VOL. 20, NO. 2-2, 2021, 98-102
www.elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

# Fault Injection Test on Error Mitigated Circuit of Partial TMR in FPGA

**Muhamad Aidil Azri Salleh**, **Ili Shairah Abdul Halim*** and **Siti Lailatul Mohd Hassan**

Integrated Sensors Research Group (DERIA), School of Electrical Engineering, College of Engineering, Universiti Teknologi MARA Shah Alam, Selangor

*Corresponding author: shairah@uitm.edu.my

**Abstract:** SRAM-based Field Programmable Gate Arrays (FPGAs) are vulnerable to Single Event Upsets (SEU), affecting the reliability of embedded system devices. Triple Modular Redundancy (TMR) has been proposed in the literature to mitigate error within the configuration memory of an SRAM-based FPGA. In this paper, TMR and Partial TMR were compared through an FPGA development board of DE1- SoC (System on Chip). Fault modules are inserted in the architecture to emulate fault injection on the circuit under test (CUT). The result from the fault injection test shows that TMR gave a 50% passing rate, whereas Partial TMR gave an additional 16.67% passing rate on top of normal TMR.

**Keywords:** Fault-Tolerant, Partial Modular Redundancy, Redundancy, Three Modular Redundancy.



*Article History: received 25 May 2021; accepted 12 June 2021; published 15 September 2021.*

## 1. INTRODUCTION

The International Roadmap for Devices and Systems has reported the technology issues in semiconductor size-reduction causing challenges in reliability [1]. Therefore the reliability [2] has become vital for integrated circuit designers and semiconductor manufacturers in reducing the error probability throughout the system operation. In recent research, fault emulation and fault modelling are methods used to analyze the reliability and performance of electronics devices. A Reconfigurable system such as Field-Programmable Gate Array (FPGA) is a platform that is commonly to emulate fault injection [3]–[6] of high energy particles that hit on the circuit level. FPGA is sensitive to this radiation and causing a Single Event Upset (SEU) [7]. SEU specifically is a change of state caused by one single ionizing particle striking a sensitive node in a microelectronic device. The state change results from the free charge created by ionization in or close to an important node of a logic element such as memory bit. While SEU does not damage the transistors or circuits or its functionality permanently, unlike other Single Event Effects (SEE), it still affects the device's accuracy to give a precise output, reducing its reliability.

Noticing these occurrences can bring inaccuracy to the device performances; therefore, fault tolerance circuits such as hardware redundancy increase its reliability and accuracy. Triple modular redundancy (TMR) [8] and partial three modular redundancy (Partial TMR) [9]–[11] via approximate logic circuit are proposed to solve the FPGA drawback of SEU. These methods of redundancies can increase the reliability of the device and obtain a high probability of accurate output. TMR is commonly known for its ability to mask faults by triplicating the logic and adding majority voters. Partial TMR is then proposed by using an approximate logic circuit as it can perform a different yet nearly related logic function, so it can be used for fault detection or error masking where the approximate logic circuit overlaps with the original circuit. Therefore, Partial TMR is low in granularity, scalable and can provide an optimal balance between resource usage and circuit reliability.

The objective of this work is to design and compare the redundancy module, TMR and Partial TMR, for its effectiveness and its capability of fault tolerance. Both redundancy modules are using LGSynth91 Benchmark Circuit for the golden circuit module and compared with the circuit-under-test (CUT) module. The design process is done by using Quartus Prime Lite Edition software that uses Verilog language implemented on the DE1-SoC (System on Chip) development board.

Figure 1 shows the overall framework in this work. For the TMR module, the test input vector will send input data to the golden and the CUT simultaneously. Then the output will be compared in the compare module. The comparison process is when both modules produce the same output, the test will be a pass which means that no fault has occurred during the test. Otherwise, when the modules produce different outputs, the test will be considered a failure and the device has a fault.

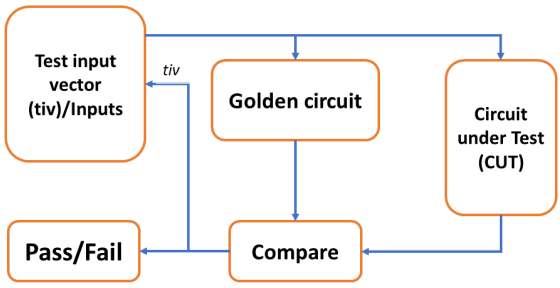


Figure 1. Fault injection test overall framework

## 2. ERROR MITIGATED CIRCUIT DESIGN ARCHITECTURE

This section describes the detailed architecture of the benchmark circuit, TMR and Partial TMR. LGSynth91 Benchmark Circuit, C17 is used as a benchmark circuit for both TMR and Partial TMR.

### 2.1 Benchmark Circuit

Figure 2 shows the circuit diagram for the C17 circuit consisting of five inputs (N1, N2, N3, N6 and N7), two outputs (N22 and N23), including six two-inputs NAND gates.

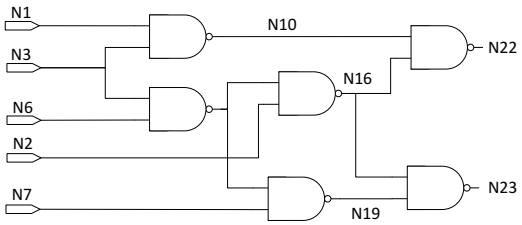


Figure 2. Circuit diagram of benchmark circuit C17

### 2.2 Three Modular Redundancy (TMR)

TMR is the most common method in error mitigation on reconfigurable systems. The redundancy is able to mask the error through the majority vote of correct output. As shown in Figure 3, TMR triplicates the same modules and connected to a voter module. The same input data is sent to all triplicated modules. If two or more redundant modules are in the fault state, the fault state will be voted out. If one or none of the redundant modules is in a fault state, the voter will vote the majority output and produce a correct state.

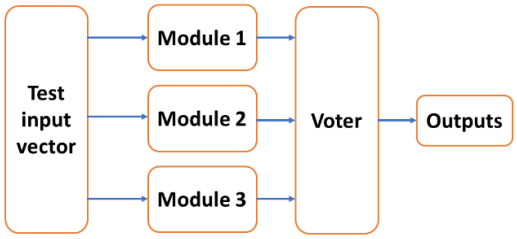


Figure 3. Block diagram of TMR

### 2.3 Partial TMR

The architecture of Partial TMR is similar to TMR's architecture and also considered as a fault-tolerant modular redundancy. As shown in Figure 4, while TMR has three duplicate modules of the C17 circuit, Partial TMR also has three redundant modules but with different input means. Although the three redundant modules still use C17 as their base of architecture, one of the three redundant modules is from the original C17 circuit; the other two applies the unique architecture of approximation logic circuit.

The concept of approximation logic circuit provides a systematic framework for the implementation of fault-tolerant combinational logic circuits. Referring to the block diagram in Figure 4, the original circuit and the approximation circuits (over and under approximation circuit) produce the same output value. Like the concept of TMR, only one fault can occur at a time its effect will be masked [10].

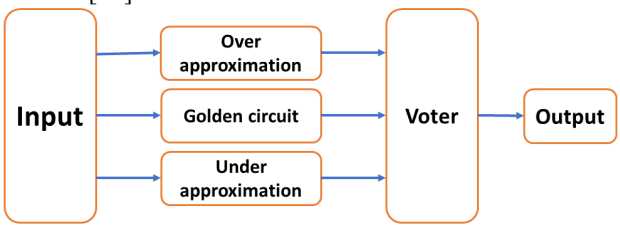


Figure 4. Block diagram of Partial TMR

Figure 5(a) shows the original Benchmark C17 schematic circuit, and Figure 5(b) and Figure 5(c) are the expected over and under approximation circuit, respectively. While the Benchmark C17 circuit is the same as previously mentioned, the over-approximation circuit's input N6 has been set to high or '1' and thus, the circuit has been simplified as in Figure 5(b). The input N3 and N6 for under approximation schematic is set to high; thus, the circuit in Figure 5(c) has been simplified and reducing the gate usage to only four gates instead of six gates in the original C17 circuit.

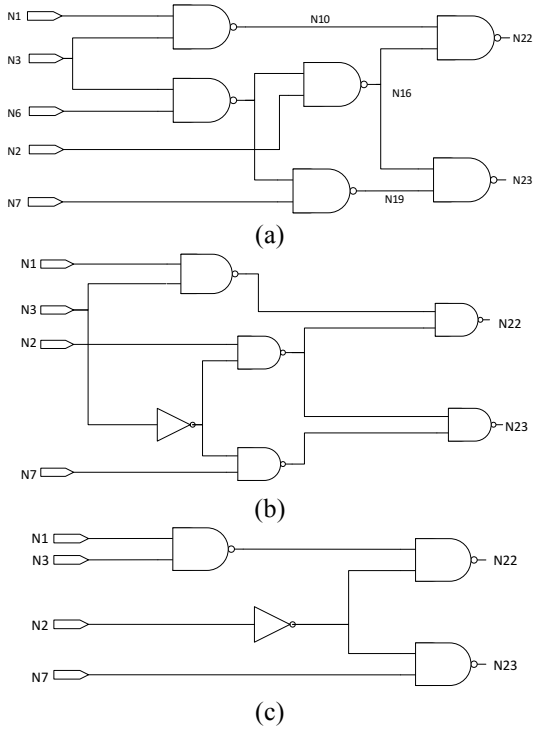


Figure 5. (a) Original C17 circuit, (b) Over-approximation circuit and (c) Under-approximation circuit design

## 3. FAULT INJECTION TEST EXPERIMENT

The fault injection test is performed by adding the fault module on the CUT in the main framework of Figure 1. The main purpose of this fault module is to emulates the SEU that changes the output (wrong output) of the benchmark circuit. Refers to Figure 6, each module if the TMR is connected to a fault module. These fault modules are controlled with external hardware in performing faulty module. The same fault module is inserted for Partial TMR during the fault injection test experiment. The same test input vector (tiv) is sent to all three modules. The same majority output will be voted and send to output before being compared to the golden circuit. Then, the output from the compare module can determine whether the run is a pass or a failure.

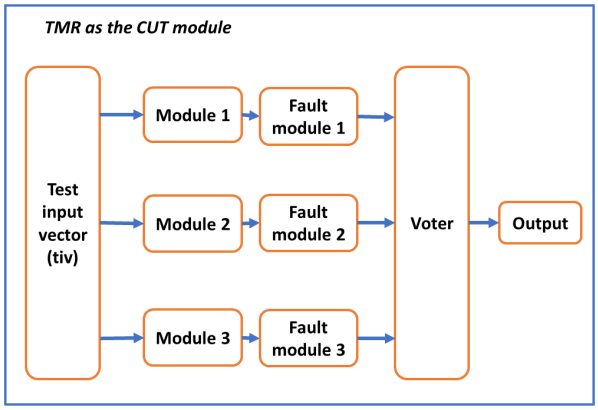


Figure 6. Fault setup block diagram for TMR module

## 4. RESULTS AND DISCUSSION

### 4.1 Register Transfer Level (RTL)

RTL viewer is a tool in Quartus Prime software that displays the designed modules in block diagrams. As in Figure 7, the RTL of the overall architecture for fault injection test shows the interfacing of block modules explained in Section 3.

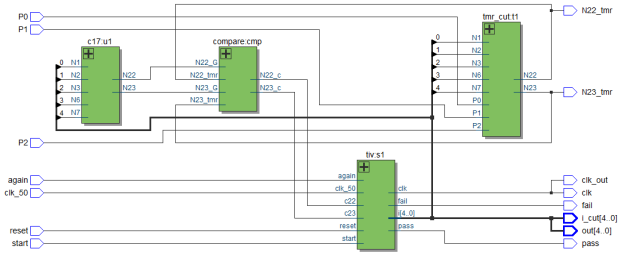

Figure 7. RTL schematic of the overall design for fault injection test with golden circuit and CUT module

First, the TMR module, *tmr_cut* as the CUT module, consists of seven inputs and eight outputs in total. *N22_tmr* and *N23_tmr* are the output results from the *tmr_cut*. *Clk_50* is external clocks to the Cyclone V DE1-SoC FPGA, and the 50MHz clock signals connected to the FPGA are used as clock sources for user logic in this work. For inputs *start*, *again* and *reset*, are assigned to do specific function in the simulation, such as, *start* is used to start the simulation and run the compilation, *again* is used to rerun the test, while *reset* is to reset all the given inputs and the produced outputs and goes into its initial condition. As for the final outputs that determine the result is the pass or fail outputs, LEDs on the FPGA boards are assigned together with the *i_cut* that displays the running inputs.

Figure 8 shows the logic gates that are used in the *c17* module, and Figure 9 shows the logic gates in *compare* module shown in the RTL viewer. The *compare* module will determine whether the results from *tmr_cut* are the same or not with *c17*. Lastly, *tiv* module is the test input vector module. This module will provide and run all the inputs for the test. This module will stop operation if all inputs are successfully tested and gives a pass result. It will also stop if the outputs from tmr_cut and c17 are not the same, thus providing a fail result.

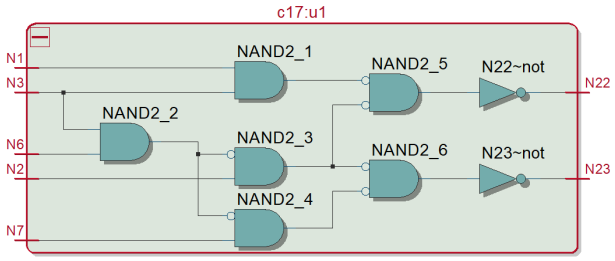


Figure 8. RTL schematic of c17 module.

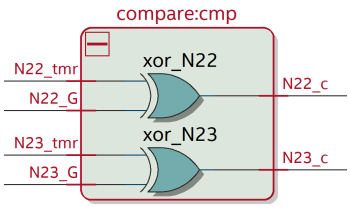


Figure 9. RTL schematic of c17 module

In Figure 10, the RTL schematic for the *tmr_cut* has five inputs, *N1, N2, N3, N6* and *N7* that supplied from the *tiv* module, three control inputs of the error module, *P0, P1,* and *P2,* that act as fault injection on the circuit, and two outputs, *N22* and *N23* indicates the output for the CUT module.

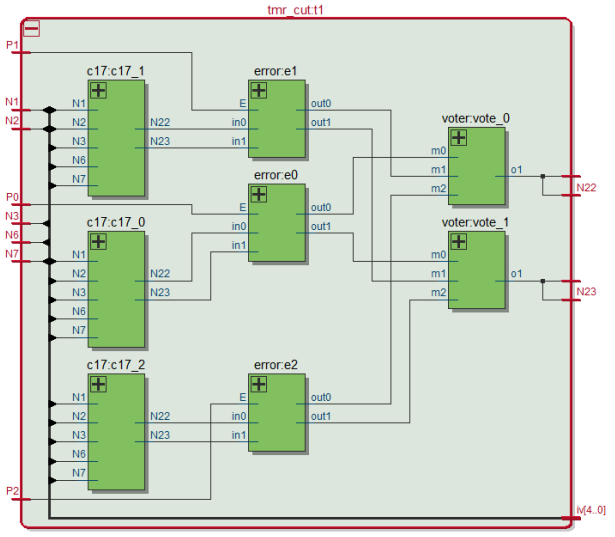


Figure 10. RTL schematic of CUT with TMR module, *tmr_cut*

For the output *iv* having five bits output data, it indicates the running five inputs. The two voter modules will take

the outputs from the error modules and vote for its majority output to become one final output for N22 and N23.

Figure 11 shows the RTL schematic for Partial TMR. Like the TMR module, the design is that it has the same types of inputs and outputs but with a slight alteration of its inputs. N1, N2, N3, N6, and N7 are the inputs for *main_gc*, which is the golden circuit module for Partial TMR.

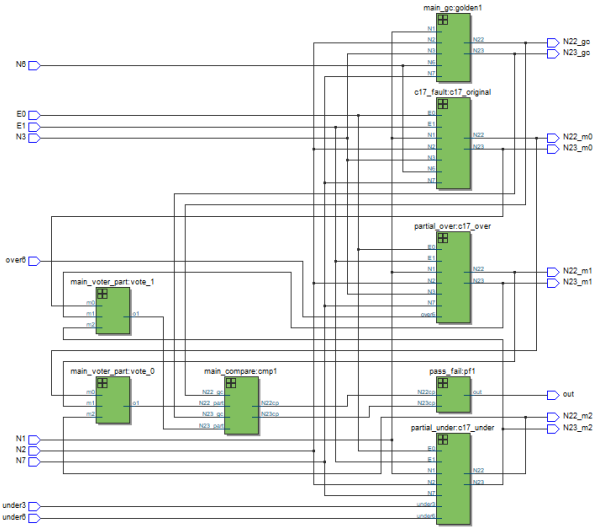


Figure 11. RTL schematic of CUT with Partial TMR

For the *c17_fault* module, it is the original C17 circuit, having the same inputs and outputs; the only difference is it includes the error modules, with *E0* and *E1* as its inputs. For its approximation modules, *c17_over* and *c17_under* have different inputs but having the same outputs as the *main_gc* module. For the *c17_over* module, the inputs are *N1, N2, N3, over6* and *N7*, together with the error inputs *E0* and *E1*. For *c17_under*, the inputs are *N1, N2, under3, under6, N7,* with two error inputs *E0* and *E1*. In *c17_over, over6* is the replacement of the input N6, while in *c17_under,* it has two inputs replaced which are *under3* and *under6*, which is initially N3 and N6, respectively.

### 4.2 Resource Utilization

Table 1 shows the logic utilization of TMR and Partial TMR obtained from the Quartus Prime software. The logic usage of TMR is four times the Partial TMR. This is due to applying the approximation circuit in Partial TMR that can reduce the number of gates.

Table 1. Resource utilization of TMR and Partial TMR

| **Devices Resources** | **Number of Resources** | |
|---|---|---|
| | ***TMR*** | ***Partial TMR*** |
| Logic utilization | 32 / 32,070 | 8 / 32,070 |

### 4.3 Hardware Result

Figure 12 shows an example output on the FPGA board where the test is pass. LEDR8 is light up to indicate the test is pass. LEDR2 to LEDR6 show that test input vector condition. The LEDs will light up when the inputs are high state (1); otherwise, the LEDs will be turned off. The switches assigned were SW0 as the start switch and three switches, SW7, SW8, and SW9, as the three error modules input. The fault injection test will pass when all the test input vector successfully run from 00000 to 11111 bits. As in Figure 12, SW9 is turned ON, which means one error module is activated. The result shows a pass (pass LED ON) as the voter capable of masking that one error.

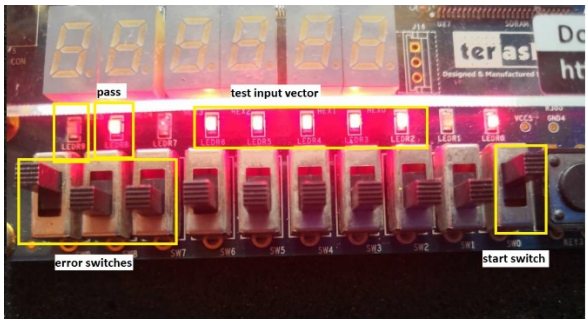


Figure 12. Pass condition on FPGA board for TMR module

Figure 13 shows the output on the FPGA board having a fail test. LEDR9 is light up to indicate the test is failed. This is due to two error modules is activated by the error switches. LEDR2 to LEDR6 show the test input vector condition. The LEDs will light up when they are in a high state (1); otherwise, they are in a low state (0). The fault injection test will fail when the output produced from the TMR module is not the same as the golden circuit module. Test input vector will stop at the current input where the TMR module become faulty and thus fail. With a total of 8 inputs (five tiv input and three error switches), a total of $2^8$ or 256 possibilities of test output. The result shows 50% passing rates of the fault injection test on the TMR circuit.

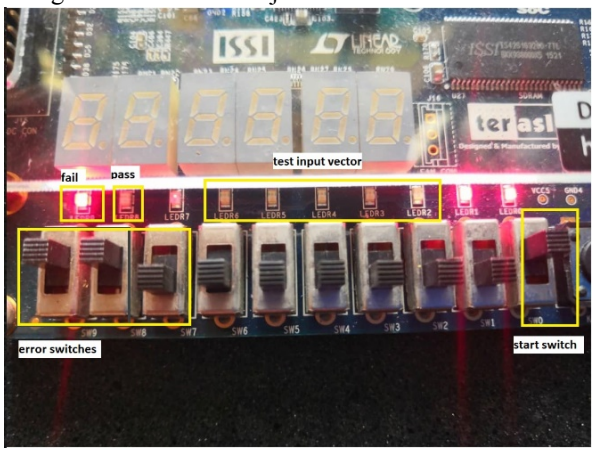


Figure 13. Fail condition on FPGA board for TMR module

A pass result of Partial TMR on the FPGA board is shown in Figure 14. For Partial TMR, specific inputs must be in a high state (1) because only some of the inputs can be compared with the golden circuit due to the logic approximation architecture design. For this setup, inputs *N3* and *N6* with the assigned switches, SW2 and SW3, respectively, are for modules *c17_original* and *golden1*. Input *over6*, assigned switch SW5 is the replacement input of *N6* for the *c17_over* module. Inputs *under3* and *under6* are the replacements for *N3* and *N6*, and their designated switches were SW6 and SW7, respectively. Error modules input for *E0* and *E1* are set to SW8 and SW9, respectively. For LED assignments, the pass or fail assignment is assigned to LED0. Whereas LED1 and LED2 are the

outputs from the *golden1* module that is the golden circuit. LED3 and LED4 are the outputs from *c17_original*, LED5 and LED6 are the outputs from the over-approximation module, and lastly LED7 and LED8 are the outputs from the under-approximation module.

Inputs of *N3*, *N6*, *over6*, *under3*, and *under6* were set to high (1) to compare Partial TMR and golden circuit output. As in Figure 14, N1 is set to high (1), and both error modules are set to low (0); hence the result shows a pass.

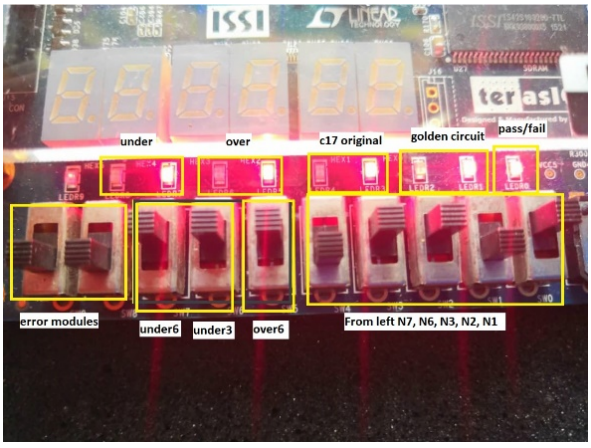


Figure 14. Partial TMR pass condition on FPGA board

For Figure 15, the fault injection test shows a fail for Partial TMR when one or all of the two redundant modules are set to high (1). Partial TMR produced different output compared with the golden circuit, concluding that the test is a fail as the LED turns OFF. With a total of 6 inputs (four tiv input and two error switches), a total of $2^6$ or 64 possibilities of test output. The result shows 66.7% passing rates of the fault injection test on the Partial TMR circuit.

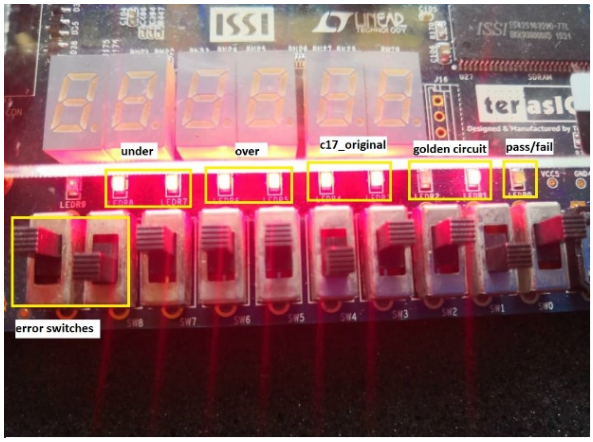


Figure 15. Partial TMR fail condition on FPGA board

## 5. CONCLUSION

In conclusion, two types of hardware redundancy for error mitigation purpose have been studied in this work. TMR and partial TMR design architecture were presented and tested its functionality. The two designs were also tested for their masking error capability using the fault injection test, where fault modules were inserted into the system. The results show that TMR can only work correctly if not more than one module is faulty. In comparison, Partial TMR has specific inputs that can be compared with the golden circuit, where it benefits to test on critical inputs. Partial TMR simplifies the golden circuit, and the modified version gives the same output as the golden circuit. TMR produces a 50% passing rate for 256 possibilities of error injections when the fault test injection takes place. Due to fewer resources reduced in Partial TMR, the number of possibilities been considered if 64 possibilities. The result shows that Partial TMR has an additional 16.67% passing rate on top of normal TMR.

## ACKNOWLEDGMENT

This material is based upon work supported by College of Engineering, Universiti Teknologi MARA.

## REFERENCES

[1] "International Roadmap for Devices and Systems: 2020," 2020.

[2] P. S. Ostler *et al.*, "SRAM FPGA Reliability Analysis for Harsh Radiation Environments," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 6, pp. 3519–3526, 2009.

[3] J. Tarrillo, J. Tonfat, L. Tambara, F. Lima, and R. Reis, "Multiple Fault Injection Platform for SRAM-based FPGA based on Ground-level Radiation Experiments," in *16th Latin-American Test Symposium (LATS)*, 2015, pp. 1–6.

[4] C.-A. Mao, Y. Xie, X. Wei, Y.-Z. Xie, and H. Chen, "FPGA-based fault injection design for 16K-point FFT processor," *J. Eng.*, vol. 2019, no. 21, pp. 7994–7997, 2019.

[5] F. Serrano, J. A. Clemente, and H. Mecha, "A Methodology to Emulate Single Event Upsets in Flip-Flops Using FPGAs through Partial Reconfiguration and Instrumentation," *IEEE Trans. Nucl. Sci.*, vol. 62, no. 4, 2015.

[6] T. Matsuzaki, K. Miyashita, K. Horiuchi, H. Shiratsuchi, I. S. A. Halim, and K. Mashiko, "Proposal of soft error injection method using two FPGA boards," *ICIC Express Lett.*, vol. 14, no. 6, pp. 555–562, 2020.

[7] M. Wirthlin, "High-reliability FPGA-based systems: Space, high-energy physics, and beyond," *Proc. IEEE*, vol. 103, no. 3, pp. 379–389, 2015.

[8] S. C. Anjankar and M. T. Kolte, "Fault Tolerant and Correction System Using Triple Modular Redundancy," *Int. J. Emerg. Eng. Res. Technol.*, vol. 2, no. 2, pp. 187–191, 2014.

[9] T. Arifeen, A. S. Hassan, and J. A. Lee, "A fault tolerant voter for approximate triple modular redundancy," *Electron.*, vol. 8, no. 3, 2019.

[10] A. J. Sanchez-Clemente, L. Entrena, R. Hrbacek, and L. Sekanina, "Error mitigation using approximate logic circuits: A comparison of probabilistic and evolutionary approaches," *IEEE Trans. Reliab.*, vol. 65, no. 4, pp. 1871–1883, 2016.

[11] A. J. Sanchez-Clemente, L. Entrena, and M. Garcia-Valderas, "Partial TMR in FPGAs Using Approximate Logic Circuits," *IEEE Trans. Nucl. Sci.*, vol. 63, no. 4, pp. 2233–2240, 2016.