10

Boundary-Scan Testing

10.1 Introduction

Although the emphasis of this book is on IC testing and designs for testability, it is important to include a chapter on Boundary-Scan, which was initially a DFT technique for printed circuit boards (PCBs). This technique requires that ICs include extra hardware to facilitate communication between the board and the various ICs on which they are mounted during testing. Also, with the present trend of a system on a chip (SOC), the ICs themselves consist of several embedded complex and diverse modules and start to beg for a DFT technique that facilitates their testing. Understanding Boundary-Scan can help in handling SOCs, as we describe in Chapter 15.

It has been customary through the late 1970s to access the various ICs on a board through a bed-of-nails fixture in an in-circuit tester. Because of the increase in device density, decrease in board size, and change to surface mounting in packaging, accessing the chips' test points became impractical. A small group of European electronics companies met in the Netherlands to discuss this problem in 1985. From the onset, consensus had been reached about the gravity of the problem and the need for a solution. The group which adopted the name *Joint Test Action Group* (JTAG), described the problems with board testing and proposed a methodology to resolve them. This proposal took the name JTAG Test Access Port and Boundary-Scan Architecture [JTAG 1988]. By 1988, the proposal gained widespread interest from industry and the support of the IEEE Computer Society. In 1990, the methodology became IEEE/ANSI Standard 1149.1 [IEEE 1990]. It was quickly adopted by the industry, to such an extent that it is now a standard feature in CAD tools and is supported by board test equipment. As early as 1990, it was embedded in some FPGAs, digital signal processors (DSPs), and other ASICs.

The use of Boundary-Scan has been extended beyond boards to circuits. [Hansen 1989] For example, it is useful for internal testing and to run BIST. In addition to testing, Boundary Scan was adopted in debugging and diagnosis using in-circuit emulation protocols [Parker 1989]. Despite the proliferation of its use, the term *Boundary-Scan* as used in this chapter is synonymous with IEEE Standard 1149.1.

In this chapter we review briefly traditional board testing and determine the rationale for the adoption of a DFT technique to facilitate its testing. In Section 10.3, the architecture of Boundary-Scan is described, and its components are detailed in Section 10.5. The TAP controller is the topic of Section 10.4. With the knowledge acquired as to the

Copyright © 2000 by John Wiley & Sons

architecture and controller, the modes of Boundary-Scan operation are given in Section 10.7. Boundary-Scan description language (BSDL) is the topic of Section 10.8. As with any other DFT technique, Boundary-Scan has its cost, briefly discussed in Section 10.9.

10.2 Traditional Board Testing

Through the late 1960s, PCBs included only discrete components and required a customized test bench. Testing could easily be completed in 3 to 4 minutes per board. As they became populated by ICs, their complexity has increased and their testing time has reached 3 to 4 hours per board [GenRad 1989]. As ATE equipment became driven by software, the testing throughput has increased.

Board testing used to consist of functional testing and in-circuit testing. The advantage of functional testing is that it can be done at speed. However, there are several disadvantages. First, it requires a long preparation time. Second, since it does not use a fault model, there is no way to verify its quality. In addition, hardly any diagnostics are possible. In-circuit testing allows testing individual components, discrete or ICs, on a fully assembled PCB. For this, all nodes must be accessible. In addition, all components need to be isolated during testing to avoid the device under test being affected by the other components on the board. This really amounts to partitioning the board and reducing testing complexity. It also allows diagnostic testing and isolating the faulty components.

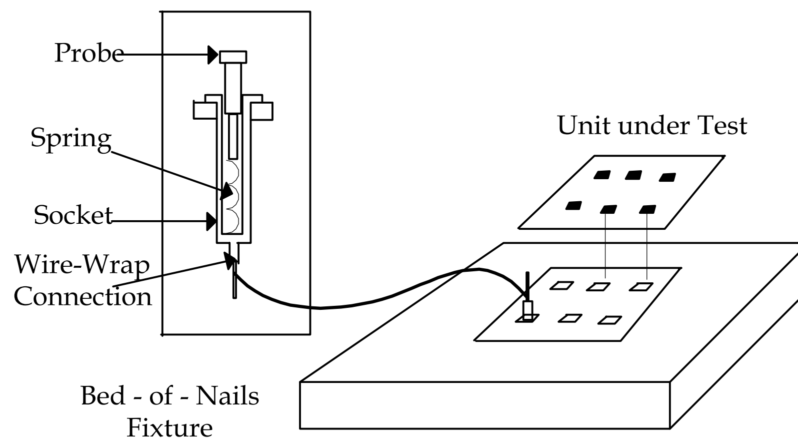

Figure 10.1 In-Circuit Testing Fixture

This type of testing was practiced through the early 1980s using a *bed-of-nails* fixture. The nails on this fixture are small spring-loaded probes that touch the tracks and device leads on the unit under test (UUT). These nails are mounted on the fixture so that each nail lines up with a device pin or a via. The bottom of the spring-loaded probes is wire-wrapped to the socket as illustrated in Fig. 10.1. Connection to the tester is realized by wiring the sockets to the ATE system. Each

Copyright © 2000 by John Wiley & Sons

custom-built bed-of-nails fixture typically has hundreds of test nails that are positioned to contact the solder pads on the bottom side of the UUT.

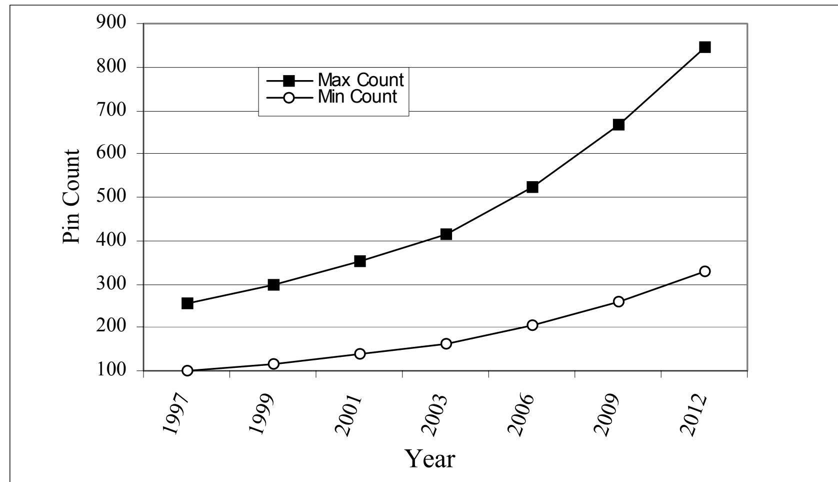

Figure 10.2 PCB Technology Trends: a) Shrinking Trace Distance,
b) Use of Surface Mount Packaging

There are some problems with in-circuit testing. First, the fixture is custom made and hence is expensive and requires a long lead time. The fixture cannot be built prior to finalizing the layout of the design. Yet, waiting until completion of the layout delays testing and therefore increases time to market. Second, the spring-loaded nails are easily damaged. Third, sometimes when testing an IC on the board, the appropriate input signals are not easily obtainable from the output of the driving IC; these signals have to be forced by the tester. This process may cause overheating of the test points and require long cooling periods; thus testing time can be prolonged. Also, the injected signals may produce noise on the board that results in wrong test response and damaging latch-up effects in CMOS circuits. These problems are exaggerated as the trace between the pins gets smaller, due to an explosion in the pin count as indicated by the trend shown in Fig. 10.2 for hand-held ICs [SIA 1997].

For modern boards, the challenge to in-circuit testing is the change in IC packaging along with rapid adoption of surface-mounted packages. The problem with these packages is pin density; that is, the pitch is getting too small. In addition, often the ICs are mounted on both sides of the board. Physical probing of test points on the board has become so difficult and costly that it motivated the search for a solution. Boundary scan is a promising alternative. According to [Parker 1998], it “actually helps one prolong the life of the in-circuit approach, because it allows the reduction of the number of nails needed to test a board while maintaining fault coverage."

Copyright © 2000 by John Wiley & Sons

10.3 Boundary-Scan Architecture

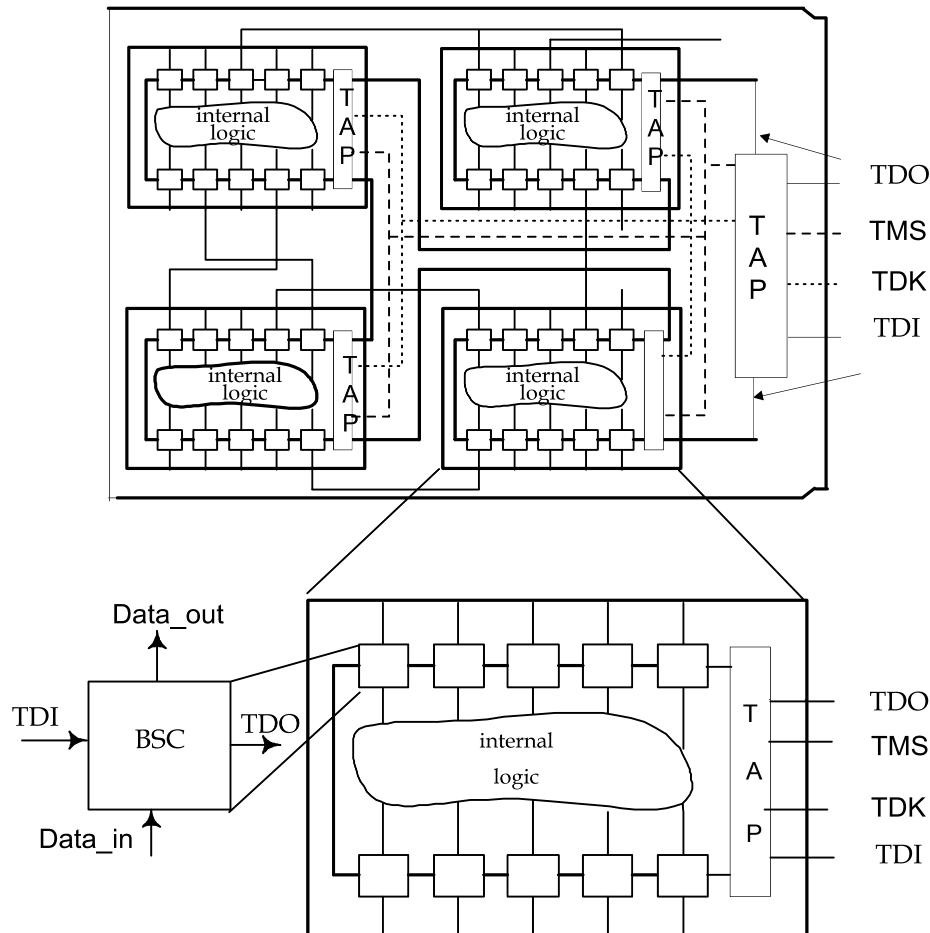

Figure 10.3 IEEE 1149.1 Boundary Scan Architecture

The general IEEE Standard 1149.1 Boundary-Scan architecture is shown in Fig. 10.3. This configuration requires that the board and each IC that is part of the Boundary-Scan include the following principal hardware components:

- A test access port (TAP) with four to five pins
- A group of registers: an instruction register (IR) and data registers (DRs)
- A TAP controller, a 16-state finite state machine

Four mandatory pins drive the TAP. They include two data pins, test data input (TDI) and test data output (TDO), and two control pins, test mode select (TMS) and test clock (TCK). It is important to mention that it is possible that some of the chips are not complying with the standard and their testing is not secured through the TAP.

Copyright © 2000 by John Wiley & Sons

The instruction register receives the instruction, then decodes it to perform the operations on the data registers. The instructions, which are described later in the chapter, include load, shift, or pause registers: IR or DR. There are two mandatory data registers, a bypass register and boundary scan register (BSR), which consists of Boundary-Scan cells (BSC).

The I/O signals of all chips enter or leave the chip through the *Boundary-Scan Cell* (BSC), that is described in Section 10.5.1. All such cells are provided with controls to allow them to be configured in a shift register and form a collar around the core logic of the chip. The TAP controller manages the exchange of data and instructions among the board and chips. Test data management is done according to a protocol dictated by the Standard. As with any standard this one, evolves according to changes in technology and design practices. In addition, due to the emergence of mixed signal chips, a new standard, P1149.4, is being established to handle analog I/O [IEEE 1994]. The reader who plans to use the standard should consult the IEEE documentation [IEEE 1990, Maunder 1990].

During normal operation, the extra hardware on the chip or board is transparent. However, during testing mode, all input signals are scanned in from the in port, TDI, and scanned out through the out port, TDO, to the board. The information from all chips is passed to the outside of the chip through a shift register formed by flip-flops around the contour of the chip. That is, every I/O pin is registered, and all flip-flops may be organized as a shift register in a fashion similar to the scan-path technique that we studied in Chapter 9.

The above described Boundary-Scan architecture allows configuring the cells for the following testing modes:

- *External testing*: interconnects between the chips
- *Internal testing*: testing of the logic within the chip

10.4 The Test Access Port

In this section we describe the various signals used by the TAP controller. They are the signals applied to the four mandatory pins and the optional test reset (TRST). The latter pin can reset the test logic asynchronously. The four mandatory pins include two data pins, test data input (TDI) and test data output (TDO), and two control pins, test mode select (TMS) and test clock (TCK). As illustrated in Fig. 10.3, the TDI of the board is connected to that of the first chip. Also, the TDO of this chip is connected to the TDI of the next chip in the chain, and so on. The TDO of the last chip in the chain is connected to that of the board's TAP. The two control signals are, however, connected to the TAPs of all the chips.

TDI and TDO are to Boundary-Scan what scan-in and scan-out are to scan path design. However, unlike scan-path design, the pins may not be one of the primary input/output pins on the chip that can be multiplexed for the two functions

Copyright © 2000 by John Wiley & Sons

as explained in Chapter 9. Both TDI and TDO are also connected to the TAP registers. Both TMS and TCK are distributed to all the chips on the board that are part of the scan design. It is very likely that the board will include logic that does not have Boundary-Scan cells.

TDI. This is the test data input which allows the introduction of test data in a fashion similar to the scan-in pin in traditional scan path. The TDI of the board is connected to its counterpart on the first chip in the scan chain. This signal is shifted in the registers at the positive edge of the TCK and, when not in use, is kept high.

TDO. This is the test data output which allows scanning out of the test data in a fashion similar to the scan-out pin in traditional scan path. The TDO of the board is connected to its counterpart on the last chip in the scan chain. Data are shifted out at the negative edge of TCK. When not in use, this input signal is kept in high impedance.

TCK. This is the test clock, which operates the testing function synchronously and independent of the system clock. It controls the transfer of data and instructions among the TAP registers and shifting the data within any of the registers.

TMS. This is the test mode select. The input stream to this pin is interpreted by the TAP controller and used to manage the various test operations. The signals are interpreted by the TAP controller, which is discussed in Section 10.6. When the TMS is not in use, it must be held high. For this, the standard requires that this be done automatically.

TRST. This is an optional signal whose purpose is to reset all testing logic asynchronously and independent of TCK. It may also be used for reset at power-on.

10.5 Registers

Figure 10.4 shows the various registers that support the testing of the board. Of these registers, only three are mandatory: the instructional register (IR), the bypass register, and the boundary-scan register (BSR), which consists of collections of the BSCs. To discuss the BSR, we first need to know about the functionality of the BSC. A brief description of each of these registers is given in this section.

10.5.1 Boundary-Scan Cell

An example of a commonly used Boundary-Scan cell (BSC) is shown in Fig. 10.5. This cell may be used at the input or output pins. During normal operation the input signal is applied to the data-in pin and passes to the internal

Copyright © 2000 by John Wiley & Sons

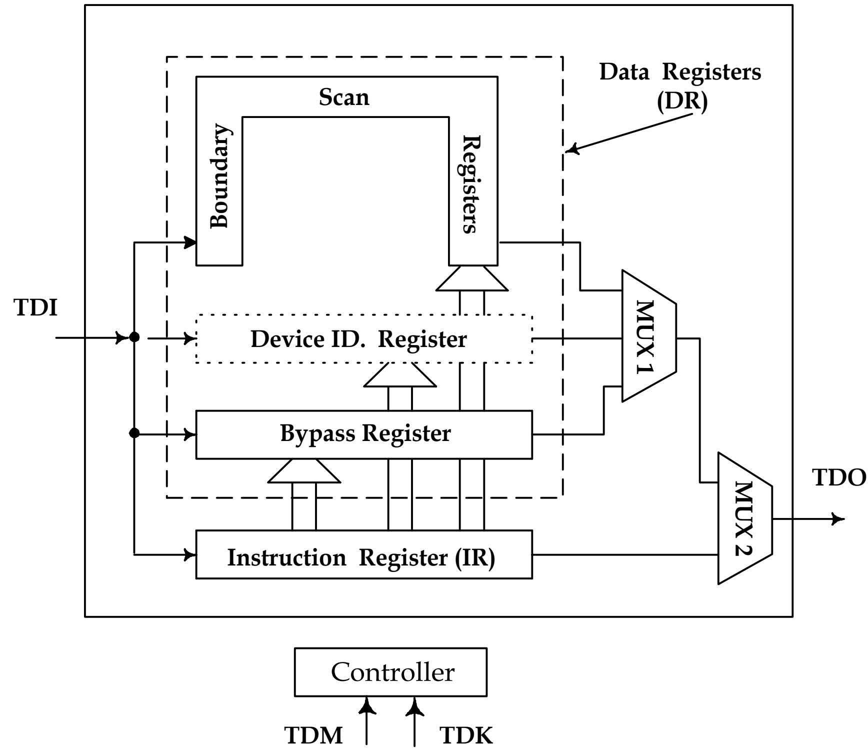

Figure 10. 4 Test Access Port -- The Registers

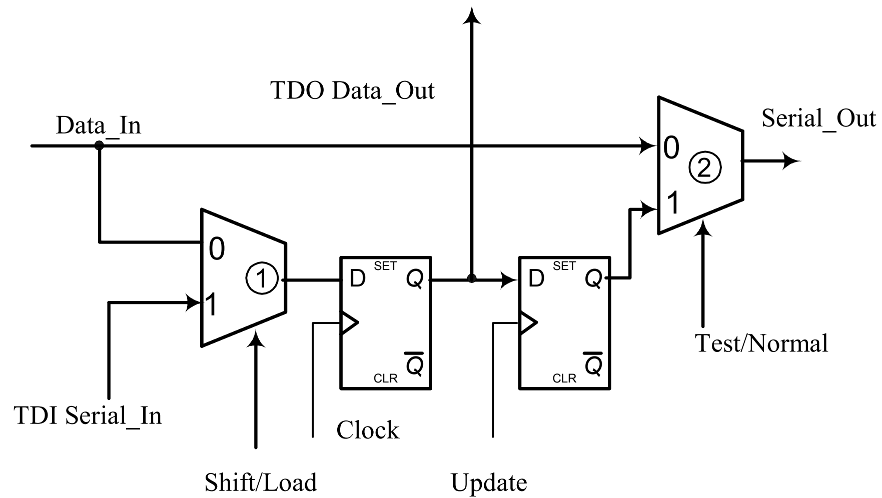

Figure 10.5 The Basic Boundary Scan Cell

logic through MUX 2. Thus the value of Test/Normal should be 0, while the Shift/Load mode may be either 0 or 1. When the same cell is used as an output pin, the data-in are from the internal logic of the chip and pass, through MUX 2, to the output of the chip.

Copyright © 2000 by John Wiley & Sons

For test mode, the data are coming through TDI; thus the Shift/Load mode should be 1. The data are latched for internal testing or to be shifted to the next BSC when the clock is activated. For internal testing, the signal needs to pass through the second storage device by enabling the Update signal. It is then passed to the chip, with Test/Normal being held high. The various configurations of the cell are also shown in Fig. 10.5.

10.5.2 Bypass Register

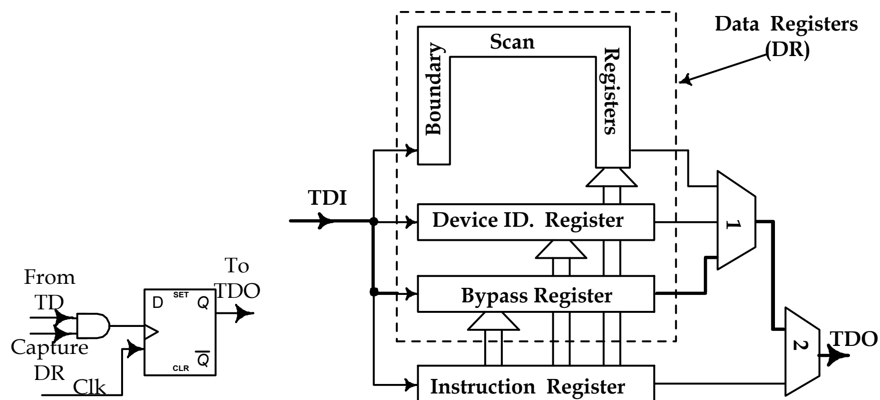

Figure 10.6 The Bypass Register: (a) Functional Representation, (b) Flow of Data

The bypass register is a single-stage register, as shown in Fig. 10.6. It is set to logic 0 at the rising edge of TCK when the TAP controller is in the Capture-DR state. Use of the bypass register allows the signal at the TDI to pass directly to the TDO of the chip, thus bypassing all the other BSCs of the chip. This approach is very useful. For example, if the board has, say, 30 chips each having 100 BSCs, there are 3000 stages in the Boundary-Scan register. Instead, when the bypass is used, there are only 129 stages in the BSR: 100 for the chips under test and one each for the other 29 chips on the board. Such an arrangement results in a reduction of both the testing time and the testing data.

10.5.3 Boundary-Scan Register

The Boundary-Scan register consists of all the BSC cells on the periphery of the chip. It is part of the testing of the interconnects and of any logic between the boundary-scan ICs on the board. This includes logic not configured in the Boundary-Scan as well as any ROM or RAM. In addition, it allows sampling and examination of the input and output signals without interfering with operation of the core logic.

Copyright © 2000 by John Wiley & Sons

10.5.4 Instruction Register

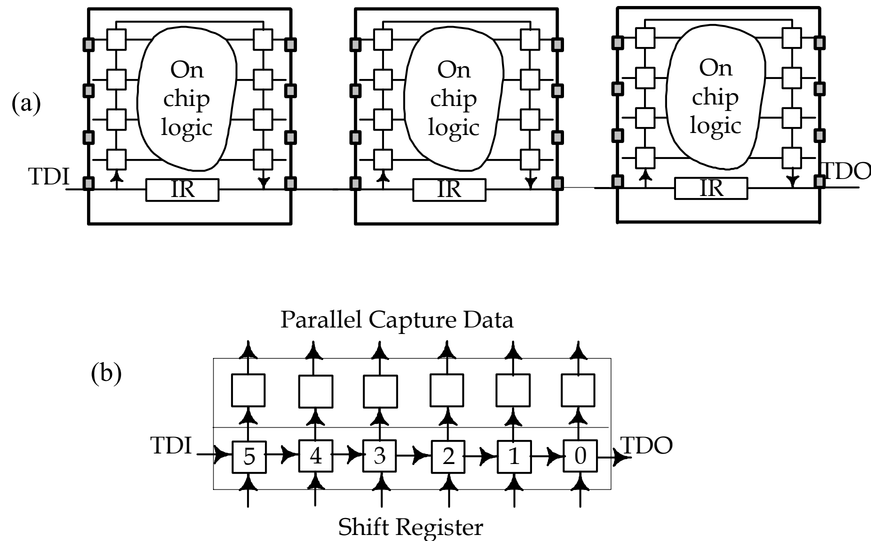

Figure 10.7 The Instruction Register

The instruction register is a serial-in, parallel-out register. Each flip-flop of the register is connected to an output latch as illustrated in Fig.10.7. The latch holds the current instruction bits latched into it from the shift register in the Update-IR state. The IR must contain at least two shift-register cells that can hold instruction data. These two mandatory cells are located nearest the serial output. They are the least significant bits (LSBs), as illustrated in Fig. 10.7.

10.5.5 The Device Identification Register

The device identification register is optional, but if included on the IC, it should comply with the standard. It must be a 32-bit -ong parallel-in and serial-out. It is intended to contain the manufacturers' number and the version number. This information facilitates verifying that the correct IC is mounted in a correct position and that it is the correct version of the chip. Unlike the other registers, the information is not passed to an output latch.

10.6 TAP Controller

The TAP controller responds to the TMS signals at the positive edge of TCK. Its main functions are (1) to load the instructions in the IR, (2) to provide control signal to load and shift the test data into TDI and out of TDO, and (3), to perform some test actions, such as capture, shift, and update test data. The TAP controller is a 16-state finite state machine that operates synchronously with TCK. Its state diagram is shown in Fig. 10.8 as defined by IEEE Standard 1149.1. Some

Copyright © 2000 by John Wiley & Sons

of the states correspond to actual operations on the data (DR) or the instructions (IR), while others allow some flexibility in the flow of operations.

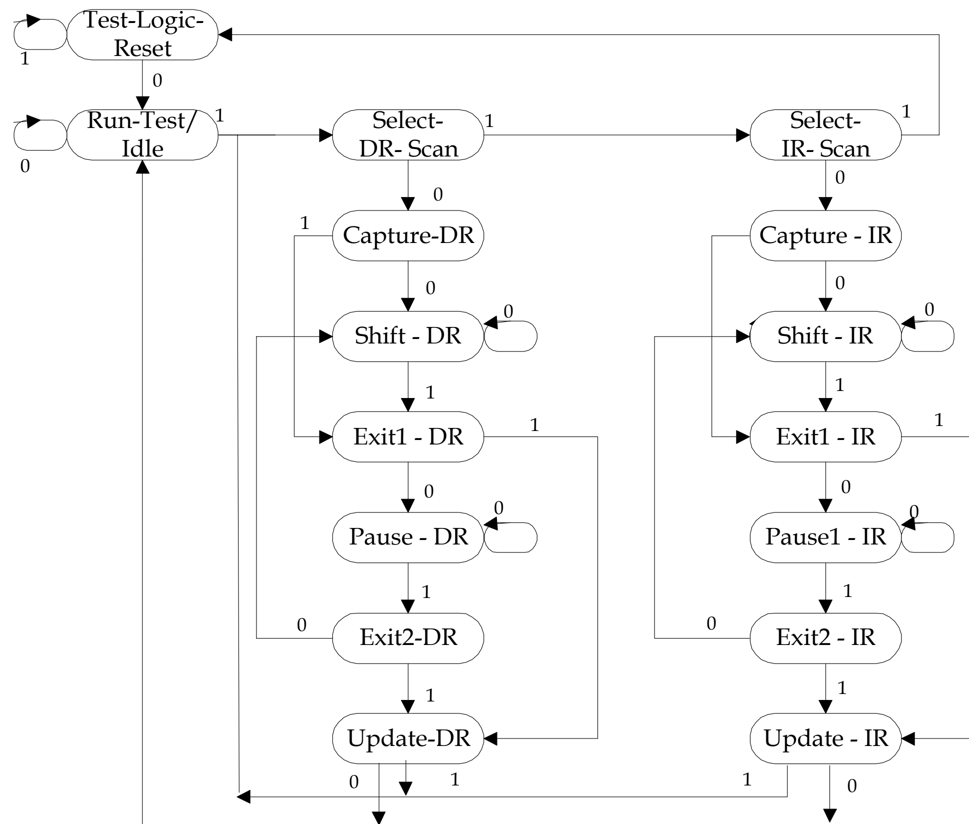

Figure 10.8 Test Access Port Controller

10.6.1 Controller's States

At power on, the controller is in the Test-Logic-Reset state. It remains in this state as long as TMS is high and the circuit is in normal operation mode. This state may also be reached if while TMS is high, TCK is clocked five times. As soon as TMS changes to logic 0, the controller is in the Run-Test-Idle state. In this state, either BIST may be initiated or the TAP idles between scan operations for other testing modes.

To start testing, the instruction needs to be loaded into IR. For this, TMS is held high and the TCK is clocked twice for the controller to reach the Select IR–Scan state. Now TDI and TDO are connected to IR, and all IR registers on the board are serially connected. Next, the controller passes to the Capture-IR state with TMS = 0. Once the instruction is loaded in the IRs, then, with TMS still low, the controller stays in the Shift-IR state for as many clock cycles as needed by the test mode. In this state, the previously captured data are shifted via TDI and via TDO, one shift register stage on each rising

Copyright © 2000 by John Wiley & Sons

edge of the TCK pulse. If shifting is not needed, TMS = 1 and the controller bypasses Shift-IR and enters the Exit1-IR. The latter state as well as any of the other exit states is temporary. At the next positive edge of the clock, there is transition to another state. If TMS = 0, the next state is Pause-IR, and the control remains in this state until TMS = 1. The Pause-IR state is needed when the shift is done in a chain of different lengths. From this state, the control goes to Exit2-IR, then to Shift-IR if TMS = 0, or to Update-IR, if TMS = 1. The controller enters this state once the shifting process has been completed. The new data are latched into their parallel outputs of the selected data registers at the falling edge of the TCK. Depending on the value of TMS, the next state is either Run-Test-Idle or Select DR-Scan.

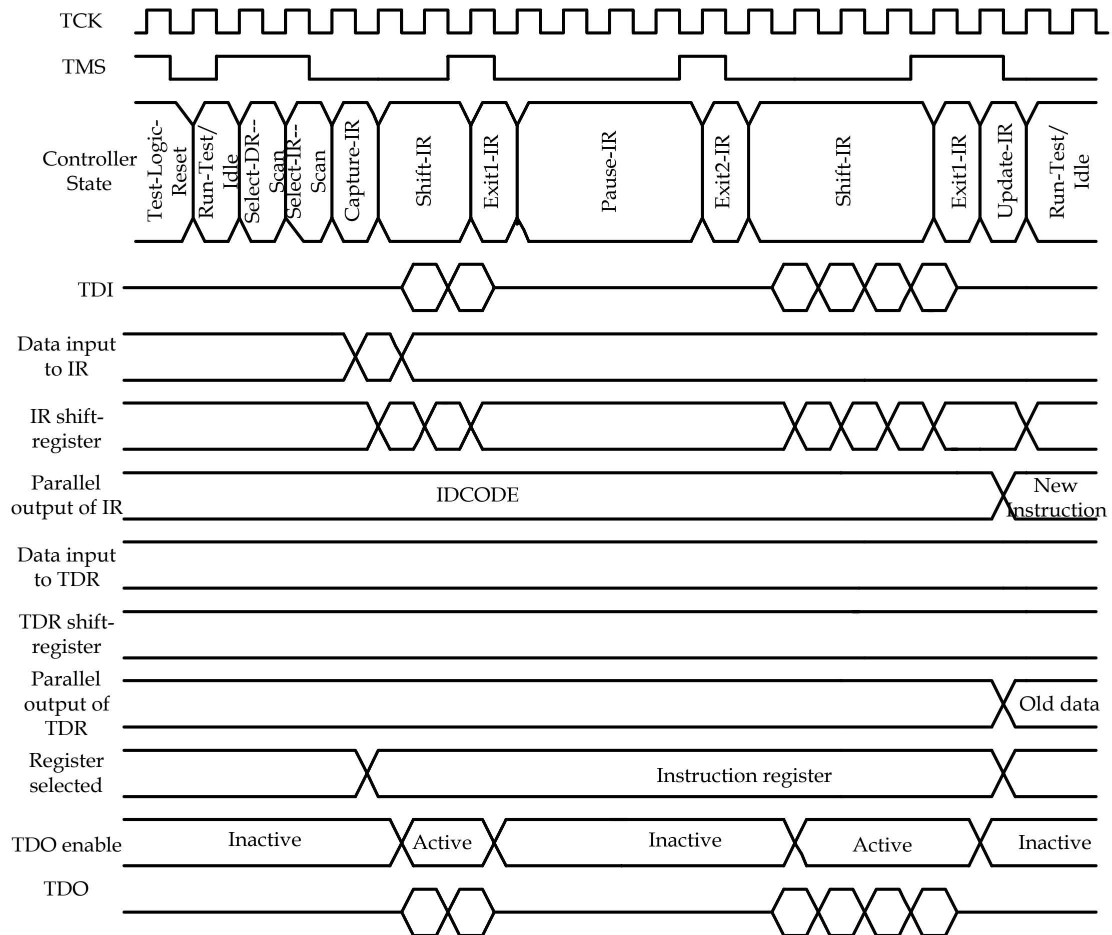

Figure 10.9 TAP Controller: Example of Timing Diagram, Instruction Scan.

Copyright © 2000 by John Wiley & Sons

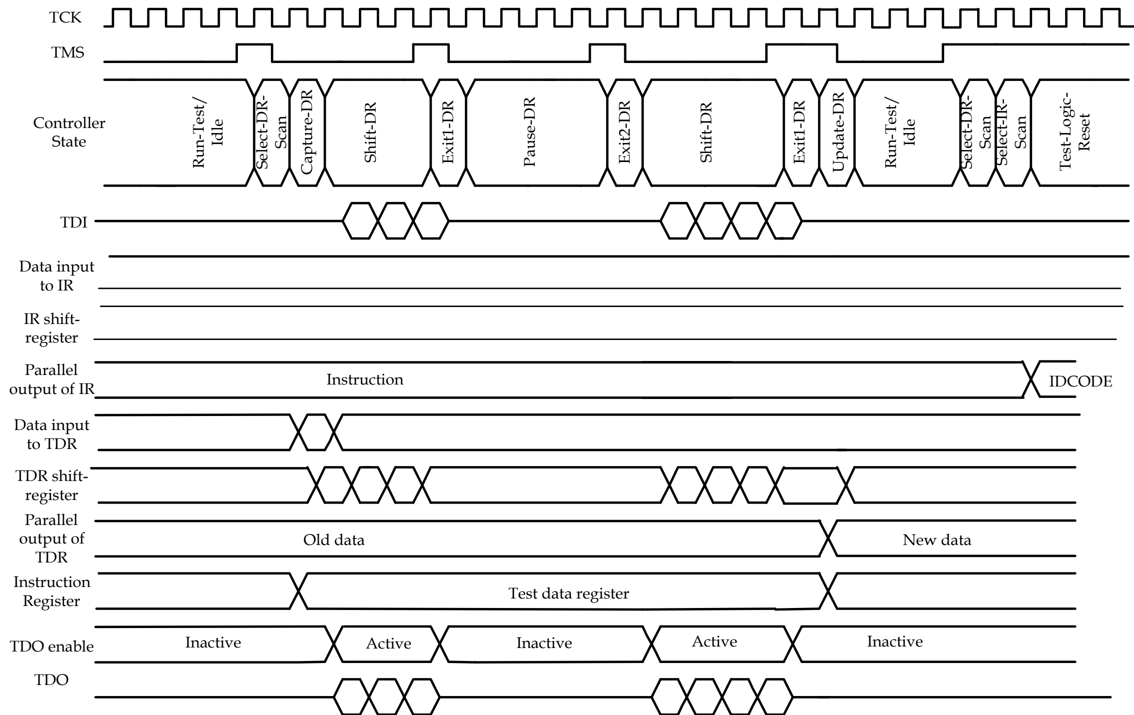

Figure 10.10 TAP Controller: Example of Timing Diagram, Data Scan

When the controller is in the DR branch of the state diagram, it performs on the IR operations similar to those described above. Examples of controller timing diagrams are shown in Figs. 10.9 and 10.10 for instruction and data scan, respectively.

10.6.2 Instruction Set

The controller utilized only a few instructions. Only three of these are mandatory: BYPASS, EXTEST, and SAMPLE/PRELOAD. The most commonly used optional instructions are IDCODE, INTEST, and RUNBIST. The instruction set can also be extended with public and private instructions. Private instructions are those intended to be used only by the manufacturer of the device, and thus do not need to be documented. The instruction is loaded in the TDI and shifted in the IR during the Shift-IR state. It is then decoded and executed.

10.6.2.1 Bypass Instruction. This mandatory instruction permits bypassing of the current IC. It places the one-bit bypass register between TDI and TDO of the chip when another IC is being tested.

10.6.2.2 EXTEST Instruction. This is a mandatory instruction, which is actually the primary reason for Boundary-Scan. It allows testing of the connectivities of the various pins of the ICs mounted on the board. The fault models used are

Copyright © 2000 by John Wiley & Sons

stuck-at, bridging fault, and opens. For present deep-submicron technology circuits, noise faults should also be considered. The latter fault terminology was used in Chapter 2 to express noise failure, such as ground bounce and crosstalk.

10.6.2.3 Sample/Preload Instruction. This instruction is mandatory and is used to scan the BSR without interrupting the normal operation of the internal logic. It supports two functions: sampling the normal operations of the chip (taking snapshots) and preloading data for another test operation into the latched parallel outputs of the BSCs. This instruction is useful in debugging prototypes in the development phase of the board design.

10.6.2.4 IDCODE Instruction. Although this optional instruction does not involve testing of the board, it helps identifying misplaced ICs. Often, it is difficult to distinguish between similar devices, and discovering the reason for malfunction of the board may take unnecessarily a long time.

10.6.2.5 INTEST Instruction. This is another optional instruction. It allows static testing of a particular IC using a bed-of-nails fixture and pin probing of ATE equipment. The test patterns are applied to the input of the chip and the response captured at the outputs. The test data are applied one at a time at the rate of TCK. For chips that need special operational speed such as those containing DRAMs that need to be refreshed, this instruction could be unusable.

10.6.2.6 RUNBIST Instruction. This instruction also is optional. It causes the execution of BIST test provided on the IC selected. It requires minimum data from outside the chip since, in BIST, the test patterns are generated internally on the chip. These patterns are applied dynamically. The instruction is run when the TAP controller is in its Run-Test-Idle state. The result of the test is captured when the controller is in Capture-DR state.

10.6.2.7 CLAMP Instruction. This optional instruction is used to control the output signal of a component to a constant level by means of a BSC. This is useful to hold values on some pins of the circuit, which are not involved in the test. These required signals are then loaded with other test patterns every time they are needed. This instruction, although useful, increases test application time.

10.6.2.8 HIGHZ Instruction. The HIGHZ instruction is optional and forces all outputs of a component to a high-impedance state. It is used, for example, when an in-circuit test is required for testing a non-BS compliant component. In this way the other components are protected from any possible back drive signal by the tester.

Copyright © 2000 by John Wiley & Sons

10.7 Modes of Operations

The TAP provides access to many test support functions. It consists of four input and one output connections. There is an optional input signal, which is used to reset the test logic synchronously. This reset function can also be performed by built-in logic in its controller. The controller receives input from TMS and TCK signals and passes the information to the IR register or the data register selected.

10.7.1 Normal Operation

During normal operation, the Boundary-Scan cell is transparent allowing the input and output signals to pass freely through the test cells, thus enabling the device to perform its intended function. The select inputs to both multiplexers of the BSC, as discussed earlier and shown in Fig. 10.5, are held low. The TAP controller is in the Test-Logic-Reset state and TMS is kept high.

10.7.2 Test Mode Operation

10.7.2.1 External Testing. Testing the interconnect between the ICs is one of the main objectives of Boundary-Scan testing. It is performed by the EXTEST instruction, which is one of the mandatory instructions of the IEEE Standard 1149.1. In addition to the interconnect circuitry, it is possible to test logic on the board: RAM, ROM or glue logic that is not configured in Boundary-Scan design. In this mode the internal logic of the ICs is isolated from its input and output pins.

First, the EXTEST instruction is loaded into the IR register and decoded. Then the testing is performed according to the following steps, which are illustrated in Fig. 10.11. Notice that we use one cell to represent all input BSCs (to the left) and one cell to represent all output BSCs. In addition, the devices that are active in a certain step are highlighted.

1. Shift-DR. In this state the stimulus data are shifted in from the board's TDI through the BSR registers to the cells related with the output pins of the ICs.
2. Update-DR. When all the data are entered, the stimulus is latched at the output cells and is thus applied to the board interconnections.
3. Capture-DR. The data on the interconnect are then captured at the input cells of the input pins of the receiving ICs. This is indicated in Fig. 10.11 as Internal Logic 2.
4. Shift-DR. The results of the test are shifted through the BSR toward TDO for observation.

Copyright © 2000 by John Wiley & Sons

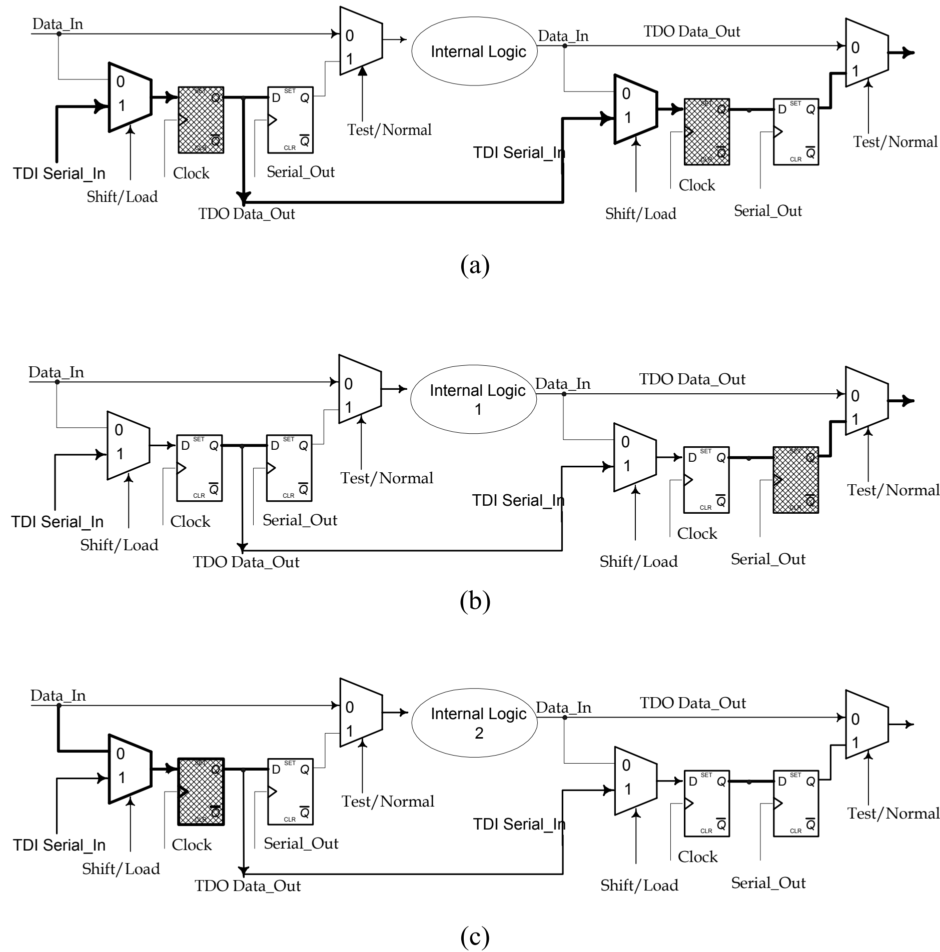

Figure 10.11 External Testing. a) Shift Stimulus in, b) Capture-DR Capture Data at the Outputs of Internal Logic 1, c) Capture-DR: Capture the Data at the Inputs of the Receiving IC, Internal Logic 2.

This procedure enables the testing of faults, such as open, stuck-at, or bridging faults. These faults are due to structural problems of the interconnect between the pins and with the pins themselves.

10.7.2.2. Testing Internal Logic. This mode may be performed by the INTEST instruction. It is one of two optional instructions that permit testing on-chip logic. This is performed in the same fashion as the external test, except that test data are applied to one of the ICs as illustrated in Fig. 10.12. The stimulus is entered in through TDI, applied to the internal logic, and then shifted out in the same way as the EXTEST:

1. Shift-DR. Shift stimulus data in from TDI through the registers of the cells related to the IC input pins.
2. Update-DR. Apply stimulus to the internal logic.
3. Capture-DR. Capture the status of the logic at the output pins of the same IC.
4. Shift-DR. Shift out the results through the BSR toward TDO for observation.

Copyright © 2000 by John Wiley & Sons

For Shift-DR states, the operations are done exactly as in the case of the external testing. Only the operations in the two other states are illustrated in Fig. 10.12. This testing allows the detection of only limited faults since it is not based on any internal DFT technique.

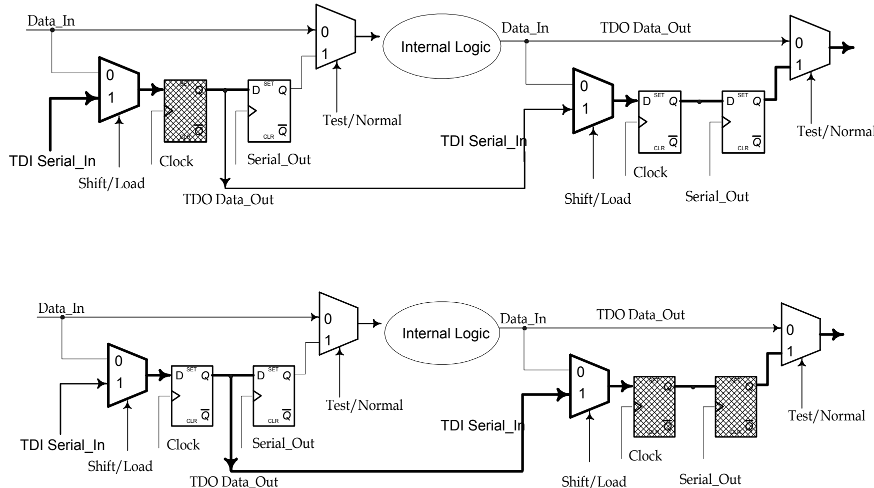

Figure 10.12 Internal Testing Mode a) Capture-DR: The Stimulus Data c) Capture-DR: Capture the Response Data.

10.7.2.3 BIST Execution. This is an optional mode. It is performed by the RUNBIST instruction while the TAP controller is in Run-Test-Idle state. In this testing the data transfer is kept to a minimum since the instruction triggers the on-chip testing. The test response is prevented from appearing at the output until the test application is completed. The results are then captured by the BSR and shifted through TDO for observation.

10.7.3 Testing the Boundary-Scan Registers

All the testing modes have made use of the Boundary-Scan registers to scan-in and scan-out the data. Thus these registers should be tested thoroughly to validate the testing of the ICs and the interconnect. This is very similar to verifying the integrity of the scan-path chain prior to using it for scan-path testing. For details of this integrity testing of the Boundary-Scan register, the reader may consult references such as [Parker 1998].

10.8 Boundary-Scan Languages

From what we have learned so far about Boundary-Scan, clearly, the basic principles are relatively simple compared to managing the data when applying the test patterns. It was thus necessary to adopt use of the standard by creating software tools to support its automation in testing. For this, Boundary-Scan Description Language (BSDL) was developed to facilitate the description of the architecture of a specific board and the flow of the data for testing. According to IEEE

Copyright © 2000 by John Wiley & Sons

Standard 1149.1-1990, it has been intended to be used by chip developers and ATE manufacturers to promote consistency throughout the industry.

This language, which was initiated by Hewlett-Packard in 1990, is based on a subset of VHDL [Parker 1990, 1991]. To facilitate data management, the language needs two attributes. First, it has to be easy to use. Second, it has to be simple to make it unambiguously parsable. In virtue of its raison d'être, the BSDL is not a general-purpose language. On the contrary, it implies knowledge of the devices and the standard; that is, how the device captures, shifts, and updates data. Thus the elements of the design that are absolutely mandatory for compliance with the IEEE Standard are not part of the language. For example, the mandatory registers are not described; they are implied. In addition, the language does not describe the logic on the chip, simply the connectivities of the Boundary-Scan registers to the terminals of this design.

Like any other VHDL language, BSDL consists of an entity and an architecture, which comprise a description of the package and of the package body. Among other parameters, the entity includes the TAP description and the package pin mapping.

10.9 Cost of Boundary-Scan Design

Initial reluctance to the use of Boundary-Scan design was primarily due to the hardware overhead, delays through different multiplexers, and the need for special design tools to incorporate the structure on the chip, board, and the tester. The cost of hardware overhead is really relative to the size of the chips and the board. Consider a small chip such as the 74H322, an octal D-latch SSI [TI 1990], the overhead is really too large to be acceptable. For relatively larger chips, this overhead is dwarfed. However, the cost of adding four pins may be a larger factor to consider.

At present, many commercial and proprietary tools that insert Boundary-Scan structure in a design. Similar to other DFT constructs, adding Boundary-Scan may increase the design time. However, it can actually shorten the manufacturing test engineering and hence the time to market [Bleeker 1993, Parker 1998]. This is because:

- Many vendors provide tools for automatic Boundary-Scan insertion.
- More people have experience with the IEEE Standard 1149.1.
- The silicon cost is declining.
- Boundary-scan technology is more appropriate for concurrent engineering development, which has been shown to shorten the design cycle.
- In addition to helping in manufacturing testing, Boundary-Scan is helpful in prototype debugging.

Copyright © 2000 by John Wiley & Sons

10.10 To Explore Further

It is important to follow the changes in the standards as they evolve. Monitoring the Web pages of the IEEE section on standards is an easy way to be informed. With the increase in mixed-signal circuits, it is also important to learn about the new standard for analog and mixed-signal circuits, IEEE Standard 1149.4 [Parker 1998]. How the board-level ATE testers handle Boundary-Scan design is another important topic. This knowledge is critical so that the designers understand the impact of their DFT constructs on the testers [Lefevre 1990, Tinaztepe 1991].

References

Bleeker, H., P. V. D. Eijnden, and F. de Jong (1993), *Boundary-Scan Test: A Practical Approach*, Kluwer Academic, Norwell, MA.

Feugate, R. J., et al. (1986), *Introduction to VLDI Testing*, Prentice Hall, Upper Saddle River, NJ.

GenRad (1989), *Introduction to multi-strategy testing*, GenRad Inc., Concord, MA.

Hansen, P. (1989), Testing conventional logic and memory clusters using boundary scan devices as virtual ATE channels, *Proc. IEEE International Test Conference*, pp. 166 – 173.

IEEE (1990), *IEEE Standard Test Access Port and Boundary-Scan Architecture*, IEEE Standard 1149.1, IEEE Press, New York.

IEEE,(1994), *IEEE Standard Module Test and Maintenance (MTM) Bus Protocol*, IEEE Standards Board, New York.

JTAG (1988), *JTAG Boundary-Scan Architecture Standard Proposal*, Version 2.0, Technical Sub-Committee of the Joint Test Action Group, Ipswitch, United Kingdom, March.

Lefevre, M. F. (1990), Functional test and diagnosis: A proposed JTAG sample mode scan tester, Paper 16.1 *Proc. International Test Conference*.

Maunder, C. M., et al. (1990), *The Test Access Port and Boundary-Scan Architecture*, Tutorial, *IEEE* Computer Society Press, Los Alamitos, CA.

Parker, K. P. (1989), The impact of Boundary-Scan on board test, *IEEE Des. Test Comput*., Vol. 6, No. 4, pp. 18 – 30.

Parker, K. P., and S. Oresjo (1990), A language for describing Boundary-Scan devices, *Proc. IEEE International Test Conference*, pp. 222-234.

Parker K. P., and S. Oresjo (1991), A language for describing Boundary-Scan devices, *J. Electron. Testing: Theory Appl.,* Vol. 2, No. 1, pp. 43 – 74.

Parker, K. P. (1998), The Boundary-Scan Handbook: Analog and Digital, Kluwer Academic, Norwell, MA.

SIA (1997), *The National Technology Roadmap for Semiconductor*, Semiconductor Industry Association.

TI (1990), *SCOPE Testability Products: Application Guide*, Texas Instruments Inc., Dallas, TX.

Tinaztepe, C. (1991), Device oriented test program generation for the sequence-per-pin test system architecture, *Proc. European Test Conference*, Munich.

Copyright © 2000 by John Wiley & Sons

Problems

10.1. Develop a synchronizing sequence for the TAP controller FSM, if any.

10.2. The IDCODE instruction is optional. Would you recommend using it? Why or Why not?

10.3. You are to implement the execution of RUNBIST on a chip. Describe all the necessary signals and their sequence to execute the instruction. Indicate any extra circuitry needed to accomplish this task.

10.4. In a PCB, you decided to use a chip that follows IEEE Standard 1149.1. All the other ICs do not use include boundary-scan architecture. Would you implement this architecture on the board level? Explain why or why not. If the answer is affirmative, would it be necessary to use the bypass instruction?

Copyright © 2000 by John Wiley & Sons

Copyright © 2000 by John Wiley & Sons