system to produce correct outputs, at least two of the three modules must produce correct outputs.

The classical reliability expression for TMR systems is optimistic because it does not consider common-mode failures. Lala observed that we must pay attention to the problem of common-mode failures (CMFs) [Lala 94]. CMFs result from failures that affect more than one module of the redundant system at the same time, generally due to a common cause. They can be design faults or operational faults due to external (such as EMI and radiation) or internal causes. For example, a radiation source causing multiple-event upsets [Reed 97] may lead to the failure of more than one module in a TMR system. There is no built in facility in conventional TMR voters to detect this situation and initiate appropriate actions.

In this paper, we describe a new voter design for TMR systems called the Word-Voter. This design is especially useful in the context of common-mode and multiple failures that affect multiple modules in a TMR system. The word-voter design is described in Sec. 2. The design is applicable to TMR systems with multiple outputs. The advantages of using the word-voters in TMR systems are discussed in Sec. 3. The additional area and delay overhead of the word-voter design compared to the conventional bit-wise voter is also analyzed in Sec. 4. In Sec. 5, we explain the design of TMR-simplex systems using the word-voter design. Section 6 discusses some interesting problems involving TMR systems with the word-voter. Finally, we conclude in Sec. 7.

2. WORD-VOTER DESIGN

This section presents the word-voter design for TMR systems. For motivation, let us consider the following example. Consider a TMR system with two outputs. Suppose that a failure source causes faults in the first and the second modules of the system. Due to the presence of the fault in Module 1, in response to a particular input combination, the module produces an output combination 10 instead of 01. Similarly, due to the presence of the fault in Module 2, the output combination obtained from it is 11. Finally, Module 3 is fault-free and produces the expected output 01. This is shown in Table 2.1.

With bit-wise voting, the voter corresponding to the first output bit produces a 1 and the one corresponding to the second output bit produces a 1. Thus, we have 11 at the system output. However, if we consider the *output word* from each module, we find that the output words from all the three modules are different. The output words from the first, second and third modules are 10, 11 and 01, respectively. This can be treated as an erroneous condition for a voter (which works on a majority voting principle) because, no two output *words* are equal. Based on this observation, we can modify the classical voter design by adding some extra circuitry that detects this error condition and produces an error. Thus, the TMR system with the word-voter is capable of maintaining data integrity for this example. In response to the error signal, appropriate action (depending on the application) can be initiated.

While the word-voting scheme is frequently used in software [Gersting 91], for TMR systems in hardware, majority voting is normally performed on a bit-by-bit basis. Figure 2.1 shows an example of a TMR system with two outputs using the word-voter.

Table 2.1. Illustration of voting in a TMR system

Module	Fault-free Outputs	Faulty Outputs
1	0 1	**1 0**
2	0 1	**1** 1
3	0 1	0 1
Bit-wise Voting	0 1	**1** 1
Word-wise Voting	0 1	No majority

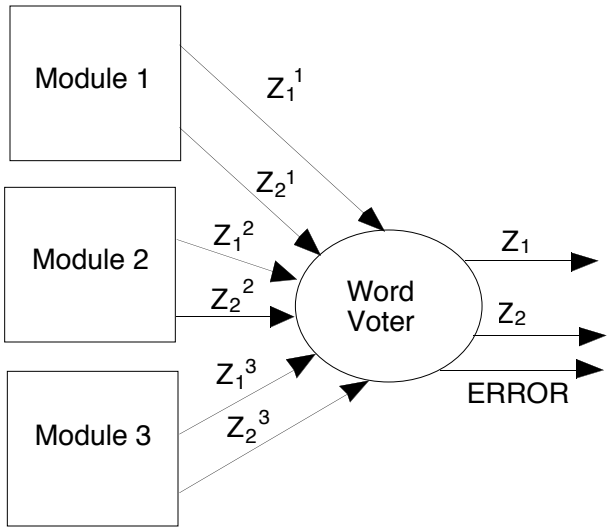

Figure 2.1. A TMR system with word-voter

Figure 2.2 shows the design of the word-voter for a TMR system where each module has n outputs. Let Z^i denote the output *vector* from the i^{th} module. Also, let Z_j^i denote the j^{th} bit of Z^i. In Fig. 2.2a, we show the matching circuit $Match_{i,j}$ which produces a 1 if and only if modules i and j produce the same output word. Following our example in Table 2.1, for a particular input combination, the three modules produce the following outputs: $Z_1^1 = 1$, $Z_2^1 = 0$, $Z_1^2 = 1$, $Z_2^2 = 1$ and $Z_1^3 = 0$, $Z_2^3 = 1$. Referring to Fig. 2.2, $Match_{1,2} = Match_{1,3} = Match_{2,3} = 0$ and the *ERROR* signal equals 1.

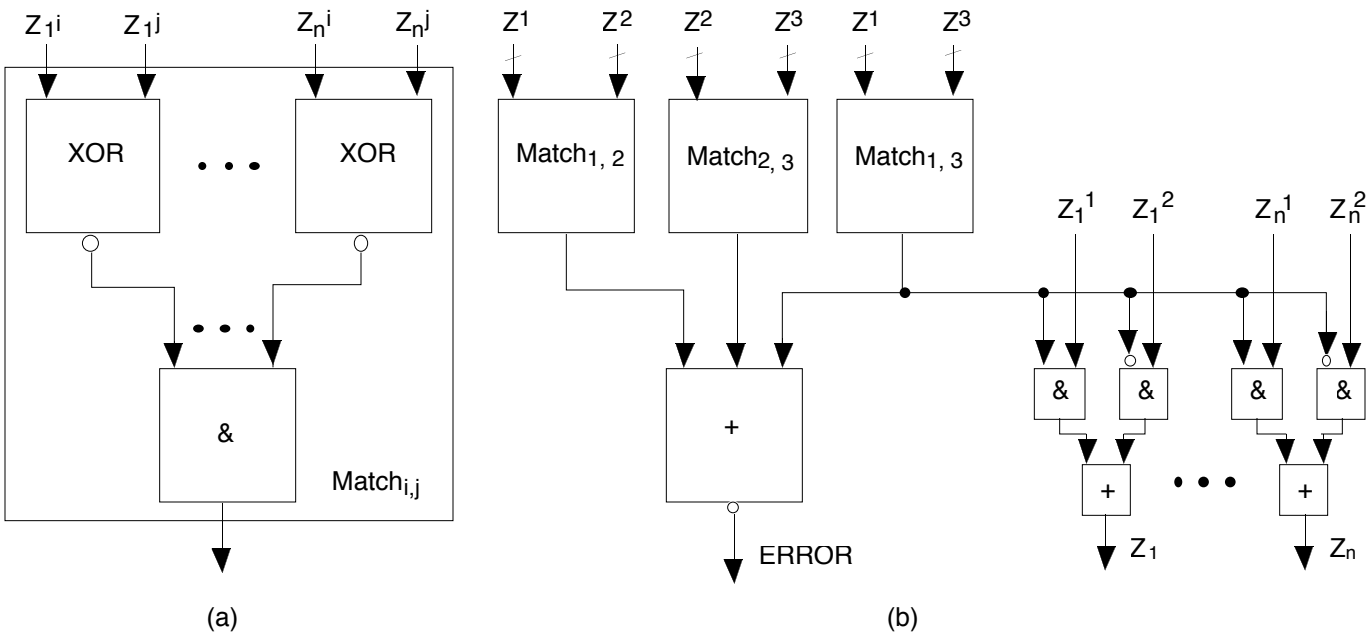

Figure 2.2. Word-Voter for a system with n outputs Z_1, …, Z_n. (a) Matching circuit $Match_{i,j}$. (b) The complete design.

Let us now consider a situation where the output words from at least two modules match. In that case, the *ERROR* signal will not be equal to 1. If the output words from modules 1 and 3 match, the TMR produces an output word that is the same as the one produced by module 1. If the output words from modules 1 and 3 do not match (i.e., $Match_{1,3} = 0$), then the output word produced by the word-voter is equal to the one produced by module 2.

Use of median voters has been proposed in the past for voting purposes in TMR systems [Dennis 74][Wong 96]. In a TMR system, a median voter selects the median of the output words from the three modules to be the output of the TMR system. For that purpose, the voter performs a magnitude comparison of the output words from the three modules. A median voter can be easily modified to operate as a word-voter. In that case, the median voter will generate an error signal if the magnitudes of the outputs of the three modules are different. Since a median voter performs magnitude comparison, its area overhead is generally more than the design in Fig. 2.2.

3. ADVANTAGES OF THE WORD-VOTER

In this section, we discuss the advantages of the word-voter design. As illustrated in Sec. 2, with the word-voter, we can detect erroneous conditions that can otherwise produce incorrect outputs. This enhances the data integrity of a TMR system. For quantifying the data integrity enhancement, we use a metric *s*. The *metric s* is defined as the probability that a TMR system with the word-voter will produce an error signal at least once before producing an incorrect output combination, in the presence of multiple module failures. In order to estimate *s*, we performed the following simulation experiments.

Two implementations of the ALU181 circuit [TI 76] and ISCAS c432 and c6288 were used for simulation purposes. We replicated three copies of a given circuit to form a TMR system. We used the word-voter for voting purposes. For each of these TMR systems, we performed 100,000 simulation runs. In each simulation run, starting from time instant 0, for each module, we generated a binary random variable to decide whether a fault is going to be injected into that module. If the random variable had a value 1, a random fault was inserted in the corresponding module. We considered only permanent stuck-at faults. We used an LFSR to generate the input sequences. We continued the process of injecting faults and applying input combinations until the TMR system produced an output word that was different from the correct output word. We estimated *s* as the percentage of cases for which the word-voter generated the error signal at least once before the system generated incorrect outputs. This number is reported in the fourth column of Table 3.1 for each of the TMR systems simulated.

Table 3.1. Simulation results

Circuit Name	# Inputs	# Outputs	Estimated value of s
alu181	14	8	92.2 %
nand181	14	8	92.5 %
c432	36	7	94 %
c6288	32	32	99 %

For example, consider the case of the ISCAS circuit c6288. The fourth column of Table 3.1 indicates that in the presence of multiple module failures, for 99% cases the system will produce the *ERROR* signal, at least once, before an incorrect output is produced. This means that,

for 99% of the cases, the system output will not be corrupt if we initiate repair action in response to the first *ERROR* signal indicated by the word-voter.

It may be observed that the estimated value of s decreases with decreasing number of outputs. This observation is also intuitively true because, the more outputs, the more is the possibility that the output words from the three copies are different. Note that if there is a single module fault, there will be no *ERROR* signal.

3.1. DATA INTEGRITY OF A TMR SYSTEM WITH WORD-VOTER

In this section, we analyze the data integrity of a TMR system with the word-voter. By *data integrity of a TMR system*, we mean the probability that the system does not produce corrupt (incorrect) outputs before indicating an erroneous situation in any of the previous cycles. Here, we present a simple analysis of the data integrity of a TMR systems using the word-voter. More sophisticated analysis can be performed by extending the analysis of redundant systems presented in [Mitra 99].

For our analysis, we consider a discrete time model of the system. In this model, time is divided into discrete increments called cycles and inputs are applied at the beginning of each cycle. Consider a TMR system with a word-voter where the probability that a module produces correct outputs in any given cycle is R_m. In addition, let us suppose that s is the probability that the word-voter in the system produces an error signal at least once before producing corrupt (incorrect) outputs when two modules fail. In this case, the data integrity of the system up to time T is given by the following expression:

$$R_m^{3T} + 3R_m^{2T}[1-(1-R_m)^T] + 3sR_m^T[1-(1-R_m)^T]^2$$

The above expression follows from the fact that when at least two modules are working correctly, the system always produces correct outputs. However, when two modules are faulty, data integrity is maintained when the system produces the error signal at least once before producing corrupt outputs. Hence, the factor s appears in the third term of the above expression.

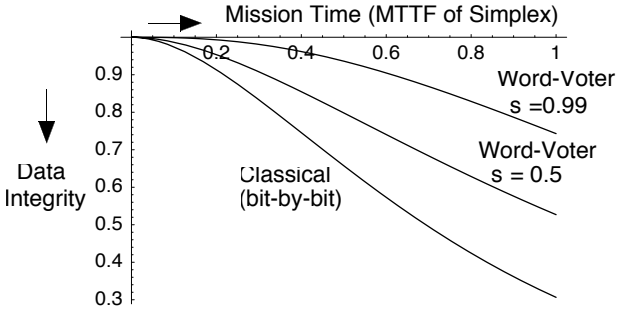

Figure 3.1. Data Integrity of a TMR System with the Word-Voter

In Fig. 3.1 we plot the data integrity of a TMR system with the word-voter for different values of s. The simplex reliability (R_m) is equal to $1 - 10^{-12}$. On the X-axis we measure the mission time of the system and 1 unit of time corresponds to the MTTF (Mean Time to Failure) of a simplex system (consisting of only one module). It is clear from Fig. 3.1 that the data integrity improves at least by an order of magnitude for increasing values of s. A TMR system with the classical bit-by-bit voter has s equal to 0. This shows the effectiveness of using word-voters for designing TMR systems with high data integrity.

As discussed earlier, the word-voter produces an error signal when the three modules of a TMR system produce different output combinations. However, that does not mean that the TMR system will always produce incorrect outputs in this situation. This is illustrated using the example in Table 3.2.

Table 3.2. Correct outputs from a TMR system in the presence of multiple module failures

Module	Fault-free Outputs	Faulty Outputs
1	0 0 0	**1** 0 0
2	0 0 0	0 **1** 0
3	0 0 0	0 0 0
Bit-wise	0 0 0	0 0 0
Word-wise	0 0 0	No majority

In the TMR system of Table 3.2, all the three modules are supposed to produce the output combination 000 in the fault-free case. However, due to the presence of faults in the first and the second modules, Module 1 and Module 2 produce output vectors 100 and 010, respectively. Although the output vectors from the three modules are different, a bit-by-bit voting scheme will produce the correct combination 000 at the system output. This situation is an example of a compensating fault, which has been studied in [Siewiorek 75]. Note that, our word-voter design will indicate an error signal in this case. Thus, with the word-voter design, error signal may be indicated for some compensating faults in the system.

4. OVERHEAD CALCULATION

Compared to a TMR system with a bit-by-bit voter, the extra hardware needed by a TMR system with the word-voter is of the order of one 2-input logic gate and three XNOR gates for each output of the system. This is because, for a TMR system with n outputs, the number of equivalent 2-input gates required by the word-voter and the bit-wise voter are $6n + 2$ and $5n$, respectively. In addition, the TMR system with the word-voter needs $3n$ XNOR gates. The derivation is shown in the next paragraph.

For a TMR system with n outputs, the word-voter design of the Fig. 2.2, the three matching circuits require $3n$ XNOR gates, 3 n-input AND gates (equivalent to $3n$ 2-input AND gates). In addition, one 3-input NOR gate is needed to generate the error signal and $2n$ 2-input AND gates and n 2-input OR gates are required to produce the TMR system outputs. The equivalent number of 2-input (AND/OR/NAND/NOR) gates, excluding the $3n$ XNOR gates, is $6n + 2$. The gates required by n simple bit-by-bit majority voting circuits are $3n$ 2-input AND gates and n 3-

input OR gates — the number of equivalent 2-input gates is $5n$. For the word-voter design of Fig. 2.2, the delay in generating the final voted output is of the order of $log_2 n$ 2-input gate delays (due to the presence of n-input AND gates in the matching circuits).

The word-voter design can be modified to guarantee that the data outputs (but *not* the *ERROR* signal) will be produced without any extra gate delay compared to conventional TMR systems. In this case, we can use the conventional bit-by-bit voters to produce the voted outputs. For generating the *ERROR* signal, we can use the part of the logic circuit of Fig. 2.2b that generates the error signal. In this case, the extra area overhead of the word-voter design over a bit-by-bit voter is 3 n-input AND gates, $3n$ XNOR gates and 1 3-input NOR gate for a TMR system with n outputs. This is equivalent to $3n$ 2-input gates and $3n$ XNOR gates.

It may be noted that, for a hybrid redundant n-output system with TMR core [Siewiorek 92] the overhead of $3n$ XNOR gates in the word-voter design disappears.

5. TMR-SIMPLEX SYSTEM WITH WORD-VOTER

The MTTF (Mean Time To Failure) of a TMR system is lower than that of the corresponding simplex system (containing only one module). Moreover, for mission time greater than seven-tenths of the MTTF of a simplex system, the TMR reliability is lower than the reliability of the corresponding simplex system. These problems can be overcome by using a TMR-Simplex system. In a TMR-Simplex system, we have full TMR as long as all the three modules are functioning correctly. As soon as one of the modules becomes faulty (disagrees with the outputs of the other two modules), the system switches to simplex mode and only *one* fault-free module is chosen to produce outputs for subsequent operations. Reliability analysis of TMR-Simplex systems has been studied in [Trivedi 82].

The word-voter design of Fig. 2.2 can be used to efficiently design TMR-Simplex systems as shown in Fig. 5.1. There are three flip-flops in the design and they are all clocked by the same clock signal (not shown in Fig. 5.1). Two of the flip-flops have clock-enable (aka gated clock, load-enable) inputs (*EN*) as shown in Fig. 5.1. The flip-flops can be set or reset by an asynchronous retry signal (not shown in Fig. 5.1). The design is somewhat similar but has less logic complexity compared to Sift-Out Modular Redundancy [De Sousa 78].

Initially, when all the modules produce correct outputs, the outputs of module 1 are produced at the voter outputs. If one of the modules (Module 1, for example) fails, $Match_{1,3}$ produces a 0 and $Match_{2,3}$ produces a 1. As a result, all the flip-flops are reset to 0, and the outputs of Module 2 are produced at the voter output. Thus, the system behaves as a simplex system consisting of Module 2 only. The *ERROR* signal will not be produced any more from this time onwards. If Module 2 fails, then $M_{1,3}$ will produce a 1 and $M_{2,3}$ will produce a 0 and the system will behave as a simplex system consisting of Module 1 only. If the output vectors produced by the three modules are all different, the *ERROR* signal will be equal to 1.

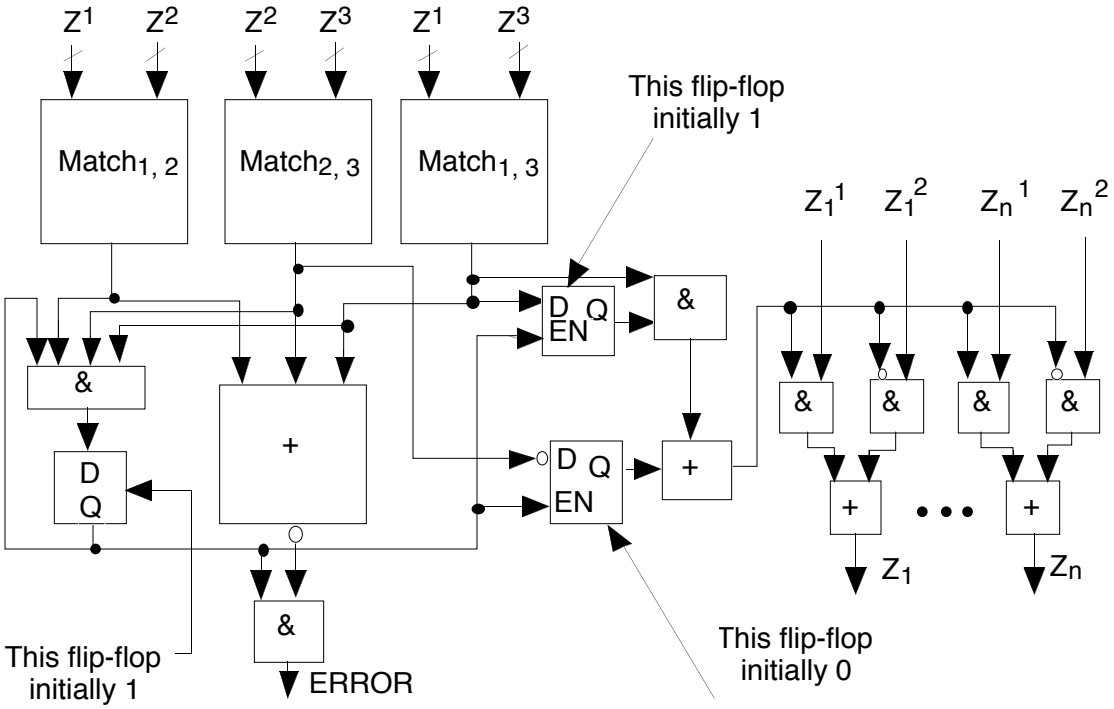

Figure 5.1. A TMR-Simplex system using a word-voter

Note that, compared to the design in Fig. 2.2, we need three extra AND gates (equivalent of five two-input AND gates), one extra two-input OR gate and three flip-flops, *independent of the number of outputs*. This shows the benefit obtained by using our word-voter in designing a TMR-simplex system.

6. INTERESTING PROBLEMS WITH THE WORD-VOTER DESIGN

This section presents several interesting problems that can be studied in the context of TMR systems with word-voters. In [Mitra 00] we studied the problem of common-mode failures (CMF) in redundant systems. We have developed a CMF model and presented a technique to design redundant systems that are protected against these modeled CMFs. The technique takes the advantage of the word-voter design and ensures that the designed TMR systems either produce correct outputs or indicate error situations when incorrect outputs are produced in the presence of the modeled CMFs. This helps us design robust TMR systems with high data integrity against the modeled common-mode failures.

Using the word-voter, we can design TMR systems for which we perform word-wise voting for only a subset of the output bits and bit-wise voting for the remaining output bits. It will be interesting to characterize the relationships among the subsets of the outputs and the system reliability and data integrity.

For systems with round-off errors, there is a possibility of non-exact matching of the outputs of the three modules. In that case, an interesting problem will be to determine thresholds. Then we can indicate error situations using the word-voter design if the outputs of the three modules do not match within the threshold.

7. SUMMARY AND CONCLUSIONS

In this paper we presented a word-voter design for TMR systems. We demonstrated that, with the word-voter, the data integrity of a TMR system improves at least by an order of magnitude over conventional systems with bit-by-bit voting. Thus, the word-voter has enhanced capabilities that can be used to design TMR systems that are protected against common-mode and multiple module failures. The area and delay overhead of the new voter design has been analyzed in this paper. We also described a technique to efficiently design a TMR-Simplex system using the word-voter. Looking at the future, many interesting problems related to redundant systems with word-voters can be studied. One of these problems that has been studied in [Mitra 00] is to develop CMF models and design TMR systems in such a way that the modeled CMFs can be detected using the enhanced capabilities of the word-voter.

8. ACKNOWLEDGMENTS

This work was supported by Defense Advanced Research Projects Agency (DARPA) under Contract No. DABT63-97-C-0024. Thanks to Dr. Nirmal Saxena of Stanford Center for Reliable Computing for his helpful comments.

9. REFERENCES

[De Sousa 78] De Sousa, P. T., and F. P. Mathur, "Sift-Out Modular Redundancy," *IEEE Trans. Computers*, Vol. C-27, No. 7, pp. 624-627, July 1978.

[Dennis 74] Dennis, N. G., "Ultra-reliable Voter Switches with a Bibliography of Mechanization," *Microelectronics and Reliability*, pp. 299-308, August 1974.

[Gersting 91] Gersting, J. L., *et al.*, "A Comparison of Voting Algorithms for N-Version Programming," *Proc. Annual Hawaii Intl. Conf. on System Sciences*, pp. 253-262, 1991.

[Lala 94] Lala, J. H. and R. E. Harper, "Architectural Principles for Safety-critical Real-time Applications," *Proc. of the IEEE*, Vol. 82, No. 1, pp. 25-40, Jan. 1994.

[Mitra 99] Mitra, S., N. R. Saxena and E. J. McCluskey, "A Design Diversity Metric and Reliability Analysis for Redundant Systems," *Proc. Intl. Test Conf.*, pp. 662-671, 1999.

[Mitra 00] Mitra, S., and E. J. McCluskey, "Design of Redundant Systems Protected Against Common-Mode Failures," *Technical Report, Center for Reliable Computing, Stanford Univ., CRC-TR-00-2*, 2000. (http://crc.stanford.edu)

[Reed 97] Reed, R., *et al.*, "Heavy Ion and Proton-Induced Single Event Multiple Upset," *IEEE Trans. on Nuclear Science*, Vol. 44, No. 6, pp. 2224-2229, July 1997.

[Siewiorek 75] Siewiorek, D. P., "Reliability Modeling of Compensating Module Failures in Majority Voted Redundancy," *IEEE Trans. Computers*, Vol. 24, No. 5, pp. 525-533, 1975.

[Siewiorek 92] Siewiorek, D. P. and R. S. Swarz, *Reliable Computer Systems: Design and Evaluation*, Digital Press, 1992.

[TI 76] *The TTL Data Book*, Texas Instruments, 1976.

[Trivedi 82] Trivedi, K.S., *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice Hall, Englewood Cliffs, NJ, USA, 1982.

[Von Neumann 56] Von Neumann, J., "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," *Automata Studies*, *Ann. of Math. Studies*, no. 34, C. E. Shannon and J. McCarthy, Eds., Princeton University Press, pp. 43-98, 1956.

[Wong 96] Wong, K., *et al.*, "A VLSI Median Voter for Fault Tolerant Signal Processing Applications," *Proc. Intl. Conf. Signal Processing*, pp. 1574-1577, 1996.