An Implementation of IEEE 1149.1 To Avoid Timing Violations and Other Practical In-compliance Improvements*

Dave Stang
Astek Corporation
Colorado Springs, CO 80907-2700
dstang@astekcorp.com

R. Dandapani
University of Colorado at Colorado Springs
Colorado Springs, CO 80933-7150
rdan@eas.uccs.edu

Abstract

An implementation of the IEEE 1149.1 standard (JTAG) is presented in this paper. Rules are given for removing gated clocks, registering of all the TAP controller outputs, and daisy-chaining the boundary-scan cell clocks, resets, and control signals in a direction opposite to that of TDI to TDO signal. Several major advantages are obtained as a result of these implementation rules. Timing issues that occur while shifting between the boundary cells when the design is in layout are eliminated. During EXTEST instruction execution, skew is introduced between the toggling pad outputs to minimize damaging power spikes. Due to the elimination of the gated clocks, scan can be inserted without additional DFT logic. A method for inserting scan is given which mostly eliminates timing issues during shifting. Since the TAP controller outputs are fully registered and the gated clocks are recommended to be enables, more observe and control locations are available for an ATPG tool to easily create a high fault coverage pattern for the JTAG logic.

Introduction

The IEEE 1149.1 standard [IEEE93], commonly called JTAG (Joint Test Action Group) standard, was created to address the time and cost issues associated with developing digital systems. The JTAG standard has allowed faster test development time and near 100% stuck-at fault coverage at the board level even when high-density surface mount parts are used. It is also possible, if implemented, to perform various BIST (Built-In Self-Test) functions using only the JTAG ports during board and system testing. JTAG has been widely accepted by industry.

Although the standard describes a systematic method to add JTAG logic to a design and shows examples of the component parts, timing issues can manifest during layout. Due to the placement requirements of the boundary-scan registers, the timing required for the gated clocks, CLOCK-DR, UPDATE-DR, CLOCK-IR, and UPDATE-DR are tight and can be time consuming to solve.

In addition, the gated clocks make it difficult to insert scan and the ability to create ATPG patterns. If scan is inserted after inserting DFT (Design for Test) logic, additional timing issues are introduced in layout. On the other hand if scan is not inserted, extracting functional patterns to test the JTAG circuitry becomes time consuming.

This paper gives a number of recommendations for implementing JTAG logic that will resolve many of the issues mentioned above. Timing problems that occur while shifting between the boundary cells when the design is in layout are mostly eliminated. During the EXTEST instruction execution, skew is introduced between the toggling pad outputs to minimize damaging power spikes. A method for inserting scan is given which mostly eliminates timing issues during shifting. Since the TAP controller outputs are fully registered and the gated clocks are recommended to be enables, more observe and control locations are available for an ATPG tool to easily create a high fault coverage pattern for the JTAG logic. The ideas presented here have been incorporated into a software tool and successfully used on ASIC projects for industry.

An extensive search of the literature (as seen from the list of references) did not yield any of the recommendations given in this paper. The paper first explains the recommendations. Then the experimental findings of applying the ideas to four industrial projects are given, followed by conclusions.

* Several ideas presented in this paper are patent pending by Astek Corporation

JTAG implementation recommendations

The implementation details discussed here are: registering of all TAP controller outputs, removal of gated clocks CLOCK-DR, UPDATE-DR, CLOCK-IR, and UPDATE-IR, and daisy-chaining the boundary-cell clocks, resets, and control signals the opposite direction of the TDI to TDO signal. These implementation recommendations do not affect the compliance to the standard. That is, if implemented, the resulting part will be fully IEEE 1149.1 compliant.

TAP controller: The recommended TAP controller implementation is shown in Figure 1.

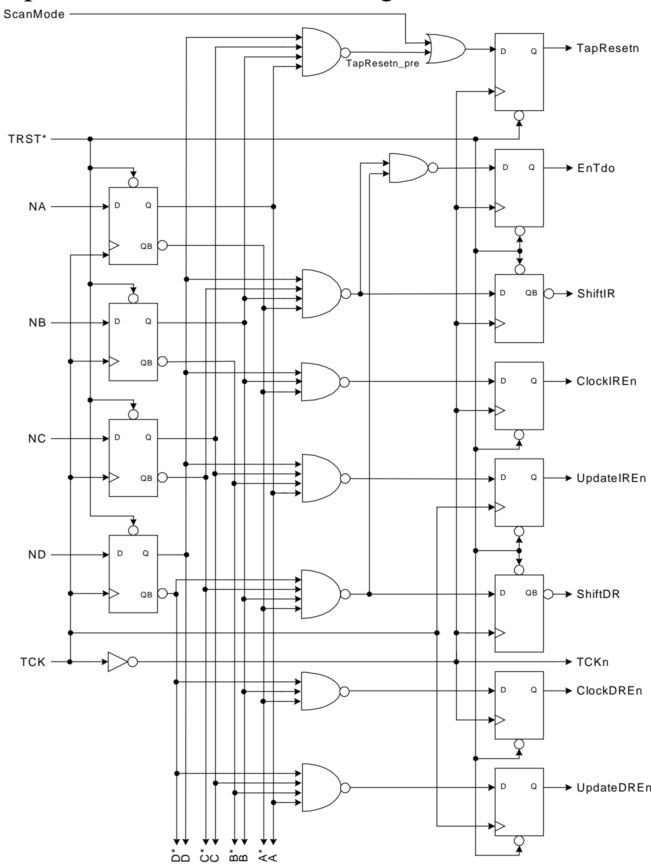

Figure 1: The TAP Controller

The NAND gates do not have TCK or the inverted TCK as inputs. This ensures that enable signals will be used instead of gated clocks in latching data into data and instruction registers. These enable signals are labeled as ClockDREn, ClockIREn, UpdateDREn, and UpdateIREn. ShiftDR and ShiftIR are unaffected because these signals are used to control multiplexers and are not gated clocks.

All the TAP controller outputs are registered on the falling edge of TCK except UpdateIREn and updateDREn, which are registered on the positive edge of TCK. This will allow all the TAP controller outputs to have close to a half TCK clock period setup and hold time to maximize the chance of meeting timing requirements.

Logic has been added to disable the internally generated reset created by the TAP controller state machine. When ScanMode is high, the internal generated reset is disabled to logic high. The asynchronous reset TRST* is unaffected by this logic. This DFT feature allows the ATPG tools to properly scan the JTAG logic. If this logic is not added, the TapResetn signal would go active low every time the TAP state machine flip-flops are in the TEST-LOGIC-RESET state during scan shifting. During all other activities, other than scan, ScanMode is to be low.

The TAP controller generates a TCKn output. It is the inverted signal of TCK. This signal is added to accommodate the shifting of test data on the negative edge of TCK using technology libraries in the industry that do not have negative edge D flip-flops.

The above implementation recommendations do not affect the next state logic given in [IEEE93].

Registers: The recommended Bypass Register, Boundary-Scan Register Cell (Control and Observe), Boundary-Scan Register Cell (Observe only) and Instruction Register implementations are shown in Figures 2-5, respectively. In each of the register implementations the following features have been incorporated:

a. A register flip-flop is clocked by TCK and/or TCKn instead of ClockDR or UpdateIR as give in [IEEE93].
b. A multiplexer is inserted at the data input to the register flip-flops. The multiplexer is controlled by an enable signal to select between new data value and the data fed back from the flip-flop. The operation of the enable signals for various registers is given in Table 1. Since the register flip-flops are receiving an externally controlled clock with enables instead of an internally generated

gated clock, EDA tools (i.e. gate synthesis, layout, & ATPG) can easily manipulate the design and gives the greatest chance of success across different technologies and manufacturing tolerances.

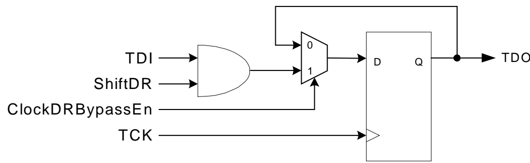

Figure 2: The Bypass Register

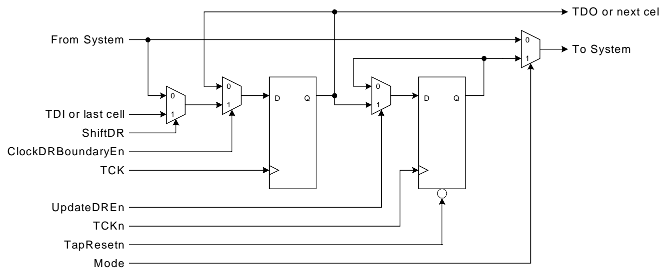

Figure 3: Boundary-Scan Cell to Control and Observe

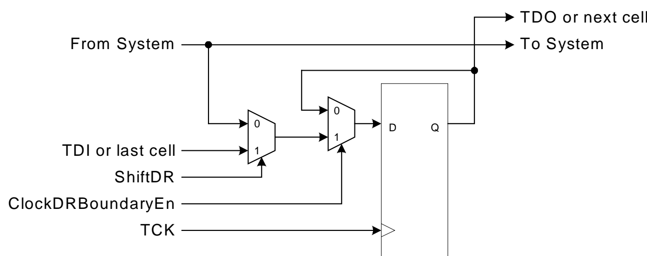

Figure 4: Boundary-Scan Cell to Observe Only

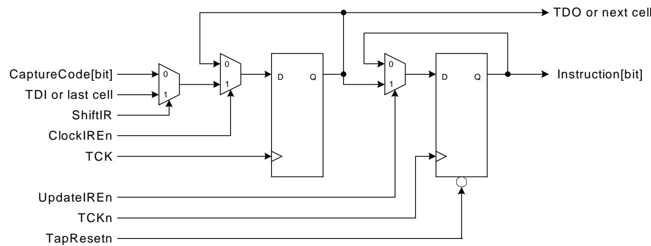

Figure 5: Instruction Register

The gated clocks CLOCK-DR, UPDATE-DR, CLOCK-IR, and UPDATE-IR are removed by changing these signals to enables. As a result, TCK & TCKn will be additional signals required by the test data registers and instruction registers. The enables will be registered for two reasons. These signals are then control and observable for ATPG pattern generation. It will also allow for easy portability across library technologies. As a result of the gated clocks changing to an enable, the shift and update registers will need a feedback loop (i.e. multiplexer).

JTAG Register	**Enable Signal**	**Enable Signal Condition**
Bypass	ClockDRBypassEn	State: CLOCK-DR Instruction: BYPASS, CLAMP, or HIGHZ Sync: To positive edge of TCK
BS Registers (Observe)	ClockDRBoundaryEn	State: CLOCK-DR Instruction: SAMPLE-PRELOAD, EXTEST Sync: To positive edge of the buffered TCK
BS Registers (Control)	UpdateDREn	State: UPDATE-DR Instruction: EXTEST Sync: To positive edge of TCKn.
Instruction (Observe)	ClockIREn	State: CLOCK-IR Sync: To positive edge of TCK
Instruction (Update)	UpdateIREn	State: UPDATE-IR Sync: To positive edge of TCKn

Table 1: Enable signals for registers

An additional note for clarification: a separate ClockDR signal will need to be generated for each test data register (Bypass, Boundary, ID code, User code, Run BIST, User defined) based on the current instruction. This will prevent all test data registers from shifting at the same time when the TAP controller is in the SHIFT-DR state. The IEEE 1149.1 is not clear on this, but it is a requirement if more than one JTAG device exists on a board. A board tester may load up the boundary-scan registers on one integrated circuit and then place that chip in bypass to load the boundary-scan registers in another integrated circuit. This logic is not shown in Figure 1, but is referenced in the modified implementation in Figures 2 through 4.

Daisy-Chain Buffering: When inserting JTAG logic into a design, the boundary-scan cells should follow the order of the package pins from TDI to TDO. The control, reset and clock signals should be routed the opposite way and buffered a predetermined number of boundary-scan cells. An initial estimate of ratio of cells to buffers is 20, but further experimentation may obtain better results. An illustration of the daisy-chain layout is shown in Figure 6.

Figure 6: Daisy-Chain Buffering

The clock and control signals (shown as bold lines in Figure 6) include TCK, TCKn, UpdateDREn, ShiftDR, Mode, ClockDrBoundaryEn, and TapResetn. By following this implementation recommendation, the delay timing created from routing signals in layout, the input logic in the boundary-scan cells, and the buffering of clock and control signals ensure that setup and hold time will be met in the D flip-flops. Scan insertion of the boundary-cells follows the same order from TDI to TDO. Since this is the same path as the TDI to TDO signal, timing for scan signals is met for the same reason that the TDI to TDO signal meets timing. This method also adds a skew to each grouping of boundary-scan cells to smooth out the power spike caused by the output pads toggling when the JTAG EXTEST instruction is implemented.

The advantages of implementing daisy-chain buffering of the boundary-scan clock and control signals is that it nearly guarantees that timing will be meet in layout. This method allows the boundary-scan cells to be easily placed close to the pads to minimize pad input and output timing and frees up routing resources to work on the timing issues in the core of the design.

If scan is inserted, the scan test signal will follow the same path as the JTAG test data. The timing of the scan signal will not be discussed because it is similar to the description of the JTAG test data signal that is explained next.

There are two basic timing scenarios in the daisy-chain buffer chain. The first scenario includes buffering of clock and control signals and is shown in Figure 7 between Pad B and Pad C. The second scenario does not include buffering of the clock and control signals and is shown in Figure 7 between Pad A and Pad B.

In the scenario of a buffer between boundary-scan cells as shown in Figure 7 between Pad B and Pad C, the clock and control signals get to the B boundary-scan cell later than the C boundary-scan cell by the amount of the buffer and routing delays. The test data signal is going the other way and is delayed by the two 2-to-1 multiplexers (or 3-to-1 multiplexer optimized). This ensures the D-flip-flop has more than adequate hold time. Setup time will not be an issue because the buffer and routing delays of the control and clock signals will be significantly less than half TCK period.

In the scenario without a buffer between boundary-scan cells as shown in Figure 7 between Pad A and Pad B, the clock and control signals get to the A

boundary-scan cell later than the B boundary-scan cell by routing delays. The test data signal is going the other way and is delayed by the two 2-to-1 multiplexers (or one 3-to-1 multiplexer optimized) and by routing delays. This ensures the D-flip-flop has adequate hold time. Setup time will not be an issue because the routing delays of the control and clock signals will be significantly less than half TCK period in time.

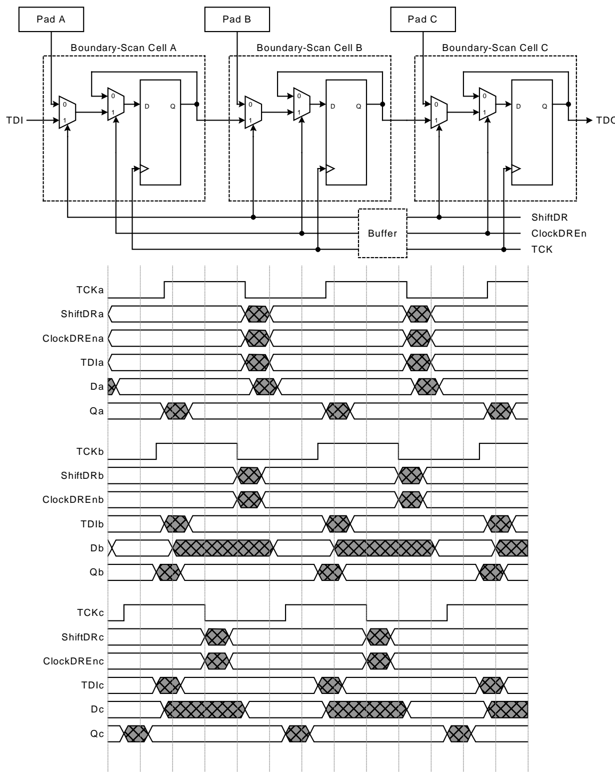

Figure 7: Daisy-Chain Buffering Timing

Pad Output Timing Skew: Since a skew is added to the TCK clock due to the daisy-chain buffering of the TCK signal, the pad outputs toggle at different times during the JTAG EXTEST instruction in groups that the boundary-scan cells are buffered. The buffering skew can be observed between the Pad B and Pad C TCK signal. This effect minimizes possible damaging power spikes due to the current surge caused by all the outputs toggling at the same time.

Calculating the Speed of the Daisy-Chain Buffer Circuitry: A potential downfall of the daisy-chain buffer technique is that it is slower than the tree (or star) buffer insertion method. This may be particularly true on high pin count package integrated circuits. Although slower speeds are obtained, the speed is more than adequate for almost all JTAG applications. An example timing calculation is discussed next.

The maximum speed of the shifting of data can be calculated if timing and load information is known about the buffers. Suppose the buffers that are being used for the daisy chain is as shown in Figure 6 and we are going to add a buffer for every 20 boundary-scan cells. The number of devices the buffer will be driving is 21 (20 boundary-scan cells and the next buffer). A load of 21 devices for this example buffer is about 1.6 ns as shown in Figure 8. If the boundary-scan chain being implemented is 300 cells long, the control, clock, and reset boundary-scan signals will traverse through (300 cells) / (20 cells per buffer) = 15 buffers. This will take a total time of (15 buffers) * (1.6 ns/buffer) = 24 ns. The 24 ns would be the time for half a TCK clock cycle. The frequency would be the inverse of double this number: 1/(2 * 24 ns) = ~ 20.8 MHz.

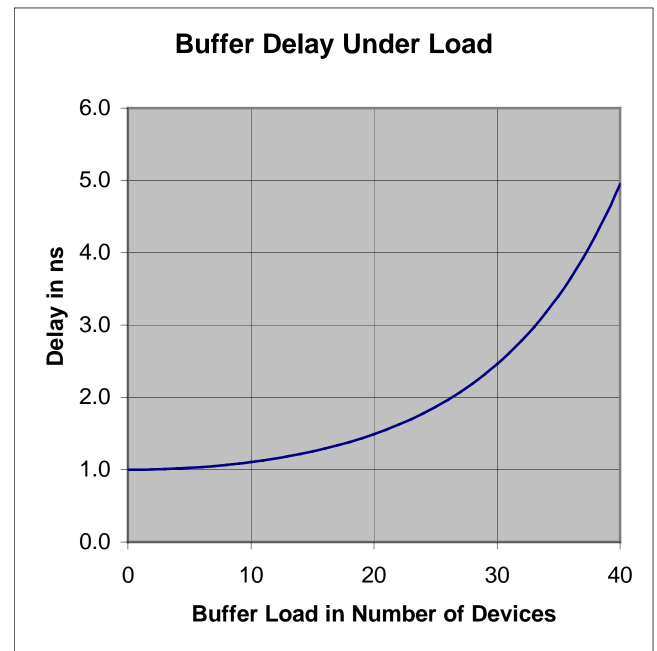

Figure 8: Buffer Delay Under Load Example Curve

Findings

The modified IEEE 1149.1 design implementation has been incorporated into a commercially available software tool called ABC (Astek Boundary-scan Compiler). The buffering of the control and clock signals was set to occur every 20 boundary-scan cells using a large buffer. The method and design described in this paper has been used on four mid-

sized and one small ASIC project. All four parts are currently being manufactured and tested with no noted JTAG issues from the board manufacturers. A summary of two of these designs is shown in Table 2 with more details of each example design in Table 3 and Table 4.

JTAG Module	Cell count	Module Instances	Total
TapController 2-bit instruction w/ CLAMP	100	1	100
Daisy-chain buffers	8	6	42
BC_1	6	103	618
BC_4	3	11	33
		Total	793

Table 2: Data Results from two Designs

JTAG Module	Cell count	Module Instances	Total
TapController 2-bit instruction	81	1	81
Daisy-chain buffers	8	13	104
BC_1	8*	168	1344
BC_4	5*	91	455
		Total	1984

Table 3: Detail Data from Example 1
Note*: Over-constraining of design resulted in additional buffers to meet timing.

Ex	Total Cells	JTAG Cells	Boundary Scan Length	Tech	Total ATPG
1	26254	793	114	0.35u	98.3%
2	22113	1984	259	0.35u	98.8%

Table 4: Detail Data from Example 2

Layout Issues: The boundary-scan cells were moved to an area close to the pad. This insured the pad input and output timing were met in mission mode. Running the simulation ATPG and JTAG patterns using timing generated from layout resulted in not having to add buffers to meet timing across manufacturing tolerances in all projects. Since a skew is added to the TCK clock by the daisy-chain buffering of the TCK signal, the pad outputs toggle at different times during the EXTEST instruction. This technique minimizes possible damaging power spikes due to the current caused by all the outputs toggling at the same time. In all the projects that had the JTAG inserted using the techniques describe in this paper, none of the parts have had any reported power damage.

Figure 9 shows setup and figure 10 shows hold from the design example 1 using data from post layout. It was verified that no additional buffers were added in the TDI to TDO flow. The technology used was 0.35um and the D flip-flop setup time requirement of 1.826ns and hold time requirement of 0.179ns. The TCK clock was constrained to 10ns (100 MHz) for illustration purposes. The high speed obtained is due to

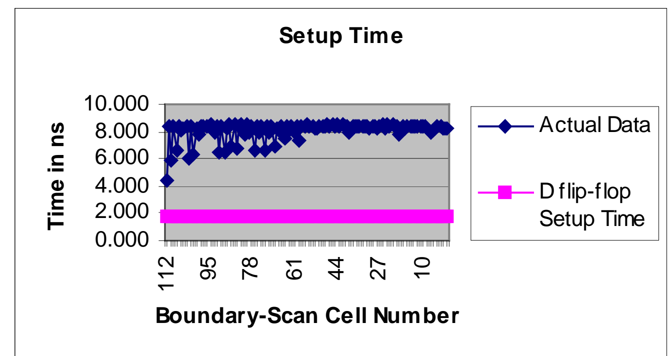

Figure 9: Setup Timing Results from Design Example 2

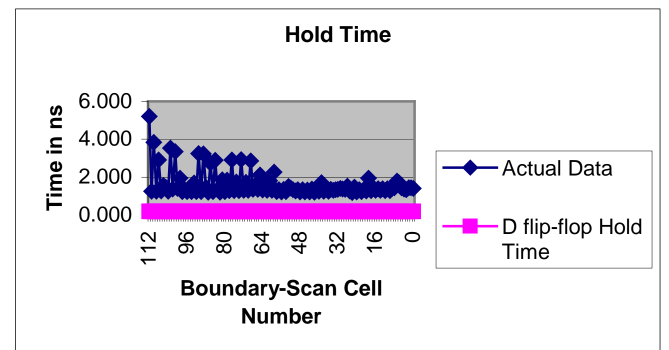

Figure 10: Hold Timing Results from Design Example 1

the short boundary-scan cell length of 114. The setup time had a range of 4.427 to 8.500 ns. The hold time had a range of 1.183 to 5.210 ns.

Gate Count: More gates are used in the modified design. Although the gate count in the individual blocks of the modified implementations are more than the IEEE example, the number of buffers required to meet timing in layout is most likely

reduced. This will become particularly true in large package integrated circuits. An experiment to compare gate count (including buffers) between the two methods was not performed because it is perceived that the minimal gate count required for test circuitry by the industry is not critical.

JTAG Simulation Speeds: The largest ASIC project had a boundary-scan chain of 292 cells and the JTAG circuitry was implemented .35um technology library. It was verified that the JTAG logic would work at a speed of 30 MHz. A disadvantage of using this insertion method is that a maximum speed will not be obtained, but a speed of 30 MHz was more than adequate. Additional speed may be possible by modifying both the size of the buffer and the number of boundary-scan cells each buffer drives. This was not tried because additional speed was not required or needed. It should be also noted that a faster JTAG design is a trade-off with power spikes during the EXTEST instruction when all the outputs are possibly toggling.

Scan Insertion and ATPG: Scan was inserted into several medium ASIC projects. The scan chain followed the same path as the TDI to TDO signal. Scan was inserted with minimal scan tool effort. The resulting ATPG patterns resulted in >95% stuck-at fault coverage in the JTAG logic in all projects. The untested logic was mostly due to the masking of the internal generated TAP reset.

Conclusions

A number of recommendations to implement JTAG logic are introduced in this paper. The recommendations were shown to be valid using three different verification methods: 1. The resulting implementation follows all the rules in [IEEE93], 2. It was shown that the resulting implementation is functionally equivalent to the examples given in [IEEE93], and 3. The resulting implementation has been successfully implemented on several industrial designs.

The implementation recommendations proposed in this paper provides a systematic method to implement JTAG into an integrated circuit that is portable across technology libraries and nearly guarantees timing in layout. In addition, this same method of meeting timing, can be used effectively to meet timing for scan insertion which allows fast and effective manufacture testing using ATPG patterns. This method also inserts skew between the output pads toggling during the EXTEST instruction to help minimize power surges. The time and effort that is eliminated in layout and for IC test pattern generation makes these recommendations an important addition to any JTAG insertion tool.

References

[Bhavsar97] "A Method for Synchronizing IEEE 1149.1 Test Access Port for Chip Level Testability Access", Dilip Bhavsar, Proceedings of the 11th International Conference on VLSI Design, Chennai, India, January 1998, pp. 289-292

[Bleeker93] "Boundary-Scan Test A Practical Approach", Harry Bleeker, Peter van den Eijnden, Frans de Jong, Kluwer Academic Publishers, 1993

[Bruce91] "Implementing 1149.1 on CMOS Microprocessors", W. C. Bruce, Michael G. Gallup, Grady Giles, and Tom Munns, International Test Conference Proceedings, Nashville, TN, October 1991, pp. 879-886

[Caldwell92] "Is IEEE 1149.1 Boundary Scan Cost Effective: A Simple Case Study", Barry Caldwell and Tom Langford, International Test Conference Proceedings, Baltimore, MD, October 1992, pp. 106-109

[Chiles91] "Using Boundary Scan Description Language in Design", Dick Chiles and John DeJaco, International Test Conference Proceedings, Nashville, TN, October 1991, pp. 865-868

[Eklow94] "Optimizing Boundary Scan in a Proprietary Environment", William Eklow, International Test Conference Proceedings, Washington DC, October 1994, page 1024

[Greene93] "Benefits of Boundary-Scan to In-Circuit Test", David A. Greene, International Test Conference Proceedings, Baltimore, MD, October 1993, page 263

[Hilla92] "Boundary Scan Testing for Multichip Modules", Stephen C. Hilla, International Test

Conference Proceedings, Baltimore, MD, October 1992, pp. 224-231

[IEEE93] "IEEE Standard Test Access Port and Boundary-Scan Architecture", IEEE Standard 1149.1-1993, IEEE Standards Board, New York, Institute of Electrical and Electronics Engineers, October, 1993

[Jarwala94] "Designing 'Dual Personality' IEEE 1149.1 Compliant Multi-Chip Modules", Najmi Jarwala, International Test Conference Proceedings, Washington DC, October 1994, pp. 446-455

[Jong91] "Testing the Integrity of the Boundary Scan Test Infrastructure", Frans de Jong, Fran van der Heyden, International Test Conference Proceedings, Nashville, TN, October 1991, pp. 106-112

[Lefebvre90] "Functional Test and Diagnosis: A Proposed JTAG Sample Mode Scan Tester", Mark F. Lefebvre, International Test Conference Proceedings, Washington DC, September 1990, pp. 294-303

[Muris90] "Integrating Boundary Scan Test into an ASIC Design Flow", Math Muris, International Test Conference Proceedings, Washington DC, September 1990, pp. 472-477

[Oakland00] "Considerations for Implementing IEEE 1149.1 on System-on-a-Chip Integrated Circuits", Steven F. Oakland, International Test Conference Proceedings, Atlantic City, NJ, October 2000, pp. 628-637

[Parker92] "The Boundary-Scan Handbook", Kenneth P. Parker, Kluwer Academic Publishers, 1992

[Parker94] "Observations on the 1149.x Family of Standards", Kenneth P. Parker, International Test Conference Proceedings, Washington DC, October 1994, page 1023

[Parker98] "The Boundary-Scan Handbook Second Edition Analog and Digital", Kenneth P. Parker, Kluwer Academic Publishers, 1998

[Raymond95] "Algorithmic Extraction of BSDL from 1149.1-compliant Sample ICs", Douglas W. Raymond, D. Eugene Wedge, Philip J. Stringer, Harold W. Ng, Suzanne T. Jennings, Craig T. Pynn, and Winsor Soule, Jr, International Test Conference Proceedings, Washington DC, October 1995, pp. 561-568

[Reed95] "Improving Board and System Test: A Proposal to Integrate Boundary Scan and Iddq", Douglas Reed, Jason Doege, and Antonio Rubio, International Test Conference Proceedings, Washington DC, October 1995, pp. 577-585

[Robinson93a] "Technology Independent Boundary Scan Synthesis (Design Flow Issues)", Markus F. Robinson, European Design Automation Conference, Los Alamitos, CA, 1993, pp. 416-421

[Robinson93b] "Technology-Independent Boundary Scan Synthesis (Technology and Physical Issues)", Markus F. Robinson, Frederic Mailhot, and Jin Konsevich, International Test Conference Proceedings, Baltimore, MD, October 1993, pp. 157-166

[Singh97] "A Symbolic Simulation-based ANSI/IEEE Std 1149.1 Compliance Checker and BSDL Generator", Harbinder Singh, Girish Patankar and James Beausang, International Test Conference Proceedings, Washington DC, October 1997, pp. 256-246

[Standbridge01] "Suggested Design Provisions for Boundary-Scan Test", James Standbridge, Electronic Product Design, UK, January 2001, pp. 28-34

[Whetsel95] "Improved Boundary Scan Design", Lee Whetsel, International Test Conference Proceedings, Washington DC, October 1995, pp. 851-860

[Xianyang99] "An Improved Design of JTAG Controller and Its Logic Implementation", Jiang Xianyang, Shen Xubang, Zhang Tianxu, Journal of Huazhong University of Science and Technology, January 2001, pp. 63-65

[XuBang93] “Design and Implementation of a JTAG Boundary-Scan Interface Controller”, Shen Xubang, Liang SongHai, Proceedings of the Second Asian Test Symposium, Beijing, China, November 1993, pp. 215-218.