www.ietdl.org

Published in IET Computers & Digital Techniques
Received on 18th August 2009
Revised on 17th August 2010
doi: 10.1049/iet-cdt.2009.0075

ISSN 1751-8601

Novel hazard-free majority voter for *N*-modular redundancy-based fault tolerance in asynchronous circuits

S. Almukhaizim[1] *O. Sinanoglu*[2]

[1]Computer Engineering Department, College of Engineering & Petroleum, Kuwait University, P.O. Box – 5969, Safat, Kuwait
[2]Computer Engineering Department, New York University Abu Dhabi, P.O. Box 129188, Abu Dhabi, UAE
E-mail: sobeeh.almukhaizim@ku.edu.kw

Abstract: *N*-modular redundancy (NMR) is the simplest and most effective fault-tolerant design method for integrated circuits, where *N* copies of a circuit are employed and a majority voter produces the voted output. Asynchronous circuits, however, exhibit various characteristics that limit the applicability of NMR. Specifically, the hazard-free property of the output in these circuits must be preserved when hardware providing fault tolerance, such as a majority voter, is added. In this work, we first demonstrate that a typical majority voter design would fail to preserve the hazard-free property of its response. We then propose a hazard-free majority voter design for the triple-modular redundancy fault-tolerance design paradigm, which enters an output-holding state to preserve the output value when transient errors may be sensitised to its inputs. By exploring various conditions to exit from the output-holding state, we describe several extensions of the voter into an NMR one, each yielding a distinct implementation with different tolerance characteristics and area cost. We generalise this extension based on the exit condition and analyse the associated tolerance capability of the extended NMR voter. Finally, the proposed hazard-free voter is simulated using HSPICE, and detailed area cost formulations are derived for the proposed voter designs.

1 Introduction

Clock-less designs promise improved characteristics over their synchronous counterparts, including power reduction, low EMI, and clock distribution network and skew problem elimination. As a result, several advanced asynchronous design styles have been carving an increasing niche in the market [1]. Moreover, their low power consumption and electromagnetic noise emission make them highly suitable in mission-critical applications such as avionics [2]. Nonetheless, the projected increase in the soft error failure rate of digital integrated circuits has sparked the development of numerous error protection strategies for synchronous [3–8] and asynchronous [9–18] circuits. In asynchronous circuits, however, the complexity in developing error protection strategies stems from the fact that these methods must be tailored to the particularities of the asynchronous design style under consideration. Specifically, the absence of a clock in asynchronous circuits leads to their design under certain communication protocols, which define the properties of the responses that the circuit will generate to its environment. Among the key properties of these protocols is the requirement that the changes in the outputs are hazard free. Since no clock is used, synchronisation between the circuit and its environment is based on the fact that any change in the output of the circuit signifies the completion of an evaluation cycle, and, accordingly, different methodologies to design hazard-free asynchronous circuits have been developed [19–22]. In essence, the hazard-free property implies that any fault-tolerance design strategy should not only preserve the functional correctness of the responses but also their hazard-free property. We stress the hazard-free requirement, as 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.

One of the most effective fault-tolerance design methods is *N*-modular redundancy (NMR) [23], where *N* copies of a circuit are utilised and a majority voter (per circuit output) produces the voted output. Consider, for example, triple-modular redundancy (TMR), which is a special case of NMR where $N = 3$. The different copies of the circuit produce their response with different delays, due to process variation, input skew and the sheer fact that the *N* circuits are separate entities [16]. Another common cause for variant delays is design diversity, that is, different implementations of the original design, which is often employed in NMR-based systems to protect against common-mode failures [24]. As the circuit copies operate with different delays, the output of the voter will change when two of its inputs (driven by the response of two of the circuit outputs) change their values. At this point of time, a transient error affecting the response of either one of these two circuit copies would change the output value of the voter erroneously. Once the third circuit copy produces its response (or the transient error disappears), the voter changes its output value to the correct response. While the above error scenario is of no significance in a synchronous TMR-based implementation (since the value of the voter is measured only when all circuit copies have produced their responses), the asynchronous counterpart would not tolerate this single error.

In this paper, we propose a hazard-free majority voter design to enable the application of NMR-based fault tolerance in asynchronous circuits. The proposed voter design enters an output-holding state that preserves the output value during the time window where transient errors may be sensitised to its output in the form of a hazard. We evaluate various conditions to exit from this output-holding state, each yielding a distinct implementation with different tolerance characteristics and area cost. We note that the proposed hazard-free majority voter is a mandatory component in order to enable the application of NMR in hazard-free asynchronous circuits, for any design style (e.g. delay insensitive, burst mode, speed independent etc.) and at any level of granularity (gate-level, block-level and system-level). Hence, our intention is not to use NMR as an added benefit to influence the choice of a particular asynchronous design style, or the most cost-effective level of granularity to apply it. Rather, and under the premise that an asynchronous implementation has been chosen owing to the advantages that it offers (e.g. power reduction, low EMI etc.), our intention is to develop necessary solutions to facilitate the application of NMR at any granularity level, the lack of which may be inhibiting this choice in the first place.

Several fault-tolerance methods for specific styles of asynchronous circuits have been previously proposed in the literature [12–18], all of which employ C-elements [25] to suppress errors. A C-element, which has also been used to tolerate errors in synchronous circuits [3, 4], is a state-holding component that waits for an input and a delayed version of it to agree on a logic value before it changes its state to this value. Whereas C-elements successfully tolerate transient errors in Quasi Delay-Insensitive (QDI) and burst-mode circuits, the fault-tolerant design suffers from several shortcomings when compared with an NMR-based implementation, albeit at the cost of higher area overhead. First, and since a C-element waits for all its inputs to agree before it changes its output value, the performance of the circuit is dependent on the duration of the transient error. An NMR-based design, however, produces the correct output when more than half of the circuit copies generate their response. Therefore the performance of an NMR-based fault-tolerant design is independent of the duration of transient errors. More importantly, C-elements cannot tolerate permanent errors in the circuit or its duplicate; hence, permanent errors force the fault-tolerant design built using C-elements into a deadlock state. NMR-based designs, however, would tolerate permanent errors in the circuit copies.

The paper is organised as follows. We start, in Section 2, by providing a research motivation to illustrate the problem of using a typical TMR majority voter in asynchronous circuits. In Section 3, we describe the proposed hazard-free TMR voter design, which utilises additional hardware in order to force the voter to retain its value after two of the circuit copies change their logic value. Subsequently, in Section 4, we illustrate the extension of the proposed hazard-free TMR voter design into a hazard-free NMR voter. Initially, we present two directions for this extension, each with different error-tolerance capabilities and area cost. Then, we generalise this extension in terms of the associated error-tolerance capability of the extended NMR voter. Finally, in Section 5, we provide a detailed area cost analysis for each of the presented extensions.

2 Motivation

Consider the transistor-level implementation of a TMR majority voter shown in Fig. 1*a*, and assume that A_1, A_2 and A_3 are the three input signals to the majority voter. In order to perform TMR-based fault tolerance for an asynchronous circuit, the circuit is triplicated and the outputs are connected as inputs to the majority voter, as illustrated in Fig. 1*b*. In this example, we simulate the output behaviour 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.) is

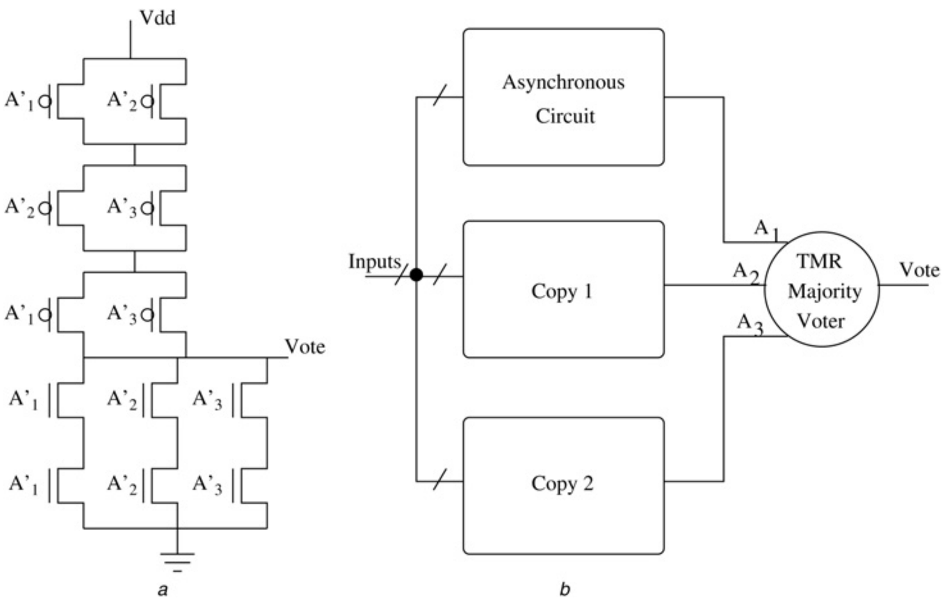

Fig. 1 *Typical TMR majority voter design and its use in a TMR-based fault-tolerant asynchronous circuit*

a Typical TMR majority voter design
b TMR implementation

doi: 10.1049/iet-cdt.2009.0075

required by incorporating proper fabrication model parameters. In these experiments, the simulations were performed using HSPICE and 500 nm AMI technology.

Assume that the outputs of the asynchronous circuits in Fig. 1*b* change from logic 0 to logic 1 during normal operation. Furthermore, assume that the change in these signals occurs 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 back to logic 0, as illustrated in Fig. 2. The output response of the majority voter for the above input stimulus is illustrated in Fig. 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 sensitised to 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 owing 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 owing to a transient error.

As demonstrated through this simulation example, transient errors sensitised to one of the inputs of the TMR majority voter may produce a hazard at its output. Since no clock is used in asynchronous designs, synchronisation 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 behaviour may be misinterpreted by the environment as a logic value change, resulting in erroneous system-level results. Therefore a hazard-free majority voter is required that is capable of producing the correct voted output, as well as preserving its logic value, when transient errors are sensitised to its inputs.

3 Proposed hazard-free TMR majority voter design

Towards designing a hazard-free majority voter, we first analyse the transient behaviour of a typical TMR majority voter and derive the hazard sensitisation condition wherein a single faulty circuit copy may lead to a hazard at the output of the voter. Then, building on this observation, we present the proposed hazard-free TMR voter design and discuss its implementation details.

3.1 Hazard sensitisation condition in TMR voters

A transient error may be sensitised to the output of a TMR majority voter, in the form of a hazard, depending on the values of the inputs to the voter when the error occurred. Thus, we need to distinguish the following two states

1. s_{safe}, the state where the outputs of all the three circuit copies agree, and
2. $s_{\text{sensitised}}$, the state where the outputs of two of the three circuit copies agree.

As the state names imply, a transient (or permanent) error affecting one of the circuit copies would not alter the output of the voter in s_{safe}. However, this error may lead to a hazard in $s_{\text{sensitised}}$. We elaborate next on the output behaviour of the voter when an error occurs in one of these two states.

Under the single-error assumption of TMR, the output of the voter is correct when the outputs of all the three circuit copies agree (i.e. s_{safe}) as no error occurs in any of the circuit copies. On the other hand, the output of the voter

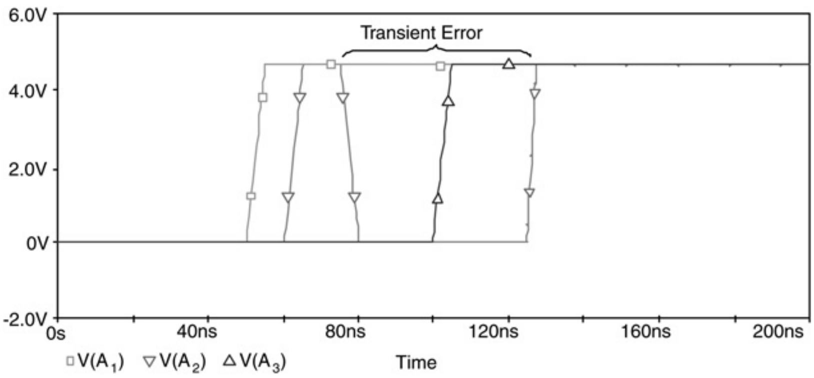

Fig. 2 *Input stimulus to the TMR majority voter with a transient error affecting input A_2*

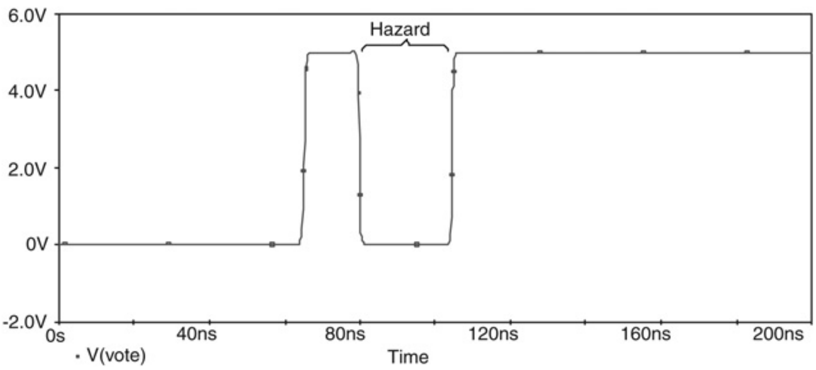

Fig. 3 *Output of the TMR majority voter*

doi: 10.1049/iet-cdt.2009.0075

might be correct when the outputs of only two copies agree ($s_{sensitised}$), depending on which circuit copy is defective and when the error is activated. In the first case (where the error is in the circuit copy that does not agree with the other two copies, as illustrated in Fig. 4*a*), the output of the voter is correct and hazard free, regardless of when the error was activated. In the second case (where the error is in one of the circuit copies that agree, as illustrated in Figs. 4*b* and *c*), the output of the voter is correct only if one of the two copies that agree produces the correct output value. Specifically, the output value of the voter in the presence of an error is correct at time t_2 in Fig. 4*b* and incorrect at time t_3 in Fig. 4*c*. The justification that the response of the voter in the event of premature firing is correct is based on the fact that the delays of the asynchronous circuit copies are not identical. Thus, when the output of the voter prematurely fires, one of the circuit copies must have produced the correct response, which is equal to the output of the majority voter. In both cases, the output of the voter is always correct when it first changes its value at time t_2. After this point of time, however, a potential transient (or newly occurring permanent) error affecting one of these two copies would be sensitised to the output of the voter, until either the third circuit copy produces the correct output value (e.g. t_4 in Figs. 4*b* and *c*) or the error is transient and disappears. Either way, a hazard appears in the response of the voter as the sensitisation condition is met, and hence the naming $s_{sensitised}$.

In order to preserve the hazard-free behaviour of the majority voter, the voter should be forced to retain its output value once the error sensitisation condition is detected (i.e. $s_{sensitised}$), thus forbidding any error-induced transitions at its output. We note that the output of the voter is interpreted by the environment as an acknowledgement of transitions at the output of all circuits, despite the fact that some of the circuits are still in computation. While the environment retracts the inputs to all circuits, the fundamental operation mode guarantees that the outputs stabilise before new inputs arrive. Hence, the basic phases of computation are guaranteed to be separated. This 'output-holding state' should continue until all the voter inputs agree, which, under the single-error assumption, is guaranteed to occur when the transient/permanent error is not sensitised to the otherwise disagreeing input of the voter (i.e. the error is not activated or propagated). At this point, no error in a single circuit copy can reflect to the voter output anyways and, therefore the 'output-holding state' can be exit safely. We describe in the next section how to control the output of the majority voter by entering and exiting the 'output-holding state'.

3.2 Proposed voter design

The proposed hazard-free TMR majority voter design is illustrated in Fig. 5. 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 *i*1 when the error sensitisation condition is met (i.e. the 'output-holding state' should be entered), and the output of the typical majority voter at *i*0, 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 (four back-to-back inverters), which together form a transition detector circuit. A transition on the output of the hazard-free voter indicates that the error sensitisation 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-holding state'. At this point, the multiplexer sensitises the feedback loop, forcing the output

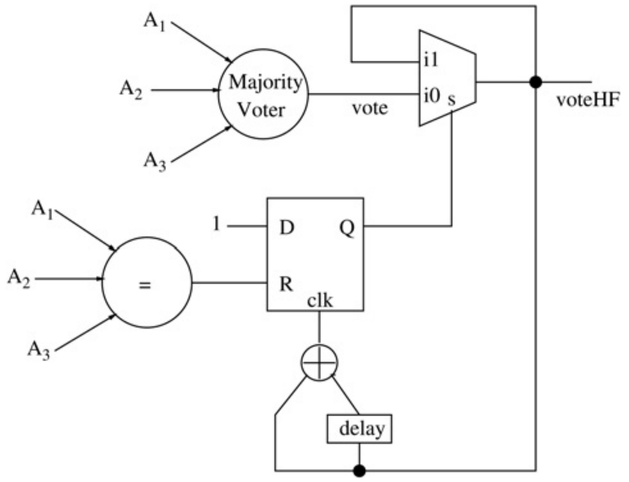

Fig. 5 *High-level design of the proposed hazard-free TMR majority voter*

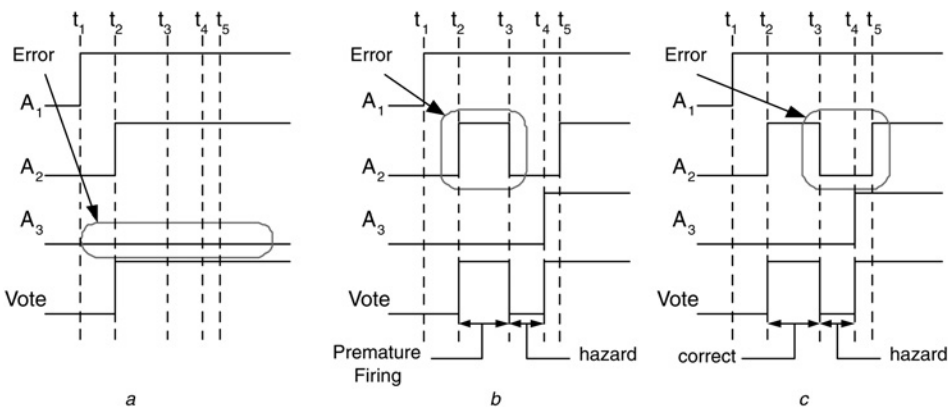

Fig. 4 *Possible input stimulus to the TMR majority voter when two circuit copies agree*

a Error scenario in A_3; vote signal is hazard free
b Error scenario in A_2; premature firing and hazard in vote signal
c Error scenario in A_2; hazard in vote signal

of the voter to retain its value. Once the error sensitisation condition is no longer met (i.e. when all of the inputs to the voter agree), the voter exits the 'output-holding state' by changing the state of the flip-flop to logic 0. This is performed by using the equality indicator, which resets the state of the flip-flop. Subsequently, the multiplexer selects the output of the typical majority voter until the next entry to the 'output-holding state'.

A functional description of the proposed design is as follows. After two of the inputs to the voter (A_1, A_2 and A_3) change their value, the 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 any transient/permanent error sensitised to 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 Fig. 2 is illustrated in Fig. 6. It can be seen from this figure that, subsequent to the change in the value of A_2 from logic 1 to logic 0 owing 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 Fig. 7. Since the D input of the flip-flop is held to logic 1, the implementation of this logic block can be 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 multiplexer 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.

3.3 Implementation issues

In this section, we investigate the implementation issues associated with the design of the proposed majority voter, which are summarised and discussed below:

- *Delay element:* The delay at the input of the XOR gate is designed by using four back-to-back inverters, which generate a pulse at the output of the XOR gate (i.e. the clock of the FF in Fig. 5) with width equal to the delay of four inverters. A pulse of such width satisfies the pulse width and set-up constraints of the custom FF, which is equal to the delay of three inverters (notice that the *R* signal will always change before the clk signal in the reduced FF illustrated in Fig. 7). Hence, and since the delay of four inverters is greater than the minimum pulse requirement on

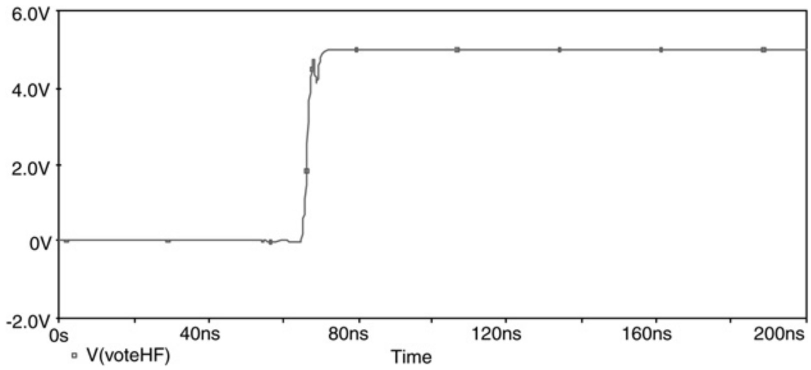

Fig. 6 *Output of the proposed hazard-free majority voter*

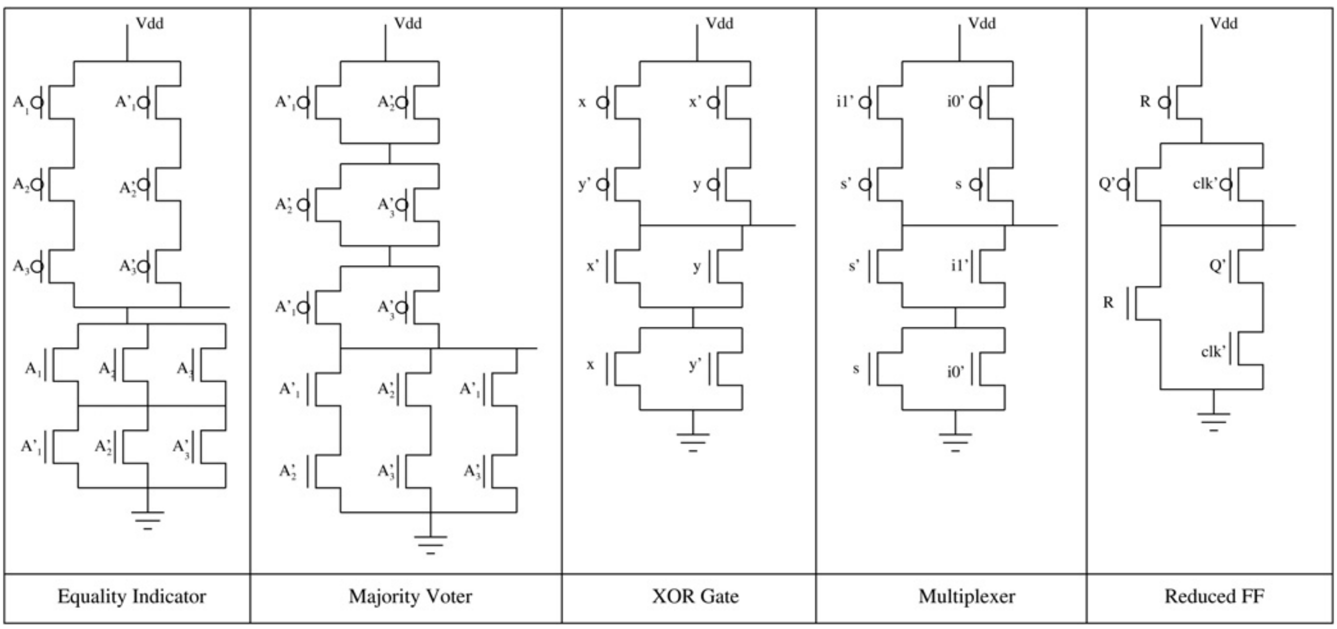

Fig. 7 *Low-level implementation of the logic blocks*

IET Comput. Digit. Tech., 2011, Vol. 5, Iss. 4, pp. 306–315
doi: 10.1049/iet-cdt.2009.0075

the clk input of the FF, the correct switching at the output of the FF is ensured. We note that the delay assumptions based on the use of four inverters render the design problematic to delay faults, in which case additional delay can also be added without affecting the functionality of the voter.

- *Short hazards:* A hazard produced at the output of the typical majority voter in the proposed hazard-free voter design may propagate to the output of the multiplexer, if the duration of the hazard is less than the delay necessary for the multiplexer to select $i1$ instead of $i0$. Hence, a race condition exists between the vote signal and the select line of the multiplexer. Essentially, this implies that hazards shorter than the necessary time for the multiplexer to switch from one data input to the other one would propagate to the output of the proposed voter (and may lead to the metastability and/or oscillation in the proposed design). Specifically, the duration of this time window equals the sum of the delays of the XOR gate, the FF and the 0-to-1 transition of the multiplexer, which is equal to the delay of five inverters. In order to suppress all short hazards, a two-input C-element is inserted between the output of the typical majority voter (vote in Fig. 5) and input $i0$ of the multiplexer. The inputs to the C-element are the vote signal, and a delayed version of the vote signal. By appropriately controlling this delay, all short hazards are suppressed. We have performed several HSPICE simulations of the proposed hazard-free voter design, implemented using the 500 nm AMI technology, and identified that short hazards of duration between 0.6 and 0.9 ns would propagate to the output of the voter. When the C-element was inserted and the vote signal was delayed using four back-to-back inverters, all short hazards were suppressed, producing the same hazard-free output behaviour illustrated in Fig. 6. Unlike the shortcomings of C-elements when used as a stand-alone voter, the use of a C-element as part of the proposed majority voter is highly effective in tolerating transient errors when sensitised to its inputs; its inability to tolerate permanent errors poses no limitation on the ability of the proposed voter design, which includes the C-element, to tolerate short hazards.
- *Signal ordering of R and clk:* The reduced FF implementation gives priority to the R input over the clk input. Thus, and even if the clk transition aligns with the reset of the FF, the FF will operate correctly.

We note that the proposed hazard-free majority voter is the Achilles heel of any fault-tolerant method in asynchronous circuits that utilises the proposed design, just as a typical majority voter is the weak point of TMR; an error affecting the voter (the hazard-free design or the typical one) in both cases can neither be detected nor corrected, regardless of the source of the error (e.g. stuck-at fault, delay fault etc.). The above discussion also holds in the implementation of the proposed NMR extensions that are described next.

4 Extension to hazard-free NMR majority voter design

We present herein several extension methods of the hazard-free TMR voter design to an NMR one, with each method differing in the condition for exiting from the 'output-holding state'. Therefore each of these extensions requires a different area cost (in identifying the associated exit condition) and provides a different tolerance level. First, in Section 4.1, we present a direct extension of the hazard-free TMR voter to an NMR one that tolerates $\lfloor N/2 \rfloor$ transient errors but only a single permanent error. Then, in Section 4.2, we present a modified extension of the hazard-free TMR voter to an NMR one that tolerates $\lceil (N-1)/4 \rceil$ transient errors and $\lfloor N/2 \rfloor$ permanent errors. Finally, in Section 4.3, we generalise the condition for exiting from the 'output-holding state', resulting in a formulation that defines the tolerance capability of any voter in terms of transient and permanent errors. The theorem may be used to identify the necessary exit condition that delivers the tolerance capability for the voter as desired by the designer based on the criticality of the application and its operating environment.

4.1 Direct extension

The proposed hazard-free TMR majority voter is capable of tolerating a single error. While we focused our attention on the tolerance of transient errors in the simulation in Section 3, it can be easily verified that the proposed TMR voter design is also capable of tolerating a single permanent error, which is the capability expected out of any TMR voter. For a hazard-free NMR majority voter, the objective is to tolerate $\lfloor N/2 \rfloor$ permanent and/or transient errors. As will be elaborated on subsequently, direct extension of the hazard-free TMR voter design in Section 3 would result in an implementation that tolerates $\lfloor N/2 \rfloor$ transient errors but only a single permanent error.

The hazard-free TMR majority voter design in Fig. 5 can be extended to an NMR one by simply extending the two three-input blocks (namely, the majority voter and the equality indicator) into N-input blocks. In this direct extension, the first transition on the vote output, which occurs after the $\lceil N/2 \rceil$th transition on the inputs, results in an entry into the 'output-holding state'. With the extension of the equality indicator block, the exit to the state happens only when all of the N inputs agree. Thus, such a design is capable of tolerating $N - \lceil N/2 \rceil$, or equivalently, $\lfloor N/2 \rfloor$ (as N is odd) transient errors sensitised to the inputs of the voter.

Unfortunately, the above direct extension is unable to tolerate $\lfloor N/2 \rfloor$ permanent errors. Specifically, if permanent errors occur in two circuit copies, with one sensitised as a stuck-at-0 on one voter input and the other one as a stuck-at-1 on another input, the hazard-free voter erroneously remains in the 'output-holding state' forever, as all inputs would never agree. As a result, the majority voter would fail to reflect to the changes in its inputs (i.e. the outputs of the circuit and its copies), rendering the design incapable of tolerating these permanent errors that are stuck-at in different directions. This cannot be the case in the proposed hazard-free TMR design where the worst-case scenario foresees a single permanent error, which does not suffice to keep the voter in the 'output-holding state' forever; when the other error-free inputs receive a value that is identical to the erroneous input's stuck-at value, all the inputs to the voter agree, leading to an exit from the state. In the direct extension scheme described above, permanent errors in two circuit copies can still be tolerated, if they are sensitised as stuck-at in the same direction. When the remaining error-free inputs change their value to that of the stuck-at value, the voter exits the 'output-holding state' and this value is voted for. The tolerance of a single permanent error, however, is the capability that holds even in the worst-case scenario for the direct extension case. Thus, considering all possible error sensitisation scenarios, direct extension of the TMR voter to an NMR one guarantees the tolerance of a single permanent error only.

The tolerance capability of the direct extension approach of the hazard-free TMR voter design is summarised via the two lemmas below.

Lemma 1: Exiting the 'output-holding state' when all N inputs agree ensures the tolerance of a single permanent error.

Lemma 2: Exiting the 'output-holding state' when all N inputs agree ensures the tolerance of $\lfloor N/2 \rfloor$ transient errors.

The above analysis indicates that the problematic case for permanent errors is when at least two errors are sensitised to the inputs of the voter as stuck-at in different directions, which prevents the exit from the 'output-holding state'. Therefore the TMR to NMR extension needs to be modified such that the voter has the capability to exit from the 'output-holding state' even when permanent errors are stuck-at in different directions.

4.2 Modified extension

As the condition for exiting from the 'output-holding state' determines the permanent and transient error-tolerance capabilities of the voter, we describe next an alternative exit condition that maximises the number of permanent and transient errors that can be tolerated.

In order to handle stuck-at errors in different directions, the exit condition can be modified such that the voter exits the 'output-holding state' when s inputs agree, where $\lceil N/2 \rceil \le s \le N$. Assume that v permanent errors are sensitised as stuck-at in different directions, with i errors in one direction and j errors in the other. Then, the voter will not exit the 'output-holding state' (i.e. permanent errors would not be tolerated) if $N - s < \min(i, j)$, where min() is the minimum function. Therefore the worst-case scenario is when permanent errors are evenly divided into two groups, each sensitised as stuck-at in a different direction. Hence, in order to tolerate $\lfloor N/2 \rfloor$ permanent errors, the exit condition should account for the worst-case scenario where $\lfloor \lfloor N/2 \rfloor /2 \rfloor$ erroneous inputs are stuck-at one value, and the other $\lceil \lfloor N/2 \rfloor /2 \rceil$ inputs are stuck-at the other value.

In the following two lemmas, we modify the exit condition in order to ensure the tolerance of $\lfloor N/2 \rfloor$ permanent errors, while maximising the number of transient errors that can be tolerated.

Lemma 3: Exiting the 'output-holding state' when $\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ inputs agree ensures the tolerance of $\lfloor N/2 \rfloor$ permanent errors.

Lemma 4: Exiting the 'output-holding state' when $\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ inputs agree ensures the tolerance of $\lceil (N-1)/4 \rceil$ transient errors.

The hardware implementation of the hazard-free NMR voter is effected by replacing the equality indicator in Fig. 5 with another logic block that implements '$\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ inputs out of N inputs agree'; when at least $\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ inputs have the same logic value, this block outputs a value of one, and a zero otherwise. Also by extending the three-input majority voter in the same figure to an N-input majority voter, a hazard-free NMR majority voter with the aforementioned tolerance capabilities can be implemented. It is quite interesting to note that the proposed hazard-free TMR majority voter is a degenerate version of the NMR majority voter design described in this section. A block that implements '$\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ out of N inputs agree' degenerates to an equality indicator when N is substituted by 3.

4.3 Generalised extension

In this section, we generalise the TMR to NMR voter extension based on the exit condition and analyse the associated error-tolerance capability of the extended voter. We capture this generalisation in the following theorem:

Theorem: Exiting the 'output-holding state' when $\lceil N/2 \rceil + k$ (where $0 \le k \le \lfloor N/2 \rfloor$) inputs agree ensures the tolerance of:

(a) k transient errors.
(b) v permanent errors, where $\lfloor v/2 \rfloor \le \lfloor N/2 \rfloor - k$ and $v \le \lfloor N/2 \rfloor$.
(c) A combination of k transient and v permanent errors, where $\lfloor v/2 \rfloor \le \lfloor N/2 \rfloor - k$ and $v \le \lfloor N/2 \rfloor$.

Proof: The proof of the theorem is provided in the Appendix section at the end of the paper.

This theorem captures the potential extension types for any possible condition to exit from the 'output-holding state'. When $k = 0$, the generalised extension is no different than a typical NMR majority voter, never entering the 'output-holding state' and thus capable of tolerating $\lfloor N/2 \rfloor$ permanent errors, but no transient errors. When $k = \lfloor N/2 \rfloor$, the generalised extension degenerates to the direct extension, as $\lfloor N/2 \rfloor + \lceil N/2 \rceil = N$. For the inequality in the theorem to hold, $v = 1$ permanent errors can be tolerated, which is consistent with Lemma 1. Also, the theorem states that $k = \lfloor N/2 \rfloor$ transient errors can be tolerated, which is also consistent with Lemma 2. When $k = \lceil (N-1)/4 \rceil$, the generalised extension degenerates to the modified extension. The substitution of k with $\lceil (N-1)/4 \rceil$ in the inequality of the Theorem results in

$$\left\lfloor \frac{v}{2} \right\rfloor \le \left\lfloor \frac{N}{2} \right\rfloor - \left\lceil \frac{N-1}{4} \right\rceil$$

$$\left\lfloor \frac{v}{2} \right\rfloor \le \left\lfloor \frac{\lfloor N/2 \rfloor}{2} \right\rfloor$$

$$v \le \left\lfloor \frac{N}{2} \right\rfloor$$

where v permanent errors can be tolerated, which is consistent with Lemma 3. Furthermore, as $k = \lceil (N-1)/4 \rceil$, this many transient errors can be tolerated, which is also consistent with Lemma 4. Finally, and since the theorem captures all the possible conditions to exit from the 'output-holding state', no possible exit condition would allow the tolerance of $\lfloor N/2 \rfloor$ permanent and/or transient errors.

5 Area cost analysis

In this section, we quantify the area overhead of the proposed hazard-free NMR majority voter designs based on the presented extensions. The area overhead herein denotes the additional area cost required to make a typical majority voter operate in a hazard-free fashion. While the utilisation of a typical majority voter (or the proposed one) with N larger than 5 is impractical because of area and performance overheads, we nonetheless provide these results for completeness purposes.

The ON-set of a typical majority voter expands exponentially in N. Specifically, $\binom{N}{\lceil \frac{N}{2} \rceil}$ product terms, each with $\lceil N/2 \rceil$ literals, exist in this ON-set. Assuming

IET Comput. Digit. Tech., 2011, Vol. 5, Iss. 4, pp. 306–315
doi: 10.1049/iet-cdt.2009.0075

Table 1 Hardware cost of hazard-free NMR voter against the typical voter (direct extension)

Number of inputs	Typical majority voter	Proposed hazard-free majority voter	Area overhead, %
3	14	64	357.14
5	62	124	100.00
7	282	356	26.24
⋮	⋮	⋮	⋮
N	$2 + 2 \times \lceil \frac{N}{2} \rceil \times \binom{N}{\lceil \frac{N}{2} \rceil}$	$34 + 6 \times N + 2 \times \left\lceil \frac{N}{2} \right\rceil \times \binom{N}{\lceil \frac{N}{2} \rceil}$	$100 \times \dfrac{32 + 6 \times N}{2 + 2 \times \lceil N/2 \rceil \times \binom{N}{\lceil N/2 \rceil}}$

that no logic optimisation is effected in the implementation of a majority voter, the number of transistors in this design will be directly proportional to the product of the two terms above. An equality indicator has a two-element ON-set, with each product term having N literals. Thus, the implementation cost of this block with no logic optimisation will be proportional to N.

The area cost, in terms of the number of transistors, of a typical NMR majority voter, the proposed hazard-free NMR voter (direct extension), and the percentile area overhead are summarised in Table 1. As can be seen in the table, the area overhead rapidly drops from 357.14% down to 26.24%, when N goes from three to seven. The reason is that for larger values of N, the typical majority voter cost becomes the dominating part in the proposed design. This is further illustrated using Fig. 8, where we plot the area overhead of the proposed hazard-free majority voter against the number of circuit copies, that is, N. The curve indicates that the percentile area overhead reduces significantly as N increases.

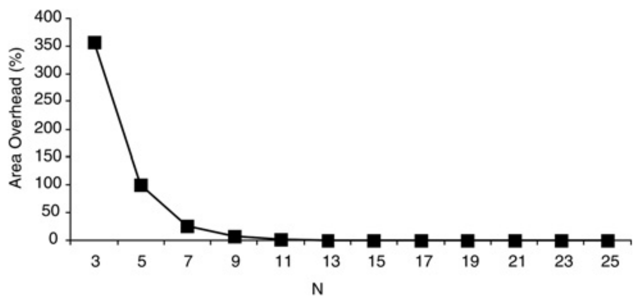

Fig. 8 *Area overhead (%) against N in the case of direct extension*

In the implementation of the '$\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ inputs out of N inputs agree' block, $\binom{N}{\lceil N/2 \rceil + \lceil (N-1)/4 \rceil}$ inputs may agree at either logic value, resulting in twice as many product terms in the ON-set, each with $\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ literals. Thus, the implementation cost of this block with no logic optimisation will be proportional to the product of these two terms. This cost along with that of a typical majority voter and the other logic blocks in Fig. 5 constitute the area cost of the proposed hazard-free NMR majority voter (modified extension).

Table 2 provides results analogous to those in Table 1, this time for the case of the modified extension. Again, for larger values of N, the typical majority voter cost becomes the dominating part in the proposed design. However, the area overhead is higher than that of the direct extension design, as the logic function '$\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ inputs out of N inputs agree' is more expensive compared to a simple equality indicator. The area overhead drops from 357.14% down to 75.89%, when N goes from 3 to 7. This is further illustrated using Fig. 9. While the general trend is reduction in area overhead, a slight increase followed by a sharper decrease occurs consistently for $N \geq 7$. The area overhead

Table 2 Hardware cost of the hazard-free NMR voter against the typical voter (modified extension)

Number of inputs	Typical majority voter	Proposed hazard-free majority voter	Area overhead, %
3	14	64	357.14
5	62	184	196.77
7	282	496	75.89
⋮	⋮	⋮	⋮
N	$2 + 2 \times \left\lceil \frac{N}{2} \right\rceil \times \binom{N}{\lceil \frac{N}{2} \rceil}$	$34 + 2 \times N + 2 \times \left\lceil \frac{N}{2} \right\rceil \times \binom{N}{\lceil \frac{N}{2} \rceil} + 4 \times \binom{N}{\lceil \frac{N}{2} \rceil + \lceil \frac{N-1}{4} \rceil} \times \left(\left\lceil \frac{N}{2} \right\rceil + \left\lceil \frac{N-1}{4} \right\rceil \right)$	$100 \times \dfrac{32 + 2 \times N + 4 \times \binom{N}{\lceil \frac{N}{2} \rceil + \lceil \frac{N-1}{4} \rceil} \times (\lceil N/2 \rceil + \lceil (N-1)/4 \rceil)}{2 + 2 \times \lceil N/2 \rceil \times \binom{N}{\lceil \frac{N}{2} \rceil}}$

doi: 10.1049/iet-cdt.2009.0075

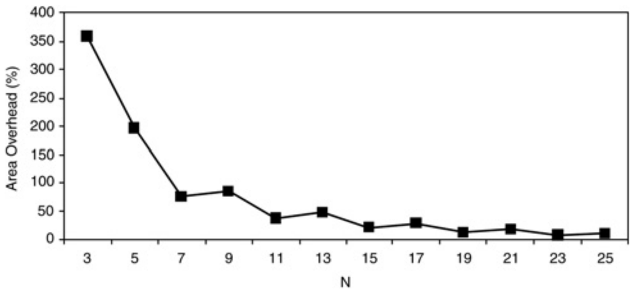

Fig. 9 *Area overhead (%) against N in the case of modified extension*

formula in Table 2 shows that the $\lceil (N-1)/4 \rceil$ term appears in the numerator, which causes an irregular increase for values of N in increments of 4. The sharper increase in the denominator, however, is more regular, as $\lceil N/2 \rceil$ grows linearly with the odd values of N. Consequently, when the increase in the numerator overcomes the increase in the denominator, local deviations from the global trend ensue for every other odd value of N.

Finally, the area cost of the proposed hazard-free NMR majority voter in the case of the generalised extension is formulated by replacing each of the $\lceil (N-1)/4 \rceil$ terms in Table 2 with k as follows

$$\mathrm{Area}_{\text{generalised extension}\,(k)} = 34 + 2 \times N + 2 \times \left\lceil \frac{N}{2} \right\rceil \times \binom{N}{\left\lceil \frac{N}{2} \right\rceil} + 4 \times \binom{N}{\left\lceil \frac{N}{2} \right\rceil + k} \times \left(\left\lceil \frac{N}{2} \right\rceil + k \right)$$

where $0 \leq k \leq \lfloor N/2 \rfloor$. The formula above evaluates the area cost of the proposed hazard-free NMR majority voter for different values of k. As mentioned in the previous section, the selection of a particular value of k dictates the number of permanent and transient errors tolerated by the proposed majority voter design.

6 Conclusions

Asynchronous circuits post new challenges to the applicability of common fault-tolerance methods developed in the synchronous domain, such as the NMR design paradigm. As demonstrated in this work, the hazard-free property of the outputs of asynchronous circuits is compromised in an NMR-based fault-tolerant system when a typical majority voter design is utilised. In order to enable the applicability of NMR for these circuits, we present in this work a hazard-free majority voter design that is capable of retaining its output value when errors are sensitised to its inputs. Therefore the majority voter achieves resiliency to transient errors that may occur in the circuit copies and reflect to voter output in the form of a hazard otherwise.

7 Acknowledgment

Preliminary version of this paper was presented by the authors at the International Design and Test Workshop [26].

8 References

1 ARM996HS™processor: available at http://www.handshakesolutions.com/products_services/ARM996HS/l

2 Brees, R.: 'CLASS: clockless logic, analysis, synthesis and systems – reducing the barrier to adoption of clockless design'. IEEE Int. Symp. on Asynchronous Circuits and Systems, 2004, Invited Talk

3 Zhao, Y., Dey, S.: 'Separate dual-transistor registers – a circuit solution for on-line testing of transient error in UDSM-IC'. IEEE Int. On-Line Testing Symp., 2003, pp. 7–11

4 Omana, M., Rossi, D., Metra, C.: 'Novel transient fault hardened static latch'. IEEE Int. Test Conf., 2003, pp. 886–892

5 Mohanram, K., Touba, N.A.: 'Cost-effective approach for reducing soft error rate in logic circuits'. IEEE Int. Test Conf., 2003, pp. 893–901

6 Nicolaidis, M.: 'Design for soft error mitigation', *IEEE Trans. Device Mater. Reliab.*, 2005, **5**, (3), pp. 405–418

7 Almukhaizim, S., Makris, Y., Yang, Y.-S., Veneris, A.: 'Seamless integration of SER in rewiring-based design space exploration'. IEEE Int. Test Conf., 2006, pp. 29.3.1–29.3.9

8 Zhou, Q., Mohanram, K.: 'Gate sizing to radiation harden combinational logic', *IEEE Trans. Comput.-Aided Des. Integr. Circuits and Syst.*, 2006, **25**, (1), pp. 155–166

9 Piestrak, S.J., Nanya, T.: 'Towards totally self-checking delay-insensitive systems'. Int. Fault-Tolerant Computing Symp., 1995, pp. 228–237

10 Yang, J.-L., Choy, C.-S., Chan, C.-F., Pun, K.-P.: 'Design for self-checking and self-timed datapath'. IEEE VLSI Test Symp., 2003, pp. 417–422

11 Hyde, P.D., Russell, G.: 'ASSEC: an asynchronous self-checking RISC-based processor'. Euromicro Symp. on Digital System Design, 2004, pp. 104–111

12 Jang, W., Martin, A.: 'SEU-tolerant QDI circuits'. IEEE Int. Symp. on Asynchronous Circuits and Systems, 2005, pp. 156–165

13 Monnet, Y., Renaudin, M., Leveugle, R.: 'Asynchronous circuits transient faults sensitivity evaluation'. ACM/IEEE Design Automation Conf., 2005, pp. 863–868

14 Monnet, Y., Renaudin, M., Leveugle, R.: 'Hardening techniques against transient faults for asynchronous circuits'. IEEE Int. On-Line Testing Symp., 2005, pp. 129–134

15 Monnet, Y., Renaudin, M., Leveugle, R.: 'Designing resistant circuits against malicious faults injection using asynchronous logic', *IEEE Trans. Comput.*, 2006, **55**, (9), pp. 1104–1115

16 Almukhaizim, S., Makris, Y.: 'Concurrent error detection methods for asynchronous burst-mode machines', *IEEE Trans. Comput.*, 2007, **56**, (6), pp. 785–798

17 Almukhaizim, S., Shi, F., Makris, Y.: 'Coping with soft errors in asynchronous burst mode machines'. IEEE Int. Symp. on Asynchronous Circuits and Systems, 2008, pp. 151–160

18 Almukhaizim, S., Shi, F., Love, E., Makris, Y.: 'Soft error tolerance and mitigation in asynchronous burst-mode circuits', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2009, **17**, (7), pp. 869–882

19 Huffman, D.A.: 'The synthesis of sequential switching networks' (Addison-Wesley, 1964)

20 Unger, S.H.: 'Asynchronous sequential switching circuits' (Wiley-Interscience, 1969)

21 Eichelberger, E.B.: 'Hazard detection in combinational and sequential switching circuits', *IBM J. Res. Dev.*, 1965, **9**, (2), pp. 90–99

22 Fuhrer, R.M., Nowick, S.M.: 'Sequential optimization of asynchronous and synchronous finite-state machines: algorithms and tools' (Kluwer Academic Publishers, 2001)

23 von Neumann, J.: 'Probabilistic logics and synthesis of reliable organisms from unreliable components', in Shannon, C.E., McCarthy, J. (Eds.): 'Automata studies' (Princeton University Press, 1956), pp. 43–98

24 Mitra, S., Saxena, N.R., McCluskey, E.J.: 'A design diversity metric and reliability analysis for redundant systems'. IEEE Int. Test Conf., 1999, pp. 662–671

25 Muller, D.E.: 'Asynchronous logics and application to information processing'. Symp. on Application of Switching Theory to Space Technology, 1962, pp. 289–297

26 Almukhaizim, S., Sinanoglu, O.: 'A hazard-free majority voter for TMR-based fault tolerance in asynchronous circuits'. IEEE Int. Design and Test Workshop, 2007

9 Appendix

This appendix contains proofs of lemmas and the theorem contained in Section 4 of the paper.

doi: 10.1049/iet-cdt.2009.0075

Proof of Lemma 3:
Proof: In the case of $\lfloor N/2 \rfloor$ permanent errors, $\lceil N/2 \rceil$ error-free inputs exist. Furthermore, the worst-case scenario is when $\lceil \lfloor N/2 \rfloor /2 \rceil = \lceil (N-1)/4 \rceil$ (as N is odd) erroneous inputs are stuck-at in one direction and the remaining $\lfloor (N-1)/4 \rfloor$ erroneous inputs are stuck-at in the other direction. When the majority voter output changes to the value that $\lceil (N-1)/4 \rceil$ erroneous inputs are stuck-at, it enters the 'output-holding state'. Upon the completion of all the error-free input changes, $\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ inputs would be agreeing. Allowing the voter to exit from the 'output-holding state' at this point enables the voter to respond to the changes in the error-free inputs. When the voter output changes to the other value, in which $\lfloor (N-1)/4 \rfloor$ erroneous inputs are stuck-at, it enters the 'output-holding state' again. This time, upon the completion of all the error-free input changes, there would be $\lceil N/2 \rceil + \lfloor (N-1)/4 \rfloor$ inputs agreeing. Even if the voter remains in the 'output-holding state', it would be voting for the correct value, in which the majority of the inputs agree. This would continue until the error-free inputs change back to the former value, in which $\lceil (N-1)/4 \rceil$ erroneous inputs are stuck-at. The voter is guaranteed to exit from the 'output-holding state' then, as explained above. Thus, $\lfloor N/2 \rfloor$ permanent errors can be tolerated if $\lceil N/2 \rceil + \lceil (N-1)/4 \rceil$ input agreement is the condition to exit from the 'output-holding state'. □

Proof of Lemma 4:
Proof: The 'output-holding state' is entered after the $\lceil N/2 \rceil$th transition in the voter inputs. Transient errors, which create transitions in the direction opposite to the error-free transitions, can be tolerated as long as the voter remains in this state; with a maximum of $\lceil (N-1)/4 \rceil$ potential transient errors, $(N - \lceil N/2 \rceil) + \lceil (N-1)/4 \rceil = \lfloor N/2 \rfloor + \lceil (N-1)/4 \rceil$ inputs will be agreeing, in which case the voter still remains in the 'output-holding state'. With one more transient error, however, the voter exits the state. Thus, $\lceil (N-1)/4 \rceil$ transient errors can be tolerated. □

Proof of Theorem:
Proof: (a) After $\lceil N/2 \rceil$ inputs of the voter change, which is when the voter output also changes, the voter enters the 'output-holding state'. Forcing the voter to remain in this state until $\lceil N/2 \rceil + k$ inputs agree ensures that even with k transient errors, there would be at least $\lceil N/2 \rceil + k - k = \lceil N/2 \rceil$ inputs agreeing at the correct value. Thus, the voter still votes for the correct value in this case. The other extremal case is when k inputs are affected by transient errors right after the $\lceil N/2 \rceil$th change in the error-free inputs of the voter. In this case, the voter still votes for the correct value that $\lceil N/2 \rceil - k$ inputs have; the incorrect value that the remaining $\lfloor N/2 \rfloor + k$ inputs have is not voted for, as an agreement among this many inputs does not suffice to exit from the 'output-holding state'. Thus, k transient errors can be tolerated.
(b) In the presence of v permanent errors, $N - v$ error-free inputs exist. In the worst-case scenario, $\lceil v/2 \rceil$ of the erroneous inputs would be stuck-at in the same direction. Thus, in order to tolerate v permanent errors, $N - v + \lceil v/2 \rceil$ input agreement should suffice to exit from the 'output-holding state', preventing the voter to remain in this state forever. In other words

$$N - v + \left\lceil \frac{v}{2} \right\rceil \geq \left\lceil \frac{N}{2} \right\rceil + k$$

$$v - \left\lceil \frac{v}{2} \right\rceil \leq N - \left\lceil \frac{N}{2} \right\rceil - k$$

$$\left\lfloor \frac{v}{2} \right\rfloor \leq \left\lfloor \frac{N}{2} \right\rfloor - k$$

must hold. Also, v must be upper bounded by $\lfloor N/2 \rfloor$, as more than this many permanent errors would obviously result in an incorrect value being voted for by the majority voter.
(c) Transient errors have no impact on whether or not the proposed voter will remain in the 'output-holding state' forever. When transient errors disappear, and they will by definition, it is the permanent errors that may potentially keep the voter in the 'output-holding state'. As stated in part (b) of the theorem, as long as $\lfloor v/2 \rfloor \leq \lfloor N/2 \rfloor - k$ and $v \leq \lfloor N/2 \rfloor$, the voter will exit this state. Thus, v permanent errors can be tolerated despite the additional transient errors.

When $\lceil N/2 \rceil$ inputs agree, the voter enters the 'output-holding state'. The fact that some permanent errors are sensitised to the agreeing inputs, and the remaining permanent errors are sensitised to the disagreeing inputs is completely irrelevant when the impact of transient errors is being investigated. For the voter to exit erroneously from the 'output-holding state', the number of disagreeing inputs must increase from $\lfloor N/2 \rfloor$ to $\lceil N/2 \rceil + k$. Thus, at least $\lceil N/2 \rceil + k - \lfloor N/2 \rfloor = k + 1$ transient errors must exist. On the other extremal case, the voter exits the 'output-holding state' upon the agreement of $\lceil N/2 \rceil + k$ inputs. At this point, for the voter to vote for the erroneous value, the number of inputs agreeing on the correct value should drop down to $\lfloor N/2 \rfloor$. Thus, $\lceil N/2 \rceil + k - \lfloor N/2 \rfloor = k + 1$ transient errors must exist in this case also. Consequently, as long as the number of transient errors is no more than k, they can all be tolerated despite the additional permanent errors. □

doi: 10.1049/iet-cdt.2009.0075