IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 December 2023, accepted 5 February 2024, date of publication 23 February 2024, date of current version 5 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3369585

RESEARCH ARTICLE

Reliability Modeling of Fault-Tolerant FPGA-Based Architectures in Space Applications for Soft and Hard Error Recovery

MANAR N. SHAKER[1], AHMED HUSSIEN[1], HASSANEIN H. AMER[2], (Life Member, IEEE), AND BEATRICE SHOKRY[2]

[1]Electronics and Communications Engineering Department, Cairo University, Giza 12613, Egypt
[2]Electronics and Communications Engineering Department, The American University in Cairo, Cairo 11835, Egypt

Corresponding author: Manar N. Shaker (manar.comm2010@gmail.com)

ABSTRACT FPGAs are currently being used in many applications due to their flexibility and re-programmability. Some of these applications operate in harsh environments. One such environment is space. Focusing on space applications, these FPGAs are subjected to soft and hard faults. For newer technology nodes, in addition to the harsh space environment, these errors are more severe. This paper investigates several fault-tolerant architectures to mitigate Single Event Upsets (SEUs), Double Event Upsets (DEUs), Triple Event Upsets (TEUs) as well as hard faults. Conventionally, for TEUs, seven copies of a module are required (7MR). Therefore, a modified 7MR architecture is studied along with two other architectures with six redundant modules: a modified 6MR architecture and a modified Triple Duplex architecture. Using Continuous Time Markov Chains (CTMCs), it is proven that, in many of the cases studied in this article, the modified Triple Duplex architecture has a higher reliability than the modified 7MR architecture. This is a counter-intuitive result. It is also proven that the modified 6MR architecture always has a lower reliability than the modified Triple Duplex architecture even though they both require six redundant modules. The ratio between the relative rates of SEUs, DEUs and TEUs plays an important role in determining the most reliable architecture. Furthermore, the Xilinx Vivado tool with the Kintex7, 7k410tfbg676 device is used to implement the modified 7MR and modified Triple Duplex voters to estimate the area and power consumed by these techniques.

INDEX TERMS DEU, fault tolerance, FPGA, reliability, SEU, TEU.

I. INTRODUCTION

In recent years, many safety-critical applications required a high level of reliability, such as automotive, biomedical, and space applications. Field Programmable Gate Arrays (FPGAs) based on Static Random Access Memory (SRAM) components are often used nowadays in these applications [1], [2]. There are many space applications where FPGA-based platforms can be very useful such as spacecrafts, satellites, and rovers. The extensive number of resources offered by programmable logic devices can be used, for instance, to increase flexibility in the on-board computer in satellites and in the automotive industry. Design modifications can be made up until a fairly late stage in the development process since FPGAs are configurable in the field. In addition, after a satellite is launched, new features and programs can be configured or updated while in space [3].

Space, in particular, has a very harsh environment since there are different particles caused by radiation such as primarily electrons, protons and heavy ions. These particles

The associate editor coordinating the review of this manuscript and approving it for publication was Shunfeng Cheng.

come from a variety of sources, such as the sun, novae, and supernovae; the energy levels of these particles vary depending on the source and the orbit. When these radiations strike an electronic device such as an SRAM- based FPGA, they may cause a problem that is primarily seen in memory elements with static cell implementations. This problem is called Single-Event Effects (SEEs) [4].

SEEs can be classified as soft errors (errors which cause no permanent damage and can be recovered such as Single Event Upsets (SEUs) or Single Bit Errors (SBEs), Single Event Transients (SETs) and Address Decoding faults) and hard errors which permanently damage the semiconductor such as Time Dependent Dielectric Breakdown (TDDB), electro migration and hot carrier effect [5], [6]. For SRAM-based FPGAs, the most frequent faults are SEUs [7].

There are several fault injection techniques for Single-Event Effects (SEEs) such as software-based techniques which use Single-Event Transient (SET) current pulses (double-exponential model) to inject SET fault [8], SEE laser testing to inject SEE faults via a laser beam [9] and heavy-ion irradiation to inject Single Event Functional Interrupt (SEFI) faults [10]. These techniques can help in the determination of the interarrival times of different failures such as SEUs/MEUs. These rates are then used in mathematical models (such as Markov models [11], [12] to calculate reliability. Note that reliability at time t is the probability of a system/module functioning correctly at time t given that it was operational at $t = 0$ [11], [12]. These models take into account the fault-tolerant techniques used in the architecture under study as well as the interarrival times of failure events and repair actions. These interarrival times are represented by the SEU/MEU failure rates and the repair rates.

For an FPGA, SEUs can cause the information stored in a configuration bit to flip, thereby changing the function of the circuit and producing an error. If the correct data is re-written in the failed configuration bit, the problem is solved, and the circuit resumes correct operation. Circuits grow more prone to upsets when technology is scaled down to improve resource integration and use less energy, to the point where Multiple-Event Upsets (MEUs) present new difficulties that cannot be ignored [13]. Particles may affect two or more memory cells (usually adjacent). While SEUs are still the most probable soft faults in FPGAs, some researchers have recently tackled MEUs [1]. This paper will focus on SEUs, Double Event Upsets (DEUs) and Triple Event Upsets (TEUs) as well as hard faults. While the probability of DEUs, TEUs and hard faults is lower than that of SEUs, they cannot be ignored in the very harsh space environment and scaled-down FPGA technology.

There are many fault-tolerant techniques that can be used to mitigate the effects of the faults considered in this work; these techniques must be able to address DEUs and TEUs. Therefore, Triple Modular Redundancy (TMR) is not appropriate here even though it is the most commonly used fault-tolerant architecture; it can only recover from SEUs and hard faults. Since TEUs are considered, at least seven identical copies of a module (7MR) must be used to still have a valid majority. However, it was shown in [14] that a variation of the conventional Triple Duplex architecture may detect and recover from SEUs, DEUs, TEUs and hard faults.

Therefore, in this paper, the focus will be on the modified 7MR technique and the modified Triple Duplex technique described in [14]. Since the platform is a SRAM-based FPGA, the Dynamic Function eXchange (DFX) will be utilized [15]; when an error is detected and identified in one, two or three modules, a DFX process is initiated to overwrite the affected module(s) with a correct partial bit file stored outside the FPGA. If the problem is due to a soft error (SEU, DEU or TEU), DFX will solve it. If the error is due to a hard fault in a module, the architecture will have to operate correctly with one less module. In some situations, the module output will not be available during DFX. The duration of DFX is most often in the range of tens to hundreds of milliseconds (especially if several partial bit files have to be downloaded [16]). There are systems which use the Retry backward error recovery technique meaning that the operating system takes the system back to a point where accurate state information is available [11]. Furthermore, the concept of the Watchdog monitor is used in many contemporary microprocessors (such as the ARM Cortex [17]) to mitigate control flow problems; the system is interrupted, and a reset is applied.

This article studies the reliability of the modified 7MR and the modified Triple Duplex architectures. Reliability is calculated by solving Markov models (CTMCs) with different failure rates. The SHARPE [18] tool is used for the calculations. As mentioned above, the fault model consists of SEUs, DEUs, TEUs and hard faults, one fault at a time. For completeness, a modified 6MR architecture is also studied and compared to the other two main architectures. It will be shown that the modified 6MR technique has the lowest reliability while the modified Triple Duplex technique has the best reliability in most cases. The modified 7MR technique has the best reliability in a few cases. This counter-intuitive result is important because the general rule is that more redundancy leads to a higher reliability. Here, this is not the case.

Other contributions of this research are:

- Designing and implementing the modified 7MR voter on the Xilinx Kintex7, 7k410tfbg676 device.
- Implementing the modified Triple Duplex voter on the Xilinx Kintex7, 7k410tfbg676 device to be able to compare it to the modified 7MR voter (using Vivado tool).
- Proving that the reliability of the modified Triple Duplex architecture is always higher than the modified 6MR architecture.

While this work focuses on FPGAs, techniques following the same reasoning, can be applied to other technologies. Memristors have recently gained a lot of attention in the literature because of their suitability for many important applications such as neuromorphic circuits [19]. However, the fault models in memristor-based circuits are different than those studied in this article. It would be interesting to

study fault tolerance in memristor-based circuits by using appropriate fault models, especially in harsh environments.

This article is organized as follows: Section II next describes the methodology used in this work. Section III then shows the reliability calculations for the modified 6MR, modified 7MR and modified Triple Duplex techniques. Section IV presents the reliability results based on different failure rate ratios and implementation results for the modified Triple Duplex and the modified 7MR voters. The results will be discussed in Section V and at last, Section VI concludes this article.

II. METHODOLOGY

SRAM-based FPGAs are often used in space applications where the environment is extremely harsh. Radiations in space can cause Single Event Upsets (SEUs), Multiple Event Upsets (MEUs) as well as hard faults. SEUs are caused by particles affecting a cell/bit in the configuration memory. These particles cause the content of the cell to toggle. However, the cell is not permanently damaged. This is sometimes referred to as a transient failure; overwriting the cell with the correct information will restore the correct functionality of the circuit. The rate of SEUs is in general much higher than that of the hard faults in a SRAM-based FPGA [20].

Particles can also affect several bits/cells simultaneously [21]. Therefore, Double Event Upsets (DEUs) can occur, but at a lower rate than that of SEUs. Finally, at an even lower rate than that of DEUs, Triple Event Upsets (TEUs) cannot be neglected.

Many fault tolerance techniques have been used to mitigate SEUs and MEUs. Septuple modular redundancy (7MR) was studied in [22]. Seven identical modules are used to recover SEUs, DEUs, TEUs and hard faults. The modified Triple Duplex was the technique used in [14]. The architecture consists of three pairs of identical modules and the fault model had SEUs, DEUs, TEUs and hard faults. References [23] and [24] showed that the scrubbing technique cannot prevent system failures due to soft errors because of the time between error occurrence and detection/recovery during which incorrect data can be propagated; furthermore, the hexa modular redundancy (6MR) scheme was investigated with a fault model consisting of SEUs and DEUs.

Three architectures will be investigated next from a reliability point of view: modified 7MR, modified 6MR and modified Triple Duplex. The fault model will be identical to the one used in [14], i.e., SEUs, DEUs, TEUs as well as hard faults. The afore-mentioned three architectures have enough redundancy to mitigate TEUs. However, before deriving the reliability of these architectures, it is important to clearly understand their fault detection and recovery mechanisms.

A. FAULT- TOLERANT TECHNIQUES

The modified 7MR architecture is shown in Fig. 1. It is composed of seven identical modules (M1, M2, M3, M4, M5, M6, and M7), a 7-input majority voter and seven XORs to compare between each module output and the voter output.

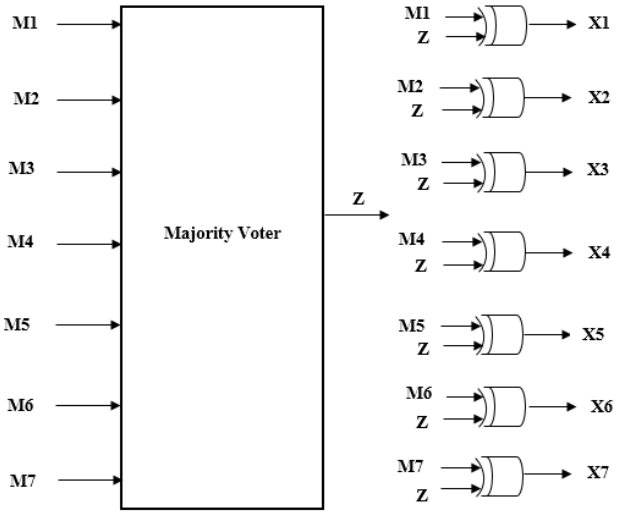


FIGURE 1. 7MR proposed architecture.

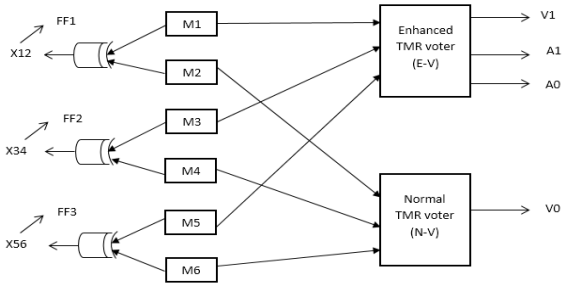


FIGURE 2. Modified Triple Duplex architecture.

The outputs of the seven XOR gates are connected to the Internal Configuration Access Port (ICAP) [25].

If there is a fault in a certain module *Mi*, its output will be different than the output of the voter and the XOR output $Xi = 1$; then, the ICAP will perform DFX on module *Mi*.

The second architecture is the modified 6MR architecture. It is identical to the modified 7MR one except that it has only six identical modules instead of seven. For the majority voter, since the number of inputs is even, it will indicate a system failure when there is no majority, i.e., three modules have the same output, and the other three modules have the opposite output.

The redundancy scheme proposed in [14] is shown in Fig. 2. This scheme is inspired by the Triple Duplex architecture initially proposed in [26]. It is composed of six identical modules (M1, M2, M3, M4, M5, and M6), three XORs (X12, X34, X56), three Flip-flops (FF1, FF2, FF3), a normal Triple Modular Redundancy (TMR) voter (N-V) and an enhanced TMR voter (E-V) [14].

The three XORs are used to compare the outputs of each pair of modules.

If the XOR output becomes '1', DFX is initiated on the corresponding pair. However, if it remains '1' after performing DFX, then one of the modules in this pair has been infected with a hard fault. This means that issuing further

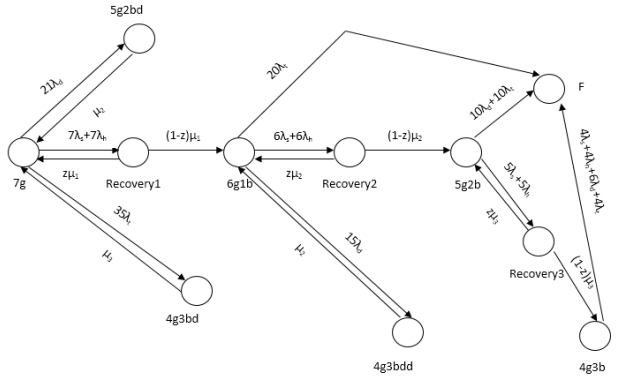


FIGURE 3. Modified 7MR Markov model.

DFX operations is useless. Accordingly, no more DFX is applied to this pair. For this purpose, a Flip-Flop is used to store the XOR status, such that its output is investigated after applying DFX; if it is still '1', the corresponding pair is excluded from the DFX list. V0 is the output of the N-V voter such that it is set to the most repeated output among the three modules' outputs. V1, A0, and A1 are the outputs of the E-V voter. V1 is a majority output as V0. A0 and A1 represent the address of the faulty module, if any. Suppose there is no faulty module, then A1A0 = 00. However, A1A0 = 01, 10, or 11 if the first, second, or third module is erroneous, respectively. If A1A0 is not '00', DFX is performed on the pair that has the faulty module [14].

The functionality of the architecture proposed in [14] is as follows. The output is only propagated to the rest of the system if V0 = V1 (both voters agree). This may happen in two situations; either both are correct (V0V1 = PP), or both are erroneous (V0V1 = P'P'). Otherwise, the system stops until V0 becomes equal to V1 again with the help of DFX. However, if the DFX is not able to return the system to such a state due to hard faults, for example, it will indicate a system failure. This is a safe failure because no incorrect data is sent to the rest of the system. It was also assumed in [14] that the rest of the system, which receives the data produced by the fault-tolerant system under study, will be able to withstand such a delay incurred by the DFX capability. For example, in the context of NCSs, the fault-tolerant system will incorporate the controller, which sends its outputs to the appropriate actuators. In NCSs, the actuators are smart and will be able to withstand such a short interruption (which is in the order of tens of ms for one module and therefore less than 500ms if DFX is applied to all six modules). On the other hand, if the system under study produces P'P', the error will not be detected, and a system failure occurs; this might be an unsafe failure.

It was proven in [14] that this scheme is always able to tolerate any SEUs, DEUs, and TEUs (soft errors) while fully recovering the system with the help of DFX. However, the system is not fully back to its initial state only if hard faults have infected the system. In some cases, even the soft faults coming after hard faults are not fully repaired, unlike the case when the system does not have a previous record of hard faults.

B. MARKOV MODELS OF FAULT-TOLERANT TECHNIQUES

The Markov model (CTMC) used to calculate the reliability of the modified 7MR architecture is shown in Fig. 3. The modified 7MR system will start with state "7g" with seven good modules and zero failed modules. When a SEU or hard error occurs in one of the seven modules, the system moves to the 'Recovery1" state. While in this state, DFX is applied on the failed module. If the problem is a SEU, DFX will repair the module by overwriting it with a correct bitstream. However, if the problem is due to a hard failure, DFX will not be able to repair the module. The transition from state "7g" to state "Recovery1" occurs at a rate of $7^*(\lambda_s + \lambda_h)$ since any of the seven modules can be affected by either a SEU or a hard fault. μ_1 is a repair rate which depends on the amount of time taken to complete the DFX action and this time is a function of the size of the bit file.

Assuming the repair time is exponentially distributed [11], this time is equal to $1/\mu_1$. From now on, $(1/\mu_i)$ will be the time required to download $\boldsymbol{i}$ modules using DFX. After a module is affected by either a SEU or a hard fault, the system moves to the "Recovery1" state. It is not known at this time whether the problem was due to a SEU or a hard fault; hence, a DFX action is initiated. If the problem was due to a SEU, it will be repaired, and the system moves back to state "7g". On the other hand, if the problem was due to a hard fault, the number of operational modules is reduced from seven to six; the system therefore moves to state "6g1b". Let z be the conditional probability that a failure is due to a SEU (temporary) given that a failure has occurred. Hence, in case of a SEU, the system returns to state "7g" at a rate $z * \mu_1$.

In case of a hard failure, the system moves from "Recovery1" state to "6g1b" (state with six good modules and one failed module with a hard error) at a rate of $(1 - z) * \mu_1$. When a DEU occurs while in state "7g", the system moves to state "5g2bd" with five good modules and two failed modules with a DEU, at a rate of $21^*\lambda_d$. It returns to state "7g" at a rate μ_2 since two modules must be downloaded. When a TEU occurs while in state "7g", the system moves to state "4g3bd" with four good modules and three bad modules with a TEU, at a rate of $35^*\lambda_t$. At a rate of μ_3, the system returns to state "7g".

The system moves from state "6g1b" to state "Recovery2" at a rate $6^* (\lambda_s + \lambda_h)$. In case of a hard failure, the system moves from state "Recovery2" to state "5g2b" (state with five good modules and two failed modules with hard errors), at a rate of $(1 - z)^* \mu_2$. However, in case of a SEU, the system returns to state "6g1b", at a rate of $z * \mu_2$. When a DEU occurs while in state "6g1b", the system moves to the "4g3bdd" state with four good modules and two failed modules with a DEU and one hard error, at a rate of $15^*\lambda_d$. At a rate of μ_2, DFX will repair the DEU and the system returns to state "6g1b". It is assumed here, that the module with the hard failure has been identified by the system and

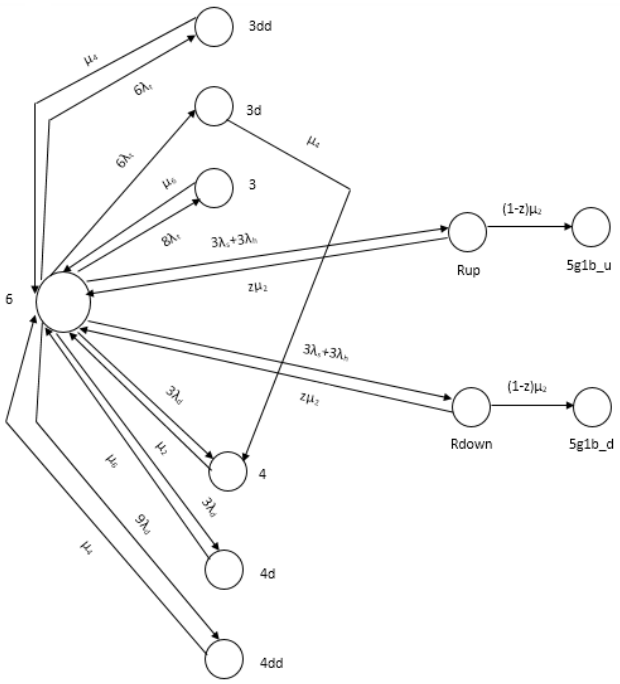


FIGURE 4. Modified Triple Duplex Markov model part 1.

that DFX will not be applied to it anymore; otherwise, the transition rate would be μ_3, i.e., three modules have to be downloaded. When a TEU occurs while in state ''6g1b'', the system moves to the Failure state ''F'', at a rate $20^*\lambda_t$.

While in state ''5g2b'', the system moves to state ''Recovery3'' at a rate of $5^*\left(\lambda_s+\lambda_h\right)$. In case of a hard error, the system moves from state ''Recovery3'' to state ''4g3b'' (four good modules and three failed modules with hard errors), at a rate $(1-z)^*\ \mu_3$ where $\mu_3 = 0.333\mu_1$. In case of a SEU, the system returns to state ''5g2b'' at a rate of $z\ ^*\mu_3$. This is because there will still be four good modules and the majority voter can produce the correct output despite three incorrect inputs. However, when a DEU or TEU occurs, the system moves from ''5g2b'' state to state ''F'', at a rate of $10^*\left(\lambda_d+\lambda_t\right)$; a maximum of only three modules will produce a correct output. Similarly, the system moves from state ''4g3b'' to state ''F'', at a rate of $4^*\left(\lambda_s+\lambda_h+\lambda_t\right)+6\lambda_d$.

Regarding the modified Triple Duplex architecture, the Markov model used to calculate its reliability is depicted in Figs. 4, 5 and 6. It can be divided into 3 parts. Part 1 (see Fig. 4) describes the states of the system starting from the error-free state till the architecture loses one of its six modules due to a hard failure; states 5g1b_u and 5g1b_d are two states with only five operational modules. Part 2 (see Fig. 5) starts from state 5g1b_u and models system behavior till state F where the entire system fails. Finally, Part 3 (see Fig. 6) starts from state 5g1b_d and ends with state F.

In Part 1, the modified Triple Duplex system will start in state ''6'' with six good modules and zero bad modules. When a SEU or hard failure occurs, the system moves to recovery states ''Rup'' and 'Rdown'' at a rate of $3^*\left(\lambda_s+\lambda_h\right)$ for both states. The rationale behind having two different states, each having five operational modules, is that the two voters used in the architecture are not identical and the architecture is not symmetrical. A failure in M1, M3 or M5 (all connected to the E_V voter will not affect the system in the same way as a failure in M2, M4 or M6 (connected to the N_M voter); the E_V voter can identify the failed module while the N_M voter cannot. As soon as an error is detected, a DFX process is initiated for the affected pair of modules (not just the failed module in the pair); if the problem was due to a SEU, the architecture returns to state ''6'' at a rate of $z^*\mu_2$ where μ_2 corresponds to the time required to download two modules and **z** is the conditional probability that the problem was due to a SEU given that a problem has occurred. Alternatively, if the problem was due to a hard fault, the architecture moves to state 5g1b_u at a rate of (1)-z)$^*\mu_2$ if the hard failure was in M1, M3 or M5 and to state 5g1b_d at the same rate if the hard failure was in M2, M4 or M6.

Starting from state ''6'', if the first problem is due to a DEU, the system moves to state ''4'' if the DEU affected M1M2 or M3M4 or M5M6, i.e., two modules belonging to the same pair. The rate is $3\lambda_d$. A DFX process will require downloading two files at a rate of μ_2. Since a DEU is a soft error, DFX will repair it and the architecture returns to state ''6''. If the DEU affects M1M3 or M1M5 or M3M5, the architecture moves to state ''4d'' (at a rate of $3\lambda_d$) which requires downloading all six modules at a rate of μ_6. For example, if M1 and M3 are affected by a DEU, X12 and X34 will both be activated, requiring a DFX process for M1, M2, M3 and M4. Furthermore, since both M1 and M3 have failed, the E-V outputs A1A0 will indicate that the failed module is M5, thereby requiring a DFX operation on the M5M6 pair. For the remaining pairs of modules which could be affected by a DEU (there are nine other pairs M1M4 or M1M6 or M2M3 or M3M6 or M2M5 or M4M5 or M2M4 or M2M6 or M4M6), the architecture moves to state ''4dd'' at a rate of $9\lambda_d$; this requires downloading four modules at a rate of μ_4. For instance, if M2 and M5 are affected by a DEU, X12 and X56 will be activated, resulting in a DFX process on M1, M2, M5 and M6. The A1A0 outputs of the E-V voter indicate that M5 has failed but DFX is already applied to M5 due to the activation of X56. Since X34 is not activated, no DFX is required for M3 or M4.

In summary, states ''4'', ''4d'' and ''4dd'' indicate that there are four good modules and two failed modules because of a DEU. Depending on which pair of modules were affected by DEUs, the repair rate will differ based on the number of bit files being downloaded. In state ''4dd'', based on the number of available good modules for the N_V voter, the system will continue working during the DFX process in six pairs M1M4 or M1M6 or M2M3 or M3M6 or M2M5 or M4M5 and in the remaining three pairs M2M4 or M2M6 or M4M6, the system will be stopped during the DFX process.

Regarding TEUs, there are twenty cases to consider. Eight of these cases require DFX at a rate of μ_6: M1M3M5, M2M4M6, M1M3M6, M1M5M4, M3M5M2, M2M4M5,

M2M6M3, and M4M6M1. Any of these eight cases will take the system from state "6" to state "3" at a rate of $8\lambda_t$ and DFX will take it back to state "6" at a rate of μ_6. For example, if M1, M3 and M5 are affected by a TEU, X12 and X34 and X56 will be activated, requiring a DFX process for M1, M2, M3, M4, M5 and M6. Furthermore, since both M1 and M3 and M5 have failed, the E-V outputs A1A0 will indicate that the system does not have any faulty module. Next consider the six cases M1M2M4, M1M2M6, M3M4M2, M3M4M6, M5M6M2 and M5M6M4. These cases will take the system to state "3dd" at a rate of $6\lambda_t$ and DFX will return the system to state "6" at a rate of μ_4. For example, if M1, M2 and M4 are affected by a TEU, X34 will be activated, requiring a DFX process for M3 and M4. Furthermore, since M1 only has failed, the E-V outputs A1A0 will indicate that the failed module is M1, thereby requiring a DFX operation on the M1M2 pair in addition to DFX for M3 and M4, so downloading four modules is required. The remaining six cases are different; take, for example, a TEU affecting M1, M2 and M3; X34 will be activated, requiring a DFX process for M3 and M4. Furthermore, since both M1 and M3 have failed, the E-V outputs A1A0 will indicate that the failed module is M5, thereby requiring a DFX operation on the M5M6 pair in addition to DFX for M3 and M4, so downloading four modules is required. DFX will repair the faults in M3 but both M1 and M2 are still erroneous. Now the architecture has two faulty modules as in state "4" in Fig. 4. This is why the rate from state "6" to state "3d" at a rate of $6\lambda_t$ and then, after DFX, to state "4" at a rate of μ_4.

In the previous section, the system's behavior was studied from the beginning of operation (all six modules are operational) till one of the modules suffers a hard fault. This is represented by the two states in Fig. 4: 5g1b_u where one of the modules connected to E-V has a hard fault, or state 5g1b_d where one of the modules connected to N-V suffers a hard fault. Next, the effect of a second fault, while in state 5g1b_u, is studied. Without any loss of generality, it will be assumed that M1 is the module which has failed and moved the system to state "5g1b_u".

In part 2, as shown in Fig. 5, when in state "5g1b_u", a hard fault or a SEU in the module belonging to a pair with the other module already affected by a hard fault, will take the system to state "g1_u" at a rate of $\lambda_s + \lambda_h$; the pair has irrecoverable faults in both modules since DFX cannot be applied to the pair which already has one module affected by a hard fault [14]. So, in state "g1_u", the system will have two modules belonging to the same pair (M1M2, M3M4 or M5M6) with irrecoverable faults but is still operating correctly. While in state "5g1b_u", if any of the other four modules suffers a SEU, the system either moves to state "4gd_u" at a rate of $2\lambda_s$ and returns to "5g1b_u" at a rate of μ_4 after DFX is applied to four modules or moves to state "4gdd_u" at a rate of $2\lambda_s$ and also returns to state "5g1b_u" at a rate of μ_2 after DFX is applied to two modules. For example, if a hard failure affects M1 then a SEU affects M3, X34 will be activated; so DFX is required for M3 and M4. The

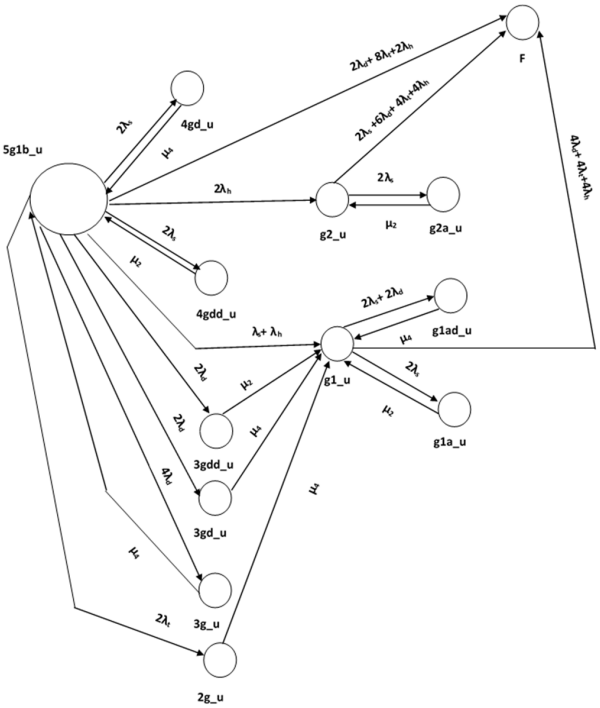


FIGURE 5. Modified Triple Duplex Markov model part 2.

A1A0 outputs of the E-V voter indicate that M5 has failed; so DFX is also required for M5 and M6. Hence, DFX is applied to M3, M4, M5 and M6, the SEU in M3 is repaired, and the system returns from state "4gd_u" to state "5g1b_u" at a rate of μ_4. Alternatively, if a hard failure affects M1 then a SEU affects M4, X34 will be activated but the A1A0 outputs of the E-V voter indicate that M1 has failed. Since the M1M2 pair is already marked as failed, no DFX will be attempted on M1M2, but it will applied to M3 and M4 since X34 was activated. Hence the system returns from state "4gdd_u" to state "5g1b_u" at a rate of μ_2.

Regarding DEUs, there are ten possibilities ($10 = {}^5C_2$). For four of these possibilities, the system moves to state "3g_u" at a rate of $4\lambda_d$ and returns to state "5g1b_u" with a rate of μ_4. Still assuming that M1 has already failed, a DEU affecting M3M5 or M3M6 or M4M5 or M4M6, will take the system to state "3g_u". For example, if the DEU affects M3 and M5, X34 and X56 will be activated while A1A0 will point to M1; therefore, DFX is applied to M3, M4, M5 and M6 and the system returns from state "3g_u" to state "5g1b_u" at a rate of μ_4. Two DEUs cause a system failure at a rate of $2\lambda_d$: a DEU affecting M3M4 or M5M6. For M3M4, for example, only X12 is activated since M1 has already failed but no DFX is applied. A1A0 indicates that M5 has failed, initiating a DFX process on M5M6, which does not fix the problems in M3M4; the system output is incorrect, and the system fails.

For the four remaining possibilities, two of them will take the system to state "3gd_u" and then to state "g1_u" at a rate of μ_4 while the other two will take the system to state "3gdd_u" and then to state "g1_u" at a rate of μ_2.

Remember that, in state ''g1_u'', both modules of a same pair have failed but the output of the system is still correct.

For TEUs, there are also ten possibilities ($10 = {}^5C_3$). Only two of these possibilities can be recovered from, namely M2M3M5 and M2M4M6. For M2M3M5 (and given that M1 had already failed before the TEU), the output of the E-V voter will be incorrect, but the output of the N-V will be correct. Both X34 and X56 will be activated, initiating a DFX process on M3M4M5M6. This will repair M4 and M5, both voter outputs will be correct, and the system moves to state ''g1_u'' where both M1 and M2 have failed. The other eight combinations of modules, if affected by a TEU, will lead to a system failure. For example, if a TEU affects M2M3M6, both voters will produce an incorrect output which is not detected by the system, therefore leading to a complete failure since incorrect data was propagated beyond the voters.

Finally, when in state ''5g1b_u'' and assuming that M1 has already failed, let one of the modules connected to the E-V voter be affected by a hard fault. As mentioned above, if the other module in the same pair suffers a hard fault, the system moves to state ''g1_u'' at a rate of $\lambda_s + \lambda_h$. If any of the two other modules (M3 or M5) connected to E-V suffers a hard fault, the output of the E-V voter will always be incorrect, and a total system failure occurs, so the system moves to the failure state at a rate $2\lambda_h$. However, if any of the modules (M4 or M6) connected to the N-V voter fails, the system does not fail but moves to state ''g2_u'' at a rate $2\lambda_h$ where each voter has one module with a hard fault and two operational modules and therefore, can still produce the correct output.

When in state ''g1_u'', where both modules in the same pair are not operational (for example, M1 and M2), most other failures will lead to a complete system failure. This is expected since one pair has completely failed. The system will only survive a few more faults. For example, if M4 fails, X34 will be activated and DFX will be applied to M3 and M4, which fixes M4 and the system returns to ''g1_u'' at a rate of μ_2.

The situation is similar when the system is in state ''g2_u'' where two modules have suffered hard faults, for example, M1 and M4. Let M5 suffer a SEU. The two voters will produce complementary outputs which prevents the system from propagating this output. X34 will be activated and a DFX on M3 and M4 will repair M4, thus returning the system to state ''g2_u''.

Regarding part 3 of the Markov model (see Fig. 6), it is similar to part 2 with some differences due to the fact (as mentioned above) that the E_V is not identical to the N_M voter. One of these differences is that, starting from state ''5g1b_d'', none of the ten combinations of modules which could be affected by a DEU, leads to a system failure; after a DEU, the system moves to states ''3g_d'', ''3gd_d'', ''3gdd_d'' or ''3gddd_d''. Next are a few examples to illustrate the effects of a DEU after one of the six modules has suffered a hard failure. Assume M2 is the module affected by a hard failure; then, any of the remaining five operational modules can be affected by a DEU. There are ten cases to consider (all possible pairs of operational modules). Note also that, since M2 has a hard failure, no DFX will be applied to M1M2 as mentioned above.

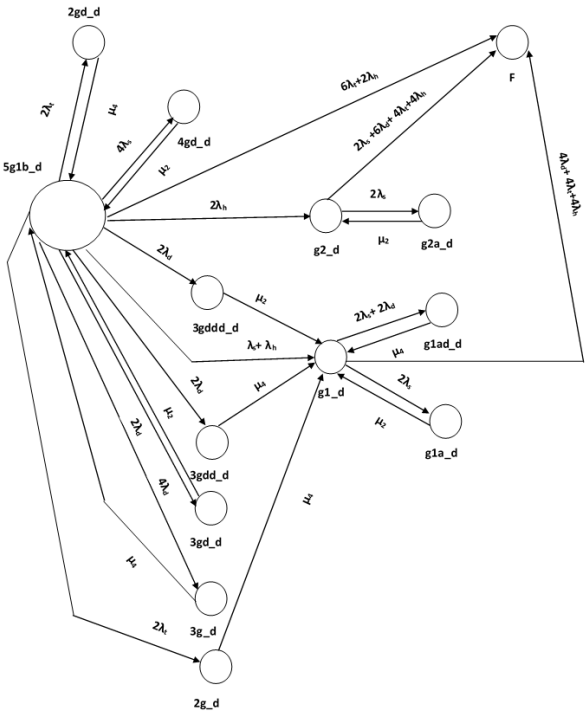


FIGURE 6. Modified Triple Duplex Markov model part 3.

Consider first a DEU affecting M3M5. M2, M3 and M5 will produce erroneous outputs. DFX will be applied to modules M3M4M5M6 (at a rate of μ_4 since 4 bit files have to be downloaded). Modules M3 and M5 will be repaired and the system will resume operation with only a hard failure in M2. In Fig. 6, these are the transitions between states ''5g1b_d'' and ''3g_d''. The other three pairs of modules which will behave similarly when affected by a DEU are: M3M5, M3M6, M4M5 and M4M6.

Consider next a DEU in M3M4. M2, M3 and M4 will produce incorrect outputs. DFX will be applied to M3M4 at a rate of μ_2 and the M3M4 pair will be operational again. A DEU in M5M6 will be treated similarly. In Fig. 6, these are the transitions between states ''5g1b_d'' and ''3gd_d''.

Consider next a DEU in M1M5. M1, M2 and M5 will be erroneous. DFX will be applied to M3M4M5M6 (at a rate of μ_4, which will only repair M5. The system will continue operating correctly with M2 having a hard failure and M1 having a SEU – this is state ''g1_d''. A DEU in M1M3 will follow the same reasoning; this is why the transition between states ''5g1b_d'' and ''3gdd_d'' is equal to $2\lambda_d$.

Finally, consider a DEU in M1M6. Modules M1, M2 and M6 will produce incorrect outputs. DFX will therefore be applied to M5M6 at a rate of μ_2. This will repair M6, the system will operate correctly but M1 and M2 will remain erroneous. This explains the transition from states ''5g1b_d'' to state ''3gddd_d'' and then to state ''g1_d''. The same reasoning is valid for pair M1M4.

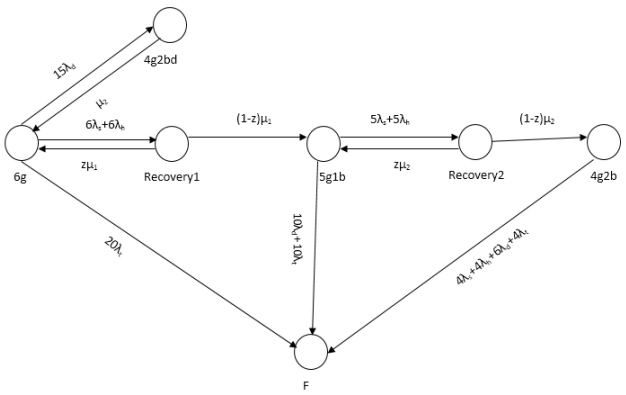


FIGURE 7. Modified 6MR Markov model.

There is another difference regarding TEUs. There are four possibilities out of ten which can be recovered from, namely M1M3M5 and M1M4M6 and M3M4M6 and M4M5M6. Two of them (M1M3M5 and M1M4M6) will take the system to state "2g_d" and then to state "g1_d" at a rate of μ_4. For M1M3M5 (and given that M2 had already failed before the TEU), the output of the E-V voter will be incorrect, but the output of the N-V will be correct. For the other two possibilities M3M4M6 and M4M5M6, the system will move to state "2gd_d" and then return to state "5g1b_d" at a rate of μ_4.

For M3M4M6 (and given that M2 had already failed before the TEU), the output of the E-V voter will be correct, but the output of the N-V will be incorrect. X56 will be activated and A1A0 will point to M3, initiating a DFX process on M3M4M5M6. This will repair M3, M4, M5 and M6, both voter outputs will be correct, and the system returns from state "2gd_d" to state "5g1b_d" at a rate of μ_4. The other six combinations of modules, if affected by a TEU, will lead to a system failure for the same reason in part 2 which is that the incorrect output will be propagated. The rest of the Markov model in Fig. 6 follows the same type of reasoning as the Markov in Fig. 5.

The modified 6MR architecture can detect SEUs, DEUs, and hard faults. The Markov model of the modified 6MR architecture is shown in Fig. 7. This architecture will start with state "6g" with six good modules and zero bad modules. When a SEU or a hard error occurs, the system moves to the "Recovery1" state, at a rate $6^*\left(\lambda_s+\lambda_h\right)$. In case of a hard error, the system moves from the "Recovery1" state to the "5g1b" state with five good modules and one bad module with a hard error, at a rate $(1-z)^*\,\mu_1$. In case of a SEU, the system returns to state "6g", at a rate $z^*\mu_1$. When DEUs occur, the system moves to state "4g2bd" with four good modules and two bad modules with DEUs, at a rate of $15\lambda_d$. When TEUs occur, the system moves to the failure state "F" at a rate $20\lambda_t$. The system moves from "5g1b" state to the "Recovery2" state at a rate $5^*\left(\lambda_s+\lambda_h\right)$. The "Recovery2" state is similar to the "Recovery1" state for deciding whether the problem is due to a SEU or a hard error after applying DFX. The system moves from the "Recovery2" state to the "4g2b" state with four good modules and two bad modules with hard errors, at a rate $(1-z)^*\,\mu_2$. The system moves from "Recovery2" state to "5g1b" state, at a rate $z^*\mu_2$. The system moves to state "F" from "5g1b" state and from "4g2b" state with rates $10^*\left(\lambda_d+\lambda_t\right)$ and $4^*\left(\lambda_s+\lambda_h+\lambda_t\right)+6\lambda_d$ respectively.

III. RELIABILITY CALCULATIONS

Reliability $(\boldsymbol{R(t)})$ is the probability that a system is alive at time $\boldsymbol{t}$ given that it was alive at $\boldsymbol{t}=\mathbf{0}$ [11]. Reliability is the probability of not being in state "F" at any time $\boldsymbol{t}$ as follows:

$$\boldsymbol{R}\,(t)=\mathbf{1}-\boldsymbol{P_F(t)} \tag{1}$$

where $\boldsymbol{P_F(t)}$ is the probability of the system being in state "F" (failure state).

The modified 6MR, modified 7MR and modified Triple duplex Markov models can be solved using the Chapman-Kolmogorov equations [11]. $\boldsymbol{P_i(t)}$ is the probability of residing in state $\boldsymbol{i}$ at time $\boldsymbol{t}$ and $\mathbf{T}$ is the transition matrix. $\boldsymbol{P_i(t)}$ and $\mathrm{T_{modified6MR}}$ for the modified 6MR Markov model are shown in (2) and (3), as shown at the bottom of the next page.

Therefore, assuming that $P_{6g}\,(0)=1$, $P_{Recovery1}\,(0)=P_{5g1b}\,(0)=P_{Recovery2}\,(0)=P_{4g2b}\,(0)=P_{4g2bd}\,(0)=P_F\,(0)=0$ and using the Transition Rate Matrix in (3) and substituting in (2), the Chapman-Kolmogorov equations can be solved in order to obtain $P_i\,(t)\,\forall i$. Then (1), (4) and (5), as shown at the bottom of the next page, are used to obtain $R\,(t)$.

Following the same reasoning, the reliability for the modified 7MR architecture can be obtained using $P_i\,(t)$ and $\mathrm{T_{modified7MR}}$ for the modified 7MR Markov model as in (6) and (7), as shown at the bottom of the next page. Therefore, assuming that $P_{7g}\,(0)=1$, $P_{Recovery1}\,(0)=\cdots\cdots\cdots\cdots=P_F\,(0)=0$ and using the Transition Rate Matrix in (7) and substituting in (6) and (8), as shown at the bottom of the next page, the Chapman-Kolmogorov equations can be solved in order to obtain $P_i\,(t)\,\forall i\{F,\ldots\ldots\ldots..,Recovery1,7g\}$. Then (1) is used to obtain $R\,(t)$. The reliability for the modified Triple Duplex architecture can be obtained using $P_i(t)$ and $T_{modifiedTripleDuplex}$ for the modified Triple Duplex Markov model as in (9), (10), as shown at the bottom of the next page, and (11).

Then,

$$\frac{dP}{dt}=P*T_{modified6MR} \tag{4}$$

And,
Then,
Then,

$$P=\left[P_6(t),P_{Rup}(t),\ldots\ldots\ldots\ldots\ldots\ldots,P_F(t)\right] \tag{11}$$

Assuming that $P_6\,(0)=1$, $P_{Rup}\,(0)=\cdots\cdots\cdots\cdots=P_F\,(0)=0$ and using the Transition Rate Matrix in (10) and substituting in (9), the Chapman-Kolmogorov equations can be solved in order to obtain $P_i\,(t)\,\forall i\{F,\ldots\ldots\ldots..,Rup,6\}$. Then (1) is used to obtain $R\,(t)$.

IV. RESULTS

A. RELIABILITY RESULTS

The Markov models for the modified Triple Duplex, modified 7MR and modified 6MR architectures are simulated using SHARPE [18] to calculate the reliability. The modules used in this article are an Artificial Neural Network (ANN) [27], a Picoblaze soft processor [16] and an Image Processing Processor [28]. The ANN is used in space applications due to its generality, suitable performance, adaptability and low energy consumption [29]. Soft processors such as the Picoblaze are widely used in space applications [30]. Image processing is used in important space applications such as Vision-Based Navigation (VBN) [31]. Let the module SEU rate be λ_s = 0.00002/hr, 0.000033/hr and 0.000086/hr based on [16], [27], [28], [32] respectively. As mentioned in [24], there is a wide range of ratios between the rates of SEUs and MEUs as well as the rates of DEUs and TEUs [1], [33], [34]. In this article, several ratios are simulated (λ_s =3 λ_d and λ_s =20 λ_d, λ_d = 5 λ_t, λ_d = 10 λ_t, λ_d = 20 λ_t and λ_d = 30 λ_t) to investigate the effect of this ratio on system reliability. Let the module hard rate $\lambda_h = \lambda_s/250$ [20] and the conditional probability z =0.9. Let μ_1 be the repair rate of one module where 1/ μ_1 is the average time to download one bit file corresponding to the module. μ_1 = 88152/hr, 53731/hr and 20665.9/hr for M1, M2 and M3 respectively based on [16], [27], [28]. μ_2 = 1/2 μ_1 as 1/ μ_2 is the time needed for downloading two bit files. Similarly, μ_3 = 1/3 μ_1, μ_4 = 1/4 μ_1, μ_5 = 1/5 μ_1 and μ_6 = 1/6 μ_1 depending on the number of bit files needed to be download and the required time. Tables 1 through 12 show the reliability (%) results for the modified 6MR, the modified 7MR and the modified Triple Duplex architectures versus time (in months) using SHARPE.

The data in Table 5 is presented in a graph (see Fig. 8) where it can be seen that the reliability of the modified Triple Duplex architecture is higher than that of the modified 7MR one. Furthermore, the reliability of the modified 6MR architecture is much lower than both of them.

B. IMPLEMENTATION RESULTS

The modified 7MR voter (as shown in Fig. 1) consists of a combinational circuit which produces the majority of the outputs of the seven identical modules. For example, if the at a certain point in time, modules M1, M4 and M5 have failed, the input to the majority voter will be: 1001100 and its output

$$\left[\frac{dP_{6g}(t)}{dt}, \frac{dP_{Recovery1}(t)}{dt}, \frac{dP_{5g1\ b}(t)}{dt}, \frac{dP_{Recovery2}(t)}{dt}, \frac{dP_{4g2\ b}(t)}{dt}, \frac{dP_{4g2\ bd}(t)}{dt}, \frac{dP_F(t)}{dt}\right]$$
$$= \left[P_{6g}(t), P_{Recovery1}(t), P_{5g1\ b}(t), P_{Recovery2}(t), P_{4g2\ b}(t), P_{4g2\ bd}(t), P_F(t)\right] * T_{modified6MR} \quad (2)$$

$$T_{modified\ 6MR} = \begin{bmatrix} -6\lambda_s - 15\lambda_d - 20\lambda_t - 6\lambda_h & 6\lambda_s + 6\lambda_h & 0 & 0 & 0 & 15\lambda_d & 20\lambda_t \\ z^*\mu_1 & -z^*\mu_1 - (1-z)^*\mu_1 & (1-z)^*\mu_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5\lambda_s - 10\lambda_d - 10\lambda_t - 5\lambda_h & 5\lambda_s + 5\lambda_h & 0 & 0 & 10\lambda_d + 10\lambda_t \\ 0 & 0 & z^*\mu_2 & -z^*\mu_2 - (1-z)^*\mu_2 & (1-z)^*\mu_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -4\lambda_s - 6\lambda_d - 4\lambda_t - 4\lambda_h & 0 & 4\lambda_s + 6\lambda_d + 4\lambda_t + 4\lambda_h \\ \mu_2 & 0 & 0 & 0 & 0 & -\mu_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & & \end{bmatrix} \quad (3)$$

$$P = \left[P_{6g}(t), P_{Recovery1}(t), P_{5g1\ b}(t), P_{Recovery2}(t), P_{4g2\ b}(t), P_{4g2\ bd}(t), P_F(t)\right] \quad (5)$$

$$\left[\frac{dP_{7g}(t)}{dt}, \frac{dP_{Recovery1}(t)}{dt}, \ldots \ldots, \frac{dP_F(t)}{dt}\right] = \left[P_{7g}(t), P_{Recovery1}(t), \ldots \ldots \ldots, P_F(t)\right] * T_{modified7MR} \quad (6)$$

$$T_{\substack{modified \\ 7MR}} = \begin{bmatrix} -7\lambda_s - 21\lambda_d - 35\lambda_t - 7\lambda_h & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\ z * \mu_1 & \ldots & \ldots & \ldots & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \ldots & \ldots & 0 & 0 \end{bmatrix} \quad (7)$$

$$P = \left[P_{7g}(t), P_{Recovery1}(t), \ldots \ldots \ldots \ldots, P_F(t)\right] \quad (8)$$

$$\left[\frac{dP_6(t)}{dt}, \frac{dP_{Rup}(t)}{dt}, \ldots \ldots, \frac{dP_F(t)}{dt}\right] = \left[P_6(t), P_{Rup}(t), \ldots \ldots \ldots, P_F(t)\right] * T_{modified\ Triple\ Duplex} \quad (9)$$

$$T_{modified\ Triple\ Duplex} = \begin{bmatrix} 6\lambda_s - 15\lambda_d - 20\lambda_t - 6\lambda_h & \ldots & \ldots & \ldots & \ldots & 0 \\ z^*\mu_2 & \ldots & \ldots & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \ldots & 0 & 0 \end{bmatrix} \quad (10)$$

TABLE 1. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.00002/h$ and $\lambda_s = 3\ \lambda_d$.

Time (Months)	$\lambda_d = 5\ \lambda_t$			$\lambda_d = 10\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	88.89	99.66	99.81	93.88	99.82	99.86
12	78.30	98.71	99.26	87.25	99.27	99.43
18	68.48	97.21	98.36	80.47	98.36	98.70
24	59.55	95.25	97.11	73.77	97.11	97.68
30	51.51	92.90	95.56	67.28	95.54	96.37

TABLE 2. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.00002/h$ and $\lambda_s = 3\ \lambda_d$.

Time (Months)	$\lambda_d = 20\ \lambda_t$			$\lambda_d = 30\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	96.57	99.90	99.88	97.46	99.93	99.89
12	92.27	99.58	99.52	93.97	99.68	99.55
18	87.46	99.01	98.89	89.88	99.22	98.95
24	82.39	98.18	97.98	85.43	98.54	98.08
30	77.24	97.09	96.81	80.81	97.63	96.95

TABLE 3. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.00002/h$ and $\lambda_s = 20\ \lambda_d$.

Time (Months)	$\lambda_d = 5\ \lambda_t$			$\lambda_d = 10\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	98.19	99.94	99.97	99.03	99.97	99.97
12	96.20	99.78	99.88	97.84	99.88	99.91
18	94.04	99.51	99.73	96.42	99.72	99.79
24	91.70	99.12	99.51	94.77	99.47	99.62
30	89.20	98.60	99.24	92.92	99.14	99.40

TABLE 4. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.00002/h$ and $\lambda_s = 20\ \lambda_d$.

Time (Months)	$\lambda_d = 20\ \lambda_t$			$\lambda_d = 30\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	99.45	99.98	99.98	99.59	99.98	99.98
12	98.66	99.93	99.92	98.95	99.94	99.93
18	97.63	99.82	99.82	98.04	99.86	99.83
24	96.35	99.66	99.68	96.89	99.72	99.70
30	94.83	99.42	99.49	95.50	99.51	99.51

TABLE 5. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.000033/h$ and $\lambda_s = 3\ \lambda_d$.

Time (Months)	$\lambda_d = 5\ \lambda_t$			$\lambda_d = 10\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	81.52	99.09	99.49	89.50	99.50	99.61
12	65.02	96.60	97.98	78.23	98.00	98.41
18	51.08	92.89	95.58	67.31	95.58	96.40
24	39.74	88.34	92.44	57.36	92.40	93.69
30	30.72	83.26	88.74	48.55	88.62	90.40

TABLE 6. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.000033/h$ and $\lambda_s = 3\ \lambda_d$.

Time (Months)	$\lambda_d = 20\ \lambda_t$			$\lambda_d = 30\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	93.78	99.71	99.68	95.26	99.79	99.70
12	85.82	98.77	98.63	88.52	99.04	98.71
18	77.30	97.12	96.84	80.96	97.67	96.98
24	68.94	94.83	94.36	73.33	95.71	94.59
30	61.10	91.98	91.30	65.99	93.22	91.61

TABLE 7. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.000033/h$ and $\lambda_s = 20\ \lambda_d$.

Time (Months)	$\lambda_d = 5\ \lambda_t$			$\lambda_d = 10\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	96.92	99.85	99.91	98.28	99.92	99.93
12	93.35	99.40	99.67	95.94	99.65	99.74
18	89.31	98.63	99.26	93.01	99.16	99.42
24	84.93	97.53	98.68	89.58	98.41	98.94
30	80.31	96.11	97.94	85.78	97.39	98.33

TABLE 8. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.000033/h$ and $\lambda_s = 20\ \lambda_d$.

Time (Months)	$\lambda_d = 20\ \lambda_t$			$\lambda_d = 30\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	98.97	99.95	99.95	99.20	99.96	99.95
12	97.27	99.78	99.78	97.71	99.82	99.80
18	94.92	99.43	99.50	95.55	99.52	99.52
24	92.02	98.87	99.08	92.82	99.01	99.12
30	88.67	98.05	98.53	89.63	98.27	98.60

will be "0" since there are four "0s" in the input and only three "1s". Alternatively, the input can be 0110011 and the output will be "1". The system will tolerate the failure of $\leq$3 modules. To be able to apply DFX to a failed module (to repair the module in case the problem was due to a soft failure), it is necessary to identify the failed module. This is why each module output is compared to the output of the majority voter via an XOR gate. A "1" at the output of an

TABLE 9. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.000086$/h and $\lambda_s = 3\lambda_d$.

Time (Months)	$\lambda_d = 5\ \lambda_t$			$\lambda_d = 10\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	56.22	94.47	96.62	71.50	96.63	97.28
12	28.97	82.11	87.88	46.76	87.75	89.63
18	14.46	67.98	76.58	29.68	76.15	79.16
24	7.15	54.71	64.86	18.66	64.05	67.85
30	3.59	43.33	53.91	11.75	52.75	56.98

TABLE 10. Reliability (%) for modified 6Mr, modified7Mr and modified Triple duplex using $\lambda_s = 0.000086$/h and $\lambda_s = 3\ \lambda_d$.

Time (Months)	$\lambda_d = 20\ \lambda_t$			$\lambda_d = 30\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	80.65	97.85	97.63	83.96	98.28	97.75
12	59.47	91.30	90.58	64.43	92.62	90.91
18	42.61	81.87	80.60	48.08	84.10	81.10
24	30.26	71.29	69.57	35.56	74.26	70.17
30	21.47	60.77	58.77	26.28	64.21	59.41

TABLE 11. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.000086$/h and $\lambda_s = 20\ \lambda_d$.

Time (Months)	$\lambda_d = 5\ \lambda_t$			$\lambda_d = 10\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	90.94	98.97	99.44	94.22	99.38	99.56
12	79.28	95.76	97.77	84.91	97.13	98.18
18	66.80	90.48	95.09	73.89	93.02	95.89
24	54.90	83.59	91.60	62.69	87.27	92.83
30	44.33	75.71	87.52	52.24	80.36	89.16

TABLE 12. Reliability (%) for modified 6Mr, modified 7Mr and modified Triple duplex using $\lambda_s = 0.000086$/h and $\lambda_s = 20\ \lambda_d$.

Time (Months)	$\lambda_d = 20\ \lambda_t$			$\lambda_d = 30\ \lambda_t$		
	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex	Mod. 6MR	Mod. 7MR	Mod. Triple Duplex
0	100	100	100	100	100	100
6	95.91	99.59	99.62	96.48	99.66	99.65
12	87.88	97.84	98.40	88.89	98.08	98.47
18	77.72	94.36	96.30	79.05	94.82	96.44
24	67.01	89.25	93.45	68.52	89.93	93.67
30	56.73	82.90	90.00	58.32	83.78	90.29

XOR gate indicates that the module output connected to this XOR is incorrect and a DFX action is initiated to attempt repairing the module.

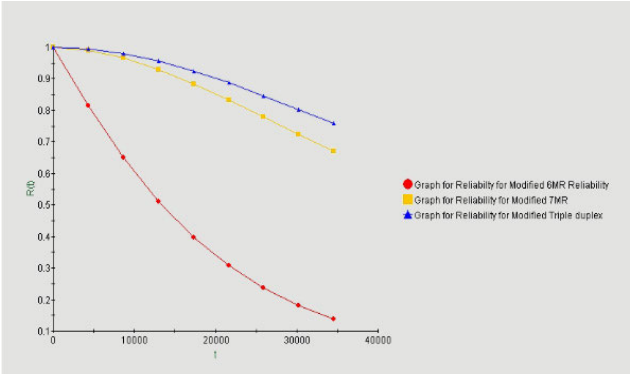


FIGURE 8. Reliability for modified 6MR, modified 7MR and modified Triple duplex based on Table 5.

Regarding the modified Triple Duplex voter (shown in Fig. 2), it consists of three modules: A normal TMR voter (N-V), and Enhanced TMR voter (E-V) and three XOR gates. The N-V voter is a conventional majority voter with three inputs; it produces the majority of the three inputs which is V0 [11]. The E-V voter is similar to the one described in [35]; it produces the correct output (V1) and another two outputs A1A0 to indicate the identity of the failed module. A XOR gate is used to compare the outputs of the two modules in a pair. The outputs of these three XOR gates along with A1A0, determine the pairs requiring a DFX action.

The modified 7MR voter and modified Triple Duplex voter were implemented using Xilinx Vivado tools targeting Kintex7, 7k410tfbg676 device. Table 13 shows the resource utilization for the voters.

V. DISCUSSION

As shown in Tables 1 through 12, the modified 6MR has the lowest reliability in all cases compared to the modified 7MR and the modified Triple Duplex.

The reliability of the modified Triple Duplex architecture is better than the reliability of the modified 7MR one in most cases as shown in Tables 1, 3, 5, 7, 8, 9, 11 and 12. The reliability of the modified 7MR architecture is better than that of the modified Triple Duplex one in a few cases as shown in Tables 2, 6 and 10. The reliability of both techniques are approximately equal in Table 4.

Based on the parameter values used above, it can be concluded that the modified Triple Duplex technique is always better than the modified 7MR one in case $\lambda_d < 20\ \lambda_t$ whatever the value of λ_s and whatever the ratio between λ_s and λ_d. If λ_s increases (module size increases), the improvement in reliability of the modified Triple Duplex architecture increases; for example, the reliability of the modified Triple Duplex improves from 99.66% to 99.81% in case $\lambda_s = 0.00002$/hr and improves from 94.47% to 96.62% in case $\lambda_s = 0.000086$/hr.

In case of $\lambda_d \geq 20\ \lambda_t$, it will depend on the ratio between λ_s and λ_d. If $\lambda_s = 3\ \lambda_d$, the modified 7MR is always better

TABLE 13. Implementation results of modified 7MR and modified Triple Duplex voters.

	Modified 7MR Voter	Modified Triple Duplex Voter
Slices LUTs	2	3
Slices FFs	8	11
BRAM (18kb each)	0	0

than the modified Triple Duplex. If $\lambda_s = 20\ \lambda_d$, the modified Triple Duplex is always better than the modified 7MR.

As shown in Table 13, the modified Triple Duplex voter has a higher cost than the modified 7MR one since the modified Triple Duplex uses more LUTs than the modified 7MR and consumes more power than the modified 7MR.

It is important to distinguish between architecture reliability and voter reliability. Since the voter is a single point of failure in this work, system reliability is equal to the product of the reliability of the fault-tolerant architecture (modified Triple Duplex or modified 7MR) and the voter reliability. However, since the voter usually requires much less resources than a single module, its reliability is usually assumed to be equal to 100% [11], [12] and system reliability is approximately equal to the fault-tolerant architecture reliability. This is the approach taken in this work.

In Table 13, voter resources are compared. The modified 7MR voter requires less resources than the modified Triple Duplex voter. In Table 14, and assuming that the redundant module is a picoblaze processor (as in section IV-A), resources required by the modified 7MR architecture are compared to those required by the modified Triple Duplex architecture from the point of view of module resources.

Table 15 combines the data in Tables 13 and 14 and shows the resources required by the complete architecture, i.e., redundant modules in addition to the voter. It is observed that the savings in slice LUTs, Slice FFs and BRAM are all around 14%. Also note that the resources required by any of the two voters is lower that those required by six or seven picoblaze processor. Even though the modified 7MR voter uses slightly less resources than the modified Triple Duplex voter, the main saving comes from going from 7 redundant modules to only 6, taking into account that voter resources are negligible with respect to resources utilized by the redundant module. In addition, both voter reliabilities are assumed to be equal to 100%; hence, only the architecture reliabilities are compared in Tables 1 through 12 and it can be seen that the modified Triple Duplex architecture has a higher reliability than the modified 7MR architecture in most of the cases considered in this work. This is a counter intuitive result.

From the power consumption point of view, assume that the picoblaze is performing a simple space application such as attitude control for example. Furthermore, assume that the picoblaze produces an output once per second (i.e., 1Hz) [36]. Simulations showed that the dynamic power consumption of the voters at 1Hz is negligible. Therefore, Table 15 only shows the dynamic power consumed by seven picoblaze processors (modified 7MR) and six picoblaze processors (modified Triple Duplex).

TABLE 14. Implementation results of 6 Picoblaze and 7 Picoblaze modules.

	6 Picoblaze	7 Picoblaze
Slices LUTs	624	728
Slices FFs	524	610
BRAM (18kb each)	6	7

TABLE 15. Implementation results of complete modified 7MR and modified Triple Duplex using Picoblaze.

	Mod. Triple Duplex	Mod. 7MR	Savings (%)
Slices LUTs	627	730	14.11
Slices FFs	535	618	13.43
BRAM (18kb each)	6	7	14.29
Dynamic Power @ 100MHz	0.020	0.024	16.67

TABLE 16. Implementation results of one Microblaze.

Slices LUTs	1286
Slices FFs	994
BRAM (36kb each)	2
Static power (W)	0.188
Dynamic power (W) @ frequency=100 MHz	0.118
Total on-chip power (W)	0.305

From another point of view, and in the context of space applications, consider for example an AI-based system running on a microblaze (more suitable for an AI-based application than a picoblaze) for imaging or satellite collision avoidance; let this system produce one output per second and let the microblase operate at a frequency of 100MHz. Table 16 shows the resources and power consumptions of one microblase processor. Regarding the voters, the dynamic power (at a frequency of 1Hz) was found to be neglegible.

VI. CONCLUSION

Recently, many applications are implemented using SRAM-based FPGAs such as space, biomedical, automotive applications, etc. These FPGAs are vulnerable to radiation causing SEUs and MEUs (DEUs and TEUs) in space applications as the space environment is a relatively harsher environment.

In this article, the modified 6MR, modified 7MR and modified Triple Duplex techniques were studied and their reliability was calculated using the SHARPE tool. It was observed that the modified 6MR has the lowest reliability and the reliability of the modified 7MR and modified Triple Duplex depends on the ratio between the SEU, DEUs and TEUs rates. The modified 7MR architecture has the highest reliability in a few scenarios but the modified Triple Duplex architecture has the highest reliability in most scenarios studied in this article. This is a counter intuitive result since the normal rule is that the greater the redundancy, the higher the reliability. However, it is important to remember that the conclusions reached in this article depend on the modules and failure rates used; the contribution of this research is

that it was proven that there may be situations where six modules would be more reliable than seven modules, besides the savings in resources and in power consumption.

Finally, the modified 7MR and the modified Triple Duplex voters were implemented using the Vivado tool with the Kintex7, 7k410tfbg676 device. It was shown that the modified Triple Duplex voter consumes more resources and power than 7MR voter. Note however, that these resources are much lower than those used by one module.

FUTURE WORK

Suggested future work consists of studying other fault tolerance techniques to mitigate SEU, DEUs, TEUs and hard errors, targeting higher reliability, less resources, and lower power consumption.

REFERENCES

[1] P. Rech, J.-M. Galliere, P. Girard, A. Griffoni, J. Boch, F. Wrobel, F. Saigne, and L. Dilillo, "Neutron-induced multiple bit upsets on two commercial SRAMs under dynamic-stress," *IEEE Trans. Nucl. Sci.*, vol. 59, no. 4, pp. 893–899, Aug. 2012.

[2] S. Shreejith, S. A. Fahmy, and M. Lukasiewycz, "Reconfigurable computing in next-generation automotive networks," *IEEE Embedded Syst. Lett.*, vol. 5, no. 1, pp. 12–15, Mar. 2013.

[3] F. Kastensmidt and P. Rech, *FPGAs and Parallel Architectures for Aerospace Applications. Soft Errors and Fault-tolerant Design*. Cham, Switzerland: Springer, 2016.

[4] C. Bolchini and C. Sandionigi, "Fault classification for SRAM-based FPGAs in the space environment for fault mitigation," *IEEE Embedded Syst. Lett.*, vol. 2, no. 4, pp. 107–110, Dec. 2010.

[5] G. I. Alkady, N. A. El-Araby, M. B. Abdelhalim, H. H. Amer, and A. H. Madian, "Dynamic fault recovery using partial reconfiguration for highly reliable FPGAs," in *Proc. 4th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2015, pp. 56–59.

[6] T. S. Nidhin, A. Bhattacharyya, R. P. Behera, T. Jayanthi, and K. Velusamy, "Understanding radiation effects in SRAM-based field programmable gate arrays for implementing instrumentation and control systems of nuclear power plants," *Nucl. Eng. Technol.*, vol. 49, no. 8, pp. 1589–1599, Dec. 2017, doi: 10.1016/j.net.2017.09.002.

[7] Z. Gao, J. Zhu, R. Han, Z. Xu, A. Ullah, and P. Reviriego, "Design and implementation of configuration memory SEU-tolerant Viterbi decoders in SRAM-based FPGAs," *IEEE Trans. Nanotechnol.*, vol. 18, pp. 691–699, 2019.

[8] A. Fernandez-Alvarez, M. Portela-Garcia, M. Garcia-Valderas, C. Lopez-Ongil, S. S. Ibáñez, and S. E. Meana, "Assessing SET sensitivity of mixed-signal circuits at early design stages," in *Proc. 34th Conf. Design Circuits Integr. Syst. (DCIS)*, Nov. 2019, pp. 1–6.

[9] V. Pouget, "Laser testing for single-event effects: Basics and use cases," in *Proc. Eur. Conf. Radiat. Effects Compon. Syst.*, Sep. 2021, pp. 1–38.

[10] M. Ceschia, M. Bellato, M. Menichelli, A. Papi, J. Wyss, and A. Paccagnella, "Heavy ion irradiation of SRAM-based FPGA's," in *Proc. Mil. Aerosp. Appl. Program. Devices Technol. (MAPLD) Conf.*, 2001, pp. 1–7.

[11] D. P. Siewiorek and R. S. Swarz, *Reliable Computer Systems Design Evaluation*. Natick, MA, USA: A. K. Peters, 1998.

[12] K. S. Trivedi and A. Bobbio, *Reliability and Availability Engineering-Modeling, Analysis, and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[13] C. Frenkel, J.-D. Legat, and D. Bol, "A partial reconfiguration-based scheme to mitigate multiple-bit upsets for FPGAs in low-cost space applications," in *Proc. 10th Int. Symp. Reconfigurable Commun.-Centric Syst. Chip*, Jun. 2015, pp. 1–7.

[14] B. Shokry, D. G. Mahmoud, H. H. Amer, M. Shatta, G. I. Alkady, R. M. Daoud, I. Adly, M. N. Shaker, and T. Refaat, "Work-in-progress: Triple event upset tolerant area-efficient FPGA-based system for space applications and nuclear plants," in *Proc. 16th IEEE Int. Conf. Factory Commun. Syst. (WFCS)*, Apr. 2020, pp. 1–4.

[15] *Vivado Design Suite User Guide. Dynamic Function Exchange*, Xilinx, San Jose, CA, USA, May 2023.

[16] C. Gauer, B. J. LaMeres, and D. Racek, "Spatial avoidance of hardware faults using FPGA partial reconfiguration of tile-based soft processors," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, USA, Mar. 2010, pp. 1–11.

[17] J. W. Valvano, *Embedded Systems: Introduction to ARM Cortex-M Microcontrollers*. Scotts Valley, CA, USA: CreateSpace Independ. Publishing Platform, 2012.

[18] (2020). *DUKE Sharp Portal*. Accessed: Nov. 20, 2020. [Online]. Available: http://sharpe.pratt.duke.edu

[19] J. Sun, Y. Wang, P. Liu, S. Wen, and Y. Wang, "Memristor-based neural network circuit with multimode generalization and differentiation on Pavlov associative memory," *IEEE Trans. Cybern.*, vol. 53, no. 5, pp. 3351–3362, May 2023.

[20] *Device Reliability Report*, Xilinx, Scotts Valley, CA, USA, May 2018.

[21] A. Dutta and N. A. Touba, "Multiple bit upset tolerant memory using a selective cycle avoidance based SEC-DED-DAEC code," in *Proc. 25th IEEE VLSI Test Symmposium (VTS)*, Berkeley, CA, USA, May 2007, pp. 349–354.

[22] S. Radhakrishnan, T. Nirmalraj, S. Ashwin, V. Elamaran, and R. K. Karn, "Fault tolerant carry save adders—A NMR configuration approach," in *Proc. Int. Conf. Control, Power, Commun. Comput. Technol. (ICCPCCT)*, Kannur, India, Mar. 2018, pp. 210–215.

[23] M. N. Shaker, A. Hussien, G. I. Alkady, H. H. Amer, and I. Adly, "Mitigating the effect of multiple event upsets in FPGA-based automotive applications," in *Proc. 8th Medit. Conf. Embedded Comput. (MECO)*, Budva, Montenegro, Jun. 2019, pp. 1–4.

[24] M. N. Shaker, A. Hussien, G. I. Alkady, H. H. Amer, and I. Adly, "FPGA-based reliable fault secure design for protection against single and multiple soft errors," *Electronics*, vol. 9, no. 12, p. 2064, Dec. 2020, doi: 10.3390/electronics9122064.

[25] *Xilinx Partial Reconfiguration User Guide*, Xilinx, Scotts Valley, CA, USA, Apr. 2013.

[26] B. Johnson, *Design and Analysis of Fault-tolerant Digital Systems*. Boston, MA, USA: Addison-Wesley, 1989.

[27] Y. Wang, J. Xu, Y. Han, H. Li, and X. Li, "DeepBurning: Automatic generation of FPGA-based learning accelerators for the neural network family," in *Proc. 53rd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2016, pp. 1–6.

[28] F. M. Siddiqui, M. Russell, B. Bardak, R. Woods, and K. Rafferty, "IPPro: FPGA based image processing processor," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Oct. 2014, pp. 1–6.

[29] M. Zakaria, A. S. Mabrouka, and S. Sarhan, "Artificial neural network: A brief overview," *Int. J. Eng. Res. Appl.*, vol. 4, no. 2, pp. 7–12, Feb. 2014.

[30] N. A. Harward, M. R. Gardiner, L. W. Hsiao, and M. J. Wirthlin, "Estimating soft processor soft error sensitivity through fault injection," in *Proc. IEEE 23rd Annu. Int. Symp. Field-Programmable Custom Comput. Mach.*, Vancouver, BC, Canada, May 2015, pp. 143–150.

[31] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, and G. Furano, "High-performance embedded computing in space: Evaluation of platforms for vision-based navigation," *J. Aerosp. Inf. Syst.*, vol. 15, no. 4, pp. 178–192, Apr. 2018, doi: 10.2514/1.i010555.

[32] D. S. Lee, M. Wirthlin, G. Swift, and A. C. Le, "Single-event characterization of the 28 nm Xilinx Kintex-7 field-programmable gate array under heavy ion irradiation," in *Proc. IEEE Radiat. Effects Data Workshop (REDW)*, Paris, France, Jul. 2014, pp. 1–5.

[33] J. Tonfat, F. Lima Kastensmidt, P. Rech, R. Reis, and H. M. Quinn, "Analyzing the effectiveness of a frame-level redundancy scrubbing technique for SRAM-based FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 6, pp. 3080–3087, Dec. 2015.

[34] L. A. Tambara, F. L. Kastensmidt, N. H. Medina, N. Added, V. A. P. Aguiar, F. Aguirre, E. L. A. Macchione, and M. A. G. Silveira, "Heavy ions induced single event upsets testing of the 28 nm Xilinx zynq-7000 all programmable SoC," in *Proc. IEEE Radiat. Effects Data Workshop (REDW)*, Boston, MA, USA, Jul. 2015, pp. 1–6.

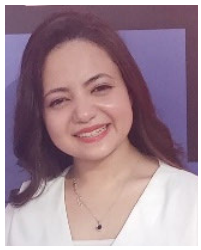
[35] D. G. Mahmoud, G. I. Alkady, H. H. Amer, R. M. Daoud, I. Adly, Y. Essam, H. A. Ismail, and K. N. Sorour, "Fault secure FPGA-based TMR voter," in *Proc. 7th Medit. Conf. Embedded Comput. (MECO)*, Budva, Montenegro, Jun. 2018, pp. 1–4.

[36] P. Acquatella, E.-J. Van Kampen, and Q. P. Chu, "A sampled-data form of incremental nonlinear dynamic inversion for spacecraft attitude control," in *Proc. AIAA SCITECH Forum*, Jan. 2022, p. 0761, doi: 10.2514/6.2022-0761.

MANAR N. SHAKER received the B.S. and M.S. degrees from the Electronics and Communications Engineering Department, Cairo University, Egypt, in 2010 and 2014, respectively. She is currently pursuing the Ph.D. degree. Since 2014, she has been with the Electronics and Communication Engineering Department, The American University in Cairo, Egypt, as a Teaching Assistant. She is also a Research Assistant with the SEAD Group. From 2018 to 2020, she was with the Electronics and Communications Engineering Department, Cairo University, as a Research Assistant. Her research interests include FPGAs, fault modeling, and reliability modeling.

AHMED HUSSIEN was born in Cairo, Egypt, in 1960. He received the B.Sc. degree in electronics and electrical communications engineering from Cairo University, Giza, Egypt, in 1983, and the M.Sc. and Ph.D. degrees from the Electronics and Electrical Communications Engineering (EECE) Department, Faculty of Engineering, Cairo University, in 1987 and 1992, respectively. He has been a Faculty Member with the EECE Department, Faculty of Engineering, Cairo University, since 1983, where he is currently a Professor. He has been the Vice Director of the Design Laboratory for Electronics and Communication Systems (DLECS), since 2006. He teaches several courses on analog and digital electronics and signal processing at Cairo University and other universities and institutes in Egypt. He has been in the electronics and communications industry for over 32 years of hands-on practical knowledge. His research interests include signal processing, embedded systems, and FPGA-based systems using a broad range of tools and technologies on diverse types of platforms with a solid track record of supervising many mega projects and experience in electronic design and technology integration.

HASSANEIN H. AMER (Life Member, IEEE) received the B.Sc. degree in electronics engineering from Cairo University, in 1978, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, CA, USA, in 1983 and 1987, respectively. He founded the SEAD Group, in 2003. He is currently a Professor with the Electronics and Communications Engineering Department, The American University in Cairo, Egypt. He is also the Founding Chair of the Electronics and Communications Engineering Department. His research interests include reliability and testing of digital and mixed-signal circuits, reliability modeling, fault modeling in VLSI, networked control systems, precision agriculture, vehicular embedded systems, and wireless sensor networks.



BEATRICE SHOKRY received the B.Sc. degree in electronics and communications engineering (computer science) from The American University in Cairo (AUC), in 2023. She has been a member of the SEAD Research Group, since 2018. Her research interests include fault tolerance, FPGAs, hardware security, and distributed systems.