Sift-Out Modular Redundancy

PAULO T. DE SOUSA, MEMBER, IEEE, AND FRANCIS P. MATHUR

***Abstract*—A fault-tolerance technique for digital systems, Sift-Out Modular Redundancy, is proposed and designed. An appropriate number of identical channels are provided for each module. The number of channels depend upon the particular application, and all channels are active as long as they are fault-free. Upon the failure of a channel, its contribution to the module output ceases. The configuration tolerates up to $L - 2$ channel failures, if L is the initial number of channels. Sift-out redundancy is easy to implement, and shows several advantages when compared to existing redundancy techniques.**

***Index Terms*—Fault-tolerant computing, modular redundancy, reliability, responsive structure, restoring organ, sift-out modular redundancy.**

INTRODUCTION

FAULT-TOLERANT computing can be achieved by modular redundancy. The item to be made fault-tolerant is replicated a number of times. The identical channels created are organized in an ultrareliable structure. This structure guards against the effects of some types of faults expected to occur in the computer. Fault-masking and spare-switching have been the two most common forms of modular redundancy.

Redundant structures providing fault-masking are usually called static, massive, or masking; all channels are active throughout the mission time. The failing of a channel is "masked" by the good channels, keeping the overall structure output correct. The maximal number of channels that can fail without disrupting the masking process is the fault tolerance F. Majority voting logic (TMR [1], NMR [2]), provides examples of massive structures.

Structures that use spare-switching contain two categories of channels: the active channels that make up the functional core of the structure, and the passive or spare channels. The spare channels are in a standby or dormant state. Whenever a functionally active channel fails, a passive channel is activated and replaces it. Dynamic, replacement, and selective redundancy are names given to this type of structure. Standby [3], serial/parallel [4], and hybrid [5], [2], are examples of selective structures.

Recently, new redundancy schemes have been proposed that do not quite fit in any of the two divisions just mentioned. In this third division of redundant structures, there are no standby channels. All the channels are active at the beginning of the mission time. However, upon the occurrence of a failure, the structure reconfigures itself in such a way that the contribution of the failed channel is reduced or eliminated. TMR/Simplex [6], NMR/Simplex [7], and some adaptive schemes [8] are examples of this type of structure.

The structure proposed in this paper is to be included in this group of responsive redundant structures. It will be referred to as Sift-Out Redundancy, and several advantages over existing techniques will be shown.

SIFT-OUT REDUNDANCY

Scheme

When using a sift-out configuration, the system is organized into L identical channels where L is any integer. The channels are synchronized with one another and perform simultaneous operations. Each channel is active as long as it is fault-free. Whenever one of the channels fail, its contribution to the system output ceases. The system becomes an $(L - 1)$ redundancy scheme. Upon the occurrence of a new failure, the process repeats itself.

Sift-out redundancy has a fault tolerance $F = L - 2$. $(L - 2)$ channels can fail and the module will still operate correctly. When the module is reduced to two channels and one of them fails, the system is unable to detect which one failed. Sift-out is a 2-out-of-L structure, or more emphatically, an L-down-to-two redundancy.

Implementation

To implement a sift-out redundant structure, a restoring organ is placed at the outputs of the L channels. In the following implementation, it is assumed that no more than one channel may fail at the same instant of time. The restoring organ compares the output signals. If one of the signals disagrees with the others, the corresponding channel is "sifted out." The signal of the "good" channels is selected without need for voting. The diagram of Fig. 1 shows the main components of the restoring organ which include the comparator, detector, and collector.

The comparator is a set of $\binom{L}{2}$ EXCLUSIVE OR gates. This is illustrated in Fig. 2 where $L = 4$.

The detector (see Fig. 3) is a sequential circuit with $[\binom{L}{2} + L]$ NOR gates. The signal F_i is equal to zero when channel i is fault-free. F_i is equal to 1 when channel i has failed. For example, let channel 1 be the first channel to fail. It will disagree with the other channels causing lines E_{12}, E_{13}, and E_{14} to hold a logical value 1. Line F_1 will then be set to 1, and the feedback loop will force it to stay that way. A

Manuscript received October 3, 1975; revised October 1977.

P. T. de Sousa was with the Department of Electrical Engineering, University of Luanda, Luanda, Angola. He is now with the Advanced Systems Development Department, Collins Commercial Telecommunications Group, Rockwell International, Dallas, TX 75081.

F. P. Mathur was with the Computer Science Section, Department of Mathematics, Wayne State University, Detroit, MI 48202. He is now with the Computer Science Section, California State Polytechnic University, Pomona, CA 91766.

0018-9340/78/0700-0624$00.75 © 1978 IEEE

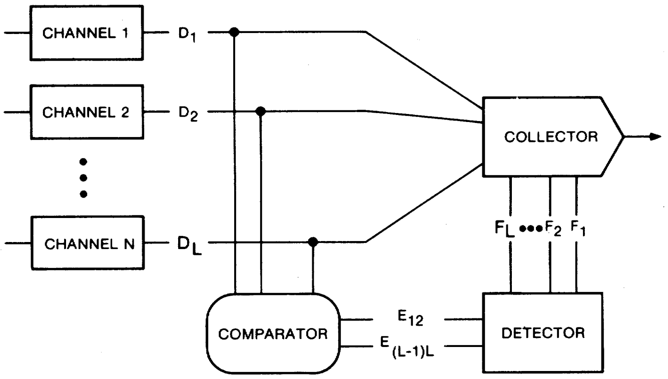


Fig. 1. Sift-out redundancy.

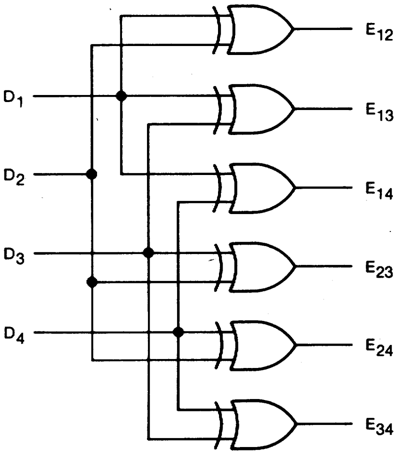


Fig. 2. Comparator for a 4-channel sift-out redundancy.

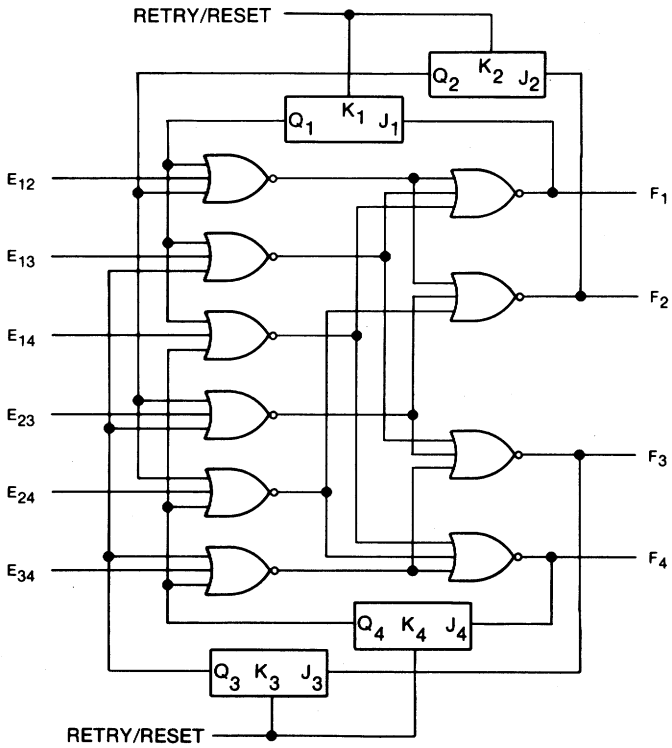


Fig. 3. Detector for a 4-channel sift-out redundancy.

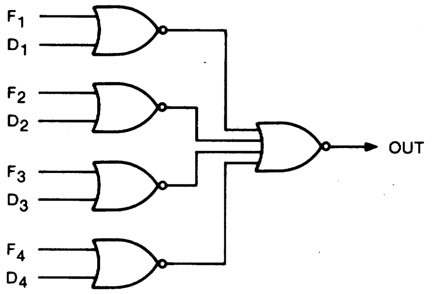


Fig. 4. Collector for a 4-channel sift-out redundancy.

clocked flip-flop is added in each feedback loop to prevent race conditions and to provide a reset/retry procedure. Such flip-flops make the structure tolerant to transient failures and facilitate initial checkout.

The final step is the collector, with $(L + 1)$ NOR gates. Each good channel feeds one input $(\bar{D}_i)$ to the last NOR gate. Each bad channel provides a logical value 0 as input to the last gate. The output of this gate is the correct output of the system, provided that at least two channels are good. Fig. 4 shows the collector when $L = 4$.

Comparison with Other Techniques

The sift-out redundancy will now be discussed in comparison with other redundancy techniques that have been used to provide ultrareliable digital systems.

Triple Modular Redundancy (TMR)

In the basic TMR configuration, the system is organized into three identical channels that feed a voting element. The voting element compares the output signals of the three channels and selects the signal on which the majority of the channels agree.

The TMR organization is one of the oldest forms of redundancy and has been considered the most promising for universal application [9]. However, the process that makes TMR fault-tolerant also makes it difficult to maintain. To analyze the performance of a malfunctioned system, error detection and fault isolation are necessary. The TMR majority voting mechanism masks a bad channel, but at the same time, complicates the detection of the error. To overcome this difficulty, extra hardware has been incorporated into TMR organized computer systems [9], [10].

A sift-out configuration with three channels has the same fault tolerance as a TMR configuration. Since only one channel is allowed to fail, there is no need to "sift" a failed channel out, and the restoring organ can be simplified (Fig. 5). The scheme already has the built-in capability of automatic error detection and fault isolation. The value of the variable F_i provides immediate information about the state of channel i (good if $F_i = 0$; bad if $F_i = 1$). This is an important advantage in commercial computers where redundancy is considered primarily for easing maintenance operations rather than improving reliability [11].

One advantage of TMR is illustrated when dividing a given system into subsystems; the voting elements can be triplicated in order to mask errors in these elements (see [12]). The restoring organ of sift-out redundancy can be made redundant in the same manner. This way the system can tolerate failures in the restoring organ itself.

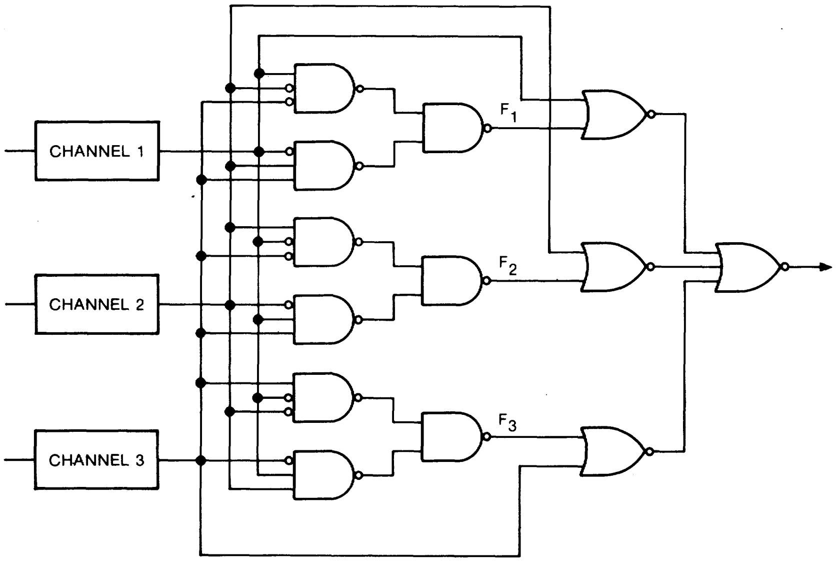


Fig. 5. Sift-out redundancy with three channels.

N-tuple Modular Redundancy (NMR)

In an NMR system, each nonredundant module is replicated an odd number (N) of times. The N identical channels feed a majority voting element. The structure works as long as a majority of the channels are fault-free.

The fault tolerance of an NMR configuration is only $F = (N - 1)/2$; the fault tolerance of a sift-out configuration with the same number of channels is $F = N - 2$. When comparing the NMR voting unit with the sift-out restoring organ, the former is found to be less complex than the latter for small values of N, but the situation inverts as N increases. In addition, the disadvantages already mentioned for the TMR, of which NMR is a generalization, are to be considered.

Hybrid Modular Redundancy (HMR)

Hybrid redundancy has been developed as a means of achieving greater reliability and longer failure-free operations than those achieved by TMR or NMR systems [2]. HMR consists of an NMR core and S standby spare channels. The restoring organ includes, besides the NMR voter, a disagreement detector and a switching network. If the disagreement detector finds that the output of a channel in the NMR core does not match the outputs of the other channels, the switching network replaces it by one of the standby channels.

Hybrid redundant systems combine the advantages of NMR systems (instant internal fault-masking) and standby systems (increased reliability for long time missions). Due to greater fault tolerance, they yield a more efficient hardware utilization than the NMR systems. If the restoring organ is assumed perfect, the most efficient use for a total of $L(= N + S)$ channels would be to construct the hybrid system with a TMR core ($N = 3$) and $S(= L - 3)$ standby spares. However, due to the switch complexity, the actual reliability of this scheme may be less than a design with more channels in the core ($N > 3$) [13]. The implementation of the restoring organ of a hybrid system is not straightforward, and requires a fairly complicated switch. Increasing the number S of spares complicates the switch so much, that beyond a given point the overall reliability starts to degrade [14].

Sift-out redundancy has a fault tolerance as high or higher than hybrid redundancy, as well as a simpler implementation.

Self-Purging Redundancy

Self-purging redundancy [15] is, like sift-out redundancy, a responsive structure with an L-down-to-2 strategy. Self-purging redundancy has L channels feeding a threshold voter. Errors are detected by comparison of the channel output with the voter output. When a channel fails, its output is forced to zero. This is logically equivalent to disconnecting failed modules from the voter.

The self-purging concept originated with Pierce [8] and a relay implementation appeared in [16]. The self-purging restoring organ consists of a threshold voter and L elementary switches, each one made up of an EXCLUSIVE OR gate, a status flip-flop, and an AND gate.

For a three-channel implementation, this restoring organ requires three flip-flops, three EXCLUSIVE OR gates, and seven elementary gates. The sift-out restoring organ requires only thirteen elementary gates (Fig. 5). For $L > 3$, the self-purging has a less complex restoring organ than the sift-out. For example, with 4 channels, self-purging requires 4 flip-flops, 4 EXCLUSIVE OR, and 11 elementary gates; the numbers for sift-out are: 4, 6, and 15.

Sift-out tolerates multiple stuck-at-1 failures (up to $L - 2$). With a dual implementation of the collector (Fig. 6), it will tolerate multiple stuck-at-0 failures. If the relative occurrence frequency of stuck-at-1 and stuck-at-0 failures is known, the most appropriate of the two schemes can be chosen. A mechanism that enables self-purging systems to tolerate multiple failures of either type has been proposed [5], [15], but the price paid in complexity increase is high.

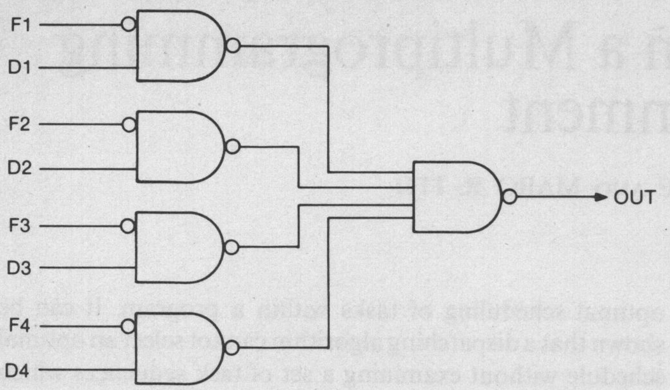


Fig. 6. Collector for a 4-channel sift-out redundancy, tolerating multiple stuck-at-0 failures.

Conclusions

A responsive type of modular redundancy technique has been introduced and analyzed. A straightforward implementation with logical gates was developed.

The technique, sift-out redundancy, was favorably viewed when compared with older redundancies.

The advantages of sift-out redundancy include the following:

1) inherent fault detection;
2) fault-isolation capability, facilitating diagnosis and self-repair;
3) adjustable order of redundancy;
4) efficient use of hardware;
5) straightforward implementation.

These advantages make the sift-out redundancy particularly suitable for the following.

1) Logic circuits whose continuous real-time operation is essential, such as the "hard-core" of ultrareliable computers.

2) Systems that need to be ultrareliable for over a long period of time.

3) Complex multiprocessor systems in which similar subsystems are used to perform critical and subcritical tasks. Since a failure is easily detected, subsystems "in pain" [10] can be readily assigned to noncritical tasks.

4) Redundant systems with easy maintenance requirements.

Acknowledgment

The constructive comments made by the referees caused a substantial improvement of the paper.

References

[1] J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," in *Automata Studies.* Princeton, NJ: Princeton University Press, 1956, pp. 43–98.

[2] F. P. Mathur and A. Avizienis, "Reliability analysis and architecture of a hybrid-redundant digital system: Generalized triple modular redundancy with self-repair," in *1970 Spring Joint Computer Conf., AFIPS Conf. Proc.*, vol. 36, May 1970, pp. 375–383.

[3] L. Fein, "The place of self-repairing facilities in computers with deadlines to meet," in *Proc. Eastern Computer Conf.*, 1957, pp. 111–115.

[4] F. P. Mathur and P. T. de Sousa, "Reliability modeling and analysis of general modular redundant systems," *IEEE Trans. Rel.*, vol. R-24, pp. 296–299, Dec. 1975.

[5] J. Goldberg, K. N. Levitt, and R. A. Short, "Techniques for the realization of ultra-reliable spaceborne computers," Stanford Res. Inst., Menlo Park, CA, Final Rep.—Phase I, Project 5580, Sept. 1966.

[6] M. Ball and F. Hardie, "IBM proposes triple-redundant aerospace computer," *Comput. Design*, pp. 34–36, Nov. 1967.

[7] F. P. Mathur and P. T. de Sousa, "Reliability models of NMR systems," *IEEE Trans. Rel.*, vol. R-24, pp. 108–113, June 1975.

[8] W. H. Pierce, *Failure-Tolerant Computer Design.* New York: Academic, 1965.

[9] M. Ball and F. Hardie, "Self-repair in a TMR computer," *Comput. Design*, vol. 8, pp. 54–57, Feb. 1969.

[10] S. L. Hight and D. P. Petersen, "Dissent in a majority voting system," *IEEE Trans. Comput.*, vol. C-22, pp. 168–171, Feb. 1973.

[11] M. Ball and F. Hardie, "Redundancy for better maintenance of computer systems," *Comput. Design*, vol. 8, pp. 50–52, Jan. 1969.

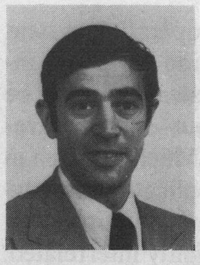
[12] A. D. Friedman and P. R. Menon, *Fault Detection in Digital Circuits.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[13] D. P. Siewiorek and E. J. McCluskey, "Switch complexity in systems with hybrid redundancy," *IEEE Trans. Comput.*, vol. C-22, pp. 276–282, Mar. 1973.

[14] R. C. Ogus, "Fault-tolerance of the iterative cell array switch for hybrid redundancy," *IEEE Trans. Comput.*, vol. C-23, pp. 667–681, July 1974.

[15] J. Losq, "A highly efficient redundancy scheme: Self-purging redundancy," *IEEE Trans. Comput.*, vol. C-25, pp. 569–578, June 1976.

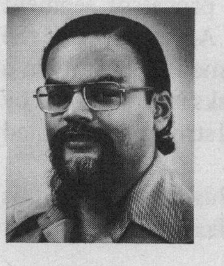
[16] R. Teoste, "Digital circuit redundancy," *IEEE Trans. Rel.*, vol. R-13, pp. 42–61, June 1964.



Paulo T. de Sousa (S'73–M'76) was born in Nova Lisboa, Angola, on January 25, 1947. He received the "Licenciatura" in electrical engineering from the University of Luanda, Luanda, Angola, in 1971, and the M.S. and Ph.D. degrees from the University of Missouri at Columbia in 1972 and 1976, respectively. He lectured at the University of Luanda in 1971 and 1975, and received a Fellowship from the same University while going to graduate school.

In 1976 he joined the Commercial Telecommunications Group of Rockwell International, Dallas, TX, where he has been engaged in performance and reliability analysis of multiprocessor systems and networks and in teletraffic engineering.

Dr. de Sousa is a past Rotary Foundation Fellow and a member of Tau Beta Pi, Eta Kappa Nu, NSPE, and NMA.



Francis P. Mathur received the B.E.E. (honors) degree from the National University of Ireland, University College, Dublin, in 1963, and the M.S.E.E. degree and the Ph.D. degrees with distinction in computer science from the University of California, Los Angeles, in 1967 and 1970, respectively.

He is currently Professor of Mathematics and Coordinator of Computer Science Section at the California State Polytechnic University, Pomona, CA. He has formerly served on the faculties of Wayne State University and the University of Missouri at Columbia. He has published extensively in the area of fault-tolerant computing, is the originator of the hybrid redundancy concept, and is a consultant to industry in these and related areas. He is professionally quite active, being an Associate Editor of the *Computer* magazine and the *Journal of Design Automation and Fault-Tolerant Computing.* He is principal advisor to the Computer Center at the Sri Aurobindo International Center of Education, Pondicherry, India. He served an extensive career in industry, notably he was seven years with NASA's Jet Propulsion Laboratory in Pasadena where he contributed to the JPL-STAR Computer project and the MARS-ROBOTICS project principally. He also served for two years as an Industrial Research Engineer with Bell and Howell Research Labs in Pasadena.

Dr. Mathur received the NASA Apollo Achievement Award in 1969.