A Hazard-Free Majority Voter for TMR-Based Fault Tolerance in Asynchronous Circuits

Sobeeh Almukhaizim
Computer Engineering Department
Kuwait University

Ozgur Sinanoglu
Mathematics and Computer Science Department
Kuwait University

Abstract

Triple-Modular Redundancy (TMR) is the simplest and most effective fault tolerant design method for ICs, where three copies of a given circuit are employed and a majority voter produces the voted output. Asynchronous circuits, however, exhibit various characteristics that limit the applicability of TMR on such designs. Specifically, the difficulty stems from the fact that asynchronous circuits communicate with their environment through hazard-free output transitions. This hazard-free property needs to be preserved when hardware providing fault tolerance is added. Therefore, performing TMR-based fault tolerance in asynchronous circuits requires the development of a hazard-free majority voter. In this work, we first demonstrate how and why a typical majority voter design would fail to preserve the hazard-free property of its response when a transient error occurs at one of its inputs. We then propose a novel hazard-free majority voter design for the TMR-based fault tolerant method. Finally, using PSpice simulations, we verify the effectiveness of the proposed voter design in preserving the hazard-free property of the response of an asynchronous circuit, and we assess its area overhead over that of a typical majority voter.

1 Introduction

The projected increase in the Soft Error failure Rate (SER) of near-future CMOS technologies has sparked numerous efforts to develop soft error protection mechanisms for digital ICs [1, 2, 3, 4]. Since the majority of commercial ICs available in the marketplace follows the clocked design paradigm, most of these efforts target synchronous circuits. Yet clockless design has recently received increased attention and several advanced asynchronous design styles have been carving an increasing niche in the market [5]. Moreover, due to their low power consumption and electromagnetic noise emission, asynchronous circuits are gaining a particularly strong foothold in mission-critical applications, such as avionics [6], wherein reliability is key. Unfortunately, error protection methods developed for synchronous circuits are not directly portable to the asynchronous domain. Therefore, error protection techniques tailored to the particularities of asynchronous circuits are required.

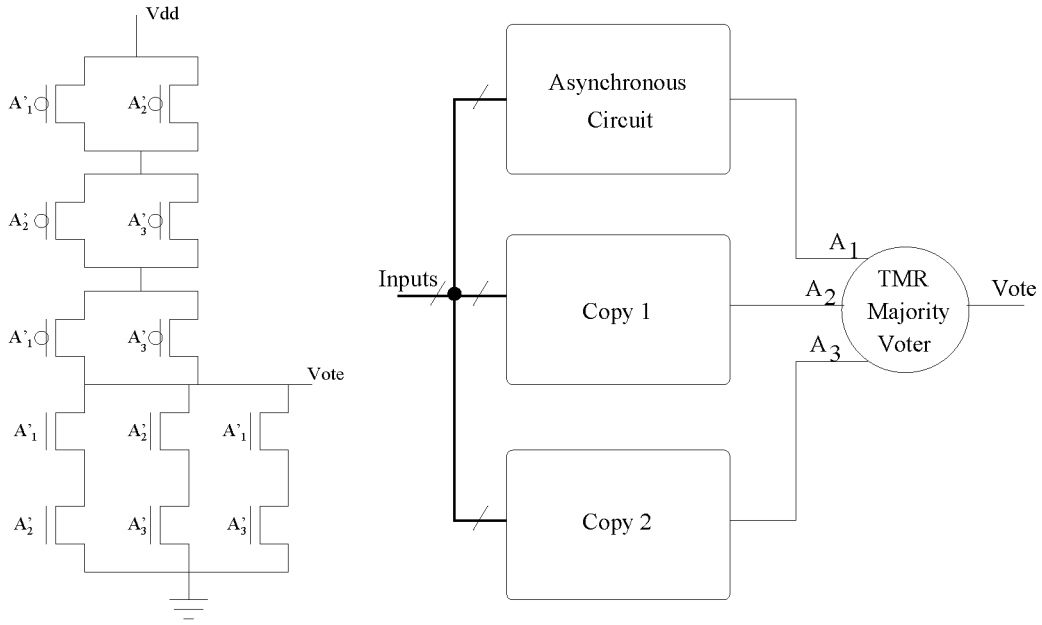
In this paper, we address the problem of performing fault tolerance in asynchronous circuits using the Triple-Modular Redundancy (TMR) design methodology [7]. TMR employs three copies of a given circuit and a majority voter to decide the final output based on their responses. Thus, any error(s) affecting a single circuit copy can be tolerated. While the applicability of TMR on synchronous circuits is straightforward, its asynchronous counterpart is intrinsically different, limiting the applicability of TMR for these circuits. The problem stems from the fact that asynchronous circuits communicate with their environment through hazard-free output transitions. This hazard-free property needs to be preserved when hardware providing fault tolerance is added. Otherwise, a hazard may be misinterpreted by the environment as a logic value change, resulting in erroneous interaction with the circuit and, by extension, erroneous system-level results.

When a TMR majority voter is employed in asynchronous circuits, the output of the majority voter becomes susceptible to transient errors affecting one of its inputs, potentially producing a logic hazard. Specifically, since three copies of an asynchronous circuit may produce their responses with different delays [1], the output of the voter will change when two of its inputs change their value. Before the third input changes, a transient error affecting either one of the circuit copies producing these inputs would change the output value of the voter and, hence, would lead to a hazard. Therefore, a hazard-free implementation of the majority voter is required.

Our aim is to enable the application of TMR-based fault tolerance on asynchronous circuits by developing a majority voter design that preserves the hazard-free property of its response in the presence of transient errors at its inputs. We start in Section 2 with a review of related work in fault tolerance for asynchronous circuits. Then, in Section 3, we provide a research motivation of our work to illustrate the problem of using a typical majority voter in asynchronous circuits. Finally, in Section 4, we describe the proposed hazard-free voter design, which utilizes additional hardware in order to force the voter to retain its value after two of the circuit copies change their logic value. Therefore, transient errors affecting one of the circuit copies will not change the output value of the voter and, hence, will be tolerated.

[1] Process variation, input skew, and the sheer fact that the circuits are separate entities are few of the reasons why three identical circuits may operate with different delays [8].

978-1-4244-1825-1/07/$25.00 ©2007 IEEE



a. Typical TMR Majority Voter Design

b. TMR Implementation

Figure 1. Typical TMR Majority Voter Design and its Use in a TMR-Based Fault Tolerant Asynchronous Circuit

2 Related Work

Several fault tolerance methods for specific styles of asynchronous circuits have been previously proposed in the literature. However, none of these approaches yields an implementation that has the same tolerance capabilities or performance of a TMR-based design. Below, we summarize these studies and describe the reasons why TMR is more generic and powerful than these methods.

Jang *et al.* [9] investigated the development of fault tolerance methods for the class of Quasi-Delay Insensitive (QDI) circuits. In order to make a QDI circuit soft error tolerant, every gate is duplicated and each pair of nominally identical outputs is fed to two C-elements [10]. A C-element, which has also been used to tolerate soft errors in synchronous circuits [11, 12], is a state-holding component that waits for all of its inputs to agree on a logic value before it changes its state to this value. Hence, a transient error at a gate is blocked by the correct value of the duplicate gate and does not propagate to the output of the C-element.

Monnet *et al.* [13] presented several soft error hardening methods for QDI circuits. A QDI circuit is divided in their analysis into a computational logic part and a memory part. In order to harden the circuit, the authors employ three methods, based on duplication of the computational part and expansion of the C-element in the memory part, synchronization of linked channels when available, and synchronization using a redundant control circuit, when no linked channels are available. Again, similar to [9], the use of C-elements is crucial to the success of these hardening methods.

While C-elements successfully tolerate transient errors in QDI circuits, the produced implementation suffers from several shortcomings when compared to a TMR-based design. Specifically, C-elements do not tolerate permanent faults in the circuit or its duplicate; hence, permanent faults would lead the fault tolerant design into a deadlock state. A TMR-based design, however, would tolerate permanent faults. Moreover, since a C-element waits for all of its inputs to agree before it changes its output, the performance of the circuit is dependent on the duration of the transient error. A TMR-based design, however, produces the correct output when two of the circuit copies generate their responses. Therefore, the performance of a TMR-based fault tolerant design is independent of the duration of transient errors.

Despite the above advantages of TMR-based fault tolerance over other fault tolerance approaches, its applicability on asynchronous circuits is compromised by the hazard-free property of these circuit. In the next section, we demonstrate how a TMR-based fault tolerant design of an asynchronous circuit would not preserve the hazard-free property of its response in the presence of transient errors.

3 Motivation

Consider the transistor-level implementation of a TMR majority voter shown in Figure 1.a, and assume that A_1, A_2 and A_3 are the three input signals to the majority voter. We want to perform TMR-based fault tolerance for an asynchronous circuit by triplicating it and connecting its outputs

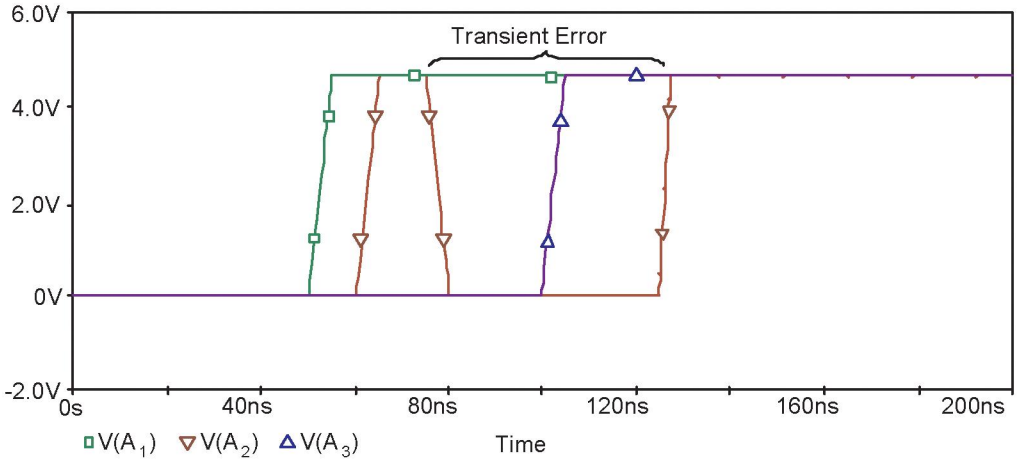


Figure 2. Input Stimulus to the Majority Voter with a Transient Error Affecting Input A_2

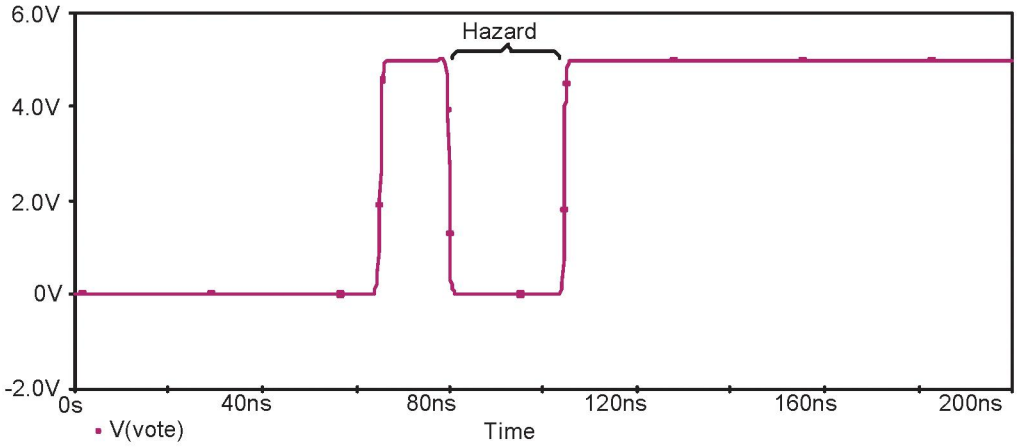


Figure 3. Output of the Majority Voter

to the inputs of the majority voter, as illustrated in Figure 1.b. Furthermore, our objective is to simulate the output behavior of the majority voter in the presence of a transient error at one of its inputs. Thus, accurate transient response (rise/fall time, propagation delay, etc.) of the circuit is required by incorporating proper fabrication model parameters. In these experiments, we perform the simulations using OrCAD PSpice and MOSIS fabrication parameters.

Assume that the outputs of the asynchronous circuits in Figure 1.b change from logic 0 to logic 1 during normal operation. Furthermore, assume that the change in these signals happens in an increasing order (i.e. A_1 changes first, followed by A_2, and finally A_3) and that a transient error occurs at A_2 that flips its value to logic 0, as illustrated in Figure 2. The output response of the majority voter for the above input stimulus is illustrated in Figure 3. As can be seen in the figure, the output value of the majority voter changes to logic 1 following the change in A_1 and A_2. At this point of time and until A_3 changes to logic 1, any transient error at A_1 or A_2 will change the output value of the majority voter back to logic 0. Specifically, this is the case for A_2, which changes back to logic 0 due to a transient error. At this point of time, the output of the majority voter temporarily changes to logic 0 until A_3 changes to logic 1, after which the output value of the majority voter flips back to the correct value of logic 1. Thus, a hazard is observed at the output of the majority voter due to a transient error at A_2.

As demonstrated through this simulation example, transient errors affecting one of the inputs to the majority voter may produce a hazard at its output. Since no clock is used in asynchronous designs, synchronization between the circuit and its environment is based on the fact that any change in the output of the circuit signifies completion of an evaluation cycle. Hence, hazardous behavior may be misinterpreted by the environment as a logic value change, resulting in erroneous interaction with the circuit and, by extension, erroneous system-level results. Therefore, a hazard-free majority voter that is capable of producing the correct voted output, as well as preserving its logic value, in the presence of transient errors is required.

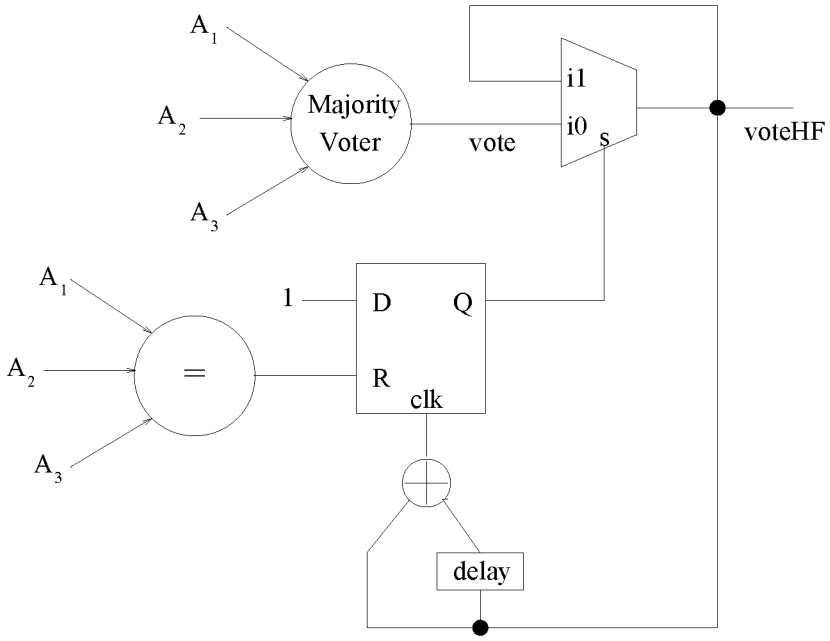


Figure 4. High-Level Design of the Proposed Hazard-Free Majority Voter

4 Hazard-Free TMR Majority Voter Design

The objective of the proposed TMR majority voter is to achieve hazard-free resiliency to potential transient errors in one of the three circuit copies. In order to achieve such hazard-free error resiliency, the majority voter must account for the transient behavior of the circuit copies. For this purpose, we need to distinguish between the following states of the circuit copies: i) the state where the outputs of all the three circuit copies agree, and ii) the state where the outputs of two of the three circuit copies agree. While a transient error affecting one of the circuit copies would not alter the output of the voter in the former state, it may lead to a hazard in the latter state. We elaborate next on the output behavior of the voter when an error occurs in one of these two states.

When the outputs of all the three circuit copies agree, the output of the voter is correct since no error occurs in any of the circuit copies. Moreover, if the outputs of only two copies agree and under the single error assumption of TMR, one of the two copies must be producing the correct output value and the output value of the voter is concluded to also be correct. Yet, potential transient errors affecting the output value of one of these two copies would be sensitized to the output of the voter. Specifically, a transient error at one of these two circuits would make the output of the voter temporarily incorrect. Hence, the TMR majority voter fails to mask a single transient error. The erroneous output value of the voter will remain until either the the third circuit copy produces the correct output value (e.g. see the input stimulus in Figure 2) or the transient error disappears; either way, a hazard is observed in the response of the voter.

Following the above analysis, errors may be sensitized to the output of the voter when the state of only two of the output values of the circuit copies agree. We refer to such state as one where the *error sensitization condition* is met. In order to preserve the hazard-free behavior of the majority voter, we insert additional hardware in the proposed voter design in order to detect the error sensitization condition. Thus, following the change at the output of the majority voter and the identification that the error sensitization condition has occurred, the hazard-free voter should be forced to retain its output value, forbidding any error-induced transitions at its output. This "output freezing state" should continue until all the voter inputs agree. At this point of time, no error in a single circuit copy can reflect to the voter output anyways and, therefore, the output of the majority voter would be correct.

The high-level design of the proposed hazard-free TMR majority voter is illustrated in Figure 4. The output of a typical majority voter drives a multiplexer, whose other data input is the output of the proposed hazard-free voter. The multiplexer selects the hazard-free output of the voter at $i1$ when the error sensitization condition is met (i.e. the "output freezing state" should be entered), and the output of the typical majority voter at $i0$, otherwise. The select line of the multiplexer is driven by a flip-flop output. The clock input of this flip-flop is connected to an XOR gate along with a delay unit, i.e., a transition detector circuit. A transition on the output of the hazard-free voter indicates that the error sensitization condition is met, which triggers a rising transition on the clock input of the flip-flop and stores the logic value 1. This event indicates the entry to the aforementioned "output freezing state". At this point, the multiplexer sensitizes

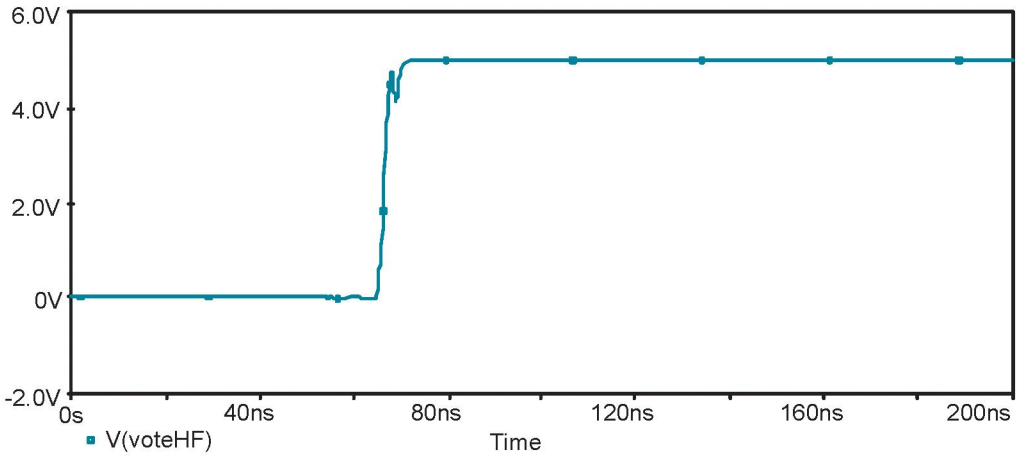


Figure 5. Output of the Proposed Hazard-Free Majority Voter

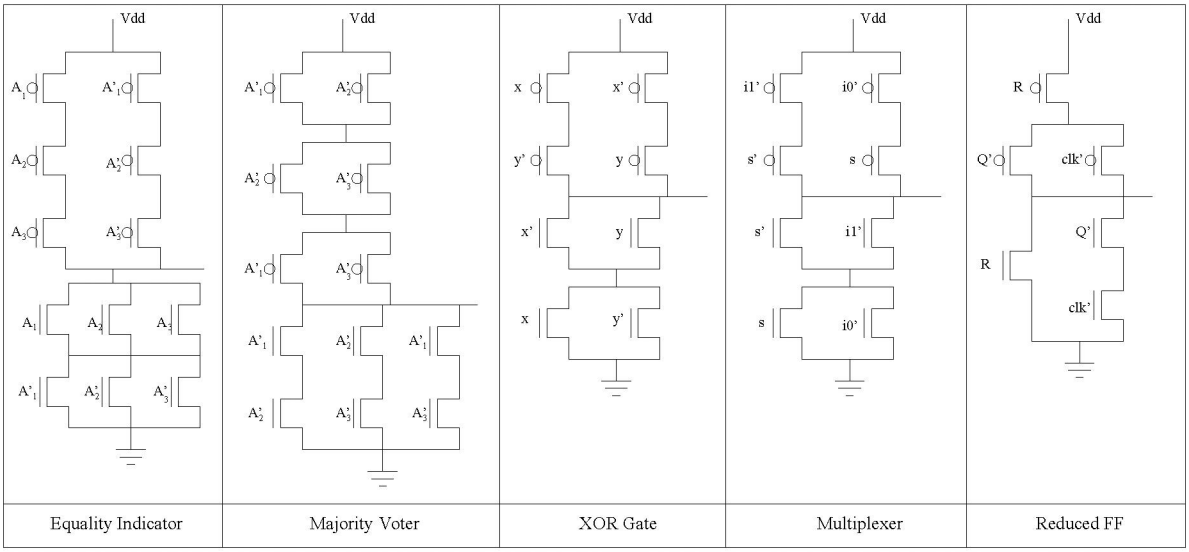


Figure 6. Low-level Implementation of the Logic Blocks

the feedback loop, forcing the output of the voter to retain its value. Once the error sensitization condition is no longer met, i.e., when all of the inputs to the voter agree, the "output freezing state" is exited by changing the state of the flip-flop to logic 0. This is performed using the equality indicator, which resets the state of the flip-flop. Subsequently, the multiplexer selects the output of the typical majority voter. The pieces within the hazard-free design, excluding the cost of a typical majority voter, constitute the area overhead.

A functional description of the proposed design is as follows. Subsequent to the *second* change in value of the inputs to the voter (A_1, A_2, and A_3), the proposed hazard-free voter is forced to retain its output value until the value of all the inputs match. Such a characteristic built-in the design exhibits resiliency to transient errors on a single input, which is the assumed worst-case scenario in a TMR design. For example, the output response of the proposed hazard-free TMR majority voter to the input stimulus in Figure 2 is illustrated in Figure 5. It can be verified from this figure that, subsequent to the change in value of A_2 from logic 1 to logic 0 due to an error, the voter blocks the error from propagating to its output and retains its value.

Finally, a low-level implementation of the logic blocks in the proposed hazard-free majority voter design is illustrated in Figure 6. Since the D input of the flip-flop is held to logic 1, the implementation of this logic block can be significantly simplified. Furthermore, in our actual design, we achieve further reduction in the area cost by implementing the complement of the majority voter as the multiplexer input, since the multiplexor requires the complement of the majority vote anyway. Thus, the implementation of the complement function is more cost-effective. In total, the proposed hazard-free TMR majority voter requires 64 transistors, including those necessary for inversions and for the delay unit.

5 Conclusions

Asynchronous circuits post new challenges to the applicability of common fault tolerance methods developed in the synchronous domain, such as TMR. As demonstrated in this work, the hazard-free property of the outputs of asynchronous circuits is compromised in a TMR-based fault tolerant system by the typical majority voter design. In order to enable the applicability of TMR for these circuits, we presented in this work a hazard-free majority voter design that is capable of retaining its output value in the presence of errors at its inputs. Therefore, transient errors affecting one of the circuit copies will not change the output value of the voter and, hence, will be tolerated.

References

[1] K. Mohanram and N. A. Touba, "Cost-effective approach for reducing soft error rate in logic circuits," in *IEEE International Test Conference*, 2003, pp. 893–901.

[2] M. Nicolaidis, "Design for soft error mitigation," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, pp. 405–418, 2005.

[3] S. Almukhaizim, Y. Makris, Y.-S. Yang, and A. Veneris, "Seamless integration of SER in rewiring-based design space exploration," in *IEEE International Test Conference*, 2006, pp. 29.3.1–29.3.9.

[4] Q. Zhou and K. Mohanram, "Gate sizing to radiation harden combinational logic," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 1, pp. 155–166, 2006.

[5] "ARM996HS^{TM}processor," `http://www.handshakesolutions.com/products_services/ARM996HS/Index.html`.

[6] R. Brees, "CLASS: Clockless logic, analysis, synthesis and systems - reducing the barrier to adoption of clockless design," in *IEEE International Symposium on Asynchronous Circuits and Systems*, 2004, Invited Talk.

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

[8] S. Almukhaizim and Y. Makris, "Concurrent error detection methods for asynchronous burst-mode machines," *IEEE Transactions on Computers*, vol. 56, no. 6, pp. 785–798, 2007.

[9] W. Jang and A. Martin, "SEU-tolerant QDI circuits," in *IEEE International Symposium on Asynchronous Circuits and Systems*, 2005, pp. 156–165.

[10] David E. Muller, "Asynchronous logics and application to information processing," in *Symposium on the Application of Switching Theory to Space Technology*. 1962, pp. 289–297, Stanford University Press.

[11] Y. Zhao and S. Dey, "Separate dual-transistor registers - a circuit solution for on-line testing of transient error in UDSM-IC," in *IEEE International On-Line Testing Symposium*, 2003, pp. 7–11.

[12] M. Omana, D. Rossi, and C. Metra, "Novel transient fault hardened static latch," *IEEE International Test Conference*, pp. 886–892, 2003.

[13] Y. Monnet, M. Renaudin, and R. Leveugle, "Designing resistant circuits against malicious faults injection using asynchronous logic," *IEEE Transactions on Computers*, vol. 55, no. 9, pp. 1104–1115, 2006.