

Boundary-Scan Test

Boundary-Scan Test

A Practical Approach

by

Harry Bleeker

Peter van den Eijnden

*Fluke/Philips Test & Measurement,
Eindhoven, The Netherlands*

and

Frans de Jong

*Philips Research Laboratories,
Eindhoven, The Netherlands*



SPRINGER SCIENCE+BUSINESS MEDIA, B.V.

ISBN 978-1-4613-6371-2 ISBN 978-1-4615-3132-6 (eBook)
DOI 10.1007/978-1-4615-3132-6

This printing is a digital duplication of the original edition.

Printed on acid-free paper

All Rights Reserved
© 1993 Springer Science+Business Media Dordrecht
Originally published by Kluwer Academic Publishers in 1993
Softcover reprint of the hardcover 1st edition 1993
No part of the material protected by this copyright notice may be reproduced or
utilized in any form or by any means, electronic or mechanical,
including photocopying, recording or by any information storage and
retrieval system, without written permission from the copyright owner.

CONTENTS

LIST OF FIGURES ix

PREFACE xv

Chapter 1

PCB TESTING 1

MINIATURIZATION IN ELECTRONICS 1

ROAD BLOCKS FOR CONVENTIONAL PCB TEST METHODS 4

THE SOLUTION FOR PCB TESTING 5

Test Control Logic 10

Example Boundary-Scan Circuits 13

Constituent Elements of the BST Path 16

Chapter 2

THE BOUNDARY-SCAN TEST STANDARD 19

THE BST ARCHITECTURE 19

TEST ACCESS PORT 20

TAP CONTROLLER 22

State Descriptions 26

THE INSTRUCTION REGISTER 30

TEST DATA REGISTERS 32

The Bypass Register 34

The Boundary-Scan Register 35

A Boundary-Scan Cell 36

System Pin Related Cells 37

The Device ID Register	41
Design-Specific Registers	43
INSTRUCTIONS	43
Public and Private Instructions	43
The BYPASS Instruction	44
The SAMPLE/PRELOAD Instruction	44
The EXTEST Instruction	45
The INTEST Instruction	47
The RUNBIST Instruction	48
The CLAMP Instruction	48
The IDCODE Instruction	49
The USERCODE Instruction	49
The HIGHZ Instruction	49
DOCUMENTATION REQUIREMENTS	49

Chapter 3

HARDWARE TEST INNOVATIONS	51
PROVISIONS AT BOARD LEVEL	51
Concurrent Sampling	51
The Boundary-Scan Master	54
Memory Board Testing	57
SYSTEM-LEVEL TEST SUPPORT	60
Embedded Go/No-Go Test With BST	60
System Backplane Test Bus	61
ON CHIP PROVISIONS	67
BST on WSI designs	68
BST on Multichip Modules	69
BST on MOS Designs	71
Digital Bus Monitor	74
Adjustable Scan Path Lengths	78
Path-Delay Measurements	80

Chapter 4

BST DESIGN LANGUAGES	85
BSDL DESCRIPTION	85
The Entity Section	86
The Package Description	94
The Package Body	95
Remnant BSDL Functions	96
BSDL DESIGN EXAMPLE	98
Design Methodology	98

The Circuitry	99
A BOUNDARY-SCAN REGISTER COMPILER	103
The Technology Library	104
Compiler Input and Output Files	106
Results	108
TEST SPECIFICATION LANGUAGES	109
Test Interface Layer	109
General Format Test Vectors	114
The Serial Vector Format	120

Chapter 5

PCB TEST STRATEGY BACKGROUNDS	123
TESTING THE INTEGRITY OF THE BST CHAIN	123
The Tested Functions	123
The Test Steps	127
PCB PRODUCTION FAULTS	132
Interconnect Structures	132
Considering Opens	134
Considering Shorts	136
Considering Multiple Faults	139
Coming to a Model	140
TEST ALGORITHMS	140
The Binary Counting Test Sequence	141
The Minimal-Weight Sequence	144
The Walking One Sequence	144
The Diagonally Independent Sequence	145
The Maximal Independent Set	146
The Order Independent Test Sequence	147
DIAGNOSTICS	148
2-Net Shorts	148
Multiple-Net Shorts	149
Aliasing Test Results	151
Confounding Test Results	151
Single and Multiple Step Tests	152
Adaptive Test Algorithm	152
Applying the Max-Independence Algorithm	153
Combined Faults in One Net	155
CLUSTER TESTING	157
Impact on Fault Detection	159
Interconnect Test of Memory Clusters	161
ARCHITECTURE OF A BOUNDARY-SCAN TEST FLOW	166
Serializer Considerations	169

Chapter 6

MANAGEMENT ASPECTS	173
COMING TO THE STANDARD	173
Effects of Boundary-Scan Testing	176
MANAGEMENT ROLE	177
Introduce Concurrent Engineering	178
Prescribe Design For Testability	182
Check Impact on Reliability	183
Care for Quality	185
BENEFITS OF BOUNDARY-SCAN TEST	188
Shorter Times To Market	188
Lower Capital Investments	191
Cost Considerations for ICs	191
PCB Cost Considerations	192
SUMMARY	194
 APPENDIX	 197
 GLOSSARY	 215
 REFERENCES	 219
 INDEX	 223

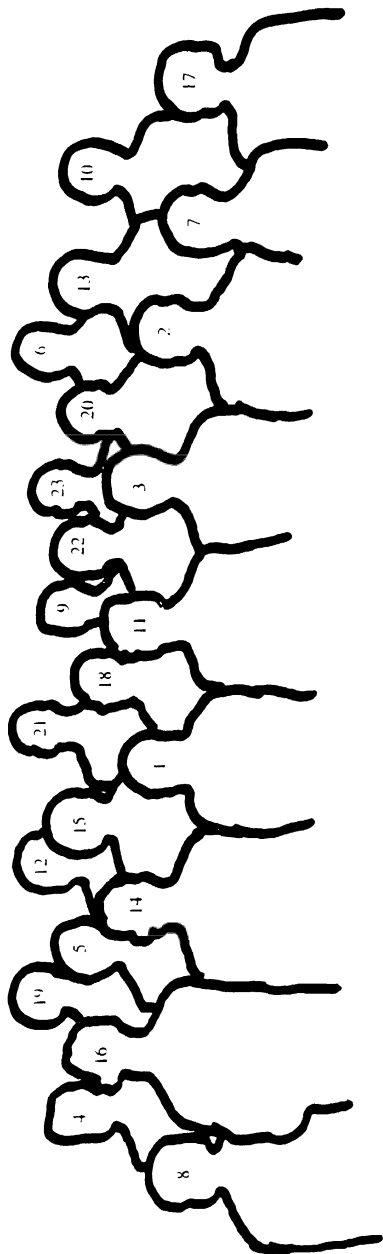
LIST OF FIGURES

Fig. 1-1	A printed circuit board containing ASICs	1
Fig. 1-2	Reduction of trace distances on PCBs and in ICs	2
Fig. 1-3	Electron microscope picture of part of a PCB	3
Fig. 1-4	Electron microscope picture of a soldered IC on a PCB	3
Fig. 1-5	Conventional test pins for a bed-of-nails fixture	4
Fig. 1-6	Many interconnections between three ICs	6
Fig. 1-7	Added shift registers provide scan path	6
Fig. 1-8	Action 1: shift test data to output cell	7
Fig. 1-9	Actions 2+3: apply test data and capture result	7
Fig. 1-10	Action 4: shift captured results out	7
Fig. 1-11	Components of a Boundary-Scan Cell in an IC	8
Fig. 1-12	ShiftDR state	8
Fig. 1-13	Update-DR state for EXTEST	9
Fig. 1-14	Capture-DR state for EXTEST	9
Fig. 1-15	Update-DR state for INTEST	11
Fig. 1-16	Capture-DR state for INTEST	11
Fig. 1-17	BST infrastructure on PCB	13
Fig. 1-18	Test control logic for BST	12
Fig. 1-19	IC including Boundary-Scan Register and Test Control	13
Fig. 1-20	PCB configuration with all IC BSRs connected in series	14
Fig. 1-21	Example of a commonly used Boundary-Scan cell	15
Fig. 1-22	An example of an Instruction Register cell	16
Fig. 1-23	BST path between two ICs	17
Fig. 2-1	The standard Boundary-Scan architecture	20
Fig. 2-2	BST configuration with two parallel paths	22
Fig. 2-3	State diagram of the TAP Controller	24
Fig. 2-4	Example timing diagram of TAP Controller	26
Fig. 2-5	Signal flow (bold lines) during Test-Logic-Reset	27
Fig. 2-6	Signal flow (bold lines) in BS cell during Capture-DR	28

Fig. 2-7	Signal flow (bold lines) in BS cell during Shift-DR	28
Fig. 2-8	Signal flow (bold lines) in BS cell during Update-DR	29
Fig. 2-9	Signal flow (bold lines) in IR cell during Capture-IR	29
Fig. 2-10	Signal flow (bold lines) in IR cell during Shift-IR	30
Fig. 2-11	Signal flow (bold lines) in IR cell during Update-IR	30
Fig. 2-12	ICs set for various tests	31
Fig. 2-13	Set-up of an Instruction Register	31
Fig. 2-14	Overview of the Test Data Registers	33
Fig. 2-15	Example of a combined test data register	34
Fig. 2-16	Implementation of a Bypass Register	35
Fig. 2-17	An implementation of a Boundary-Scan Cell	37
Fig. 2-18	Minimum BSC configuration for input pins	37
Fig. 2-19	Boundary-Scan cells with various output buffers	38
Fig. 2-20	Cell configuration at a 3-state output pin	39
Fig. 2-21	One BS cell controls several 3-state outputs	40
Fig. 2-22	Cell configuration at a bidirectional system pin	40
Fig. 2-23	Structure of a Device Identification Register	42
Fig. 2-24	An example of a Device ID design.	42
Fig. 2-25	Data flow during SAMPLE instruction	45
Fig. 2-26	Data flow during PRELOAD instruction	46
Fig. 2-27	Data flow during EXTEST instruction	46
Fig. 2-28	Data flow during INTEST instruction	47
Fig. 3-1	Board structure for test with concurrent sampling	53
Fig. 3-2	Block diagram of the Boundary-Scan Master	55
Fig. 3-3	Boundary-Scan interface of the MCERT chip	59
Fig. 3-4	Power-On-Self-Test for a BST board	61
Fig. 3-5	Architecture for system testing	63
Fig. 3-6	Blockdiagram of the BT-Link unit	64
Fig. 3-7	Timing of BT-Link control logic	65
Fig. 3-8	Chip arrangement for full wafer probe testing	68
Fig. 3-9	Chip arrangement for Boundary-Scan testing	69
Fig. 3-10	Integrated BS cell for Motorola 68040 address pins	73
Fig. 3-11	TI's DBM architecture	75
Fig. 3-12	Partitioning a scan test data register	79
Fig. 3-13	Scannable Double-Strobe flip-flop	83
Fig. 4-1	An implementation of a Boundary-Scan Cell	92
Fig. 4-2	Symbolic model of a boundary cell	95
Fig. 4-3	Core of octal D-latch	101
Fig. 4-4	Octal D-latch with Boundary-Scan chain added	102
Fig. 4-5	Structure of BSR compiler	106
Fig. 4-6	Example circuit	117
Fig. 5-1	Some PCB net structures	134

Fig. 5-2	Sample net with four nodes	135
Fig. 5-3	Some single and multiple stuck-open faults	136
Fig. 5-4	All possible shorts between four simple nets	138
Fig. 5-5	A set of simple nets	143
Fig. 5-6	Multiple faults in one net	155
Fig. 5-7	Circuit containing BST and Non-BST logic	157
Fig. 5-8	Timing of RAM accesses	162
Fig. 5-9	Block diagram of a Boundary-Scan tester	167
Fig. 5-10	Screen display showing identification faults	169
Fig. 5-11	Screen display showing net bridging faults	170
Fig. 6-1	Relation between complexity and speed of electronic systems	173
Fig. 6-2	Relation between commitment and expenditure	178
Fig. 6-3	Costs for (A) sequential and (B) concurrent engineering	180
Fig. 6-4	PCB design cycle with and without BST	181
Fig. 6-5	Difference in profit due to design delay	190
Fig. A-1	Measuring Set-up and Hold times	206
Fig. A-2	Measuring propagation delays	206





1	Rod Tulloss (AT&T, U.S.A.)	13	Ulrich Ludemann (Nixdorf Computer, Germany)
2	Ben Bennetts (Bennetts Associates, U.K.)	14	Harry Bleeker (Philips (chairman), The Netherlands)
3	Colin Maunder (British Telecom, U.K.)	15	Dirk van de Lagemaat (Philips (secretary), The Netherlands)
4	Ian Fisher (Computer Automation, U.K.)	16	Frans Beenker (Philips, The Netherlands)
5	David Richards (Digital Equipment Corp., U.S.A.)	17	Wim Sauerwald (Philips, The Netherlands)
6	Eskild Jensen (ElektronikCentralen, Denmark)	18	Pat Diamond (Plessey (secretary), U.K.)
7	Walter Ghisler (Ericsson, Sweden)	19	Erwin Trischler (Siemens, Germany)
8	Prabhat Varma (GEC HRC, U.K.)	20	Derek Roskell (Texas Instruments, U.K.)
9	Frans Mosselvelde (Facron-Schlumberger, The Netherlands)	21	Pete Fleming (Texas Instruments, U.S.A.)
10	Malcolm Wallace (GenRad, U.K.)	22	Lee Whetsel (Texas Instruments, U.S.A.)
11	Tom Williams (IBM Boulder, U.S.A.)	23	Michel Parot (Thomson-CSF, France)
12	Gordon Hannah (Marconi Instruments, U.K.)		

PREFACE

The ever-increasing miniaturization of digital electronic components is hampering the conventional testing of Printed Circuit Boards (PCBs) by means of bed-of-nails fixtures. Basically this is caused by the very high scale of integration of ICs, through which packages with hundreds of pins at very small pitches of down to a fraction of a millimetre, have become available. As a consequence the trace distances between the copper tracks on a printed circuit board came down to the same value. Not only the required small physical dimensions of the test nails have made conventional testing unfeasible, but also the complexity to provide test signals for the many hundreds of test nails has grown out of limits.

Therefore a new board test methodology had to be invented. Following the evolution in the IC test technology, Boundary-Scan testing has become the new approach to PCB testing. By taking precautions in the design of the IC (design for testability), testing on PCB level can be simplified to a great extent. This condition has been essential for the success of the introduction of Boundary-Scan Test (BST) at board level.

Nowadays Boundary-Scan testing is embraced worldwide by almost all PCB manufacturers. The main reason is, of course, cost saving, emerging in various phases of the PCB life cycle. In the *design* phase, testing time is saved during prototyping. In the *factory*, lower test preparation time, time savings in fault diagnosis and much cheaper test equipment has led to a reduction in test costs of over 50%. Moreover, the factory throughput has increased. In the *field service* phase, tremendous savings are made: in the prices of test equipment, in test preparation times and reduced spare board inventories. In this way BST has led to a cost reduction in PCB production of as much as 70%, *after* including the extra development costs for the IC precautions.

The simplicity of BST also makes such cost reductions profitable for small volume PCB production. And since Boundary-Scan Testing involves the whole PCB life cycle, it is more than just the introduction of a new design technology, it is an integral production approach. Therefore BST concerns the global strategy of a company to which also top management should pay attention.

Due to its importance, the architecture of the BST technology has lead to a world standard: IEEE Std 1149.1 [1]. Applying this standard in the IC designs allows PCB designers to use ICs of various manufacturers and yet prepare *standard* test methods and test equipment for the whole life cycle of the product.

This book is an aid to introducing BST into a company. Chapter 1 states the PCB test problem and its solution. Chapter 2 provides a tutorial on the standard for the Boundary-Scan Architecture. The next chapters 3 and 4 describe the innovations for the hardware and the software BST support respectively. Developments in BST are explored with the aid of various examples and results as they have been obtained and published by world leading electronics companies. Chapter 5 describes various basics supporting the actual test technology; which PCB production faults are met, how are they detected and to what degree can they be diagnosed. Chapter 6 gives a management overview of the aspects to be considered in introducing BST in a production organisation, including organizational and cost saving aspects. The book ends with a glossary of terms and an alphabetical index, for quick reference.

The readers of this book are expected to have a basic understanding of ICs and digital circuitry.

Acknowledgements

Many contributions are necessary to complete a book at such a technological level as this. The authors wish to thank their colleagues for their discussions and positive comments in preparing the text. In particular Harm Tiersma should be mentioned here for his remarks concerning the principles of testing and his conscientious reading and correcting of the draft. Special credit must be given to Rien van Erk for his editing and writing efforts, without which this book would probably not have come to fruition.

Chapter 1

PCB TESTING

MINIATURIZATION IN ELECTRONICS

Printed Circuit Boards (PCBs) add the most value to electronics hardware. Over the years, PCBs have become loaded with more components and hence have become increasingly complex and expensive. This is mainly caused by the ongoing miniaturization in electronics.

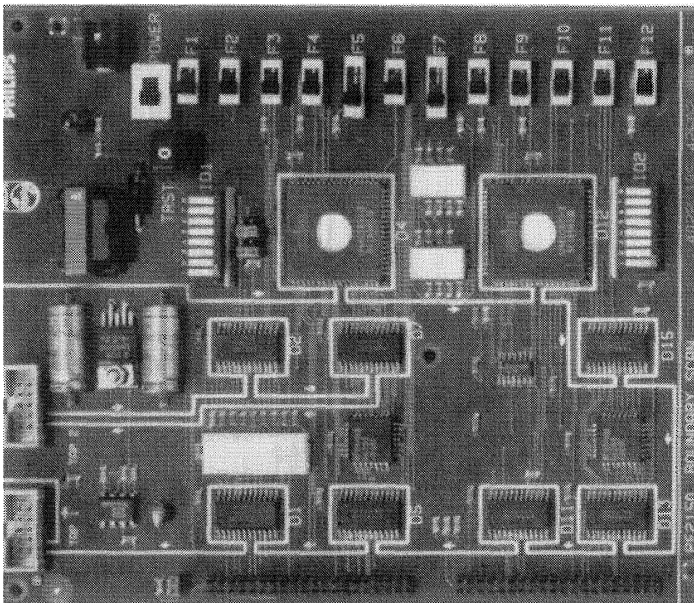


Fig. 1-1 A printed circuit board containing ASICs

To start with, the miniaturization within integrated circuits (ICs) themselves has progressed into the sub-micron technology. This has led to an increased number of gates and a very large number of functions per chip. Consequently, many more pins per IC are needed and hence ICs come in bigger packages with more leads (>500 pins) at smaller pitches (0.3 mm). Therefore, the distance between the foot prints of the ICs on the PCB is becoming equally small.

Direct mounting of chips on both sides of the PCB, using Tape Automated Bonding (TAB), Chip On Board (COB) etc. have caused further miniaturization of the hardware implementation.

To even further increase the number of electronic functions per unit of surface, other technologies are becoming available: Multi Chip Modules (MCMs). These are functional modules built up of dies directly mounted on top of several dielectric and metallization layers supported by a ceramic multi-layer substrate. The component density is very high and accessibility other than via the input and output pins of the MCM is non-existent.

Figure 1-2 shows the reduction of trace distances on PCBs and in ICs.

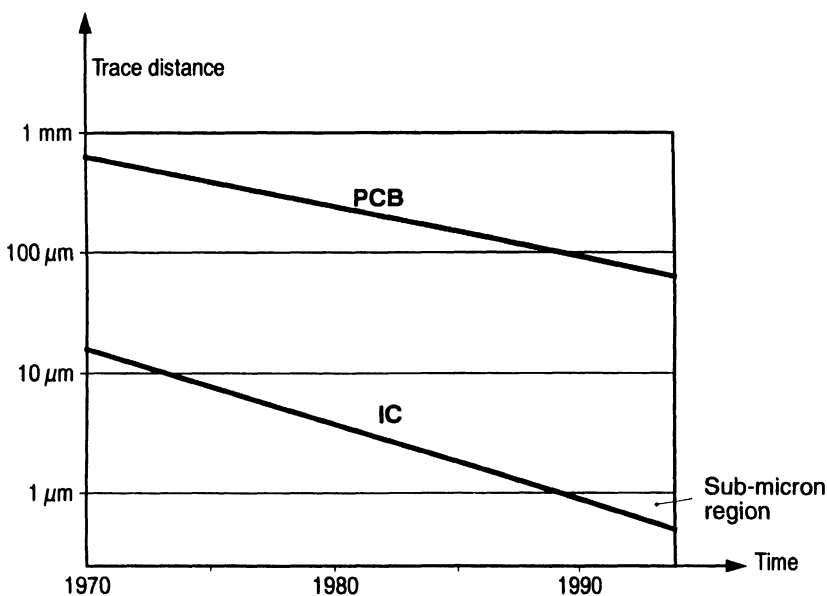


Fig. 1-2 Reduction of trace distances on PCBs and in ICs

Figures 1-3 and 1-4 show, respectively, a part of a PCB and a part of an IC which is soldered onto that PCB. These pictures are taken with a Scanning Electron Microscope and the pitch of the IC's contact pins is 25 mil ($= \frac{1}{40}$ inch ≈ 0.63 mm).

The ongoing miniaturization has made it more and more difficult to access these "highly loaded" PCBs mechanically, with fixtures, for in-circuit testing. Moreover, the test equipment became so expensive in the 1980s that it severely affects the profitability of producing PCBs. Electronic companies were, therefore, looking for low-cost test methods based on the sofar hardly explored design for testability.

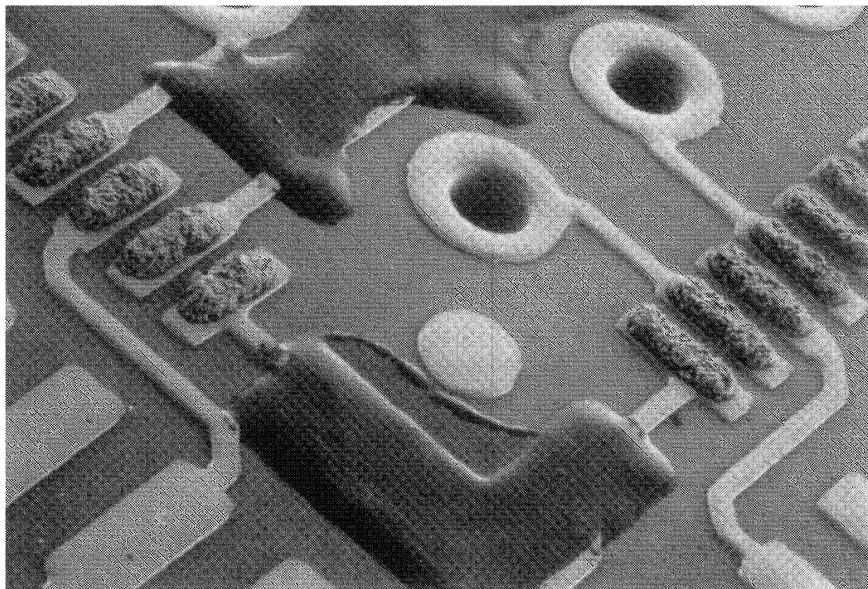


Fig. 1-3 Electron microscope picture of part of a PCB

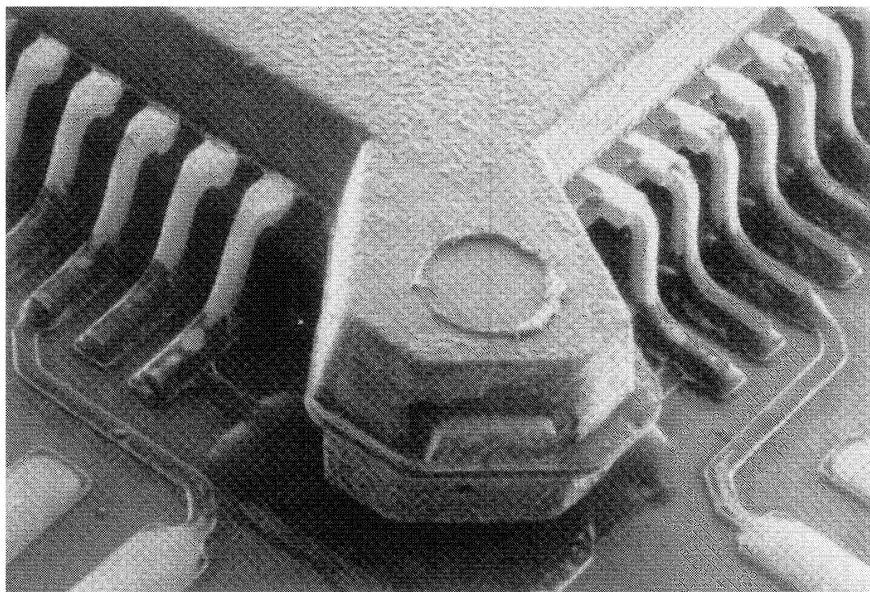


Fig. 1-4 Electron microscope picture of a soldered IC on a PCB

Figure 1-5 shows a drawing of some different test pins used in a bed-of-nails fixture. The pitch of the test pins is $\frac{1}{4}$ " and they have a diameter of about 0.8 mm.

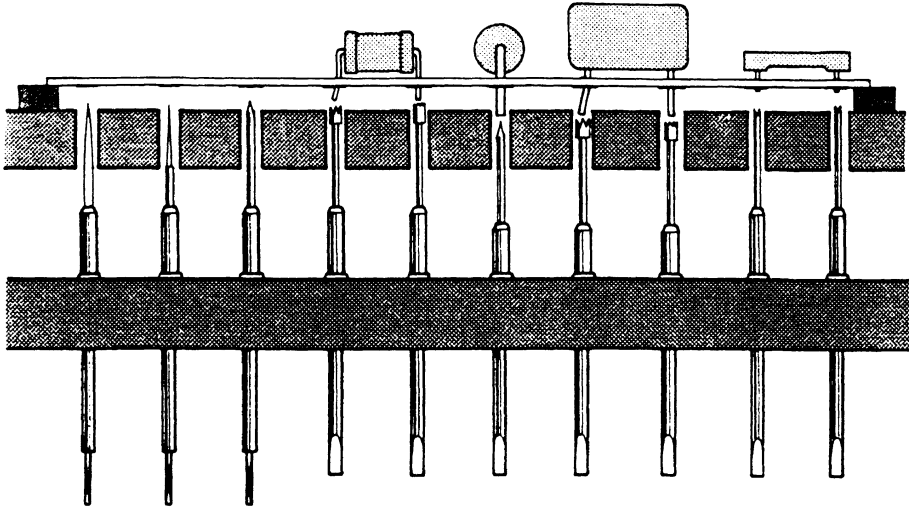


Fig. 1-5 Conventional test pins for a bed-of-nails fixture

It is clear that these pins can not be used for testing purposes at the IC connections in figure 1-4, where the pitch was 0.63 mm.

ROAD BLOCKS FOR CONVENTIONAL PCB TEST METHODS

With *in-circuit testing*, physical contact is made to the components (for example ICs) and the copper tracks on the PCB (test pads) by means of probes or bed-of-nails technology. Input signals are applied to the component's input pins via probes or needles, and the response at the output pins is examined through the probes or needles. For the plated through hole (PTH) technology this method was easy to perform. But the continual miniaturization created the first obstacle to in-circuit testing.

The PCB technology with surface mounted devices (SMDs) *on both sides* of the PCB is another reason why mechanical probing becomes practically impossible.

Next, the advantages of in-circuit testing (ICT), the use of standard test pattern sets for ICs, from which a test program could easily be assembled, have evaporated due to the increasing use of Application Specific ICs (ASICs). Each ASIC requires a separate test set which is not contained in the library of standard test sets. The vendor of the in-circuit test library is unable to supply these tests. Test sets for complex standard VLSI devices (e.g. microprocessors) are not available or very expensive.

Finally, VLSIs require long test sequences, for instance, for initialization. If outputs of ICs are forced (back-driven) during such a sequence, there may be an impact on the overall reliability of the PCBs.

Summarizing, the programming of automatic test equipment (ATE), the mechanical test access and the quality of testing in the conventional way are reaching their limits of feasibility.

The solution to these problems arises from the IC test technology. To overcome the test preparation and fault coverage problems faced in the IC-test technology in the 70's, the scan test technology evolved and became a good solution for digital electronics. The enormous increase in observability and controllability achieved by scan technology forms the base for Boundary-Scan Test technology for PCBs.

THE SOLUTION FOR PCB TESTING

The main point to ensure while manufacturing a PCB is to make sure that all components are mounted properly at the right place on the board and that the interconnections between the components are as prescribed in the design. So, after production, one of the main items to be tested is whether or not the connections between the components (for example ICs) are 100% correct.

The basic idea to circumvent the above mentioned access problem for test purposes was to add a shift register cell next to each input/output (I/O) pin of the component. During test mode these cells are used to control the status of an output pin (high or low) and read the states of an input pin (high or low). This allows for testing the the board interconnections. During normal mode the cells are 'transparent'.

Figures 1-6 and 1-7 show the idea of adding the shift register cells. Figure 1-6 depicts three ICs with 'many' interconnections. The input signals (I) and the output signals (O) are the I/O signals present at the board edge connector.

In figure 1-7 shift register cells have been added between the IC's 'core logic' and the I/O pins to provide a serial test data path through all ICs.

Since the shift register cells are located at the IC's boundary (I/O pins) these cells are referred to as Boundary-Scan Cells (BSCs) and the composed shift register is called the Boundary-Scan Register (BSR). According to the IEEE Std 1149.1 [1], the input to the serial test data path is called Test Data Input (TDI) and the output is called Test Data Output (TDO).

To permit testing of the PCB's connecting copper tracks, the interconnections, the cells must support the following test actions.

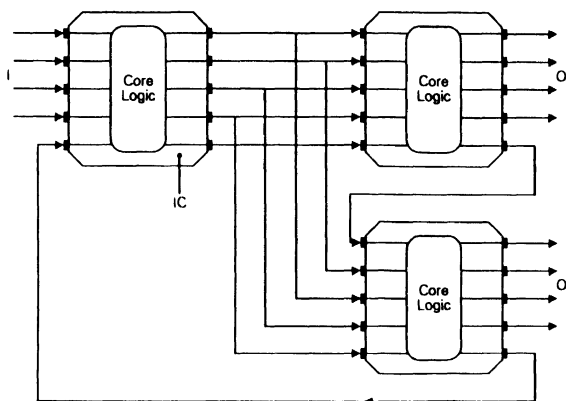


Fig. 1-6 Many interconnections between three ICs

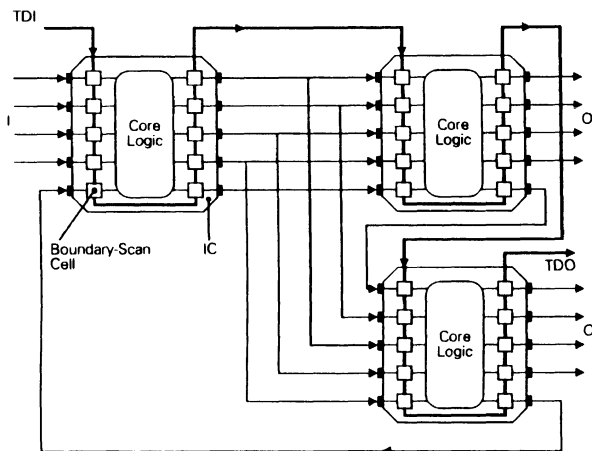


Fig. 1-7 Added shift registers provide scan path

- Action 1. *Shift-DR*: shift stimulus data in from TDI through the registers to the cells related with the output pins of the IC(s).
- Action 2. *Update-DR*: update these output cells and apply the stimuli to the board interconnections.
- Action 3. *Capture-DR*: capture the status of the board's interconnections at the input pins of the receiving IC(s).
- Action 4. *Shift-DR*: shift out the results through the BSR towards TDO for examination.

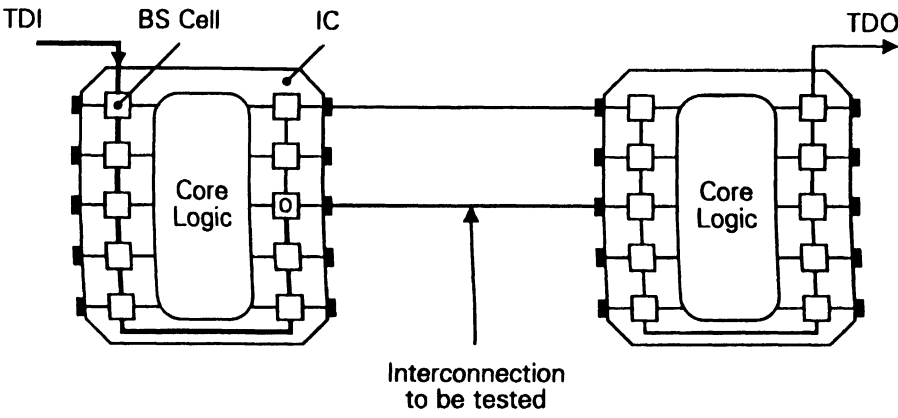


Fig. 1-8 Action 1: shift test data to output cell

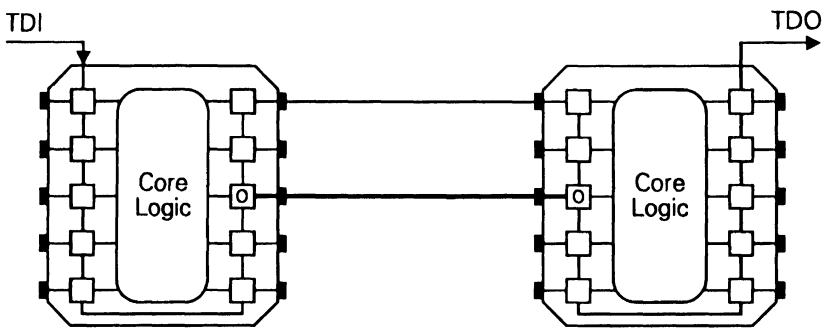


Fig. 1-9 Actions 2+3: apply test data and capture result

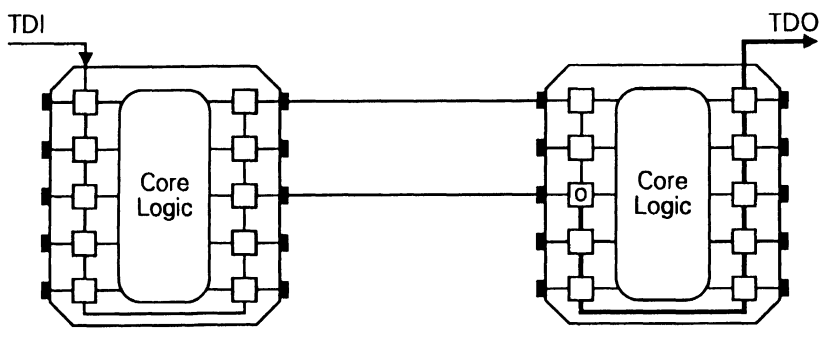


Fig. 1-10 Action 4: shift captured results out

Figures 1-8, 1-9 and 1-10 depict the respective test actions. The test stimulus is indicated here as a bit of value '0' in the Boundary-Scan path.

Note that the actions 2 and 3 perform the actual test: apply test data to the PCB tracks between the ICs and capture the status of these tracks. The two shift actions (1 and 4) do not in fact contribute to the test results.

Testing of the interconnection between the ICs is referred to as external test or *EXTTEST*. The circuit diagram in figure 1-11 supports these functions.

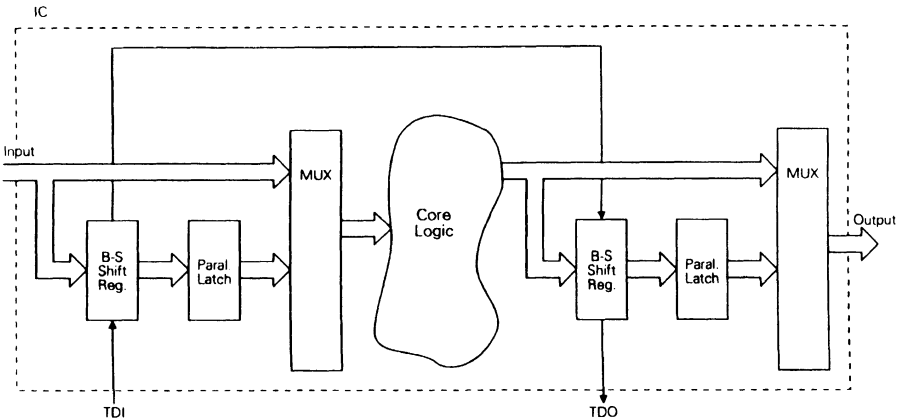


Fig. 1-11 Components of a Boundary-Scan Cell in an IC

Figure 1-12 shows the data flow during the first test action, shift the stimuli (data) through the registers, from input TDI to the output TDO. This action is called Shift-DR. The emboldened lines between TDI and TDO plus the shaded Boundary-Scan (BS) registers depict the signal path.

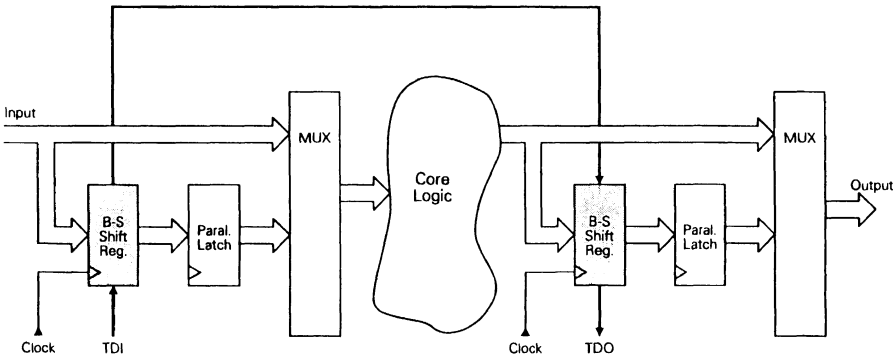


Fig. 1-12 ShiftDR state

Notice that in figure 1-12 the Boundary-Scan design comprises, a parallel latch (flip-flop) and a multiplexer (MUX) in addition to the shift-register stages. The latches have been included to assure a stable signal at the parallel outputs while data (test stimuli and test results) are shifted through the scan path (BSR). The purpose of the multiplexer is discussed later.

The figures 1-13 and 1-14 show the signal flows for Update-DR (above Action 2) and Capture-DR (above Action 3) respectively.

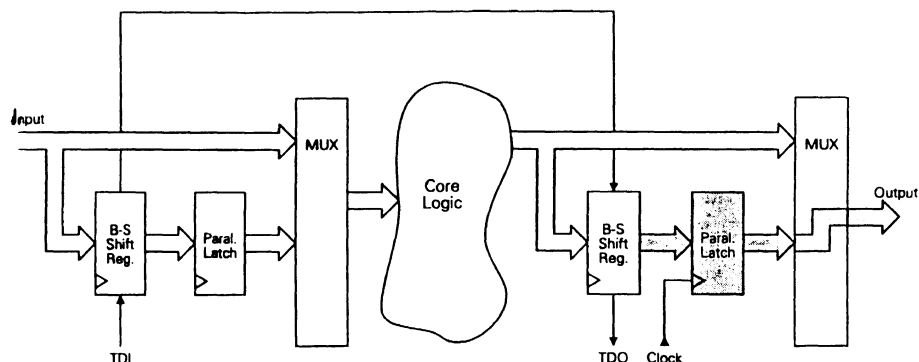


Fig. 1-13 Update-DR state for EXTEST

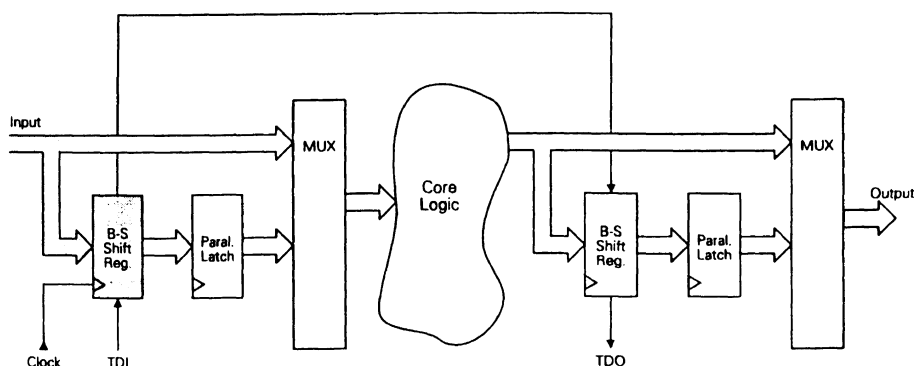


Fig. 1-14 Capture-DR state for EXTEST

Remember that these test steps still concern the EXTEST. The shift state for Action 4 is the same as for Step 1 of course.

The actions indicated in these figures are triggered by the indicated Clock signals. The clock signal triggers the actions in the Boundary-Scan cells. In the Update-DR stage of the EXTEST an electrical signal is then applied to a copper track on the

PCB, which arrives virtually at the same time at the receiving input of the appropriate IC(s), see figure 1-9. Here the Capture-DR action is also triggered by the clock pulse. It is prescribed that the Update-DR action must be triggered at the falling edge (negative slope) of the clock pulse whilst the Capture-DR action is triggered at the rising edge (positive slope) of the clock pulse. Note in the last three figures (1-12 through 1-14) that the EXTEST does not in any way interfere with the functioning of the core logic.

In *conclusion* it can be stated that a board interconnect test can be performed with the aid of Boundary-Scan Registers using the described procedure. It has also been shown that this can be done without interfering with the core logic's operation. Given the registers at the I/O pins of the ICs, it can be seen that an internal test of the IC logic (core logic) can also easily be realized. The testing of the internal logic of an IC is referred to as internal test or *INTEST*. For the *INTEST* the same test actions as described for the *EXTEST* are applicable, only the Update-DR and Capture-DR actions concern now the Boundary-Scan cells at the input and output pins of the IC respectively. The following test sequence describes the actions.

- Action 1. *Shift-DR*: shift the data (test stimuli) through the registers to the cells associated with the input pins of the IC(s).
- Action 2. *Update-DR*: update these input cells and apply the stimuli to the core logic.
- Action 3. *Capture-DR*: capture the status of the core logic outputs in the output cells of the IC(s).
- Action 4. *Shift-DR*: shift out the results through the BSR of the IC(s) for examination.

The signal flow during the shift actions is the same as in figure 1-12. The signal flows for the test actions 2 and 3 are shown in figures 1-15 and 1-16 (shaded).

Test Control Logic

As discussed above, several test control signals are used to control the Boundary-Scan shift-registers, the parallel latches and the multiplexers. It would be unacceptable to define an IC package pin for each control signal. Therefore, a control logic has been designed consisting of a sequencer that can be controlled by a serial protocol. In addition to this sequencer an Instruction Register (IR) is required to select the different tests to be performed, for example *INTEST* and *EXTEST*. The combinations of the sequencer states and the IR instructions are used to generate the internal control signals for the various test operations.

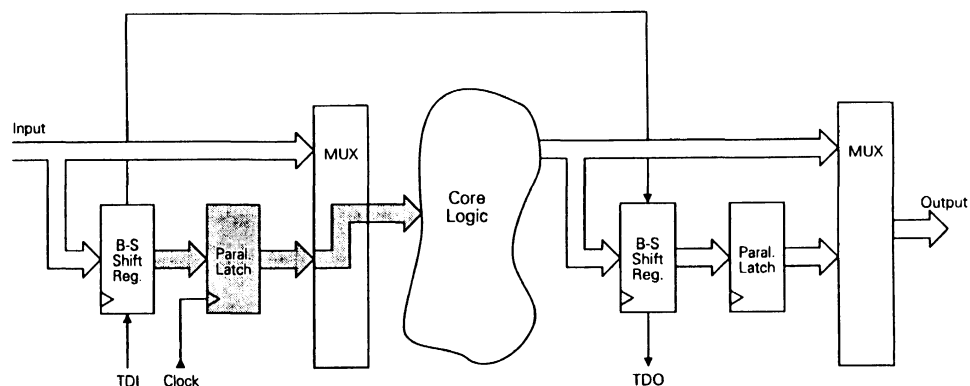


Fig. 1-15 Update-DR state for INTEST

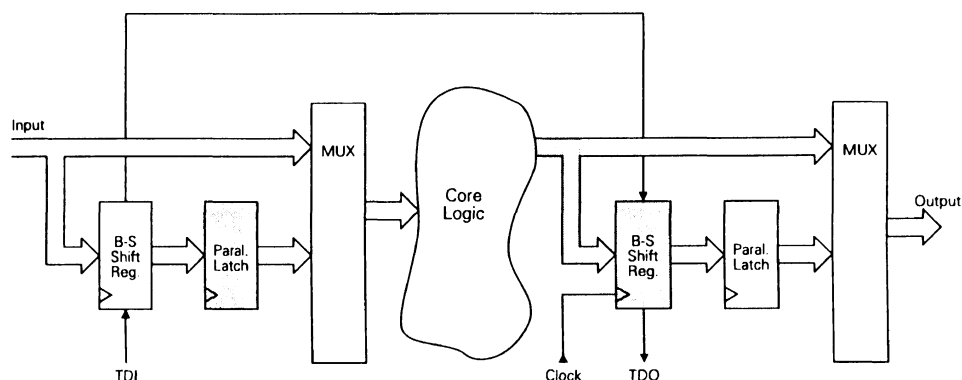


Fig. 1-16 Capture-DR state for INTEST

The sequencer is controlled by two signal lines: Test Mode Select (TMS) and Test Clock (TCK). The whole Boundary-Scan test logic requires at least four dedicated test pins at the IC package, thus additional to its other functional pins. The four pins (TMS, TCK, TDI and TDO) are referred to as the Test Access Port or TAP. The entire test control logic is consequently called the TAP Controller.

Remark. Since all these test provisions have been standardized in the IEEE Std 1149.1 [1], the 'standard' names for the signals and the IC pins are used further in this book. An overview of the birth of the Boundary-Scan standard is given in chapter 6.

At board level, refer back to figure 1-7, the test control lines TMS and TCK have to be applied to all components on the board that are suited for BST. Figure 1-17 includes the infrastructure for the TMS and TCK signals.

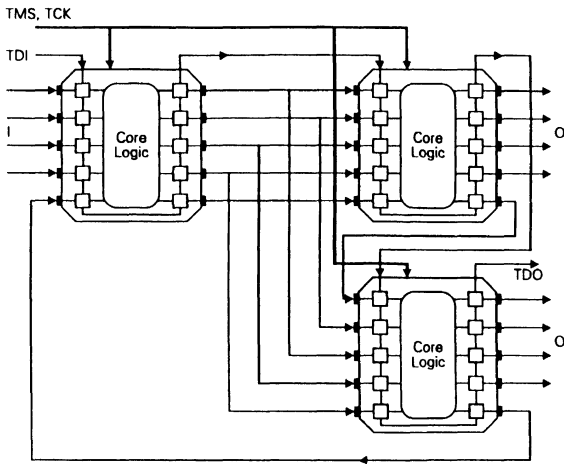


Fig. 1-17 BST infrastructure on PCB

Figure 1-17 shows that the TMS and TCK signals are applied to all relevant components in parallel (thus all TAP Controllers are always in the same state), whereas the test path, TDI to TDO, is a serial path all over the printed circuit board.

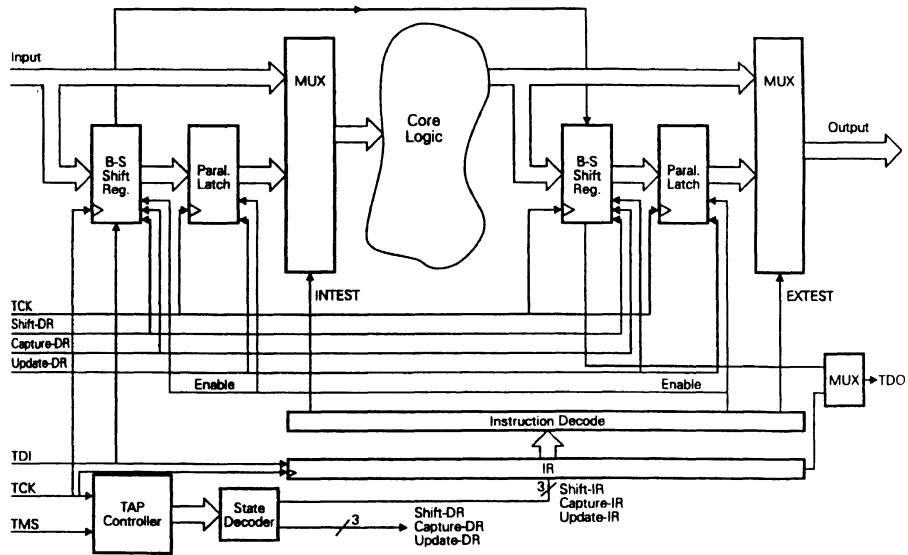


Fig. 1-18 Test control logic for BST

The way that these signals control the Boundary-Scan Test logic is depicted in figure 1-18. Here it can be seen that, under control of the State Decoder, either the

Instruction Register's serial path or the Boundary-Scan shift register stages can be connected between the IC's TDI and TDO pins. Figure 1-19 shows at IC level the test control logic and the serially connected Boundary-Scan cells. In practice also the IC input and output buffers are present, as drawn in figure 1-19.

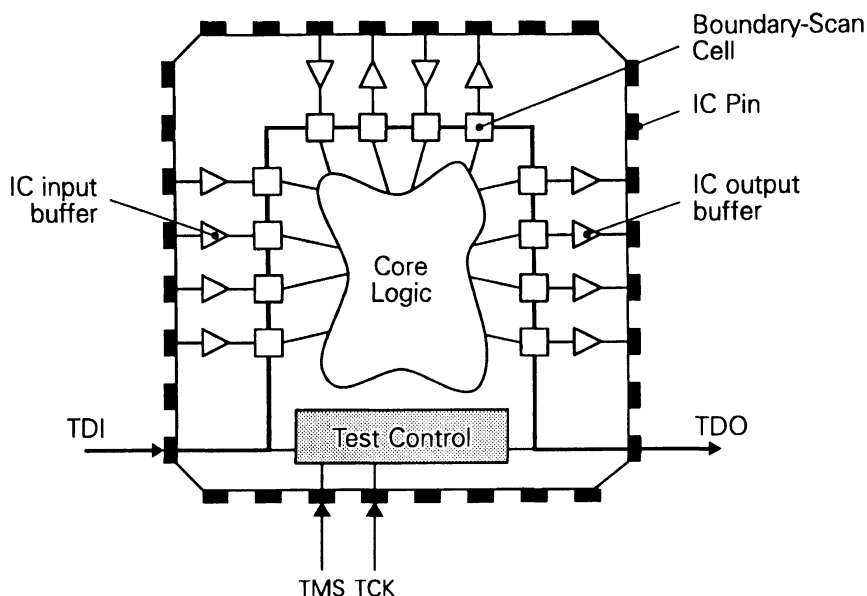


Fig. 1-19 IC including Boundary-Scan Register and Test Control

Figure 1-20 shows an example of a board level Boundary-Scan test configuration consisting of one serial test path.

Remember that the actual testing of the printed circuit board concerns the connections between the ICs, denoted as *net1*, *net2* ... *netn* (EXTEST), but the test architecture can also be used for checking the internal operation of the chip's core logic (INTEST).

Example Boundary-Scan Circuits

For a board level test procedure the Boundary-Scan registers should have three important properties:

1. they must be 'invisible' during normal operation of the IC's core logic,

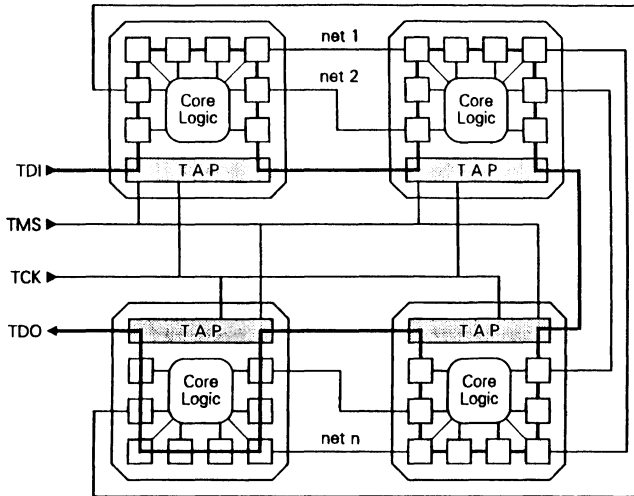


Fig. 1-20 PCB configuration with all IC BSRs connected in series

2. they must be able to isolate the IC interconnections on the board from the IC cores to allow testing the external IC connections, and
3. they should be able to isolate the IC from its surroundings on the PCB to also allow testing of the internal core logic.

An example of a commonly used cell design is given in figure 1-21.

The test properties are obtained as follows. To support normal operation (Test/Normal) the multiplexer 2 (MUX2) is kept at logical '0'. The control signal (Shift/Load) of MUX1 can be kept at '1' or '0'.

The BS cell depicted in figure 1-21 can be used both at the input and at the output related IC pins. For example, for an *INTEST* of the core logic (see figure 1-19) the BS cells at the IC *input* receive data shifted-in via TDI with MUX1 set to '1' (Shift) and clocked to the core logic with MUX2 set to '1' (Test). At the core *output* the BS cells must capture the resulting output signal for which MUX1 is set to '0' (Data from system). These signals are clocked towards TDO for examination, for which MUX2 is kept at '1'.

Note that the above depicted Boundary-Scan cell is just one sample representation. Other circuits and test features are discussed in chapter 2.

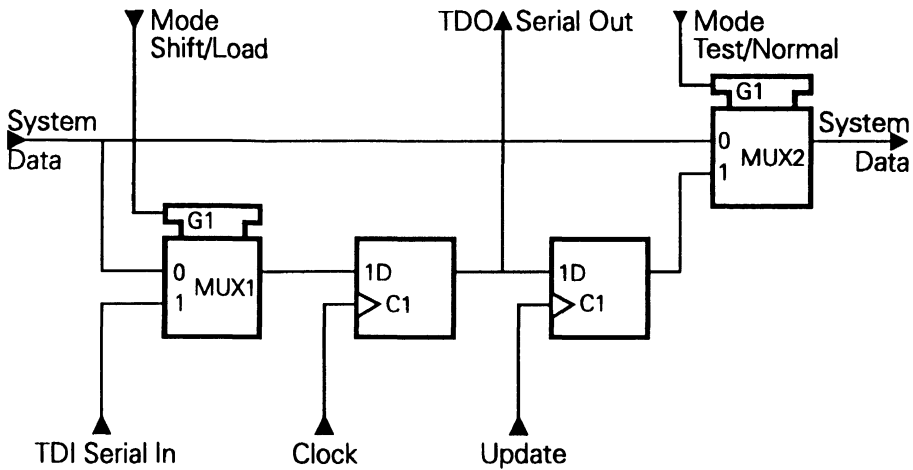


Fig. 1-21 Example of a commonly used Boundary-Scan cell

During testing, an IC's Boundary-Scan Register can also stay idle, thus without interfering with the in-core logic. For example, when the IC is not used in a board test it is senseless to include the IC's BS Register (BSR) in the PCB Boundary-Scan path. In such cases the BSR is short cut by the so called Bypass register, a one-bit register between the TDI and TDO pins to make the board test faster, thus more efficient.

The Bypass Register may be selected by loading the appropriate instruction into the Instruction Register. The presence of instructions opens a wide variety of test possibilities. For instance it is possible to have a Device Identification Register implemented in the component (IC), by means of which it can be tested whether the right component is mounted on the right place on the PCB. Or, it is possible to have manufacturer specific instructions for several purposes, e.g. emulation, self-test, test access to internal scan registers, etc. Figure 1-22 shows an example of an Instruction Register cell.

The IR operation and applications are detailed in chapter 2.

The TAP Controller (see figure 1-18) is a 16-state sequential machine operating synchronously with the TCK and responding to the TMS signal. Under control of the TAP Controller the instruction and test data are shifted from the TDI, through the registers and out again towards the TDO pin.

A Boundary-Scan test consists of four steps:

1. shift in and decode instructions, i.e. select a particular data register,
2. shift in test data,
3. execute the test and

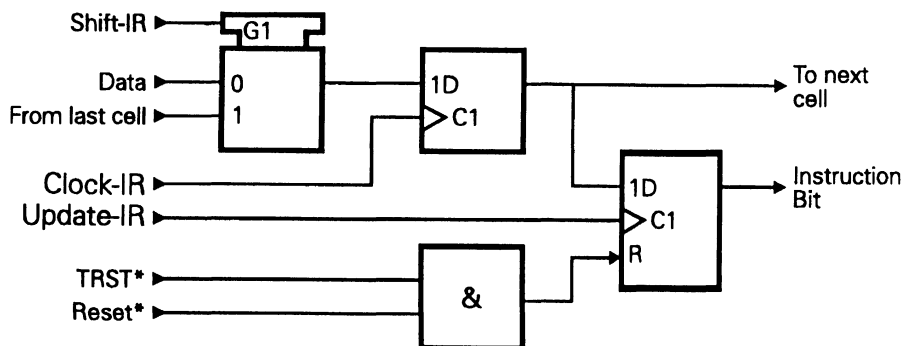


Fig. 1-22 An example of an Instruction Register cell

All the actions can be controlled by the TAP Controller and are described in chapter 2.

Constituent Elements of the BST Path

As mentioned above, applying Boundary-Scan Test (BST) on a PCB implies that Boundary-Scan (BS) cells are placed along the IC's input/output pins. For test purposes, these BS cells replace the conventional *physical* test pins (which can be placed on the IC contact pins during testing) by '*virtual*' test pins inside the IC.

In the conventional way an interconnecting copper track (or 'net') between the output of IC1 and the input of IC2 is tested by placing two test pins on the net nodes and applying test signals to one node while measuring the responses at the other node. These nodes (measuring points) could be physically located at the PCB soldering joints of the IC pins.

In the case of BST, the conventional nodes become '*virtual*' nodes located inside the BS cells. Including part of the IC itself in the test path means that the test path becomes longer, so more possible connection points are to be considered when a fault is detected. Figure 1-23 (introduced in [2]) gives an idea of the whole BST test path between two ICs.

The signal test path between two '*virtual*' test pins now includes:

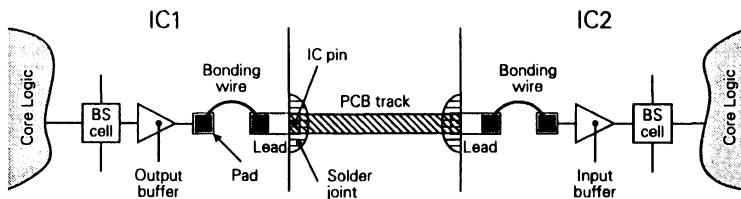


Fig. 1-23 BST path between two ICs

- the Output and Input Boundary-Scan cells,
- the Output and Input buffers,
- the Bonding pads, four in total,
- the Bonding wires, two in total,
- the IC pin soldering joints, two in total,
- the PCB copper track.

Obviously all these points can give rise to faults. Although the ICs are tested usually before leaving the supplier's factory, the ICs can, once mounted on the PCB, show faults. This can be caused by component handling during loading the PCB, for example due to mechanical or thermal shocks.

It should be noted that a deduced 'open' fault (path interrupted) from a BS test does not indicate on which of the above mentioned points the interruption occurs. For example the connections of the two soldering points and the PCB copper track in between can be faultless while the input buffer of IC2 is blown up due to electro static discharge, which then may cause the 'open' error. If one output is connected to several input pins, then further location of such a fault can be deduced with proper test patterns for all input pins involved (for details see chapter 5).

Chapter 2

THE BOUNDARY-SCAN TEST STANDARD

This chapter describes the elements defined by the IEEE Std 1149.1 for the Standard Test Access Port and Boundary-Scan Architecture. First the constituent parts of the Boundary-Scan architecture (both the mandatory and the optional items), are described followed by the instructions. The final section gives some rules which should be followed when BST designs are to be documented. Numerous descriptions are clarified, each with a practical example, making this chapter a tutorial for BST. For more detailed specifications of the BST standard please refer to the listing in the Appendix of this book, where these specifications are summarized for those readers who do not have the official IEEE Std 1149.1 manual readily available.

THE BST ARCHITECTURE

The IEEE Std 1149.1 distinguishes four basic hardware elements:

1. the Test Access Port (TAP),
2. the TAP Controller,
3. the Instruction Register (IR) and
4. a group of Test Data Registers (TDRs).

The first three elements and part of the group of TDRs are mandatory and *must* be supported by components which are compliant with the standard. The mandatory TDRs are the Boundary-Scan Register (BSR) and the Bypass register. The optional Test Data Registers comprise a Device Identification Register and one or more Design-Specific test data registers. Although the latter registers are optional, they must, if present in a component, obey the rules described in the IEEE Std 1149.1. The following sections detail each of these elements.

Figure 2-1 shows the standard Boundary-Scan architecture.

The operation of the Instruction and the Test Data registers is controlled by signals coming from the TAP Controller. The TAP controller is a sequential circuit which receives its controlling signals through the Test Access Port (TAP). The TAP only consists of IC contact pins. The standard TAP requires at least four IC pins (see also previous chapter) while a fifth pin may be provided for an optional active low reset signal TRST* (Test Reset), which can reset the test logic asynchronously to

the test clock signal (TCK). Usually, however, the reset operation is achieved through the TAP Controller.

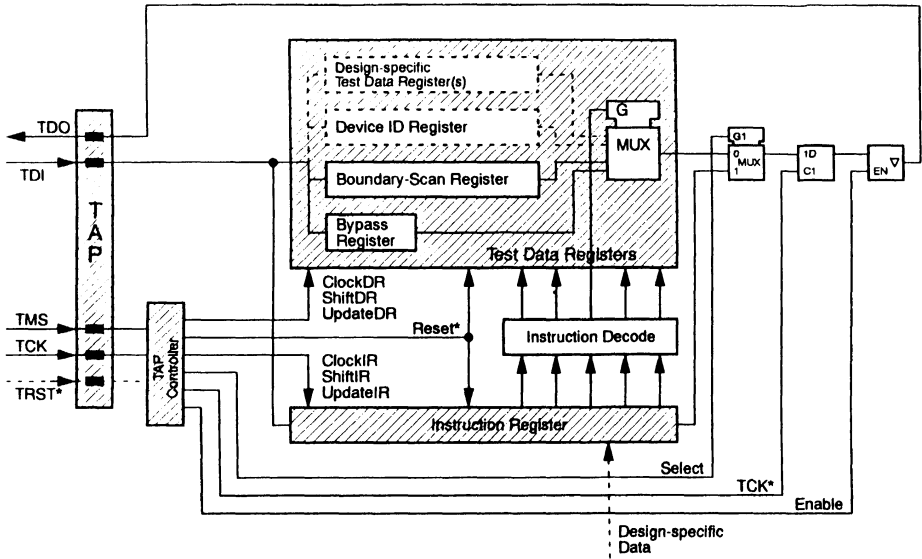


Fig. 2-1 The standard Boundary-Scan architecture
Note: The allowed optional registers and signals are shown in dotted lines.

The TAP Controller is driven by the signals Test Mode Select (TMS) and Test Clock (TCK), obtained from test execution hardware, supplied either externally from Automatic Test Equipment (ATE) or internally from system logic. More details are described in later sections.

Note in figure 2-1 that each of the named Registers can be selected into the serial signal path from TDI through TDO. The last multiplexer (MUX), as seen from TDI, selects either the Instruction Register or a Data Register. This multiplexer is directly controlled by the TAP controller. The preceding MUX selects which of the available data registers is to be switched into the TDI-TDO path, and is controlled by the decoded instruction.

TEST ACCESS PORT

The TAP provides access to the many test support functions built into an IC. It consists of four input connections, one of which is optional, and one output connection. The optional input connection is the TRST* pin.

The IEEE Std 1149.1 requires that the TAP connections are not used for any purpose other than testing.

The test signals connected to the Test Access Port have the following functions.

- *The Test Clock Input (TCK)*

The TCK signal allows the Boundary-Scan part of the ICs to operate synchronously and independently of the built-in system clocks. A PCB may include various ICs that contain a component-specific clock. To remain independent of such clock frequencies, the TCK must *not* interfere with any system clock. However, for some optional tests a synchronization of the two clocks may be required. TCK permits clocking test instructions and data into or out of the register cells. Shifting in data from the TDI input pin must occur on the rising edge (positive slope) of the TCK pulse and shifting out data towards TDO occurs at the falling edge (negative slope) of TCK. Loading of data at the system input pins occurs at the rising edge of TCK.

- *The Test Mode Select Input (TMS)*

The logic signals (0s and 1s) received at the TMS input are interpreted by the TAP Controller to control the test operations. The TMS signals are sampled at the rising edge (positive slope) of the TCK pulses. The signals are decoded in the TAP Controller to generate the required control signals inside the chip. When TMS is not driven it must be held at a logic 1. This may be obtained by a pull-up resistor on the TMS input pin in the TAP.

- *The Test Data Input (TDI)*

Serial input data applied to this port is fed either into the instruction register or into a test data register, depending on the states of the TAP Controller. The input data is shifted in at the rising edge (positive slope) of the TCK pulses. When TDI is not driven, a logic 1 must be present. This may be obtained by a pull-up resistor on the TDI input pin in the TAP.

- *The Test Data Output (TDO)*

Depending on the state of the TAP controller, the contents of either the instruction register or a data register are serially shifted out towards the TDO. The data out of the TDO is clocked at the falling edge (negative slope) of the TCK pulses. When no data is shifted through the cells, the TDO driver is set to an *inactive* state, for example to high impedance.

- *The Test Reset Input (TRST*)*

The TAP's test logic is asynchronously forced into its reset mode when a logic 0 is applied to the TRST* pin. Therefore, a non connected TRST* pin must react as if a logic 1 was applied. TRST* is an *optional* signal, because in any case the test logic must be designed such that it can be reset under control of the TMS and TCK signals. A TRST* signal can be used to force the TAP

Controller logic into its reset state *independently* of the TCK and TMS signals at power-up time, as is required by the standard. The optional TRST* pin requires one more IC input pin for test purposes.

Figure 2-2 shows a board level configuration on which the TRST* is applied. At the same time this figure shows, as an example, an application where two TMS signals are used to test the upper and lower serial path independently (compare with figure 1-20). The latter is allowed because, as will be discussed later on in this chapter, when no data is shifted through the cells, the TDO driver is set to an inactive (high impedance) state.

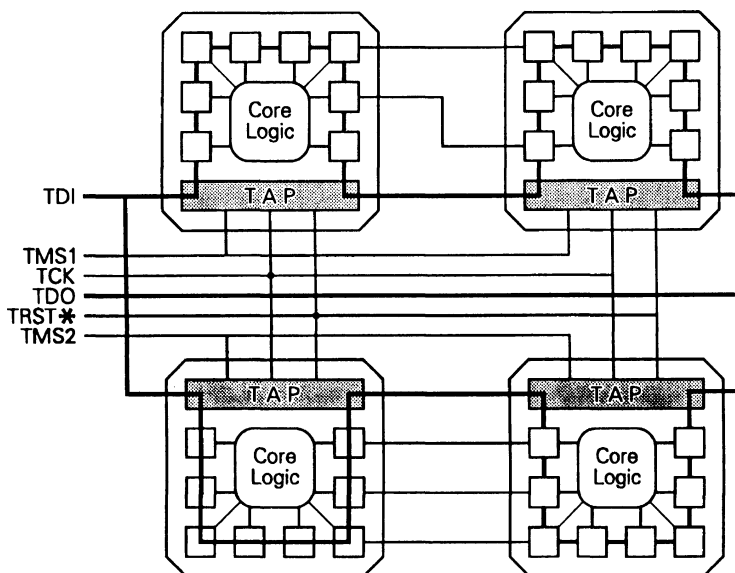


Fig. 2-2 BST configuration with two parallel paths

But care should be taken here, because the TMS signals must be controlled such that both the serial paths do not attempt to shift out signals simultaneously. A better practice in such cases is to use one serial net (figure 1-20) and sequentially set the ICs of one part of the serial path in the Bypass mode while testing the other part. The Bypass option is already mentioned in chapter 1 and is further discussed later on in this chapter.

TAP CONTROLLER

The basic function of the TAP Controller is to generate clock and control signals required for the correct sequence of operations of the circuitry connected to its

output; that is, the Instruction Register and the Test Data Registers. Its main functions are:

- provide signals to allow loading the instructions into the Instruction Register,
- provide signals to shift test data into (TDI) and test result data out of (TDO) the shift registers,
- perform test actions such as capture, shift and update test data.

The operation of the TAP Controller is explained by means of its state diagram as defined by the IEEE Std 1149.1, see figure 2-3. All state transitions (as indicated by the arrows) within the TAP Controller occur on the rising edge of the TCK pulse, whereas actions in the connected test logic (registers etc.) occur at either the rising or the falling edge of the TCK. The values with the arrows are the TMS values at the rising edge (positive slope) of TCK.

Note that in figure 2-3 some states are shaded and others not. The non-shaded states are 'auxiliary' states, i.e. they do not initiate a system action but are included to provide 'process control'. For example if the test engineer wants to leave the *Capture-DR* state for an *Update-DR* action *without* a shift operation, then the path through *Exit1-DR* can be used. The *Pause-DR* state is included to provide a wait time during shifting to allow for example a test pattern generator to fetch/prepare test data. Shifting can be resumed by leaving the *Exit2-DR* state with TMS held at 0. These control sequences may cost one or two extra TCK cycles in the test procedure, but the alternative to provide these control functions at the cost of one or more extra test pins (in addition to TMS and TCK) at the IC package, which is not a reasonable option.

After switching on the system, it is essential to ensure that shortly after power-up the TAP Controller is in its *Test-Logic-Reset* state, allowing the system busses and wired junctions to be controlled by the system itself. As will be shown, the TAP Controller can be set synchronously into its *Test-Logic-Reset* state following five rising edges of the TCK, provided TMS is held at logic 1. However, the worst-case time required to reach this state may exceed the one at which damage could occur. It can also not be guaranteed that the clock will be running at the time at which power is applied.

These problems can be solved in various ways, such as:

- inclusion of a power-up reset within the ICs,
- inclusion of an asynchronous reset (TRST*) facility within the IC, by which the TAP Controller is reset asynchronously with respect to the TCK,

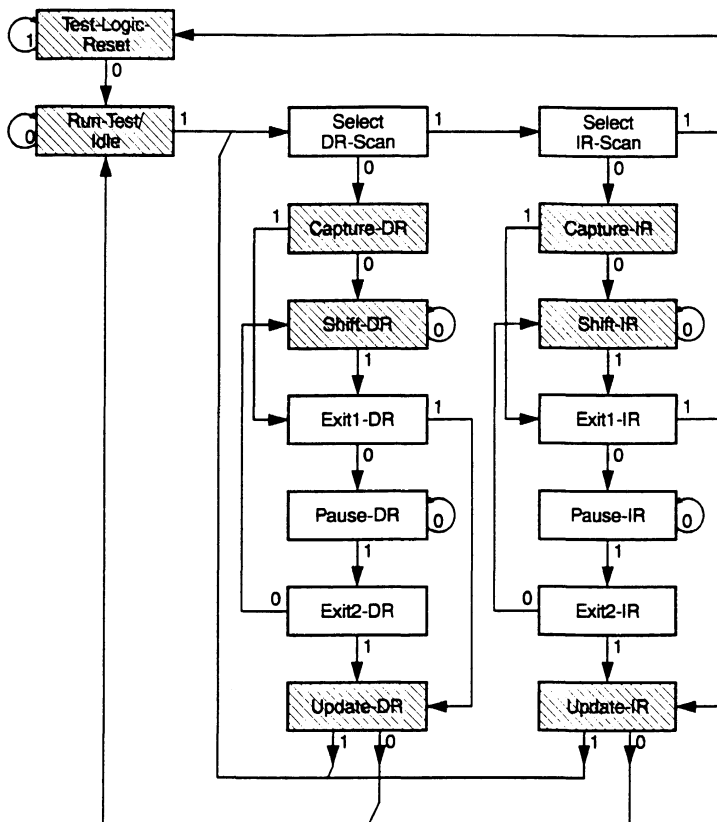


Fig. 2-3 State diagram of the TAP Controller

- asymmetric design of the latches or registers used to construct the TAP Controller.

Where a power-up reset facility is provided within the component, it can be used to initialize both the system and the test logic circuitry.

Now, suppose that after power-up using one of the above options the TAP Controller has reached its *Test-Logic-Reset* state. As long as the TMS value is held at 1, the controller remains in this state (see diagram). In order to leave this state, the level of the TMS is set to 0. Then, at the next rising edge (positive slope) of the TCK pulse, the TAP Controller leaves the *Test-Logic-Reset* state and enters the *Run-Test/Idle* state. The *Run-Test* part of this name relates to the start of a built-in self-

test and the part *Idle* applies to all other cases. As long as TMS is held at 0, the controller remains in the *Run-Test/Idle* state (see figure 2-3).

The first step in the test procedure is to load an instruction, so the '-IR' branch (right column) of the state diagram must be entered. As a result the Instruction Register (IR) will be connected between the TDI and TDO pins. In the '-DR' branch of the state diagram (middle column) the test Data Registers (DR) operate. To select the '-IR' branch from the *Run-Test/Idle* state, two TCK pulses (rising edges) are needed with TMS kept at 1. With this action the *Select DR-Scan* state is passed. At board level all IRs are serially connected, once the TAP Controller is in the '-IR' branch. Once in the *Select IR-Scan* state, the *Capture-IR* and the *Shift-IR* states can be entered by making TMS a 0 for the next two TCK pulses. Shift actions are continued as long as the TMS line is kept at 0 in the *Shift-IR* state of the controller for as many clock pulses as required. When TMS is next set to 1, the shift process is ended and the next state *Exit-IR* is entered. When TMS is again set to level 1 the *Exit-IR* state in the controller is left and the *Update-IR* is entered. The instruction data are now clocked into the parallel output stages of the IRs and the new instructions become valid in the ICs. The 'Exit' states in the TAP Controller are temporary states, because whatever the TMS value is, this state is left at the next rising edge of the TCK pulse. In between the *Pause-IR* is present in the '-IR' branch of the state diagram. When in this state the TMS signal is kept at 0, the TAP Controller remains in this state. This is needed with multiple chains of different lengths. If after a pause the shift process must be resumed, the *Pause-IR* state is left with TMS at 1. Then *Exit2-IR* is entered which should be left then with TMS at level 0 to bring the controller back to its *Shift-IR* state.

After the *Update-IR* state the TAP Controller enters either the *Run-Test/Idle* state (TMS at 0) or directly to the *Select DR-Scan* state (TMS at 1). When the controller is in the '-DR' branch, the selected test data registers are serialized and similar actions as described here for the '-IR' can be initiated for the test data registers.

In this way the TAP Controller supports all test procedures that may be required by the test engineer. Figure 2-4 shows a timing diagram of the sequences described above, except that the *Pause-IR* is left via the *Exit2-IR* state.

Note in the state diagram (figure 2-3) that, in whichever state the TAP Controller is, a maximum of five TCK pulses will always reset the controller in its *Test-Logic-Reset* state, provided TMS is held at a logical 1.

The *non-shaded* states in figure 2-3 (*Select-DR-Scan*, *Select-IR-Scan*, *Exit1-DR*, *Exit1-IR*, *Pause-DR*, *Pause-IR*, *Exit2-DR* and *Exit2-IR*) are used to control the required test sequence. The *Pause* is provided to temporarily halt the shifting process. This state can be used, for example, to allow synchronization between TCK and system clock signals, when needed for certain tests. These 'Pause' states are maintained as long as the level of TMS is kept low (logical 0). Further to that, the

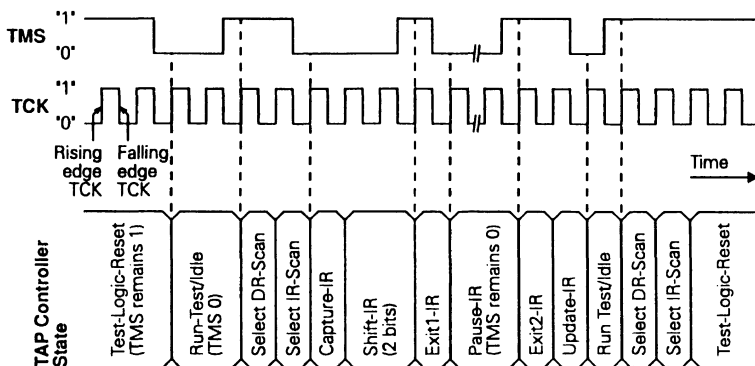


Fig. 2-4 Example timing diagram of TAP Controller

state *Exit1* is present in order to provide a three way branch starting from *Capture* to *Shift* or (via the *Exit1*) to either *Pause* or *Update* just by using one control signal (TMS). In the same way *Exit2* is provided to allow a three way branch starting from *Pause* to either stay in the *Pause* state or leave *Pause* (via the *Exit2*) to either *Update* or *Shift*. So with this construction a three way branch is controlled with one signal (TMS), for which two control signals were required otherwise.

It should be noted that when an *Exit* state is left with the TMS at a logical 1, an *Update* action takes place, meaning that the scanning process has terminated.

In the following subsection the operation of all possible TAP Controller states are described, following the formal specifications.

State Descriptions

All controller states shown shaded in the state diagram (figure 2-3) control the changes of the connected test logic. The resulting signal flows through the Boundary-Scan Cell and the Instruction Register are described below, using figures 1-21 and 1-22 respectively.

- *Test-Logic-Reset*

In this controller state (see figure 2-5) all *test* logic is disabled, i.e. all core system logic operates normally. Whatever the state of the controller is, it will enter the *Test-Logic-Reset* state when the TMS signal is held high (logic 1) for at least five rising edges of the TCK pulse. The exact required number of TCK pulses depends on the current state of the TAP Controller. The controller remains in this state while TMS is held high. If the optional signal TRST* is

present in the TAP, it can be used to force the controller to the *Test-Logic-Reset* state at once (asynchronous reset), at any moment during circuit operation.

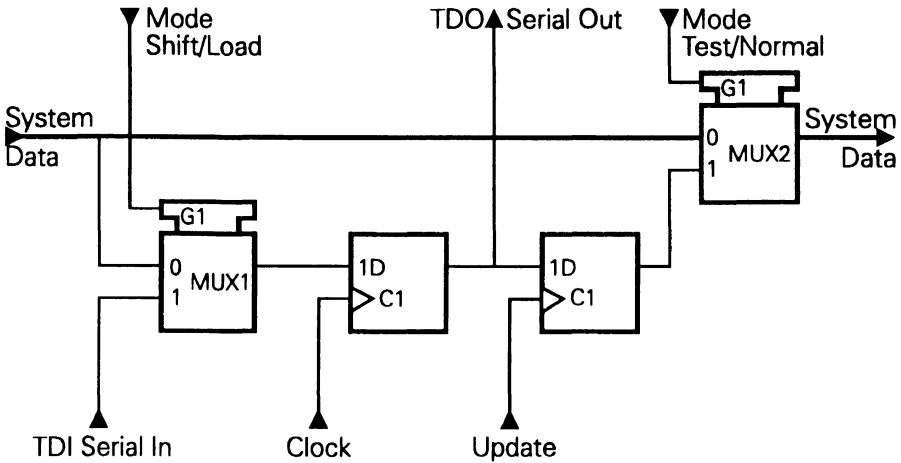


Fig. 2-5 Signal flow (bold lines) during Test-Logic-Reset

- *Run-Test/Idle*

This is a controller state between the various scan operations, see previous section. Once the controller is in this state, it will stay there as long as the TMS signal is held low (logical 0). The operation of the connected test logic depends on the instruction contained in the instruction register. If the instruction causes a built-in self-test function to execute (e.g. the RUNBIST instruction) the TAP Controller has to wait for its completion. This can be done by holding the TAP Controller in the Run-Test/Idle state. If an instruction does not cause built-in self test functions to execute, such as select a data register to scan, then all selected test data registers will stay *Idle* while the TAP Controller is in the Run-Test/Idle state. The current instruction does not change while the controller is in this state.

- *Capture-DR*

In this controller state (see figure 2-6), data is parallel-loaded from the parallel inputs into the selected test data register. The register retains its previous state if it does not have a parallel input or if capturing is not required for the selected test. The action takes place at the rising edge (positive slope) of the TCK pulses.

- *Shift-DR*

In this controller state (see figure 2-7), the previously captured data is shifted out towards the TDO, one shift-register stage on each rising edge (positive slope) of the TCK pulse. If a selected test data register is not placed in the serial path, it retains its previous state.

falling edge (negative slope) of the TCK pulses. The shift-register stage in a selected test data register retains its previous state.

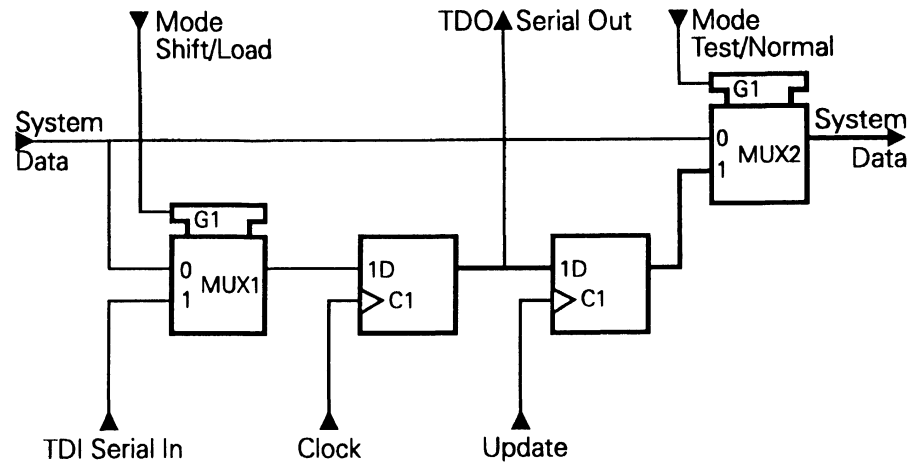


Fig. 2-8 Signal flow (bold lines) in BS cell during Update-DR

• *Capture-IR*

In this controller state (see figure 2-9), the previously shifted-in instruction data is parallel-loaded into the shift-register stage of the instruction register. In addition, design-specific data may be loaded into a shift-register stage which may not to be set to a fixed value. In this way the IR-path between TDI and TDO on the board (see figure 2-12) can be checked as to whether or not the instructions can be shifted in correctly or not. The action takes place at the rising edge (positive slope) of the TCK pulses. Test data registers which are selected by the current instruction retain their previous state.

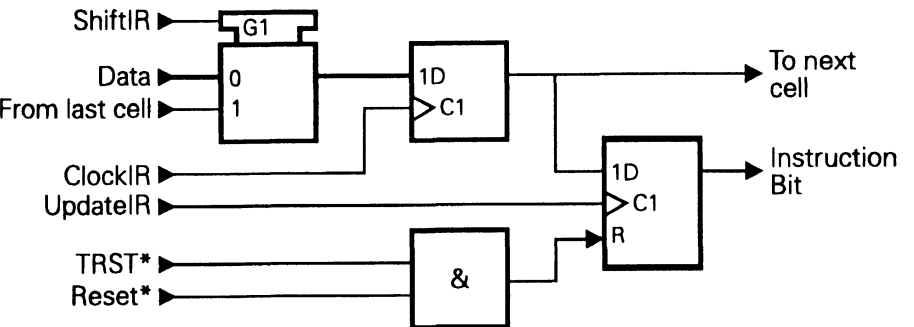


Fig. 2-9 Signal flow (bold lines) in IR cell during Capture-IR

• *Shift-IR*

In this controller state (see figure 2-10), the previously captured data is shifted out towards the TDO, one shift-register stage on each rising edge (positive slope) of the TCK pulse. Selected test data registers retain their previous state.

The current instruction does not change in this controller state.

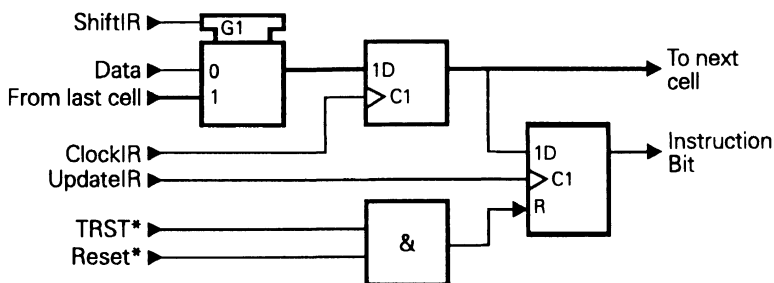


Fig. 2-10 Signal flow (bold lines) in IR cell during Shift-IR

- *Update-IR*

The shifted-in instruction data is loaded from the shift-register stage into the parallel instruction register (see figure 2-11). The new instruction becomes valid when the TAP Controller is in this state. All the test data shift-register stages which are selected by the current instruction retain their previous state. The action takes place at the falling edge (negative slope) of the TCK pulses.

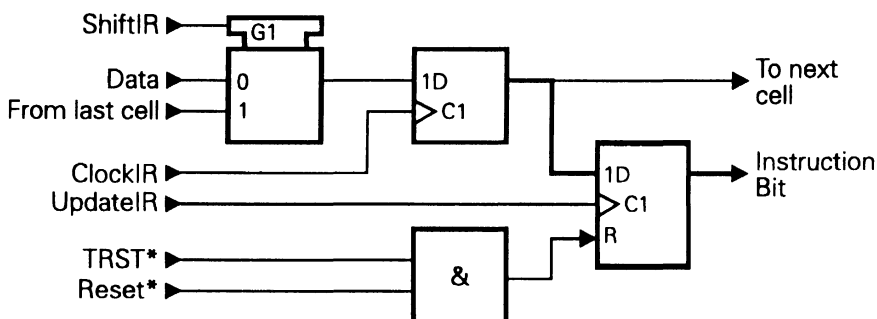


Fig. 2-11 Signal flow (bold lines) in IR cell during Update-IR

THE INSTRUCTION REGISTER

The Instruction Register allows test instructions to be shifted into every IC along the PCB-level path. Therefore, in the Shift-IR controller state, all IC instruction registers are connected in series at board level between TDI and TDO. A test instruction defines the test data register to be addressed and the test to be performed. This may vary per IC on the board, as is depicted in figure 2-12.

The IEEE Std 1149.1 defines a number of mandatory and optional instructions. Further design-specific instructions may be added by the designer of the IC.

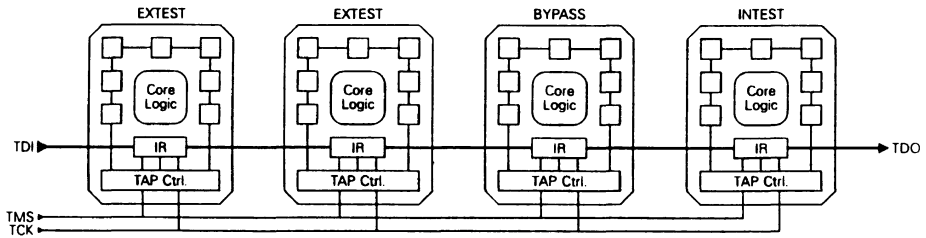


Fig. 2-12 ICs set for various tests

The design of the Instruction Register is a serial-in parallel-out register (see figure 1-22). Each Instruction Register cell has a shift-register flip-flop and a parallel output latch. The shift-register flip-flop holds the instruction bit moving through the Instruction Register. The parallel output latch holds the current instruction bit, latched into it from the shift-register during the *Update-IR* state. The latched instruction can only be terminated/changed when the TAP Controller is in its *Update-IR* or *Test-Logic-Reset* state. No inversion of data between the serial input and the serial output of the Instruction Register is allowed to take place.

The Instruction Register must contain at least two shift-register-based cells which can hold instruction data. These two mandatory cells are located nearest to the serial outputs, i.e. they are the least significant bits. Figure 2-13 shows the set-up of an Instruction Register.

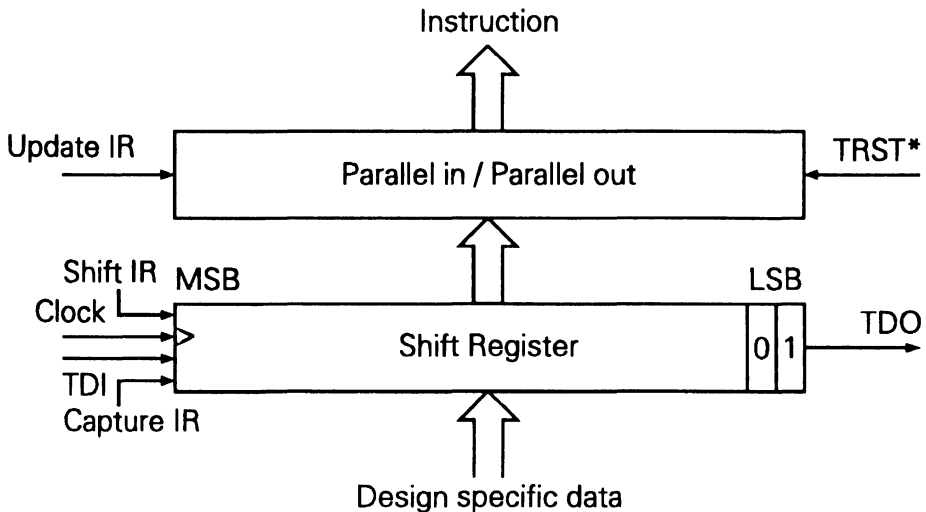


Fig. 2-13 Set-up of an Instruction Register

The values of the bits are 0 and 1 (the 1 is the least significant bit). These fixed values are used in locating faults in the serial path through the ICs on the PCB. The fixed binary "01" pattern must be loaded in the *Capture-IR* state of the TAP Controller.

Additional parallel inputs are allowed to permit capture of design-specific information. In cases where no design-specific instructions are defined, it is recommended that the design-specific cells are designed such that the parallel inputs load a fixed logic value (0 or 1) in the *Capture-IR* state of the TAP Controller.

Table 2-1 shows the operation of the Instruction Register (IR) in each of the states of the TAP Controller.

Table 2-1 IR Operation in Each TAP Controller State

Controller state	Shift-Register Stage	Parallel output
<i>Test-Logic-Reset</i>	Undefined	Set to give the IDCODE (or BYPASS) instruction
<i>Capture-IR</i>	Load 01 into LSBs and design-specific data or fixed values into MSBs	Retain last state
<i>Shift-IR</i>	Shift towards serial output	Retain last state
<i>Exit1-IR</i>	Retain last state	Retain last state
<i>Exit2-IR</i>	Retain last state	Retain last state
<i>Pause-IR</i>	Retain last state	Retain last state
<i>Update-IR</i>	Retain last state	Load from shift-register stage into decoder
All other states	Undefined	Retain last state

Referring back to the description of the TAP Controller state diagram along with figure 2-3, it can be noted from the table that the states *Exit1-IR*, *Exit2-IR* and *Pause-IR* do not influence the operation of the test logic.

TEST DATA REGISTERS

The Boundary-Scan Architecture must contain a minimum of two test data registers: the Bypass Register and the Boundary-Scan Register. A third register, the Device Identification Register, is optional. One or more additional design-specific test data registers are allowed too. The design-specific registers may, but need not, be

publicly addressable. Additional registers are accessed by adding instruction codes to the architecture’s test-instruction set.

Each test data register included in the architecture has a fixed length as designed by the IC designer and can be accessed through one or more instructions that can be shifted into the Instruction Register. Figure 2-14 shows a detail of the standard Boundary-Scan Architecture (compare with figure 2-1).

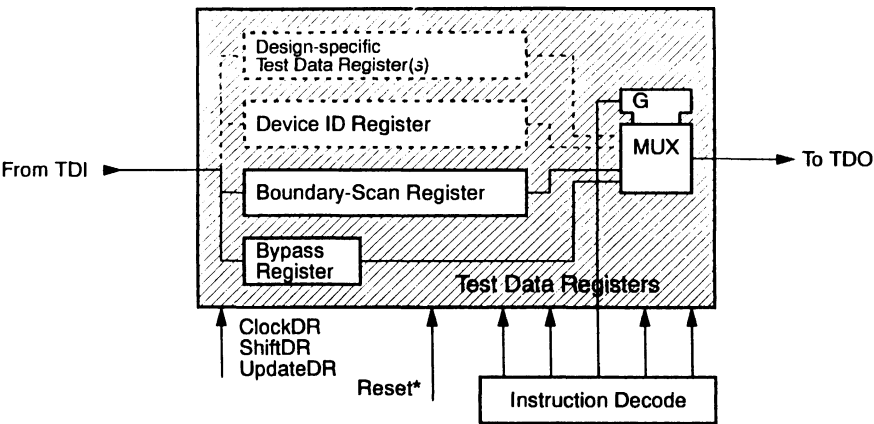


Fig. 2-14 Overview of the Test Data Registers
Note: The allowed options are shown in dotted lines.

The depicted Test Data Registers are described in more detail in the following sections.

It should be mentioned here that the length of a test data register must be fixed (defined by the IC designer), independent of the instruction by which it is accessed. A named test data register may share registers from other logic circuitries, but each part and each addressable combination of parts must have a unique name and the resulting design must be compatible with the IEEE Std 1149.1.

Figure 2-15 shows an example.

Table 2-2 shows the operation of the Test Data Registers in each of the states of the TAP Controller.

Referring back to the description of the TAP Controller state diagram along with figure 2-3, it can be noted from the table that the states *Exit1-IR*, *Exit2-IR* and *Pause-IR* do not influence the operation of the test logic.

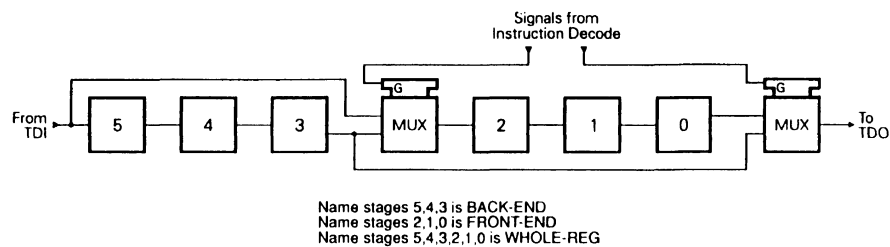


Fig. 2-15 Example of a combined test data register

Table 2-2 Operation of the Test Data Register Enabled for Shifting

Controller state	Action
<i>Capture-DR</i>	Load data at parallel input into shift-register stage. Parallel output register or latch retains last state.
<i>Shift-DR</i>	Shift data towards serial output. Parallel output register or latch, retains state.
<i>Exit1-DR</i>	Retain last state.
<i>Exit2-DR</i>	Retain last state.
<i>Pause-DR</i>	Retain last state.
<i>Update-DR</i>	Load parallel output register or latch from shift-register stage. Shift-register stage retains state.
All other controller states	Registers which have a parallel output maintain their last state of the output; otherwise undefined.

The Bypass Register

The mandatory Bypass Register is a single stage shift-register. When selected, the shift-register is set to a logic zero on the rising edge (positive slope) of the TCK with the TAP Controller in its *Capture-DR* state. It then provides a minimum length serial path for the test data shifting from the component’s (IC) TDI to its TDO.

The benefit of the Bypass Register is a shortened scan path through the Boundary-Scan architecture when scan access of the test data registers is not required, or it can be used to leave the IC in functional mode during testing. Chapter 1 also describes some applications of the Bypass register.

Suppose a printed circuit board has 50 ICs and the Boundary-Scan registers of all ICs are connected into a single serial chain. If the average component register length is 100 stages, then the total serial path length contains 5000 stages. If only one IC in this chain is to be tested (with the *INTEST* instruction) then the remaining 49 ICs can be bypassed. For this test of one IC, only 100 stages plus 49 bypass stages are required. Now, instead of all 5000, only 149 (about 3%) of the stages are involved. This reduces testing time considerably in two ways:

1. The test cycle is shortened, in this example, by 97%.
2. Only the 100 bits for that one IC under test are of importance. Hence, the rest of the bits need not be stored in the automatic test equipment, which shortens diagnosis times.

Figure 2-16 shows an example of an implementation of a Bypass Register or cell.

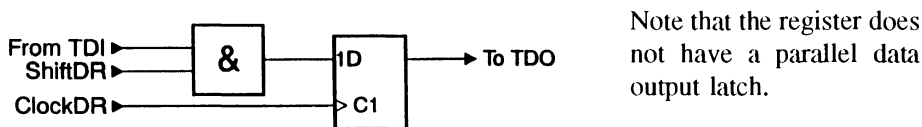


Fig. 2-16 Implementation of a Bypass Register

When the Bypass Register is selected, the shift-register stage is set to a

logical zero on the rising edge (positive slope) of the TCK pulse when the TAP Controller is in its *Capture-DR* state. This allows to test the presence or absence of the optional device identification register. When the device ID is present, then a constant logic 1 must be loaded in its LSB. When the *IDCODE* instruction (see the next section) is loaded into the Instruction Register, then a subsequent data register scan cycle causes the first data bit of each component to be shifted out. So when this bit is a 1, then the optional device ID register is present, otherwise the instruction selects the Bypass register, out of which a logic 0 is shifted.

The Boundary-Scan Register

The mandatory Boundary-Scan Register (BSR) consists of a series of Boundary-Scan Cells (BSCs) arranged to form a scan path around the core logic of the IC (see figure 1-19). It provides the following features:

- It allows testing of circuitry external to the component (IC). A typical example is the interconnect test. Also logic circuitry embedded between Boundary-Scan testable devices (RAM, ROM, glue logic) can be tested, using the *EXTTEST* instruction.

- It allows testing of the core logic (INTEST), while also providing defined conditions at the periphery of the core logic.
- It allows sampling and examination of the input and output signals without interfering with the operation of the core logic.
- And last but not least: it can stay idle, thereby showing virtually no load to the core logic when signals are flowing through the system.

Chapter 1 describes how the test data flow in a component (IC) is controlled by the Boundary-Scan Register for various instructions and figure 1-18 shows the interconnections between the Instruction Register, the Boundary-Scan cells and the TAP Controller.

A Boundary-Scan Cell

For ease of understanding, an example of a 'universal' Boundary-Scan Cell (BSC) is considered. The example is not prescribed by the IEEE Std 1149.1 standard nor is it considered here as a preferred configuration. But it is a configuration that could be used at both the input and the output pins of an IC. However, not all elements of the cell are always needed in a particular application and the building blocks composing the cell can also be different.

A Boundary-Scan register cell must be dedicated to the test logic only, i.e. it shall never be used for system operation. In order to meet the requirements of all instructions, it must be possible in a BSC to move test data through the register without interfering with the normal system operation. This can be achieved by making the shift-register stages and their parallel output latches a dedicated part of the cell.

Figure 2.17 shows an implementation of a Boundary-Scan Cell which can be used at both input and output pins. This configuration is often referred to in publications. However, it is certainly not the best one on all occasions. A simpler cell can for example be used at the input pins, if one follows strictly the minimum requirements of the IEEE Std 1149.1, see figure 2-18. This cell may be preferred over the previous one in delay sensitive circuits, for example at clock inputs.

The minimum requirement follows from the mandatory instructions as prescribed in the IEEE Std 1149.1: the *EXTEST* and the *SAMPLE/PRELOAD* instructions. So for the input cell no more logic elements are necessary, because the test results from a PCB interconnect test with the *EXTEST* instruction can be captured and further shifted towards the TDO pin. Obviously, the test of core logic of an IC which is mounted on a PCB is an option in the standard, for which the *INTEST* instruction is prescribed if it is supported by the chip manufacturer.

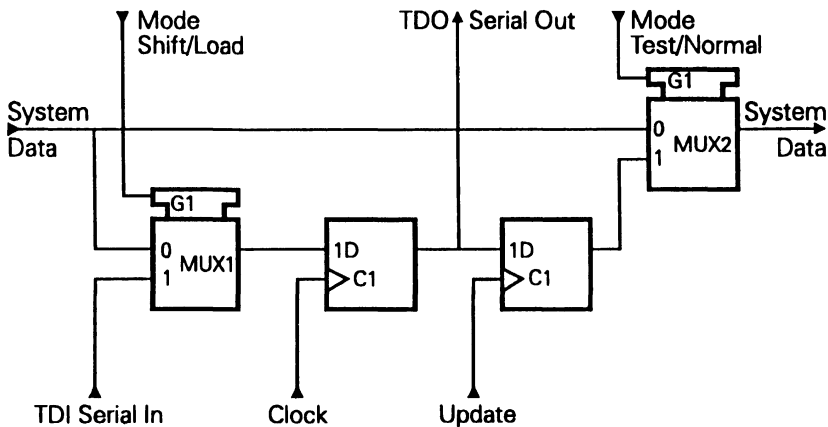


Fig. 2-17 An implementation of a Boundary-Scan Cell

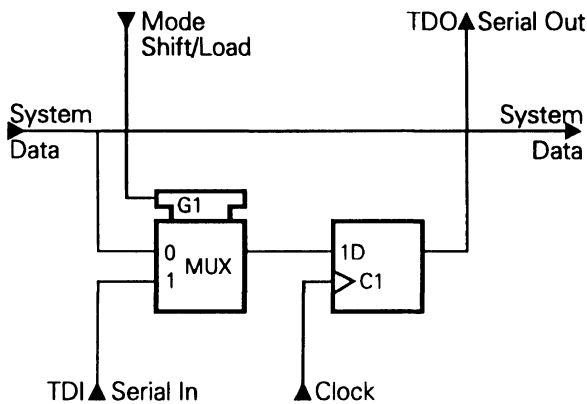


Fig. 2-18 Minimum BSC configuration for input pins

At the output pins, the cell depicted in figure 2-17 fulfils the minimum requirement.

System Pin Related Cells

Next to the mandatory overall specifications of the Boundary-Scan Register cells, additional specifications are valid for cells connected to the various types of *component or system pins*. Figure 2-19 shows schematically how the Boundary-Scan Cells surround the core logic and provide signals for the various output buffers. Control of the BSCs is provided by the TAP Controller. Some details concerning the various output cells are given below.

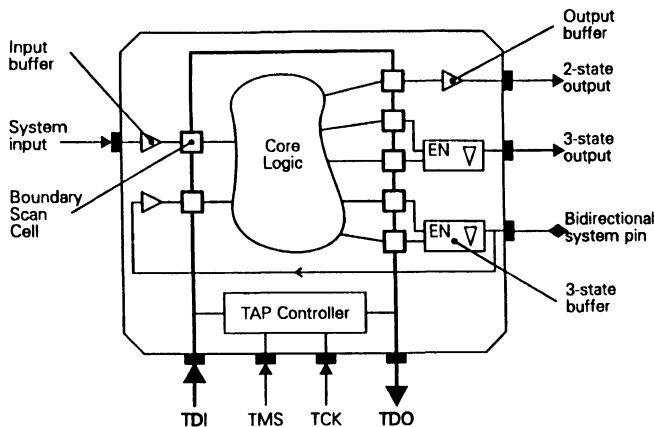


Fig. 2-19 Boundary-Scan cells with various output buffers

Notice that the discussed cells may also transmit signals coming from the purely digital core logic into a mixed analog/digital circuit block.

Note: In the descriptions below various instruction names are mentioned. These instructions are explained in a subsequent section.

- *Cells at 2-State Output Pins*

At 2-state output pins the signals can be set only at a high or a low logical level at any given instant. Therefore, only one Boundary-Scan cell per pin is sufficient to observe or control the state of the pin (see figure 2-19). Problems may be encountered when Boundary-Scan cells drive signals into logic blocks external to the component. Such an external block may contain asynchronous logic that will be set into undesirable states when shifting patterns appear at its input. Also, Boundary-Scan output signals may be fed into a clock input of the external block, which may produce hazardous effects if the logic is not shielded from the shifting patterns.

Therefore, the Boundary-Scan cell should be designed such that these problems are guaranteed to be avoided. The example cell in figure 2-17 fulfils this rule. In particular the parallel output register or latch can affect the state of an output driver at a system pin as in figure 2-19. This ensures that, while the *EXTEST* instruction is selected, the data driven from a component to an external logic block changes only after completion of the shifting process, at the update state transition.

- *Cells at 3-State Output Pins*

At 3-state output pins the signals can be actively set at a high or a low logical level *and* the pin can be held at an inactive state. Therefore, at least two Boundary-Scan cells per pin are needed to observe or control the state of the pin (see figure 2-19).

The 3-state system output pin may drive a wired junction on PCB level, using the *EXTEST* instruction. Therefore it must be possible to drive the wired junction from each of the possible driving pins independently. To that purpose the 3-state and bidirectional output drivers can be used.

At PCB level the output pin may also drive a component of which the core logic is being tested using the *INTEST* or *RUNBIST* instruction. Two options are available to avoid contention in such cases:

1. The state of the system pin can be fully user defined like a 2-state output pin.
2. The system pin can be driven into its inactive high-impedance state. This is allowed since components must tolerate high-impedance conditions during normal system operation.

Figure 2-20 shows the details for such a configuration.

The internal connections between the Boundary-Scan Cells may be as follows: The control lines of both cells (*Shift-DR*, *Clock-DR*, *Update-DR* and *Mode*) are connected in parallel. The TDO signal of the *Output cell* is connected to the TDI of the *Control cell*. In this way the TDI/TDO serial shift-register path is maintained. But other internal connections between the two cells are also possible.

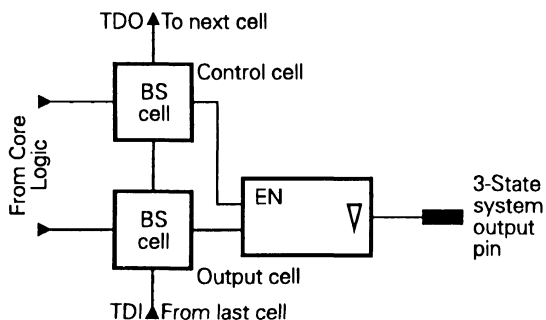


Fig. 2-20 Cell configuration at a 3-state output pin

Note: When two more or more 3-states pins are present (e.g. connected to a bus) it is allowable to control the respective enable stages with one Boundary-Scan cell. Figure 2-21 shows an example.

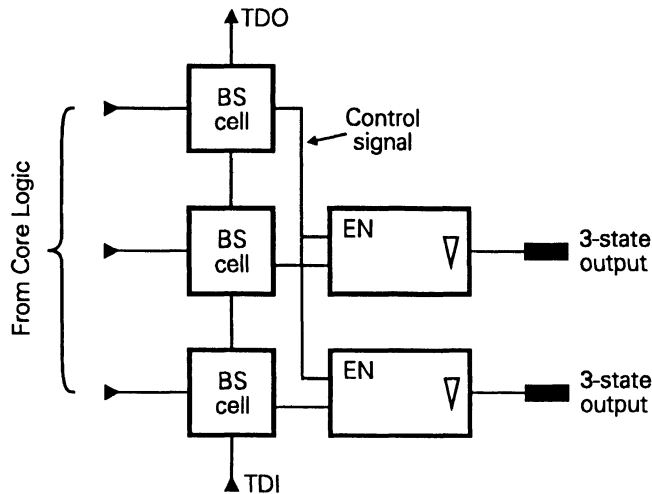


Fig. 2-21 One BS cell controls several 3-state outputs

- *Cells at Bidirectional Pins*

Bidirectional pins may be either a 2-state (for example formed as a combination of an open-collector output and an input) or a 3-state pin (for example formed as a combination of a 3-state output and an input). As a matter of fact, the IEEE Std 1149.1 requirements are a merge of those for both the 2-state or 3-state output pins and the system input pins.

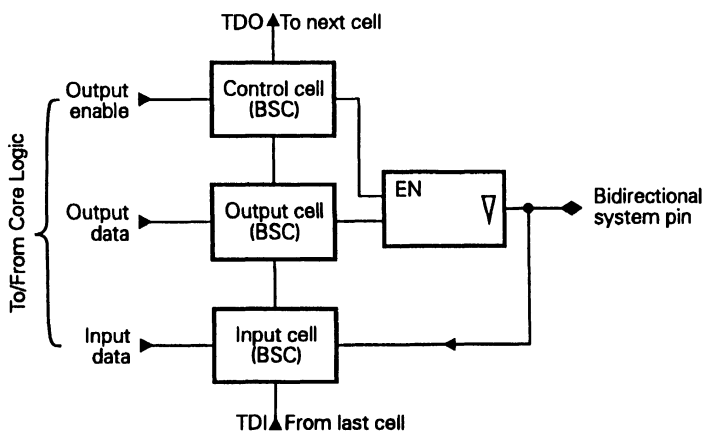


Fig. 2-22 Cell configuration at a bidirectional system pin

Figure 2-22 shows the details of a configuration at a 3-state output pin. The internal connections between the Boundary-Scan Cells for the bidirectional output pin are the same as in the 3-state output pin, except that the parallel output flip-flop of the control cell may have an extra input for the *Reset** signal.

Note: One output cell for several pins is allowed.

The Device ID Register

The Device Identification Register is an optional register that may be included in an IC. But *if* it is included, it must follow the requirements described in the IEEE Std 1149.1.

The Device ID Register provides binary information about the manufacturer's name, part number and version number of the component (IC). This is important, for example for the following applications.

- In the factory, it allows verification that the correct IC has been mounted on the proper place on the PCB.
- When a component (IC) has been replaced, the version number of the replacement can be checked, and if required the test program can be modified.
- It may be desirable to blindly interrogate a PCB design by a controller unit in order to determine the type of each component on each board location without further functional knowledge of the design.
- When a PCB is added to a configuration at system level, the system test program can be adjusted to the new PCB and the ICs mounted on it.
- The correct programming of off-line programmed ICs can be checked.

The register must consist of a 32 bits shift-register, parallel-in and serial-out. The register does not need a parallel output latch. Like the bypass register, the normal operation of the IC can continue while the ID register is in use.

The structure of the data loaded into the Device ID Register in response to the *IDCODE* is shown in figure 2-23.

The least significant bit (LSB) is always loaded with a logical 1. This is done in conjunction with the Bypass Register, which is always set to a logical 0. If the optional Device ID Register is not present, the Bypass Register is chosen when an *IDCODE* instruction is executed. If subsequently the first bit shifted out of the component during a test data scan is a 0 (zero), then it can be deduced that the

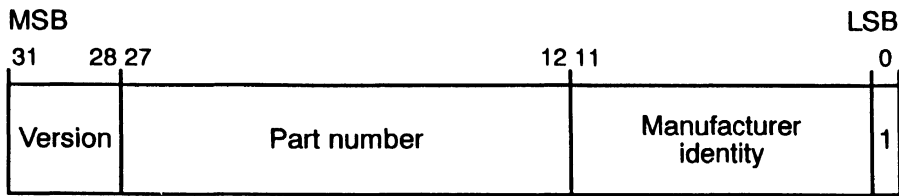


Fig. 2-23 Structure of a Device Identification Register

component has no Device ID register. The bits 1 through 11 contain an 11 bits manufacturer code, which is derived from a scheme controlled by the Joint Electron Device Engineering Council, abbreviated as JEDEC [3]. This scheme allows 2032 manufacturer identifications. The IEEE Std 1149.1 prescribes that the manufacturer code *00001111111* is not permitted for components claiming compatibility with the standard. This code can then be used as a known invalid code in test algorithms. The bits 12 through 27 allow 2^{16} ($\approx 65,000$) part number codes. The remaining bits 28 through 31 permit 16 variants of a part to be distinguished, for example between chips showing a different behaviour on instructions or have different responses to data sent or received through the TAP.

The need for interrogation of the board component becomes evident when on a board components are used that can be user programmed for certain functions. Another application may be in systems which can be configured with boards for various functions, possibly supplied from different manufacturers.

Figure 2-24 shows an example configuration of a Device ID Register cell.

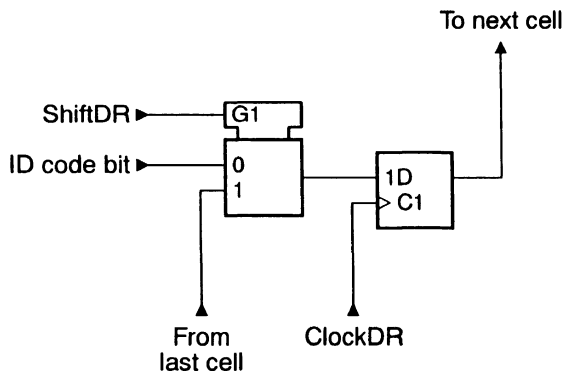


Fig. 2-24 An example of a Device ID design.

Design-Specific Registers

The manufacturer may add test data registers dedicated to his own design. For this purpose, design-specific instructions have been developed. In figure 2-1 at the beginning of this chapter, these options are shown in dotted lines.

One point should be made here. The manufacturer may decide whether he makes the instructions for the design-specific register available in his component catalogue or not. If not, he needs the design-specific registers only for his own in-house testing.

INSTRUCTIONS

Instructions are serially loaded from the TDI into the test logic during an instruction-register scan cycle, with the TAP Controller in its *Shift-IR* state. Subsequently the instructions are decoded to achieve two basic functions.

1. Select the set of test data registers that may operate while the instruction is current. Non-selected test data registers must be controlled so that they do not hamper the normal on-board operation of the related ICs.
2. Define the serial test data path that is used to shift data between TDI and TDO during data register scanning.

Instruction codes that are not used to provide control of test logic must be equivalent to the *BYPASS* instruction (described later). This makes sure that a test data register is always connected between TDI and TDO for every possible instruction code. So every instruction code will produce a defined response.

The IEEE Std 1149.1 specifications of the instructions cover:

- the instructions, whether they are mandatory or optional,
- test data registers which are to be connected in the serial TDI/TDO path,
- binary code patterns (if applicable), and
- the data flow through Boundary-Scan Register cells and/or core logic signals.

Public and Private Instructions

Public instructions are documented and supplied by the manufacturer along with the component. These instructions provide the component user with access to the test

features, so that the user can apply the instructions in test tasks on component, PCB or system level.

In addition to the public instructions, private instructions exist, which are available solely for the use of the component manufacturer. The operation of private instructions need not be documented. If private instructions are utilized in a component, it remains the responsibility of the vendor to clearly identify those instruction codes that, if selected, will damage the component or cause hazardous operation of the component. The following subsections describe the public Boundary-Scan instructions.

The BYPASS Instruction

The *BYPASS* instruction is the only instruction defined by the IEEE Std 1149.1 that can be applied when the Bypass Register is to be selected in the TDI-TDO path. The binary code for the *BYPASS* instruction must be {111...1} or said as 'all 1s'. The use of this register and the purpose of the code is already described in previous sections, including chapter 1. The instructions can be entered by holding the TDI at the logical high level.

Note that when the TDI is not terminated (e.g. caused by an open on the board level), it behaves as if the input was high (logical 1). This causes the Bypass Register to be selected following an instruction-scan cycle, so avoiding any unwanted interference with the normal IC operation.

If the optional Device ID Register is not present, then the *BYPASS* instruction is forced into the latches at the parallel outputs of the Instruction Register when the TAP Controller is in its *Test-Logic-Reset* state. This ensures a complete serial TDI/TDO path in either case.

The SAMPLE/PRELOAD Instruction

This is a mandatory instruction used to scan the Boundary-Scan register without interfering with the normal operation of the system core logic. Received system data at the input pins is supplied to the core logic without modification and system output data is driven to the output pins without interference. Meanwhile samples of these signals can be taken and the sampled data can be shifted through the Boundary-Scan registers.

So the instruction supports two functions:

1. It allows a *SAMPLE* ('snapshot') of the normal operation of a component (IC) to be taken for examination.

- 2. Prior to the selection of another test operation, a *PRELOAD* can take place of data values into the latched parallel outputs of the Boundary-Scan cells.

The *SAMPLE* mode is a useful tool for debugging of prototypes in the development phase of a board design. In the sampling mode data are captured with the TAP Controller in its *Capture-DR* state at the rising edges of the TCK pulses. Subsequently these data can be shifted out in the *Shift-DR* state of the controller. The *PRELOAD* is selected when the user prepares an *EXTEST* by shifting in beforehand the data which must be driven out from the chip's output pins into the PCB net using the TAP controller in its *Update-DR* state.

An application of the *PRELOAD* function is the following. Directly after power-up the data in the Boundary-Scan registers at the output cells are not known. If, as a first test step, an *EXTEST* instruction is loaded into the Instruction Register, then the execution of the test may result in hazardous signals being output to the connected inputs on the net. Therefore, the Boundary-Scan registers at the output (driving) pins should be loaded beforehand with known data using the *PRELOAD* instruction.

Figures 2-25 and 2-26 show the input and output cells when the *SAMPLE* respectively the *PRELOAD* instructions is selected. The data flows in the Boundary-Scan cells are shown in bold lines.

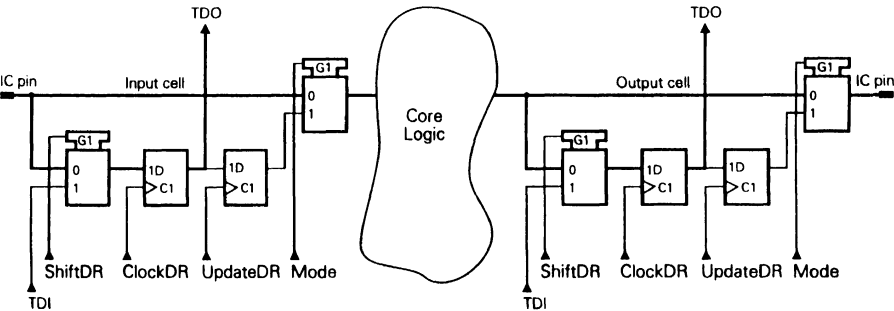


Fig. 2-25 Data flow during *SAMPLE* instruction

The EXTEST Instruction

Since the board-level interconnect testing is one of the primary reasons for introducing Boundary-Scan testing, the *EXTEST* is a mandatory instruction within the IEEE Std 1149.1. This is most important and specifically provided to allow board-level testing of *opens*, *stuck-at* or *bridging errors* etc. This instruction also allows testing of clusters of components which do not themselves incorporate

connected output pins are kept at HIGHZ. This avoids Boundary-Scan Cells at the output pins being back-driven (overdriven) with an unknown signal value, which may cause damage to the component.

The INTEST Instruction

When the *INTEST* is selected, the Boundary-Scan test substitutes testing by means of the bed-of-nails fixture and pin probing of automatic test equipment (ATE) for component (IC) testing.

This is an optional instruction which, if provided, must comply with the IEEE Std 1149.1. The *INTEST* instruction allows static (slow-speed) testing of the core logic after the component is mounted on the PCB. Following this instruction, test signals are shifted in one at a time and applied to the core logic. As explained in chapter 1, the operation resembles the *EXTEST*; only the role of the IC's output and input pins is reversed.

Figure 2-28 shows the data paths through the Boundary-Scan Cells applied at the input for the *INTEST* and shifting out the test results from the output side of the core logic.

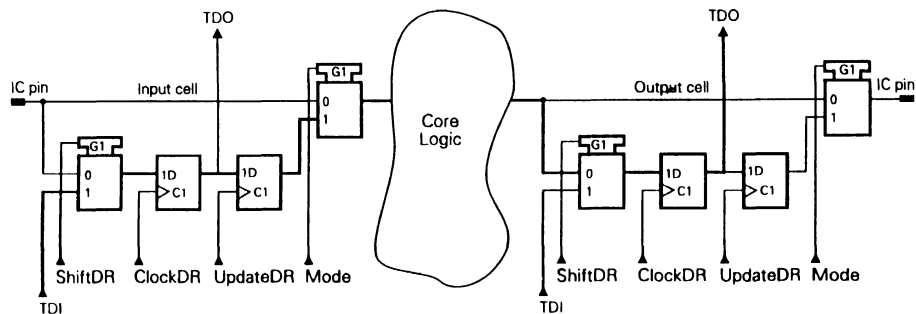


Fig. 2-28 Data flow during INTEST instruction

As stated above, the test signals are applied one at a time at the rate of TCK. This requires that the core logic can operate at this rate. For devices containing dynamic logic, such as DRAM memories, refreshment of the data cells may require a much higher frequency than can be obtained with this test method. Unless core logic provisions have been taken internally (e.g. synchronization of the system clock with TCK), this test method could be unusable for these devices. Therefore the *RUNBIST* may be best suited (see next subsection).

Between the input of the signal and the capture of the response, the core logic may receive clock signals. These clock signals may be obtained in various ways while the *INTEST* instruction is executed. Some examples are given in the description of the IEEE Std 1149.1.

The RUNBIST Instruction

The Run Built-In Self-Test is an optional instruction which, if provided, must comply with the IEEE Std 1149.1. The *RUNBIST* instruction causes the execution of a self-contained self-test of the component (IC) without the need to apply complex data patterns. With the *RUNBIST* instruction the correct operation of a component (IC) can be tested dynamically, without the need to load complex data and without the need for single-step operation (as required for the *INTEST* instruction). The IEEE Std 1149.1 requires that this self-test can only be executed with the TAP Controller in its *Run-Test/Idle* state. Only then the self-test can run to completion, provided that the controller remains long enough in the *Run-Test/Idle* state, measured for example in clock pulses applied to the core logic. The result of the self-test is captured from the selected test data register with the controller in its *Capture-DR* state, after which it can be shifted out for examination.

The timing constraints have been added to ensure that the built-in self-tests of *all* components involved are completed in one test run. If on a PCB components are mounted with different specifications of self-test duration times, then the IEEE Std 1149.1 ensures that enough time has passed for the completion of the built-in self-tests of all components.

When the self-test is running, the Boundary-Scan cells are used to hold the component's outputs to a fixed value. In this way the signals generated in the core logic during the self-test can not enter the PCB nets, so that unwanted values at the receiving ends of the nets can not damage the connected logic.

Finally it should be mentioned that, when the *RUNBIST* instruction is applied, the test results for all versions of a component must be the same.

The CLAMP Instruction

The optional *CLAMP* instruction [4] is used to control the output signal of a component to a constant level by means of a Boundary-Scan cell. In such cases the Bypass Register is connected in the TDI-TDO path on the PCB. The instruction is used for instance with cluster testing, where it can be necessary to apply 'static' guarding values to those pins of a logic circuitry which are *not* involved in a test. The required signal values are loaded together with all test stimuli, both at the start

of the test and each time a new test pattern is loaded. As such the *CLAMP* instruction does increase the test pattern, thus slightly reducing the overall test rate.

The IDCODE Instruction

If a Device Identification Register is included in a component, the *IDCODE* is forced into the Instruction Register's parallel output latches while the TAP Controller is in its *Test-Logic-Reset* state. This means of access to the Device Identification Register permits blind interrogation of components assembled onto a PCB, making it possible to determine what components are mounted on a board. The possible applications of the Device ID register are described above in the relevant subsection.

The USERCODE Instruction

The *USERCODE* must be provided by the manufacturer if the Device Identification Register is included in a component *and* the component is user-programmable. This instruction is only required if the programming can not be determined through the use of the test logic. When selected, this instruction loads the user-programmable identification code into the Device Identification Register, at a rising edge of TCK and the TAP Controller in its *Capture-DR* state.

The HIGHZ Instruction

The optional *HIGHZ* instruction [4] forces *all* output of a component to an inactive drive state, for example to a high impedance state. The application of the *HIGHZ* instruction is found in situations where besides Boundary-Scan Test, a conventional in-circuit test is still required, e.g. for testing non-BST compatible components. The in-circuit tester may drive signals back to the component's output pins where hazards may occur if its output impedance is not high.

DOCUMENTATION REQUIREMENTS

Manufacturers can claim that their components comply with the IEEE Std 1149.1 for Boundary-Scan testing. Such components must comply with all relevant specifications described in the standard. The level of conformance to the standard may vary, however, depending on the range of test operations that is supported. The claim must clearly identify the subset of the public instructions that is supported (see next table).

Table 2-3 Public Instructions

Instruction	Status
<i>BYPASS</i>	Mandatory
<i>SAMPLE/PRELOAD</i>	Mandatory
<i>EXTEST</i>	Mandatory
<i>INTEST</i>	Optional
<i>RUNBIST</i>	Optional
<i>IDCODE</i>	Optional
<i>USERCODE</i>	Optional

The minimum requirement for conformance with the IEEE Std 1149.1 is set to ensure that the IC user can perform two basic test tasks:

- 1. examine the operation of a prototype system (e.g. IC), and
- 2. test assembled products for assembly-induced defects during manufacturing.

Further it is strongly recommended that the components support either the *INTEST* or the *RUNBIST* instruction or both, in order to enable efficient and comprehensive verification of internal component operation at the board and system level.

ASIC vendors may claim conformance with the IEEE Std 1149.1 if they can prove that an interconnection of cells, if made, will produce a component that meets the requirements of the standard.

With the sole exception of the Device Identification code, each second source component must react on all public instructions in exactly the same way as the components from the prime source supplier. This ensures exchangeability of components on a PCB, which is beneficial with respect to prepared test programs in the manufacturing phase and in field service.

The operation of the component’s test logic must be fully documented if conformance is to be claimed for that component. The information, specified in the standard, must be supplied by the component supplier. The component purchaser needs this information for use in development tests, production, field service and other activities.

GLOSSARY

The following glossary of terms and abbreviations reflects terms used in this book but may not reflect terms commonly used in electronics. Terms printed in **bold** in the descriptions below are defined themselves in the glossary.

2-State Output Pin –

A device output pin at which the signal can be only at a logical high or a logical low level at any given instant.

3-State Output Pin –

A **2-state output pin** which can *also* be held in an inactive state, for example at high impedance, in which the pin's output logic can not determine the logical level at the connected external circuitry.

Aliasing Test Result –

Response of a test of a particular net where a faulty result looks the same as a fault-free result of the test on another net, so that no unique diagnosis of the fault location is possible.

ASIC Application Specific Integrated Circuit

ATE Automatic Test Equipment

ATPG Automatic Test Pattern Generator

Bidirectional Pin –

A device contact pin at which a signal can be either received from or transmitted to the connected external circuitry.

Binary Counting Sequence –

A test method at which a net to be tested is supplied with a unique sequence of test bits, such that the applied binary sequence reflects the decimal number subsequently given to each net.

BITL **BST** Integrated Testvector List

Blind Interrogation –

Retrieving information out of a component (e.g. from a device's identification register) without further knowledge of the operation of the component concerned, thus providing for a quick check.

BPV **BST** Parallel Vector

BS Boundary-Scan

BSC Boundary-Scan Cell

BSDL Boundary-Scan Description Language

BSR Boundary-Scan Register

BST Boundary-Scan Test

BTPGTM Boundary-scan Test Pattern Generator (Trade Mark of Philips Industrial Electronics B.V.)

BTSL Boundary-scan Test Specification Language

BYPASS Instruction for **BST** defined in the **IEEE Std 1149.1**.

CLAMP Optional instruction for **BST** defined in the **IEEE Std 1149.1**.

Capture	A BST operation at which data present on an input signal line is loaded into a memory element of the BSR .
Cluster	In the context used in this book, a piece of logic on a PCB surrounded by BST compatible components but which itself is not compatible to the IEEE Std 1149.1 .
COB	Chip On Board; chip directly mounted on printed circuit board, without being packaged first.
Confounding Test Result –	Response of a test of a net by which the faulty test result looks the same as a faulty result of the test on another net, so that no unique diagnosis of the fault location is possible.
Core logic	As opposed to the test logic , the functional logic performing the wanted non-test functions.
DFT	Design For Testability
DR	Data Register
DUT	Device Under Test
EDIF	Electronic Data Interchange Format; an international standard language to describe electronic circuits.
<i>EXTTEST</i>	External test, an instruction for BST defined in the IEEE Std 1149.1 .
<i>HIGHZ</i>	Optional instruction for BST defined in the IEEE Std 1149.1 .
ICT	In-Circuit Test
<i>IDCODE</i>	Identity code, an optional instruction for BST defined in the IEEE Std 1149.1 .
IEEE Std 1149.1 –	Abbreviation for 'IEEE Standard Test Access Port and Boundary-Scan Architecture' [1].
Interconnect Test –	A test for the correctness of all PCB level interconnects.
<i>INTEST</i>	Internal test, an optional instruction for BST defined in the IEEE Std 1149.1 .
IR	Instruction Register
JEDEC	Joint Electron Device Engineering Council, an international body maintaining standards on electrical and electronic devices.
JTAG	Joint Test Action Group; group of test engineers representing world-wide electronics companies preparing the BST method that resulted in the IEEE Std 1149.1 .
LSB	Least Significant Bit
MCM	Multi Chip Module
MDA	Manufacturing Defects Analyzer; tester for electronic devices (PCBs) in a manufacturing environment.
MSB	Most Significant Bit
MTBF	Mean Time Between Failures
Net	Configuration of connected-through copper tracks/wires on a PCB .

Open	A name of a fault caused by an interrupted electrical connection, for example between a component pin and the PCB track or caused by a broken copper track.
PCB	Printed Circuit Board
PLB	Programmable Logic Devices
PPM	Parts Per Million
Private Instruction –	An IEEE Std 1149.1 instruction which is solely meant for internal use by the component (IC) manufacturer.
PTH	Plated Through Hole; a PCB at which the drilled holes are internally metal plated.
PTV	Parallel Test Vector, a number of test signals which are supplied simultaneously to various test nodes.
Public Instruction –	An IEEE Std 1149.1 instruction providing the end user of a component (IC) with access to test features, documented by the component supplier.
PWB	Printed Wiring Board
<i>RUNBIST</i>	Run Built-In Self Test, an optional instruction for BST defined in the IEEE Std 1149.1 .
<i>SAMPLE/PRELOAD</i> –	Optional instruction for BST defined in the IEEE Std 1149.1 .
Scan design	A technique used in digital electronics in which shift-register paths are implemented to improve testability.
Scan path	The shift-register path as used in a scan design, e.g. a BSR chain.
Shift	A defined operation of a serial shift register (e.g. Boundary-Scan Register) at which data is shifted through the serial path.
Short	A name of a fault caused by an electrical connection between two or more signal pins or copper tracks/wires on a PCB .
Signature analysis –	A technique of compressing test (result) data into a smaller number of bits, which are subsequently compared with expected (stored) data for fault detection.
SMD	Surface Mounted Devices
SR	Shift Register
Stuck-at	An name for a fault that causes a PCB signal connection to be fixed at logical level '1' or '0', independent of the driving signals.
STV	Sequential/Serial Test Vector, applied to a single test node over a period of time.
SVF	Serial Vector Format
System logic	As opposed to the test logic , the functional logic performing the required non-test functions.
TAB	Tape Automated Bonding; technology to automatically place components on a PCB .
TAP	Test Access Port defined by IEEE Std 1149.1

TCK	Test ClocK input pin in the TAP
TDI	Test Data Input pin in the TAP
TDO	Test Data Output pin in the TAP
Test logic	As opposed to the core logic , the logic performing the required test functions, for example a Boundary-Scan Test.
TIM	Testability IMprover; BSR compiler of Philips Electronics N.V.
TMS	Test Mode Select input pin in the TAP
TPG	Test Pattern Generator
TRST*	Test ReSeT input pin (optional) in the TAP
Update	A defined operation of a serial shift register element (e.g. Boundary-Scan Cell) at which the content of the scan cell is transferred to an output element.
<i>USERCODE</i>	User identity code, an optional instruction for BST defined in the IEEE Std 1149.1 .
UUT	Unit Under Test
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuit
Virtual Access –	As opposed to physical access, a term used to indicate that Boundary-Scan Cells can be used to access board connections and through these connections the inputs and outputs of non- BST devices (clusters).
VLSI	Very Large Scale Integration
Walking Sequence –	A sequence of test patterns at which the test vectors contain only one '1' between all zeros (walking 1) or vice versa (walking '0'). This vector shifts though the scan path such that the logical value, either the '1' or the '0', is applied to all nets, one at a time.
WSI	Wafer Scale Integration; technology used to integrate electronic products on a wafer, instead of on a PCB for example.

REFERENCES

- [1] IEEE Std 1149.1-1990: IEEE Standard Test Access Port and Boundary-Scan Architecture. Published by the Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017, USA.
- [2] F. de Jong et al.: Boundary-Scan Test, Test Methodology and Fault Modelling. *Journal of Electronic Testing - Theory and Applications*, Vol. 2, No. 1 March 1991, pp. 77-88.
- [3] Joint Electron Device Engineering Council: Standard Manufacturer's Identification Code. JEDEC publication 106-A from JEDEC, 2001 Eye Street NW, Washington DC 20006, USA.
- [4] Proposed instruction as supplement to IEEE Std 1149.1: Draft P1149.1a/D9, June 26 1992.
- [5] K.D. Wagner and T.W. Williams: Enhancing Board Functional Self-Test by Concurrent Sampling. *Proc. ITC 1991*, pp. 633-640.
- [6] N. Jarwala and C.W. Yau: Achieving Board-Level BIST Using the Boundary-Scan Master. *Proc. ITC 1991*, pp. 649-658.
- [7] P. Raghavachari: Circuit Pack BIST from System to Factory - the MCERT Chip. *Proc. ITC 1991*, pp.641-648.
- [8] IEEE Std. P1149.5: Standard Backplane Module Test and Maintenance (MTM) Bus Protocol.
- [9] D. Bhavsar: An Architecture for Extending the IEEE Standard 1149.1 Test Access Port to System Backplanes. *Proc. ITC 1991*, pp.768-776.
- [10] P.A. Wyatt and J.I. Raffel: Restructable VLSI - a Demonstrated Wafer Scale Technology. *Proc. Intl. Conf. on WSI*, 1989, pp. 13-20.
- [11] D.L. Landis and P. Singh: Optimal Placement of IEEE 1149.1 Test Port and Boundary-Scan Resources for Wafer Scale Integration. *Proc. ITC 1990*, pp. 120-126.
- [12] K.E. Posse: A Design-for-Testability Architecture for Multichip Modules. *Proc. ITC 1991*, pp. 113-121.
- [13] W.C. Bruce et al.: Implementing 1149.1 on CMOS Microprocessors. *Proc. ITC 1991*, pp.879-886.
- [14] L. Whetsel, Event Qualification, a Gateway to At-Speed System Testing. *Proc. ITC 1990*, pp. 135-141.
- [15] L. Whetsel: An IEEE 1149.1 Based Logic/Signature Analyzer in a Chip. *Proc. ITC 1991*, pp. 869-878.

- [16] S.P. Moreley and R.A. Marlett: Selectable Length Partial Scan, a Method to Reduce Vector Length. Proc. ITC 1991, pp. 385-392.
- [17] B.I. Dervisoglu and G.E. Stong: DFT, Using Scanpath Techniques for Path-Delay Test and Measurement. Proc. ITC 1991, pp. 365-374.
- [18] K.P. Parker and S. Oresjo: A Language for Describing Boundary-Scan Devices. Proc. ITC 1990, pp. 222-234.
- [19] K.P. Parker and S. Oresjo: A Language for Describing Boundary-Scan Devices. Journal of Electronic Testing - Theory and Applications, Vol. 2, No. 1 March 1991, pp. 43-74.
- [20] IEEE Standard Number 1076-1987: IEEE Standard VHDL Language Reference.
- [21] D. Chiles and J. DeJaco: Using Boundary Scan Description Language in Design. Proc. ITC 1991, pp.865-868.
- [22] M. Muris: Integrating Boundary Scan Test Into an ASIC Design Flow. Proc. ITC 1990, pp. 472-477.
- [23] EDIF Version 2.0.0: EIA Interim Standard No. 44, May 1987.
- [24] R.G. Bennets: Design of Testable Logic Circuits. Addison-Wesley, 1984.
- [25] Software package from the alliance FLUKE and Philips Industrial Electronics B.V., Eindhoven, The Netherlands.
- [26] W.F. Driessen and F. de Jong: Test Interface Layer, an Application Layer for Boundary-Scan testers. Paper prepared for publication.
- [27] F. de Jong: Boundary-Scan Test Used at Board Level. Proc. ITC 1990, pp. 235-242.
- [28] P.E. Fleming, D. McClean and P. Hansen: A Format for Transportable and Reusable Scan-Based Vectors. Texas Instruments Incorporated, October 1991.
- [29] F. de Jong: Testing the Integrity of the Boundary-Scan Test Infrastructure. Proc. ITC 1991, pp. 106-112.
- [30] W.K. Kautz: Testing of Faults in Wiring Interconnects. IEEE Transactions on Computers, Vol. C-23, No. 4, April 1987, pp. 358-363.
- [31] A. Hassan et al.: Testing and Diagnosis of Interconnects Using Boundary-Scan Architecture. Proc. ITC 1988, pp. 126-137.
- [32] P. Wagner: Interconnect Testing With Boundary-Scan. Proc. ITC 1987, pp. 52-57.

- [33] N. Jarwala and C.W. Yau: A New Framework for Analyzing Test Generation and Dignosis Algoritms for Wiring Interconnects. Proc. ITC 1989, pp.63-70.
- [34] C.W. Yau and N. Jarwala: A Unified Theory for Designing Optimal Test Generation and Diagnosis Algorithms for Board Interconnects. Proc. ITC 1989, pp. 71-77.
- [35] J-C. Lien and M.A. Breuer: Maximal Diagnosis for Wiring Networks. Proc. ITC 1991, pp. 96-105.
- [36] M.L. Flichtenbaum and G.D. Robinson: Scan Test Architectures for Digital Board Testers. Proc. ITC 1990, pp. 304-310.
- [37] G.D. Robinson and J.G. Deshayes: Interconnect Testing of Boards With Partial Boundary-Scan. Proc. ITC 1990, pp. 572-581.
- [38] F. de Jong and A.J. de Wind van Wijngaarden: Memory Interconnection Test at Board Level. Paper submitted to the IEEE ITC Committee, June 1992.
- [39] D. Bhavsar: Testing Interconnections to Static RAMs. IEEE Design & Test of Computers, June 1991, pp. 63-71.
- [40] W-T Cheng et al.: Diagnosis for Wiring Interconnects. Proc. ITC 1990, pp. 565-571.
- [41] BTPG(TM), Boundary-scan Test Pattern Generator, is a Trademark of FLUKE and Philips Industrial Electronics B.V., Eindhoven, The Netherlands.
- [42] P. Hansen: Testing Conventional Logic and Memory Clusters Using Boundary-Scan Devices as Virtual ATE Channels. Proc. ITC 1989, pp. 166-173.
- [43] F.P.M. Beenker: Systematic and Structured Methods for Digital Board Testing. Proc. ITC 1985, pp. 380-385.
- [44] C.E. Stroud: Distractions in Design For Testability and Built-In Self-Test. Proc. ITC 1991, p.1112.
- [45] W. Maly: Improving the Quality of Test Education. Proc. ITC 1991, p.1119.
- [46] M.A. Breuer: Obstacles and an Approach Towards Concurrent Engineering. ITC 1990, pp. 260-261.
- [47] P.C. Maxwell: The Interaction of Test and Quality. Proc. ITC 1991, p. 1120.
- [48] K. Rose: Quality in Test Education? Proc. ITC 1991, p.1121.
- [49] C.H. House and R.L. Price: The Return Map, Tracking Product Teams. Harvard Business Review, January-February 1991.
- [50] M.E. Levit and J.A. Abraham: The Economics of Scan Design. Proc. ITC 1989, pp. 869-874.

- [51] M.G. Gallup et al.: Testability Features of the 68040. Proc. ITC 1990, pp.749-757.
- [52] J. Miles et al.: A Test Economics Model & Cost-Benefit Analysis of Boundary-Scan. Proc. European Test Conference 1991, pp. 375-384.

INDEX

- 2-net short 148
- 2-state output pin 38, 199
- 3-state output pin 39, 199
- Adaptive test algorithm 153
- Added value of test 185
- Address line faults 163
- Aliasing test result 151
- ATE 5
- Backdriving 47
- Backplane test bus 61
- Bed-of-nails fixture 4, 114
- Bidirectional pin 40, 200
- Binary counting sequence 142
- BITL 114
- Blind interrogation 41
- Board connect test 35
- Bonding
 - Pads 17
 - Wires 17
- Boundary-Scan
 - Description language 85
- Boundary-Scan Cell 5, 36, 72
- Boundary-Scan Master 54
- Boundary-Scan Register 5, 32, 35, 198
- Boundary-Scan Test
 - Basic idea 5
 - Basics 123
 - Dataflow diagram 166
 - Test algorithms 140
 - Test path 16
- BPV 115
- BSC 5, 35, 36
- BSDL 85
 - Entity 86
 - Package 94
 - Package Body 95
- BSR 5, 35
 - Compiler 103, 106
 - Library 104
 - Library elements 105
 - Technology library 104
- BST xiii, 12
 - Integrated testvector list 114
 - On MOS designs 71
 - On multichip modules 69
 - Parallel vector 114
- BTPG 166
- BTSL 110
 - Instructions 118
- BYPASS instruction 44, 207
- Bypass Register 32, 34
- Capture-DR 27
- Capture-IR 29
- Cells at 2-state output pin 38, 199
- Cells at 3-state output pin 39, 199
- Cells at bidirectional pin 40, 200
- Cells at system clock input pin 200
- Cells at system input pin 201
- Chip On Board 2
- CLAMP instruction 48, 207
- Cluster test 157
- Component pin 37
- Component-specific clock 21
- Concurrent engineering 178, 181, 187
- Concurrent sampling 51
- Confounding degree 152
- Confounding test result 151
- Control cell 39
- Cost/market model 189
- Creation process 187
- Data line faults 164
- DBM 74
- Design costs 179
- Design For Test 176, 182
- Design-specific registers 43
- Device Identification Register 202
- Device Identification Register 41
- DFT 176, 182
- Diagnostics 168
 - Representation 168
- Die 70
- Digital Bus Monitor 74
- Documentation requirements 49, 202
- Don't care bits 72
- Double-strobe flip-flop 81, 82
- Driving output 141
- EDIF netlist 108
- Education 183
- Embedded go/no-go test 60
- Event Qualification Module 77
- Exit1-DR 25
- Exit1-IR 25
- Exit2-DR 25
- Exit2-IR 25
- EXTTEST instruction 45, 208
- Fault
 - Coverage 143
 - Detection 132

- Diagnosis 132, 140, 148
- Model 140
- Modelling 132
- Multiple 139
- Open 132, 134
- Short 136
- Single-net 134
- Stuck-at 134
- Fault detection clusters 159
- General format test vector 115
- Glossary of terms 215
- Go/no-go test 60
- HIGHZ instruction 49, 209
- IDCODE instruction 49, 209
- IEEE P1149.5 60
- In-circuit emulation 52
- In-circuit testing 4
- Inactive driver state 21, 22
- Infrastructure test 123
- Initialization 23
- Instruction decoding 72
- Instruction Register 19, 30, 206
- Interconnect structures 132
- Integrity test BST chain 123
- Intermittent errors 132
- Internal clock signals 80
- INTEST instruction 47, 72, 210
- IR 19, 32
- JEDEC 42
- JETAG 174
- JTAG 175
- Latched parallel output 28
- Life cycle costs 178
- Make_BS 98
- Management aspects 173
- Manufacturing costs 179
- Manufacturing Defects Analyzer 194
- Masking data 53
- Max-independence algorithm 153
- MCM 69
- MDA 194
- Memory board testing 57
- Memory Clusters
 - Interconnect test 161, 165
- Memory testing 70
- Minimum-size sequence 142
- Multi Chip Modules 2, 69
- Multi-net faults 134
- Multiple Faults 139
- Multiple-net short 149
- Multiple-step test 152
- Non-BST logic 157
- Open faults 134
- Order independent sequence 147
- Output cell 39
- Overdriving 47
- Pad drivers 70
- Parallel Test Vectors 141
- Path-delay measurements 80
- Pattern generation 166
- Power-on-self-test 60
- Powering-up 23
- Private instructions 44
- Programmable Clock Interface 77
- Public instructions 43, 211
- Quality 185
 - Costs 179
- RAM interconnection test 162
- Reliability 183
- ROM interconnection test 165
- Run-Test/Idle 24, 27
- RUNBIST instruction 48, 210
- SAMPLE/PRELOAD instruction 44, 72, 212
- Select-DR-Scan 25
- Select-IR-Scan 25
- Sequential Test Vectors 141
- Serial Test Vectors 141
- Serial vector format 120
- Service costs 180
- Shift-DR 27
- Shift-IR 29
- Shorts 137
 - And-short 137
 - Combinations 137
 - Conventional test 160
 - Or-short 137
 - Weak short 137
- Signature analysis 53, 76
- Single-net faults 134
- Single-step test 152
- Skew errors 52
- Slot address 64
- Soldering joints 17
- State diagram 23
- State transitions 23
- Static errors 80, 132
- Structure independent sequence 147
- Stuck-at 0 134
- Stuck-at 1 134
- Stuck-open faults 134
- SVF 120
- Synchronized TCK 74
- System clock 21, 74
- System level testing 60
- System pin 37
- System pin related cells 37
- TAP 11, 19

- TAP Controller 11, 19, 22, 72, 213
- Tape Automated Bonding 2
- TCK 20, 21
- TDI 8, 21
- TDO 8, 21
- TDR 19
- Test
 - Algorithms 140
 - BST implementation 124
 - BST infrastructure 123
 - BST net integrity 127, 140, 160
 - Clusters 157
 - IC's BST logic 124
 - Infrastructure 123
 - Interactions 160
 - Interconnect 133
 - Memory clusters 161
 - Memory interconnect 161
 - Pattern 140
 - Power supply 124
 - Stuck-at 134
 - TCK net 125
 - TDI and TDO nets 125
 - TMS net 125
 - TRST* line 126
- Test Access Port 19
- Test Clock 20, 21
- Test Data Input 21
- Test Data Output 21
- Test Data Registers 19, 33, 213
- Test execution 168
- Test interface layer 109
- Test Mode Select 20, 21
- Test Reset Input 21
- Test strategy 123
- Test vector length 78
- Test-Logic-Reset 24, 26
- Tester architecture 166
- TIL 109
- TIM 103
- Timing dependency 80
- TMS 20, 21
- Trace Memory 76
- TRST* 19, 21
- Update-DR 28
- Update-IR 30
- USERCODE instruction 49, 212
- VHDL 85
- VHSIC 85
- Virtual node 16
- Wafer Scale Integration 68
- Walking one sequence 145
- Walking zero sequence 145
- WSI 68