Performance Analysis of Transient Fault-Injection and Fault-Tolerant System for Digital Circuits on FPGA

Sharath Kumar Y N[1]
Department of Electrical and Electronics Engineering
Dayananda Sagar College of Engineering, Bengaluru, India

Dinesha P[2]
Department of Electronics and Communication Engineering
Dayananda Sagar College of Engineering, Bengaluru, India

***Abstract*—A Fault-Tolerant System is necessary to improve the reliability of digital circuits with the presence of Fault Injection and also improves the system performance with better Fault Coverage. In this work, an efficient Transient Fault-Injection system (FIS) and Fault-Tolerant System (FTS) are designed for digital circuits. The FIS includes Berlekamp Massey Algorithm (BMA) based LFSRs, with fault logic followed by one – hot-encoder register, which generates the faults. The FTS is designed using Triple-Modular-Redundancy (TMR) and Dual Modular- Redundancy (DMR). The TMR module is designed using the Majority Voter Logic (MVL), and DMR is designed using Self-Voter Logic (SVL) for digital circuits such as synchronous and asynchronous circuits. The four different MVL approaches are designed in the TMR module for digital circuits. The FIS-FTS module is designed on Xilinx-ISE 14.7 environment and implemented on Artix-7 FPGA. The synthesis results include chip area, gate count, delay, and power are analyzed along with fault tolerance, and coverage for given digital circuits. The fault tolerance is analyzed using Modelsim-simulator. The FIS-FTS module covers an average of 99.17% fault coverage for both synchronous and asynchronous circuits.**

Keywords*—Digital circuits; transient fault; fault injection; fault tolerant; triple modular redundancy; dual modular redundancy; majority voter logic; self-voter logic

I. Introduction

Designing the electronic system by concerning the reliability and availability features are used in many critical applications like aerospace, military, transportation, and avionics. These electronics systems provide continuous support while performing real-time applications. If any attacks have occurred on these systems, it affected the overall system performance and led to failure. The fault injection is a process for estimating or evaluating the fault-tolerant system. There are many FIS methods are available based on hardware FIS, simulation-based FIS, and emulation based FIS. The hardware-based FIS injects high Laser beams or ion beams to circuits. The software-based FIS has static and dynamic approaches for analyzing the high computational overheads with high accuracy. The emulation based FIS has hardware reconfiguration, and circuits instrumentations approach. The dependability can tolerate system failures by providing better services, including threats, attributes, and means. Many available attributes include reliability, safety, confidentiality, security, integrity, maintainability, and availability creates dependability in any of the electronics systems. The faults, errors, failures are a chain of threats, which affect the system or component. The means have methods or techniques which include fault tolerance, fault preventions, fault forecasting, and fault removal [1-2].

The triple modular redundancy technique is used to protect logic circuits against soft errors. The original logic circuit is repeated three times, and the output is obtained from the majority voter. Each replicated circuit works independently from the other circuit. If an error occurs in one circuit, it is masked by TMR by majority voting and thus propagates the error-free output. On the other hand, the dual modular redundancy method, the logic circuit is duplicated and works in parallel. The main goal of the DMR technique is to achieve SET fault mitigation similar to that of the TMR technique, and it consumes less power and area compared to that of the TMR technique.

The FTS is used to extend the dependability of the electronic system. The FIS can be tolerant of any kind of faults in the system with expected and predictive ways. The faults are tolerated by the system using redundancies. There are three redundancies, like space, time, and information, based on system failures. These FIS have also implemented either in software-based and hardware-based approaches, which depend upon the system requirements. The FTS applies to many application fields, including cloud computing, distributed systems, and specific applications. The cloud computing [3] are facing with networking faults, service expiry faults, physical faults, process faults. The distributing system [4] is dependent on availability, safety, reliability, and maintainability. These distributing systems are tolerated by faults using reactive FTS and proactive FTS. Reactive FTS includes retry, checkpointing, message logging, and replication.

Similarly, proactive FTS includes software rejuvenation, self-healing, and preemptive migration. The specific applications [5] include reactive FTS and proactive FTS tolerate montage, inspiral, Sipht, epigenomics, and cyber shake. The FPGA based in-operations faults [6] are occurred due to aging, the stress in terms of heat and voltage, and high energy particle impact along with environmental changes. These FPGA based in-operations faults are recovered by using redundancy approaches like TMR, DMR, and other approaches. The FTS is designed for math circuits [7], which identifies the critical gates. The critical gates evaluate the reliability in two ways, nominal reliability and practical reliability. The single event upset is prevented using hardware, software, hybrid, adaptive, and adaptive hybrid approaches are available. The hardware approaches include TMR, DMR, and N-Modular Redundancy. Similarly, software approaches use Error Detection – Duplication Instruction (EDDI), which runs on the processor. The hybrid approach is a combination of software, hardware, and Error Correction Codes (ECC) [8-10].

In this work, the transient fault injection and fault-tolerant system module are designed using TMR based majority voter logic, and DMR based self-voter logic for fault analysis with coverage for digital circuits like synchronous and asynchronous circuits. Section II describes the review of the existing approaches of FIS and FTS for different applications. The proposed FIS-FTS module using TMR and DMR is elaborated in Section III. Section IV analyzes the performance evaluation of the FIS-FTS module with synthesis and fault-tolerant results for digital circuits. Finally, conclusion and future work are discussed in Section V.

II. Related Work

This section describes the review of recent existing works on FIS-FTS modules for different applications for software and hardware approaches. Balasubramanian et al. [11] present combinational circuit based fault tolerance and redundant logic insertion with improvements and estimating fault-masking using the truth-cum fault enumeration table. The fault-tolerant module is designed using 28/32nm CMOS technology by analyzing the figure-of-merit and Power Delay Product (PDP). Schweizer et al. [12] present the FPGA based FIS with fast and accurate analysis for ISCAS-83 based gate circuits. The statistical analysis between RTL/Gate level circuits is mapped for place and route logic in FPGA Design. Li et al. [13] present the CMOS based soft-error sophisticated design for DMR with a low power mechanism. The complementary-DMR is used to mitigate soft errors, which reduces the voting circuit area, error rate, and better time than TMR approaches. The DMR uses separate two circuits and merging circuit to overcome the soft error to improve the area. Sheikh et al. [14] present the DMR based FTS for combinational circuits, which includes C-element (CEL) based DMR to overcome TMR and selective transistor redundancy (STR) modules. The protected C-element provided better fault-tolerance and applied for LGSynth'91 benchmark circuits. The results are analyzed for area-overhead, circuit reliability using DMR, DMR-CEL, and TMR logic modules.

Arifeen et al. [15] describe the approximate-TMR based fault-tolerant voter module to overcome the area –overhead issues in TMR based modules. Approximate-TMR also tolerates the internal faults in voter inputs. Along with full protection to critical parts and partial protection to the rest of the system. The work analysis the PDP, transistors, and reliable calculation using the CMOS approach. The fault-diagnosis system is a module by mandaogade et al. [16] using FIS for discrete systems. The fault-diagnosing system includes a circuit under test (CUT) for 32-bit AND gate, control logic, stimulus generator, and output response. Oliveira et al. [17] present a short circuit based fault- diagnosis (SC-FD) for fault-tolerant voltage-source-inverter (VSI). The reconfigurable feature is incorporated in VSI like two-auxiliary switches, fast fuse, and fast legs. The VSI based decision system is to isolate the fault conditions on switches using diagnostic variables.

The fault-tolerant system used in many other advanced applications includes System-on-Chip (SoC), Network-on-Chip (NoC), and others. Chekmarev et al. [18] present the On-Chip Debugging (OCD) using the modification of FIS for processor cores of SoC. The fault-injection system includes on-chip bus connection of OCD for processor cores, fault-injector, external interface controller, and controller memory acts as a system under test which is connected externally to memory for fault coverage analysis. Hybrid FIS for NoC is presented by Coelho et al. [19], to improve reliability and NoC performance. The FIS has SET, SEU with timer, and is connected with target FPGA and is interfaced with 4x4 NoC via AXI-Interface. Guruprasad et al. [20] present the fault and congestion-free NoC design where faults are tolerated in NoC using Error Code Correction techniques with better fault coverage. Podivinsky et al. [21] present the hardware-based robot controller with a fault-tolerant mechanism, which evaluates single and multiple fault-injection analysis with electronic failures.

III. FIS-FTS Module for Digital Circuits

The fault-injection system, along with the fault-tolerant system, is represented in detail using TMR and DMR methods for digital circuits (both synchronous and asynchronous circuits). The FIS and FTS modules are explained in the following section.

A. Fault Injection System (FIS)

The fault-injection system is designed to randomly generate the faults using the Berlekamp Massey algorithm based on LFSRs and used in FTS. The fault-injection system using LFSR is represented in Fig. 1. The FIS includes BMA based on two LFSRs, fault logic, and multiplexor. The BMA based LFSR provides better execution time than conventional LFSRs. The two LFSR uses the following polynomials for the random data generations in equations (1) and (2).

$$LFSR_1 = x^3 + x + 1 \tag{1}$$

$$LFSR_2 = x^3 + x \tag{2}$$

The BMA based LFSR generates faults based on the above two equations. The control signal load is used in the initial process setup. When the load is activated, the default values assigned to LFSR registers, else the polynomial values are updated in registers. The two LFSR values are XOR'ed in the fault logic process. The fault logic acts as a select line to the multiplexor. The one hot-encoder using shift register generates the encoded data are XORed with feedback register output for the creation of faults in a more random manner. If the fault logic output acts as a select line, if it is 1, then the XORed output is fault data else user data is considered as not a fault data. These faults are considered as transient faults in FIS. These fault data is injected into the FTS for digital circuits, checks the status of the circuits, and analyzes the fault coverage.

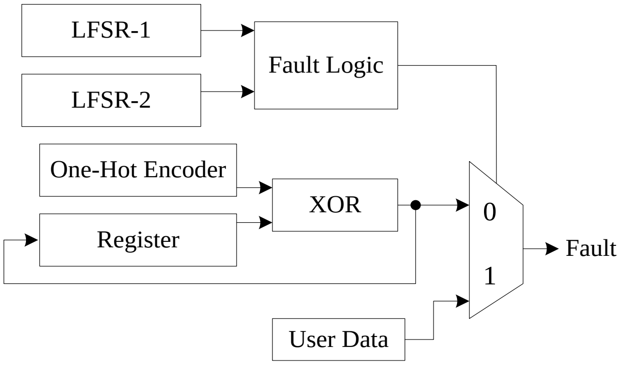


Fig. 1. Fault Injection System using LFSR.

B. Fault-Tolerant System (FTS) using TMR and DMR

The fault-tolerant system is used to tolerate the injected faults in digital circuits. The digital circuits include both synchronous and asynchronous circuits that are considered in designs using triple modular redundancy and dual modular redundancy. The FTS has a TMR module using the majority voter logic and DMR based self-voter logic for fault analysis. The FTS using TMR based MVL has represented in Fig. 2. The asynchronous circuits include 4-bit ripple-carry adder (RCA), 4:16 decoder, 4:2 encoders, and comparator has been used in the FTS process.

Similarly, the synchronous circuits include 4-bit data flip-flop, 4-bit shift register, 4-bit counter, and 4-bit read-only memory (ROM) have been considered in the FTS design process. These circuits output's like X, Y, and Z has considered in the four MVL approaches. The MVL is a simple and most popular approach for detecting and tolerating the faults with coverage in the most successful way. The hardware complexity and performance of the system will be improved by using MVL in TMR. So in this FTS, the different and possible TMR based MVL approaches are designed and analyzed for fault coverage.

The TMR module has constructed using four different MVL approaches and analyzed for both synchronous and asynchronous circuits are represented in Fig. 3. The TMR based four MVL design-1, 2, 3, and 4 approaches are also represented in Fig. 3(a-d).

The TMR based MVL Design-1 is a basic approach that includes three AND gate and one OR gate with equation (3) for output O1 function generation as follows.

$$O_1 = (X.Y) + (Y.Z) + (X.Z) \tag{3}$$

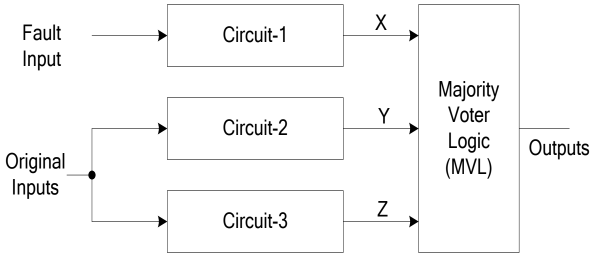


Fig. 2. Fault-Tolerant System using TMR.

The TMR based MVL Design-2 approaches 2 XOR gate, 1 inverter gate, one AND gate, and 1 multiplexor, and its output function O_2 is represented in the following equations (4-6).

$$P = (X \wedge Y),\ Q = (Y \wedge Z),\ \text{and } R = {\sim}Q \tag{4}$$

$$Sel_1 = P.R \tag{5}$$

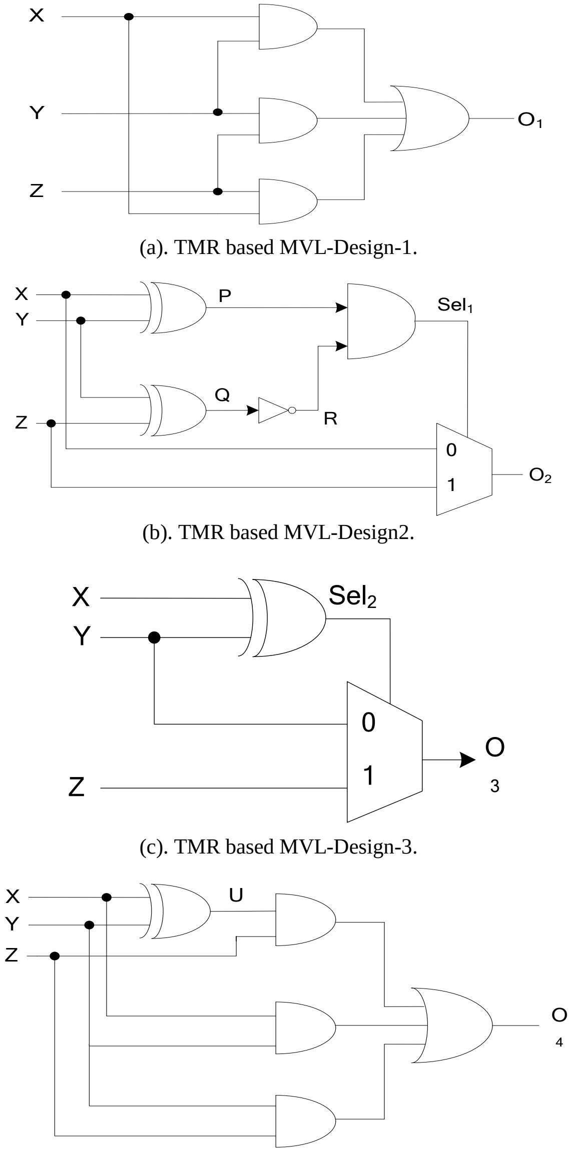
$$O_2 = \text{if } (Sel_1 = 0) \text{ then } X \text{ else } Z \tag{6}$$

Where the representation of basic gates like AND gate is '.' (Dot), OR gate is '+,' XOR gate is '^,' and the Inverter gate is '~.'

The TMR based MVL Design-3 approach has 1 XOR gate and 1 multiplexor, and its output function O_3 is represented in the following equations (7-8).

$$Sel_2 = X.Y \tag{7}$$

$$O_3 = \text{if } (Sel_2 = 0) \text{ then } Y \text{ else } Z \tag{8}$$



(a). TMR based MVL-Design-1.

(b). TMR based MVL-Design2.

(c). TMR based MVL-Design-3.

Fig. 3. (d). TMR based MVL-Design-4.

The TMR based MVL Design-4 approach has 1 XOR gate and 1 multiplexor, and its output function O_4 is represented in the following equations (9-10).

$$U= X.Y \quad (9)$$

$$O_4 = (U.Z) + (X.Y) + (Y.Z) \quad (10)$$

The TMR based MVL design approaches works based on two out of three majority logic. The TMR based MVL Design-two approach uses more basic gates than the other three approaches. The TMR modules triplicate the digital circuits, and the majority voter logic filters outs the single event transients. The TMR based modules can control the common-mode SETs failures.

The dual modular redundancy acts as self-voter logic and has two inputs and one output. If any of the two inputs changed, the output would change the state. The DMR based SVL aims to reduce the chip area overhead than TMR based modules. The DMR based SVL has represented in Fig. 4. The DMR module uses only two circuits, assigns the faults to the first circuit, and maintains the original input in the second circuit.

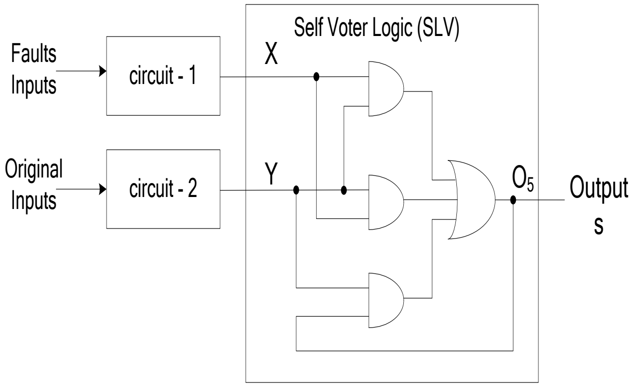


Fig. 4. Fault-Tolerant System using DMR.

The DMR based SVL design approach has three AND gate followed by the OR gate, and its output function O_5 is represented in the following equation (11).

$$O_5 = (X.Y) + (X.Y) + (Y.O_5) \quad (11)$$

The self-check mechanism is incorporated in the DMR module by providing output O_5 data as a feedback input to the SVL module. SET protection applies to clock and data inputs. The FTS module using TMR and DMR logic provides a single event upset and SET protection for the digital circuit inputs.

IV. Results and Discussion

The Results and discussion of Fault injection system -Fault Tolerant system (FIS-FTS) for digital circuits, which includes synchronous and asynchronous circuits, are analyzed with detailed performance metrics like Chip area, combinational delay, and total power along with fault coverage. The fault-tolerant and coverage are analyzed in a simulator tool based on the number of injected faults. The FIS-FTS module is synthesized using Xilinx ISE 14.7 environment and simulated using Modelsim 6.5f simulator and implemented and prototyped using the Artix-7 FPGA device.

A. Synthesis Results

The FIS-FTS module chip-area utilization for Asynchronous Circuits using different voter logic are tabulated in Table I. And graphical representation for the same is shown in Fig. 5. The FTS is designed using TMR module with four different Majority voter logic (MVL) like TMR-D1, TMR-D2, TMR-D3, and TMR-D4. Similarly, FTS is also designed using the DMR module with self-voter logic (SLV).

The FIS-FTS module area utilization is evaluated using LUT's and Gate count (GC). The Utilization of LUT's is <1% in Artix-7 FPGA for all the asynchronous circuits like 4-bit RCA, decoder, encoder, and comparator in different voter logic's includes TMR and DMR. The Gate count (GC) is calculated using many Flip-flops (Registers), XOR-Gates, and Multiplexors, along with Block RAM's (BRAMs) and comparators. The gate count values differ based on the asynchronous circuits used in FTS. The Decoder circuit uses Three BRAMs, and Comparator circuits use six comparisons in FTS based TMR D1 to TMR-D4 modules. Similarly, in DMR based FTS, the Decoder circuit uses two BRAMs, and Comparator circuits use four comparators along with Registers, XOR gates, and Multiplexors. Overall, The DMR based FIS-FTS module using Asynchronous Circuits consumes less gate count (GC) than TMR based FIS-FTS module.

TABLE I. FIS-FTS Module Area utilization for Asynchronous Circuits

Asynchronous Circuits	Resources	TMR-D1	TMR-D2	TMR-D3	TMR-D4	DMR
4-bit RCA	LUT's	31	40	31	31	29
	GC	74	79	77	74	66
Decoder	LUT's	41	41	41	41	50
	GC	53	56	55	53	52
Encoder	LUT's	28	33	28	28	36
	GC	59	62	61	59	56
Comparator	LUT's	32	32	32	32	29
	GC	71	77	74	71	64

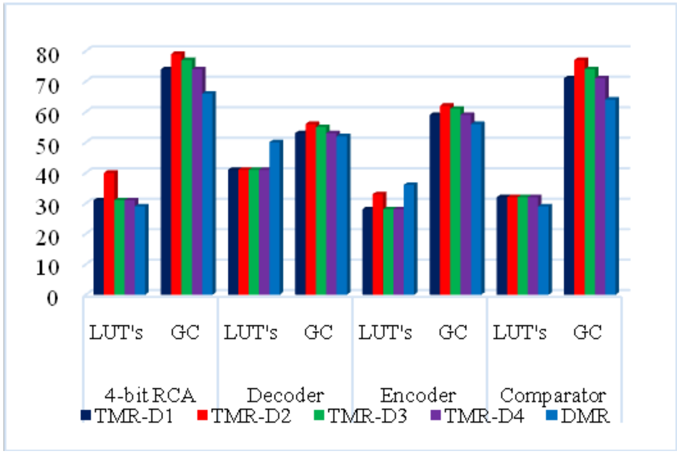


Fig. 5. Graphical Representation of FIS-FTS Module Area utilization for Asynchronous Circuits.

The Combinational delay (ns) are analyzed for the FIS-FTS module using Asynchronous circuits and represented in Fig. 6. The Asynchronous circuits like 4-bit RCA, decoder, and comparator generates better combinational delay using TMR based FIS-FTS module than DMR based FIS-FTS module. The TMR-Design-4 gives better delay for 1.68ns, 1.002ns, and 2.115ns than DMR Based design like 2.26ns, 1.709ns, and 2.517ns for RCA, Decoder, and comparator respectively.

The FIS-FTS module resource utilization for Synchronous Circuits using TMR based MVL designs and DMR based SVL are tabulated in Table II, and graphical representation for the same is shown in Fig. 7. The FIS-FTS module utilizes a < 1% LUTs chip area on Artix-7 FPGA for all the synchronous circuits like Data Flip-flop, Shift register, counter, and ROM using TMR based MVL and DMR based SVL.

In Gate Count Calculation, the counter circuit uses additionally 3 adders for all TMR based MVL designs and 2 adders for DMR based SVL designs. The ROM circuits use 3 BRAMsfor all TMR based MVL designs and 2 adders for DMR based SVL design along with Registers, XOR gates, and Multiplexors. The DMR based FIS-FTS module utilizes less GC than TMR Based FIS-FTS module. The LUTs utilization for all the Synchronous Circuits based on TMR and DMR modules on an average is 26.

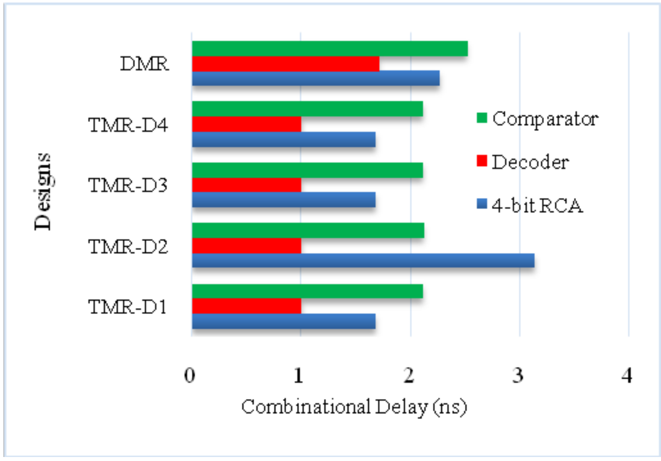


Fig. 6. Combinational Delay Analysis of Asynchronous Circuits in FIS-FTS Module.

TABLE II. FIS-FTS MODULE AREA UTILIZATION FOR SYNCHRONOUS CIRCUITS

Synchronous Circuits	Resources	TMR-D1	TMR-D2	TMR-D3	TMR-D4	DMR
Data Flip-flop	LUT's	25	31	25	25	22
	GC	62	65	64	62	58
Shift Register	LUT's	25	25	25	25	21
	GC	59	62	61	59	56
Counter	LUT's	41	63	41	41	54
	GC	65	68	67	65	60
ROM	LUT's	28	28	28	28	22
	GC	65	68	67	65	52

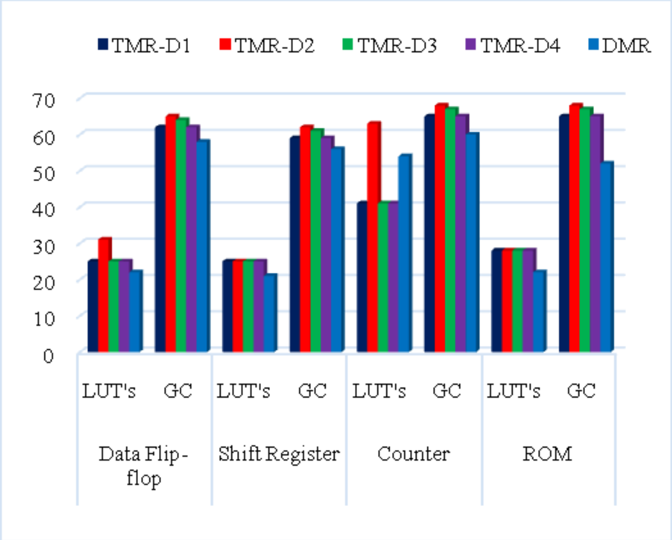


Fig. 7. Graphical Representation of FIS-FTS Module Area utilization for Synchronous Circuits.

The TMR Based FIS-FTS module for Synchronous Circuits and Asynchronous Circuits works at 606.50MHz, and DMR based FIS-FTS module works at 523.80MHz. The DMR Based FIS-FTS design utilizes less chip area than TMR based FIS-FTS designs on Artix-7 FPGA. For both Synchronous Circuits and Asynchronous Circuits.

The Power utilization of digital circuits (Synchronous Circuits and Asynchronous Circuits) in the FIS-FTS module is represented in Fig. 8. The DMR based FIS-FTS utilizes less total power (W) than the TMR based FIS-FTS module. The Xilinx –Xpower analyzer is used for power calculation at 100MHz clock frequency. The TMR Design-2 Approach utilizes more power than all other approaches in the FIS-FTS module. The ROM circuits use an average of 0.086W for all the FTS based design approaches. The power utilization of all digital circuits in the FIS-FTS module is in the range of 0.086 to 0.092W, which is quite less and suitable real-time FTS based digital circuits.

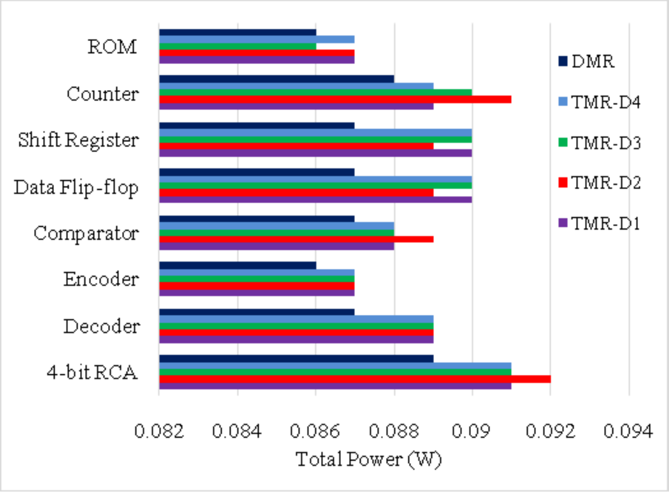


Fig. 8. Total Power utilization of Digital Circuits in FIS-FTS Module using different Voter Logic.

B. Performance Analysis

The performance analysis is evaluated using % fault coverage of digital circuits using the FIS-FTS module. The % fault coverage is calculated using the below formula:

$$\% \text{ fault coverage} = \frac{Number of Faults tolerated}{Total number of faults injected} X\ 100$$

The Number of Transient Faults injected to the FIS-FTS module for all digital circuits is 100. The calculation of tolerated faults or fault coverage is calculated by analyzing the simulation waveform results. Based on simulation analysis for all the digital circuits in FIS-FTS modulation, the number of tolerated faults is calculated and is tabulated in Table III.

The TMR-Design-2 method in the FIS-FTS module tolerates 98 in RCA, 95 in the encoder, 94 in Data Flip-flop, and 92 faults in the counter. Similarly, DMR based FIS-FTS module tolerate 94 faults in data-FFs and 94 faults in the shift register. The TMR based Design-1, 3, and 4 methods are tolerated with 100% fault coverage for all the digital circuits. The TMR Design-2 and DMR based FIS-FTS are tolerated around with fault coverage of 96.42% and 98.5%, respectively.

C. Comparative Results

The comparative results are analyzed for Full adder circuits using different FTS methods [22] with the proposed TMR-D3 and DMR methods, tabulated in Table IV. The existing FTS methods and present works are synthesized on the same Spartan-3 FPGA. The Existing TMR and Novel Fault-tolerant (NFT) FTS methods use 5 slices, whereas TMR-D3 uses only 1 slice, and DMR uses 2 slices. The combinational delay (ns) of TMR and NFT applies 13.36ns and 9.97ns, respectively. The proposed TMR-D3 uses a delay of 7.824ns, and DMR uses 8.262ns. The Total power utilization of TMR and NFT consumes 0.135W and 0.129W, respectively. The proposed work TMR-D3 uses total power of 0.28W, and DMR consumes0.3W. The proposed FTS methods (TMR-D3and DMR) are better Chip areas (Slices and LUT's), Combinational delay, and Total power than the existing FTS approaches [22].

TABLE III. % FAULT COVERAGE USING FIS-FTS MODULE FOR DIGITAL CIRCUITS

Digital Circuits	**TMR-D1**	**TMR-D2**	**TMR-D3**	**TMR-D4**	**DMR**
4-bit RCA	100	98	100	100	100
Decoder	100	100	100	100	100
Encoder	100	95	100	100	100
Comparator	100	100	100	100	100
D-FF	100	94	100	100	94
Shift Register	100	100	100	100	94
Counter	100	88	100	100	100
ROM	100	100	100	100	100
% Fault Coverage	**100**	**96.42**	**100**	**100**	**98.5**

TABLE IV. COMPARATIVE RESULTS FOR FULL ADDER USING DIFFERENT FTS METHODS

FTS Method	**Slices**	**LUT's**	**Delay (ns)**	**Power (W)**
TMR [22]	5	9	13.364	0.135
NFT [22]	5	8	9.977	0.129
TMR-D3 (This work)	1	5	7.824	0.28
DMR (This work)	2	4	8.262	0.3

V. CONCLUSION AND FUTURE WORK

In this Manuscript, An efficient Fault-Injection system (FIS), followed by a Fault-Tolerant System (FTS), is designed for Synchronous Circuits and Asynchronous Circuits. The FTS is designed using TMR and DMR methods. The TMR method works based on the Majority voter logic (MVL), and DMR based self-voter logic (SVL). The TMR based four different MVL approaches are designed and analyzed for both Synchronous Circuits and Asynchronous Circuits. The TMR Based Design-1, Design-3, and Design -4 are better than design -2 by concerning the hardware constraints. The DMR based FIS-FTS has better Area (LUTs), Gate-Count, and power utilization than four-TMR based FIS –FTS in both Synchronous Circuits and Asynchronous Circuits. The TMR Based FIS-FTS using Design-1, Design-3, Design-4 approaches are 100% fault-tolerant, while design-2 has 96.42% fault-tolerant, and DMR Based FIS-FTS has 98.5% fault-tolerant for both Synchronous Circuits and Asynchronous Circuits. In the future, the FIS-FTS module is incorporated in Commination and networking applications. The proposed FTS methods are compared with the existing FTS approaches for the Full adder circuit with better resource utilization.

REFERENCES

[1] Eslami, Mohammad, Behnam Ghavami, Mohsen Raji, and Ali Mahani. "A survey on fault injection methods of digital integrated circuits." Integration (2019).

[2] Kooli, Maha, and Giorgio Di Natale. "A survey on simulation-based fault injection tools for complex systems." In 2014 9th IEEE International Conference On Design & Technology of Integrated Systems In Nanoscale Era (DTIS), pp. 1-6. IEEE, 2014.

[3] Kumari, Priti, and Parmeet Kaur. "A survey of fault tolerance in cloud computing." Journal of King Saud University-Computer and Information Sciences (2018).

[4] Ledmi, Abdeldjalil, Hakim Bendjenna, and Sofiane Mounine Hemam. "Fault Tolerance in Distributed Systems: A Survey." In 2018 3rd International Conference on Pattern Analysis and Intelligent Systems (PAIS), pp. 1-5. IEEE, 2018.

[5] Singh, Shefali, and Inderveer Chana. "Survey on Fault-Tolerant Techniques in Scientific Applications." In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 618-624. IEEE, 2018.

[6] Balach, J., & Novak, O. (2010, September). Reconfigurable Fault-Tolerant System Synchronization. In 2010 13th Euromicro Conference on Digital System Design: Architectures, Methods, and Tools (pp. 817-820). IEEE.

[7] Ban, Tian, and Gutemberg GS Junior. "Critical gates identification for fault-tolerant design in math circuits." Journal of Electrical and Computer Engineering 2017 (2017).

[8] Hamilton, Nicolas, Scott Graham, Timothy Carbino, James Petrosky, and Addison Betances. "Adaptive-Hybrid Redundancy with Error Injection." Electronics 8, no. 11, (2019): 1266.

[9] Agiakatsikas, Dimitris, Ganghee Lee, Thomas Mitchell, Ediz Cetin, and Oliver Diessel. "From C to Fault-Tolerant FPGA-Based Systems." In 2018 IEEE 26th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM), pp. 212-212. IEEE, 2018.

[10] Lu, Wentao, Ruping Wang, Chenhui Zeng, Chenyan Liu, and Xin Wang. "A General Fault Injection Method Based on JTAG." In 2018 Prognostics and System Health Management Conference (PHM-Chongqing), pp. 604-608. IEEE, 2018.

[11] Balasubramanian, P., and R. T. Naayagi. "Redundant logic insertion and fault tolerance improvement in combinational circuits." In 2017 International Conference on Circuits, System and Simulation (ICCSS), pp. 6-13. IEEE, 2017.

[12] Schweizer, Thomas, Dustin Peterson, Johannes M. Kühn, Tommy Kuhn, and Wolfgang Rosenstiel. "A fast and accurate FPGA-based fault injection system." In 2013 IEEE 21st Annual International Symposium on Field-Programmable Custom Computing Machines, pp. 236-236. IEEE, 2013.

[13] Li, Yan, Yufeng Li, Han Jie, Jianhao Hu, Fan Yang, Xuan Zeng, Bruce Cockburn, and Jie Chen. "feedback-based low-power soft-error-tolerant design for dual-modular redundancy." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 26, no. 8 (2018): 1585-1589.

[14] Sheikh, Ahmad T., and Aiman H. El-Maleh. "Double Modular Redundancy (DMR) Based Fault Tolerance Technique for Combinational Circuits." Journal of Circuits, Systems and Computers 27, no. 06, (2018): 1850097.

[15] Arifeen, Tooba, Abdus Sami Hassan, and Jeong-A. Lee. "A Fault-Tolerant Voter for Approximate Triple Modular Redundancy." Electronics 8, no. 3 (2019): 332.

[16] Mandaogade, Nitin Nemikumar, and Prashant V. Ingole. "Development of Fault Diagnosis System Using Fault Injection for Discrete Architectures." In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1-6. IEEE, 2018.

[17] Oliveira, Andre BM, Robson L. Moreno, and Enio R. Ribeiro. "Digital short-circuit diagnosis and reconfiguration of a fault-tolerant VSI." In 2017 IEEE Southern Power Electronics Conference (SPEC), pp. 1-6. IEEE, 2017.

[18] Chekmarev, S. A., V. Kh Khanov, and O. A. Antamoshkin. "Modification of fault injection method via on-chip debugging for processor cores of systems-on-chip." In 2015 International Siberian Conference on Control and Communications (SibCon), pp. 1-4. IEEE, 2015.

[19] Coelho, Alexandre, Nacer-Eddine Zergainoh, and Raoul Velazco. "NoCFI: A Hybrid Fault Injection Method for Networks-On-Chip." In 2019 IEEE Latin American Test Symposium (LATS), pp. 1-6. IEEE, 2019.

[20] Guruprasad S.P and Chandrasekar B.S, "Design of an Efficient Fault and Congestion Free NoC Design using Adaptive Routing on FPGA." International Journal of Recent Technology and Engineering (IJRTE), Vol.8, Issue 3, pp.1055-1062, September 2019.

[21] Podivinsky, Jakub, Jakub Lojda, and Zdenek Kotasek. "An Experimental Evaluation of Fault-Tolerant FPGA-Based Robot Controller." In 2018 IEEE East-West Design & Test Symposium (EWDTS), pp. 1-7. IEEE, 2018.

[22] Kshirsagar, R. V., and R. M. Patrikar. "A novel fault-tolerant design and an algorithm for tolerating faults in digital circuits." In 2008 3rd International Design and Test Workshop, pp. 148-153. IEEE, 2008.