Digest of FastAbstracts: FTCS-28

The Twenty-Eighth Annual International Symposium on Fault-Tolerant Computing

June 23-25, 1998 Munich, Germany

Sponsored by IEEE Technical Committee on Fault-Tolerant Computing

Dependable Circuits Based on the XMR Architecture

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

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

Abstract

The new XMR architecture is a TMR-like design aiming single IC integration and easy of design. An enhanced BST infrastructure replaces one replica and the Voter, allowing error confinement and a timely recovery.

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

Introduction

Many Fault-Tolerant (**F-T**) systems in automotive, railway, industrial control and even aerospace, are *real-time* designs requiring *error confinement* but, usually, may allow a small *latency* interval to *resume operation*, if the outputs are set to a *known-safe* state meanwhile [1]. This opens a wide field of application to dependable Integrated Circuits (**IC**) providing these features.

Most safety-critical hardware designs are *self-checking* (***S-C***) or *replication* based architectures [2, 3], but the integration of these architectures in VLSI ICs (ASICs or PLDs) has advantages and drawbacks: the many gates available are not accompanied by pin number, impacting accessibility, and redundancy is desirable to a certain level only [4]. Replication, being interesting in a VLSI (a ***TMR*** for example), is however hampered by *common mode faults (cmf)*; S-C designs facing permanent faults must be *duplex* [5]. *Duplication* of the *mission circuit* is a good compromise in VLSI ICs but has not enough information to resume a correct operation and may only provide a *fail-stop* solution [6]. When a *timely recovery* is acceptable, the *latency* delay, of many milliseconds or even seconds [7], allows another solution to be envisaged, with the help of the Boundary Scan (***BS***) Test (**BST**) 1149.1 infrastructure [8].

The new architecture presented, *XMR*, may be seen as a sub-group of *N-Modular Redundancy* (**NMR**) designs: *a duplication design in which an* 1149.1 compliant *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, a system-level BST interface required in many designs [9].

XMR is presented for combinatory designs first; the extension to sequential logic is considered in the end.

Background and Objectives

To start with we must clarify some terms here:
- *CUT*: the *circuit under test*, or the *mission circuit.*
- *Module*: a 2-CUT *IC* with BST, expected to behave as a Fault Containment Region (**FCR**) [1].
- *Fault model*: single-faults only considered at first. Replication based designs are vulnerable to *cmf*, but a XMR IC is able to detect most permanent *cmf.*

Error confinement means an *immediate* detection avoiding the error to spread out the module. To *resume a correct operation* means to feed the outputs with right CUT (always running) as soon as the fault is located. A latency interval shorter than the system's *time granularity* allows to *correct the error*, otherwise we talk about *correct recovery* only.

Our objective is to design dependable ICs easily, with low overhead and improved reliability. According to the theory of information applied to digital testing [10], the *probability of detecting* faults is directly related to the *quantity of information*; then, the *fault location* interval is inversely dependent on this quantity, related here to the redundancy mainly. Since a 2-CUT module may recover when the fault is located, the XMR base idea is to accept a delayed fault location in change with hardware overhead. Considering features as VLSI IC limitations, fault model and latency, concurrent test and design diversity, the objectives for a XMR IC where defined:

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

The XMR architecture relies on POST [11], a *self-synchronized* BST infrastructure enhanced to reduce on-line scan requirements. The BST-controller (**BSμC**, 1-4KB ROM, 8 mandatory pins) provides additional information on-line, scanning the *input* and *output test patterns* (VTi+VTo) previously defined for *functional verification* and stored in the internal ROM; when an input match occurs the output may perform several functions. POST was now optimized to verify a 2-CUT architecture *concurrently*, enhanced to provide decisions, and adds a *partial design diversity* to face *cmf* arising from replication inside a single IC.

The XMR Architecture

A XMR module has 2 CUTs, a set of input BS cells to detect input matches (**D**), the set of *enhanced output* BS cells to provide decisions, and the 1149.1 *std* TAP controller. CUT comparison confines the error in the out cells, driving the output (**Z**) to a known-safe state.

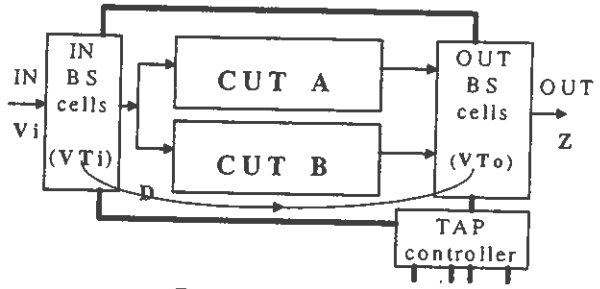


Fig.01- The XMR architecture

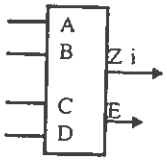
Independently, the BSμC scans the input and output test vectors (VTi+VTo) continuously [11]; upon occurrence of an input match, the output cells have information to vote and disable the faulty *CUT*, if any.

Let **Q** and $\mathbf{Q_M}$ 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 stored in the BSμC); a XMR module deals with $Q_M=Q*(1+1+\rho)$, allowing a *TMR-like* behavior with *discrete* voting and 4 modes of operation:

1. *stand-alone* ($\rho=0$): error confinement, known output.
2. *normal* ($Q_M=Q*(1+1+\rho)$): error confinement, known output, recovery .
3. *single-CUT* ($Q_M=Q*(1+\rho)$): supervised in POST mode to provide a timely error detection.
4. *survival* ($Q_M=Q*\rho$): VTo is injected if Vi=VTi [11].

To speed-up error treatment decisions must occur at BS cells level, requiring to redesign one multiplexer in 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), upon occurrence of an input match (D).



A disagreement activates the error signal (**E**) to force the cell outputs (Zi) into one of 4 states:

- *0, 1, high-impedance*: are design dependent to provide a *known output*, and allow any output combination.
- *to freeze the current output*: cells need an extra latch.

The following input match disables the faulty *CUT*, and this interval may be estimated as function of the inputs (*n*). Assuming a new input every T interval and equi-probable distribution, the average delay is $t=(2^{n-1})*T$, shown in the table for a 10Mhz operation frequency. Partition may allow any number of inputs.

n	t (ms)
8	0,01
12	0,20
16	3,28
20	52,43

Extension to sequential designs

A XMR module with 2 sequential CUT always confines the error. Fault location requires cells mirroring the memory elements, but the higher number of states enlarges the detection intervals. Partition is a possible solution to get sub-CUTs with no more than 18-20 *inputs plus memory elements*. Wrong states must not return to valid states or a faulty *CUT* may disable the right *CUT*.

Conclusions

The truth table to redesign the MUX, coherent with a TMR, increases the cell hardware near 50%, with a signal path delay of 4 mandatory gates (2 in the std cell). The cell is tested in POST mode by reading the *true* output on-line. Assuming that the BST infrastructure overheads near 5% [12], a XMR module may have **X<2.1** (IC hardware only). Functional simulation test patterns (possibly derived from the specification) are reused, enhancing the *design diversity* provided through BST, and allow to detect permanent *cmf* according to their (single-) *fault coverage* [11]. The XMR-BST satisfies all the objectives defined, is 1149.1 compliant and may run in association with other F-T solutions.

Sequential circuits, reliable BS cell design and BST chain reliability features need further attention.

REFERENCES

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

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

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

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

6- E Bohl, R Stephan, W Glauert,"The Architecture of the Fail-Stop Controller AE11", IEEE IOLTW, Greece, 1997, pp.47-52.

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

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

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

10- V.D. Agrawal, "An Information Theoretic Approach to Digital Testing", *IEEE T. Computers*, V.C-30,pp.582-7, 1981.

11- 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.

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