On-line Parity Control Enhances 1149.1 BST Reliability

José Miguel Vieira dos Santos
jmvs@dee.isep.ipp.pt
ISEP- I. S. Engenharia do Porto
R. S. Tomé - 4200 Porto - Portugal

José Manuel Martins Ferreira
jmf@fe.up.pt
FEUP-Universidade do Porto
Rua Bragas 4050 Porto - Portugal

Abstract

Nowadays, the 1149.1 BST std is being proposed in several applications for on-line operation, but it provides no way to verify the integrity and correctness of scanned data, because it was not designed to do so, off-line. However when the scan data is used for immediate decisions, we need to have confidence in the string entering or leaving the target IC.

In this paper we show how the BST-controller may read the parity of this string immediately after the scan operation, with almost no changes to the TAP controller.

Keywords: On-line test, BST, DFT, Dependable VLSI.

Introduction

The 1149.1 BST infrastructure [1] was designed for off-line testes mainly. In this mode, the **CUT** (Circuit Under Test) has usually no, or little, activity during the scan operation, which will run smoothly: *noise* is a minor concern. Besides, in case of error, there will be enough time to repeat the test, providing the necessary debug to know if the fault is in the CUT or in the BST infrastructure operation.

However, the 1149.1 std is now being suggested for a lot of operations requiring reliable scan data. Concurrent sampling [2], self-checking systems [3], thermal monitoring [4], Fail-safe processors and systems [5], and scan-bist [6], are, among several other applications, cases in which the CUT may induce noise and degrade scan operations. TCK frequencies are also increasing and another source of errors.

Traditionally, BST data is not targeted for an immediate action into the CUT operation: it is read back to the BST controller and scan errors are not really critical, mainly if the BS-cells are in observability mode only. In case of error, the scan operation may be repeated, reading the data again and check results. Even in a std off-line scan operation, errors may result from faults in the CUT, or from scan errors.

The worst case, however, are 2 situations in which the error is critical, and we have no control on the consequences:

1. When scanning instructions to the **TAP** (*Test Access Port*), data misinterpretation may lead to the storage of an erroneous IR (Instruction Register) instruction. In consequence, the BST cells may be set to controllability modes, and impact CUT operation.
2. In recent proposals for ICs tolerating faults, the BST infrastructure provides additional information to the IC, under the control of the on-board BST-controller (**BSμC**). The BS infrastructure is reused to monitor the CUT [7] and also to provide decisions, disabling the faulty CUT in a 2-CUT design inside ICs tolerating permanent faults [8]. This architecture is named **XMR,** which stands for an *incomplete-TMR* architecture.

Since the BST data leads to immediate action here, we need feed-back on the integrity of the bits entering or leaving the target IC. This kind of problems was already addressed in [9], to increase the reliability of TAP instructions through an Interrupt scheme. However, in this approach the parity bit value must be constant and this imposes a lot of constraints. The size of the IR strings, the need for an Interrupt chain

and an additional line (TINT), also lead this solution not to be fully compliant to the 1149.1 std. Furthermore, the circuitry to handle and control an Interrupt infrastructure may easily become more complex than the BSμC required in our approach.

On-line Parity Check

Our proposal is to calculate on-line the parity of the string being scanned and provide this result through TDO. This way, the BSμC reads the parity immediately after the scan operation, and may repeat the cycle if not correctly received.
In our scheme the CUTs (or ICs) may be accessed in series or in parallel, allowing much faster scan operation, which requires independent TMS lines only [1, 7]. Our main interest are the scan operations *to* the IC, aiming a fast response to avoid store erroneous data, and since usually we have no need to change TAP instructions during on-line operation, the first target are DR (Data) operations. This mechanism will be presented for DR cycles, but is similar for instruction (IR) cycles.
In the figure 1 we can see the two possible parity generation schemes, inside the target IC:

- *Series*: Ps will be generated as soon the bits are received, which allows a very simple hardware.
- *Parallel*: Tp is from the new values stored in the register scan cells, which needs a little more hardware but is more reliable.

Both ways will be ready at the same time and if the probability of errors inside the IC is low, the first solution is preferable.
So, in our case we consider Ps connected to the multiplexer input P and we may see in figure 2 the simulation diagram in which the parity bit is provided in the Exit1 state of the TAP controller cicle.

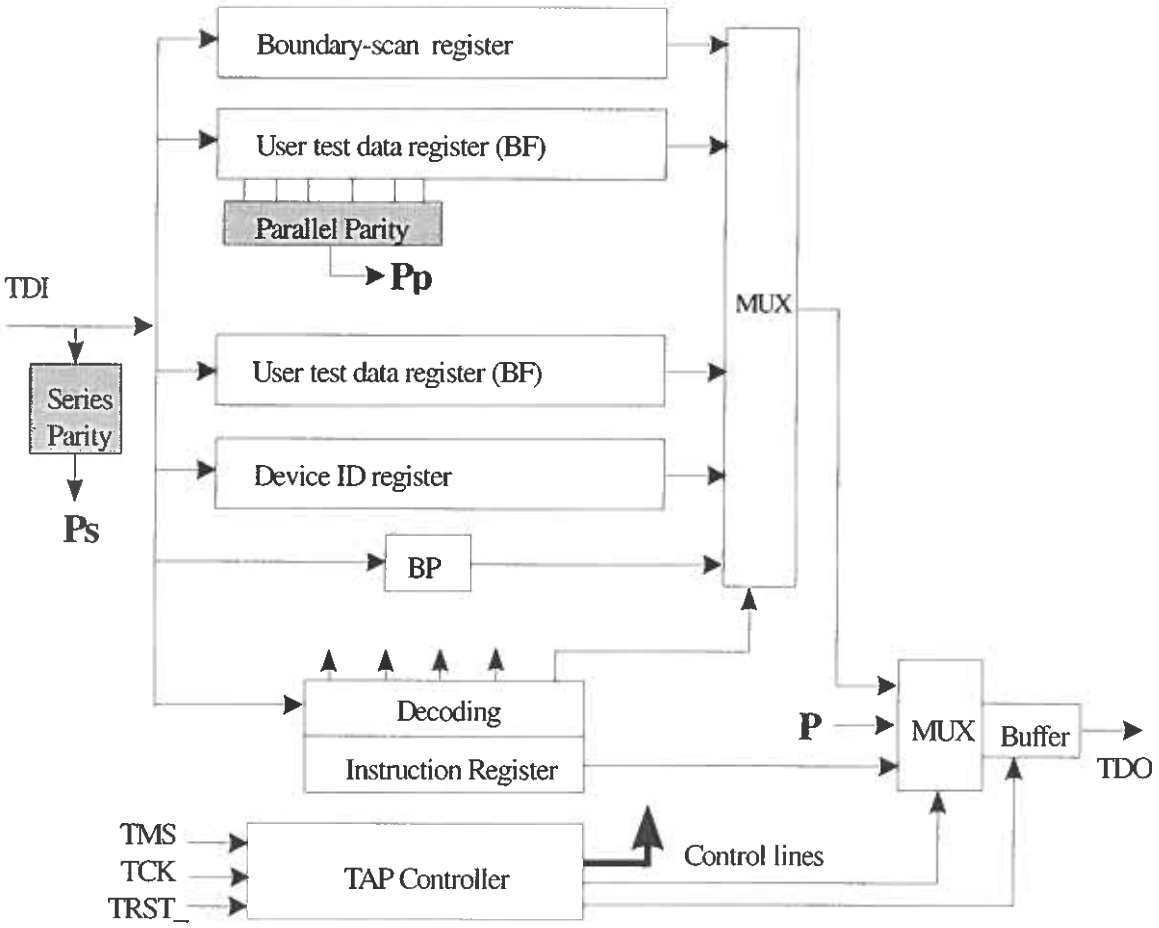

Figure. 1

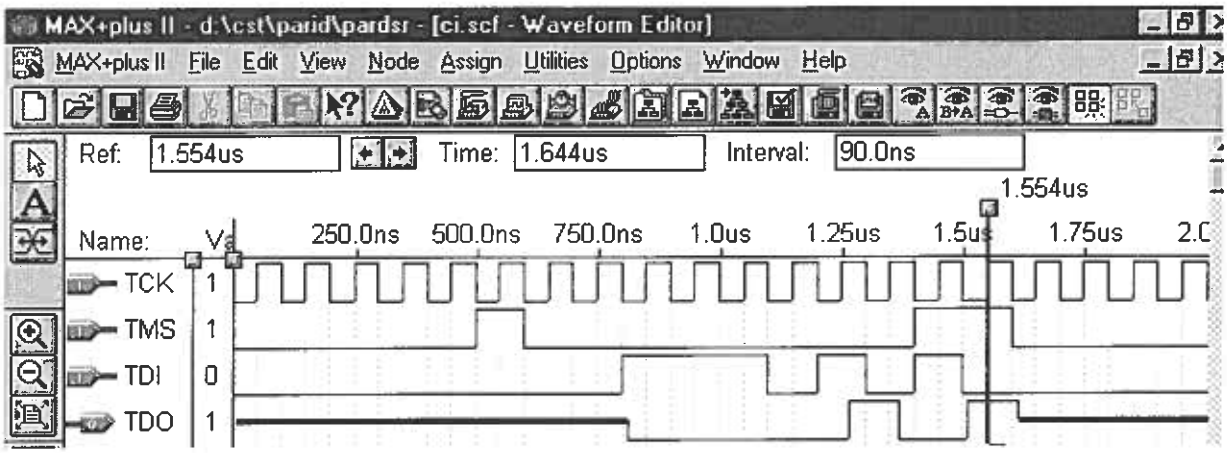

Figure. 2

The TAP control

The above diagram corresponds to the TAP control cycle represented in figure 3. The IEEE 1149.1 BST std, defines that in normal operation the last scan bit is transferred when the TAP goes from Shift-DR to Exit1-DR. After this sate, the TAP goes (normally) through Update-DR, which ends the DR cycle, while TDO remains in High-Impedance, because there is no data to transfer.

Obviously, in this case, the new data will be stored, either good or bad, because the TAP goes to the Update-DR (or IR) state. The solutions to this problem, is present in the figure 4. In this case, if the scanned data is incorrect the BST-controller may return to the Shift state and repeat the scan operation again, without storing bad data into the register cells.

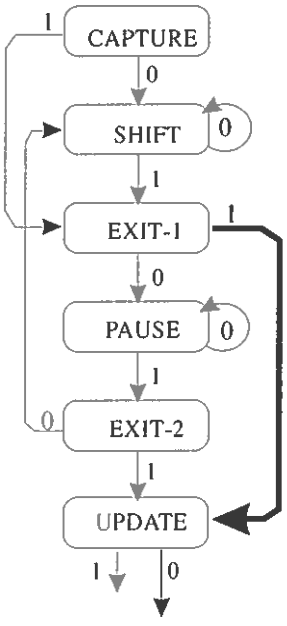

Figure 3- TAP Control: option 1

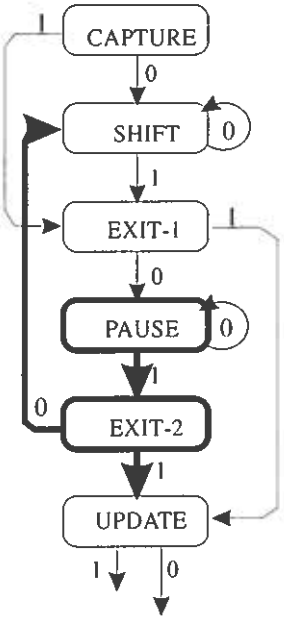

Figure 4- TAP Control: option 2

Final comments

We have proposed a solution which allows the BST-controller to read back the parity of the string received in the target IC, in order to verify scan data integrity. This means almost no hardware overhead but the parity generator, which in the series generation scheme represents less than 1% of the BST infra-structure.

The proposal is not compliant to the 1149.1 std during a single TCK cycle only, but this feature will be not detected by 1149.1 compliant components. Furthermore the proposal has several advantages:

- No impact on the CUT performance,
- No additional data needs to be scanned,
- Standard and simplified BS cells allowed with no changes,
- minimal overhead to the BST: <1% to the std.

The dynamic overhead is zero with option one and two TCK cycles only with option two.

This mode can also be applied to the output string, so that the BSμC may check data leaving the target IC TDO in the same way.

Selected Papers

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

2- D. Wagner and T. W. Williams, "Enhancing Board Functional Self-Test by Concurrent Sampling", IEEE ITC 1991, pp. 633-40.

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

4 - J.M. Karam, M. Lubaszewski, B. Courtois, "Thermal Monitoring of Self-Checking Systems", 2nd *IEEE IOLTW'96 Compendium of Papers*, Biarritz, pp. 6-12, 1996.

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

6 - G. Kiefer, H-J Wunderlich, "Deterministic BIST with Multiple Scan Chains", *IEEE ETW'98 Compendium of Papers*, Sitges, Spain, pp. 39-43, 1998

7- J.M.V Santos, J.M.M. Ferreira, "The POST Approach to On-Line Failure Detection Based on BST", pp.27-32, *Proceedings XII DCIS'97*, Sevilla, Spain, 1997.

8 - J.M.V Santos, J.M.M. Ferreira, "Joining Boundary-Scan to NMR Architectures: The XMR Architecture", *Proceedings XIII DCIS'98*, pp. 630-4, Madrid, Spain, November 1998

9 - C. M. Maunder and R. E. Tulloss, editors. *The Test Access Port and B-S Architecture*, IEEE, 1990, Chpt. 20 by: P.F. McHugh and L. Whetsel, "Adding Parity and Interrupts to IEEE Std. 1149.1,pp 205-213.