Majority Voters Robustness under the Presence of Permanent Faults

Ingrid F. V. Oliveira, Rafael B. Schvittz and Paulo F. Butzen
Grupo de Sistemas Digitais e Embarcados – GSDE
Center for Computational Science – C3
Universidade Federal do Rio Grande – FURG
{ingridoliveira, rafaelschivittz, paulobutzen}@furg.br

ABSTRACT

Fault tolerance is a major factor for circuits in critical applications. Redundancy techniques are commonly adopted to increase reliability of circuits, data integrity and availability. The most frequently used technique is the Triple Modular Redundancy. This technique guarantees a correct output even in the presence of a single defective module. However, it is not guaranteed that the system will work if this single fault occurs in the majority voter, being the critical part of TMR. For this reason, alternative architectures are proposed in the literature to improve the robustness of this block. In this context, it is important to analyze their expected behavior under the presence of faults. This work analyzes majority voter architectures under the presence of permanent faults at transistor level, evaluating how robust is the architecture considering the concepts of Fault Masking Ratio.

Keywords

Majority voter, nanotechnology, permanent faults, reliability.

1. INTRODUCTION

The evolution of electronic devices follows the progress of integrated circuits (ICs) through the technology scaling. The technology scaling results in circuits with higher performance and an increasing number of functionalities. In contrast, the circuits present higher transistor density, higher complexity and higher manufacturing defects probabilities. Because of these effects, yield and reliability are becoming a major concern in IC design, especially for circuits working in harsh environments [1].

The field of fault tolerance continually faces the challenge of maintaining an acceptable level of service of a system, even in presence of faults [2]. To deal with this issue, hardware redundancy remains the most adopted technique, especially the Triple Modular Redundancy (TMR). The TMR technique consists in three identical modules, which accomplish the same job, connected into a voter that selects the correct output. Therefore, the voter is the weak part of the TMR system. When a fault occurs in a majority voter, it may cause an observed error, changing the expected output signal [3].

In the last years, several majority voter architectures were proposed in the literature to improve the robustness of this TMR component [4-6]. Therewith, the main goal of this work is to analyze the behavior of majority voter architectures proposed in [4-6] under the presence of permanent faults Stuck-Open, Stuck-On and Gate Oxide Short.

This paper is organized as follows: Section II shows the permanent faults explored in this work. Section III describes the behavior of a TMR system and a majority voter. Section IV presents the methodology used to analyze the majority voter implementations. Section V presents the results and finally, section VI presents the final remarks.

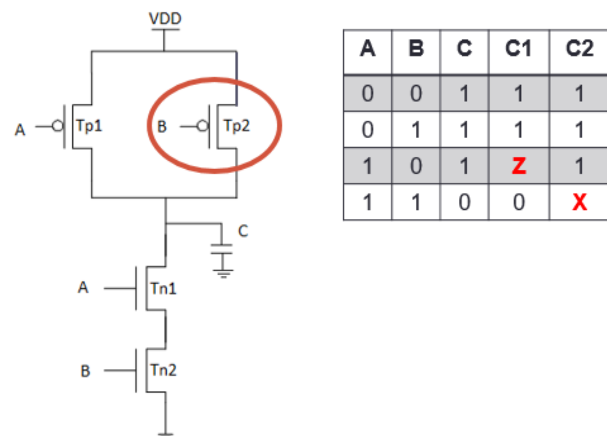
2. Permanent faults

Permanent faults are always present in the circuit due to defects in the manufacturing process and can also be caused by aging effects. These faults can be modeled at logical level (e.g. stuck-at) and at transistor level (e.g. stuck-open, stuck-on, gate oxide short). Fault modeling at logical level is the most commonly used fault model because of its simplicity. However, fault modeling at transistor level is generally more accurate to represent defects than logic level fault models. In this way, this work uses in its analysis transistor level fault models Stuck-Open (SOF), Stuck-On (SOnF) and Gate Oxide Short (GOS).

2.1 Stuck-Open faults (SOF)

When a Stuck-Open fault occurs, the transistor is permanently switched off. In other words, when a transistor presents a SOF, the connection between the drain and source terminal, regardless of the signal applied to the gate, will never exist. To exemplify this fault model, in Fig. 1 it is possible to observe a NAND2 logic gate implementation, where C is its expected output and C1 is its output considering the occurrence of a SOF in transistor Tp2.

Considering an input vector AB=00. Even with a SOF in Tp2, pull-up network still conducts the expected signal to output, because Tp1 and Tp2 transistors are in parallel. That is, even if one of them fails, the other continues to lead the correct signal to output, masking the fault effects. The same occurs with an input vector AB=01, transistor Tp1 send the right signal to output, independently of input B signal, and the pull-down network continues switched off.



A	B	C	C1	C2
0	0	1	1	1
0	1	1	1	1
1	0	1	Z	1
1	1	0	0	X

Figure 1. NAND2 logic gate with SOF/SOnF in Tp2.

With AB=11 as the input vector, the logic gate presents an opposite behavior when compared to the input vector AB=00. In this case, the pull-down network conducts the expected signal to output. The

SOF in Tp2 does not interfere, since Tp2 is located at the pull-up network, that in this situation, is switched off.

Otherwise, when the input vector is AB=10, the expected behavior would be similar to AB=01, but since Tp2 is switched off, because of a SOF, when needed to be switched on, all paths in the pull-up network are blocked, making the output signal float in high impedance Z. When Z occurs, it is necessary to observe the previous state to analyze if the fault will be masked or if an error will be observed. If the previous state is AB=00 or AB=01, the capacitor is loaded, leading the actual state output to '1', masking the permanent fault. However, when the previous state is AB=11, the capacitor is unloaded leading the actual state output to '0', this way an error can be observed.

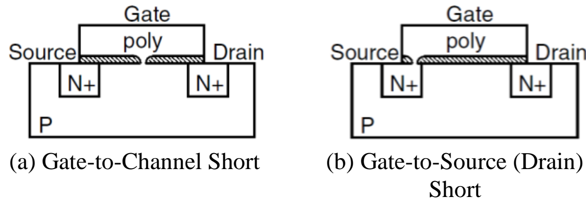
2.2 Stuck-On faults (SOnF)

If a transistor presents a SOnF, it is permanently conducting (switched on). In other words, a SOnF behavior is the opposite of a SOF. Fig. 1 presents a NAND2 logic gate that exemplifies the fault behavior, where C is the expected output and C2 is the output under the presence of a SOnF in transistor Tp2. Considering the input vectors AB=00 and AB=10, in these vectors the pull-up network is responsible to make a path between VDD and output node. In these vectors the transistor Tp2 should be conducting, because of the signal "0" applied on its gate terminal, thus the fault does not interfere in the logic gate result, being masked.

With AB=01 as the input vector, the transistor Tp2 should be switched off, however it is conducting because of the stuck-on fault. Since transistors Tp1 and Tp2 are in parallel, and Tp1 is already making a path in the pull-up network, thus the SOnF is also masked. Although when the input vector is AB=11, a path in the pull-down network is formed, but because of the SOnF in Tp2 a path is also formed in the pull-up network. Since both networks are conducting, the output is in a low impedance (X) state.

2.3 Gate Oxide Short (GOS)

A Gate Oxide Short is an electrical connection through the oxide between the gate and the channel or the source (or drain) terminal, as observed in Fig. 2. In both types of GOS, an unwanted path of current emerges through the oxide of the gate. When a GOS defect occurs as in Fig. 2(a), it causes a short circuit between gate terminal and the transistor channel. This type of GOS is usually caused by gate oxide imperfections or Si surface defects. If a GOS defect is observed as shown in Fig. 2(b), a short circuit between gate and source (or drain) terminals occurs. Shorts like this are often caused by electrical discharges (ESD) or electrical overstress (EOS) [7].



(a) Gate-to-Channel Short (b) Gate-to-Source (Drain) Short

Figure 2. Gate oxide short defect types [8].

3. TMR system and the majority voter

Nanoelectronic systems have become more sensitive to faults and defects due to transistor shrinking. Therewith, many applications need to ensure a high level of reliability. Hardware redundancy remains the most adopted technique to deal with this issue. A widely used fault tolerant technique is the Triple Modular Redundancy (TMR) [9]. As shown in Fig. 3, a TMR system is composed of three identical modules performing the same function and a majority voter.

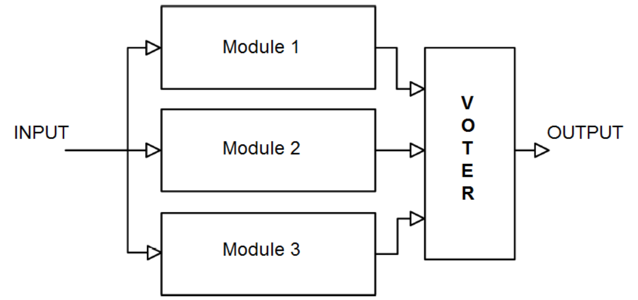


Figure 3. TMR system architecture.

The majority voter compares the output of each module bit-by-bit to vote the correct output. The majority voter function has three inputs ("A", "B" and "C") and one output V, and decides the output by majority as showed in the corresponding canonic Boolean expression given in Eq. 1.

$$V = AB + AC + BC \tag{1}$$

The idea behind the TMR is that a defective module propagating an error can be masked for the two other fault-free modules and can guarantee a full masking to a single fault. A voter is used in these implementations to set through majority vote the possibly correct output. In an ideal TMR system, the reliability of the voter circuit should be much higher than the reliability of the other TMR components. However, in practice the voter is the critical part of the circuit because if anything happens to the majority voter, the TMR structure may get faulty [10]. This shows the importance of it being studied.

In this work, six majority voter architectures are explored [4-6] and the circuits are presented in Fig. 4. The most conventional implementation of Eq. 1 is the CLASSICAL voter, presented in Fig. 4(d). This voter is mainly used in studies to compare with other majority voter architectures. However, its structure is considered less robustness to faults. Another majority voter architecture used in this analysis is the CMOS voter, as shown in Fig. 4(f), which is a similar implementation of the CLASSICAL voter but using a complex logic gate. The main advantage of this implementation is the low number of transistors involved and it is considered more robust than the classical implementation, however in section V it is possible to observe a different behavior for one of the faults.

In Fig. 4(c) can be observed another topology used in this study and it is proposed by [4], the KSHIRSAGAR voter. It is a proposed fault-tolerant voter based in a priority encoder that selects the output to a multiplexer to implement the majority function. This circuit was design to tolerate stuck-at and transient faults. The BAN voter, shown in Fig. 4(a) is a simplification of the KSHIRSAGAR circuit. According to [5], a voter with less transistors has lower probability of having a fault. Therefore, the main advantage of the circuit is the reduced power consumption area, since it was implemented with 18 transistors against 36 transistors of KSHIRSAGAR voter.

Another proposed fault-tolerant voter is shown in Fig. 4(b). The BALA voter is proposed by [6] to be more robust than the other architectures listed above (CLASSICAL, BAN and KSHIRSAGAR). This architecture is composed by a 2-input OR gate whose result is connected to the complex gate that implements the Boolean function of Eq. 1, similarly to the complex logic gate used in the CMOS voter. And in Fig. 4(e) it is possible to observe the circuit of the BALA CMOS voter. This implementation is the BALA voter as a complex logic gate.

4. Methodology

The architectures showed in Fig. 4 were analyzed at a logical level when proposed in the literature. This analysis is often performed at system level due the scalability. However, majority voters are specific and relatively small circuits, making it possible to accomplish a more detailed analysis at transistor level. Transistor level analysis would allow a more accurate results, turning it interesting to use these results to confirm (or not) the results showed in the literature.

Therefore, the objective of this case study is to compare the behavior of six majority voter architectures under the presence of transistor level permanent faults and to analyze the robustness of this block. A fault masking ratio (FMR) is used as a parameter to evaluate the robustness of the circuits.

To accomplish this paper goal a switch-level analysis is realized manually, considering a single fault event in the voter circuit. This way, since the analysis only considers a single fault in the majority voter, the only input vector combinations used in this study are ABC=000 and ABC=111, when all modules are functioning correctly fault-free.

.

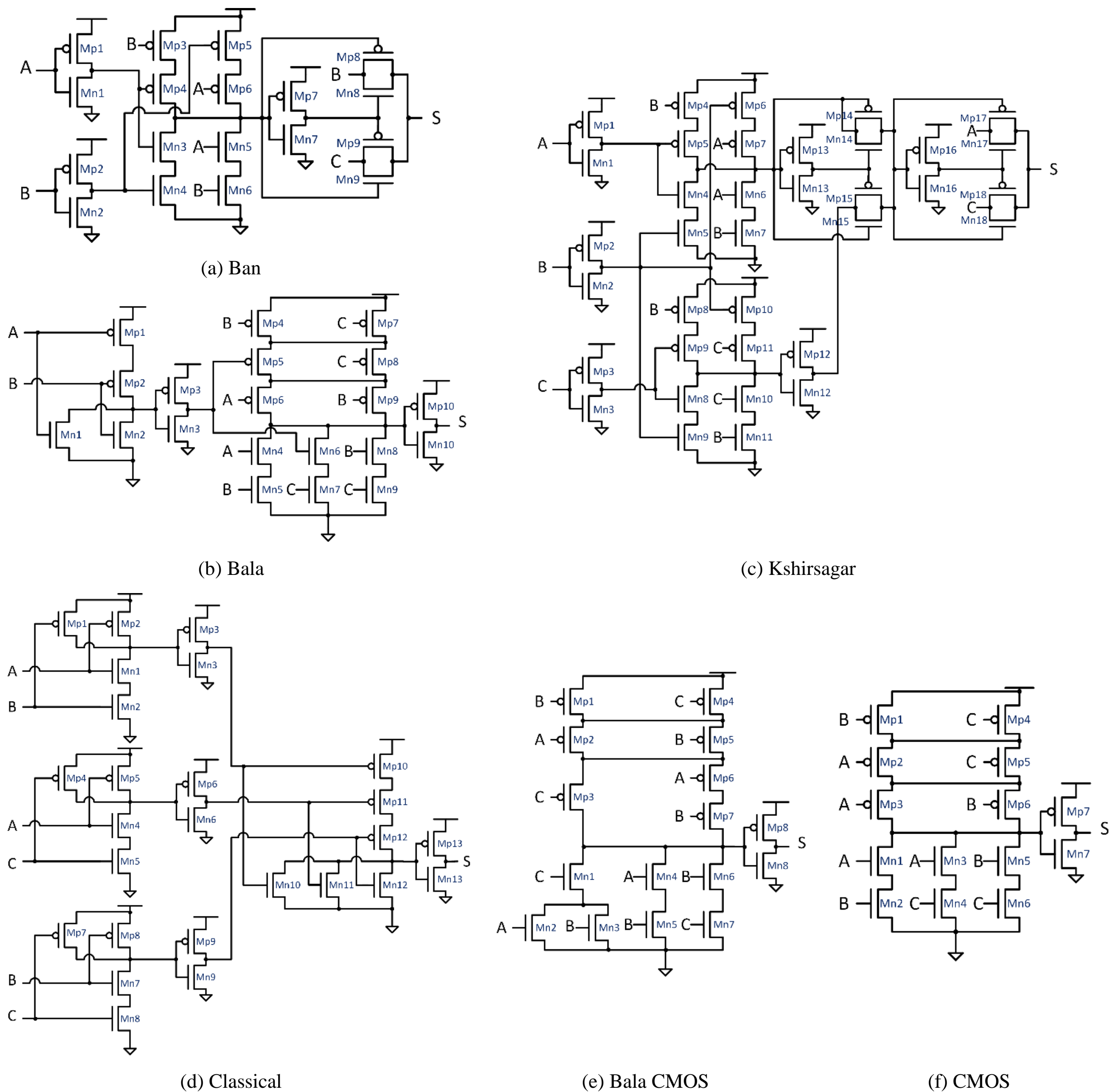


Figure 4. Majority Voters.

Every transistor of the voter circuit is analyzed under the presence of SOnF, SOF and GOS faults, for both input vectors, and how it may or not affect the output signal. Data are obtained at each analysis, as the number of sensitive transistor that can cause an unexpected error in the output. All acquired information is important to analyze the robustness of the architectures under study. To measure the reliability of the architectures, a fault masking ratio is calculated according to Eq. 2. The FMR is calculated as the ratio of the sum of transistors affected by permanent faults which are masked at the output (TM), divided by the total number of transistors (N) considering both evaluated input vectors: “000” and “111”. The closer to 1 the value of the FMR is, more reliable the block is.

$$FMR = \frac{\sum TM}{N} \quad (2)$$

The permanent faults analyzed are listed and described in section II. The Stuck-Open fault is modelled a bit different from usual. Normally, when analyzing this fault, it is necessary to observe the previous state and how it may affect the high impedance state output, in other words, the output assumes the value loaded in the capacitor from the previous state. However, since in this work only two input vectors are analyzed, when all the modules are fault-free, there is no need to observe the previous state. To exemplify, if the voter input is ABC=111 the expected output should be “1”, but because of a SOF the output is in a high impedance state. Looking to the previous state, the only possible state is ABC=000 with output “0”. This way, the high impedance state assumes the “0” value and an error is observed, and since this is the only previous state possible, it will never be masked when a “Z” happens. Thus, this analysis already assumes the worst and only case if “Z” happens, assuming an observation error.

The GOS fault used is a gate-to-channel GOS. This GOS type is chosen in this analysis because it is widely used in the literature when modelling a GOS fault. It occurs due the fact that the probability of a gate-to-channel fault occurring is higher than a gate-to-source (or drain) occurring, since the channel area is bigger than the overlay area.

5. Results

Table 1 contains the FMR values for each majority voter. Each column refers to the FMR results obtained when the circuit was under the influence of the permanent faults indicated in the header. As closer the FMR value is to “1”, more robust the architecture is under the presence of a certain fault.

Analyzing the data obtained from the permanent faults SOnF and SOF in Table 1, it is possible to observe identical values for both faults. This occurs due to the complementary behavior of the faults. Two architectures obtained the maximum value of FMR (1), KSHIRSAGAR and BAN voters. These voter circuits reached excellent results due to their similar architecture that uses a multiplexer and pass transistors.

TABLE 1 – FMR values of the majority voter architectures

Majority Voters	FMR (%)		
	SOnF	SOF	GOS
Classical	0.846	0.846	0.673
CMOS	0.929	0.929	0.500
Bala	0.950	0.950	0.650
Bala CMOS	0.938	0.938	0.500
Ban	1	1	0.944
Kshirsagar	1	1	0.972

Observing the third column, where the FMR results under the presence of the GOS fault are listed, there is a significant fall in the numbers compared to the other faults analyzed in this paper. The CMOS and BALA CMOS voters presented the worst FMR values, which is an interesting fact since both architectures are complex logic gate implementations of CLÁSSICAL and BALA architectures, respectively, that reached more reasonable FMR numbers. At last, KSHIRSAGAR and BAN voter reached the best fault masking ratio results under the presence of a gate-to-channel GOS fault. Although there is a small difference in their FMR numbers, both architectures had just one case where the fault caused a low impedance state in the output.

6. Final remarks and future works

This work compares the behavior of different majority voter architectures under the presence of transistor level permanent faults. The block robustness is analyzed using the FMR value, which is calculated for each architecture under the effects of each fault. KSHIRSAGAR and BAN voter presented the highest FMR values.

The next step is expand the analysis to transient faults to provide a broader study on the effects of faults in majority voters and how it affects its robustness. In addition, it will be investigated the effectiveness in robustness improvements when transistor redundancy techniques is applied in the voters.

7. REFERENCES

[1] D. T. Franco, J.-F. Naviner, and L. Naviner, “Yield and reliability issues in nanoelectronic technologies,” Ann. Telecommun. - Ann. Des Télécommunications, vol. 61, no. 11–12, pp. 1422–1457, 2006.

[2] I. Koren and C. M. Krishna, *Fault Tolerant Systems*. Morgan Kaufmann, 2010.

[3] J. Vial, et al. "Using TMR architectures for yield improvement." 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems. IEEE, 2008.

[4] R. V Kshirsagar and R. M. Patrikar, “Design of a novel fault-tolerant voter circuit for TMR implementation to improve reliability in digital circuits,” Microelectron. Reliab., vol. 49, 2009.

[5] T. Ban and L. A. De Barros Naviner, “A simple fault-tolerant digital voter circuit in TMR nanoarchitectures,” NEWCAS, 2010.

[6] P. Balasubramanian, K. Prasad, and N. E. Mastorakis. "A fault tolerance improved majority voter for TMR system architectures." WSEAS Transactions on Circuits and Systems, v. 15, n. 14, p. 108-122, 2016.

[7] J. M. Soden and C. F. Hawkins. "Test considerations for gate oxide shorts in CMOS ICs." IEEE Design & Test of Computers 3.4, p. 56-64, 1986.

[8] M. Renovell, et al. "Modeling gate oxide short defects in CMOS minimum transistors." Test Workshop, 2002. Proceedings. The Seventh IEEE European. IEEE, 2002.

[9] J. Von Neumann, “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.

[10] M. Sadeghi, et al. "The study of hardware redundancy techniques to provide a fault tolerant system." Cumhuriyet Science Journal, v. 36, n. 4, p. 236-245, 2015