A NOVEL MULTIPLE ERRORS RECOVERY TECHNIQUE IN TMR SYSTEMS

[1]Jyothi Joms, [2]V K Vanitha Rugmoni
[1]Department of ECE, Viswajyothi College Of Engineering, Vazhakulam, Muvattupuzha, India
[2]Assistant Profrssor, Department of ECE, Viswajyothi College Of Engineering, Vazhakulam, Muvattupuzha, India
Email: [1]jyojoms8@gmail.com

Abstract-
Embedded systems are widely used in safety critical applications like process control in industries and in medical field for patient life support monitoring etc. It should be equipped with appropriate error recovery techniques, as any type of faults in the system are in tolerable. Triple modular redundancy (TMR) is a fault tolerant system. It can mask single fault in a module but it cannot recover from the fault. A scan chain technique (ScTMR) can recover from a single fault but it cannot detect latent fault. So a scan-chain-based multiple error recovery technique for triple modular redundancy systems (SMERTMR) which 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 order to detect and correct multiple errors in SMERTMR technique there are two modes of operation which are comparison mode and recovery mode. In order to reduce the area overhead of SMERTMR a new technique called modified SMERTMR technique is proposed here. In this proposed technique the two modes are combined and the area is reduced. Upon detection of any mismatch, the faulty modules are located and the state of a fault-free module is copied into the faulty modules to recover from the fault.

Index Terms-Triple modular redundancy, scan chain based technique, roll-forward technique

I. INTRODUCTION

The use of embedded systems in safety-critical applications such as process control, and patient life support monitoring has become a common trend[1]. Such a system often has both timing constraints and fault tolerance requirements. To meet the reliability requirement, such embedded systems should be equipped with appropriate error detection and correction mechanisms.

One of the widely used fault-tolerant techniques in safety-critical applications is triple modular redundancy (TMR). Triple Module Redundancy (TMR), first proposed by Von Neumann, is one such technique where a module is replicated three times and the output extracted from a majority voter. Triple modular redundancy (TMR) is a fault-tolerant form of N modular redundancy, in which three systems perform a process and that result is processed by a majority-voting system to produce a single output[2]. If any one of the three systems fails, the other two systems can produce correct mask the fault. The main drawback of modular redundancy technique is excessive area overhead as the base design is implemented thrice. Another 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. i.e., TMR system can withstand only single upset at any instant of time, thus, if two redundant modules are simultaneously upset, then the output cannot be guaranteed to be correct[4]. Also, if two modules are permanently damaged, the whole TMR system has to be discarded[5].

A TMR-based technique applicable to generalpurpose circuits is called Scan chain based TMR (ScTMR), provides recovery for both transient and permanent errors in TMR systems. ScTMR uses a rollforward approach and employs the scan chain implemented in the circuits for testability purposes to recover the system fault-free state. Although ScTMR significantly reduces the probability of TMR failures, it suffers from two major shortcomings. First, ScTMR cannot recover a single faulty module in the TMR system in the presence of latent faults. A fault is referred to as latent if it is not propagated to the system outputs but does cause a mismatch between the states of the TMR modules. Note that, in the presence of a mismatch between the states of the TMR modules, once an error is detected at the output of either of the modules, the system will fail to restore its fault-free state. Second, ScTMR is unable to recover the system if multiple faults are simultaneously manifested to the outputs of two modules.

A scan-chain-based roll-forward error recovery technique for TMR-based systems, which addresses the shortcomings of ScTMR is presented. The proposed technique, called scan chain-based multiple error recovery TMR (SMERTMR), 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 is the first roll-forward error recovery technique for a TMR-based system that has the capability of error recovery in the presence of multiple latent faults as well as two faulty modules. The main idea behind SMERTMR is to reuse 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 order to reduce the area of SMERTMR technique another technique called modified SMERTMR is introduced. Modified SMERTMR can be obtained by combining the comparison mode and recovery mode of SMERTMR. Thus we can reduce the area overhead and power consumption. So a modified error recovery technique is proposed here. Simulation part of the project is done in Modelsim and Xilinx. Implementation part has been done using software Xilinx 12.4 on Spartan3E FPGA kit. The rest of this paper is organized as follows. Section II, section III and section IV describes overview of this work. Implementation and simulation results are given in section V. Finally, section VI concludes this work.

II. OVERVIEW OF THE SCTMR TECHNIQUE

The block diagram of the ScTMR technique is shown in fig 1. As shown in this figure, the ScTMR includes three redundant modules, a voter and a controller[10].

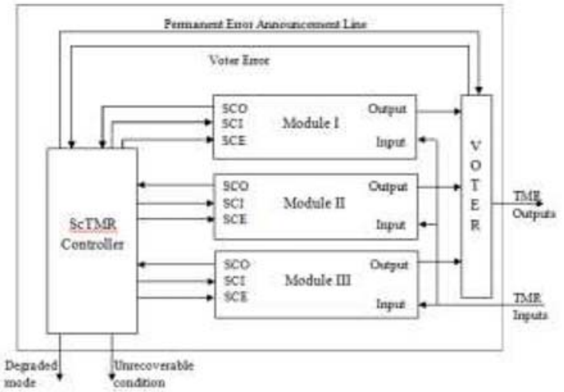

Fig. 1. ScTMR Block Diagram

In this architecture, once an error is detected by the voter, the ScTMR controller identifies the error type (transient or permanent) and triggers an appropriate recovery mechanism to eliminate the error from the system [8]. This is achieved by copying the state of a fault free module to the detected faulty module using the scan-chain circuitry. The recovery process is done through the scanchain input (SCI), scan-chain output (SCO), and scanchain enable signals instructed by the ScTMR controller.

The state diagram of a system (fig. 2) protected by the ScTMR technique is shown below. Primarily, the system is in the normal state and, upon detection of an error 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 (permanent or transient). If a permanent fault is detected in one of the modules, the system is degraded to a M/C configuration. In case of detecting a transient fault, the recovery process is performed to bring the system to the fault-free, i.e, the normal state. Upon detection of multiple transient faults, the recovery process is terminated and the system will be halted immediately to provide a fail-safe state.

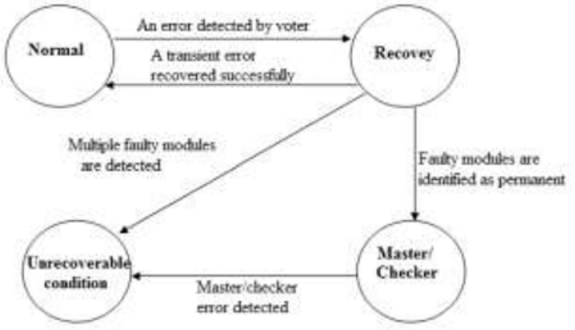
Normal
An error detected by voter
Recovey
A transient error recovered successfully
Multiple faulty modules are detected
Faulty modules are identified as permanent
Unrecoverable condition
Master/ Checker
Master/checker error detected

Fig. 2. ScTMR State Diagram

A. Voter

In a TMR system, detection and correction of a faulty module is a challenging issue and is still an ongoing research[7],[8],[9]. In particular, a wrong detection or inability to locate the faulty module can significantly affect the system reliability. To address this issue, a voter that can identify the faulty module is presented here. The architecture of the voter is depicted in fig 3.

As shown in the fig 3, three comparators (C_{12}, C_{13}, and C_{23}) are used to represent any mismatch between TMR modules. As an example, TE_{23} signal is activated once a mismatch between Outputs II and III is detected. If one of the modules generates an erroneous output (e.g., Output I), two of the comparators (here, C_{12} and C_{13}) will activate the mismatch signals (here, TE_{12} and TE_{13}) and only one of the comparators (here, C_{23}) will not activate the corresponding mismatch signal (here, TE_{23}). In case of a faulty comparator (e.g., C_{13}), only the corresponding signal (here, TE_{13}) is activated and the other signals (here, T E_{12} and T E_{23}) are deactivated. This voter can also detect and recover from permanent faults. In order to detect permanent faults[6], the proposed voter employs three input signals (named Pr_{12}, Pr_{13}, and Pr_{23}), which are derived by the ScTMR controller. In the normal state and during transient error recovery process, these three signals are deactivated (Pr_{12} = Pr_{13} = Pr_{23} = 0) and as a result, the values of E_{12}, E_{13}, and E_{23} become equal to TE_{12}, TE_{13}, and TE_{23}, respectively. Upon detection of a permanent fault, Pr_{12}, Pr_{13}, and Pr_{23} will be activated by the ScTMR controller.

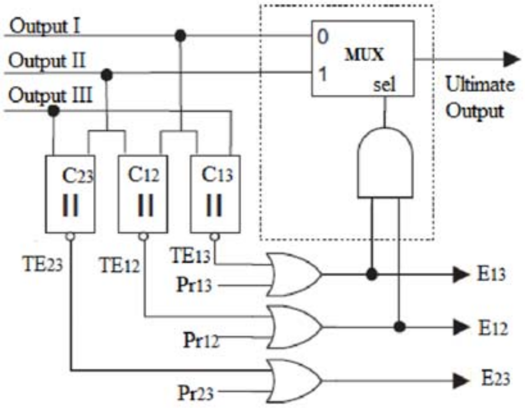

Fig. 3.Voter

B. RECOVERY MECHANISM

The ScTMR controller is used for both transient and permanent fault recovery. Once an error is detected by the voter, it alerts the ScTMR controller using an error signal.

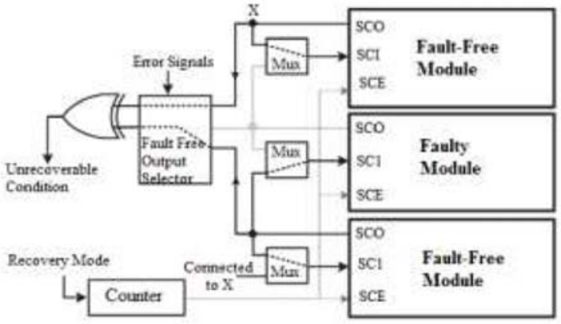

Fig. 4. ScTMR in Recovery Mode

The ScTMR controller then changes the system state from the normal operation mode to the recovery mode to restore the correct state of the system using the states of fault-free modules. Block diagram of the ScTMR controller configuration in recovery mode is shown in fig.4. During the recovery process, the flip-flop values of fault-free modules are shifted out using the scan chain and are copied to the corresponding flip-flops in the faulty module. Using this configuration, the state of a fault-free module can be copied into the state of the faulty module after shifting the fault-free module by L_{sc} clock cycles (L_{sc} is the length of the scan chains). While shifting the flipflop values, a counter is used to enumerate the number of clock cycles. Upon activation of the recovery mode, the counter is first loaded by L_{sc}. Afterwards; the counter is decremented by one at each clock cycle. Once the counter reaches zero, the recovery process will be completed. ScTMR has also the ability of distinguishing between permanent and transient faults. 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. As an example, if module II becomes faulty, MRFM is equal to 2. Upon detection of another faulty module, the faulty module number is compared with the previous faulty module number stored in MRFM. If these two numbers are equal, the ScTMR controller increments the NCF register by 1; otherwise, it resets the NCF. 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. The limitations of ScTMR are that it cannot detect latent fault and cannot recover if multiple modules are faulty.

III. ARCHITECTURE OF SMERTMR

ScTMR technique can recover from a transient fault only if it manifests in the module outputs. This is because only modules outputs are compared by the voter. In the ScTMR technique, if a fault occurs in one of the TMR modules while there is a latent fault in the other modules, ScTMR fails to recover the correct state of the system because of having two faulty modules. In this case, during the recovery operation, ScTMR detects that there are two faulty modules and enters the unrecoverable condition. Unlike the ScTMR technique that only compares the modules outputs; the SMERTMR technique compares the internal states of the modules to detect possible latent faults.

In the SMERTMR technique, the comparison mode is activated when an error is detected by the voter. As mention earlier, in the SMERTMR technique the internal states of TMR modules are compared together, while in the ScTMR technique only the module outputs are compared by the voter. Fig 5 shows the state diagram of SMERTMR. In the normal mode, upon detection of an error by the voter, 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.

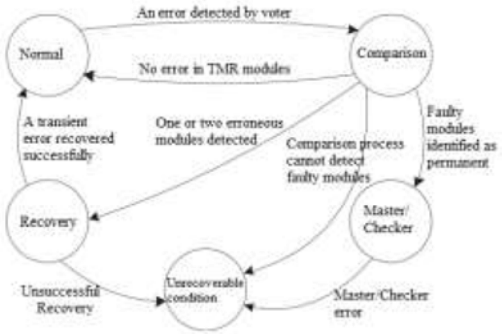
An error detected by voter
Normal
Comparison
No error in TMR modules
A transient error recovered successfully
One or two erroneous modules detected
Comparison process cannot detect faulty modules
Faulty modules identified as permanent
Recovery
Master/ Checker
Unrecoverable condition
Unsuccessful Recovery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

Fig. 5. 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.

A simplified block diagram of the SMERTMR controller circuit working in the comparison mode is shown in fig 6. 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 6, there are three counters, namely, counter$_{12}$, counter$_{13}$, and counter$_{23}$, to store the number of mismatches between each module pairs.

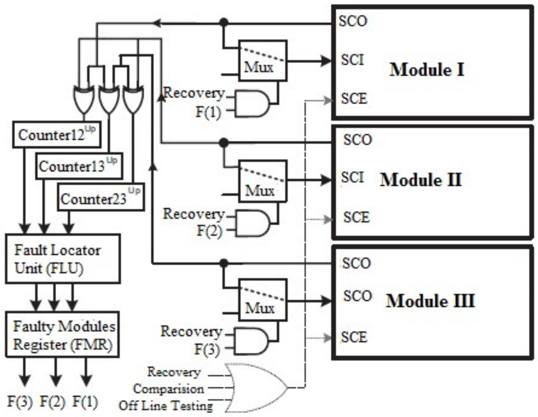

Fig. 6. SMERTMR in Comparison Mode

SMERTMR system can detect and locate faulty modules by comparing the internal states of the system modules. 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. All modules are fault-free: In this case, all three counters will be equal to zero.
2. There is only one faulty module: Let us assume that module i is faulty and it contains x erroneous flip flops and the other modules, i.e., modules j and k are fault-free. In this case, we will have counter$_{i\,j}$ = counter$_{ik}$ = x. Note that, since both modules j and k are fault-free, counter$_{jk}$ will be equal to zero (i.e., counter$_{jk}$ = 0). After extracting the number of mismatches, the system enters the recovery mode and the state of module i is recovered using the state of either modules j or k.
3. There are two faulty modules: Suppose that there are two faulty modules (e.g. modules i and j) and one faultfree module (here, module k). Let A and B be the sets of erroneous flip-flops in modules i and j , respectively. Here, the faulty modules may have either no common erroneous flip-flops (A∩B=Ø) or at least one common erroneous flip-flop (A∩B≠Ø). Assume that the number of erroneous flip-flops in modules i and j are denoted with x and y, respectively. In case A∩B=Ø, counter$_{ik}$ = x, counter$_{jk}$ = y, and counter$_{ij}$ = x + y. In case A∩B≠Ø, 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 when there is no common erroneous flipflop in the faulty modules (A∩B=Ø). In the latter case, where there is at least one common erroneous flip-flop (A∩B≠Ø), SMERTMR is not able to locate the faulty modules.
4. All modules are faulty: In this case, SMERTMR is not able to locate the faulty modules and it enters the unrecoverable condition. So the system gets halts and further process will not happen and this condition is worse.

Algorithm for Faulty Modules Detection is given below:

```
1.  Cntr_ij=Cntr_ik=Cntr_jk=0 then
2.  next_state  Normal
3.  else if i,j,k : (Cntr_ij=Cntr_ik) & (Cntr_jk=0) then
4.  next_state Recovery
5.  FMR i
6.  else if i,j,k : (Cntr_jk=x) & (Cntr_ik=y) & (Cntr_ij=x+y) then
7.  next_state Recovery
8.  FMR i,j
9.  else
10. next_state  Unrecoverable condition
11. end
```

In the SMERTMR technique, upon completion of the comparison mode, the fault locater unit (FLU) will determine the faulty modules. Algorithm outlines how faulty modules are detected by the FLU. As can be seen, if all counters become zero, there is no faulty module and consequently the system returns to its normal mode. The condition statement in line 3 checks the existence of one faulty module. If there is only one faulty module, two out of three counters will have the same non-zero value while the third counter will be equal to zero. The condition statement in line 6 checks the existence of two faulty modules with no common faulty flip-flops. In the last two cases, the system enters the recovery mode to restore the correct state of the faulty modules using the state of the fault-free modules. If none of the previous conditions is valid, the system enters the unrecoverable condition (line 10). The FLU stores the faulty module numbers in a register named the faulty modules register (FMR). As an example, if FMR is equal to 110, it means that modules I and II are faulty. This information is used by the SMERTMR controller during the recovery mode.

B. Recovery mechanism

After the identification of fault-free and faulty modules by the FLU unit at the end of the comparison process, 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. A simplified block diagram of the SMERTMR controller circuit in the recovery mode is shown in fig 7.

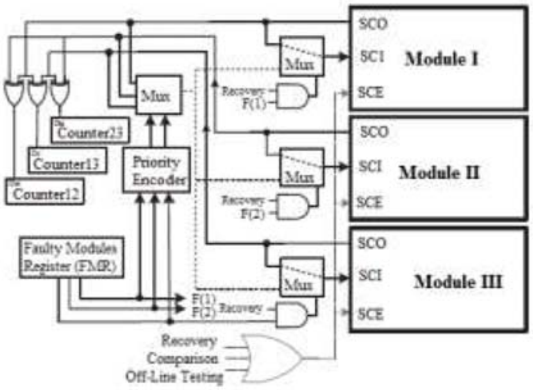

Fig. 7. SMERTMR in Recovery Mode

In recovery mode, the SMERTMR controller enables the scan chains of the SMERTMR modules and configures the multiplexers as follows: 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. As shown in fig 7, the value of the FMR register is used in the recovery mode to select the incoming driver of the appropriate signal driver for the SCI signals. Using the configuration shown in fig 7, the state of one of the faultfree modules is copied into the faulty modules after L_{sc} clock cycles.

While shifting out the states of the modules in the recovery mode, similar to the comparison mode, they are also compared to find any mismatch due to faults occurring in the recovery process. During the recovery process, whenever a mismatch is detected, the corresponding counter value containing the number of mismatches is decreased by one unit. At the end of the recovery process, all counters should be zero. This is because, for each mismatch, the corresponding counter is incremented by one unit during the comparison process and is decremented by one unit during the recovery process. If either of the counters is nonzero at the end of the recovery process, it is indicative of another fault occurrence during the recovery process. In this case, the SMERTMR system would enter the unrecoverable condition since such faults cannot be located. Permanent error detection in SMERTMR is similar to that in the ScTMR technique. 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.

IV. ARCHITECTURE OF PROPOSED SMERTMR

Proposed SMERTMR is an improvement from the existing SMERTMR technique. It addresses the disadvantages of SMERTMR. The fault tolerant system should occupy only minimum area and use less power. The proposed system can detect the error and to recover the same using minimum circuitry and less time. The system also aims to deal with the condition of multiple faults arriving at multiple modules. The system is actually a TMR system with an additional controller. It is responsible for both error detection and recovery. This in turn reduces the time and area required. Comparison and recovery operations are done at the same time. After comparison of each bit, the mismatched bits are rewritten at the same time.

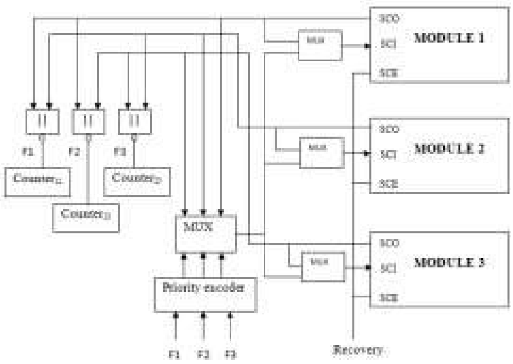

Fig 8. Block diagram of proposed SMERTMR controller

In proposed SMERTMR technique, the comparison mode and recovery mode of SMERTMR is combined to reduce the area. Mismatch detected in the comparison process is recovered at the same time. The block diagram of modified SMERTMR is shown in fig 8.The priority encoder and the mux will select the fault free output. This fault free output is connected to the scan chain input of the faulty module. By this copying the internal state of faulty module will change to fault free module. The working of modified SMERTMR is same as that of conventional SMERTMR technique but comparison process and recovery process is combined here.

Advantages of proposed SMERTMR compared to SMERTMR technique are:

1. There is minimum number of logic blocks in the new design.
2. Area required for this method is reduced.

A. Limitatons of proposed SMERTMR

Proposed SMERTMR offers features such as detection and correction of multiple bit errors and fast error recovery, but it suffers from few limitations.

- SMERTMR can be applied only to TMR systems where the three replicas are synchronized at any clock cycle and share exactly the same microarchitecture. Here, all three replicas have the same implementation and exactly the same number of flipflops.
- The three replicas must be designed and fabricated in such a ways to reduce common failure modes. Proposed SMERTMR is not applicable to those TMR systems where there is common failure state. i.e., this technique cannot recover the faulty module if two modules have same erroneous flip-flop.
- Proposed SMERTMR technique cannot be used if all three modules are faulty.

V. EXPERIMENTAL RESULTS

A. ScTMR TECHNIQUE

The simulation results for the error recovery using ScTMR technique in a sequential circuit is obtained as shown in fig 9.

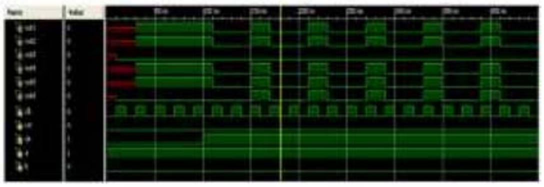

Fig. 9. Wave form of ScTMR technique

ScTMR technique can recover if single module is faulty. Here module 3 is faulty. Out1, out2 and out3 are the outputs of three modules. When se=0, out1 and out2 are 1, but out3 is 0. In the sequential circuit there are three flipflops. So when se=1, after three positive clock edge the module 3 will recover ie., out4, out5 and out6 have same value.

B. SMERTMR Technique

The simulation results for the error recovery using SMERTMR technique is obtained as shown in fig 10.

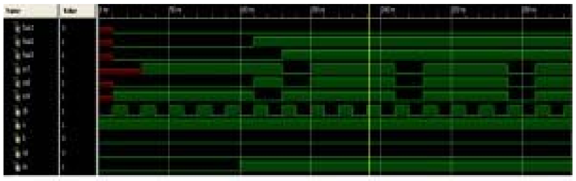

Fig. 10. Wave form of SMERTMR

SMERTMR technique can recover if multiple modules are faulty. Here module 2 and module3 are faulty. so7, so8 and so9 are the outputs of three modules. In this sequential circuit there are three flipflops. When se=0, normal operation will occur and the fault will be detected by the voter. The outputs fout1, fout2 and fout3 show the faulty module. If fout1 is high then module 1 is faulty and if both fout1 and fout2 are high then module 1 and module 2 are faulty. In this waveform fout2 and fout3 are high. So here module 2 and module 3 are faulty. When se=1, after three positive clock edge the module 2 and module3 will recover from the fault. i.e., so7, so8 and so9 have same value. After completion of scan chain technique, all outputs will have same value.

C. Proposed SMERTMR technique

The simulation results for the error recovery using proposed SMERTMR technique in a sequential circuit is obtained as shown in fig 11.

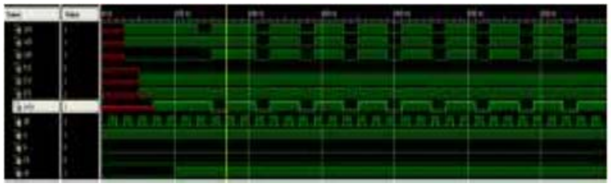

Fig. 11. Waveform of proposed SMERTMR

Proposed SMERTMR technique can recover if multiple modules are faulty. In this output waveform, module 2 and module 3 are faulty. so4, so5 and so6 correspond to the output of sequential circuit. The inputs to the sequential circuit are a and b. The outputs f12, f23 and f13 show the faulty module. If f12 is high then module 1 is faulty and if f13 is high then module 2 is faulty if both f12 and f13 are high then module 1 and module 2 are faulty. In this waveform f13 and f23 are high. So here module 2 and module 3 are faulty. The recovery process occurs during scan chain technique. When scan chain enable (se) is low, then the circuit will be in normal mode. So when se=1, the circuit goes to scan chain mode. So, after three positive clock edge the outputs so4, so5 and so6 become same. Thus module 2 and module 3 will recover from the fault. After three positive edge clock faulty modules will recover from the fault by copying internal state of fault-free module to the faulty module.

D. Comparison

The comparison diagram of conventional SMERTMR and proposed SMERTMR technique is shown in fig 12.

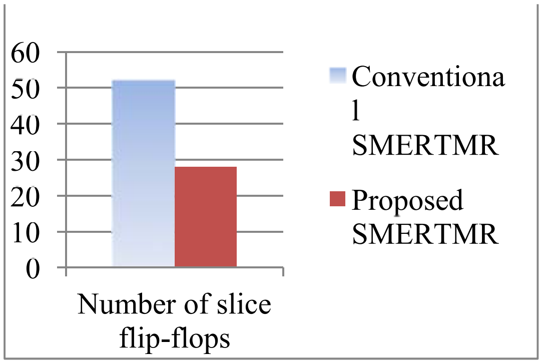

Fig 12 Comparison of conventional SMERTMR and proposed SMERTMR technique

From the above diagram the number of slice flipflops used by the conventional SMERTMR is greater than the proposed SMERTMR. Thus the area of proposed SMERTMR is less than conventional SMERTMR.

VI. CONCLUSION

The system is designed for the detection and recovery of multiple errors in TMR based systems for safety critical applications. The existing system ScTMR cannot recover faulty module in the TMR system in the presence of latent faults and cannot recover the system if multiple faults are presented. In order to overcome these limitations of ScTMR a roll-forward technique, called SMERTMR is presented here. It reuses the scan chain flip flops for the process of error detection and recovery. Here the roll forward error recovery technique is used in which the state of the fault free module is copied in to the faulty one. Thus a recovery of the same is done. It can recover multiple errors in TMR system and can detect latent faults in the module. In the proposed technique, built-in design for testability resources available in digital circuits to recover faulty modules in the presence of multiple transient faults is employed. Comparison and recovery mode of conventional SMERTMR technique is combined in modified SMERTMR technique. Number of logic blocks in modified SMERTMR is reduced, so area is reduced.

This technique can be implemented in processors where efficiency is more important than area. This technique can be used in safety critical applications.

REFERENCE

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] M. Zhang, S. Mitra, T. M. Mak, N. Seifert, N. J. Wang, Q. Shi, K. S. Kim, N. R. Shanbhag, and S. J. Patel, "Sequential element design with built-in soft error resilience," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 12, pp. 1368–1378, Dec. 2006.

[6] 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.

[7] Deepa Jose, Dr P Nirmal Kumar and A.David Naveen Dhas, "Implementation of Power optimized VLSI Designs for Reliable Processing using Majority circuit", in *Annual IEEE Indian Conference (INDICON)*, 2013.

[8] H. El-Razouk and Z. Abid, "A New TransistorRedundant Voter for Defect-Tolerant Digital Circuits", IEEE *Trans. Comput*, May 2006.

[9] P. Balasubramanian, and N. E. Mastorakis, "A Standard Cell Based Voter for use in TMR Implementation", ISBN.

[10] 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.