10LTW98

XMR : The Incomplete TMR Solution

Jose Miguel V. Santos
FEUP/ ISEP- R. S. Tome, 4200 Porto, Portugal
jmvs@dee.isep.ipp.pt

Jose Manuel M. Ferreira
FEUP- Rua Bragas, 4050 Porto, Portugal
jmf@fe.up.pt

Abstract

XMR is a new subclass of NMR architectures: an enhanced BST infrastructure helps a duplication-based design to provide a TMR-like behavior. Error confinement and a timely recovery are possible with $X \in [2, 3[$. allowing single IC integration and easy of design.

Keywords: *Fault-Tolerance, Concurrent Test, BST, Fault-Location.*

Introduction

Dependable Integrated Circuits (IC) providing error confinement and quickly resuming the correct operation may find a wide field of application in many Real-Time designs, such as automotive, railway, medical, nuclear power plants and industrial control units. Fault-Tolerant (F-T) systems require immediate *error confinement* but, usually, a small interval is allowed to *resume operation*, provided the outputs are set to a *known-safe* state meanwhile.

Most safety-critical hardware designs are *replication* or *self-checking* architectures [1, 2], both with advantages and drawbacks, and chosen according to the features of the application. The integration of these architectures in VLSI ICs has several problems: the many gates available are not accompanied by pin number and accessibility, and high-redundancy designs are interesting but not desirable [3, 4]. Replication, being acceptable in a VLSI, is however hampered by *common mode faults (cmf)*. Self-checking properties may also be masked by many technology specific faults. When a timely recovery is acceptable, mainly in ground-based systems, the *latency* delay allows another solution to be envisaged, with the help of the Boundary Scan Test (BST) infrastructure [5].

The new architecture presented, *XMR*, may be seen as a sub-group of *N- Modular Redundancy* (NMR) designs: *a duplication design in which an enhanced BST infrastructure provides decisions when the replicas disagree*. Working alone, a XMR IC may only confine errors, but a timely recovery is available when supported by a BST-controller.

The paper presents XMR for combinatory designs and considers the extension to sequential logic.

I - Background

To begin with we must clarify the meaning of some terms in the paper:

- *CUT*: the *circuit under test*, or *mission circuit*, according to the specification.
- *Module*: an IC (or part of it) expected to behave as a Fault Containment Region (FCR) [6] in dependable systems. Many configurations are possible, but we are concerned with 2-CUT modules with BST.
- *Fault model*: as replication based designs are vulnerable to *cmf*, only single faults (permanent and temporary) are considered at first; yet, the XMR architecture is able to detect most permanent *cmf*, and they will be discussed later.

Error confinement means an *immediate* detection avoiding the error to spread out the module.

To *resume a correct operation* means not necessarily to correct the error, but only to connect the module outputs to the good CUT, always running meanwhile, as soon as the faulty CUT is located. The error will be corrected if the latency interval is shorter than the *time granularity* of the system; otherwise we may only talk about *correct recovery*, acceptable if the circuitry fed by the outputs is not sequential, as when a XMR IC drives actuators.

Error *correction* needs redundant information, and the traditional solutions are:

- *Replication*: Triple Modular Redundancy (TMR) is the most known of *NMR* architectures (N≥3). The advantages are an immediate decision (*voting*) and, since no coding circuitry is required, each *CUT* reaches the highest reliability, together with the easy of design. The main drawbacks in single IC designs are the weakness of replication architectures to *cmf*, and the hardware required, Voter included, which impacts the *yield, power dissipation* and the *MTBF*.
- *Self-Checking designs:* they confine the error but may only provide a *fail-stop* solution [7]. A F-T design requires a *duplex* Self-Checking, mandatory if permanent faults must be considered, leading the hardware overhead to approach tha' of a TMR [8].
- *Error Correcting Codes*: requiring to redesign the *CUT* deeply, and very sensitive to unidirectional errors (critical inside VLSI) they fall outside our approach.

II - Design Tradeoffs and Objectives

In order to design dependable ICs easily, with low hardware overhead and improved reliability, several features must be considered, namely:

- *VLSI ICs*: they have many advantages, but are handicapped by the faults which degrade replication and self-checking designs [9, 10], mainly *cmf* and unidirectional errors. Since VLSI designs (ASIC, PLD) tend to be pin limited, not core limited, *some redundancy* is acceptable but many reasons suggest to avoid more complex solutions [3,1,4]. As redundancy degrades yield, independent ICs are preferable for high replication designs.

- *Duplication design:* an IC with 2 *CUT* and output comparison, reaches a much better yield and wider MTBF than a TMR, but may only provide a *fail-stop* solut:on, even if one *CUT* still works correctly. Compared to self-checking design [11], duplication is an easier approach with little more hardware, even if their targets are not exactly the same.

-*Fault model*: as all replication architectures, duplication has an important characteristic: no fault model is required for single-CUT faults. A fault model is only needed to verify the design through a restricted set of patterns, and this allows a XMR module to detect permanent *cmf* "included" in the (single-)fault model defined for design verification of the CUT.

- *Concurrent Verification:* Intermittent faults are becoming increasingly important as a result of thermal problems in VLSI ICs (see III.3). Permanent faults also are a real problem in small IC process geometries. Production defects may go undetected after burn-in, and some of them may show a "memory" behavior invalidating off-line tests [12]. Concurrent monitoring is becoming mandatory.

- *Latency*: most F-T systems are real-time designs and need error confinement, but may allow a small delay to resume. A very restrictive interval is fuel control in jet-planes every 20-50ms [6], and ground-based designs may sustain many milliseconds or even seconds [13]. This acceptable latency will be exploited.

- *Design Diversity (DD)*: is always desirable but usually means an increased cost. Replication designs will benefit when DD is included to face *cmf*: a simple idea is to provide a partial DD through an already existing support, the BST infrastructure.

- *IEEE 1149.1 std compliance*: recent tendencies [14] show BST highly probable in all new designs, as a result of two strong reasons: dedicated ATE is being replaced by more generic (flexible and cheaper) ATE interfacing the BST infrastructure, and BST seems to be the best solution to test defects arising from the high pin counting of today devices. Any solution must then be 1149.1 compliant.

Following the above considerations, the objectives for a XMR module (or IC) were defined as:

1. *concurrent* monitoring,
2. *duplication* based IC with no coding circuitry,
3. *reuse of BST* to *compare* outputs and *vote*.
4. *error confinement*: provides a known output,
5. *fault location*: allows to resume operation,
6. *partial DD* to detect *cmf*.

The XMR architecture relies on POST [15], a self-synchronized BST infrastructure with some features to reduce scan requirements, allowing to extend BST interest to concurrent (on-line) monitoring.

In a CUT supervised by POST the BS cells are organized as IN and OUT groups. This is not really mandatory but simplifies the general idea, and pins with a similar I/O function are grouped usually.

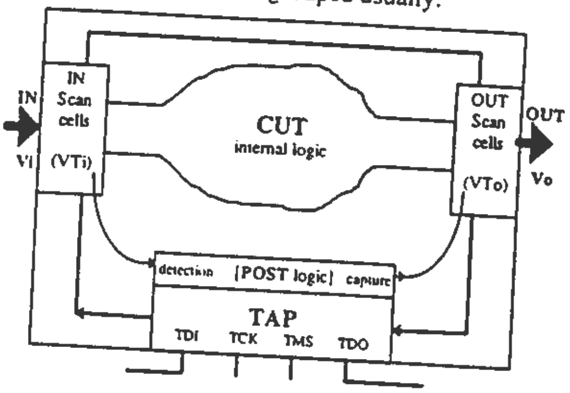


Fig.01 - POST architecture

A BST-controller (BSμC), required in many designs for system-level BST interface [16, 17], scans on-line the input and output *test patterns* set (VTi+VTo) stored in its internal ROM. Generated for functional verification, these vectors are the only thing to be reprogrammed. When an input match occurs (Vi=VTi) the input group generates a signal allowing the output group to perform one of several functions: output capture, checking, or output *injection* bypassing the internal CUT logic. A single BSμC may scan patterns for several XMR modules, each one accessed independently, one at a time.

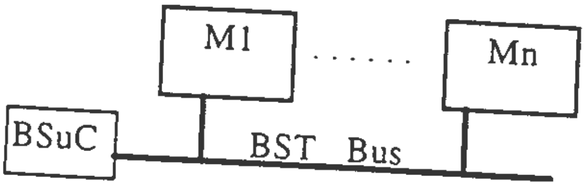


Fig.02 - The BSμC accessing multiple Modules

The input pattern search is done inside each module by POST logic, while the BSμC supervises the other modules [15]. Now POST was optimized to verify *concurrently* a 2-CUT architecture and enhanced to provide decisions.

III - The XMR Architecture

According to the theory of information applied to digital testing [18], the *probability of detecting* faults is directly related to the *quantity of information*; so the *fault location* interval is inversely dependent on this quantity. Since fault location allows a 2-CUT module to *recover*, and the information depends on the redundancy, the XMR base idea is to accept a delayed fault location in change with hardware overhead.

A XMR module, with 2 CUT, has information enough to confine single-CUT errors through comparison. The additional information required to disable the faulty *CUT* is dealt ***through*** and ***into*** the scan cells. The XMR-BST infrastructure, with a *standard* TAP and 2 common sets of cells, may supply discrete data: the set of input cells (observation-only or standard) generates the input detection signal D, and the set of *enhanced output* BS cells provides decisions.

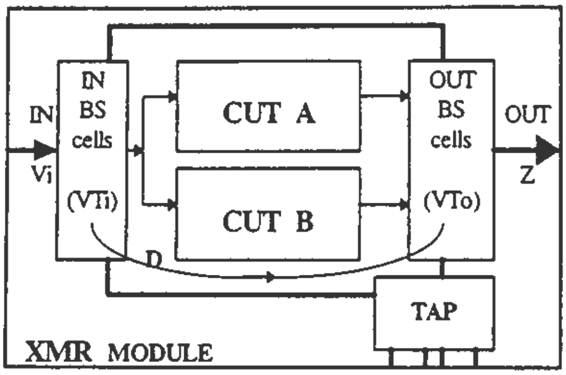


Fig.03 - The XMR architecture

Output comparison of the 2 CUT always confines single-CUT errors, driving the IC outputs (Z) into a known state. Independently, the BSµC shifts the input and output test vectors (VTi+VTo) continuously; upon occurrence of an input match (Vi=VTi), the output BS cells have information enough to vote and disable the faulty *CUT*, if any.

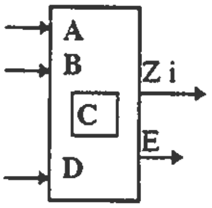
Let Q and Q_M be the *quantity of information* in each *CUT* and in the module respectively, and $\rho \in [0,1[$ the fraction of information accessible through BST (the patterns generated for *functional verification*, stored in the BSµC ROM); a XMR module deals with $Q_M = Q*(1+1+\rho)$, allowing a *TMR-like* behavior with discrete voting and $X \in [2,3[$.

Since ρ is defined by the set of patterns generated for functional verification, the best set will decrease the average time for a match, allowing to resume operation faster. Being known that 90% or more of the faults are temporary, combinatory modules may recover easily. This is one more reason to justify the cost of a complete debug and simulation.

III.1- The Voting cells and the BSµC

In our approach we aim not changing the BS cells, and POST was developed to work with standard cells or even *Observability-only* cells, which do not add delays to the signal path. But, to speed-up error treatment, decisions must occur at output cell level avoiding to scan data back. This allows to simplify the BSµC, and requires to redesign the multiplexers of the output cells, for a double function:

1. *comparison* of replicas (A, B).
2. *voting* among replicas (A, B) and the content of a cell latch (C), when an input match occurs (D).



Comparison is always working, but voting needs the additional information provided by the BSµC. Zi is the output bit and E the error signal, to force the output BS cells into one of 4 states:

- 0, 1, high-impedance
- to freeze the output.

The 3 first values are design or technology dependent, and may be combined to provide any *known-safe* output pattern (as usual during power-up), each cell being pre-defined separately. To hold the current (good) output, the BS cell needs an extra latch.

The truth table to redesign the MUXs is coherent with a TMR, and the result increases the BS cell hardware near 50%, without changing the TAP. Assuming that the BST infrastructure overheads 5-7% to a circuit [19,14], a XMR IC may be expected to have **X<2.1**. The signal path in the voting scan cell has 3 mandatory gates, against 2 in the standard 1149.1 cell, but gate count optimization adds one more gate to the path (4).

The BSµC may verify the integrity of the BST infrastructure [20]. The voting cells are designed to involve the cell decision logic in the capture of the *true* output (Zi), when they are scanned out to read Vo (Z) in POST mode, and the BSµC compares the capture (Vo) with the expected pattern (VTo). This transparent and near complete self-test to the cells may avoid the need to design *self-testing* cells. When performed on-line, voting has to be disabled but comparison is active.

The BSµC required to support XMR operation is a Finite State Machine with no software, 8 mandatory pins only, and 1-4KB of ROM to store the test vectors. It may be built inside the XMR IC, but from a dependability point of view an external BSµC is preferable. The 1149.1 BST standard is *not* defined at system level [21], and systems compliant with IEEE 1149.5 std require a Module Test and Maintenance (MTM) interface in each PCB [16]; the BSµC may join both functions, and support several XMR modules.

III.2 - Operating modes and Latency interval

A XMR module, dealing with $Q_M=Q^*(1+1+\rho)$, allows 4 modes of operation:

1. *stand-alone* $\{\rho=0\}$: without the BSµC, or if disabled, the output cells may only compare the CUT outputs; a mismatch confines the error, and sets the output to a known-safe word (assuming non-degraded cells).
2. *normal* $\{Q_M=Q^*(1+1+\rho)\}$: both CUT are good and the module is supported by a BSµC. Error confinement is immediate, and a known output provided until operation is resumed.
3. *single-CUT* $\{Q_M=Q^*(1+\rho)\}$: a XMR IC may resume operation with the right CUT supervised in POST mode to provide a timely error detection. This mode may be maintained indefinitely or until a reset operation is allowed (for sequential CUTs).
4. *survival* $\{Q_M= Q^*\rho\}$: a last resource if both CUT fail permanently; the test patterns in ROM may still be applied at the outputs (VTo), one at a time and when a VTi matches Vi. Obviously restricted, this mode goes a step further fail-stop designs. Typical applications may allow an elevator to continue a blind shuttle service, an automotive ABS to provide a limited service, or avoid to shutdown a fuel-injection engine, even if fuel economy is not guaranteed.

When the CUT disagree, the module outputs the safe pattern until the next input match disables the faulty *CUT*. This latency interval can be estimated as a function of the *CUT* number of inputs (*n*). Assuming an *equi-probable* distribution and a new input every interval T, the average delay is given by $t=(2^{n-1})*T$, shown below for 3 frequencies. Low input rates can also be verified as shown in [15].

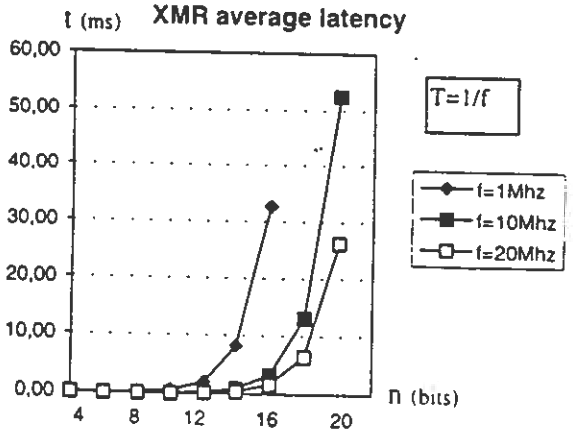


Fig.03- Average latency for recovery

A *normal* input distribution allows to select a set of patterns with higher probability of existence, decreasing the average delay.

III.3 -Common Mode and Intermittent Faults

The VT set, needed in XMR designs for the recovery process, may be derived directly from the specification enhancing the *design diversity* provided through BST. Its fault coverage is relevant for *cmf* detection only.
Permanent *cmf* must really be considered as design and production defects. After debug and burn-in, and in normal conditions, permanent faults are mostly a long time process, and simultaneous occurrence in both CUT may be assumed to have a low probability. On the other hand, as voting occurs with every input match and is always working, the probability of detection of permanent *cmf* is related to the *fault coverage* of the test patterns [15]. Temporary *cmf* may also be detected, but this process needs a further analysis.

Intermittent faults are an increasingly important question as integration density grows: pin number, power dissipation and thermal problems may originate *intermittent* bond wire defects [10]. Large ICs, approved in production tests, may show opens in one or more pins as soon the temperature rises, and return to a perfect operation later, unhappily when being tested. The soldering process may also degrade bond wires, and replication of pins has been suggested [4].
Off-line test methods may hardly detect these defects, even with BST and performed at nominal temperature, and concurrent verification becomes highly desirable.

The XMR architecture is able to detect many input pin and external defects affecting input patterns: below a given rate of input matches the BSµC may consider the system working *not properly* and signal a *non-fatal* error; the module may be allowed to continue or, if required, tested off-line scanning all the test patterns.

IV- Extension to sequential design

A XMR module with sequential CUTs always confines the errors. Fault location requires to add *observation* scan cells mirroring the internal memory elements, but the higher number of states requires more VT's and enlarges the detection intervals. We have a solution to search *multiple pattern combinations* and speed up input matches as far as required, without changing the IN BS cells, but not yet to deal with all the corresponding responses.
One solution is to scan data back and let the BSµC read the outputs and decide, but slows down the voting mechanism. Partition is more effective, and a sequential XMR sub-*CUT* seems acceptable (concerning latency delays) if the number of inputs *plus* memory elements is no greater than 18-20.
A XMR module resumes operation without resetting the right *CUT*; however, wrong states must not return to valid states, or a faulty *CUT* with a valid combination may be selected first, disabling the right *CUT*. We do not have a better answer for this problem yet.

Conclusions

XMR is a new subgroup of NMR architectures: a 2-CUT design helped by an enhanced BST infrastructure to provide a TMR-like behavior. A XMR IC may work as stand-alone or supported by a BSμC. Error confinement is always immediate for single CUT faults. Sequential XMR designs are possible through partition but take a bit more to resume.

The XMR-BST infrastructure is fully IEEE 1149.1 compliant, and may work in association with other F-T solutions. The main advantages are:

- easy of design
- transparent monitoring
- hardware overhead acceptable in VLSI ICs
- very low impact on performance
- reuse of simulation patterns
- ability to detect *cmf*
- low cost

Currently the following points are being addressed:

- voting scan cells with robust design and reduced signal path delay,
- sequential circuits, concerning recovery from temporary faults mainly.

Replication based designs are many times built of independent parts, because of *cmf* and repair, but are more expensive also. Since repair implies usually to switch the system off, and most permanent *cmf* are detectable here, a XMR IC is useful if temporary *cmf* are not relevant, which may be acceptable after debug and burn-in. It must be noticed that a TMR and a duplex self-checking can't deal with *cmf* too. High-safety systems may require physical independence of the replicas, but, even here, replicas built of XMR ICs can be an advantage over the single CUT counterpart.

REFERENCES

1- P.K. Lala, *Fault Tolerant, Fault Testable HW Design*, Prentice/Hall International, 1985.

2- M. Nicolaidis, S. Noraz, B. Courtois, "A Generalized Theory of Fail-Safe Systems", *FTCS-19 Digest of Papers*, IEEE Comp. Society Press, 1989, pp398-406.

3- I. Koren, A.D. Singh, "Fault Tolerance in VLSI Circuits", *IEEE Computer*, pp73-82, July 1990.

4- E. Stroud, A.E. Barbour, "Testability and Test Generation for Major Voting Fault-Tolerant Circuits ", IEEE *JETTA*, 4, pp. 201-14, 1993.

5 - *IEEE Standard 1149.1 Test Access Port and Boundary-Scan Architecture*, IEEE Inc, NY, 1990.

6 - J.H. Lala, R.E. Harper, "Architectural Principles for Safety-Critical Real-Time Applications", *Proc. IEEE*, V82 n1, Jan 1994, pp25-40.

7- E. Bohl, R. Stephan, W. Glauert, "The Architecture of the Fail-Stop Controller AE11", *3rd IEEE IOLTW'97*, Crete, Greece, 1997, pp. 47-52.

8- M. Lubaszewski, B. Courtois, "On the design of Self-Checking Boundary Scannable Boards", *Proc. of ITC*, 1992, IEEE, pp.372-81.

9 - M. Nicolaidis, "Shorts in Self-Checking Circuits", *IEEE Journal of Electronic Test: Theory and Applications* 1, pp.257-73, 1991.

10 - M. D'Abreu, A. Stokes, L. Sassoon, "Defect Analysis-Impact on Yield Improvement and to the Design of Manufacturable ICs", *IEEE ETW'97, Compendium of Papers*, Session 4, Italy, 1997.

11 - R.O. Duarte, I.A. Noufal, M. Nicolaidis, "A CAD Framework for Efficient Self-Checking Data Path Design", *3rd IEEE IOLTW*, Greece, 1997, pp28-35.

12 - M.d'Abreu, P. Isakanian, S. Hunjan, "Understanding of the Fabrication Process- Key to Design and Test of Mixed Signal Integrated Circuits", *IEEE ETW'98, Compendium of Papers*, pp.156-9, Barcelona, 1998.

13- C. Kuntzsch, F. Mayer, K. Ronge, "A Novel Approach for an On-Line Selftest Architecture using ASIC Circuits in a Multi-Channel System", 3rd IEEE IOLTW, Crete, Greece, 1997, pp165-8.

14 - *IEEE ETW'98, Industrial Presentations, Compendium of Papers*, Sitges, Barcelona, Spain, 1998, May 27-9.

15- J.V. Santos, J.M. Ferreira, "Failure Detection and Boundary Scan: a Pseudo On-Line approach (POST)", *3rd IEEE IOLTW*, Crete, Greece, July 1997, pp160-4.

16 - *IEEE Standard 1149.5 , Module Test and Maintenance Bus Protocol*, IEEE Inc., NY, 1994.

17 - *Texas Inst. IEEE 1149.1 Testability Primer*, 1994, SSYA002B, http://www.ti.com/sc/docs/jtag/jtag2.htm.

18- V.D. Agrawal, "An Information Theoretic Approach to Digital Testing", *IEEE Trans. on Computers*, Vol. C-30, pp. 582-587, 1981.

19- A.L. Crouch, C. Pyron, "Impact of JTAG/1149.1 Testability on Reliability", *GOMAC* 1989, pp83-90.

20 - F. Jong, F. Heyden, "Testing the Integrity of the Boundary Scan Test Infrastructure", IEEE, *Proc. of ITC*, 1991, pp106-12.

21- T.J. Chakraborty, "On-line Test Method Using BS", *3rd IEEE IOLTW'97*, Greece, 1997, pp156-9.