Scan-chain-based multiple error recovery in TMR systems (SMERTMR)

Matúš Liščinský

April 27, 2020

Faculty of Information Technology, Brno University of Technology

Introduction

Nowadays, the use of embedded systems in safety-critical applications has become a common trend. Such a system often has some fault-tolerance requirements, and to meet them:

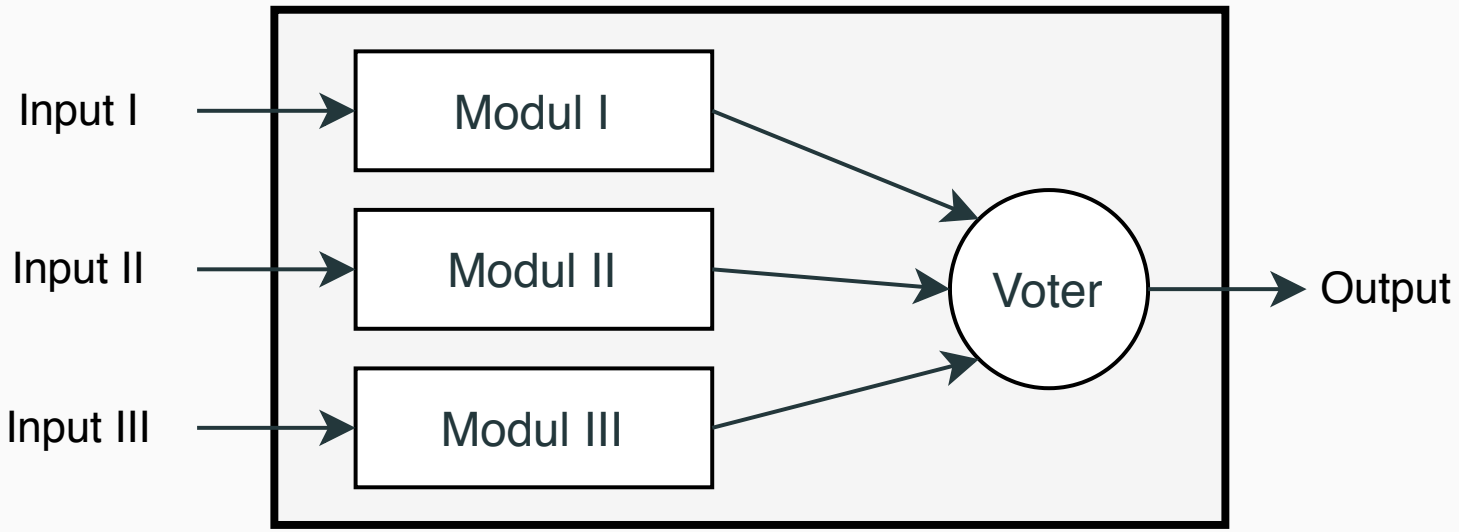
- embedded systems should be equipped with appropriate error **detection** and **correction** mechanisms
- with minimum performance overhead

One of the well-known and widely used fault-tolerant techniques is triple modular redundancy (TMR).

Triple Modular Redundancy

Traditional TMR System

- consists of three redundant modules and a voter at the modules outputs

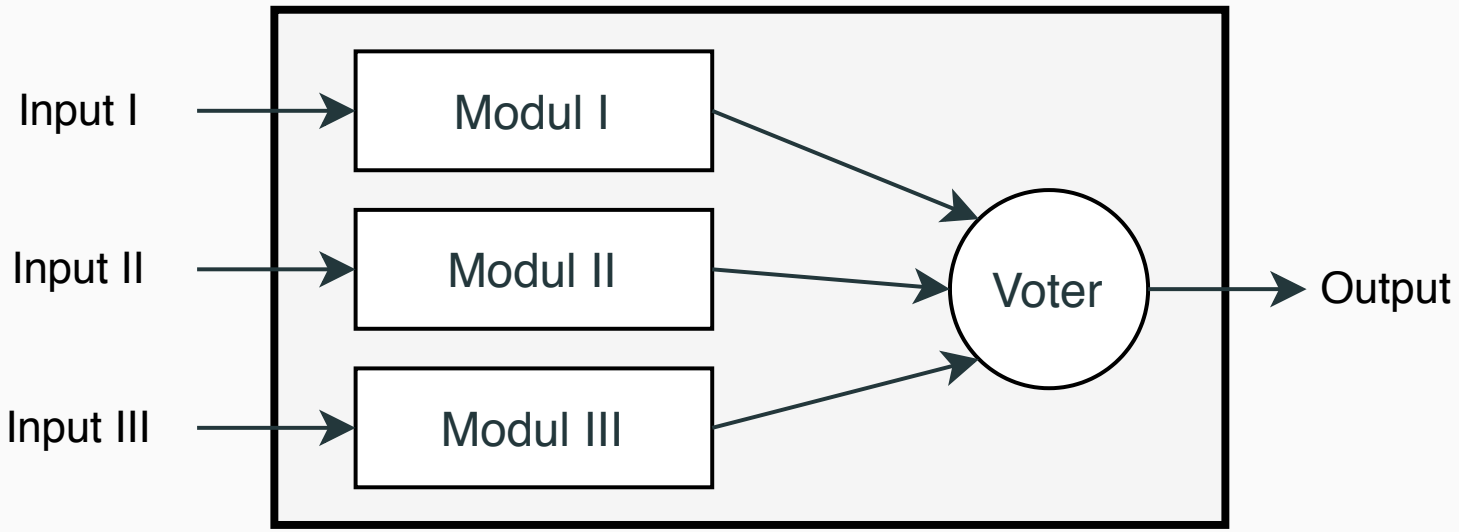


- **disadvantage:** inability to work with faulty voter or when more than one module is faulty
- the absence of appropriate recovery mechanisms increases the probability of TMR failure
- Solution:

Triple Modular Redundancy

Traditional TMR System

- consists of three redundant modules and a voter at the modules outputs



- **disadvantage:** inability to work with faulty voter or when more than one module is faulty
- the absence of appropriate recovery mechanisms increases the probability of TMR failure
- Solution: *transient error recovery technique*

Transient Error Recovery Techniques

Two approaches:

- roll-back
  - once an error is detected, the faulty module will re-execute the entire process
  - not suitable for applications with tight deadline - rollback increase the performance overhead, we may cross the deadline
- roll-forward
  - the correct state of module is copied from a fault-free module to faulty module
  - no need to re-compute, can be used in tight deadline applications

ScTMR

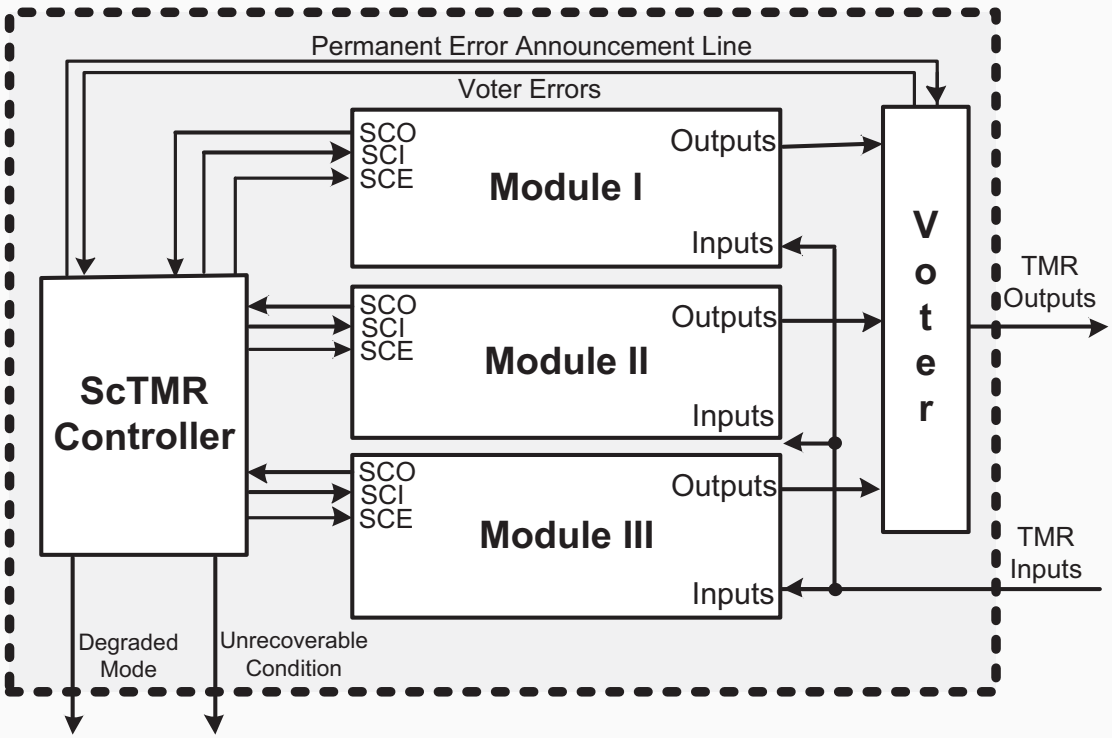
Scan **C**hain-Based Error Recovery Technique for **TMR** Systems

- roll-forward approach
- 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

- cost-effective technique implemented in the circuits to provide a simple way for testing combinational and sequential circuits
- flip-flops are chained together through a long shift register circuit
- multiplexer is used in front of each flip-flop to switch between the normal and testing operations

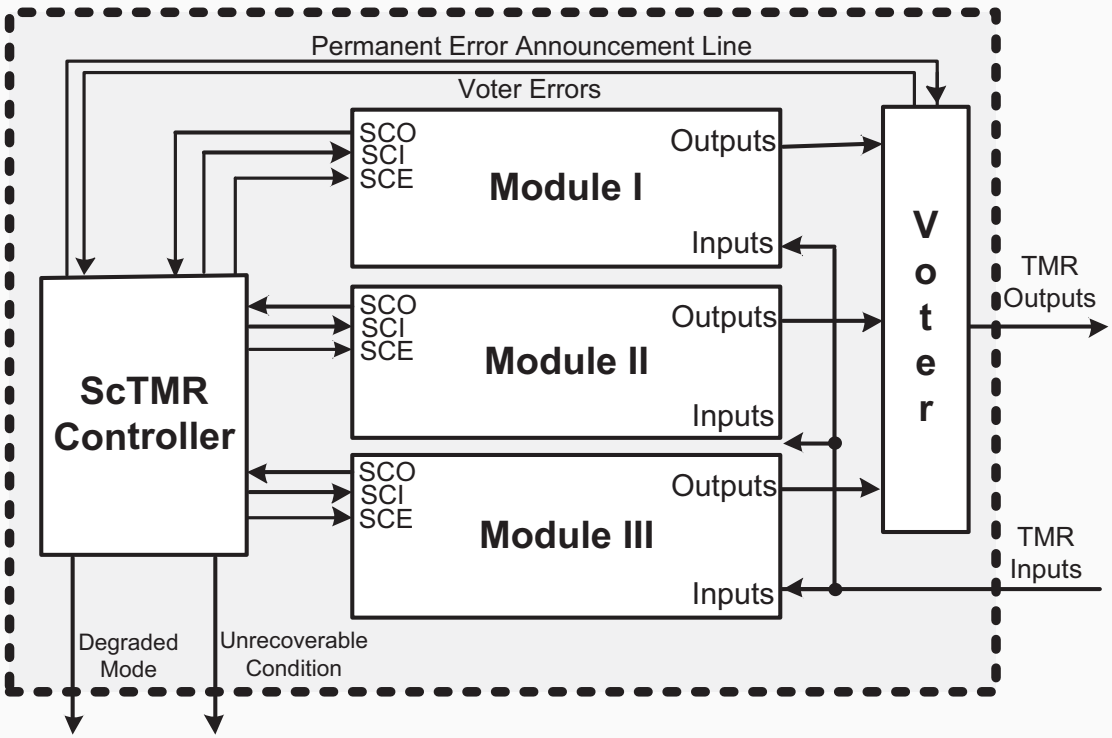
ScTMR Architecture



ScTMR architecture consists of:

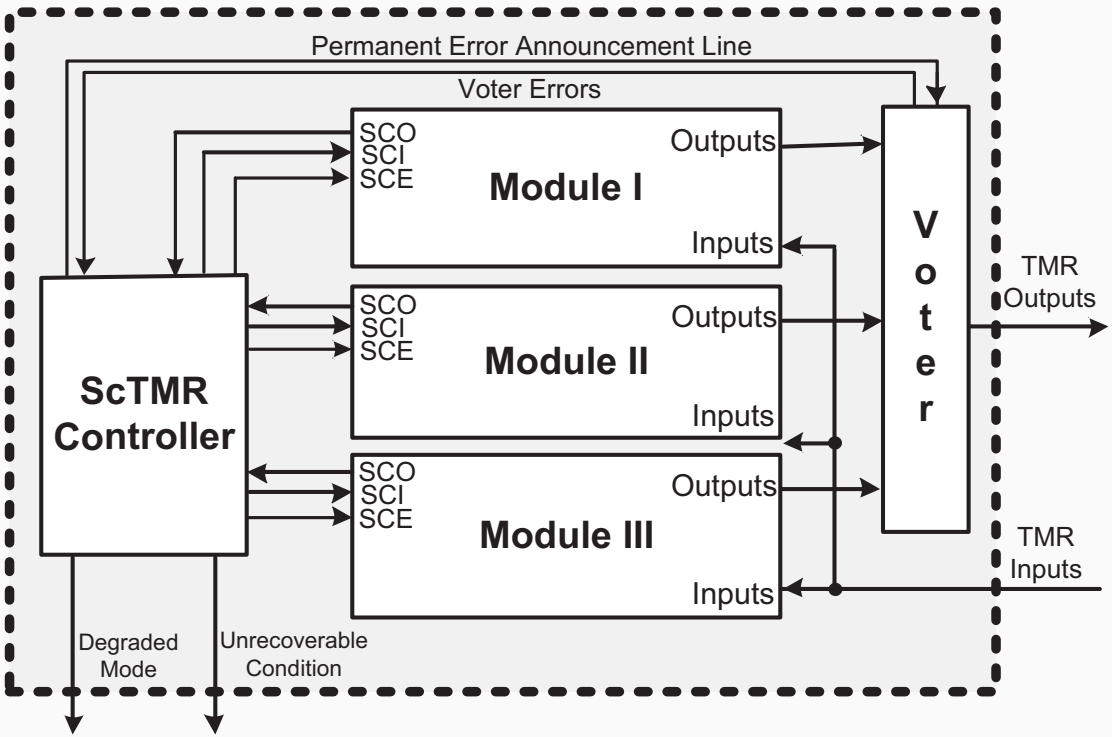
- three identical modules
- **ScTMR controller**
- **voter** - detects errors and reports these errors to the ScTMR controller

ScTMR Controller



- identifies the error type and use an appropriate mechanism to remove the error effects from the system
- uses scan chains to copy the state of a fault-free module into the faulty module
- monitors and controls the scan chain signals: SCI, SCO, SCE

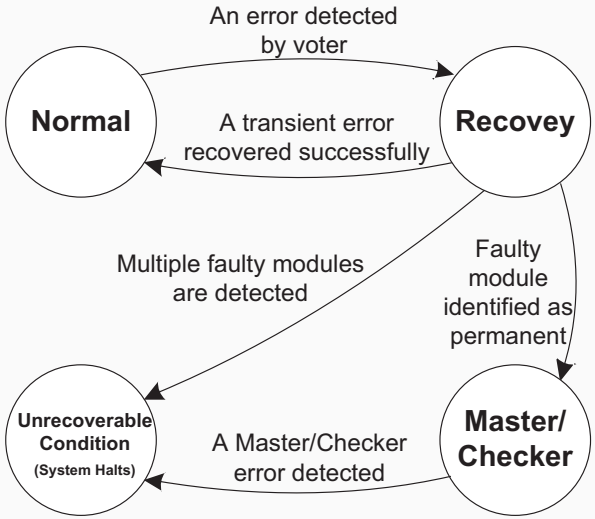
ScTMR Controller



When error is detected by the voter it gives an error signal to the controller. Then the controller 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.

ScTMR States

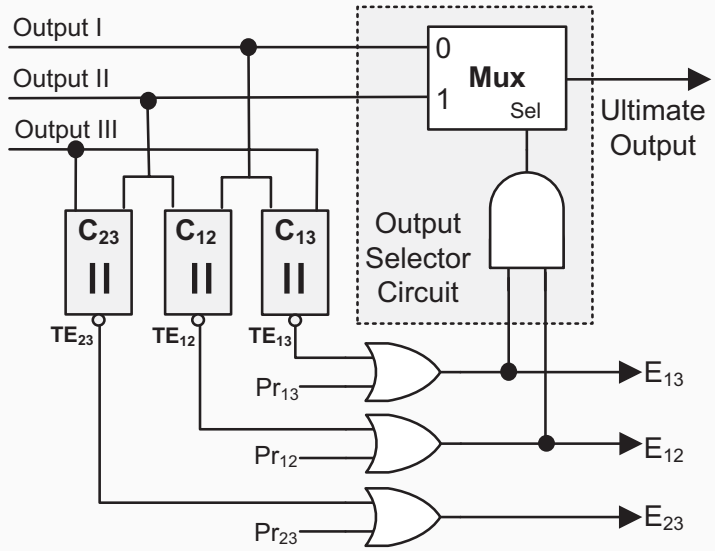
- initially - system in the ***normal*** state
- detected error by the voter - ***recovery*** process is initiated
- recovery process - ScTMR controller detects the faulty module as well as the fault type:



- **permanent fault** - degradation to a ***Master/Checker*** configuration
- **transient fault** - recovery process brings the system to the fault-free state
- **multiple transient faults** - termination of recovery process and the system halts itself immediately (***unrecoverable condition***)

ScTMR Voter

- has the capability of locating the faulty module and also detecting and locating faults within the comparators
- **comparators** (C_{12}, C_{13}, and C_{23}) compare the outputs of the TMR modules

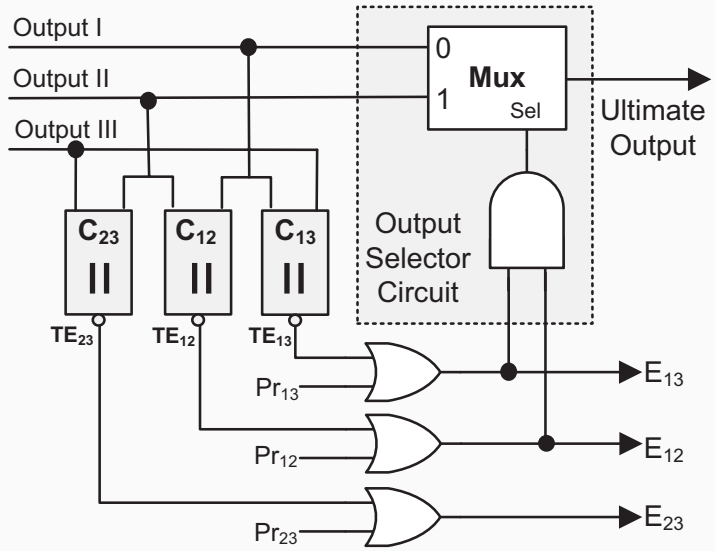


- **error signals** (TE_{12}, TE_{13}, and TE_{23}) shows mismatch between TMR modules - occurred error is manifested to the output of the faulty module and the corresponding output becomes erroneous - the error is then detected by comparators

Example: If Output I become inaccurate, the error is detected by C_{12} and C_{13} and therefore both error signals TE_{12} and TE_{13} are activated.
Question: What if one of the comparators becomes faulty ?

ScTMR Voter

- has the capability of locating the faulty module and also detecting and locating faults within the comparators
- **comparators** (C_{12}, C_{13}, and C_{23}) compare the outputs of the TMR modules



- **error signals** (TE_{12}, TE_{13}, and TE_{23}) shows mismatch between TMR modules - occurred error is manifested to the output of the faulty module and the corresponding output becomes erroneous - the error is then detected by comparators

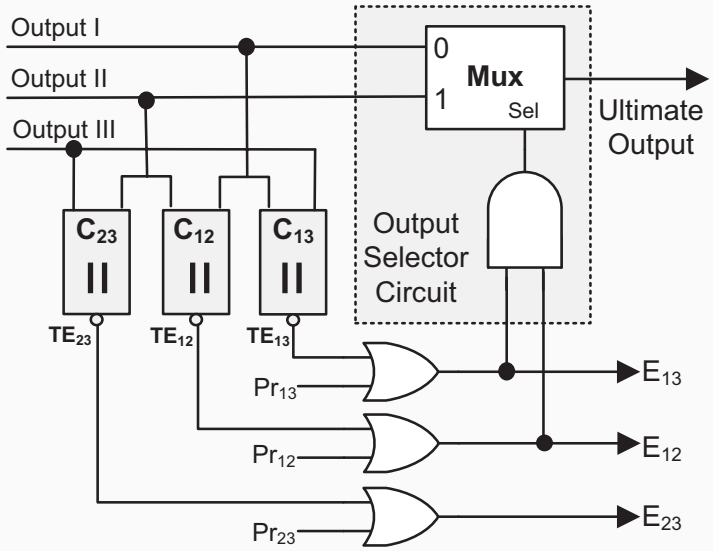
Example: If Output I become inaccurate, the error is detected by C_{12} and C_{13} and therefore both error signals TE_{12} and TE_{13} are activated.

Question: What if one of the comparators becomes faulty ?

Only the corresponding error signal is activated.

ScTMR Voter

- voter employs **three input signals** Pr_{12} Pr_{13}, and Pr_{23} for detection the permanent faults
- signals are derived by the ScTMR controller and set to zero before a fault is identified as a permanent

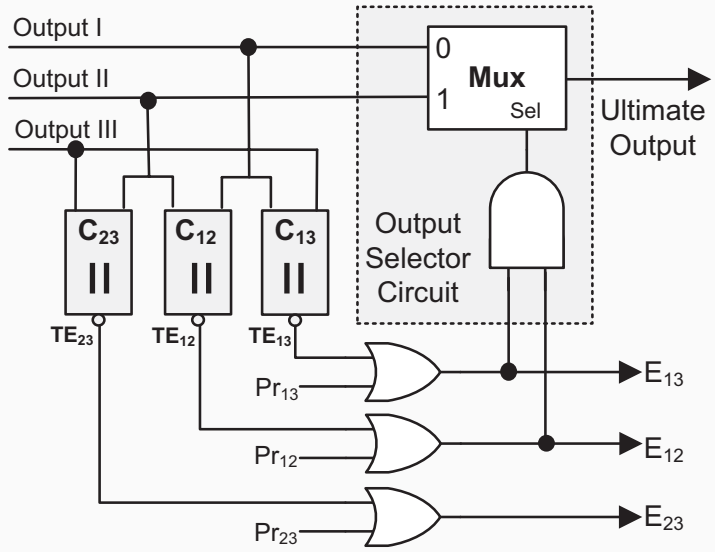


- **Output Selector Circuit**
  - implemented by a 2-to-1 multiplexer
  - selects the error-free output and gives the ultimate error-free output
  - the faulty module and the voter output are identified using different values of error signals

ScTMR Voter

Identifying faulty module and selecting correct voter output using error signals:

E_{12}	E_{13}	E_{23}	Faulty Module	Output
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



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 becomes faulty, output II will be selected by the selector circuit as the error-free output of the system. If all the error signals are active then this condition is called ***unrecoverable condition***.

ScTMR - Permanent vs Transient Fault

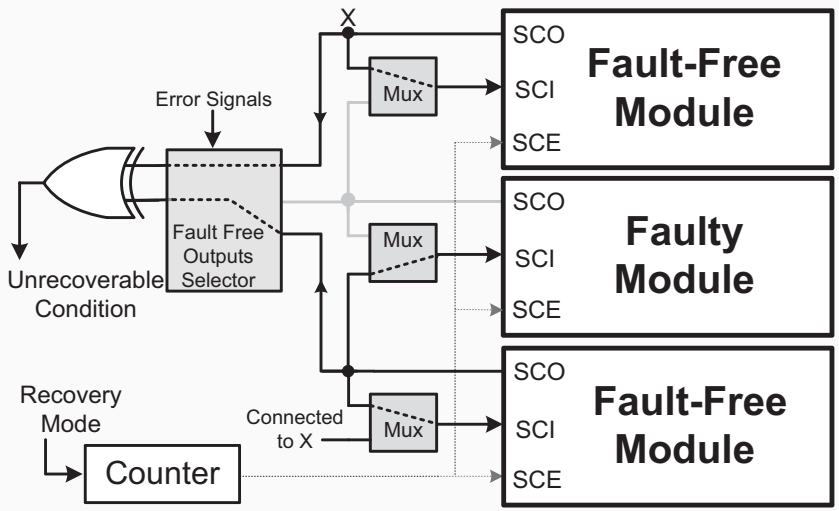
ScTMR controller can identify the fault type - permanent/transient, using two internal registers:

- ***MRFM*** - *Most Recent Faulty Module*
- ***NCF*** - *Number of Consecutive Faults*

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 ScTMR controller enables the corresponding Pr error signal and the system will go into ***Master/Checker*** configuration.

ScTMR Recovery Mode

- upon activation of the error signal by the voter, the ScTMR controller switches from ***normal*** mode to the ***recovery*** mode



- ***recovery mode***:
  - internal states of the two fault-free modules are shifted out (with scan chains)
  - if the states of fault-free modules do not match: controller will indicate ***unrecoverable condition*** (two faulty modules within the system)
  - otherwise: state of one of the fault-free modules are copied into the faulty module (using scan chains and multiplexers configuration) and system will return to its ***normal*** mode

ScTMR Shortcomings

The ScTMR technique has some **disadvantages**:

1. ScTMR cannot recover a single faulty module in the TMR system in the presence of *latent faults*
   latent fault - not propagated to the system output but causes a mismatch between the states of TMR modules
   Note: In the presence of a mismatch between the states of the TMR modules, the system will fail to restore its fault-free state when an error is detected, so the system will fail to restore its fault-free state.
2. ScTMR is unable to recover the system if multiple faults are simultaneously manifested to the outputs of two modules

Solution: more sophisticated technique

ScTMR Shortcomings

The ScTMR technique has some **disadvantages**:

1. ScTMR cannot recover a single faulty module in the TMR system in the presence of *latent faults*
   latent fault - not propagated to the system output but causes a mismatch between the states of TMR modules
   Note: In the presence of a mismatch between the states of the TMR modules, the system will fail to restore its fault-free state when an error is detected, so the system will fail to restore its fault-free state.
2. ScTMR is unable to recover the system if multiple faults are simultaneously manifested to the outputs of two modules

Solution: more sophisticated technique - **SMERTMR**

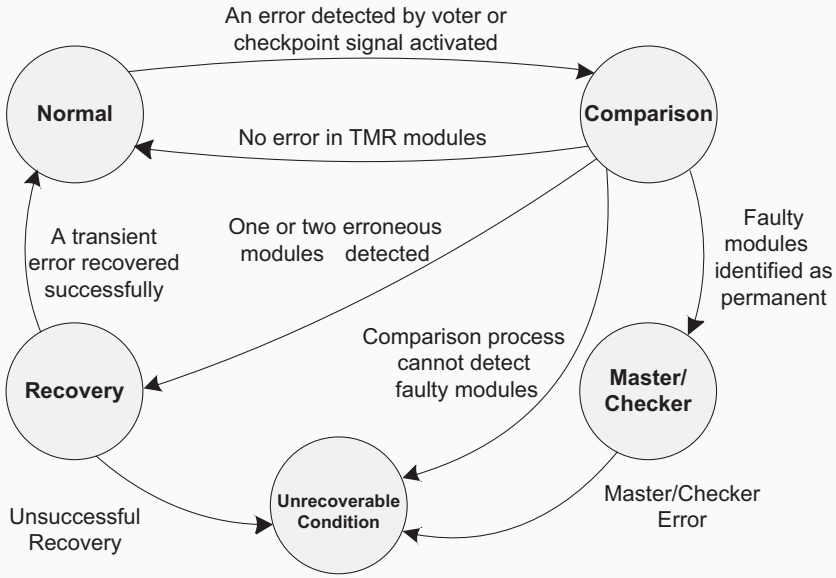
SMERTMR

Scan-Chain-Based **M**ultiple **E**rror **R**ecovery in **TMR** Systems

- ScTMR with the ability to
  - locate and remove latent faults in TMR modules
  - recover the system from multiple faults affecting two TMR modules
- employs a ***comparison mode*** in order to locate latent faults
- ***comparison mode:***
  - internal states of modules are compared with each other to extract the number of mismatches for each comparison pair
  - faulty modules are detected and located using the results of pair comparisons

SMERTMR States

- *comparison mode* is activated in two cases:
  1. when an error is detected by the voter or
  2. when the checkpoint signal is activated.
- checkpoint signal
  - activated during slack times
  - used to intentionally trigger the comparison mode in order to eliminate latent faults



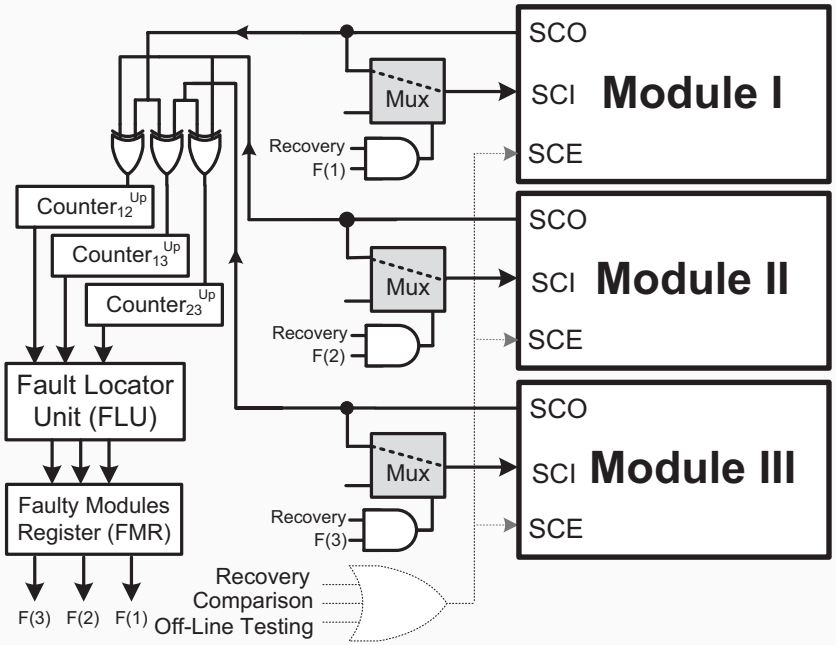
- when the the voter detects an error, it gives an error signal to SMERTMR controller, then the controller switches the mode to ***comparison*** and after detecting the faulty modules switch its state to ***recovery*** mode in order to recover the faulty modules

SMERTMR: Comparison mode

- in this mode:
  - the internal states of the TMR modules are compared with each other
  - 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
- during the comparison mode, SMERTMR can also detect **permanent faults** in one module
- in case of permanent fault, the system enters the ***Master/Checker*** (M/C) mode
- Note: 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.

SMERTMR in Comparison mode

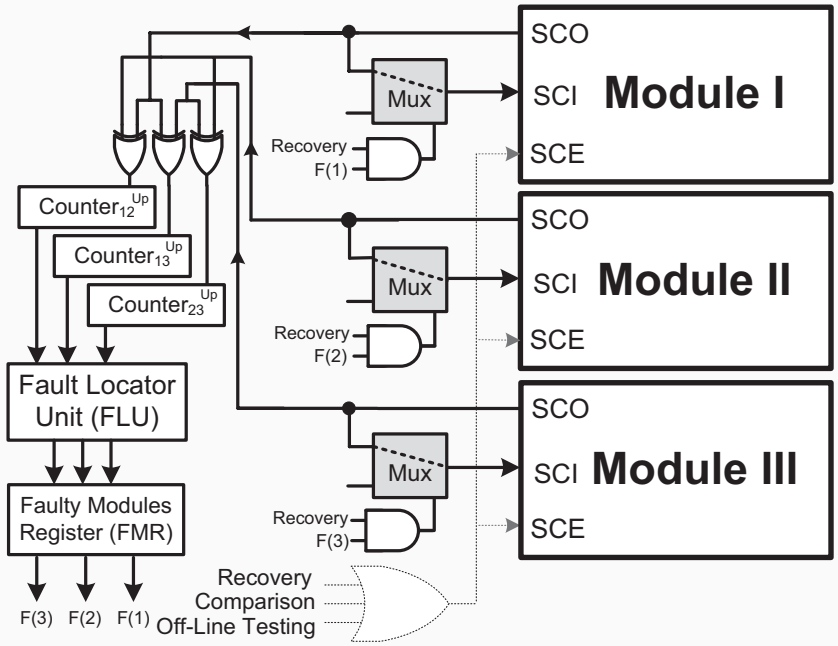
- the internal states of all TMR modules are shifted out using the scan chains and all module pairs (I/II, I/III, and II/III) are compared
- ***three counters*** $Counter_{12}$, $Counter_{13}$, and $Counter_{23}$ stores the number of mismatches between each module pairs



- Example: $Counter_{12}$ stores the number of mismatches between modules I and II

SMERTMR: Fault Location

- *FLU* - fault locator unit, determines the faulty modules as follows:
- Suppose that there is an SMERTMR system including three modules named *i*, *j*, and *k*. The system may be in the following four situations:



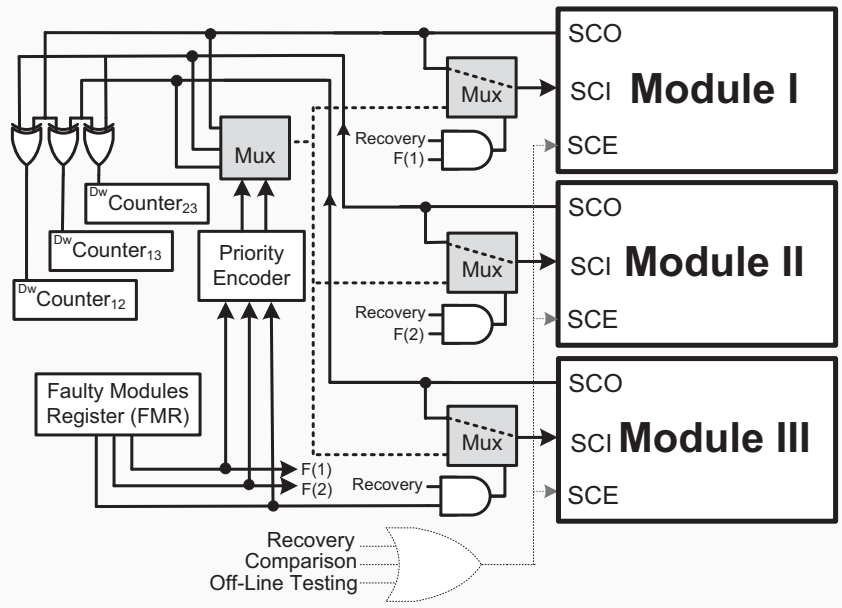
1. **all** modules are **fault-free**: all three counters will be equal to zero
2. **only one faulty** module: assume that module *i* is faulty and it contains *x* erroneous flip-flops, then $counter_{ij} = counter_{ik} = x$ the state of module *i* is recovered using the state of module *j* or *k*
3. **two faulty** modules (e.g., modules *i* and *j*) faulty modules may have either:
   - no common erroneous flip-flops or
   - at least one common erroneous flip-flop

SMERTMR: Fault Location

3. continuation
   Assume that the number of erroneous flip-flops in modules i and j are denoted with x and y, respectively.
   - In case of no common erroneous flip-flops, $\text{counter}_{ik} = x$, $\text{counter}_{jk} = y$, and $\text{counter}_{ij} = x + y$. Therefore SMERTMR **can effectively detect and locate** the faulty modules by comparing the values of the counters.
   - In case of at least one common erroneous flip-flop, the counters will have the following values: $\text{counter}_{ik} = x$, $\text{counter}_{jk} = y$, and $0 < \text{counter}_{ij} < x + y$. In this situation SMERTMR **is not able to locate** the faulty modules, because it is not distinguishable from a case in which there are three faulty modules.
4. **all** modules are **faulty**: SMERTMR is not able to locate the faulty modules and it enters the unrecoverable condition

SMERTMR in Recovery Mode

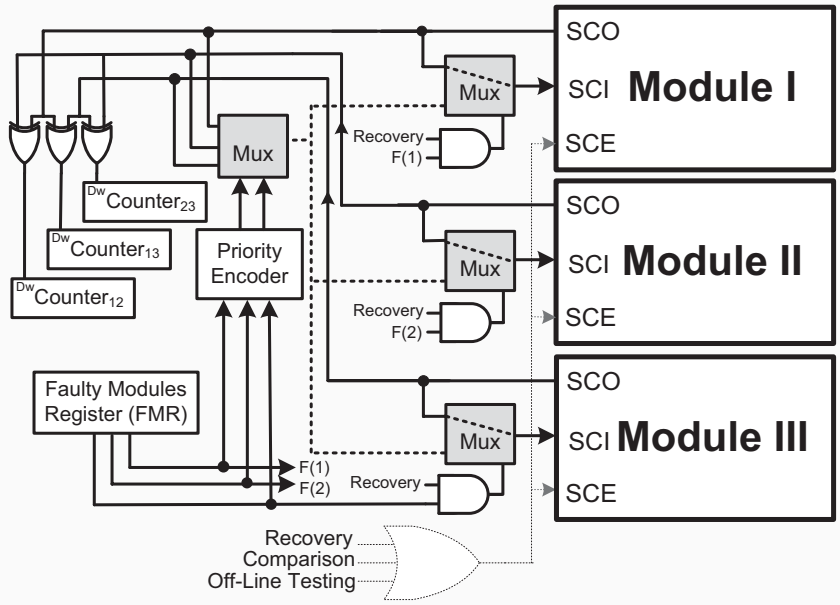
- the system enters the ***recovery mode*** if there is one or two faulty modules
- the state of the faulty module is recovered by the state of fault-free modules using the employed scan chains



- the SCI signal of fault-free modules is connected to the SCO signal of the same module, and the SCI signal of the faulty module is connected to the SCO of one of the fault-free modules
- 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

SMERTMR in Recovery Mode

- 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



- 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 and the SMERTMR system will enter the ***unrecoverable condition*** since such faults cannot be located
- **permanent error** detection in SMERTMR is similar to that in the ScTMR technique explained previously

Conclusion

EXPERIMENTAL EVALUATION

- SMERTMR detects, locates, and corrects **100%** and **99.7%** of multiple faults affecting **single** modules and **two** faulty modules, respectively
- compared to the traditional TMR system, **area** and the **performance** overheads of SMERTMR are **less than 3%** and **1%**, respectively
- analytical assessment demonstrated that SMERTMR improves the **reliability** of TMR systems by **several orders of magnitude** compared to the state-of-the-art techniques

Questions?

References

- Scan-chain-based multiple error recovery in TMR systems (SMERTMR)
  https://ieeexplore.ieee.org/document/7019111
- ScTMR: A scan chain-based error recovery technique for TMR systems in safety-critical applications
  https://ieeexplore.ieee.org/document/5763277
- Low-Cost Scan-Chain-Based Technique to Recover Multiple Errors in TMR Systems
  https://ieeexplore.ieee.org/document/6307891