MAJORITY LOGIC

Majority Logic is used to implement a majority decision mechanism. In particular, an Majority Logic Block (MLB) can be regarded as a black box that receives possibly different data values at its inputs, and gives, at its output, the data value present on the majority of its inputs. For instance, if a MLB receives the three input data “1”, “1” and “0”, it gives the output data “1”.

MLBs are generally used as component blocks of digital electronic systems devoted to critical operations, like control systems of nuclear plants, flight control systems of aircrafts, speed control systems of trains, onboard control systems of satellites, etc. The correct uninterrupted operation of these systems is mandatory because their malfunction could cause catastrophic consequences, such as loss of human life or huge economical loss. To avoid such losses, the system must be designed to guarantee its correct behavior (that is, the behavior expected in the fault-free case) despite the occurrence of internal faults.

In fact, electronic systems are prone to faults. Faults occur during the system’s manufacturing process and during its operation. For instance, in the presence of a high electrical field, high current density within a circuit metal line might cause the line to break. This might make the faulty circuit provide an output datum different from the correct one, for example, a “0” rather than a “1”. If the faulty signal, for instance, is a signal that activates the alarm device of a train’s speed control system when equal to “1”, it is obvious that a faulty “0” may cause the whole system to become ineffective with possible catastrophic consequences.

Unfortunately, faults of this kind (and faults of a different kind) might occur within a digital electronic system. The likelihood of their occurrence is reduced by using proper electronic components, but ensuring that they never occur is impractical.

Hence, if the correct operation of a system is crucially important, some precautions must be taken to avoid the catastrophic consequences that faults might produce. The techniques used to reach this goal are generally called fault-tolerant techniques.

In particular, to guarantee that a system tolerates its possible internal faults (that is, to ensure that no system malfunction occurs because of such faults), redundancy is used. As an example, to guarantee that a train’s speed control system tolerates faults in the circuit activating the alarm device, redundant copies of such a circuit are used. If these redundant copies are properly isolated from one another, a single fault during the system operation affects only one copy, hence only this copy provides incorrect (or faulty) output data, whereas the other copy (or copies) gives correct output data.

However, it should be obvious that redundancy alone is not sufficient to ensure that a system tolerates its possible internal faults. To reach this goal, we need some decision criterion that allows us to determine which data value, among those present at the output of the redundant copies, can be regarded as correct, and which **one(s)** as incorrect. This decision criterion normally is the majority criterion, implemented by an MLB.

Of course, three is the minimum number of copies of the same circuit (or, more generally, module) that we must have to allow the MLB’s selection of the majority data value.

The use of MLBs and redundant copies of the same module characterize the fault-tolerant N-Modular Redundancy (NMR) systems where, as previously introduced, N must be ≥ 3. The idea to use N-Modular Redundancy and majority logic blocks (also called majority “voting blocks”, or “voters”) to achieve fault-tolerance was first introduced in (2) and has been adopted for several critical applications, such as space exploration missions and nuclear reactor protection (3, 4).

In the particular case where N = 3, these systems are called Triple Modular Redundancy (TMR) systems. Hence a TMR system consists of: (1) three copies of the original, non fault-tolerant module, that concurrently process the same information (where a module is simply a circuit, a gate, or even a microprocessor, depending on system’s related choices), (2) n MLBs (where n is the number of outputs of the replicated module), each comparing bit-by-bit the corresponding outputs of the three replicated modules, and producing at its output the majority data value among those at its inputs (Figure 1). Moreover, suitable techniques are generally adopted at the system level to avoid exhausting all of the system’s redundancy as modules get successively faulty.

Obviously, MLBs play a critical role in the fault-tolerant systems described. In fact, the correct operation of the whole system is compromised if an MLB becomes faulty. For instance, it is obvious that, if the output of an MLB is affected by a stuck-at “1” fault (for instance, because of a short between an output line and the circuit power supply), the faulty MLB always gives an output “1”, regardless of the data values at its inputs (that is, also when the data value on the majority of its inputs is a “0”). Similar problems may occur because of different kinds of faults possibly affecting the input (5), internal and output lines/nodes of a MLB.

In the remainder of this article we consider the problem of designing MLBs for TMR and NMR systems and the case of faulty MLBs.

MAJORITY LOGIC FOR TRIPLE MODULAR REDUNDANCY SYSTEMS

As previously introduced, an MLB can be regarded as a black box that must produce at its output the datum value present on the majority of its inputs. To distinguish such a majority datum value, an MLB must have at least three inputs. When this is the case, the MLB must satisfy the truth table shown in Table 1, where Z denotes the output of the MLB, and $X = (X_1, X_2, X_3)$ is the MLB input vector.

Hence, the MLB can be implemented (at the logic level) by a two-level NAND-NAND (Figure 2) or NOR-NOR (Figure 3) circuit. Equivalently, a two level AND-OR, or OR-AND implementation can be considered (6).

Of course, the electrical level implementations of these MLBs depend on the technology adopted. As a significant example, if the Complementary MOS (CMOS) technology is used, the NAND-NAND majority logic block can be im-

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

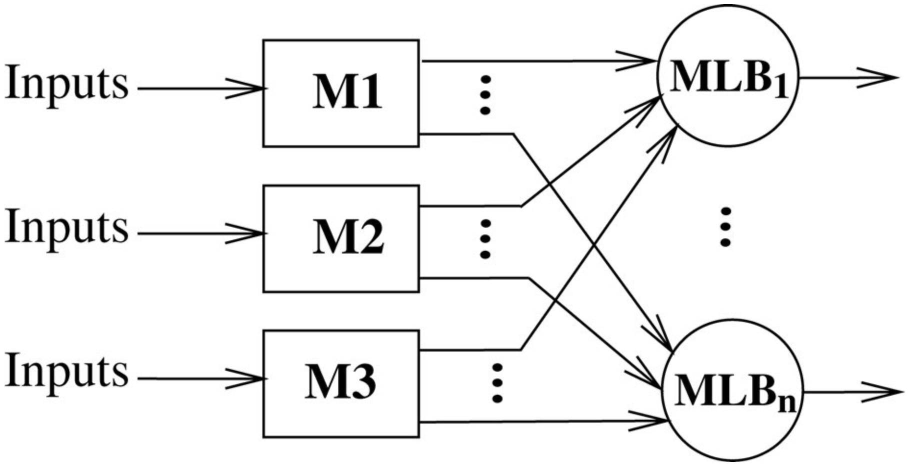

Figure 1. Representation of a Triple Modular Redundancy (TMR) system.

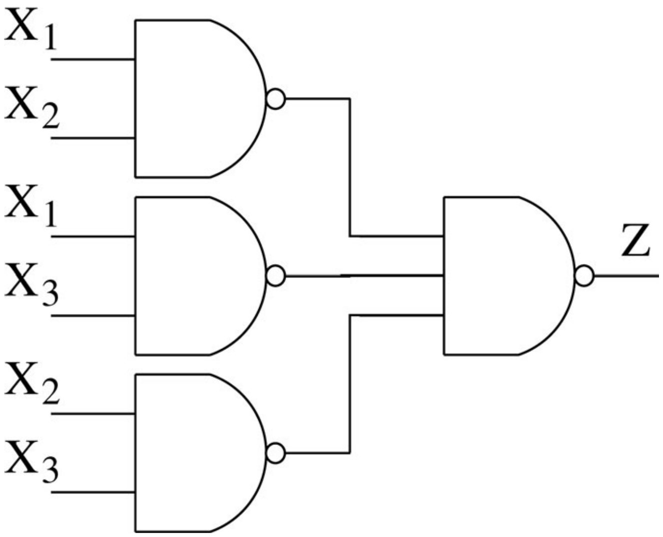

Figure 2. NAND-NAND logical implementation of a majority logic block for a TMR system.

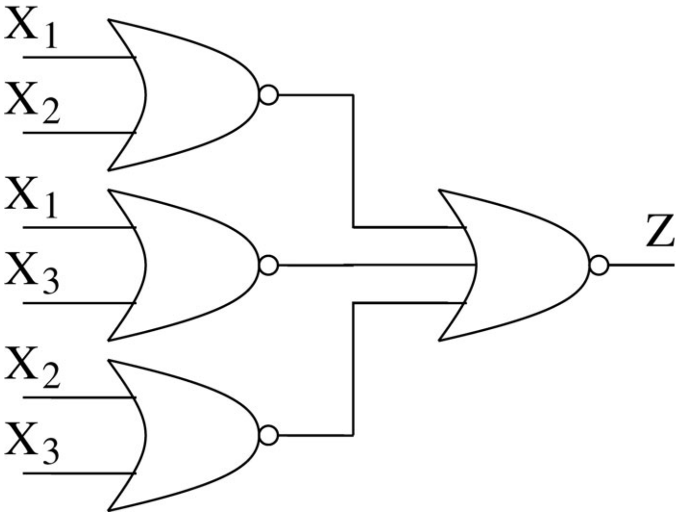

Figure 3. NOR–NOR logical implementation of a majority logic block for a TMR system.

plemented by means of the circuit shown in Figure 4. Alternatively, the circuit shown in Figure 5, for instance, can be used (6). Other possible electrical implementations of MLBs for TMR systems (Figure 6 and Figure 7) were proposed in (7, 8). Different from the conventional MLB realizations considered up to now, these MLBs produce an output error message in case of internal faults that make them give an incorrect majority output datum value. The behavior of these circuits is described later while dealing with the problems due to faults affecting MLBs.

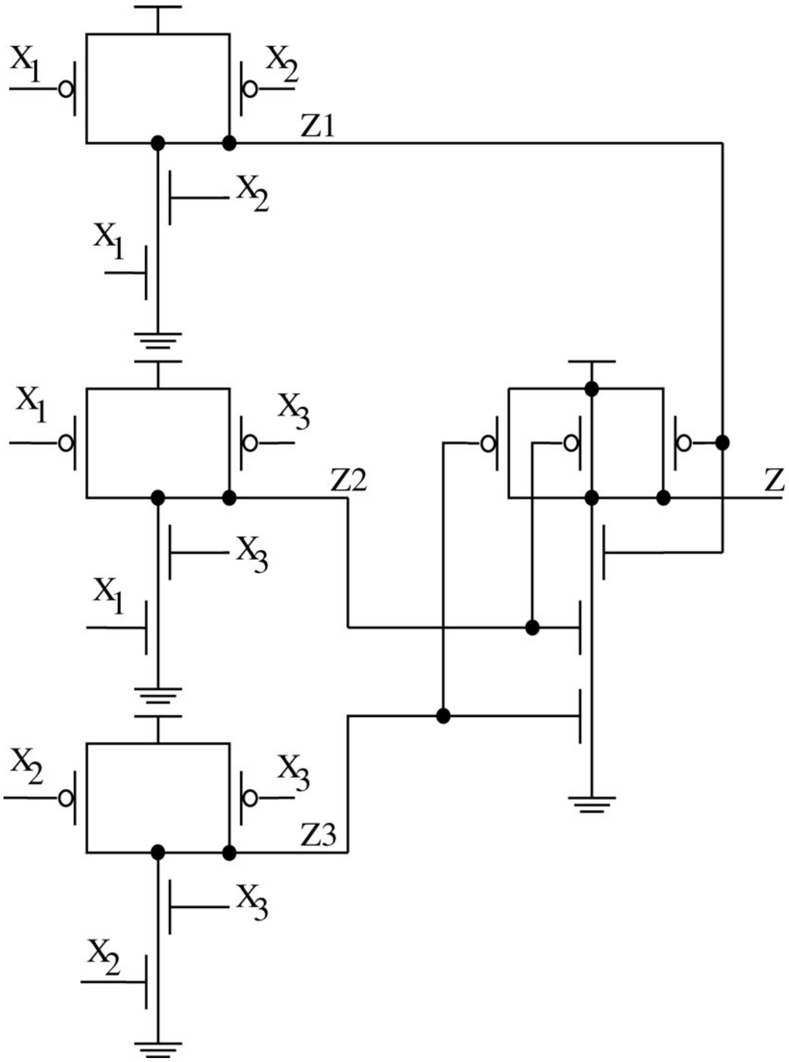

Figure 4. Example 1 of a possible CMOS implementation of the NAND–NAND majority logic block in Figure 2.

Table 1. Truth Table of a Three-Input Majority Logic Block

X_1 X_2 X_3	Z
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	1

As regards the number of errors in X tolerated by a general TMR system, it is evident that, if only one of the MLB input data is incorrect (that is either X_1, X_2, or X_3), the MLB produces the correct output datum value. Instead, if two input data are incorrect, the MLB produces the incorrect output datum value. Hence, a TMR system tolerates only a single error on the corresponding outputs of the replicated modules. If a higher number of errors must be tolerated (because of system level requirements) an NMR system with n MLBs (where n is the number of outputs of the replicated module), each with N inputs (N > 3), must be used.

MAJORITY LOGIC FOR N MODULAR REDUNDANCY SYSTEMS

Tto tolerate E errors in the vector $X = (X_1, X_2, \ldots, X_\mathrm{N})$ given to the input of an MLB of an NMR system, N and E must satisfy the following equation (9):

$$N \geq 2 \cdot E + 1.$$

In fact, when this is the case, the number of erroneous bits E is smaller than the number of correct bits $(N - E)$.

Barbour and Wojcik (6) added an upper bound to the value of N of an NMR system tolerating E errors. In particular, they transformed Eq. (1) into:

$$2 \cdot E + 1 \leq N \leq (E + 1)^2.$$

Moreover, they proposed possible two-level implementations of MLBs for NMR systems, whose values of N and E satisfy Eq. (2). These are the implementations most widely used for MLBs of general NMR systems.

In particular, the first level of the proposed MLBs consists of $\binom{N}{K}$ AND (or OR, or NAND, or NOR) gates, where K is a parameter [called a "grouping parameter" (6)] that

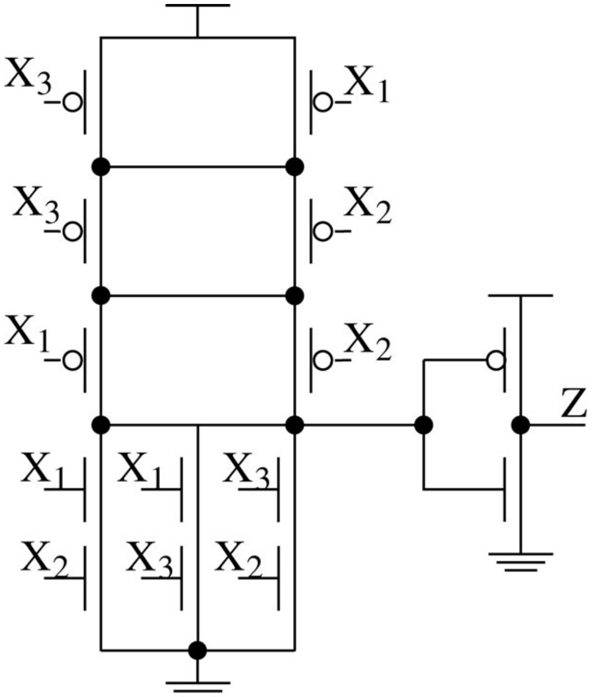

Figure 5. Example 2 of a possible CMOS implementation of the NAND-NAND majority logic block in Figure 2.

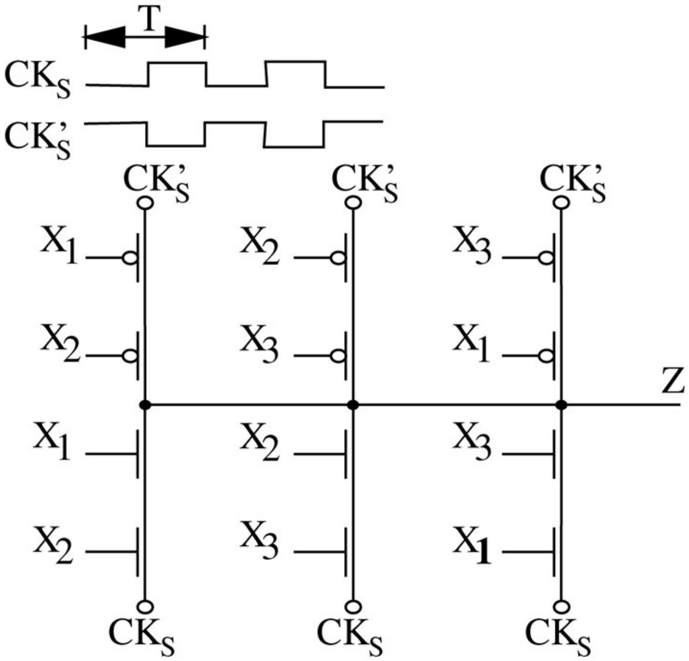

Figure 6. Electrical implementation of the majority logic block giving an output error message in case of internal faults affecting its correct operation proposed in (7).

must satisfy the following condition:

$$E + 1 \leq K \leq N - E.$$

The inputs to these first level gates are the $\binom{N}{K}$ combinations of K elements out of the N elements of the MLB input vector X.

The second level of the proposed MLBs consists of a simple OR (or AND, or NAND, or NOR) gate, depending on whether AND (or OR, or NAND, or NOR, respectively) gates are used at the first level.

As an example, the general NAND-NAND implementation of such an MLB, with generic "grouping parameter" K, is shown in Figure 8. We can easily verify that, if N = 3 and K = 2, this implementation equals that reported in Figure 2.

The derived electrical level implementations of these MLBs are straightforward.

An alternative implementation of MLBs for NMR systems was proposed in (10). These MLBs (Figure 9) are suit-

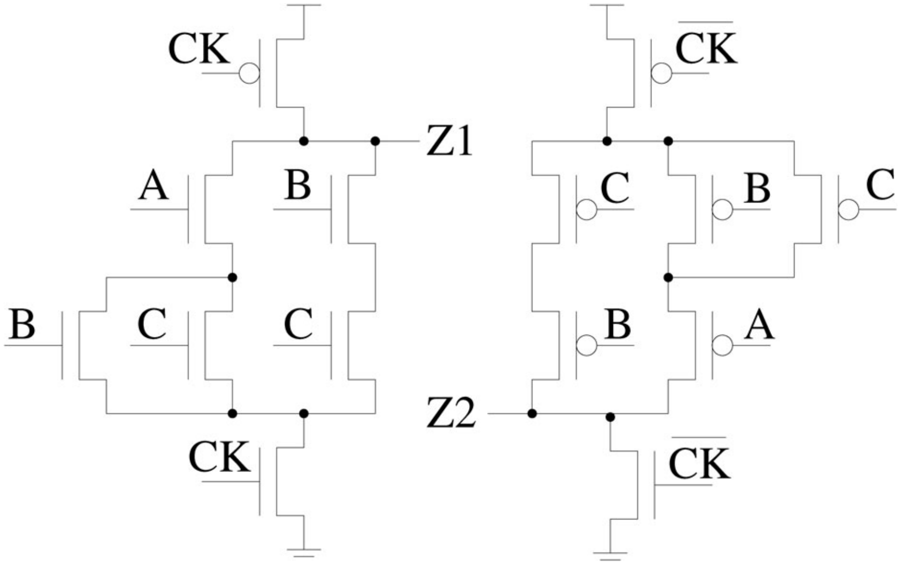

Figure 7. Electrical implementation of the majority logic block giving an output error message in case of internal faults affecting its correct operation proposed in (8).

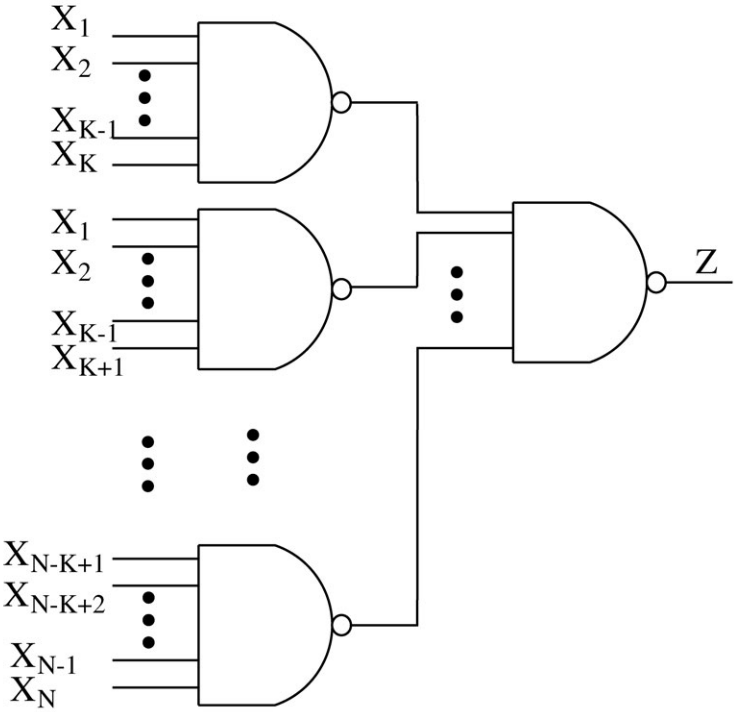

Figure 8. NAND-NAND logical implementation of a majority logic block for an NMR system with “grouping parameter” = K.

able for NMR dynamic systems, that is, in systems where, differing from the conventional cases considered until now, the number of processed replicated modules (hence the number of inputs of the MLBs) can be dynamically changed during the system’s operation.

PROBLEMS IN CASE OF FAULTY MAJORITY LOGIC

As previously mentioned, if the MLB of a TMR (NMR) system becomes faulty, it might produce at its output an incorrect majority datum value, hence making the adoption of the TMR (NMR) fault-tolerant technique useless.

To deal with this problem, MLBs can be themselves replicated (11), as schematically shown in Figure 10 for a TMR system. Note that in this and in the following figures, we use a common symbol to represent n MLBs, and the n outputs of the replicated modules, respectively. It is obvious that, if this scheme is adopted, faults affecting one of the replicated MLBs that make it produce incorrect output data can be tolerated. However, similar to the case of replicated modules only, in this case the problem of distinguishing the datum value provided by the majority of the replicated MLBs also arises. Of course, if other MLBs (receiving the outputs of the replicated MLBs) are used to fulfill this purpose, the problem discussed here is simply translated to these final MLBs.

An alternative strategy for dealing with this problem is to renounce the fault-tolerance requirement of the MLBs

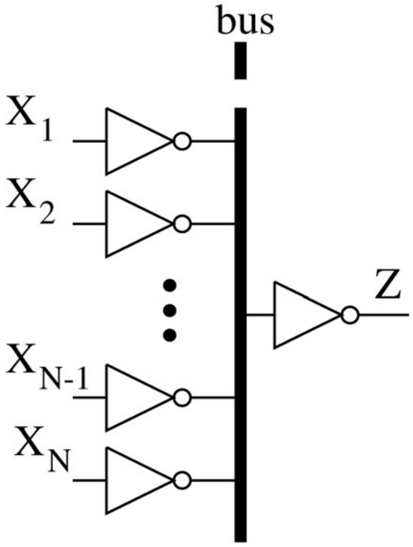

Figure 9. Majority logic block suitable to NMR dynamic systems introduced in (11).

and to require only that the MLBs give an output error message in case of internal faults making them produce incorrect output data. Such an error message can be exploited at the system level to avoid the dangerous consequences possibly resulting from incorrect MLB operation. For instance, if an error message is received by one of the MLBs of a train's speed control system, a system level recovery procedure can be automatically started, eventually making the train stop.

MLBs of this kind can be found in (1–8).

In particular, in (1), the MLBs are duplicated (Figure 11), and their outputs are given to the inputs of a comparator that verifies whether or not such outputs are equal to one another. In the case of a disagreement, the comparator gives an output error message. It is obvious that this strategy guarantees that, in case of faults affecting an MLB that make it produce erroneous majority output data, the comparator gives an output error message.

However, compared with the case where single MLBs are used, this solution implies an inevitable increase of the area overhead and power consumption costs of the fault-tolerant system.

To reduce these costs in TMR systems, the MLB introduced in (7) or (8) can be adopted.

The electrical structure of the MLB in (7) is shown in Figure 6, where CK_S and CK'_S denote a periodic signal whose period is T and an opposite waveform, respectively.

This MLB provides an output error message when internal faults occur that make it give an output incorrect majority datum value. This behavior is guaranteed for all MLB node stuck-at, transistor stuck-on, transistor stuck-open and resistive bridging faults (whose values of the connecting resistance are in the range 0 Ω to 6 kΩ).

In the fault-free case, this MLB gives as its output: (1) a signal equal to CK'_S if the majority of its input signals is equal to "0"; (2) a signal equal to CK_S if the majority of its input signals is equal to "1" (Table 2). Hence, in the fault-free case, this MLB gives at its output a signal assuming both high and low logic values within the same period T. In particular, the logic value produced when $CK_S = 1$ is equal to the majority datum value (*MV* in Table 2) among those given to its input during such a period T.

Table 2. Truth Table of the Majority Logic Block in (6), and Corresponding Majority Datum Value

X_1 X_2 X_3	Z	MV
0 0 0	CK'_S	0
0 0 1	CK'_S	0
0 1 0	CK'_S	0
0 1 1	CK'_S	1
1 0 0	CK'_S	0
1 0 1	CK_S	1
1 1 0	CK_S	1
1 1 1	CK_S	1

Instead, in case of an internal fault of the kind previously reported, either the MLB is not affected by the fault (that is it continues providing the correct majority output data), or it produces an output error message (in particular a signal that does not change logic value within T). If a fault does not affect the MLB and following internal faults occur, either the MLB is not affected by these internal faults or it provides an output error message. This condition holds for all possible sequences of internal faults under the general assumptions that internal faults occur one at a time during the operation of an integrated circuit and that the time interval elapsing between two succeed-

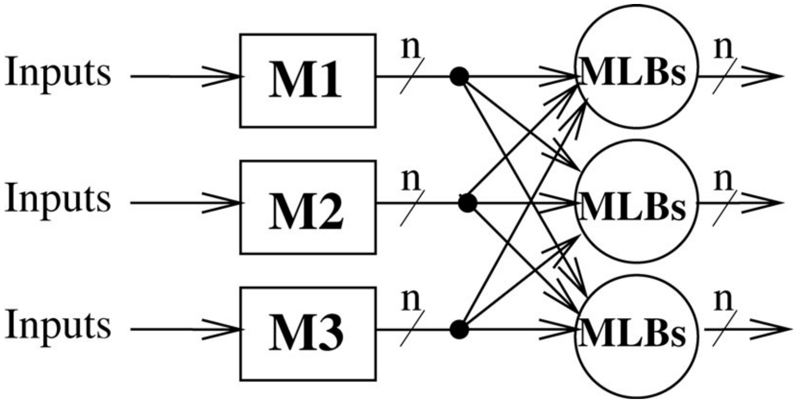

Figure 10. Triplicated MLB scheme proposed in (12). In priciple, triplication of the MLB allows faults that could possibly affect one of the replicated MLBs to be tolerated. In practice, problems arise in correctly discriminating the value given by the majority of the replicated MLBs.

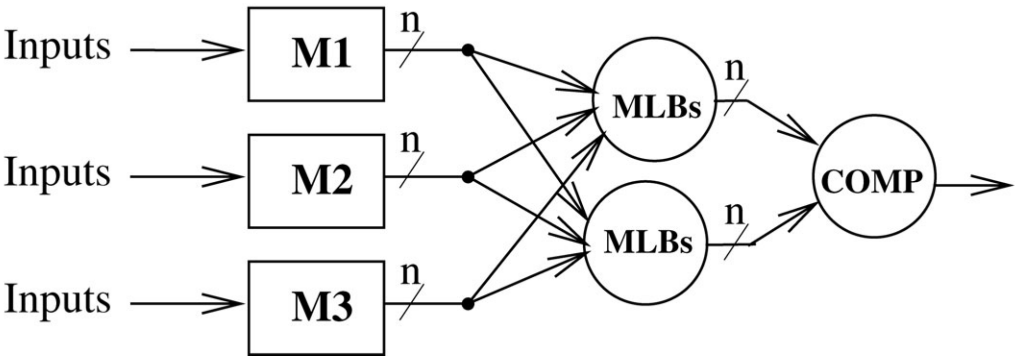

Figure 11. Duplicated MLB scheme presented in (1). Faults affecting one of the duplicated MLBs can not be tolerated, but can be detected by the output comparator.

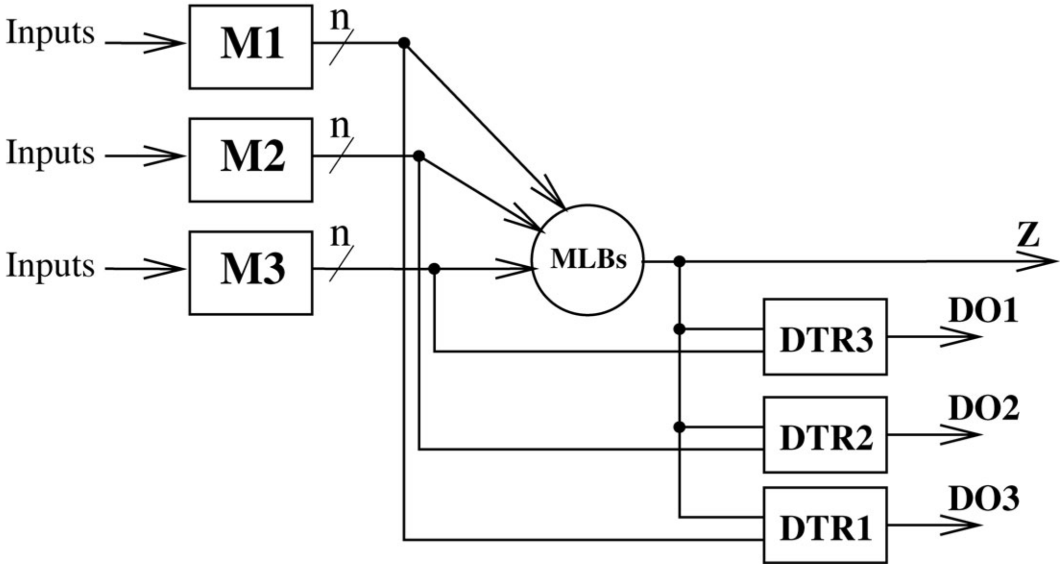

Figure 12. Detecting scheme possibly adopted to avoid the exhaustion of a TMR system redundancy. The detectors ($DTR_\mathrm{i}, i = 1, 2, 3$) reveal the disagreement between the outputs of the replicated modules and the majority output data given by the MLBs.

ing faults is long enough to allow the fault-free modules to produce both high and low output data values.

The presence of an error message at the output of this MLB can be simply revealed. For instance a flip-flop sampling the MLB output on both the CK_S rising and falling edges (12) can be used.

This MLB implementation can also be extended to NMR systems (13). In this case, compared with the electrical structure shown in Figure 6: (1) the number of parallel pull-up/pull-down branches changes from $\binom{3}{2}$ to $\binom{N}{(N+1)/2}$, (2) each branch consists of $(N + 1)/2$ series transistors, rather than of 2. Obviously these conditions may limit the maximal value of N for which this implementation can be conveniently used.

The electrical structure of the MLB in (8) is shown in Figure 11, where CK_S and CK'_S denote a periodic signal with period T, and an opposite waveform, respectively.

Similarly to the MLB in (7), also this MLB provides an output error message in case of the occurrence of internal faults making it give an output incorrect majority data value. This behavior is guaranteed for all MLB's node stuck-at, transistor stuck-on, transistor stuck-open and resistive bridging faults (with values of the connecting resistance in the range 0 Ω to 6 kΩ), but for the bridging fault between its two outputs, which should consequently be avoided, for instance by proper design of the circuit layout.

In the fault-free case, this MLB gives as its output: (1) $(Z1_\mathrm{i}, Z2_\mathrm{i}) = (1, 1)$, if the majority of its input signals is equal to "0"; (2) $(Z1_\mathrm{i}, Z2_\mathrm{i}) = (0, 0)$, if the majority of its input signals is equal to "1".

In case of internal faults of the kind previously reported, this MLB behaves as the MLB in (7), with the difference that the provided error message is an output word with $Z1_\mathrm{i} \neq Z2_\mathrm{i}$, rather than a a signal that does not change logic value within T (as for the MLB in (7)).

Compared to the MLB in (7), the MLB in (8) features the advantage of being faster (offering a reduction of the input/output propagation delay of approximately the 80%), thus being more suitable to high speed, high reliability systems, while requiring comparable power consumption. This is achieved at the cost of a small increase in area overhead (14). Additionally, as proven in (14), the MLB in (8) continues to work properly (i.e., either it continues to provide the correct majority vote, or it produces an output error message) even when affected by high leakage currents, which are expected to become a major concern for dynamic CMOS circuits.

As previously introduced, suitable techniques must be also adopted to avoid exhausting the TMR (NMR) system's redundancy because modules become successively faulty. To fulfill this purpose, detectors that reveal the disagreement between the outputs of the MLBs and of the replicated modules can be used (15), as shown schematically in Figure 11. A possible electrical implementation of detectors of this kind, suitable for use together with the MLBs described, can be found in (13).

Voters operating on a word basis, rather than the traditional bit-by-bit basis, have also been proposed (16), which are able to provide an output error message, should the words produced by the three replicated modules of a TMR system differ from each other.

Finally, several critical applications require that the MLBs of the used TMR (NMR) systems provide "fail-safe" outputs; that is, signals whose value is either correct or safe (where, as an example, the red color is the "safe" value for traffic control lights). Possible implementations of this kind of MLBs can be found in (17).

BIBLIOGRAPHY

1. K. G. Shin and H. Kim, A Time Redundancy Approach to TMR Failures Using Fault-State Likelihoods, *IEEE Trans. Comput.*, vol.**43**,pp. 1151–1162,October 1994.
2. J. V. Neumann, Probabilistic logics and the synthesis of reliable organisms from unreliable components, *Automata Studies, Ann. of Math. Studies*, no. 34,pp. 43–98, 1956.
3. C. E. Stroud and A. E. Barbour, Testability and Test Generation for Majority Voting Fault-Tolerant Circuits, *J. of Electronic Testing: Theory and Applications*, Vol.**4**,pp. 201–214, 1993.
4. D. Audet, N. Gagnon, and Y. Savaria, Quantitative Comparisons of TMR Implementations in a Multiprocessor System, in Proc. of 2nd IEEE Int. On-Line Testing Work.,pp. 196–199, 1996.
5. M. Favalli, and C. Metra, TMR Voting in the Presence of Crosstalk Faults at the Voter Inputs, *IEEE Trans. on Reliability*, pp. 342–348,September 2004.
6. A. E. Barbour and A. S. Wojcik, A General, Constructive Approach to Fault-Tolerant Design Using Redundancy, *IEEE Trans. Comput.*, pp. 15–29,January 1989.
7. C. Metra, M. Favalli, and B. Riccò, Compact and Low Power Self-Checking Voting Scheme, in Proc. of IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems,pp. 137–145, 1997.
8. J.-M. Cazeaux, D. Rossi, C. Metra, New High Speed CMOS Self-Checking Voter, IEEE Proc. of the 10th IEEE International On-Line Testing Symposium,pp. 58–63, 2004.
9. W. H. Pierce, *Failure-Tolerant Computer Design*, New York: Academic, 1965.
10. N.-E. Belabbes, A. J. Guterman, and Y. Savaria, Ratioed Voter Circuit for Testing and Fault-Tolerance in VLSI Processing Arrays, *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol.**43**,pp. 143–152,February 1996.
11. P. K. Lala, *Fault Tolerant and Fault Testable Hardware Design*, Englewood Cliffs, NJ: Prentice-Hall International, 1985.
12. M. Afghahi and J. Yuan, Double edge triggered D flip-flop for high speed CMOS circuits, *IEEE J. of Solid State Circuit*, vol.**SC-26**,pp. 1168–1170,August 1991.
13. C. Metra, M. Favalli, and B. Riccò, On-Line Self-Testing Voting and Detecting Schemes for TMR Systems, *J. of Microelectronic Systems Integration*, vol.**5**, no. 4,pp. 261–273, 1997.
14. J.-M. Cazeaux, D. Rossi, C. Metra, Self-Checking Voter for High Speed TMR Systems, *J. of Electronic Testing: Theory and Applications*, pp. 377–389, 2005.
15. N. Gaitanis, The Design of Totally Self-Checking TMR Fault-Tolerant Systems, *IEEE Trans. Comput.*, vol.**37**,pp. 1450–1454,November 1988.
16. S. Mitra, E. J. McCluskey, Word-Voter: A New Voter Design for Triple Modular Redundant Systems, IEEE Proc. of the 18th IEEE VLSI Test Symposium,pp. 465–470, 2000.
17. M. Nicolaidis, Fail-Safe Interfaces for VLSI: Theoretical Foundations and Implementation, *IEEE Trans. Comput.*, vol.**47**,pp. 62–77,January 1988.

Cecilia Metra
D.E.I.S. University of Bologna,
Viale Risorgimento 2,
Bologna, Italy