Majority and minority voted redundancy scheme for safety-critical applications with error/no-error signaling logic

Balasubramanian, Padmanabhan; Maskell, Douglas; Mastorakis, Nikos

2018

Balasubramanian, P., Maskell, D., & Mastorakis, N. (2018). Majority and Minority Voted Redundancy Scheme for Safety-Critical Applications with Error/No-Error Signaling Logic. Electronics, 7(11), 272-. doi:10.3390/electronics7110272

https://hdl.handle.net/10356/88763

https://doi.org/10.3390/electronics7110272 electronics

MDPI

Article

Majority and Minority Voted Redundancy Scheme for Safety-Critical Applications with Error/No-Error Signaling Logic [†]

Padmanabhan Balasubramanian [1,*], Douglas Maskell [1] and Nikos Mastorakis [2]

[1] School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore; asdouglas@ntu.edu.sg

[2] Department of Industrial Engineering, Technical University of Sofia, bulevard Sveti Kliment Ohridski 8, Sofia 1000, Bulgaria; mastor@tu-sofia.bg

[*] Correspondence: balasubramanian@ntu.edu.sg; Tel.: +65-6790-4745

[†] An abridged version of this work (Balasubramanian, P. et al., 2018) is published in the proceedings of the 61st IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, Ontario, Canada, 5–8 August 2018.

Received: 21 September 2018; Accepted: 22 October 2018; Published: 24 October 2018

Abstract: In the era of nanoelectronics, multiple faults or failures of function blocks are likely to occur. To withstand these, higher levels of redundancy are suggested to be employed in at least the sensitive portions of a circuit or system. In this context, the N-modular redundancy (NMR) scheme may be used to guard against the multiple faults or failures of function blocks. However, the NMR scheme would exacerbate the weight, cost, and design metrics to implement higher-order redundancy. Hence, as an alternative to the NMR, the majority and minority voted redundancy (MMR) scheme was proposed recently. However, the proposal was restricted to the basic implementation with no provision for indicating the correct or the incorrect operation of the MMR. Hence in this work, we present the MMR scheme with the error/no-error signaling logic (ESL). Example NMR circuits without and with the ESL (NMRESL), and example MMR circuits without and with the proposed ESL (MMRESL) were implemented to achieve similar degrees of fault tolerance using a 32/28-nm CMOS technology. The results show that, on average, the proposed MMRESL circuits have 18.9% less critical path delay, dissipate 64.8% less power, and require 49.5% less silicon area compared to their counterpart NMRESL circuits.

Keywords: redundancy; fault tolerance; low power; ASIC; digital circuits; standard cells; CMOS

1. Introduction

Nanoelectronic circuits and systems are found to be more prone to multiple faults or failures [1] due to harsh environmental phenomena such as radiation [2–6] and/or aging [7,8]. Hence, when such circuits or systems are deployed in safety-critical applications such as aerospace, defense, nuclear plants, etc., redundancy is incorporated by default to cope with the arbitrary fault(s) or failure(s) of constituent function blocks, which are subject to a pre-defined fault tolerance bound. Redundancy implies the use of identical function block(s) in additional to the original function block while designing a circuit or a system for a safety-critical application, where the function block may be a sub-circuit or a sub-system. Redundancy is important in safety-critical circuits and systems to cope with the arbitrary fault(s) or failure(s) of the constituent function blocks. In this context, the N-modular redundancy (NMR) scheme, which is well known, is widely used [9,10]. However, the drawbacks with the NMR are: (i) in order to increase the redundancy by an order of magnitude, two extra function blocks should be introduced, which would exacerbate the weight, cost, and design metrics; and (ii) the sizes of the majority of voters that were used in the NMR scheme would substantially increase with increases in the level of redundancy.

To mitigate the impact of multiple faults or failures on nanoelectronics circuits and systems, higher levels of redundancy are suggested to be used. Since it will be exorbitant to implement high levels of redundancy for an entire circuit or system (say, based on the NMR), the progressive module redundancy (PMR) approach was suggested [11]. PMR is an architectural suggestion that vouches for the selective implementation of high levels of redundancy for the more vulnerable portions of a circuit or system and the implementation of minimum redundancy for the less vulnerable portions of a circuit or system. However, the implementation of higher-order NMR for the more vulnerable portions of a circuit or system would still be expensive. Hence, as an efficient alternative to NMR, the majority and minority voted redundancy (MMR) scheme was proposed in [12] targeting safety-critical applications. However, just the basic implementation of the MMR scheme was considered in [12] with no provision for indicating the correct or the incorrect operation of the MMR through error/no-error signaling logic (ESL). In this article, we build upon our previous work [12] by presenting an ESL for the MMR scheme.

In [13], an ESL for the NMR scheme was presented. The ESL is important for any redundancy scheme, because if the ESL signals no error, then the outputs of the redundancy scheme are reliable, i.e., dependable, and if the ESL signals error, then the outputs of the redundancy scheme are not reliable i.e., non-dependable. Hence, without the ESL, the correct operation of a redundancy scheme is only assumed, which may be incorrect and may even cause a catastrophic failure. Hence, the ESL avoids assuming the correct operation of a redundancy scheme and thereby contributes to the safety of a circuit or system. However, there are bounds associated with the operation of the ESL, which will be discussed later.

The rest of the article is organized as follows. Section 2 discusses the NMR scheme and briefs the operation of the NMR circuits without and with the ESL (NMRESL). Section 3 describes the example MMR circuits without and with the proposed ESL, i.e., the MMRESL. Example NMR and NMRESL circuits, and their counterpart MMR and MMRESL circuits, were considered for physical implementation, and their design metrics are given in Section 4 and compared. Finally, Section 5 provides the conclusions.

2. NMR Scheme and NMRESL

2.1. NMR Scheme

In the NMR scheme, as portrayed in Figure 1, N identical function blocks, where N is odd, are used, and the correct operation of at least (N + 1)/2 function blocks is required. The maximum fault tolerance of the NMR scheme is (N − 1)/2. The outputs of the N identical function blocks viz. B_1 to B_N are given to a voter, which performs the majority voting and produces the NMR output (NMRO).

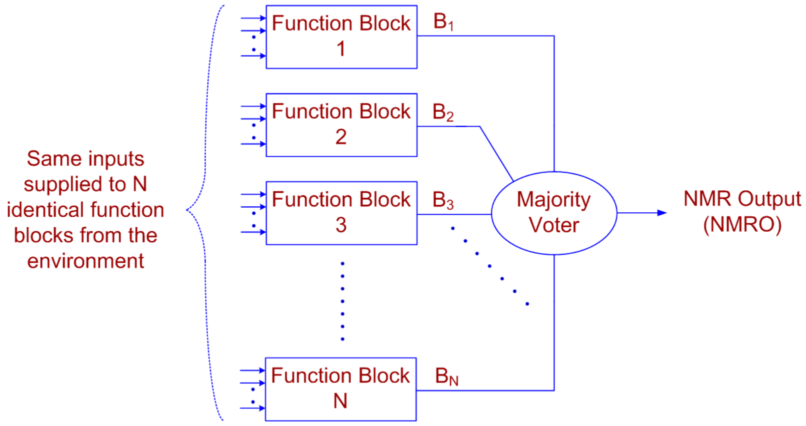

Figure 1. Block schematic of the N-modular redundancy (NMR) scheme.

The 3MR represents the basic i.e., the minimum version of the NMR that uses three identical function blocks and can mask the fault or failure of a maximum of one function block. The 5MR, 7MR, and 9MR versions of the NMR use five, seven, and nine function blocks, respectively, and can mask the faults or failures of a maximum of two, three, and four function blocks. Hence, two function blocks should be added to the NMR scheme to increase its fault tolerance by an order of magnitude.

Figures 2–4 show the 5MR, 7MR, and 9MR majority voters designed using the multiplexer (MUX) logic, as suggested in [14]. B_1 up to B_9 represent the outputs of the identical function blocks, which serve as the inputs for the NMR majority voters, and 5MRO, 7MRO, and 9MRO represent the outputs of the 5MR, 7MR, and 9MR implementations. In Figure 3, the complex gate OA221 can be replaced by the complex gate OA211, but because the OA211 gate is not available in the standard digital cell library [15], the OA221 gate has been used instead. For implementation using [15], the MUX-based 5MR, 7MR, and 9MR majority voters respectively consume 13.47 μm^2, 34.31 μm^2, and 63.79 μm^2 of silicon. The almost doubling of the areas of the majority voters when progressing from one level of redundancy to the next is due to the increase in the number of dominant majority conditions, which is governed by the mathematical combination: O [$NC_{(N+1)/2}$].

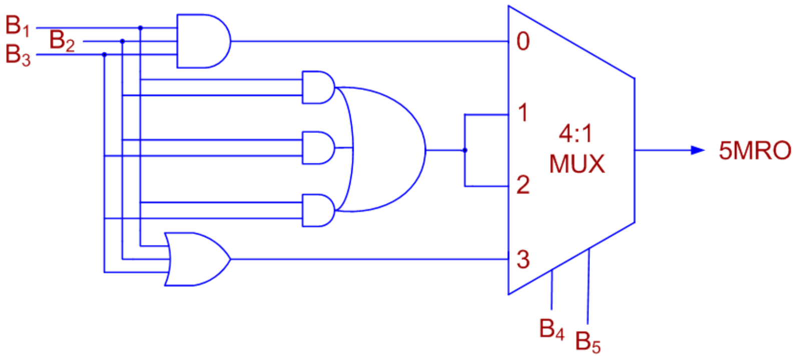

Figure 2. Multiplexer (MUX)-based 5MR majority voter.

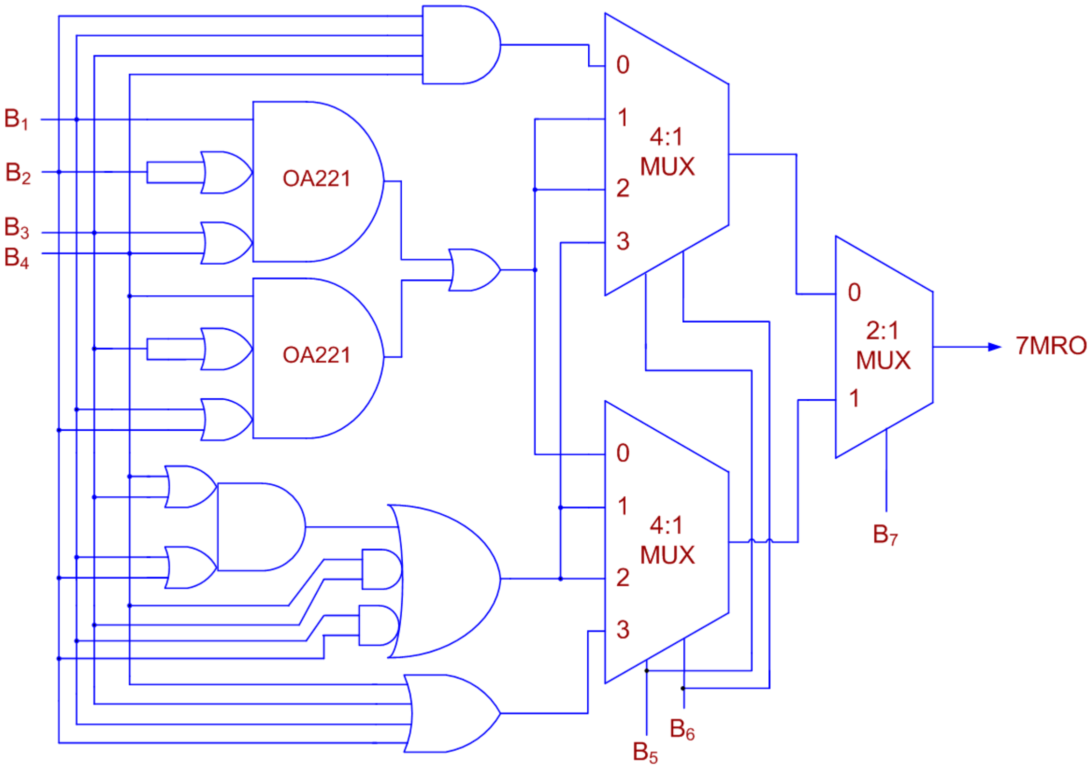

Figure 3. MUX-based 7MR majority voter.

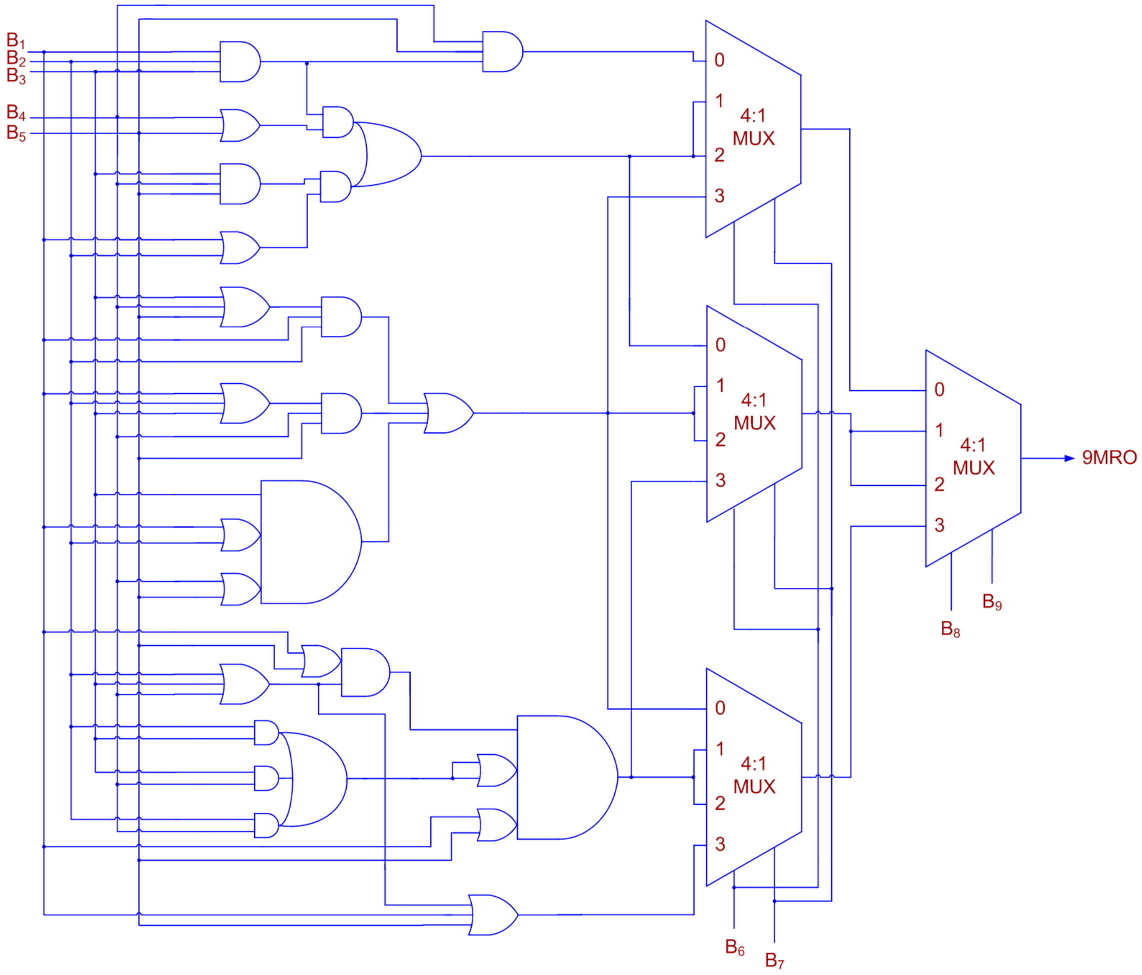

Figure 4. MUX-based 9MR majority voter.

To explain what a dominant majority condition is and the difference between a normal majority condition and a dominant majority condition, let us consider the 5MR implementation for an example. Considering that B_1, B_2, B_3, B_4, and B_5 are the outputs of the five identical function blocks, which are supplied as the inputs to the 5MR majority voter that is shown in Figure 2, its output is expressed by Equation (1). The total number of majority conditions (including the dominant majority conditions) underlying a NMR implementation is generically governed by O $[2^{N-1}]$. Of the 16 majority conditions listed in Equation (1), the first 10 majority conditions are said to be dominant, as they are irredundant for the physical realization of the 5MR majority voter, while the remainder of the majority conditions can be eliminated by applying the absorption axiom of Boolean algebra; for example, according to the absorption law, X + XY = X. Nevertheless, while estimating the reliability of a NMR implementation, all of the majority conditions should be considered:

$$\begin{aligned}5\mathrm{MRO} = {} & \mathrm{B_1B_2B_3} + \mathrm{B_1B_2B_4} + \mathrm{B_1B_2B_5} + \mathrm{B_1B_3B_4} + \mathrm{B_1B_3B_5} + \mathrm{B_1B_4B_5} + \mathrm{B_2B_3B_4} + \mathrm{B_2B_3B_5} + \mathrm{B_2B_4B_5} + \\ & B_3B_4B_5 + B_1B_2B_3B_4 + B_1B_2B_3B_5 + B_1B_2B_4B_5 + B_1B_3B_4B_5 + B_2B_3B_4B_5 + B_1B_2B_3B_4B_5\end{aligned} \quad (1)$$

Let the reliability of a function block, which signifies its correct operation, be expressed as R_F, which is inherently a function of time *t*. Also, since identical function blocks are used, the reliabilities of the function blocks are considered equal. Given this, the reliabilities of the 5MR, 7MR, and 9MR implementations are given by Equations (2)–(4) respectively. Since the majority voter is generally small compared to the function block, the perfect behavior of the majority voters is assumed in Equations

(2)–(4) for simplicity, i.e., the reliability of the voter is equated to 1. Also, the fault(s) or failure(s) of the function block(s) are assumed to be statistically independent.

$$R_{5MR} = 10R_F^{\ 3}\ (1 - R_F)^2 + 5R_F^{\ 4}\ (1 - R_F) + R_F^{\ 5} \tag{2}$$

$$R_{7MR} = 35R_F^{\ 4}\ (1 - R_F)^3 + 21R_F^{\ 5}\ (1 - R_F)^2 + 7R_F^{\ 6}\ (1 - R_F) + R_F^{\ 7} \tag{3}$$

$$R_{9MR} = 126R_F^{\ 5}\ (1 - R_F)^4 + 84R_F^{\ 6}\ (1 - R_F)^3 + 36R_F^{\ 7}\ (1 - R_F)^2 + 9R_F^{\ 8}\ (1 - R_F) + R_F^{\ 9} \tag{4}$$

The terms present on the right side of Equations (2)–(4) result from the mathematical combinations corresponding to the correct operation of majority of the function blocks and the incorrect operation of the remaining function blocks, i.e., $R_F^{\ K}$ implies that a majority K out of the N function blocks are operating correctly, and $(1 - R_F)^{N-K}$ implies that the remaining (N − K) function blocks are faulty or have failed. For example, the first term on the right side of Equation (2) specifies the condition of any three out of the five function blocks maintaining the correct operation and the faulty state or the failure of the remaining two function blocks. The second term specifies the condition of any four out of the five function blocks operating correctly, and the fault or the failure of the remaining function block. The third term specifies the (ideal) condition of all of the function blocks maintaining the correct operation.

2.2. Example NMRESL and Its Operation

A system health monitor for the NMR scheme was presented in [13], which consists of the fault warning logic (FWL) and the ESL. The FWL would issue a warning signal (binary 1) whenever any output of any function block is contrary to the corresponding output(s) of any of the remaining function block(s). As such, a fault warning or no-fault warning issued by the FWL would not be able to provide clear information about the correct or the incorrect operation of a NMR implementation, but the ESL can confirm the correct or the incorrect operation. Hence, in this work, we discard the FWL and consider only the ESL for a generic NMR implementation. The design of the ESL for the NMR scheme is complex and sophisticated, because it is dependent on the order of the NMR [13], and an interested reader is suggested to refer to [13] for the details. However, for a quick reference and to make this article self-contained, the design of the ESL for a 5MR implementation is discussed below.

The 5MR scheme along with the ESL is shown in Figure 5. There are five function blocks, and let each function block consist of two outputs. (B1, C1), (B2, C2), (B3, C3), (B4, C4), and (B5, C5) represent the corresponding dual outputs of the function blocks 1, 2, 3, 4, and 5, respectively. The portion of the circuit highlighted in blue lines depicts the typical 5MR implementation consisting of the function blocks and the five-input majority voters, which produce the primary outputs 5MRO1 and 5MRO2. The sub-circuit highlighted in red lines depicts the 5MRESL, and 5MRESLO denotes the output of the ESL. To briefly mention the components of the 5MRESL shown in Figure 5, for example, B(1,2) refers to the output of a two-input inclusive OR (XNOR) gate that has B_1 and B_2 as inputs. An XNOR gate basically checks for the logical equivalence of its inputs. If the inputs to an XNOR gate are logically equivalent, it would output 1; otherwise, it would output 0. (1,2) refers to the output of a two-input AND gate whose output determines whether the corresponding outputs of the function blocks 1 and 2 are equivalent or not. If (1,2) = 1, it implies that $B_1 = B_2$ and $C_1 = C_2$, confirming that the function blocks 1 and 2 produce the same outputs. On the contrary, if (1,2) = 0, it implies that $B_1 \neq B_2$ and/or $C_1 \neq C_2$, confirming that the function blocks 1 and 2 do not produce the same outputs, thus indicating that either of these function blocks has become faulty or failed. (1,2,3) represents the output of the next-level two-input AND gate, which receives as inputs (1,2) and (2,3). If (1,2) and (2,3) are 1, then (1,2,3) = 1, implying that the function blocks 1, 2, and 3 produce the same outputs. Supposing if (1,2,3) = 0, it signifies that one or more of the function blocks 1, 2, and 3 are faulty or have failed.

To briefly explain the operation of the 5MRESL implementation, let us consider two example scenarios with respect to Figure 5. Firstly, let us assume that three out of the five function blocks in Figure 5 operate correctly (say, function blocks 1, 2, and 3 operate correctly) and produce the correct output, and that function blocks 4 and 5 have become faulty or failed. Regardless of whether the correct outputs of the function blocks (1, 2, and 3) are binary 1 or 0, the outputs of the two-input XNOR gates labeled B(1,2), B(1,3), and B(2,3) would be 1. Similarly, the outputs of the two-input XNOR gates labeled C(1,2), C(1,3), and C(2,3) would be 1. Therefore, (1,2) = (1,3) = (2,3) = 1, and hence (1,2,3) = 1, which is given as an input to the four-input OR gate labeled G1. Subsequently, G1 would output 1, since one of its inputs is 1, and because the four-input NOR gate (G3) receives 1 as one of its inputs, it would output 0 on 5MRESLO, thus signaling no-error.

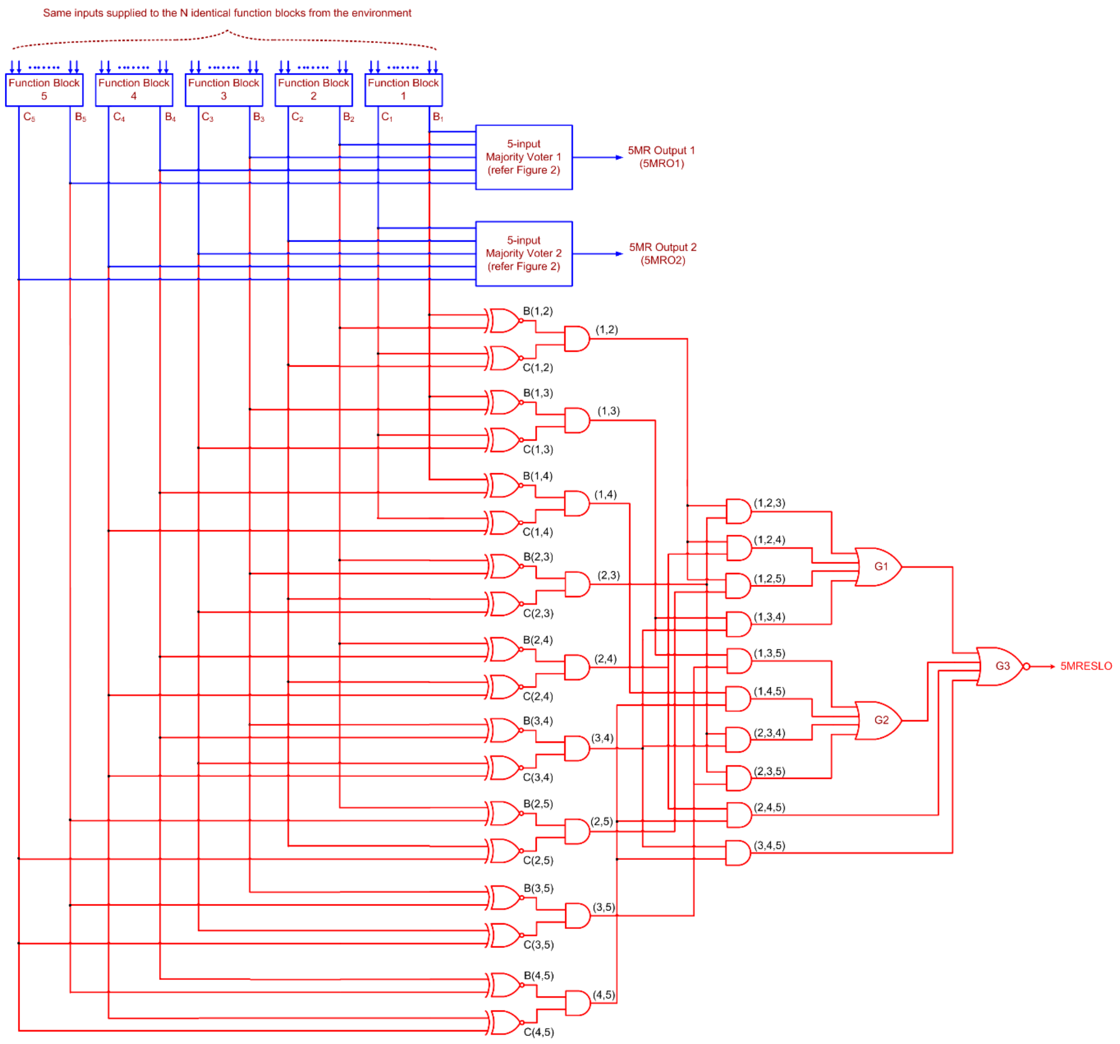

Figure 5. 5MR implementation with error/no-error signaling logic (ESL).

Secondly, let us assume that only function blocks 1 and 2 operate correctly, and that the function blocks 3, 4, and 5 have become faulty or failed. Further, let us assume that the function blocks 3, 4, and 5 do not experience common-mode faults i.e., they do not agree to produce the same incorrect outputs. However, for this assumption, 'no-error' would be erratically signaled by the ESL, since the ESL will consider that the function blocks 3, 4, and 5 are maintaining the correct operation, which is not true. This is a limitation of the 5MRESL circuit, and this limitation is inherent in even the basic NMR circuit, as remarked in [16]. In general, in any NMR implementation, if (N + 1)/2 function blocks or more would agree to produce the same incorrect outputs due to any common-mode faults affecting them, then the output of the NMR implementation would be contrary to the factual, and this condition will not be signaled as an incorrect operational state by the NMRESL [13]. On the contrary, if only a minority of the faulty or failed function blocks may agree to produce the same error outputs due to the common-mode faults affecting them, this will not affect the operation of the NMRESL.

As per the second assumption that the function blocks 1 and 2 alone operate correctly in Figure 5, $B_1 = B_2$ and $C_1 = C_2$. Also, let us randomly assume that $B_3 \neq B_4$ but $B_4 = B_5$, and $C_3 \neq C_4$ but $C_3 = C_5$. Given this scenario, $B_1 = B_2 = B_3$ may be a possibility, and $C_1 = C_2 = C_4$ may also be a possibility. This is because a Boolean variable can assume either binary 0 or 1. As a result, B(1,2) = B(1,3) = B(2,3) = B(4,5) = C(1,2) = C(1,4) = C(2,4) = C(3,5) = 1, and B(1,4) = B(1,5) = B(2,4) = B(2,5) = B(3,4) = B(3,5) = C(1,3) = C(1,5) = C(2,3) = C(2,5) = C(3,4) = C(4,5) = 0. Therefore, (1,2) = 1, but (1,3) = (1,4) = (2,3) = (2,4) = (3,4) = (2,5) = (3,5) = (4,5) = 0. Eventually, this results in (1,2,3) = (1,2,4) = (1,2,5) = (1,3,4) = (1,3,5) = (1,4,5) = (2,3,4) = (2,3,5) = (2,4,5) = (3,4,5) = 0, meaning that all of the inputs to the four-input OR gates G1 and G2 are 0, and hence all of the inputs to the four-input NOR gate G3 is 0, and so the output of the 5MRESL circuit viz. 5MRELSO = 1, implying the 5MR implementation is in error, and its outputs are not dependable.

3. MMR Scheme and MMRESL

The basic MMR scheme was proposed by us in an earlier paper [12], without the ESL. The generic architecture of the MMR scheme, including the ESL, is shown in Figure 6. The blue lines depict the basic MMR architecture and the red lines depict the ESL of the MMR (MMRESL).

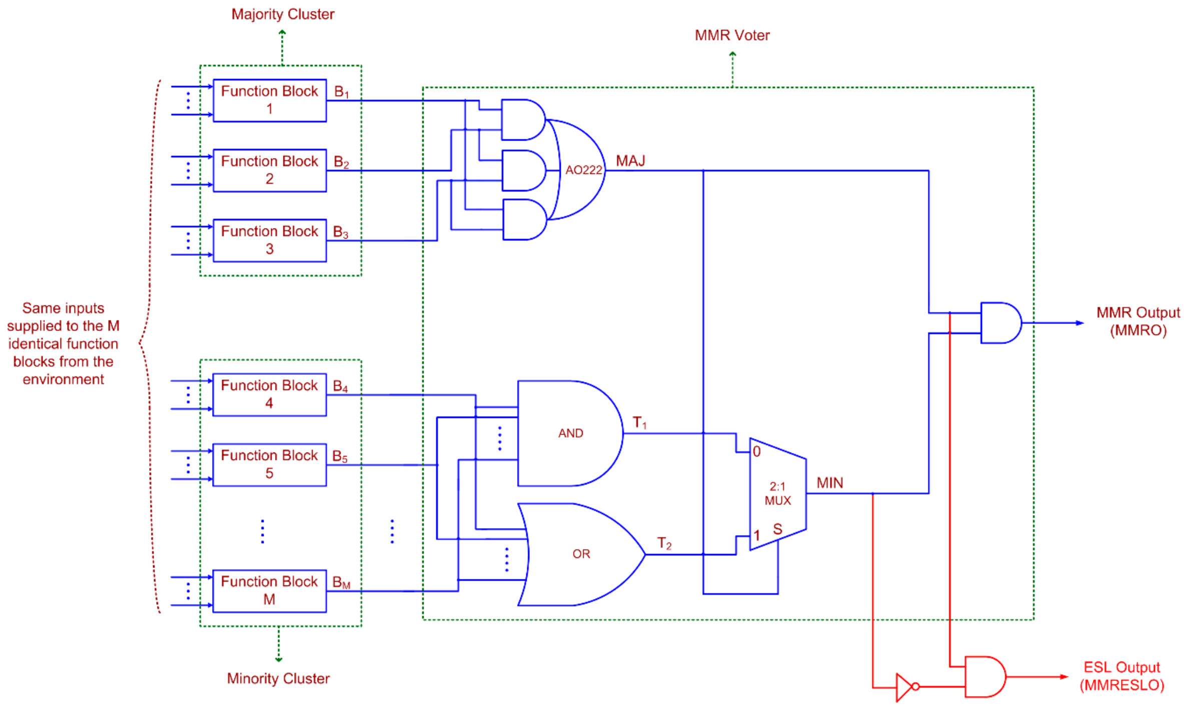

Figure 6. Majority and minority voted redundancy (MMR) scheme with the proposed ESL.

In the MMR scheme, (M − 1) copies of the original function block are used, and the M identical function blocks are split into two clusters, namely the 'majority cluster' and the 'minority cluster', as shown in Figure 6. Three function blocks comprise the majority cluster, and the remaining (M − 3) function blocks comprise the minority cluster. The Boolean majority condition is imposed on the function blocks constituting the majority cluster, which implies that at least two out of the three function blocks 1, 2, and 3 should maintain the correct operation. The relaxed Boolean minority condition is imposed on the function blocks constituting the minority cluster, and thus it would suffice even if any one of the function blocks in the minority cluster operates correctly. Overall, at least three out of the M function blocks should maintain the correct operation in the MMR scheme, and hence the fault tolerance of the MMR scheme is specified as (M − 3).

The MMR voter is marked in Figure 6. For every output of the function block, the MMR voter would consist of an AO222 complex gate, a (M − 3)-input AND gate, a (M − 3)-input OR gate, and a 2:1 multiplexer (i.e., 2:1 MUX). The outputs of the function blocks 1, 2, and 3 are given to the AO222 gate [17], which performs majority voting on the three inputs B_1, B_2, and B_3, and produces the internal output MAJ. The outputs of the remainder of the function blocks 4 to M are given to an AND gate and an OR gate, which have the same fan-in of (M − 3). T_1 represents the output of the (M − 3)-input AND gate, and T_2 represents the output of the (M − 3)-input OR gate. T_1 and T_2 are given as inputs to the 2:1 MUX, whose select input is MAJ. Hence, if MAJ = 0, T_1 is selected, and its value is forwarded to the output of the 2:1 MUX, which is labeled MIN. If MAJ = 1, then T_2 is selected, and MIN = T_2. The logical conjunction of MAJ and MIN yields the primary output of the MMR implementation viz. MMRO. The ESL of the MMR scheme consists of an inverter that complements MIN. The ESL also consists of a two-input AND gate, and the logical conjunction of MAJ and the complement of MIN yields the MMRESL output i.e., MMRESLO. If function blocks with multiple outputs are used in an MMR implementation, then the ESL will contain as many two-input AND gates and inverters as are commensurate with the number of outputs from the function blocks. The outputs of all of the ESL circuitry can be combined using an OR gate, which may be decomposed arbitrarily, to produce the ESL output of the MMR implementation.

We will use the notation K-of-M while referring to the MMR scheme for our discussion, which signifies that K out of the M function blocks in a MMR implementation operate correctly. Hence, a three-of-five MMR implementation can mask the faults or failures of a maximum of two function blocks similar to the 5MR implementation; a three-of-six MMR implementation can mask the faults or failures of maximum of three function blocks similar to the 7MR implementation; and a three-of-seven MMR implementation can mask the faults or failures of maximum of four function blocks similar to the 9MR implementation. The three-of-six and three-of-seven MMR implementations provide the same degrees of fault tolerance as the 7MR and 9MR implementations despite requiring one and two function blocks less than their counterparts. This could help to reduce the cost, weight, and design metrics of the former compared to the latter.

The reliabilities of the three-of-five, three-of-six, and three-of-seven MMR implementations are given by Equations (5)–(7) based on the assumption of perfect MMR voters. Let us interpret the reliability components of the three-of-five MMR implementation for an example. In Equation (5), the first term on the right side specifies the condition of any two function blocks in the majority cluster and any one function block in the minority cluster operating correctly. The second term specifies the condition of either of any two function blocks in the majority cluster and both the function blocks in the minority cluster operating correctly, or the correct operation of all three function blocks in the majority cluster and just one function block in the minority cluster. The third term on the right side specifies the (ideal) condition of all five function blocks in the three-of-five MMR implementation maintaining the correct operation:

$$R_{\text{3-of-5 MMR}} = 6R_F^{\,3}\,(1 - R_F)^2 + 5R_F^{\,4}\,(1 - R_F) + R_F^{\,5} \tag{5}$$

$$R_{\text{3-of-6 MMR}} = 9R_F^{\,3}\,(1 - R_F)^3 + 12R_F^{\,4}\,(1 - R_F)^2 + 6R_F^{\,5}\,(1 - R_F) + R_F^{\,6} \tag{6}$$

$$R_{\text{3-of-7 MMR}} = 12R_F^{\,3}\,(1 - R_F)^4 + 22R_F^{\,4}\,(1 - R_F)^3 + 18R_F^{\,5}\,(1 - R_F)^2 + 7R_F^{\,6}\,(1 - R_F) + R_F^{\,7} \tag{7}$$

The reliabilities of the NMR and counterpart MMR implementations are plotted in Figure 7 as a function of the reliability of the constituent function blocks, and they exhibit a close correlation. Considering the reliability of a function block to be in the range of 0.9 to 0.99, which is quite common for a safety-critical application, the MMR implementations were found to have 1.12% less reliability than the NMR implementations, on average. This is the trade-off that is involved in achieving reductions in the number of function blocks, design metrics, weight, and cost.

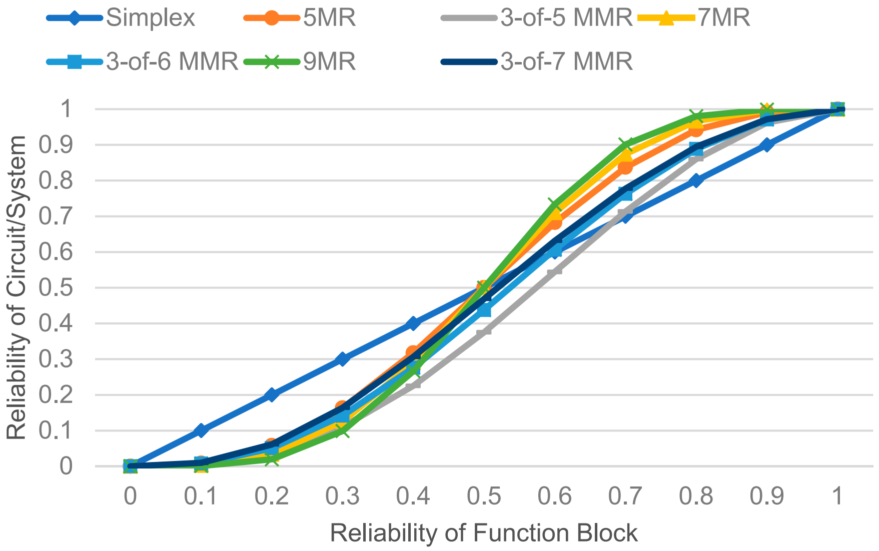

Figure 7. Comparison of reliabilities of NMR and MMR implementations, assuming the perfect behavior of the voters. The reliability of a simplex circuit/system, with no redundancy, is equal to R_F.

A higher priority is inherently accorded to the majority cluster compared to the minority cluster in the MMR scheme. This is because the Boolean majority condition is unambiguous, while the Boolean minority condition may be ambiguous. To understand why this is so, let us presume that the function blocks 1, 2, and 4 in Figure 6 produce the correct output, and that function block 3 and function blocks 5 to M are faulty or have failed. Given this, since two out of the three function blocks produce the same correct output in the majority cluster, the Boolean majority condition will unambiguously determine the output of the majority cluster as MAJ = B_1 = B_2. On the other hand, given that only function block 4 produces the correct output, this cannot be unambiguously interpreted as the output of the minority cluster. This is because it can be argued that the outputs of the function block 5 to M also correspond to the Boolean minority, since the Boolean minority condition primarily specifies at least one correct output. Hence, there arises an ambiguity in determining the correct output of the minority cluster based on the Boolean minority condition. For example, if B_4 = 0, and B_5 up to B_M assumes 1, both 0 and 1 can correspond to the Boolean minority, since B_4 is 0 and at least one of B_5 up to B_M is 1. For this input combination, T_1 = 0 and T_2 = 1. So, the choice of T_1 or T_2 as the correct output of the minority cluster should have to be decided, and a decision should be taken based on the value of MAJ, which is the output of the majority cluster. This explains why *the correct operation of the majority cluster is crucial in an MMR implementation and cannot be compromised (to overcome the ambiguity with the Boolean minority condition), while the correct operation of the minority cluster may not always be crucial*. In fact, a complete failure of the minority cluster can be successfully masked under certain circumstances, and this will be explained through Table 1.

Under the minority cluster column in Table 1, 'B_4–B_M' represented by '0–0' implies that B_4 up to B_M assume 0; 'B_4–B_M' represented by '0–1' implies that B_4 assumes 0, and B_5 up to B_M may assume 1; and 'B_4–B_M' represented by '1–0' implies that B_4 assumes 1, and B_5 up to B_M may assume 0. The possible operational scenarios for the MMR scheme are captured in Table 1.

Scenario 1 indicates the ideal condition of both the majority and minority clusters operating perfectly i.e., the function blocks in both the clusters maintain the correct operation. Obviously, in this scenario, the state of the MMR output (i.e., MMRO) would be correct. Scenario 2 highlights the condition where the majority cluster is imperfect due to a faulty function block and outputs 0 due to any two out of the three function blocks outputting 0, and the minority cluster is imperfect. However, at least one of the function blocks in the minority cluster maintains the correct operation and outputs 0. In this scenario, MAJ = 0, and T_1 is selected, which implies that MIN equates to 0. Hence, MMRO = 0, which is correct. Scenario 3 is similar to Scenario 2, except that MMRO = 1 because MAJ = MIN

= 1, since two of the function blocks in the majority cluster output 1, and at least one of the function blocks in the minority cluster also outputs 1. With respect to scenarios 1, 2, and 3, the MMRESL output (MMRESLO) is 0, thus implying no-error.

Table 1. Illustrating the operation of the MMR and the MMRESL.

Majority Cluster			Minority Cluster			MMR Voter Internal Outputs		MMR Output	MMR Output State (Correct/Error)	MMRESL Output (MMRESLO) (0—Correct; 1—Error)
B_1	B_2	B_3	B_4	–	B_M	MAJ	MIN	MMRO		
Scenario 1: Majority and Minority Clusters are perfect										
0	0	0	0	–	0	0	0	0	Correct	0
1	1	1	1	–	1	1	1	1	Correct	0
Scenario 2: Majority and Minority Clusters are not perfect, and Majority Cluster outputs 0										
0	0	1	0	–	1	0	0	0	Correct	0
0	1	0	0	–	1	0	0	0	Correct	0
1	0	0	0	–	1	0	0	0	Correct	0
Scenario 3: Majority and Minority Clusters are not perfect, and Majority Cluster outputs 1										
1	1	0	1	–	0	1	1	1	Correct	0
1	0	1	1	–	0	1	1	1	Correct	0
0	1	1	1	–	0	1	1	1	Correct	0
Scenario 4: Majority Cluster is not perfect and outputs 0, *and Minority Cluster completely fails*										
0	0	1	1	–	1	0	1	0	Correct	0
0	1	0	1	–	1	0	1	0	Correct	0
1	0	0	1	–	1	0	1	0	Correct	0
Scenario 5: Majority Cluster is not perfect and outputs 1, *and Minority Cluster completely fails*										
1	1	0	0	–	0	1	0	0	Error	1
1	0	1	0	–	0	1	0	0	Error	1
0	1	1	0	—	0	1	0	0	Error	1

Scenarios 4 and 5 depict the conditions where the majority cluster is imperfect, and the minority cluster fails completely. Although the MMR implementation is not warranted to operate correctly under scenarios 4 and 5, Scenario 4 showcases the innate error resiliency of the MMR scheme, which is captured by the proposed ESL, and Scenario 5 showcases the importance and the need for the ESL. With respect to Scenario 4, if the majority cluster is not perfect and outputs 0 due to any two of the constituent function blocks outputting 0 and given that the minority cluster has completely failed (i.e., all of its constituent function blocks output 1), MAJ = 0 and MIN = 1, and hence MMRO = 0, which is factually correct, since the output of the MMR scheme is primarily dictated by the output of the majority cluster. The correct state of the MMR output under Scenario 4 is confirmed by the MMRESL, where MMRESLO = 0, thus implying no-error. This shows the MMR scheme maintains the correct operation even under an undesirable and unwarranted Scenario 4. Supposing Scenario 5 occurs, where the majority cluster is not perfect and outputs 1 due to two of its function blocks outputting 1 and that the minority cluster has completely failed (i.e., all of its function blocks output 0), MAJ = 1 and MIN = 0. This implies that MMRO = 0, which is incorrect, since the output of the MMR scheme does not tally with the output of the majority cluster i.e., MMRO $\neq$ MAJ. Under this scenario, the proposed MMRESL would output 1 on MMRESLO, implying the error in the operation of the MMR scheme. Considering all five scenarios which were discussed, it may be evident that the proposed MMRESL provides useful information about the correct or the incorrect operational state of a MMR implementation while encompassing the error resiliency of the MMR scheme.

Figure 8 shows an example three-of-five MMR implementation along with the ESL. Comparing this with the 5MR implementation featuring the ESL that is shown in Figure 5, it may be noted that the former requires a considerably smaller number of gates than the latter while featuring the same fault tolerance, which is expected to translate into reductions in the design metrics for a physical implementation.

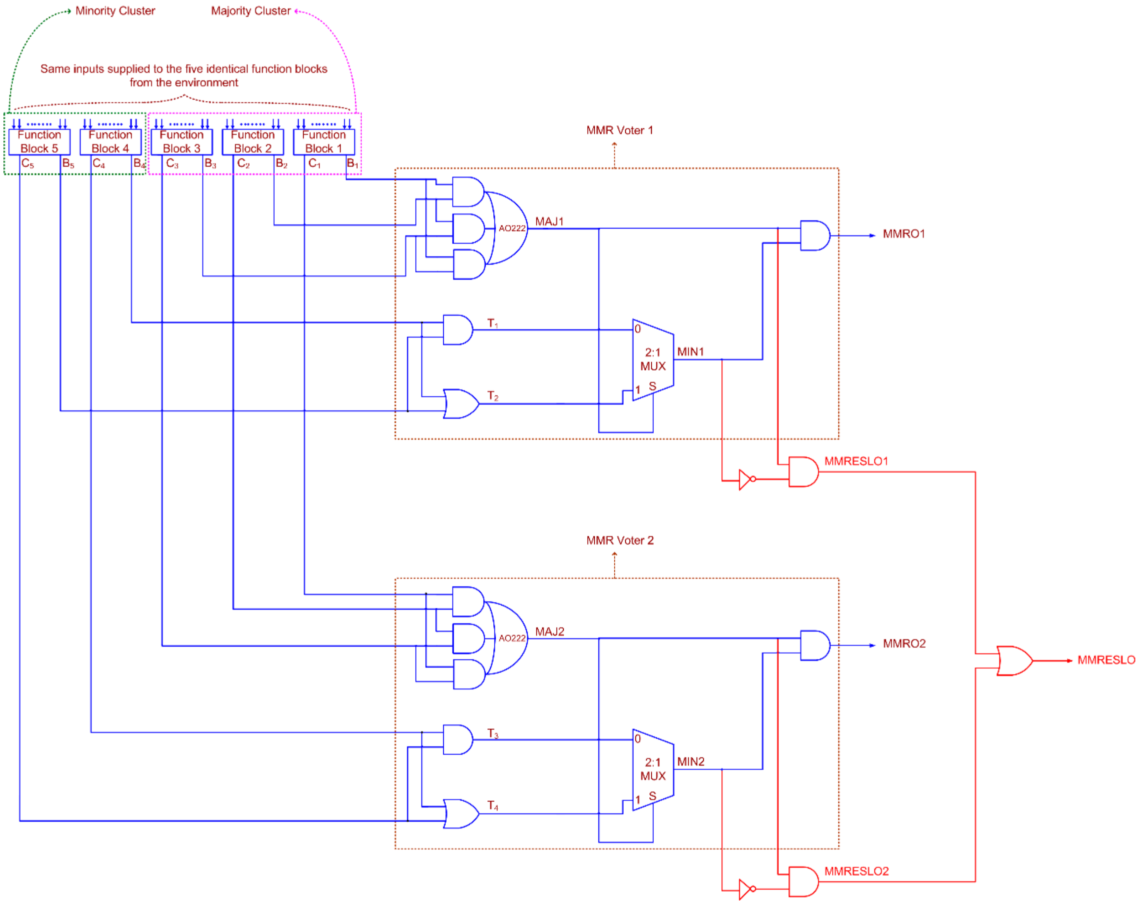

Figure 8. Example three-of-five MMR implementation with the proposed ESL.

4. Results and Discussion

5MR, 7MR, and 9MR circuits, and three-of-five MMR, three-of-six MMR, and three-of-seven MMR circuits with and without the ESL were physically implemented using a 32/28 nm CMOS standard digital cell library [15]. A 4 × 4 array multiplier was considered as the function block, which has eight input bits and produces eight output bits. The array multiplier requires 16 two-input AND gates, four half adders, and eight full adders for physical realization. The AND gate, half-adder, and full-adder cells from the library [15] were utilized to construct the array multiplier, which consumes 84.38 μm^2 of silicon. Functional simulations were performed to verify the functionalities of the redundant circuits using test benches, which included all of the distinct input vectors corresponding to the multiplier. The test benches were supplied at time intervals of 2.5 ns (400 MHz). The switching activity data captured through the functional simulations were used to estimate the average power dissipation using Synopsys tools. Default wire loads were included while performing the simulations, and the areas and the critical path delays were also estimated. The design metrics corresponding to the example NMR and MMR circuits without and with the ESL are given in Table 2.

The power-delay product (PDP) is a well-known and widely used low power metric for digital circuits and systems. Hence, the PDP of the redundant circuits were calculated and normalized. To perform normalization, the highest PDP value of a redundant circuit corresponding to a specific degree of fault tolerance was chosen as the reference, and this reference value was used to divide the actual PDP values of all of the redundant circuits without and with the ESL, which correspond to the same degree of fault tolerance. The normalized PDP values are given in Table 1. Although the least value of PDP is desirable, the PDP is traded-off for the provision of the ESL here. The provision of the ESL is important, as it infuses a confidence into interpreting the correct or the incorrect operation of a redundancy scheme, and the absence of the ESL would lead to presuming the correct operation of a redundancy scheme, which may not always be true.

Table 2. Design parameters of NMR and counterpart MMR circuits without and with the ESL, estimated using a 32/28nm CMOS process. PDP: power-delay product.

Type of Redundancy	Critical Path Delay (ns)	Area (μm^2)	Power Dissipation (μW)	Normalized PDP
		Maximum fault tolerance of two function blocks		
5MR	0.98	529.64	120.7	0.543
5MRESL	1.31	935.25	166.2	1
3-of-5 MMR	1.01	523.54	116.4	0.54
3-of-5 MMRESL	1.20	559.12	126.2	0.696
		Maximum fault tolerance of three function blocks		
7MR	1.12	865.11	191.2	0.535
7MRESL	1.44	1685.48	277.9	1
3-of-6 MMR	1.01	611.98	137.0	0.346
3-of-6 MMRESL	1.20	647.56	146.8	0.44
		Maximum fault tolerance of four function blocks		
9MR	1.23	1269.70	278.5	0.469
9MRESL	1.69	2917.08	431.9	1
3-of-7 MMR	1.01	708.55	159.3	0.22
3-of-7 MMRESL	1.20	744.13	169.0	0.278

The critical path delays of the NMR circuits are given by the sum of the propagation delays of a function block and the corresponding majority voters. Since the majority voters of the NMR circuits would differ in structure due to increases in the logic gates and the logic levels with increases in the order of redundancy (as portrayed by Figures 2–4), the critical path delays of the NMR circuits would increase with increases in the order of redundancy, as noticed in Table 2. The critical path delays of the NMRESL circuits are given by the sum of the propagation delays of a function block, the corresponding majority voters, and the corresponding ESL circuits. The ESL portion of the NMRESL circuits would considerably increase with increases in the order of redundancy. As a result, the critical path delays of the NMRESL circuits are also expected to increase with increases in the order of redundancy, as seen in Table 2. In the case of the MMR circuits, their critical path delays are dependent upon the propagation delay of a function block and the propagation delay of the corresponding MMR voter. The propagation delay of a MMR voter is dependent on the propagation delays of an AO222 gate, a 2:1 MUX, and a final two-input AND gate. Given this, the critical path delays of the MMR circuits would be the same, thanks to the regularity implicit in the MMR architecture. In the case of the MMRESL circuits, their critical path delays comprise the propagation delays of a function block, the corresponding MMR voter, and the corresponding ESL portion. The ESL part of the MMR circuits feature a uniform logic realization comprising an inverter and a two-input AND gate with respect to each primary output of the function block. The internal outputs of the MMRESL (for example, MMRESLO1 and MMRESLO2, as shown in Figure 8) can be combined using an OR gate or an OR gate tree, depending upon the number of primary outputs produced by the function blocks. The ESL portion of the MMRESL circuits would be the same, regardless of the order of redundancy, and hence the critical path delays of the MMRESL circuits will be the same, as noticed in Table 2.

The critical path delays of the NMRESL and MMRESL circuits will be greater than the critical path delays of the basic NMR and MMR circuits due to the presence of the ESL in the former, which are absent in the latter. From Table 2, it is found that the averaged critical path delay of the 5MR, 7MR, and 9MR circuits is less than the averaged critical path delay of the 5MRESL, 7MRESL, and 9MRESL circuits by 25%, and the averaged critical path delay of the three-of-five, three-of-six, and three-of-seven MMR circuits is less than the averaged critical path delay of the three-of-five, three-of-six, and three-of-seven MMESL circuits by 15.8%. Also, the averaged critical path delay of the three-of-five, three-of-six, and three-of-seven MMRESL circuits is less than the averaged critical path delay of the 5MRESL, 7MRESL, and 9MRESL circuits by 18.9%.

From Table 2, it is seen that the areas of the NMR circuits are larger than the areas of the MMR circuits. This is due to two reasons: (i) the 7MR and 9MR circuits require 1 and 2 function blocks more than the three-of-six and three-of-seven MMR circuits, respectively; and (ii) the areas of the NMR majority voters are larger than the areas of the counterpart MMR voters. The normalized areas of the various NMR and counterpart MMR voters are depicted in Figure 9a. The area of the 9MR majority voter is the maximum among the various voters, and this was considered as the baseline value to divide the actual areas of all of the NMR and MMR voters to perform normalization. On average, the MMR voters require a 63.5% smaller silicon footprint compared to their counterpart NMR voters. Further, the areas of the ESL of the MMR circuits represent a very small percentage compared to the area occupancies of the ESL part of the counterpart NMR circuits. Figure 9b shows the normalized area occupancies of the NMRESL circuits and the corresponding MMRESL circuits, given in percentages. The ESL portion of the 9MRESL circuit is found to occupy the maximum area, and so this value was used to perform the normalization. On average, the ESL part of the MMRESL circuits requires 26× less area than the ESL part of their counterpart NMRESL circuits. From Table 2, it is found that on average, the MMR circuits occupy 30.8% less area than the corresponding NMR circuits, and the MMRESL circuits occupy 64.8% less area than the corresponding NMRESL circuits. The proposed MMRESL circuits require 26.8% less silicon than even the corresponding NMR circuits without ESL, which is a notable advantage.

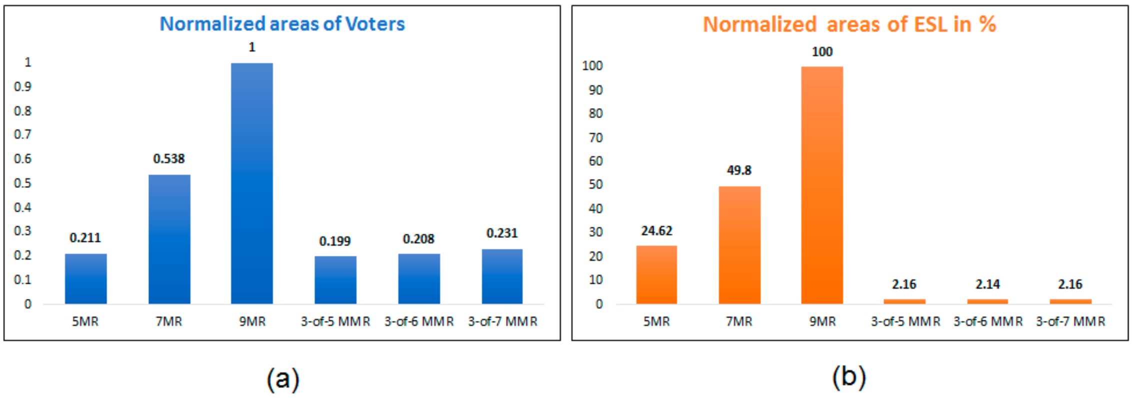

Figure 9. Normalized areas of: (**a**) voters of NMR and counterpart MMR implementations; (**b**) ESL portion of NMR (NMRESL) and counterpart MMRESL implementations.

Since the averaged area of the NMR and NMRESL circuits is greater than the averaged area of the MMRESL circuits, the latter are likely to dissipate less power than the former. From Table 2, it is found that on average, the MMRESL circuits dissipate 25.1% less power compared to the NMR circuits, and 49.5% less power than the NMRESL circuits. Further, it is noted that the proposed MMRESL circuits, on average, achieve an 8.7% reduction in the PDP compared to the basic NMR circuits and a 52.9% reduction in the PDP compared to the NMRESL circuits.

5. Conclusions

This article presented a new ESL circuit for the recently proposed MMR scheme, which forms an attractive alternative to the NMR scheme for the efficient design of circuits and systems that are meant for safety-critical applications. The provision of the ESL is important to be able to make an informed judgment about the correct or the incorrect operation of a redundant implementation. However, for the ESL, the correct operation of a redundancy scheme would be assumed, which may not always be true and may be dangerous. The ESL basically provides a clarity into ascertaining the operational state of a safety-critical circuit or system in real-time. This could be useful information to initiate appropriate remedial action, preemptively or during a scheduled maintenance. Example NMR and MMR circuits without and with the ESL, which embed similar degrees of fault tolerance, were physically implemented using a 32/28-nm CMOS technology, and their design metrics were estimated. It is found that on average, the proposed MMRESL circuits achieve: (i) respective reductions in area, power, and PDP by 26.8%, 25.2%, and 8.7% compared to the basic NMR circuits without ESL; and (ii) respective reductions in delay, area, power, and PDP by 18.9%, 64.8%, 49.6%, and 52.9% compared to the NMRESL circuits. Compared to the basic NMR circuits, on average, the NMRESL circuits report increases in the critical path delay, area, and power dissipation by 33.3%, 107.8%, and 48.4% respectively. However, compared to the basic MMR circuits, on average, the MMRESL circuits report respective increases in the critical path delay, area, and power dissipation by just 18.8%, 5.8%, and 7%; these represent the minor trade-offs to be made to obtain useful information about the operational state of a MMR implementation in real-time.

Author Contributions: Conceptualization, P.B.; Methodology, P.B., D.M.; Validation, P.B.; Formal Analysis, P.B., D.M., N.M.; Investigation, P.B.; Resources, D.M., N.M.; Data Curation, P.B., D.M.; Writing-Original Draft Preparation, P.B.; Visualization, P.B.; Supervision, D.M., N.M.; Software, D.M.; Project Administration, D.M.; Funding Acquisition, D.M.

Funding: This research was funded by the Academic Research Fund (AcRF) Tier-2 research award of the Ministry of Education (MOE), Singapore grant number MOE2017-T2-1-002 and by the AcRF Tier-1 research award of MOE, Singapore grant number RG132/16.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Miskov-Zivanov, N.; Marculescu, D. Multiple transient faults in combinational and sequential circuits: A systematic approach. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2010**, *29*, 1614–1627. [CrossRef]
2. Baumann, R.C. Radiation-induced soft errors in advanced semiconductor technologies. *IEEE Trans. Device Mater. Reliab.* **2005**, *5*, 305–316. [CrossRef]
3. Quinn, H.; Graham, P.; Krone, J.; Caffrey, M.; Rezgui, S. Radiation-induced multi-bit upsets in SRAM-based FPGAs. *IEEE Trans. Nucl. Sci.* **2005**, *52*, 2455–2461. [CrossRef]
4. Seifert, N.; Slankard, P.; Kirsch, M.; Narasimham, B.; Zia, V.; Brookreson, C.; Vo, A.; Mitra, S.; Gill, B.; Maiz, J. Radiation-induced soft error rates of advanced CMOS bulk devices. In Proceedings of the IEEE International Reliability Physics Symposium, San Jose, CA, USA, 26–30 March 2006.
5. Seifert, N.; Ambrose, V.; Gill, B.; Shi, Q.; Allmon, R.; Recchia, C.; Mukherjee, S.; Nassif, N.; Krause, J.; Pickholtz, J.; et al. On the radiation-induced soft error performance of hardened sequential elements in advanced bulk CMOS technologies. In Proceedings of the IEEE International Reliability Physics Symposium, Anaheim, CA, USA, 2–6 May 2010.
6. Mahatme, N.N.; Bhuva, B.; Gaspard, N.; Assis, T.; Xu, Y.; Marcoux, P.; Vilchis, M.; Narasimham, B.; Shih, A.; Wen, S.-J.; et al. Terrestrial SER characterization for nanoscale technologies: A comparative study. In Proceedings of the IEEE International Reliability Physics Symposium, Monterey, CA, USA, 19–23 April 2015.
7. Rossi, D.; Omaña, M.; Metra, C.; Paccagnella, A. Impact of aging phenomena on soft error susceptibility. In Proceedings of the IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology, Vancouver, BC, Canada, 3–5 October 2011.
8. Omaña, M.; Rossi, D.; Edara, T.S.; Metra, C. Impact of aging phenomena on latches' robustness. *IEEE Trans. Nanotechnol.* **2016**, *15*, 129–136. [CrossRef]
9. Johnson, B.W. *Design and Analysis of Fault-Tolerant Digital Systems*; Addison-Wesley: Boston, MA, USA, 1989; ISBN 978-0201075700.
10. Koren, I.; Krishna, C.M. *Fault-Tolerant Systems*; Morgan Kaufmann Publishers: Burlington, MA, USA, 2007; pp. 11–54, ISBN 978-0120885251.
11. Ban, T.; Naviner, L. Progressive module redundancy for fault-tolerant designs in nanoelectronics. *Microelectron. Reliab.* **2011**, *51*, 1489–1492. [CrossRef]

12. Balasubramanian, P.; Maskell, D.L.; Mastorakis, N.E. Majority and minority voted redundancy for safety-critical applications. In Proceedings of the 61st IEEE International Midwest Symposium on Circuits and Systems, Windsor, ON, Canada, 5–8 August 2018.
13. Balasubramanian, P. ASIC-based design of NMR system health monitor for mission/safety-critical applications. *SpringerPlus* **2016**, *5*, 628. [CrossRef] [PubMed]
14. Parhami, B. Voting networks. *IEEE Trans. Reliab.* **1991**, *40*, 380–394. [CrossRef]
15. Synopsys SAED_EDK32/28_CORE Databook, Revision 1.0.0, January 2012. Available online: https://www.synopsys.com/community/university-program/teaching-resources.html (accessed on 10 December 2017).
16. Mitra, S.; McCluskey, E.J. Word-voter: A new voter design for triple modular redundant systems. In Proceedings of the 18th IEEE VLSI Test Symposium, Montreal, QC, Canada, 30 April–4 May 2000.
17. Balasubramanian, P.; Mastorakis, N.E. Power, delay and area comparisons of majority voters relevant to TMR architectures. In *Recent Advances in Circuits, Systems, Signal Processing and Communications*; Mladenov, V., Ed.; WSEAS Press: Athens, Greece, 2016; pp. 110–117. ISBN 978-1618043665.