# Are Our Design For Testability Features Fault Secure ?

C. Metra
*DEIS – Univ. of Bologna (Italy)*
*cmetra@deis.unibo.it*

TM Mak
*Intel Corporation, Santa Clara (CA)*
*t.m.mak@intel.com*

M. Omaña
*DEIS – Univ. of Bologna (Italy)*
*momana@deis.unibo.it*

## Abstract

*We analyze the risks associated with faults affecting some common Design For Testability (DFT) features employed within digital products. We will show that some DFT structures may become useless, with consequent dramatic impact on test effectiveness and product quality. We borrow the Fault Secure property and we will show that it guarantees that no escapes or false acceptance of faulty products may occur because of faults within the DFT structures.*

## 1. Introduction

The continuous scaling of microelectronic technology will soon lead to billions of transistors on a single piece of silicon. Higher level of integration allows more functionality and performance improvement, but it also makes testing increasingly difficult. Meanwhile, scaling to feature sizes that are less than the wavelength of light brings increasing variations of electrical parameters, thus making it more difficult to guarantee the proper operation speed of the fabricated chips. Consequently, we are also experiencing an increasing adoption of Design For Testability (DFT), Design For Debug (DFD) and Clock Calibration (CC) schemes on the chip [2, 3].

Very often, DFT, as well as DFD and CC structures, are designed as if they are fault free by default. The fault probability actually increases with the area of the circuit elements, so increasing reliance on DFT will increase the amount of DFT circuit elements and therefore, DFT circuit are also susceptible to faults. Based on these considerations, we discuss the risks associated with faults possibly affecting some of the most commonly adopted DFT schemes of digital products.

## 2. Risks associated with DFT structure faults

Scan and BIST are probably the most widely adopted DFT strategies. So, we will begin with these.

As for Scan structures, faults may affect: i) the Test Access Port signals; ii) the Scan flip-flops.

For faults of kind i), we can reasonably expect that they are detected during the Scan chain testing phase [15]. Similar considerations hold true also for faults affecting the Scan flip-flops.

For general BIST structures, faults may on principle affect: i) the Hardware Pattern Generator; ii) the input MUX; iii) the Output Response Compactor; iv) the Comparator. Faults of kind i) might make the Circuit Under Test (CUT) receive, during the test phase, test patterns different from those expected for the case of fault free test pattern generator. We can expect that an incorrect signature is eventually produced by the Output Response Compactor, so that a *Fail* is given to the output and the faults are detected by the BIST structure, with no impact on the CUT test quality and effectiveness. Similar conditions hold true for faults of kind ii). As for faults of kind iii), we can expect that, either they make the compactor provide the signature that would have been produced in the case of no internal fault, or vice versa. In the former case, of course, the compactor faults can be tolerated with no problems. In the latter case, instead, we should distinguish two possible conditions: 1) the faulty compactor gives a signature equal to the memorized one, rather than a different one (*i.e.,* the CUT is faulty); 2) the faulty compactor gives a signature different from the memorized one, rather than an equal one (*i.e.,* the CUT is fault-free). Case 1) is of course very dangerous, but we can reasonably consider it rather unlikely. Case 2) does not create any problem, as the output comparator produces an output *Fail*, and the fault can be detected. Therefore, in both cases, no risk for the CUT test quality and effectiveness should take place. The situation is different for faults of kind iv). In fact, should even a simple stuck-at fault (SA) at the comparator output indicate a *Pass*, a *Pass* will always be recognized at the BIST output, despite the possible generation of an incorrect signature by the BIST structure itself, with consequent dramatic impact on the CUT test quality and effectiveness. Similarly, it is possible that, because of the presence of comparator internal faults (different from the output SA), a *Pass* is produced, rather than a *Fail*.

1530-1591/04 $20.00 (c) 2004 IEEE

Problems of this kind might on principle arise for any comparator generally employed within BIST structures, as well as for those within the BIST controller in [6, 7] and the sticky bit comparator in [2].

Out of these referenced comparators, that in [7] (Fig. 1) is immune from problems due to SAs affecting its output, because of its additional "testout" output [7]. Alternate solutions to make a comparator robust with respect to output SAs can be found in [8, 1]. However, these solutions may not eliminate the problem of the possible generation of a *Pass* rather than a *Fail* because of comparator internal faults.

We analyzed these problems considering the comparator most recently proposed in [7] (Fig. 1).

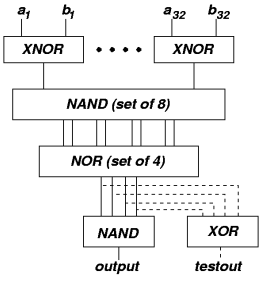


**Fig. 1 32-bit comparator presented in [7].**

We performed electrical level simulations of this comparator, considering a realistic set of possible internal faults, composed by all possible internal SAs, transistor stuck-opens (SOPs) and resistive bridgings (BFs), with values of connecting resistance (Rbrid) in the [0…6kΩ] range. We also made the simplifying assumption of having a single fault within the comparator, in order to simplify fault accounting.

We verified that the generation of a false *Pass* could occur at the BIST output, with a detrimental effect on the CUT test quality and effectiveness, for the 48% of all possible SAs, the 46% of all possible SONs and for the 10% of all possible BFs.

## 3. Fault secure DFT structures

*Fault Secureness* [4] is a well-known property in the field of self-checking circuits (SCC) [5]. We re-define it for DFT structures as follows: *a DFT structure is Fault Secure (FS) with respect to a set of internal faults F if, for every fault in F, it can never occur that the performed testing procedure gives a "Pass", rather than a "Fail".*

*Fault Secureness* guarantees that we cannot obtain a false "*Pass*", because of faults affecting the DFT structures, thus avoiding their detrimental impact on test quality and effectiveness.

We can easily verify that general Scan schemes can be considered *FS* with respect to internal faults.

As for BIST schemes, included the BIST controllers in [6, 7], and the sticky bit comparator in [2], none of them is generally *FS* with respect to its possible internal faults. In particular, for all these schemes, the critical block is the output comparator. Considering that recently proposed in [7], we have shown that problems may arise because of its internal SAs , SOPs and BFs.

As for output SAs, a possible solution could be to adopt a technique of the kind in [8], or to induce a controlled "*Fail*", as described in [1].

In order to guarantee *Fault Secureness* with respect to output SAs, as well as internal SAs, SOPs, BFs, the comparator internal structure could be changed in order to make it able to provide an output *Fail* as soon as an internal or output fault occurs. As an example, the internal structure of such comparators can be derived from that of embedded Totally Self-Checking (TSC) [5] two-rail code checkers, frequently used within SCCs, *e.g.,* that recently proposed in [9].

As for a *FS* alternate to the sticky bit comparator in [2], a two output comparator of the kind mentioned above, followed by an error indicator of the kind used within SCCs (*e.g.,* that in [10]) could be employed.

## References


[1] M. Tripp, T.M. Mak, A. Meixner, "Elimination of Traditional Functional Testing of Interface Timings at Intel", *to appear in Proc. of Int. Test Conf.,* 2003.

[2] D. D. Josephson, S. Poehlman, and V. Govan, "Debug Methodology for the McKinley Processor", in *Proc. of Int. Test Conf.,* 2001, pp. 451 – 460.

[3] T. Litt, "Support for Debugging in the Alpha 21364 Microprocessor", in *Proc. of IEEE Int. Test Conf.,* 2002, pp. 584 – 589.

[4] J. E. Smith and G. Metze, "Strongly fault-secure logic networks," *IEEE Trans. Comput*., vol. C-27, pp. 491 – 499, June 1978.

[5] W. C. Carter and P. R. Schneider, "Design of dynamically checked computers," in *Proc. IFIP '68*, Edinburgh, Scotland, pp. 878 – 883, 1968

[6] F. Karimi, F. Lombardi, "Parallel Testing of Multi-Port Static Random Access Memories for BIST", in *Proc. of The Int. Symp. on Defect and Fault Tolerance in VLSI Systems,* 2001.

[7] B. Bailey, A. Metayer, B. Svrcek, N. Tendolkar, E: Wolf, E. Fiene, M. Alexander, R. Woltenberg, R. Raina, "Test Methodology for 15's High Performance e500 Core Based on PowerPC Instruction Set Architecture", in *Proc. of IEEE Int. Test Conf.,* 2002, pp. 574—583.

[8] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices", pp. 4 – 9, 1993.

[9] M. Omaña, D. Rossi, . Metra, "High Speed and Highly Testable Parallel Two-Rail Code Checker", in *Proc. of Design, Aut. and Test in Europe Conf. ,* 2003.

[10] C. Metra, M. Favalli, B. Riccò, "On-Line Testing Scheme for Clocks' Faults", in *Proc. of Int. Test Conf.,* 1997, pp. 587—596.