Multiple Error Recovery in TMR Systems

Ms.Neethu Raj
Student: Applied Electronics and Instrumentation
Younus College of Engineering & Technology
Kollam, India
neethurj9@gmail.com

Ms. Nisha Lali.R
Assoc. professor: Dept. of ECE,
Younus College of Engineering & Technology
Kollam, India
nisha.lali.r@gmail.com

Abstract—**Nano-scale devices are continuously shrinking, operating at lower voltages and higher frequencies. This makes them more susceptible to environmental perturbations and distinguished by their high dynamic fault rates. Redundancy techniques are widely used to increase the reliability of combinational logic circuits. In this paper, a scan-chain-based multiple error recovery technique for triple modular redundancy (TMR) systems (SMERTMR) is used. The proposed technique reuses scan-chain flip-flops fabricated for testability purposes to detect and correct faulty modules in the presence of single or multiple transient faults. In the proposed technique, the manifested errors are detected at the modules' outputs, while the latent faults are detected by comparing the internal states of the TMR modules. Upon detection of any mismatch, the faulty modules are located and the state of a fault-free module is copied into the faulty modules. In case of detecting a permanent fault, the system is degraded to a master/checker configuration by disregarding the faulty module.**

Keywords—Fault-tolerant design, roll-forward error recovery, scan chain, triple modular redundancy (TMR),ScTMR,SMERTMR.

I . INTRODUCTION

Today, our society is dependent on embedded, cyber and IT systems that takeover even exceptionally critical decisions. On one hand, it reduces the risk of human error and minimizes uncertainty. On the other hand, dependability of these systems is a major concern. A variety of faults with different causes can result in system malfunctions that can lead to potentially catastrophic consequences. The rates of faults tend to increase with the increase of system complexity, decrease of semiconductor manufacturing process size, reduction of power supply voltage, and increase of operating speed. The reliance of today's embedded systems on deep sub-micron technologies has brought serious reliability challenges due to soft errors in safety critical applications. Soft errors are caused by high energy particle strikes in sensitive regions of an electronic device. Traditionally, soft errors were regarded as the main reliability threat only for space applications, but due to technology down scaling toward nanometer era, these errors have become a crucial concern in ground-level applications as well [1].

To meet the reliability requirement of safety-critical applications, embedded processors should be equipped with error detection and correction mechanisms. In addition to the reliability requirement, performance is another important issue in these applications as most of them have real-time constraints. Thus, providing fault tolerant techniques with minimum performance overhead is of decisive importance.

TMR is a well-known and widely used fault-tolerant technique [1], [2]. A traditional TMR system consisting of three redundant modules and a voter at the modules outputs has some shortcomings that should be addressed in order to be employed in safety-critical applications. A major shortcoming of the traditional TMR is its inability to cope with TMR failures [3]. TMR failure refers to a failure in a TMR system caused by multiple faulty modules or a faulty voter However, the traditional TMR is unable to recover the system state when one of its modules becomes faulty. A TMR fails if it does not properly obtain a majority decision among its modules outputs due to multiple faulty modules or a faulty voter [4].

Without an appropriate recovery mechanism, there is a high probability of having TMR failures, when running an application for a long period of time [5], [6]. To address this issue, a transient error recovery technique can be employed along with TMR technique. Normally, an error recovery technique restores the state of the system to a fault-free state once an error is detected. A majority of previous error recovery techniques in TMR systems exploit retry mechanism [4],[6],[7]. In a retry mechanism, once an error is detected, the faulty module will re-execute the entire process. Retry techniques are not suitable for tight deadline applications as they impose significant performance overhead to the system.

Rollback recovery is an alternate approach to recover from multiple fault arrivals. In rollback recovery techniques [8], the system state is backed up in some points during a program execution. Whenever an error is detected, the system is restored to the previous fault-free checkpoint. Check pointing and re-computation impose significant performance overhead which may violate the real-time requirements of safety-critical applications. In contrast, the roll forward recovery mechanisms are efficient to be used in tight deadline applications as they do not rely on re-computation. A roll-forward recovery technique has been presented in [5].The proposed technique, however, would require detailed information about the module function and cannot be applied to general purpose circuits such as processors. A TMR-based technique applicable to general-purpose circuits has been proposed in [9]. The proposed technique, called ScTMR, provides recovery for both transient and permanent errors in TMR systems [9]. ScTMR uses a roll-forward approach and employs the scan chain implemented in the circuits for testability purposes to recover the system fault-free state. Though ScTMR reduces the probability of TMR failures, it has two major shortcomings.

1. ScTMR cannot recover a single faulty module in the TMR system in the presence of latent faults (faults that are not transferred to the output).
2. ScTMR is unable to recover the system if multiple faults are simultaneously manifested to the outputs of two modules.

In this paper, technique called scan chain-based multiple error recovery TMR (SMERTMR) is employed. It has the ability to locate and remove latent faults in TMR modules as well as to recover the system from multiple faults affecting two TMR modules. SMERTMR reuses the available scan chains devoted for testability purposes in order to compare the internal states of TMR modules to locate and restore the correct state of faulty modules using the state of non-faulty modules. In case of permanent faults, the faulty module is disregarded and the system is degraded to a master/checker (M/C) configuration. As compared to other TMR-based recovery techniques, SMERTMR has negligible area overhead, as it reuses the available resources within the circuit.

The SMERTMR technique has been analytically and experimentally evaluated by implementing it in fault tolerant design of a combinational circuit. The result is compared with that of the previous ScTMR technique. As a case study the proposed technique was implemented on combinational circuit of an adder. The proposed analytical study shows that, in the presence of multiple errors, SMERTMR improves the reliability of TMR systems.

II. OVERVIEW OF ScTMR TECHNIQUE

The ScTMR technique reuses scan chains for recovering the state of the faulty module. Scan chain [9] is a cost-effective technique used in *Design for Testability* (DFT) 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 in large designs multiple scan chains are used. The number of flip-flops in each scan chain is called scan chain length (Lsc) and the number of scan chains in parallel is named scan chain width (W_{sc}).

Figure 1 shows the ScTMR block diagram. As shown in this figure, the ScTMR architecture consists of three redundant modules, a voter and a ScTMR controller. The proposed voter detects errors and reports them to the ScTMR controller. The ScTMR controller detects the error type and exploits an appropriate mechanism to remove the error effects from the system. The ScTMR controller then uses scan chains to copy the state of a fault-free module into the faulty module. The scan chain signals including Scan Chain Input (SCI), Scan Chain Output (SCO), and Scan Chain Enable (SCE) are controlled by the ScTMR controller.

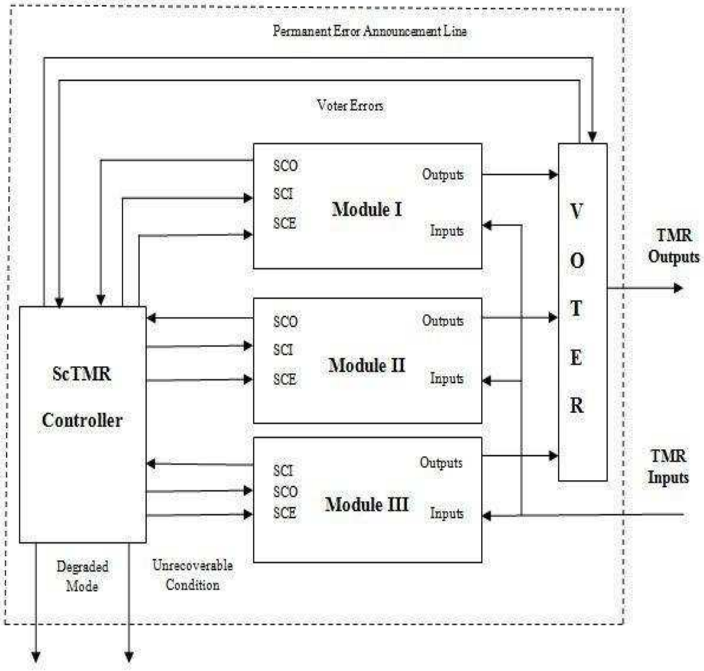

Fig.1. ScTMR Block Diagram

A. ScTMR Voter

Voter is employed to find the faulty module. The proposed voter has the capability of detecting and locating faults within the comparators, too. Figure 2 shows the proposed 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 showing mismatch between TMR modules are generated accordingly (TE_{12}, TE_{13}, and TE_{23}). If one of the TMR modules becomes faulty and the fault is not overwritten within the module, an error is manifested 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.

The voter takes three other input signals denoted with Pr_{12}, Pr_{13}, and Pr_{23}. These input signals are used to recover from permanent faults. These signals, which are derived by the ScTMR controller, are set to zero before a fault is identified as a permanent fault. In this case, E_{13}, E_{12}, and E_{23} are equal to TE_{13}, TE_{12}, and TE_{23}, respectively. The error signals (E_{13}, E_{12}, and E_{23}) are connected to both the output selector circuit and the ScTMR controller. The output selector circuit, shown in

Figure 2, selects the error-free output. The faulty module and the voter output are identified using different values of error signals according to Table I.

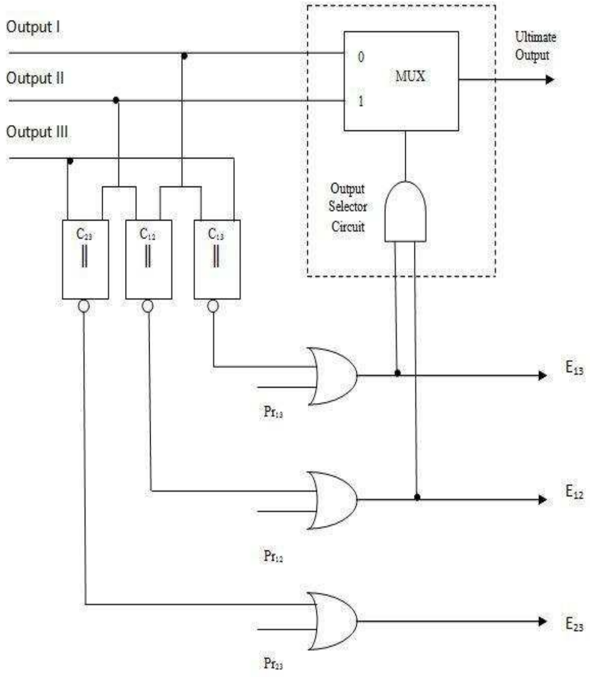

Fig. 2. ScTMR Voter

The output selector circuit can be simply implemented by a 2-to-1 multiplexer whose selector signal can be designed as depicted in Figure 2.

TABLE I
IDENTIFYING FAULTY MODULE AND SELECTING CORRECT VOTER OUTPUT

E_{12}	E_{13}	E_{23}	Faulty module	Output
0	0	0	-	Output I
0	0	1	C23	Output I
0	1	0	C13	Output I
0	1	1	Module III	Output I
1	0	0	C12	Output I
1	0	1	Module II	Output I
1	1	0	Module I	Output II
1	1	1	Unrecoverable	X

B. Error recovery mechanism

When the voter detects an error, it activates an error signal to alert the ScTMR controller. On the activation, the ScTMR switches from normal operation to recovery mode to restore the state of the faulty module.

Figure 3 shows the simplified block diagram of ScTMR controller circuit in recovery mode. In recovery mode, the internal states of two fault free modules are shifted out using scan chains and the states of one of the fault free module is copied to the faulty module. ScTMR controller enables modules scan chains and configures the multiplexers as follows: 1) the SCI signal of each fault-free module is connected to its SCO and 2) the SCI signal of faulty module is connected to the SCO of one of the fault-free module. The state of a fault-free module can be copied into the state of the faulty module after shifting the fault-free module by Lsc clock cycles. Upon activation of the recovery mode, the counter is first loaded by Lsc. Afterwards; the counter is decremented by one at each clock cycle. Once the counter reaches zero, the recovery process will be completed.

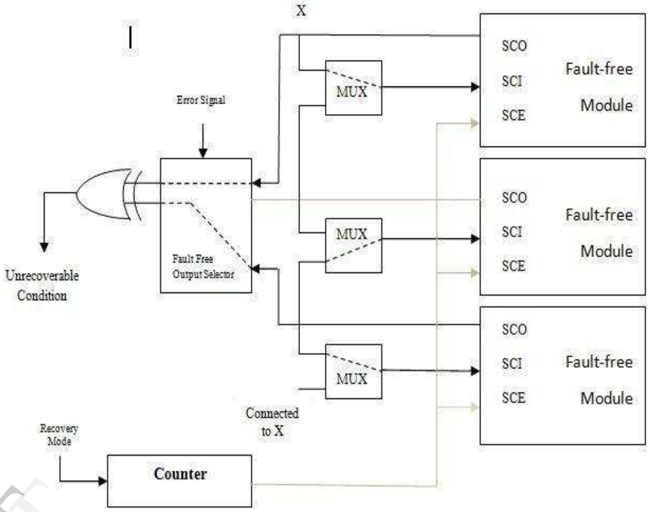

Fig. 3.ScTMR Controller in Recovery mode

ScTMR employs two internal registers named most recent faulty module (MRFM) and number of consecutive faults (NCF). MRFM holds the faulty module number detected most recently. Whenever, the value of NCF exceeds a predefined threshold value, the module is considered as a permanently faulty module. In this case, the faulty module is disregarded and the system is degraded to an M/C configuration.

III. ARCHITECTURE OF SMERTMR

Latent faults could not be identified by ScTMR technique since only the output states are compared. SMERTMR compares the internal states of the modules to detect the latent faults. In the SMERTMR technique, the comparison mode is activated in two cases: 1) when an error is detected by the voter and 2) when the checkpoint signal is activated. In the latter case, the checkpoint signal is employed to intentionally trigger the comparison mode in order to eliminate latent faults. The advantages for the SMERTMR technique are

1) If there is enough slack time to regularly perform the comparison process, the probability of having multiple faulty modules is significantly reduced.

2) Two faulty modules can be efficiently detected and located. Therefore, the faulty modules can be recovered using the states of the fault-free module.

Fig. 4 shows the state diagram of SMERTMR. In the normal mode, upon detection of an error by the voter or activation of the checkpoint signal, the system switches to the comparison mode. In the comparison mode, the internal states of the TMR modules are compared with each other to locate faulty modules and to determine the fault type. If no mismatch is found between all comparison pairs of the modules, the system returns to its normal mode. Otherwise, the system switches to the recovery mode and the recovery process is started. Finally, if the recovery process finishes successfully, the system continues to operate in the normal mode Otherwise, it enters the unrecoverable condition state resulting in a system halt. SMERTMR can also detect permanent faults in one module during the comparison mode. If it detects a permanent fault, the system enters the M/C mode. In the M/C 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.

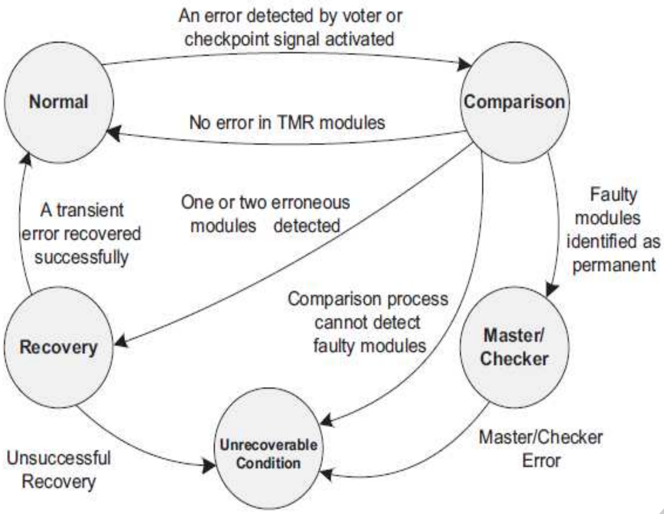
An error detected by voter or checkpoint signal activated
Normal
Comparison
No error in TMR modules
A transient error recovered successfully
One or two erroneous modules detected
Faulty modules identified as permanent
Comparison process cannot detect faulty modules
Recovery
Master/ Checker
Unrecoverable Condition
Unsuccessful Recovery
Master/Checker Error

Fig. 4.SMERTMR State Diagram

A. Comparison Process

In the SMERTMR technique, whenever the voter detects an error, it activates an error signal to alert the SMERTMR controller. Upon activation of the error signal, the SMERTMR controller switches from the normal operation to the comparison mode to locate the faulty modules. After locating the faulty modules, SMERTMR switches to the recovery mode to recover the faulty modules using the state of one of the fault-free modules. Fig. 5 shows a simplified block diagram of the SMERTMR controller circuit working in the comparison mode. In this mode, the internal states of all TMR modules are shifted out using the scan chains and all module pairs (I/II, I/III, and II/III) are compared. As shown in Fig. 5, there are three counters, namely, counter12, counter13, and counter23, to store the number of mismatches between each module pairs.

The SMERTMR controller enables scan chains of the SMERTMR modules and configures the multiplexers in such a way that the SCO signal in each module is connected to the SCI signal of the same module. During the shift operation, the internal states of the modules are compared using XOR gates. Whenever a mismatch is detected, the corresponding counter is incremented by one unit. Using this configuration, counter$_{ij}$ will contain the number of mismatches between modules i and j after Lsc clock cycles.

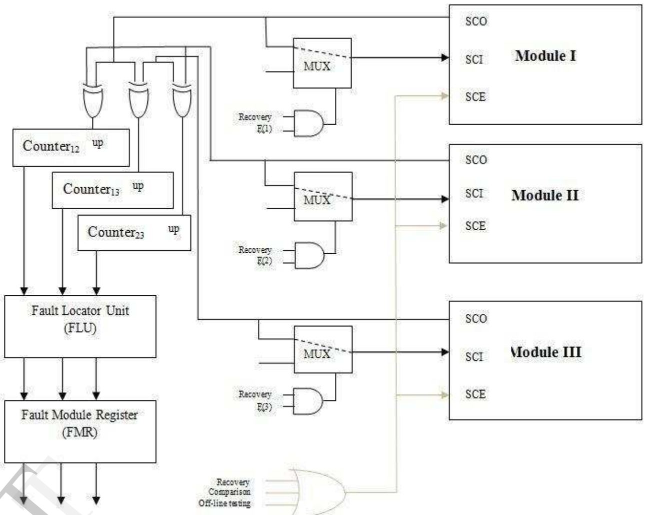

Fig. 5 SMERTMR in Comparison Mode

Suppose that there is an SMERTMR system including three modules named i , j, and k. Basically, the system may be in the following four situations.

1) When all modules are fault free then three counters will be zero
2) When there is one faulty module say module i with x erroneous flip-flops, then counterij =counter$_{ik}$ = x.
3) When there are two faulty modules say module i and module j having x and y erroneous flip-flops respectively. When there are no common erroneous flip-flops, then counter$_{ik}$ = x, counter$_{jk}$ = y, and counter$_{ij}$ = $x + y$. when there is no common erroneous flip-flops , then the counters will have the following values: counter$_{ik}$ = x, counter$_{jk}$ = y, and $0 <$ counter$_{ij}$ $< x + y$. By comparing the values of the counters, SMERTMR can effectively detect and locate the faulty modules.
4) When all modules are faulty, SMERTMR is not able to locate the faulty modules and it enters the unrecoverable condition

Upon completion of the comparison mode, the fault locator unit (FLU) will determine the faulty modules. Algorithm 1 outlines how faulty modules are detected by the FLU

B. Transient and Permanent Error Recovery Mechanisms

At the end of comparison mode, FLU detects the faulty module and the system enters the recovery mode if there is one or two faulty modules in the system. Otherwise, it returns to the normal mode. In the recovery mode, the state of the faulty module is recovered by the state of fault-free modules using the employed scan chains. Fig. 6 shows a simplified block diagram of the SMERTMR controller circuit in the recovery mode.

Algorithm 1 . Faulty Module Detection

```
If $Cntr_{ij} = Cntr_{ik} = Cntr_{jk} = 0$ then
    Next_state __ normal
Elseif ∃ i,j,k: ($Cntr_{ij} = Cntr_{ik}$ ) & $Cntr_{jk}= 0$ then
    Next_state ◄ recovery.
    FMR = i
Elseif ∃ i,j,k : ( $Cntr_{jk} = x$ ) & ( $Cntr_{ik} =y$) &
    ( $Cntr_{ij} = x+y$ ) then
    next_state ◄ recovery
    fmr =  i+j
Else
    next_state  ◄ unrecoverable condition
endif
```

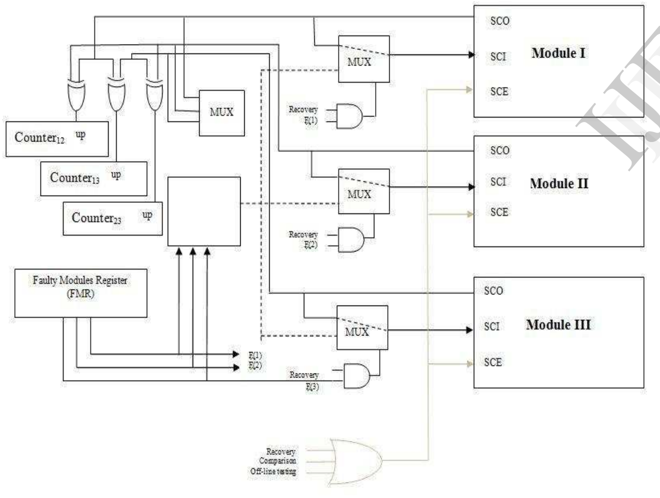

Fig. 6. SMERTMR in Recovery Mode

SMERTMR configures the scan chain like the SCI signal of fault-free modules is connected to the SCO signal of the same module. In addition, the SCI signal of the faulty module is connected to the SCO of one of the fault-free modules. The state of one of the fault-free modules is copied into the faulty modules after Lsc clock cycles. During the recovery process, the counter value will be decremented by one for each value, at the end of the recovery the counter value will be zero. If any of the counter is not zero it indicates the occurrence of fault. In that case, SMERTMR system would enter the unrecoverable condition.

SMERTMR exploits the history of faulty modules to detect permanent faults. The only difference between ScTMR and SMERTMR in permanent error detection is that ScTMR stores the module status (faulty or fault-free) based on the voter results in the MRFM register, whereas SMERTMR stores the comparison process results (saved in FMR) in the MRFM register. If the NCF in a module exceeds a predefined threshold ($T\ R$), the module is assumed to be permanently faulty. In this case, the probability that consecutive transient faults are detected as a permanent fault is equal to $1/3T\ R-1$.

IV.EXPERIMENTAL EVALUATION

In order to evaluate the efficiency of SMERTMR, we have implemented it in a fault tolerant design for combinational circuit. Then, the area and performance overheads of the SMERTMR technique are compared with those of the traditional ScTMR. The error detection and recovery capability of SMERTMR and ScTMR are evaluated by injecting stuck-at-0 and stuck-at-1 faults.

A. Case Study

Both ScTMR and SMERTMR techniques can be employed in any design equipped with scan chains for testability purpose. To validate these techniques, we have applied both techniques to a combinational design turning it to a fault tolerant design. The combinational circuit module is replicated. In module replication, a redundant module is essentially a replica of one of the original modules in the circuit. However, newly introduced redundant modules are specially customized modules. This means that redundant modules can be customized and may not be a replica of the original module.

B. Implementation Details

We have applied both ScTMR and SMERTMR techniques to the VHDL model of the combinational logic. the circuit was simulated in ModelSim PE v 6.3 e. Error was injected to the module externally. In ScTMR the error injected in single module was successfully recovered after Lsc clock cycles. When multiple errors were injected, the system went to the halt condition. In SMERTMR, multiple errors injected were successfully recovered.Fig.7 shows the simulation result of Voter.Fig.8 shows the ScTMR Recovery output and Fig. 9 shows the SERTMR Recovery output.

The voter compares the output of each module by comparing and selects the non-faulty module accordingly. The ScTMR controller recovered single faulty modules after injecting faults. The SMERTMR controller compares each of the module outputs and will recover latent as well as multiple faults from the redundant modules. The MRFM and NCF register values are automatically updated after each recovery. Whenever the NCF register value exceeds specified threshold, the system halts.

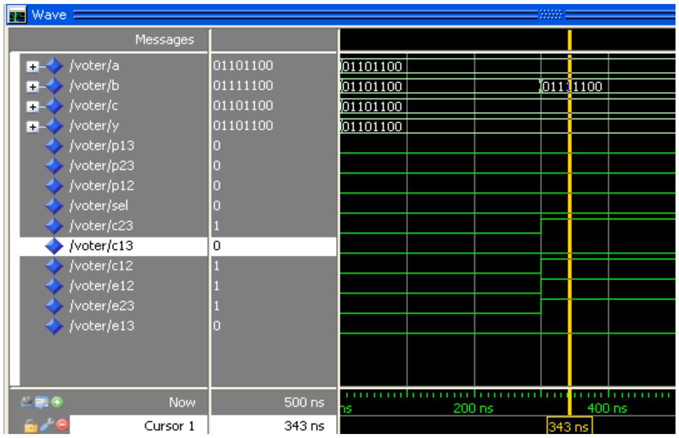

Fig.7.Simulation result of Voter Output

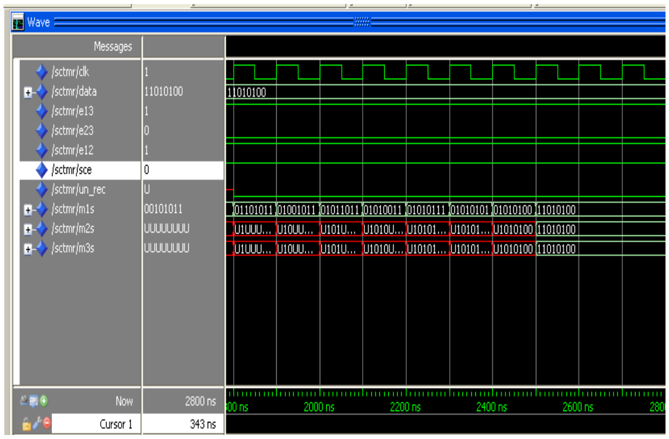

Fig.7. Simulation result of ScTMR Recovery Output

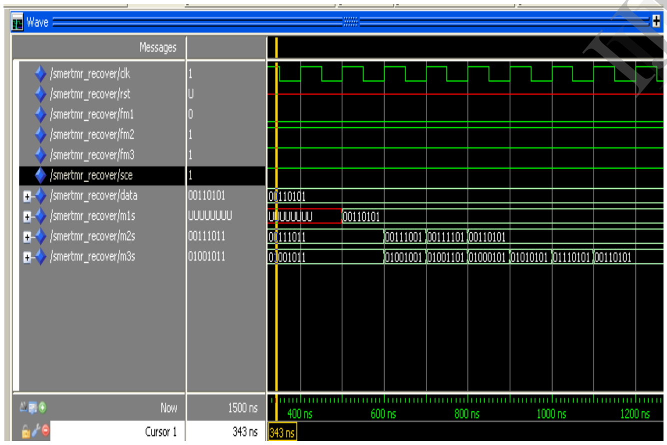

Fig.8. Simulation result of SMERTMR Recovery Output

C. Peformance Overhead

Two important factors that influence the performance of the ScTMR- and SMERTMR-based systems are the increase in the critical path delay and the time required for error recovery. The other factor influencing the performance of the ScTMR and SMERTMR-based systems is recovery time, which depends on the number of flip-flops and the width of multiple scan chains. The following equation can be used to measure the recovery time in a ScTMR-based system.

$$\text{Recovery Time} = {}_{\text{Lsc}} \times \text{Clock Period.}$$

The time required for a comparison or recovery process in the SMERTMR technique is equal to the recovery time of the ScTMR technique. This is because the scan chain is fully circulated in both techniques during the recovery process. However, the SMERTMR technique imposes twice the performance overhead to the system in the presence of a fault as compared to the ScTMR technique since both comparison and recovery processes are required for locating and correcting the faulty module in the SMERTMR technique.

D. Limitations of SMERTMR

Though SMRTMR poses many advantages, it has certain limitations, SMERTMR can be applied only to TMR systems where the three replicas are synchronized at any clock cycle and share exactly the same micro-architecture. The SMERTMR technique imposes twice the performance overhead to the system in the presence of a fault as compared to the ScTMR technique since both comparison and recovery processes are required for locating and correcting the faulty module in the SMERTMR technique.

E. Proposed System

As described earlier in section IV D, the performance overhead of SMERTMR is twice that of ScTMR since comparison and recovery processes are done separately. The proposed system implements combined comparison and recovery mode, where the recovery takes place at the time of comparison itself. The addition of spare modules along with the three redundant modules will save the system from halting whenever the threshold value of fault occurrence is reached.

V. CONCLUSION

In this paper, we presented a roll-forward technique, called SMERTMR, to recover multiple errors in TMR systems. In the proposed technique, we employed built-in design for testability resources available in combinational circuit to recover faulty modules in the presence of multiple transient faults as well as latent faults .Performance overhead of SMERTMR was found to be less than that of previous techniques.

REFERENCES

[1] 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.

[2] 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.

[3] F. Wrobel, F. Saigne, M. Gedion, J. Gasiot, and R. Schrimpf , “Radioactive nuclei induced soft errors at ground level”, IEEE Transactions on Nuclear Science , Vol.56, No.6 , pp 3437-3441, dec 2009.

[4] 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.

[5] S. Yu and E. McCluskey, “On-line testing and recovery in TMRsystems for real-time applications,” in *Proceedings of International TestConference*, 2001, pp. 240–249.

[6] K. Shin and H. Kim, “A time redundancy approach to TMR failures using fault-state likelihoods,” *IEEE Transactions on Computers*, vol.43,pp.1151-1162,Oct.1996

[7] A. Hopkins, T. Smith, and J. Lala, “FTMP: A highly reliable faulttolerant multiprocess for aircraft,” *Proceedings of the IEEE*, vol. 66,no. 10, pp. 1221–1239, oct. 1978.

[8] D. Pradhan and N. Vaidya, “Roll-forward and rollback recovery:performance-reliability trade-off,” in *Proceedings of 24th InternationalSymposium on Fault-Tolerant Computing*, jun. 1994, pp.186–195.

[9] L. Wang, C. Wu, and X. Wen, *VLSI Test Principles and Architectures:Design for Testability*. Morgan Kaufmann Publishers Inc.,2006.

[10] J. A. Abraham and D. P. Siewiorek, “*An algorithm for the accurate reliability evaluation of triple modular redundancy networks*,” *IEEETrans. Comput.,* vol. 23, no. 7, pp. 682–692, Jul. 1974.