International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH14) 28 & 29 November 2014

Scan-Chain-Based Multiple Error Recovery in TMR Systems (SMERTMR)

Firoz Ahmed Siddiqui
Research Scholar,
Department of Electronics and Communication Engineering,
NRI Institute of Information Science & Technology,
Bhopal, Madhya Pradesh, India

Prof. Puran Gour
Associate Professor,
Department of Electronics and Communication Engineering,
NRI Institute of Information Science & Technology,
Bhopal, Madhya Pradesh, India

Abstract— In this paper, we present a roll forward error recovery technique for triple modular redundancy (TMR) system based on scan-chains, calledscan-chain-based multiple error recovery technique in triple modular redundancy systems (SMERTMR). This technique uses scan-chain flip-flops to detect and correct faulty modules in the presence of single or multiple transient faults. In this technique both, the errors in the output of the voter and the error due to latent faults at the input of modules are detected. The latent faults are detected by comparing the internal states of the TMR modules. Upon detection of any mismatch between the modules output, the SMERTMR system will find the faulty modules and it can correct the faulty module by copying the state of a fault-free module into the faulty modules. Upon detection of permanent fault, the system will degrade to a master/checker configuration. In this paper we are proposing SMERTMR in modify mode which is the combination of comparison and recovery mode of SMERTMR.

Keywords- Roll-forward error recovery, Rollback error recovery, scan chain TMR (ScTMR), triple modular redundancy (TMR).

I. INTRODUCTION

Today, the embedded systems are widely used in safety-critical applications like in aviation, process control, and patient life-support monitoring system [1], [2]. These systems require fault free operation within the time limit. To meet these requirements, such embedded systems should be equipped with appropriate error detection and correction mechanisms. On the other hand, in some systems the timing constrains are ignored in achieving a high level of reliability. For example, in a rollback recovery-based system, the overall reliability is improved but the probability of missing deadline increases for certain applications because expected response time increases. In safety-critical application, timing is a very important factor and it can't be neglected. Therefore improving system reliability considering its real-time constraints is very important for safety-critical applications. Therefor in embedded processors providing fault-tolerant techniques which provide minimum performance overhead is very important.

A TMR [3], [4] system consists of three identical modules and a voter. The voter is connected at the outputs of modules which have some errors that should be removed in order to be used in safety-critical applications. A major disadvantage of the traditional TMR is its inability to work when TMR itself fails. TMR failure means the failure in a TMR system caused by more than one faulty module or a faulty voter [5]. In case if a fault occurs in two different modules and neither of the faults is recovered, it may result in a TMR malfunction. For long-term applications, the absence of appropriate recovery mechanisms increases the probability of TMR failure [6], [7]. To overcome this problem, TMR should have a transient error recovery technique. Most of the TMR-based error recovery techniques proposed so far uses roll back or retry mechanisms [1], [5], [7]–[9]. In rollback mechanism, once an error detected, the faulty module will re-execute the entire process. However these techniques are not suitable for applications in which tight deadline is required, because rollback or retry mechanism may increase the performance overhead and therefore may cross the deadline [10].

In roll forward error recovery technique the correct state of module copied from a fault free module to faulty module. As re-computation is not needed in roll forward technique,therefore this technique can be used in tight deadline applications.

The other technique called ScTMR provides recovery for both transient and permanent errors in TMR systems [11]. ScTMR uses a roll-forward approach and uses the scan chain to recover the system from faulty state. Although ScTMR has the advantage of reducing the probability of TMR failures, it has two main disadvantages. First, ScTMR cannot recover a single faulty module in the presence of latent faults [13]. A fault is referred to as latent if it does not propagate to the system output but does cause a mismatch between the states of the TMR modules. Second, ScTMR is not able to recover the system if multiple faults are simultaneously occurring in the outputs of two modules.

SMERTMR is a scan-chain-based roll-forward error recovery technique for TMR-based systems, which overcomes the disadvantages of ScTMR. The scan chain-based multiple error recovery TMR (SMERTMR), has the capability to identify and remove latent faults in TMR modules as well as to recover the system from multiple faults affecting two TMR modules. SMERTMR is the first roll-forward error recovery technique that has the capability to recover the error in the presence of multiple latent faults and two faulty modules. SMERTMR reuse the available scan chains, used for testing purposes, to compare the internal states of TMR modules to

locate the faulty module and restore the correct state from fault free module to the faulty module. In case of permanent faults, the faulty module is disregarded and the system will be degraded to the master/checker (M/C) configuration. In this configuration the offline testability is used for detection and correction of permanent fault. SMERTMR has small area overhead, as compare to other TMR-based recovery techniques, as it reuses the existing resources within the circuit.

II. SCTMR TECHNIQUE

The ScTMR technique reuses scan chains for recovering the state of the faulty module by copying the state of fault free module to the faulty module. Scan chain [8] is a cost-effective technique to provide a simple way for testing combinational and sequential circuits. In this technique, flip-flops are chained together through a long shift register circuit and a multiplexer is used in front of each flip-flop to switch between the normal and testing operations. In order to reduce the observation and loading time multiple scan chains are used. Multiple scan chains include parallel chains, which consist of approximately the same number of flip-flops. The number of flip-flops in each scan chain is called scan chain length and the number of scan chains in parallel is called scan chain width.

Figure 1 shows the block diagram of the ScTMR technique. As shown in the figure, the ScTMR architecture consists of three identical modules, a voter and a ScTMR controller. The voter detects errors and reports these errors to the ScTMR controller. The ScTMRcontroller identifies the error type and use an appropriate mechanism to remove the error effects from the system. Then the ScTMR controller uses scan chains to copy the state of a fault-free module into the faulty module. The scan chain signals,which consist ofScan Chain Input (SCI), Scan Chain Output (SCO), and Scan Chain Enable (SCE), are monitored and controlled by the ScTMR controller.

Figure 2 shows the state diagram of a system using ScTMR technique. Initially, the system is in the normal state and, when an error is detected by the voter, the recovery process is initiated. During the recovery process, the ScTMR controller detects the faulty module as well as the fault type i.e. whether the fault is permanent or transient.

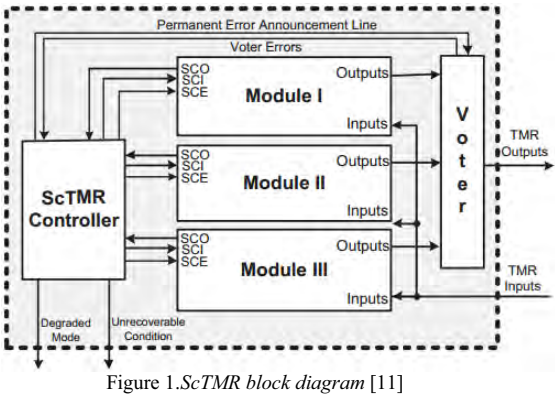

Figure 1.*ScTMR block diagram* [11]

When a permanent fault is detected in one of the modules, the system is degraded to a Master/Checker configuration. If the detected fault is a transient fault, the system will do the recovery process to bring the system to the fault-free state. When multiple transient faults are detected, the recovery process terminatesitself and the system will be halted immediately to provide a fail-safe state.

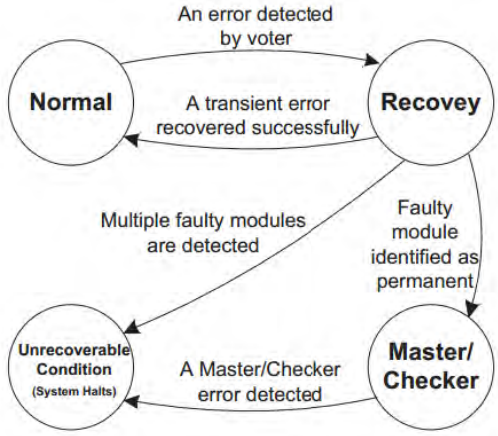

Figure. 2. *ScTMR State diagram* [16]

A. Proposed Voter

The major concern in a highly reliable TMR system is finding the faulty module [12]. To address this concern, ScTMR voter has the capability of locating the faulty module [14] [15]. The ScTMR voter also has the capability of detecting and locating faults within the comparators. Figure 3 shows the ScTMR voter architecture. In this voter, three comparators (C_{12}, C_{13}, and C_{23}) are used to compare the outputs of the TMR modules. Three error signals (TE_{12}, TE_{13}, and TE_{23}) are generated showing mismatch between TMR modules one-two, one-three and two-three respectively. If one of the TMR modules becomes faulty and the fault is not corrected within the module, an error is established to the output of the faulty module and the corresponding output becomes erroneous. Since the output of each module is compared by the outputs of the other modules, the error is detected by two comparators. For an example, if output I become inaccurate, the error is detected by C_{12} and C_{13} and therefore both error signal TE_{12} and TE_{13} are activated accordingly. On the other hand, if one of the comparators becomes faulty, only the corresponding error signal is activated.

The ScTMR voter employs three input signals denoted by Pr_{12}, Pr_{13}, and Pr_{23}. These input signals are used to detect permanent faults. These signals are derived by the ScTMR controller and are set to zero before a fault identified as a permanent fault. In this case, E_{13}, E_{12}, and E_{23} are equivalent to TE_{13}, TE_{12}, and TE_{23}, respectively.

The error signals (E_{13}, E_{12}, and E_{23}) are connected to the output selector circuit and to the ScTMR controller. The output selector circuit selects the error-free output and gives

the ultimate output which is error-free output. The faulty module and the voter output are identified using different values of error signals according to the truth table shown in Table I. As shown in this table, if either one of the comparators (C_{13}, C_{12}, and C_{23}), module II, or module III becomes faulty, the output of module I is selected by the output selector circuit as the error-free output of the system. If module I become faulty, output II will be selected by the selector circuit as the error-free output of the system. Therefore, the output selector circuit can be simply implemented by a 2-to-1 multiplexer. If all the error signals are active then this condition is called unrecoverable condition.

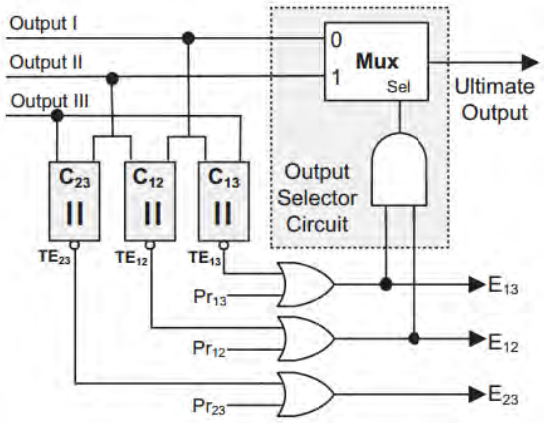

Figure 3. *Proposed Voter* [11]

B. *Transient and Permanent Error Recovery*

The ScTMR controller recovers the system from both transient and permanent faults. As soon as the error is detected by the voter it gives an error signal to the controller. As soon as the error signal is received by the controller, it changes the system state from normal mode to recovery mode and restores the correct state of the system by using the states of fault-free modules.

TABLE I. IDENTIFYING FAULTY MODULE [11]

Error Signals			Faulty Module	Output
E_{12}	E_{13}	E_{23}		
0	0	0	-	Output I
0	0	1	C_{23}	Output I
0	1	0	C_{13}	Output I
0	1	1	Module III	Output I
1	0	0	C_{12}	Output I
1	0	1	Module II	Output I
1	1	0	Module I	Output II
1	1	1	Unrecoverable	X

Figure 4 shows a block diagram of a ScTMR controller in recovery mode. ScTMR controller can also identify the fault type i.e. whether the fault is permanent or transient. For this ScTMR controller uses two internal registers, first is most recent faulty module (MRFM) and the second is the number of consecutive faults (NCF). In MRFM register the faulty module number is stored. For example if module II is faulty, MRFM is equal to 2. When another faulty module is detected, the faulty module number is compared with the number stored in MRFM register. If these numbers are equal then ScTMR controller increments NCF register by 1, however if these numbers are not equal the controller resets NCF register. Whenever the value of NCF register exceeds a predefined value, the module is considered as a permanently faulty module. In this case the system will go into Master/Checker configuration.

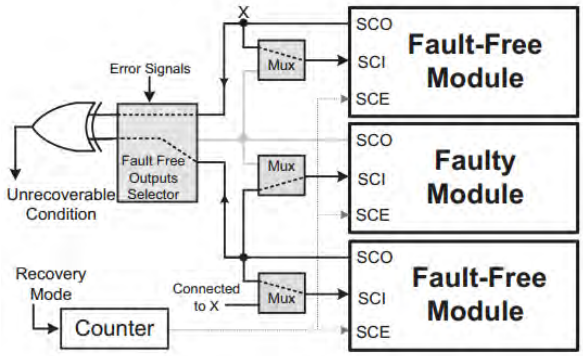

Figure 4. *ScTMR in recovery mode* [11]

III. PROPOSED TECHNIQUE

The ScTMR has some disadvantages: 1) the ScTMR technique can recover the system from a transient fault only if it displays in the output of modules. This is because ScTMR voter compares the output of modules. Because of this it may be possible that a fault remains latent in a module for a long time without propagating to the module outputs. During this period, if a second fault occurs in the other modules, then ScTMR will not recover the correct state and it will enter into unrecoverable condition. 2) If a fault occurs in any one of the TMR modules while there is a latent fault in the other modules then ScTMR fails to recover the correct state of the system because of having two faulty modules at a time. In this case ScTMR detects that there are two faulty modules and enters the unrecoverable condition.

Therefore, to avoid these disadvantages latent faults should be considered while designing the fault-tolerant systems. Therefor SMERTMR technique employing a comparison mode in order to locate latent faults. In this mode, the internal state of each TMR module is compared with the other modules to find the number of latent faults.

Figure5 shows the state diagram of SMERTMR. As shown in the state diagram initially the system will be in the normal mode. When an error is detected by the voter or checkpoint signal is activated, the system switches from normal mode to the comparison mode. In the comparison mode, the internal states of the TMR modules are compared with each other to find the faulty modules and to determine the fault type. If no mismatch is found between the pairs of the modules, the system returns to its normal mode. Otherwise, the system will go to the recovery mode and the recovery process starts. If the recovery process finishes successfully, the system returns to

the normal mode. Otherwise, it enters the unrecoverable condition and halts the system. During comparison mode SMERTMR can also detect permanent faults in one module. If SMERTMR detects a permanent fault, the system enters into the Master/Checker mode. In the Master/Checker mode, any fault in the master or the checker modules results in an unrecoverable condition. In this case, other methods such as functional testing could be exploited to locate and identify the faulty module.

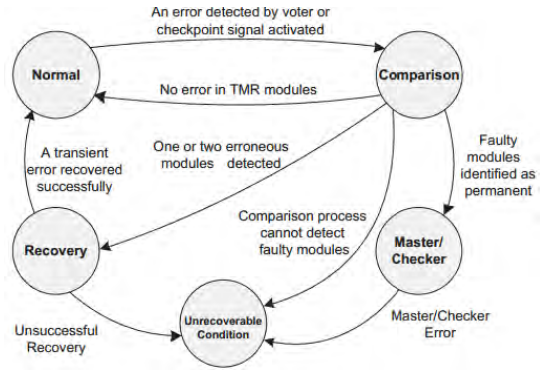

Figure5. *SMERTMR state diagram* [16]

A. *SMERTMR in Comparision mode*

The comparison mode of SMERTMR is activated by two ways: 1) when an error is detected by the voter and 2) when the checkpoint signal is activated. The checkpoint signal is used to intentionally trigger the comparison mode in order to eliminate latent faults. As soon as the voter detects an error, it gives an error signal to SMERTMR controller. Upon initiation of the error signal, the SMERTMR controller switches its state from normal mode to comparison mode. After detecting the faulty modules, SMERTMR controller switches its state from comparison mode to recovery mode in order to recover the faulty modules.

In this mode, upon detection of an error, the SMERTMR controller switches from normal mode to comparison mode. After locating the faulty module it enters into the recovery mode in order to recover the faulty module.

B. *SMERTMR in recovery mode*

Upon identification of faulty module and fault-free module the comparison process ends. The system enters into the recovery mode if it detects one or two faulty modules.In recovery mode SMERTMR controller recovers the correct state of modules. In this mode the system recovers faulty module by copying the states of fault free module to the faulty module by using scan chains.

C. *SMERTMR in modify mode*

We have proposed the new mode of SMERTMR controller which we call a modify mode. This is the combination of both comparison mode and recovery mode. In this mode we are comparing the sates of three modules with each other. Upon detection of any mismatch the position of that bit is changed by the corresponding bit of non-faulty module. For example if the comparator detects the mismatch in 4th bit of module 1 and 2, then the 4th bit of both the modules is replaced by the 4th bit of non-faulty module. This process will be continued till SMERTMR controller recovers the faulty modules.

Figure 6 shows the SMERTMR in modify mode. In this mode the state of modules are continuously monitored and compared with the correct state. Upon detection of error in any module, recovery process starts.

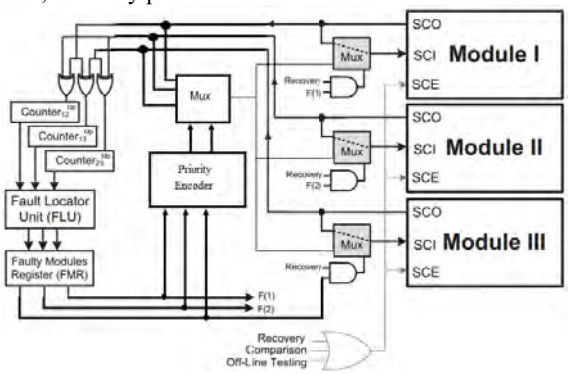

Figure 6. *SMERTMR in modify mode*

TABLE II. COMPARISON OF DIFFERENT TECHNIQUES

Design Summery/Different Techniques	ScTMR	SMERTMR in Comparison Mode	SMERTMR in Recovery Mode	SMERTMR in Modify Mode
Number of Slice Flip Flops	61	135	56	59
Total number of 4 input LUTs	189	551	171	257
Total equivalent gate count for design	2062	5412	1798	2218
Additional JTAG gate count for IOBs	720	192	1776	480

Table II shows the design comparison of different techniques. It clearly shows that the total gate count and JTAG gate count required to design SMERTMR in modify mode is less than the other techniques.The proposed technique is the combination of comparison and recovery mode.The proposed technique performs both comparison and recovery operation and recovers the faulty module.

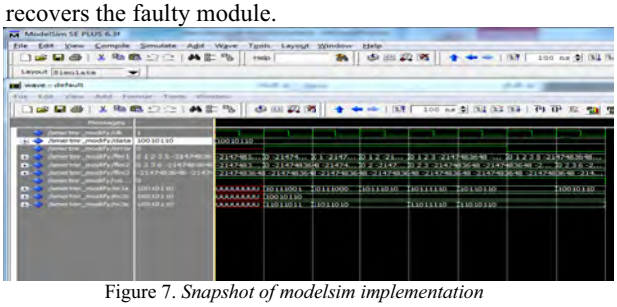

Figure 7. *Snapshot of modelsim implementation*

Figure 7 shows the snapshot of modelsim implementation. It shows that the SMERTMR controller recovers the two faulty modules by continuously comparing their states with the state of fault-free module.

IV. CONCLUSION

In this paper, we proposed a TMR based roll-forward error recovery technique SMERTMR in the modify mode.This technique is the combination of comparison and recovery mode of operation. This technique recovers multiple errors in TMR systems. In this technique, SMERTMR can recover faulty modules in the presence of multiple transient faults and latent faults. The result shows that the proposed technique is less expensive as compared to the traditional TMR systems, as it requires less number of logical gates to design.This technique can be used in embedded applications like in safety critical applications like in aviation, process control, and patient life-support monitoring system

REFERENCES

[1] A. L. J. Hopkins, T. B. Smith, and J. H. Lala, "FTMP—A highly reliable fault-tolerant multiprocess for aircraft," *Proc. IEEE, vol. 66, no. 10, pp. 1221–1239, Oct. 1978.*

[2] J. Gaisler, "A portable and fault-tolerant microprocessor based on the SPARC v8 architecture," in *Proc. Int. Conf. Depend. Syst. Netw.*, 2002, pp. 409–415.

[3] F. L. Kastensmidt, L. Sterpone, L. Carro, and M. S. Reorda, "On the optimal design of triple modular redundancy logic for SRAM-based FPGAs," in *Proc. Design Autom. Test Eur. Conf. Exhibit.*, 2005, pp. 1530–1591.

[4] P. K. Samudrala, J. Ramos, and S. Katkoori, "Selective triple modular redundancy (STMR) based single-event upset (SEU) tolerant synthesis for FPGAs," *IEEE Trans. Nucl. Sci.*, vol. 51, no. 5, pp. 2957–2969, Oct. 2004.

[5] H. Kim and K. G. Shin, "Design and analysis of an optimal instructionretry policy for TMR controller computers," *IEEE Trans. Comput.*, vol. 45, no. 11, pp. 1217–1225, Nov. 1996.

[6] S. Yu and E. J. McCluskey, "On-line testing and recovery in TMR systems for real-time applications," in *Proc. Int. Test Conf.*, 2001, pp. 240–249.

[7] K. G. Shin and H. Kim, "A time redundancy approach to TMR failures using fault-state likelihoods," *IEEE Trans. Comput.*, vol. 43, no. 10, pp. 1151–1162, Oct. 1994.

[8] P. K. Chande, A. K. Ramani, and P. C. Sharma, "Modular TMR multiprocessor system," *IEEE Trans. Ind. Electron.*, vol. 36, no. 1, pp. 34–41, Feb. 1989.

[9] S. D'Angelo, C. Metra, and G. Sechi, "Transient and permanent fault diagnosis for FPGA-based TMR systems," in *Proc. Int. Symp. Defect Fault Tolerance VLSI Syst.*, 1999, pp. 330–338.

[10] J. Yoon and H. Kim, "Time-redundant recovery policy of TMR failures using rollback and roll-forward methods," *IEE Proc. -Comput. Digit. Tech.*, vol. 147, no. 2, pp. 124–132, Mar. 2000.

[11] M. Ebrahimi, S. G. Miremadi, and H. Asadi, "ScTMR: A scan chainbased error recovery technique for TMR systems in safety-critical applications," in *Proc. Design Autom. Test Eur. Conf. Exhibit.*, 2011, pp. 1–4.

[12] G. Latif-Shabgahi and S. Bennett, "Adaptive majority voter: A novel voting algorithm for real-time fault-tolerant control systems," in *Proc.25th EUROMICRO Conf.*, vol. 2. 1999, pp. 113–120.

[13] X. Wang, "Partitioning triple modular redundancy for single event upset mitigation in FPGA," in *Proc. Int. Conf. E-Product E-Service EEntertain.*, 2010, pp. 1–4.

[14] G. Rui, C. Wei, L. Fang, D. Kui, and W. Zhiying, "Modified triple modular redundancy structure based on asynchronous circuit technique," in *Proc. 38th Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2006, pp. 184–196.

[15] X. She and K. S. McElvain, "Time multiplexed triple modular redundancy for single event upset mitigation," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 4, pp. 2443–2448, Aug. 2009.

[16] Mojtaba Ebrahimi, Seyed Ghassem Miremadi, Seyed Ghassem Miremadi, Mahdi Fazeli "Low-Cost Scan-Chain-Based Technique to Recover Multiple Errors in TMR Systems"*IEEE Trans. On very large scale integration(VLSI) Systems,* vol. 21, no. 8, Aug. 2013