A Boundary Scan-based one-channel timing analyser

Gustavo R. Alves and J. M. Martins Ferreira

Abstract

The Boundary Scan Test infrastructure is now widely implemented in the Integrated Circuit market, especially in the microprocessor and Application-Specific Integrated Circuit arena. While the structural test of Printed Circuit Boards has been considered the driving force behind its broad acceptance, the test community has also addressed the issues of prototype debug and validation. However, the more demanding requirements associated with these issues are not sufficiently covered by the mandatory and optional operating modes described in the IEEE 1149.1 Standard, especially for debugging problems associated with real-time operation. Previous work has focused on this problem, having resulted in a new set of user-defined optional instructions addressing the use of the BS register to store in real-time a sequence of contiguous vectors, captured at its parallel inputs without / until / after a certain condition is found. In this paper we describe the trade-off between input channels and storage capacity, by proposing a new operating mode where the BS register is used to capture / store an n-bit sequence captured at one single functional pin, thus acting similarly to a one-channel timing analyser. This non-intrusive operating mode may also be used for field diagnosis and other on-line operations.

1. Introduction

Boundary Scan Test (BST) [1] is now widely accepted and used for the structural test of Printed Circuit Boards (PCB). Testing a board through its BST infrastructure proceeds in three main steps, which consist of testing the BST infrastructure itself, testing the interconnects among the components (including those buried into non-BST clusters), and testing the components (mainly through the activation of component-level BIST functions). Other advantages of using BST include simple test interface, assistance on functional debug and test, and availability during field operation debugging [2, 3, 4, 5]. However, the more demanding requirements of prototype debug and validation are not yet sufficiently covered by the mandatory and optional operating modes described in the IEEE 1149.1 Standard. This is especially true when debugging problems that only occur when the system under test is working in real-time, at its full operating speed [6, 7]. Traditionally, logic analysers are used (both as timing and state analysers) for debugging this sort of problems. In this paper, we propose the use of the BS register for capturing / storing an n-bit sequence, captured at one single functional pin, thus acting similarly to a one-channel timing analyser.

This paper is organised as follows: Section 2 presents and discusses previous work in this area, and the motivations that led to our current proposal. Section 3 describes how the BS register can be used to fulfil this new operating mode, activated by a user-defined optional instruction. The sequence of steps that need to be done for performing this operation, the modifications in the group of test data registers and in the BS cell structure and control signals, are some of the items covered in detail. Section 4 concludes this paper with the final remarks and future work.

2. Background and motivation

Extending the use of BST for prototype debug and validation has already been proposed and analysed in several previous works, either in the form of adding new operating modes to this test technology or by adding new components with enhanced test functions to the system under debugging [8, 9, 10]. The case of embedded core-based systems has also been covered, by using the electronic access provided by this test infrastructure [11]. The concept of Design-for-Debug-and-Test (DfDT), in correspondence to the concept of Design-for-Testability (DfT), has also led to some variations or new optional operating modes for the BST infrastructure [12]. In this previous work, we have proposed, analysed, and implemented a new set of optional instructions that place the BST circuitry in special operating modes addressing the following debugging routines:

- Detect in real-time a breakpoint condition referring to the values present at the BS register PIs.
- Capture / store in real-time, in the BS register, a sequence of two contiguous vectors.

- Capture / store in real-time, in the BS register, a sequence of two contiguous vectors, until a certain condition is found.
- Capture / store in real-time, in the BS register, a sequence of two contiguous vectors, after a certain condition is found.

Without adding any flip-flops, these two contiguous vectors may be stored in the BS register – one in the capture / shift stage, and one in the update stage. By trading off the number of input channels (PI of all BS cells) by storage capacity (total number of flip-flops within the BS register) it is possible to implement a one-channel timing analyser with a n-bit storage capacity, where n denotes the total number of BS cells in the device. With this in mind, we proposed ourselves to develop a new operating mode for the BST infrastructure that could implement this functionality.

3. Capturing / storing n-bit sequences from a single pin

The goal of this new optional operating mode consists of using the BS register to capture and store, in real-time, an n-bit sequence corresponding to the values captured at one functional pin. This mode is selected by a user-defined optional instruction, named Memorise Sequence from Single Pin (*MSEQ1P*, in the abbreviated form). The pin that captures the values is selected by another user-defined optional instruction, named Select Pin (*SELPIN*, in the abbreviated form), which places an additional test data register in the TDI-TDO path, with a number of cells equal to the logarithm, in the basis 2, of the total number of BS register cells. The vector that identifies the position of the BS cell associated with the selected pin is shifted into this additional register, named Selected Pin (SP) register.

3.1 Sequence of steps

To capture / store an n-bit sequence it is necessary to perform the following steps:

- Shift in the *SELPIN* instruction in order to place the SP register in the device TDI-TDO path.
- Shift in the combination that selects the BS cell associated with the pin acting as the input channel. The order of the associated BS cell may be found in the device BSDL file (or data sheet). Usually, the BS cell closest to TDO is identified as BS cell [0].
- Shift in the *MSEQ1P* instruction, that will start the capture / store activity immediately after the TAP controller enters the *Run-Test/Idle* state. The captured values are stored in the BS register, which is currently selected by this optional instruction. The samples are stored in the capture / shift stage, by shifting them in a continuous loop, with the BS register acting as a circular buffer, as illustrated in Figure 1. The last sample (e.g., the most recent one) is always stored in the BS cell associated with the selected pin, while the first sample (e.g., the least recent one) is always stored in the BS cell that stands at the left of the previous one (assuming that BS cell [0] is the rightmost one and that BS cell [n-1] is the leftmost one). Notice that if the selected pin corresponds to BS cell [n-1], then the one that stands closer at the left side will be BS cell [0], as the BS register should be regarded as a circular buffer.
- To read out the stored sequence, place the TAP controller in the *Shift-DR* state, and apply cycles on TCK until the BS register contents are all shifted out. When *MSEQ1P* is active and the TAP controller moves through *Capture-DR* the BS register contents do not change.

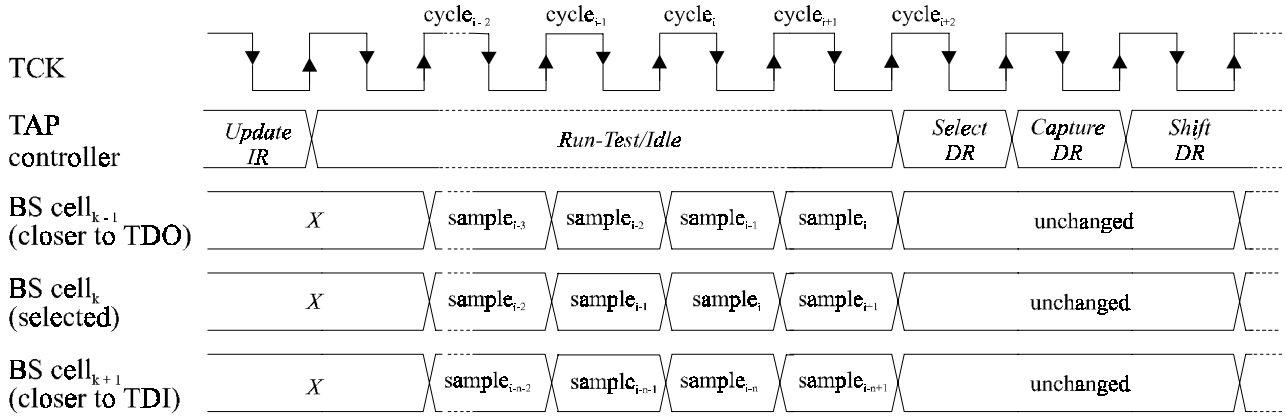

Figure 1: Timing diagram of capturing / storing an n-bit sequence in the BS register.

3.2 Modifications in the group of test data registers and in the BS cell structure

The group of test data registers of the device now includes the SP register, as illustrated in Figure 2. In order to implement the functionality associated with this operating mode, the BS cell structure has to be slightly modified, as illustrated in Figure 3. Notice that this modification is minor, a new two-input AND gate in every BS cell. The output of this gate acts as the control signal of the multiplexer that feeds the capture / shift stage. The input signals are Shift (which acts as the multiplexer control signal in 'normal' BS cells, e.g. as illustrated in [1, page 1-4]), and the line that comes from the m to 2^m decoder illustrated in Figure 4. This decoder decodes the SP register contents in order to allow the BS cell associated with the selected pin to capture the values present at its Parallel Input (PI) while the remaining cells shift the value present at their Scan Input (SI). The decoder output lines are active low so that a logic '0' forces the AND result to be false (or low), causing the multiplexer upper branch (PI) to be selected. A logic '1' in the decoder output line causes the AND result to be controlled by the Shift signal. The decoder truth table presented in Table 1 helps the reader to understand its behaviour. The decoder is enabled when the TAP controller is in *Run-Test/Idle* and *MSEQ1P* is active. When not enabled, all output lines are at logic '1', thus not interfering with the result of the additional AND gate. This way the Shift signal gains full control over the multiplexer, as in a 'normal' BS cell, thus guaranteeing the same behaviour for the three mandatory instructions (BYPASS, SAMPLE / PRELOAD and EXTEST).

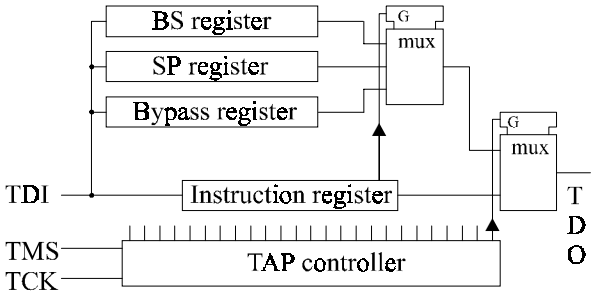

Figure 2: The group of test data registers after implementing the *SELPIN* optional instruction.

While this modification is minor, an additional one is needed in BS cell [n-1] (closest to TDI), to implement the circular buffer. The structure illustrated in Figure 5 refers exclusively to this BS cell, where it can be seen that an additional multiplexer was added in order to allow the selection between TDI and the Scan Output (SO) from BS cell [0] (thus implementing the circular buffer). When *MSEQ1P* is active and the TAP controller is in *Run-Test/Idle* the multiplexer output is connected to the BS cell [0] SO. In all other conditions, TDI will be selected. The multiplexer that feeds the capture / shift stage acts in the previously described way. If BS cell [n-1] is selected (decoder line at logic '0'), the values present at its PI will be captured and shifted along the BS register (notice that all other BS cells are used to store the captured sequence). If BS cell [n-1] is not selected (decoder line at logic '1') the values present at the SI will be captured and shifted along the BS register (until they reach the BS cell that stands immediately at the left side of the currently selected one). If the decoder is not enabled (*MSEQ1P*·*Run-Test/Idle* = false), TDI is selected as the SI of BS cell [n-1], and the Shift signal controls the multiplexer that feeds the capture / shift stage of this and all other BS cells.

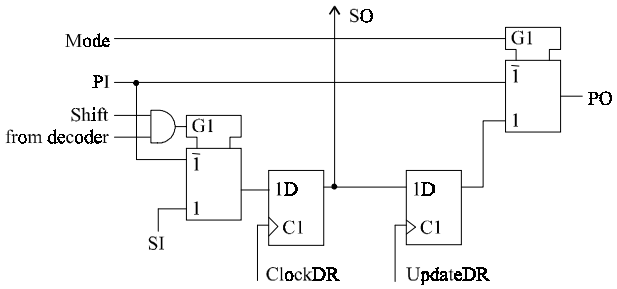

Figure 3: Modified BS cells able to support the *MSEQ1P* optional instruction.

en	SP	cell $_{n-1}$	cell $_{n-2}$	Other cells	cell $_1$	cell $_0$
0	X	1	1	all '1s'	1	1
1	0	1	1	all '1s'	1	0
1	1	1	1	all '1s'	0	1
1	…	1	1	…	1	1
1	n-2	1	0	all '1s'	1	1
1	n-1	0	1	all '1s'	1	1

Table 1: Truth table of the m to 2^m decoder.

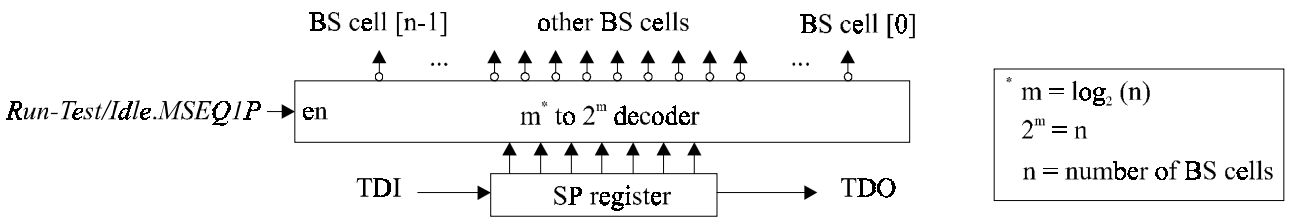

Figure 4: Decoder (m to 2^m) that identifies the BS cell associated with the selected pin, according to the combination present at the SP register.

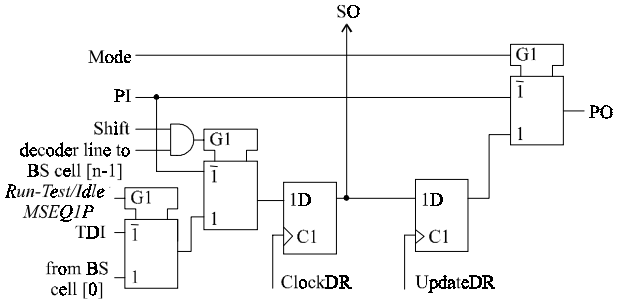

Figure 5: BS cell closest to TDI, modified in order to support the *MSEQ1P* optional instruction.

3.3 Modifications in the BS cell control signals

The modifications in the BS cell control signals are summarised in Table 2. The 1st column contains the control signals, the 2nd column presents the corresponding logic equation, for a BS cell able to implement the functionality associated with the three mandatory instructions, according to what is described in [1, chapter 5], and the 3rd column presents the new logic equations. Notice that UpdateDR and Mode hold the same logic equation, while Shift now includes the *Run-Test/Idle* state (the actions related to *MSEQ1P* occur while the TAP controller is in this state), and ClockDR is expanded in order to accommodate the two actions required by *MSEQ1P* – not capture in the *Capture-DR* state, and capture in the *Run-Test/Idle* state.

3.4 Questions to be addressed

Some questions naturally arise from the previous description. This section tries to answer some of them:

- The SP register does not need an update stage, as the possible intermediate values resulting from the shift process will not cause any misbehaviour. The clock signal of this simple shift register (without parallel load) is active when *SELPIN* is active and the TAP controller is in S*hift-DR*.
- Changing the values present in the BS register, during *Run-Test/Idle* (with *MSEQ1P* active) is allowed by the IEEE 1149.1 Standard [1, page 5-3[1]].
- Holding the values present in the BS register, during *Capture-DR* (with *MSEQ1P* active) is allowed by the IEEE 1149.1 Standard [1, page 5-4[2]].

[1] "… In *Run-Test/Idle*, activity in selected test logic occurs only when certain instructions are present."

[2] "… if capturing is not required for the selected test, then the register retains its previous state unchanged."

4. Conclusion

In this paper we proposed and described a new optional operating mode for the BST infrastructure. This proposal results from an extension of a previous work, regarding the use of this test infrastructure for prototype debug and validation. The proposed operating mode consists of using the BS register (namely the capture / shift stage) to store the values captured at one single functional pin, by means of its associated BS cell. The BS register is configured as a circular buffer though an additional multiplexer placed at the BS cell closest to TDI. Additional hardware includes a two-input AND gate for every BS cell, an m to 2^m decoder and the SP register that contains the position of the selected pin. Notice that this additional circuitry allows any BS cell to be selected as the input-channel of the captured / stored n-bit sequence. This operating mode also implies some modifications in the logic equations of the control signals feeding the BS register, although these modifications do not violate any rule stated in the IEEE 1149.1 Standard.

This new optional operating mode will now be implemented in the prototype device that is currently being used to validate the work described in [12].

5. Acknowledgements

This work has been partially supported by *Fundação para a Ciência e Tecnologia* (FCT) under contract number PBIC/C/TIT/2747/95.

6. References

[1] IEEE Standard Test Access Port and Boundary-Scan Architecture, Oct. 1993, IEEE Std. 1149.1

[2] Jerry Katz, "A Case-Study in the use of Scan in microSPARC™ testing and debug," in *ITC*, pp. 70-75, IEEE Computer Society Press, 1994.

[3] M. F. Lefébvre, "Functional Test and Diagnosis: A Proposed JTAG Sample Mode Scan Tester," in *International Test Conference*, pp. 294-303, IEEE Computer Society Press, 1990.

[4] R. Sedmak, "Boundary-Scan: Beyond Production Test," in *International Test Conference*, pp. 415-420, IEEE Computer Society Press, 1994.

[5] B. E. Whittaker and Nicole A. Doherty, "Boundary Scan Helps Solve Field Failures," in *Evaluation Engineering*, Vol. 35, nº 10, pp. 26-30, Oct. 1996.

[6] Bruce Erickson, "Selecting the Right Debugging Tool," in *Electronic Design*, vol. 43, pp. 83-98, Oct. 1995.

[7] Thomas R. Blakeslee and Jan Liband, "Real-Time Debugging Techniques: Hardware-Assisted Real-Time Debugging Techniques for Embedded Systems," in *Embedded Systems Programming*, vol. 8, nº 4, 1995.

[8] Lee Whetsel, "An IEEE 1149.1 Based Logic/Signature Analyzer in a Chip," in *International Test Conference*, pp. 869-878, IEEE Computer Society Press, 1991.

[9] Andy Halliday, Greg Young and Al Crouch, "Prototype Testing simplified by Scannable Buffers and Latches," in *ITC*, pp. 174-181, IEEE Computer Society Press, 1989.

[10] Gustavo R. Alves, Daniel Aga, Ovidiu Mosuc and José M. Ferreira, "Debug and Test of Microcontroller Based Applications using the Boundary Scan Test Infrastructure," in *Student Forum within the IEEE International Symposium on Industrial Electronics*, IEEE Industrial Electronics Society, 1997.

[11] Gustavo R. Alves and José M. Ferreira, "IEEE 1149.1 Compliance-enable Pin(s): A Solution for Embedded Microprocessor-based Systems Debug and Test," in *Digest of the 1st IEEE International Workshop on Testing Embedded Core-based Systems*, paper 4.3, IEEE Computer Society Press, 1997.

[12] Gustavo R. Alves and José M. Ferreira, "From Design-for-Test to Design-for-Debug-and-Test: Analysis of Requirements and Limitations for 1149.1" in *VLSI Test Symposium*, pp. 473-480, IEEE Computer Society Press, 1999.

Table 2: Logic equations of the BS cell control signals before and after implementing the *MSEQ1P* optional instruction.

Signal	Mandatory instructions [3]	Support of *MSEQ1P* optional instruction
UpdateDR	!TCK·*Update-DR*	Identical
ClockDR	TCK·(*Shift-DR* + *Capture-DR*)	TCK·(*Shift-DR* + *Capture-DR*·!*MSEQ1P* + *Run-Test/Idle*·*MSEQ1P)*
Shift	*Shift-DR* (registered)	*Shift-DR* + *Run-Test/Idle* (registered)
Mode	*EXTEST*	Identical

[3] ! stands for NOT, · stands for AND, + stands for OR.