

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <http://www.renesas.com/inquiry>.

## Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
  - “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
  - “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
  - “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

# H8SX Family

## Boundary Scan: Introduction

### Introduction

This Application Note describes the basics of boundary scan testing.

This Application Note provides an introductory level description. See the related Application Notes for use and application of specific devices.

Sections 1, 2, and 3 answer the questions what is a boundary scan and what can be done using boundary scan. Section 4 describes the structure of devices that support boundary scan testing, and section 5 describes the BSDL file that is required for boundary scan testing. Finally, section 6 presents the examples of the files and software required for boundary scan testing as reference material.

### Contents

1. What is Boundary Scan?.....	2
2. Boundary Scan Test Overview.....	4
3. Boundary Scan Examples.....	7
4. Device Mechanisms.....	10
5. BSDL File.....	33
6. Boundary Scan Test System Structure Example.....	44
7. References.....	46
8. Reference Documents.....	46

### 1. What is Boundary Scan?

Boundary scan is a function that tests (scans) the boundary between an IC's internal circuits and the external circuit it operates in. A check or test of an IC or printed circuit using boundary scan technology is called either a boundary scan test or a JTAG test.

The JTAG test was proposed in Europe in 1985 and standardized in 1990 as the IEEE 1149.1-1990 Standard Test Access Port and Boundary-Scan Architecture standard. The name of the committee at that time was JETAG (Joint European Test Action Group). After that, since countries from areas other than Europe joined the group, the "European" was removed from the name to create the current name, JTAG (Joint Test Action Group). Currently, the name of the committee, JTAG, and the formal name of the standard are used interchangeably to refer to this standard.

Checking and testing using the earlier in-circuit testing, in which a test probe is applied to the IC pins, has become difficult due to current IC integration levels requiring pin spacings that are narrower than those on test probes. (See figure 1(a).)

Boundary scan testing, however, resolves this problem. Devices that support boundary scan testing include shift registers (called boundary scan cells) in the boundary between the IC's internal and external circuits, and also provide boundary scan input and output pins. In a boundary scan test, the boundary scan cells acquire the IC pin I/O states and output the cell data through the boundary scan output pin. This allows the pin states to be ascertained. (See figure 1 (b).)

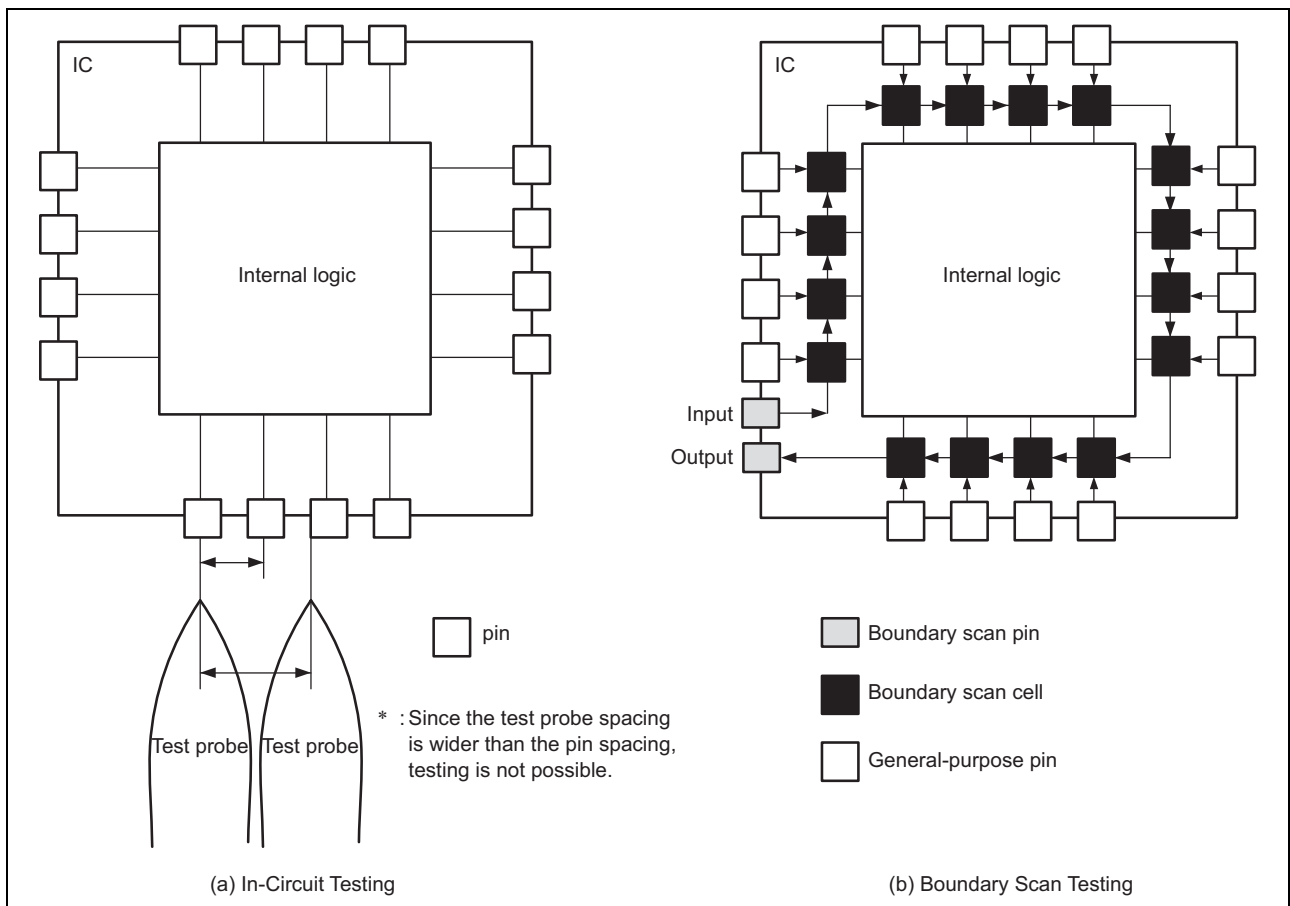


Figure 1 In-Circuit and Boundary Scan Testing

Another problem is that both the QFP (quad flat package) type (figure 2 (a)), in which pins extend from the four sides of the IC, and the BGA (ball grid array) type (figure 2 (b)), in which electrodes made of solder in the form of a small sphere are arrayed under the package, are now widely used. If one were to attempt to use an in-circuit test approach with BGA type package devices, it would be physically impossible to apply the test probes to the pins when the package is mounted since the pins are directly underneath the IC package. With boundary scan testing, however, testing is possible regardless of the form of the IC, since the IC has internal boundary scan cells.

Boundary scan testing allows the states of each pin to be investigated without influencing the IC's basic operation and it also allows IC's internal and external state testing in which the IC's basic operation is stopped.

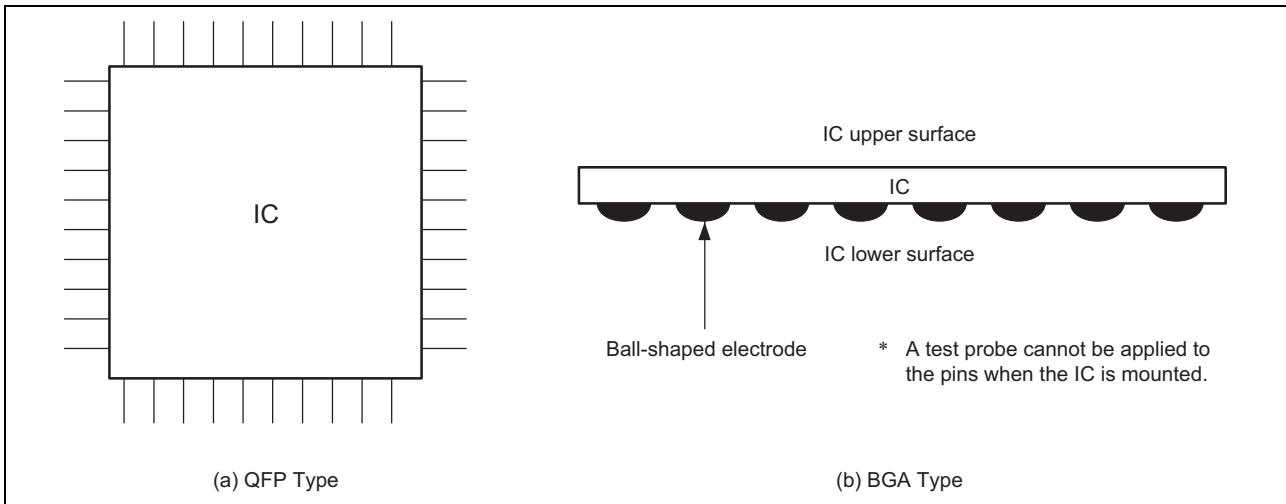


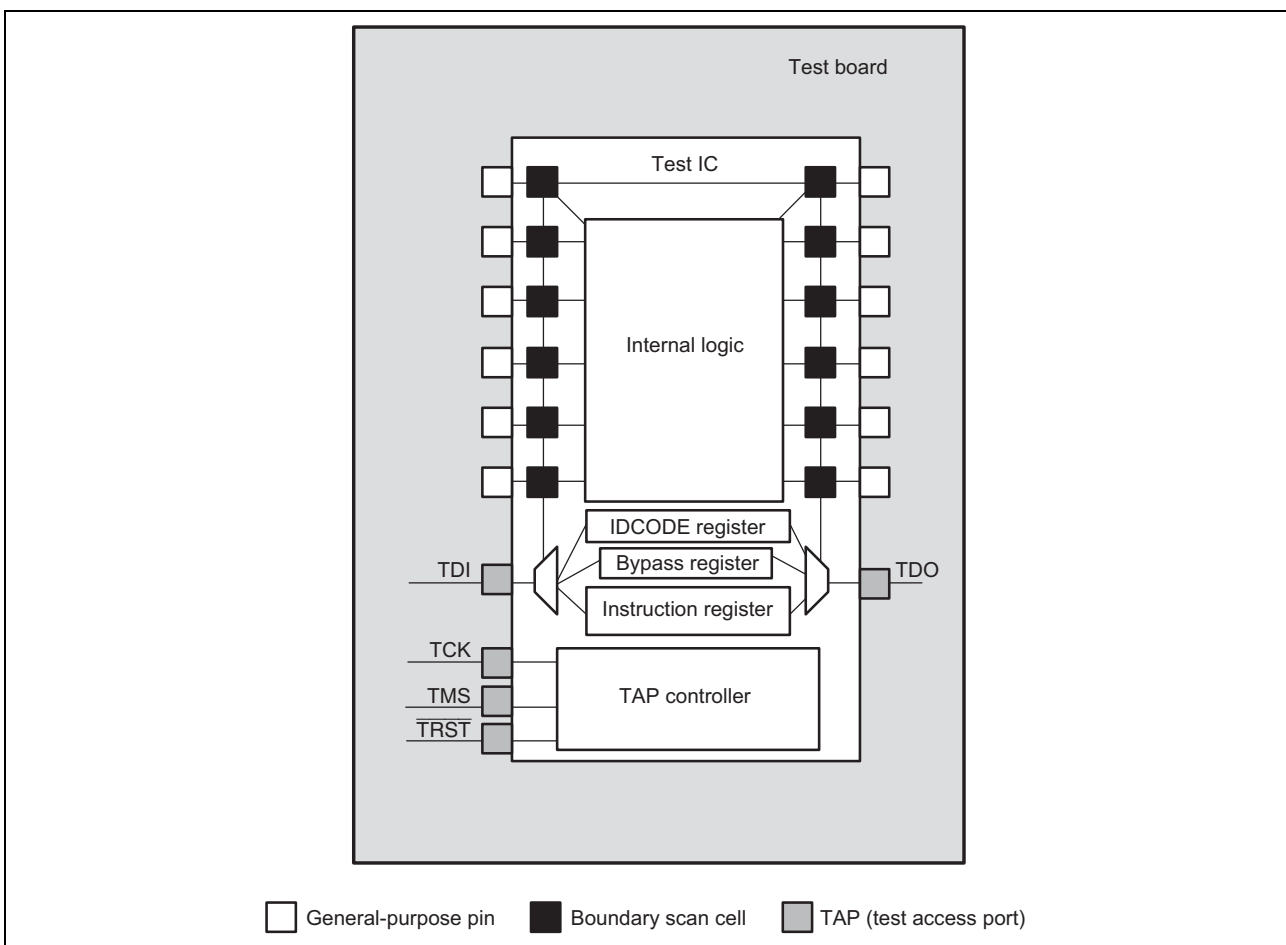
Figure 2 IC Package Types

## 2. Boundary Scan Test Overview

Boundary scan has the following features.

- Boundary scan requires only 5 additional pins
- Devices that support boundary scan include a TAP controller and various registers internally.
- The device is controlled by inputting special-purpose instructions.
- Instructions can be used to operate the device in either test mode or normal mode. In test mode, normal operation of the device is disabled and the pin states and internal logic are controlled, whereas in normal mode, the device is operated normally and the pin states and internal logic are monitored.

Figure 3 shows the structure of a boundary scan device. A boundary scan device provides five TAP (test access port) pins as external pins, and includes, independently of the internal logic (core) of the device itself, certain special-purpose circuits including a TAP controller, boundary scan cells, and an instruction register.



**Figure 3 Boundary Scan Device Structure**

Boundary scan operation is managed and controlled by a state machine called the TAP controller; the state of the TAP controller can be controlled by manipulating the TAP inputs. The TAP consists of five pins, each with a different function (table 1). The pin required for TAP controller control is the TMS pin, and the TAP controller state can be manipulated by inputting 0 or 1 to those pins. The TDI and TDO pin connection targets change according to the state of the TAP controller.

For example, when the TDI pin is connected to the instruction register, instructions can be entered and when it is connected to the boundary scan cells, the cells can be set to arbitrary values. The TDO pin operates similarly. Either the instruction register value (called the IR status word) or the boundary scan cell values can be acquired from that pin by changing the connection target. See section 4.2, TAP (Test Access Port), for details on the TAP and section 4.3, TAP Controller, for details on the TAP controller.

**Table 1 TAP Functions**

Pin	I/O	Function
TDI (Test Data In)	Input	Serial input of instructions or data
TDO (Test Data Out)	Output	Serial data output
TCK (Test Clock)	Input	Clock supply
TMS (Test Mode Select)	Input	TAP controller state control
TRST (Test Reset)	Input	TAP controller reset

Table 2 lists the registers used by boundary scan, and their functions. These registers can only be accessed by manipulating the TAP controller. While the pin states can be manipulated and monitored during a boundary scan, it is the boundary scan register that allows these functions to be implemented. The term boundary scan register refers to a shift register formed by linking together multiple boundary scan cells within the device. The boundary scan register exists in the boundary between the internal logic and the external pins, and furthermore, the boundary scan cells are connected in series. Thus input or output can be performed in a single operation from the TDI or TDO pins. See section 4.4, Registers, for details on the registers.

**Table 2 Register Types and Functions**

Register Type	Function
Instruction register	Stores and decodes instruction codes (e.g. the SAMPLE instruction).
Data register	Bypass register
	Boundary scan register
	IDCODE register
	Accepts data input to the TDI pin and bypasses it to the TDO pin.
	Stores the states of the individual pins.
	Holds the IDCODE for the device.

To control the device pin states during a boundary scan test, a special boundary scan instruction must be issued. When issuing an instruction, the TAP controller is manipulated to input the instruction code in the state where the instruction register is connected to the TDI pin. There are several types of instruction, and in addition to pin state control, there is also an instruction called IDCODE that can acquire the unique device ID. (See table 3.) Note that there are two modes for instruction operation: normal mode and test mode. In normal mode, the pin states and internal logic can be monitored in the state where the device is operated normally, and in test mode, normal device operation is stopped and the pin states and internal logic can be controlled. One typical instruction used in normal mode is the SAMPLE/PRELOAD instruction. This instruction can acquire the pin states through the boundary scan cells in normal device operating state (for a microcontroller, this would be the state where a program is being executed). One typical instruction used in test mode is the EXTEST instruction. This instruction can set the device pins to arbitrary values regardless of device operation (output of values to the pins by the internal logic is cut off). See section 4.5, Instructions, for details on the instructions.

**Table 3 Instructions and Functions**

<b>Instruction</b>	<b>Function</b>	<b>Operating Mode</b>
SAMPLE/PRELOAD	Passes pin states and internal logic data to boundary scan cells.	Normal mode
BYPASS	Bypasses input data from the TDI pin to the TDO pin.	Normal mode
EXTEST	Outputs the values of the boundary scan cells.	Test mode
IDCODE	Identifies the device IDCODE.	Normal mode
CLAMP	Outputs the values of the boundary scan cells and performs a bypass operation.	Test mode
HIGHZ	Sets all output pins to the high-impedance state and sets up a bypass.	Test mode



### 3. Boundary Scan Examples

What can be done using the boundary scan function? This section introduces two useful tests: the interconnect test and the cluster test.

#### 3.1 Interconnect Test

An interconnect test determines whether or not data is correctly transmitted from one IC to another using multiple ICs that support boundary scan testing.

Figure 4 presents a simplified image of this test. Here, a test pattern is loaded in advance into the cells in device 1, and the corresponding pin states are output with the EXTEST instruction. Then a SAMPLE/PRELOAD instruction is executed in device 2. If the correct test patterns can be verified in device 2, then we can infer that the inter-pin connections between ICs are correctly connected in the printed circuit board pattern.

If open circuits or shorts occur in the printed circuit board pattern, it will not be possible to verify the exact test pattern data in device 2.

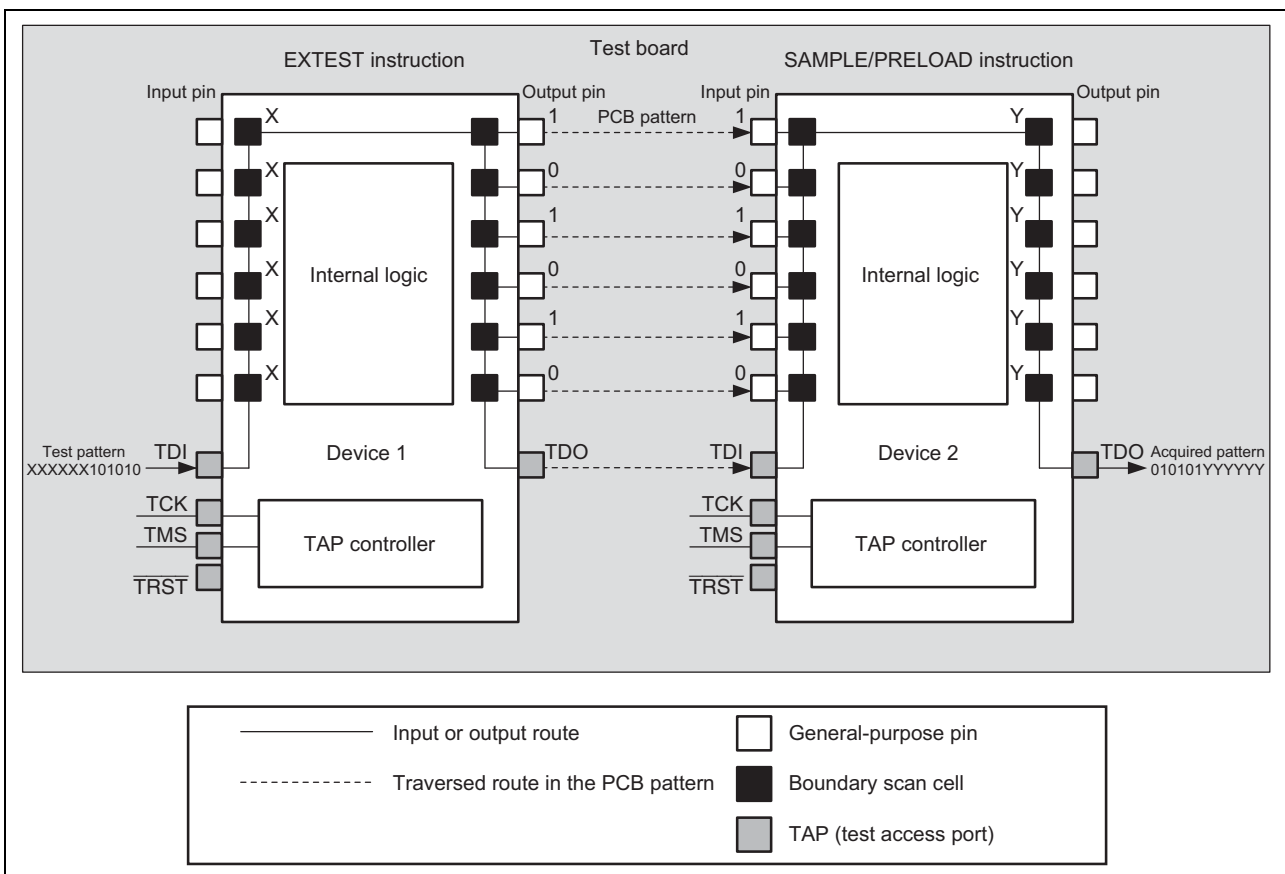


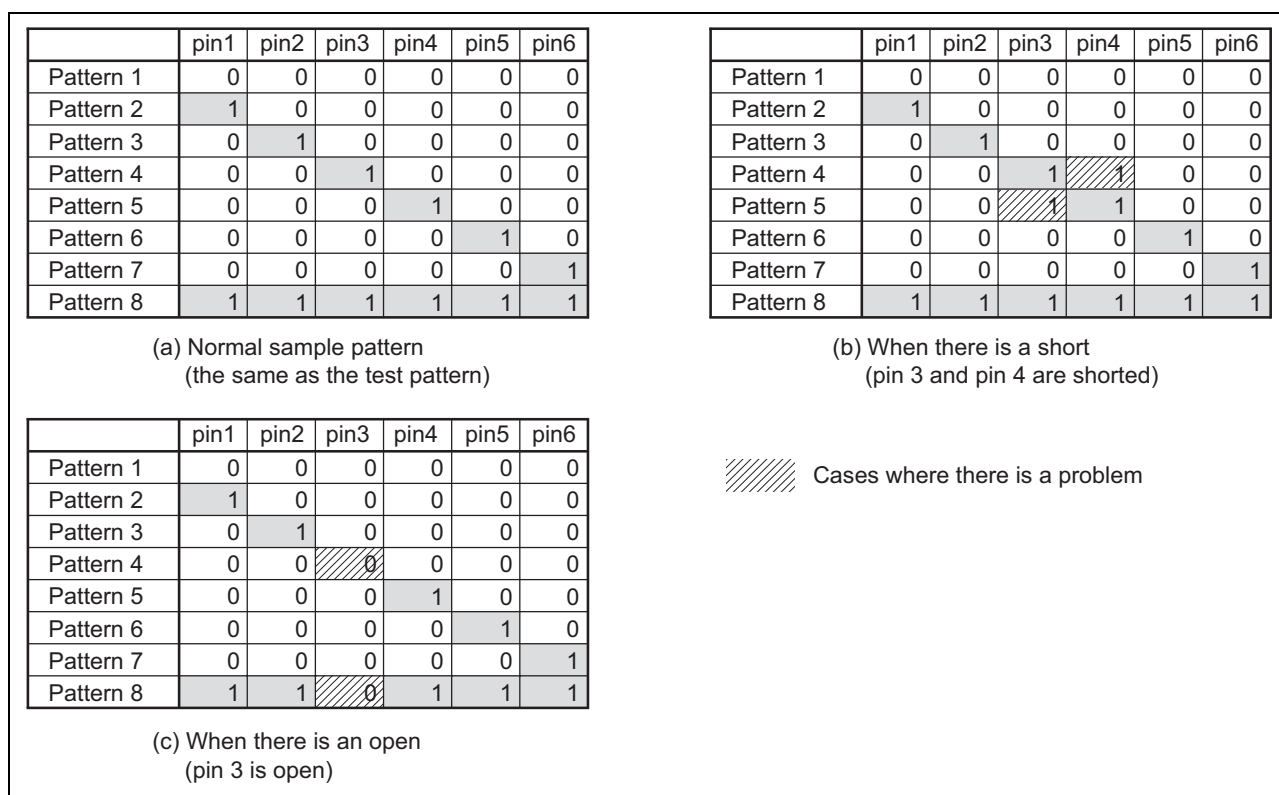
Figure 4 Interconnect Test Overview

As an example, figure 5 shows the patterns acquired when there are 8 test patterns for an interconnect test and each pattern is performed once (for a total of 8 tests) and when 6 pins are tested. In this example, pins with the same pin number are connected.

If there are no problems with connections on the printed circuit board, when the test patterns in figure 5 (a) are loaded into device 1, the same sample patterns will be acquired from the pins of device 2.

Consider, however, the case in figure 5 (b), where despite the figure 5 (a) test patterns being applied, the value 1, which differs from the logical value, is acquired both for pin 4 in pattern 4 and pin 3 in pattern 5. In this case, since the same logical value is acquired for pin 3 and pin 4 in all patterns, we can infer that pin 3 and pin 4 are shorted together on the test board.

Also consider figure 5 (c), where despite the figure 5 (a) test patterns being applied, the value "0" is acquired for pin 3 in pattern 4 and pattern 8. Here, we can infer that, since the pin 3 value for all the patterns is 0, pin 3 is open (or shorted to ground) on the printed circuit board.



**Figure 5 Sample Patterns**

### 3.2 Cluster Test

The cluster test is a method for testing the operation of devices that do not support boundary scan (including ICs, resistors, and capacitors) but that are inserted between multiple ICs that do support boundary scan on a printed circuit board. The structure of this test is the same as that of the interconnect test, but this test requires that the I/O specifications of the devices that do not support boundary scan be made explicit before the test.

In a cluster test, the test pattern is loaded into device 1 and the corresponding pin states are output with the EXTEST instruction. Then a SAMPLE/PRELOAD instruction is executed in device 2. The test pattern is transmitted from device 1 to device 2 as follows: device 1 pins -> PCB pattern -> non-boundary scan device -> PCB pattern -> device 2 pins. The way in which the test patterns are changed by correct operation of the non-boundary scan device are predicted as expected values, and if those values are verified in device 2, operation of the non-boundary scan device is judged to be normal.

Figure 6 presents an overview of this test.

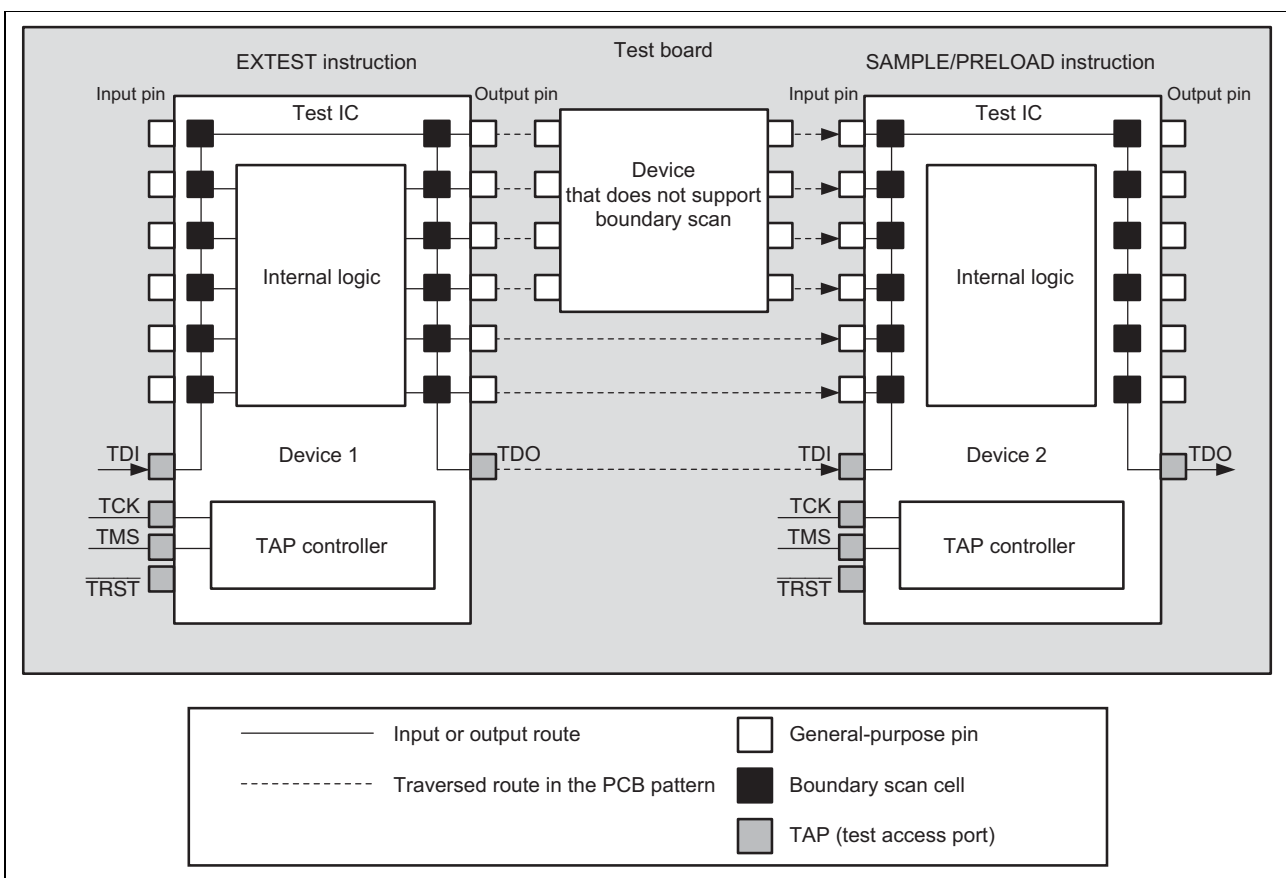


Figure 6 Cluster Test Overview

## 4. Device Mechanisms

### 4.1 Boundary Scan Device Structure

As described in section 2, Boundary Scan Test Overview, boundary scan tests are performed using the boundary scan cells, boundary scan register, TAP controller, and TAP built into the IC. The TAP is connected to a JTAG controller in the host computer. The JTAG controller has the role of transmitting and receiving the TAP signals from the test board. This section describes the TAP, TAP controller, registers, and instructions in detail.

Figure 7 shows the structure of a boundary scan test device.

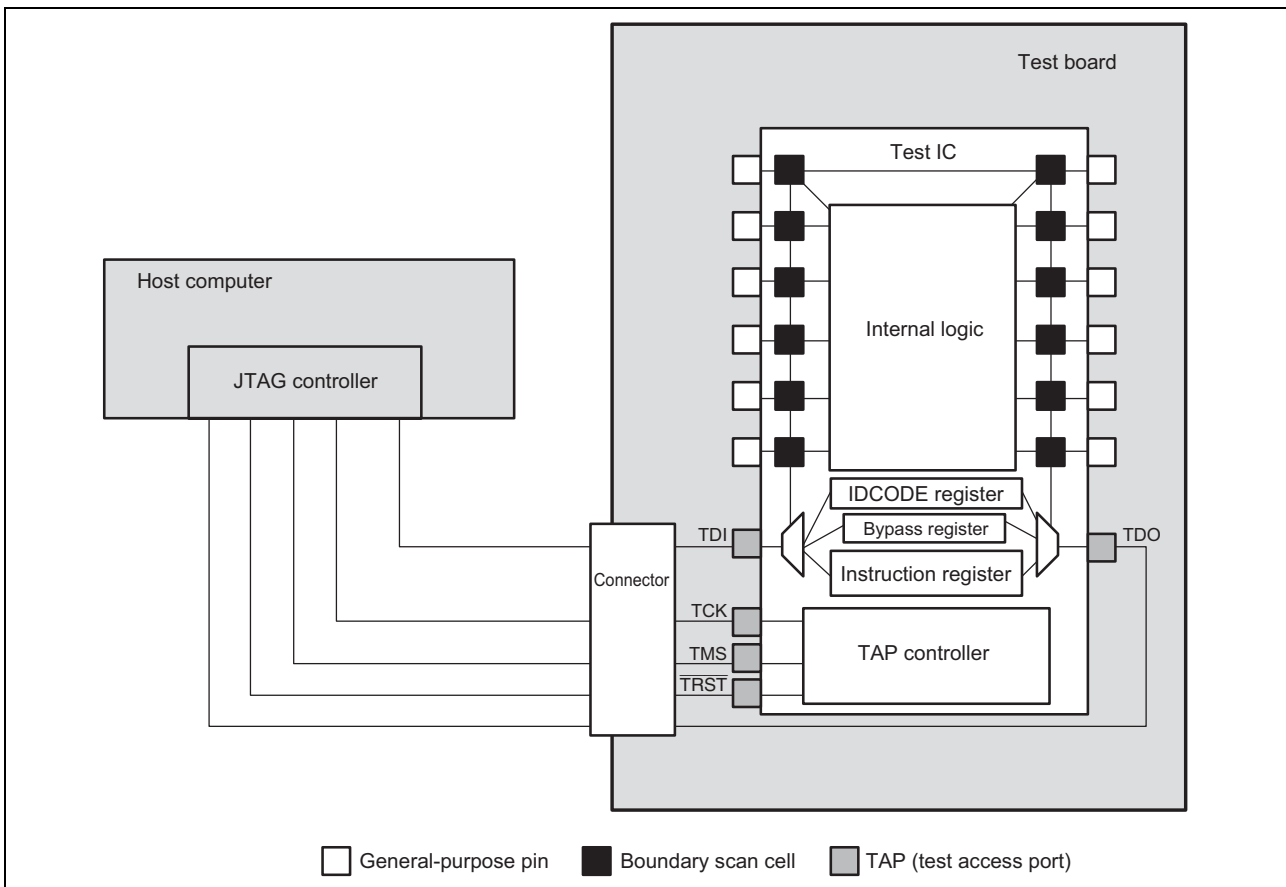


Figure 7 Boundary Scan Device Structure

### 4.2 TAP (Test Access Port)

TAP is a port used for input and output of instruction and data used for boundary scan testing. The TAP has five pins: TDI, TDO, TCK, TMS, and  $\overline{\text{TRST}}$ . These are controlled by an external host computer.

Table 4 lists the TAP functions.

**Table 4 TAP Functions**

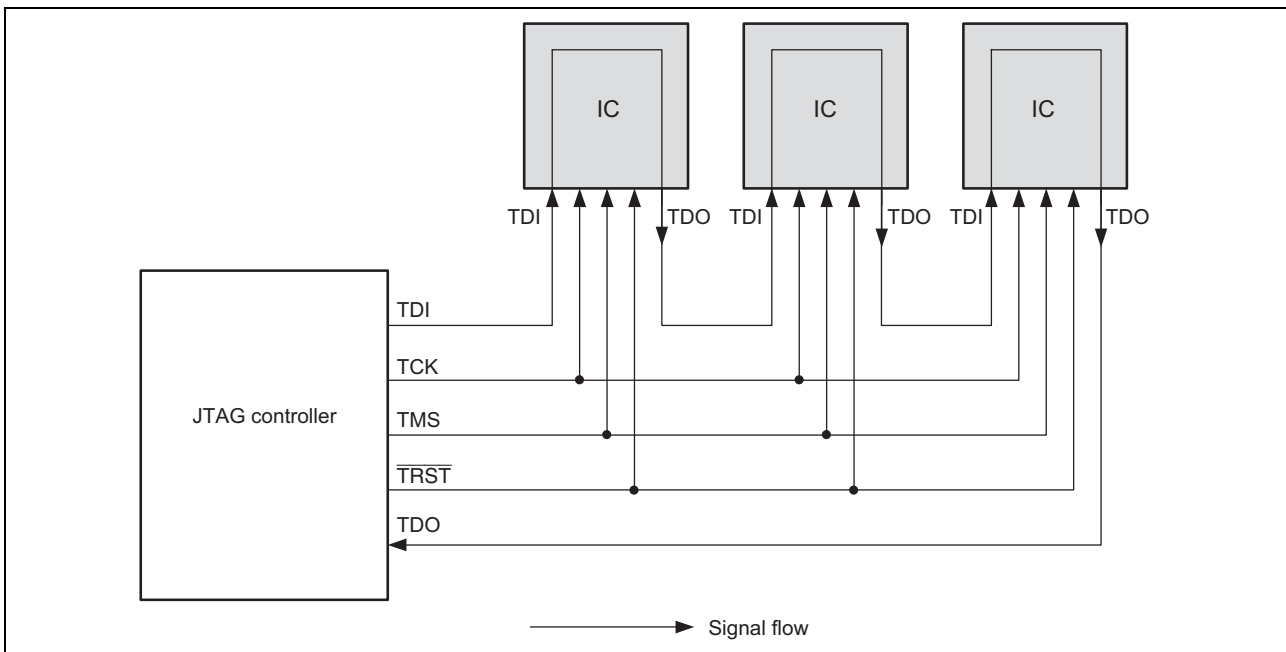
Pin	I/O	Function
TDI (Test Data In)	Input	Serial input of instructions or data
TDO (Test Data Out)	Output	Serial data output
TCK (Test Clock)	Input	Clock supply
TMS (Test Mode Select)	Input	TAP controller state control
$\overline{\text{TRST}}$ (Test Reset)	Input	TAP controller reset

The TDI, TMS, and  $\overline{\text{TRST}}$  signals are acquired in synchronization with rise of TCK, and TDO is output in synchronization with the fall of TCK. The relationship between the TCK signal and the other TAP signals is shown in the timing chart examples in section 4.5, Instructions.

Figure 8 shows a wiring example when testing is performed with multiple ICs.

The lines for the TDI and TDO pins, which perform data input and output, are connected in series between adjacent ICs. In the case shown in figure 8, the data output from the leftmost IC's TDO pin passes through the internal circuits of the center and rightmost ICs and is then input to the JTAG controller's TDO pin. This data is not modified within this path. By using this sort of wiring, it is possible to shift and output the data for multiple ICs as a single data connection.

Since the TCK, TMS, and  $\overline{\text{TRST}}$  pins are shared by all the ICs in the figure, they are connected in parallel.



**Figure 8 TAP Wiring Example for Connecting Multiple ICs**

### 4.3 TAP Controller

The TAP controller is a sequential circuit that has 16 states. The registers are controlled by these states. The transitions from one state to the next are controlled by the TAP TMS and TCK pins.

Figure 9 shows the state transition diagram for the TAP controller.

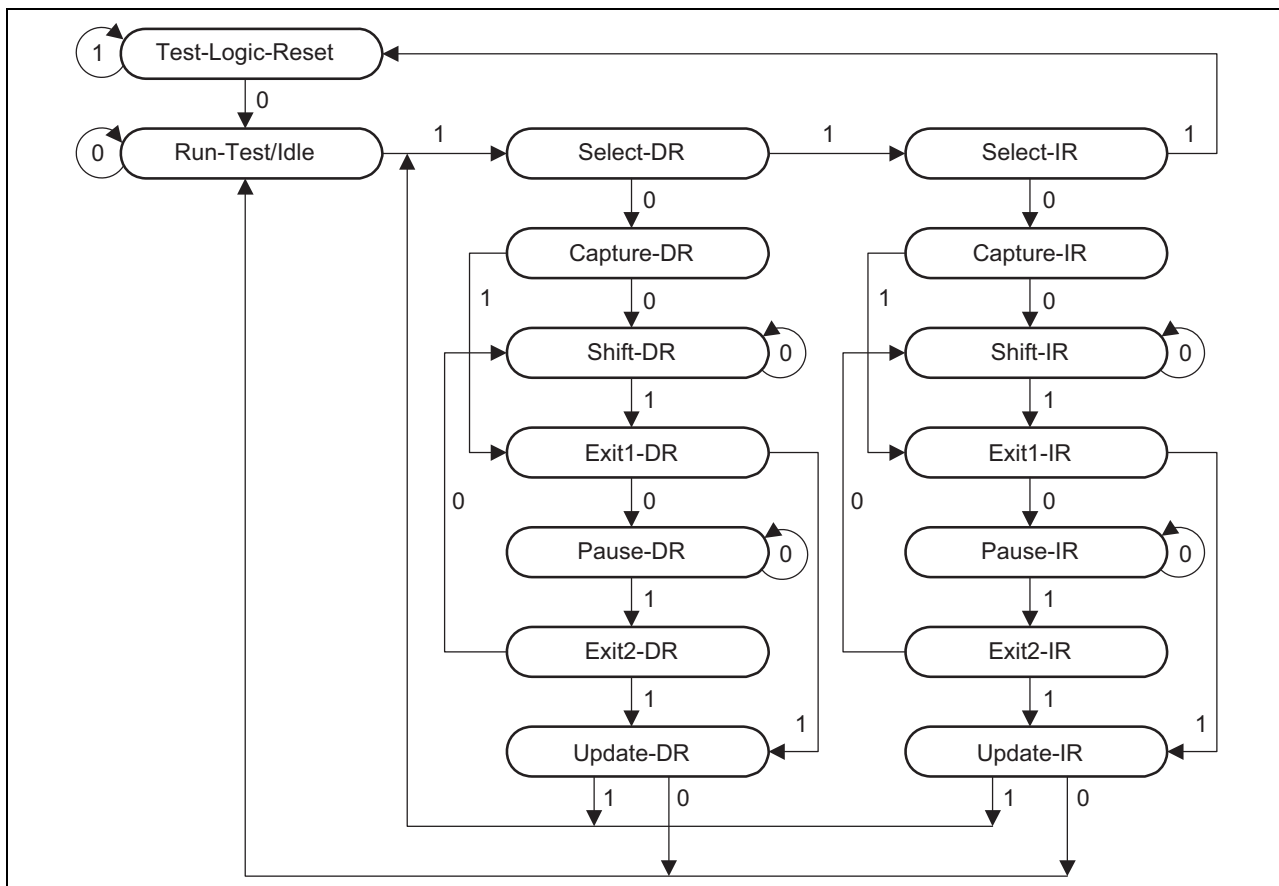


Figure 9 TAP Controller State Transition Diagram

The names enclosed in oval frames in figure 9 are the state names. The numbers 0 and 1 associated with arrows are the TMS values on the rise of TCK. The TAP states transition in the direction shown by the arrows according to the TMS value with a timing determined by the rise of TCK.

### 4.3.1 TAP Controller States

The TAP controller states consist of the Test-Logic-Reset and Run-Test/Idle states plus seven DR states and seven IR states. Here, "DR" refers to the data register, which is controlled by the instruction code. IR refers to the instruction register, which holds the instruction code. See section 4.4, Registers, for details on the registers.

Table 5 lists the 16 states of the TAP controller.

**Table 5 TAP Controller States**

State	Object Manipulated	Function
Test-Logic-Reset	—	The test logic reset state (the normal initial state)
Run-Test/Idle	—	Passing state following the Test-Logic-Reset state
Select-DR	DR	Temporary state used to select the next state
Capture-DR		Capture data in the shift register
Shift-DR		Shift the shift register data 1 bit
Exit1-DR		Temporary state used to select the next state
Pause-DR		Temporarily stop the register shift operation
Exit2-DR		Temporary state used to select the next state
Update-DR		Load the shift register data into the latch circuit
Select-IR	IR	Temporary state used to select the next state
Capture-IR		Acquire the IR status word in the shift register
Shift-IR		Shift the shift register data 1 bit
Exit1-IR		Temporary state used to select the next state
Pause-IR		Temporarily stop the register shift operation
Exit2-IR		Temporary state used to select the next state
Update-IR		Load the shift register data into the latch circuit

(1) Test-Logic-Reset state

Test-Logic-Reset is the initial state of the TAP controller. The controller enters this state at power on and when a  $\overline{\text{TRST}}$  input is applied. Also, if 5 or more TCK cycles are input with TMS remaining set to 1, the TAP controller will switch to this state. Since the same operation can be performed using TMS and TCK, the  $\overline{\text{TRST}}$  line may be left unconnected as an optional signal in some cases.

Note that an IDCODE instruction is loaded when the TAP controller enters the Test-Logic-Reset state. If the IDCODE instruction is not implemented, a BYPASS instruction will be loaded.

(2) Run-Test/Idle state

Run-Test/Idle is the idle state. The TAP controller remains in this state as long as a 0 is input to TMS. In this state, both DR and IR remain unchanged.

(3) Select state

The Select state is a temporary state used to select the next state.

(4) Capture states

The Capture states, allow data to be acquired in the shift register.

In the Capture-DR state, if a boundary scan cell is an input cell, the pin state will be acquired in the boundary scan cell, and if it is an output cell, the internal logic value will be acquired. These input and output cells are described in detail in section 4.4.2.(1).(b), Boundary Scan Register.

In the Capture-IR state, the IR status word (a fixed value) is acquired in the shift register.

(5) Shift states

In the Shift states, at the same time as shifting the data acquired in the shift register in the Capture state 1 bit at a time in the TDO direction, data is shifted in 1 bit at a time from TDI.

As long as a 0 is input to TMS, data can be shifted any number of times in the Shift state. The last bit is output when the TAP controller switches to the Exit1 state.

(6) Exit 1 states

The Exit1 states are temporary states used to select the next state.

(7) Pause states

The Pause states can temporarily stop the shift operation while a 0 is input to the TMS pin.

(8) Exit 2 states

The Exit2 states are temporary states used to select the next state.

(9) Update states

In the Update-DR state, the boundary scan call shift register value is acquired by the latch circuit.

In the Update-IR state, the shift register value is stored in the instruction register. That value is taken to be an instruction and executed.



## 4.4 Registers

There are two types of register; the instruction register and the data registers. The instruction register holds instruction codes and the data registers are controlled by those instructions.

The data registers consist of the required registers (the bypass register and the boundary scan register) and the IDCODE register, which is an optional register.

Table 6 lists the register types and their functions.

**Table 6 Register Types and Functions**

Register Type		Function
Instruction register		Stores and decodes instruction codes (e.g. the SAMPLE instruction).
Data register	Required registers	Bypass register Accepts data input to the TDI pin and bypasses it to the TDO pin.
	Optional registers	Boundary scan register Stores the states of the individual pins.
IDCODE register		Holds the IDCODE for the device.

### 4.4.1 Instruction Register

The instruction register holds an instruction code acquired from the TDI pin and is used to decode the stored instruction. Since the instruction codes differ in length and value between manufacturers, the register bit length differs with the manufacturer and device.

The instruction register is used when the TAP controller state is one of the IR states.

#### 4.4.2 Data Registers

The data registers are controlled by the instruction codes. The data registers consist of the required registers (the bypass register and the boundary scan register) and the optional IDCODE register.

The data registers can be used when the TAP controller state is one of the DR states.

For these registers, which register is used is specified with each instruction code and the specified register is connected between TDI and TDO.

##### (1) Required registers

###### (a) Bypass register

The bypass register is a 1-bit register used to bypass data between TDI and TDO.

The data input to TDI is output from TDO through the bypass register, and this allows the test path to be made shorter. This register is used to bypass unused ICs when, for example, testing multiple ICs or a circuit board.

###### (b) Boundary scan register

The boundary scan register refers to the shift register formed by connecting multiple boundary scan cells in a chain form within the device itself. The boundary scan register is placed between the general-purpose pins and the device internal logic, and holds the state of each pin. When the TAP controller state is a shift state, the values of the boundary scan cells are shifted in the direction of the TDO pin and output. The number of boundary scan cells differs with the manufacturer and device.

As shown in figure 10, each boundary scan cell consists of a 1-bit shift register and a 1-bit latch. Also, there are three types of boundary scan cell: input cells, output cells, and control cells. Input cells are cells in which the boundary scan cell input is connected to an external pin and the signal output is connected to the device internal logic. Output cells are cells in which, inversely, the boundary scan cell input is connected to the device internal logic and the signal output is connected to an external pin. Control cells are cells which enable output from an output cell.

Figure 11 shows the relationship between device pins, boundary scan cells, and internal logic. When the general-purpose pin is an input pin, an input cell is placed between the pin and the internal logic. The signal is input to the input cell from the input pin and that signal is output to the internal logic. (See figure 11 (a).)

When the general-purpose pin is an output pin, an output cell and a control cell are placed between the internal logic and the output pin. A signal is input to the output cell from the internal logic, and, only when output from the output cell is enabled by the control cell, the signal is output to the general-purpose pin. (See figure 11 (b).) It is also possible to have an output pin structure where there is no control cell and there is only an output cell. In this case the value of the output cell will always be output to the general-purpose pin.

When the general-purpose pin is a bidirectional pin, an input cell, an output cell, and a control cell are placed between the internal logic and the pin. (See figure 11 (c).) As is the case with normal input pins, the input cell inputs a signal from the general-purpose pin and outputs that signal to the internal logic. While the output cell inputs a signal from the internal logic, it can only output the signal to the general-purpose pin when the control cell enables output from the output cell.

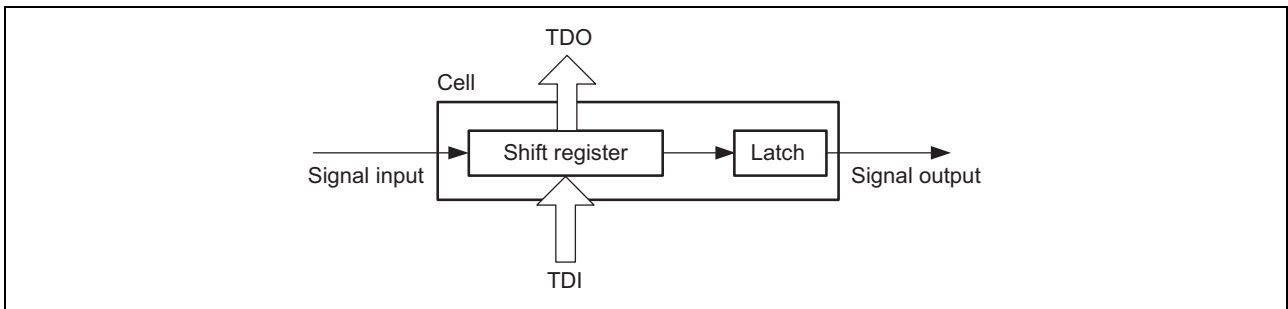


Figure 10 Cell Structure

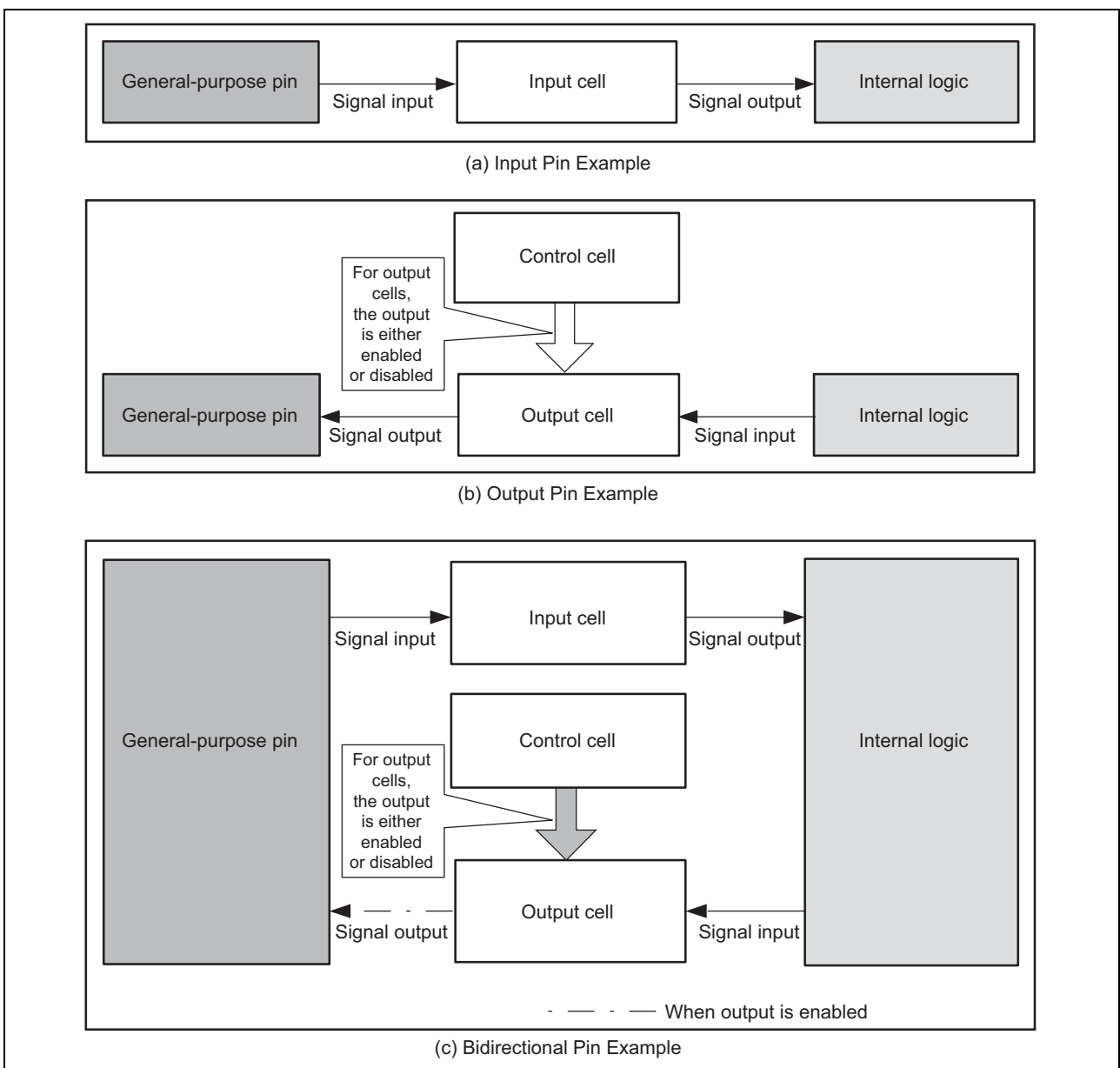


Figure 11 Cell Operation Examples

(2) Optional Registers

(a) IDCODE register

The IDCODE register stores the device IDCODE. When an IDCODE instruction is executed, the unique IDCODE for the device is output from the IDCODE register to TDO.

The IDCODE register is a 32-bit register. The upper 4 bits (bits 31 to 28) indicate the version number, the next 16 bits (bits 27 to 12) indicate the part number, and the next 11 bits (bits 11 to 1) indicate the manufacturer's ID. The low order bit is stipulated to always be set to 1 by the IEEE 1149.1 standard. (See figure 12.)

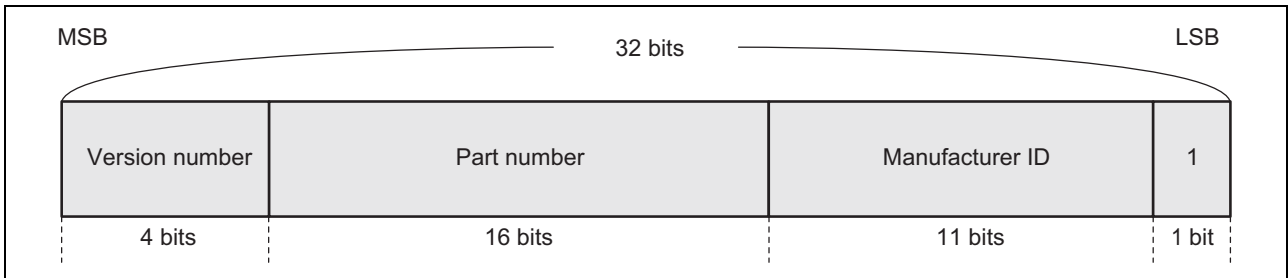


Figure 12 IDCODE Register

## 4.5 Instructions

When an instruction code is stored in the instruction register, the IC starts an operation corresponding to that instruction.

Instructions include both the required instructions, which must be implemented, and optional instructions, which do not have to be implemented. Also, there are two modes in which instructions are executed: normal mode and test mode.

In normal mode, boundary scan operations can be performed without influencing the normal operation of the device. Thus the data acquired in the boundary scan cells in the state where the device is operating can be output from the TDO pin. The normal mode instructions include the required SAMPLE/PRELOAD and BYPASS instructions and the optional IDCODE instruction.

Test mode allows tests in which the pins are isolated from the internal logic to be performed. In this mode, the internal logic cannot accept I/O from pins other than the TAP pins. This makes it possible to perform device external tests without influencing the internal logic. Test mode instructions include the required EXTEST instruction and the optional CLAMP and HIGHZ instructions.

Table 7 lists the instruction types and their functions.

**Table 7 Instruction Types and Functions**

Instruction		Function	Operating Mode
Required instruction	SAMPLE/PRELOAD	Passes pin states and internal logic data to boundary scan cells.	Normal mode
	BYPASS	Bypasses input data from the TDI pin to the TDO pin.	Normal mode
	EXTEST	Outputs the values of the boundary scan cells.	Test mode
Option instruction	IDCODE	Identifies the device IDCODE.	Normal mode
	CLAMP	Outputs the values of the boundary scan cells and performs a bypass operation.	Test mode
	HIGHZ	Sets all output pins to the high-impedance state and sets up a bypass.	Test mode

4.5.1 Required Instructions

(1) SAMPLE/PRELOAD Instruction

The SAMPLE/PRELOAD instruction acquires input data to the boundary scan cells from the IC's general-purpose pins without influencing the internal logic or external circuits. It can also acquire arbitrary data from the TDI pin and store it in the boundary scan cells. Since this is a normal mode instruction, the device can operate normally even during a boundary scan test.

The SAMPLE and PRELOAD instructions have the same instruction code, and are handled as the same SAMPLE/PRELOAD instruction. When the TAP controller is in the Capture-DR state, a sample operation is performed, and when it is in the Update-DR state, a preload operation is performed.

In the sample operation, if the boundary scan cell is an input cell, the value of the IC general-purpose pin is acquired in the boundary scan cell and if it is an output cell, the value of the internal logic is cell is acquired in the boundary scan cell.

In the preload operation, The value of the boundary scan cell is captured by the latch circuit and thus the boundary scan cell value is held fixed.

Figure 13 shows an overview of the SAMPLE/PRELOAD instruction operation.

In the SAMPLE/PRELOAD instruction, first the TAP controller switches to the Capture-DR state. Then, if the boundary scan cell is an input cell, the value of the IC general-purpose pin is acquired in the boundary scan cell, and if it is an output cell, the value of the internal logic is cell is acquired in the boundary scan cell. (See figure 13 (a).) To load an arbitrary test pattern, or if you want to verify the states of the boundary scan cells, switch to the Shift-DR state and shift the boundary scan cells as much as required to output those values from the TDO pin. (See figure 13 (b).) Finally, switch to the Update-DR state and lock the boundary scan cell values by acquiring the boundary scan cell values in the latch circuits. (See figure 13 (c).)

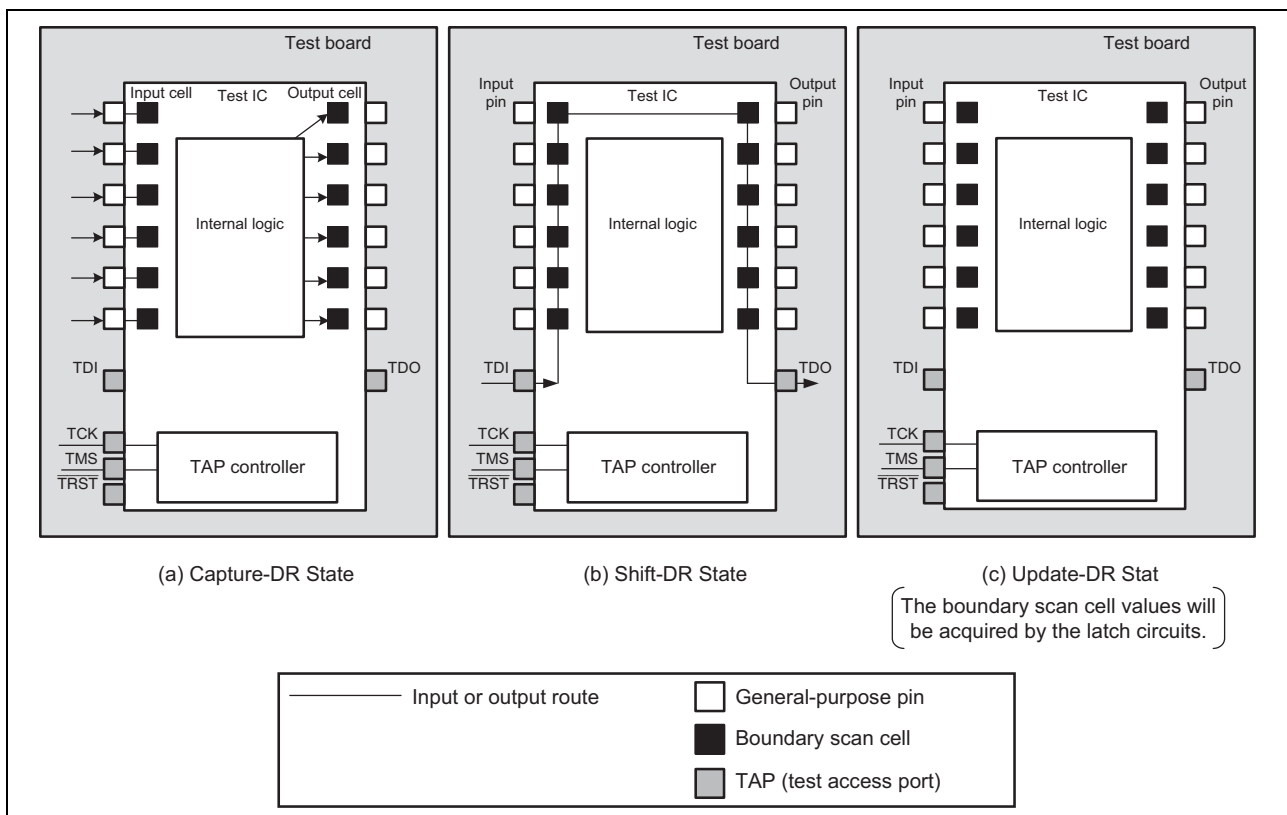


Figure 13 SAMPLE/PRELOAD Instruction

Figure 14 shows the timing chart for the SAMPLE/PRELOAD instruction.

In the timing chart, the device operation can be described from the states of four TAP pins: TCK, TMS, TDO, and TDI. The TAP control state switches as shown in figure 9 in section 4.3, TAP Controller, according to the value of TMS on the rise of TCK.

The default state for the four TAP pins is 1.

The sections A, B, and C marked at the top of figure 14 divide the chart into each of the operations of the boundary scan test. In section A, the operation that switches the TAP controller to its initial state is performed. In section B, the operation up to the point that the instruction is issued by the instruction register is performed. In section C, the instruction issued in section B is executed.

This timing chart is for the case where there are 8 bits in the instruction register and there are 15 boundary scan cells. The "X" marks in the figure indicate arbitrary values. Similar timing charts apply to the other instructions described below.

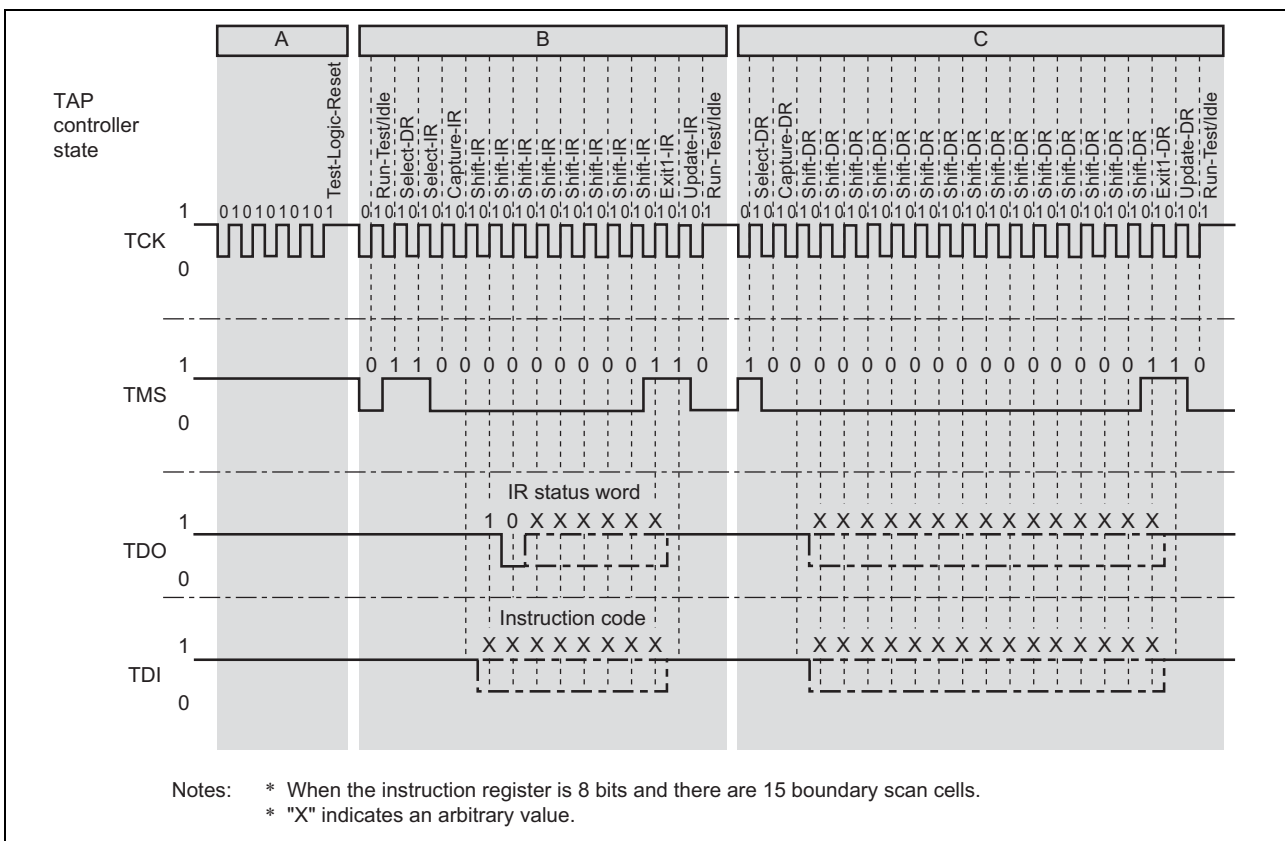


Figure 14 SAMPLE/PRELOAD Instruction Timing Chart

In the following, we describe the issuing and execution of a SAMPLE/PRELOAD instruction according to this timing chart.

In section A, the TCK pin signal rises five times while the TMS pin is 1. As a result of this, the TAP controller is guaranteed to enter the Test-Logic-Reset state on the fifth rise of TCK. Even if the controller switches to the Test-Logic-Reset state before this, it will loop in the Test-Logic-Reset state. Since section A is provided to initialize the controller state, it may be omitted if it is not necessary to initialize the controller state.

In section B, the TAP controller state is switched through the following states by the TMS pin value and the rise of the TCK pin: Run-Test/Idle state → Select-DR state → Select-IR state → Capture-IR state → Shift-IR state (8 cycles) → Exit1-IR state → Update-IR state → Run-Test/Idle state. The SAMPLE/PRELOAD instruction is executed as a result of these state transitions.

In the Capture-IR state in this transition sequence, the IR status word is captured from the instruction register.

In the Shift-IR state, one bit of the IR status word is output from the TDO pin in synchronization with the fall of TCK, and one bit of the SAMPLE/PRELOAD instruction is input from TDI. These operations are repeated for the number of bits in the instruction register (in this case, 8 times). The last bits input and output from the TDI and TDO pins are input or output on the fall of TCK when leaving the Shift-IR state.

In the Update-IR state, the SAMPLE/PRELOAD instruction stored in the instruction register is decoded and the boundary scan register is connected to TDI and TDO.

In section C, the TAP controller state is switched through the following states: Run-Test/Idle state → Select-DR state → Capture-DR state → Shift-DR state (15 cycles) → Exit1-DR state → Update-DR state → Run-Test/Idle state. The sample and preload operations are performed as a result of these state transitions.

In the Capture-DR state, the values of the general-purpose input pins and the values of the internal logic are stored in the input cells and output cells, respectively. This is the sample operation.

Next, in the Shift-DR state, the data values to be stored in the boundary scan cells (15 values in this example) are shifted in from the TDI pin.

In the Update-DR state, the boundary scan cell values are acquired by the latch circuits. This is the preload operation.

Note that it is possible to continuously acquire the input pin and internal logic values to the boundary scan cells by performing sample/preload operations by iterating the C section.



(2) BYPASS Instruction

The BYPASS instruction bypasses data from TDI to TDO using the 1-bit bypass register inserted between TDI and TDO. This allows test paths not related to the test to be shortened when multiple devices are used. Since this is a normal mode instruction, the device can operate normally even during a boundary scan test. Furthermore, registers other than the bypass register are not affected by this operation.

Figure 15 shows an overview of the BYPASS instruction operation.

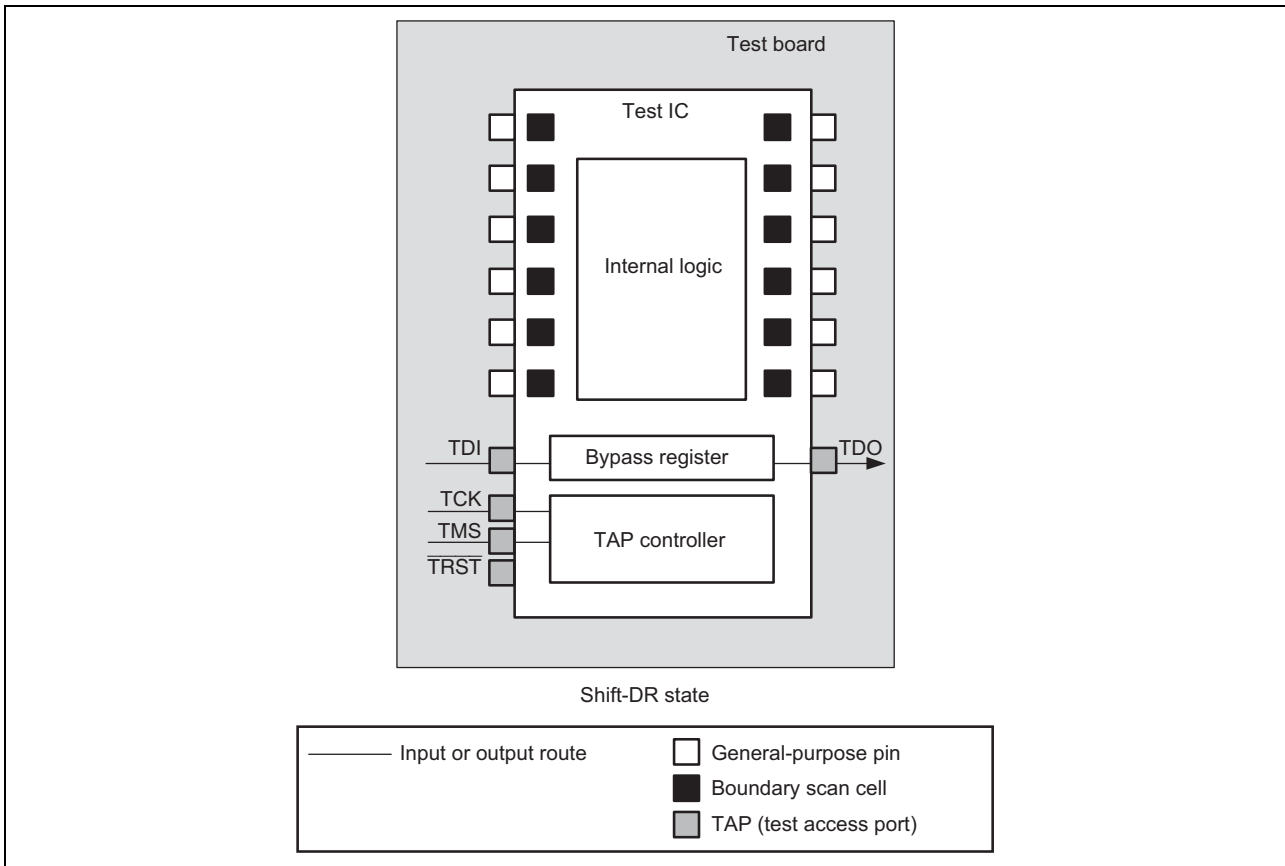
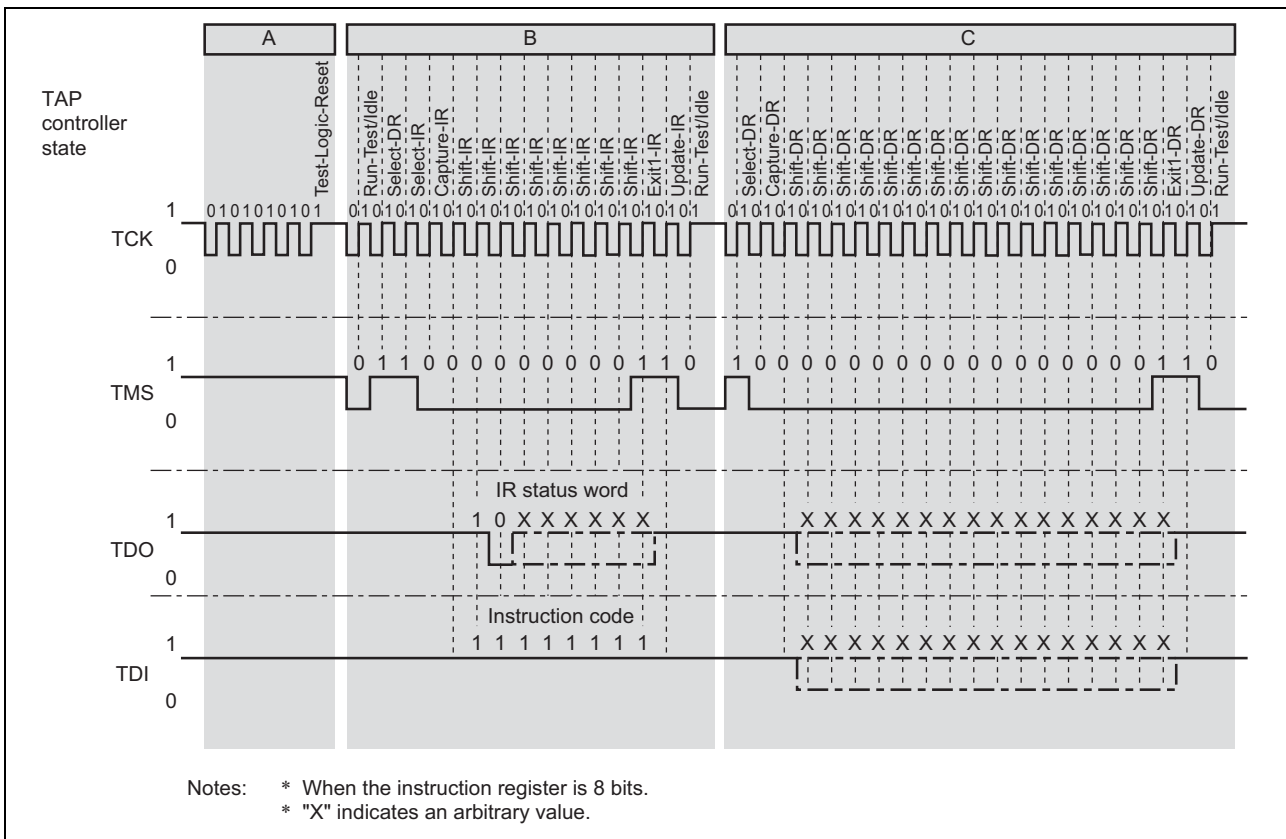


Figure 15 BYPASS Instruction

Figure 16 shows the timing chart for the BYPASS instruction.



**Figure 16 BYPASS Instruction Timing Chart**

The operation in section A initializes the TAP controller state.

In section B, the BYPASS instruction code is stored in the instruction register and decoded, and the bypass register is connected to TDI and TDO. Note that although the BYPASS instruction is stipulated to be all ones, regardless of the length of the register, in addition, other codes can also be set.

In section C, the TAP controller state is switched through the following states: Run-Test/Idle state → Select-DR state → Capture-DR state → Shift-DR state (any number of cycles) → Exit1-DR state → Update-DR state → Run-Test/Idle state. The bypass operation is performed as a result of these state transitions.

Since shifting data from TDI to TDO through the 1-bit bypass register is the main function of the BYPASS instruction, the operations in the Capture-DR and Update-DR states have no meaning.

In the Shift-DR state in this state transition sequence, the data to be bypassed is input to the bypass register from TDI one bit at a time on the rise of TCK, and the bypass register data is output from TDO one bit at a time.

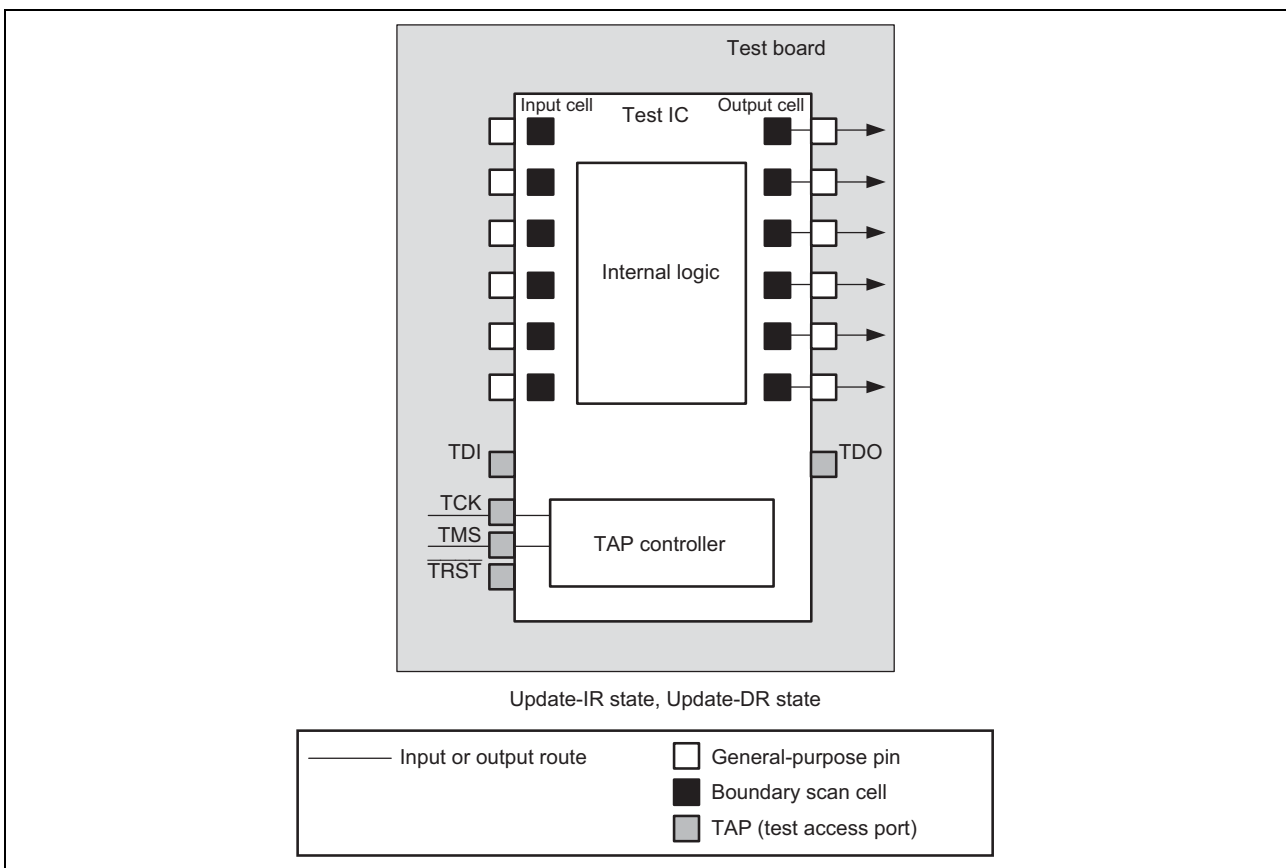
**(3) EXTEST instruction**

The EXTEST instruction is provided for testing IC external connections (printed circuit board or solder connectivity). This instruction is executed with the internal logic isolated. Since this is a test mode instruction, the devices in the boundary scan test do not operate normally.

Figure 17 shows an overview of the EXTEST instruction operation.

To execute an EXTEST instruction, data is loaded into the boundary scan cells in advance using the SAMPLE/PRELOAD instruction. Then the EXTEST instruction is executed and data is output from the IC general-purpose output pins. The EXTEST operation is performed from the point the transition sequence passes through the Update-IR state.

It is also possible to modify and output the contents of the boundary scan cells, consecutively after executing an EXTEST instruction, by switching the TAP controller through the following states: the Capture-DR state, the Shift-DR state, and the Update-DR state. In this case, the output update is performed in the Update-DR state.



**Figure 17 EXTEST Instruction**

Figure 18 shows the timing chart for the EXTEST instruction.

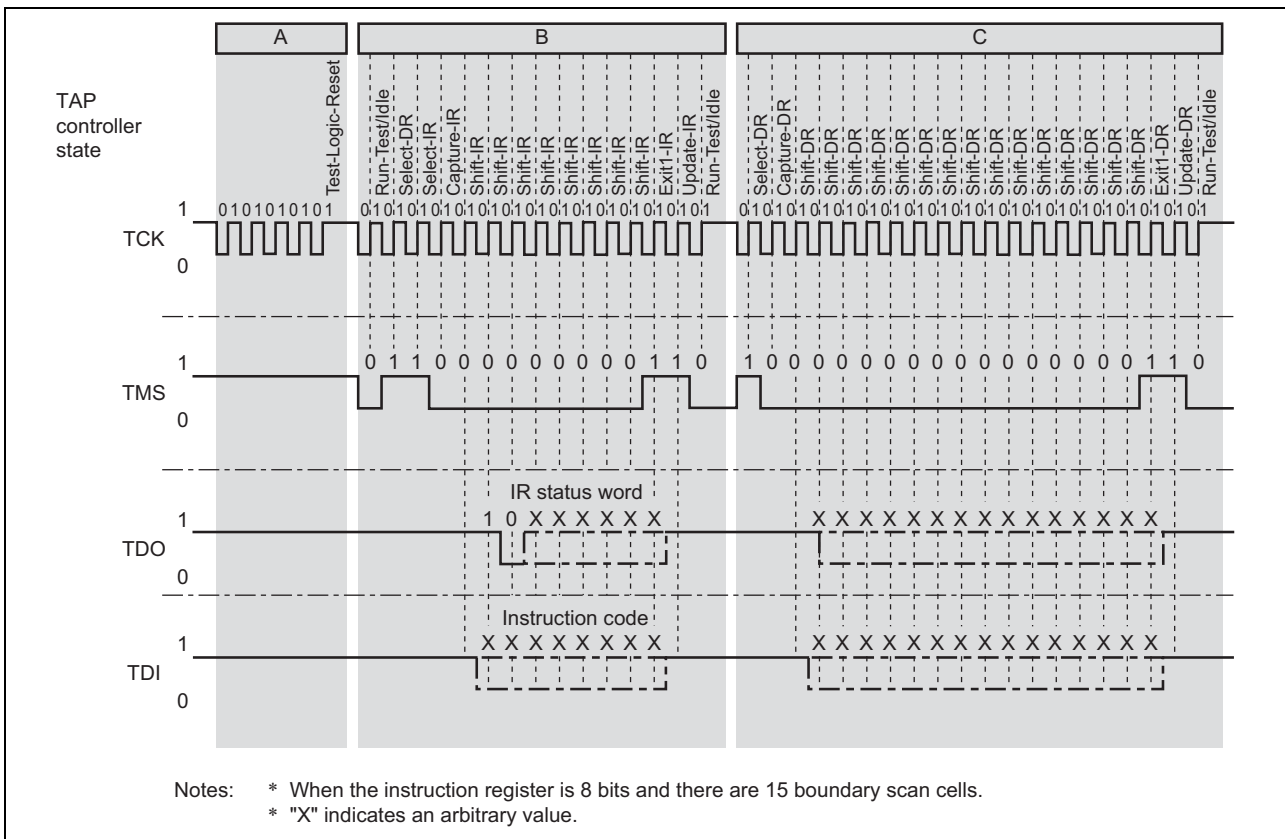


Figure 18 EXTEST Instruction Timing Chart

The operation in section A initializes the TAP controller state.

In section B, the EXTEST instruction code is stored in the instruction register and decoded, and TDI and TDO are connected to the boundary scan register. At the point the EXTEST instruction is issued, the boundary scan cell values stored in advance with the SAMPLE/PRELOAD instruction are output from the IC general-purpose output pins (this is the EXTEST operation). The EXTEST operation continues until the EXTEST instruction is cancelled.

In section C, the TAP controller state is switched through the following states: Run-Test/Idle state → Select-DR state → Capture-DR state → Shift-DR state (15 cycles) → Exit1-DR state → Update-DR state → Run-Test/Idle state. These state transitions can reload the boundary scan cell values and update the values output from the general-purpose output pins.

In the Capture-DR state during these transitions, the general-purpose input pin values and the internal logic values are stored in the input cells and output cells, respectively.

Next, in the Shift-DR state, just the number of data values to be stored in the boundary scan cells are shifted in from the TDI pin.

In the Update-DR state, the boundary scan cell values are acquired by the latch circuits. The general-purpose output pin output values are updated at the point the transition sequence passes through the Update-IR state.

Note that it is possible to acquire the input pin and internal logic values to the boundary scan cells while continuing to perform the EXTEST operation by iterating the C section.

4.5.2 Optional Instructions

(1) IDCODE instruction

The IDCODE instruction determines the IDCODE for the device. Since the IDCODE instruction is an optional instruction, it can only be used with ICs that include an IDCODE register. Since this is a normal mode instruction, the device can operate normally even during this boundary scan test operation.

Figure 19 shows an overview of the IDCODE instruction operation.

When an IDCODE instruction is executed, the IDCODE is acquired from the IDCODE register in the Capture-DR state, and it is output from TDO in the Shift-DR state. The IDCODE can be acquired by iterating the Shift-DR state 32 times (the number of bits in IDCODE).

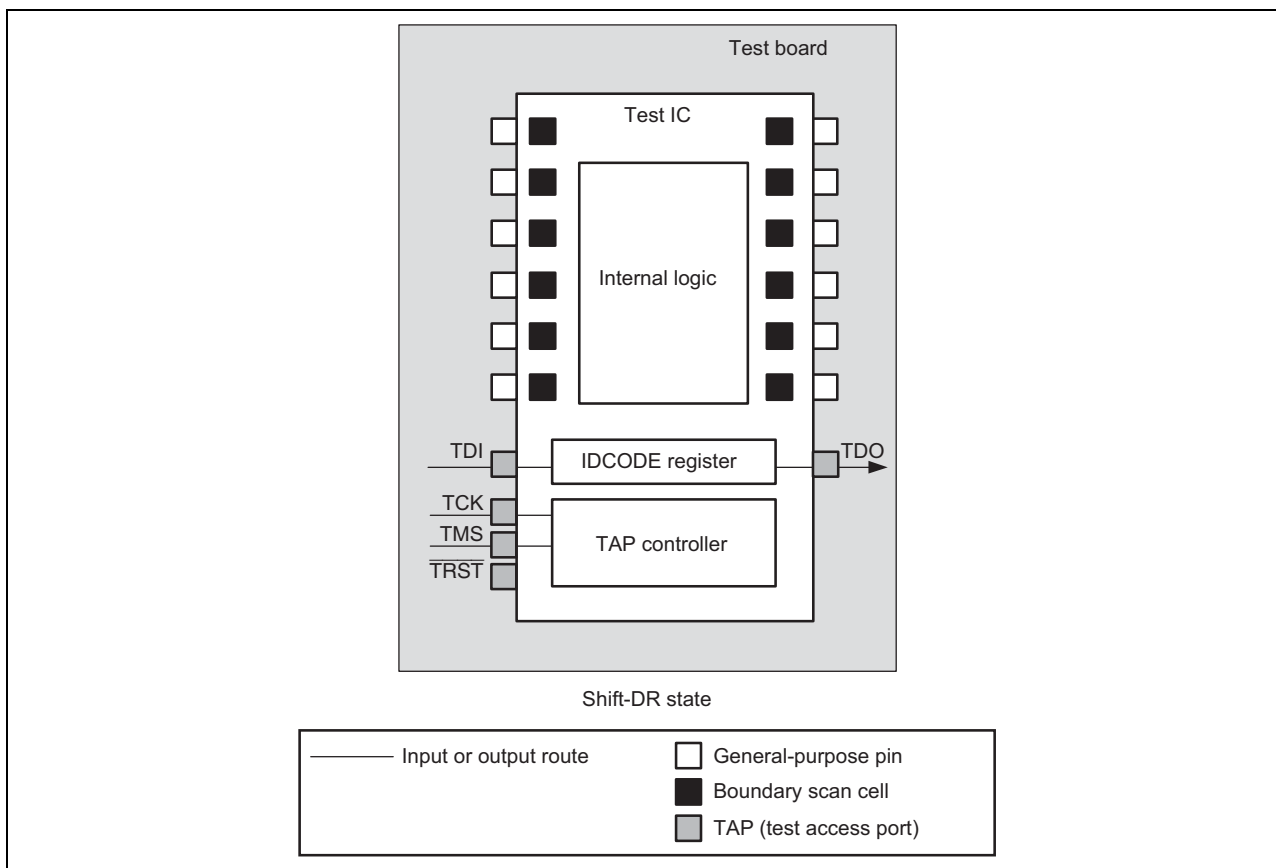
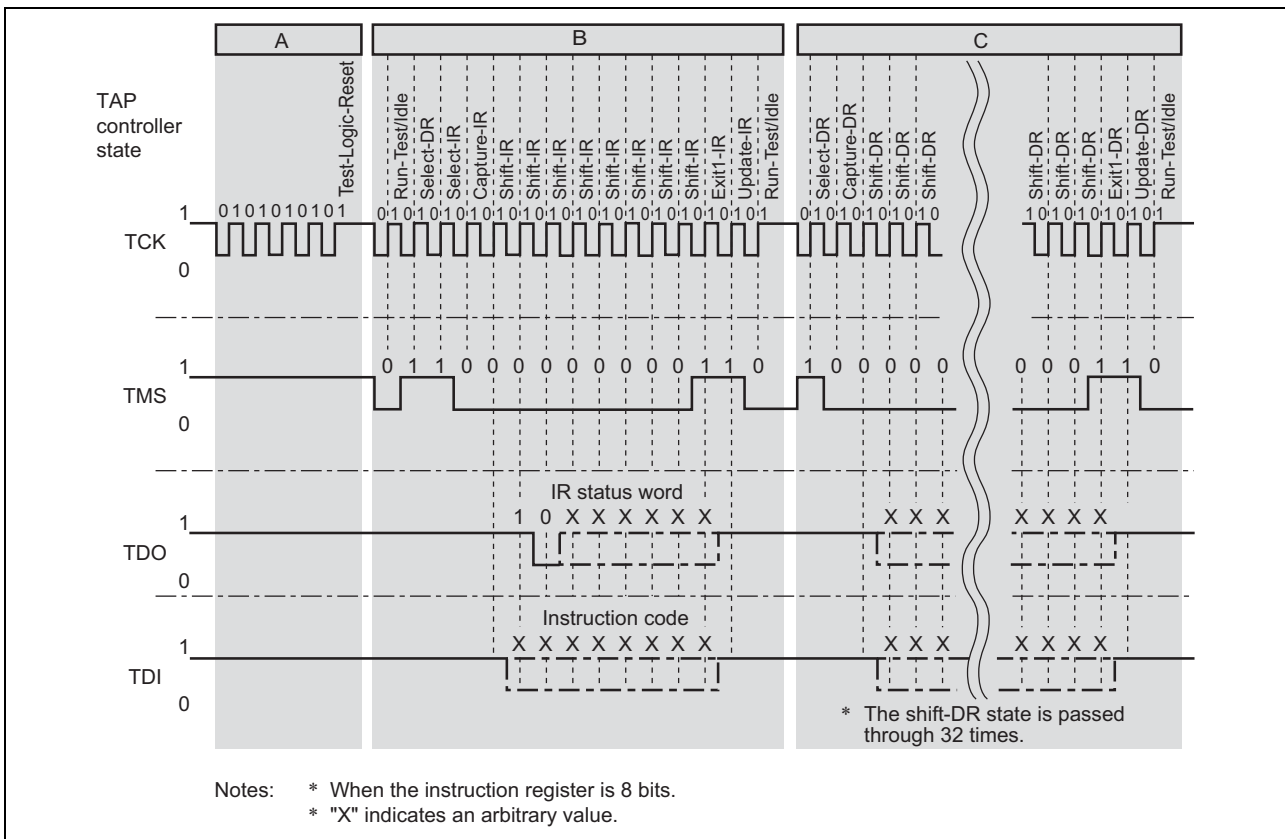


Figure 19 IDCODE Instruction

Figure 20 shows the timing chart for the IDCODE instruction.



**Figure 20 IDCODE Instruction Timing Chart**

The operation in section A initializes the TAP controller state.

In section B, the IDCODE instruction code is stored in the instruction register and decoded, and TDI and TDO are connected to the IDCODE register.

In section C, the TAP controller state is switched through the following states: Run-Test/Idle state → Select-DR state → Capture-DR state → Shift-DR state (32 cycles) → Exit1-DR state → Update-DR state → Run-Test/Idle state.

In the Capture-DR state during these transitions, the 32-bit IDCODE unique to the device is stored in the shift register from the IDCODE register.

Next, IDCODE is output from TDO one bit at a time in the Shift-DR state.

There are no special operations in the Update-DR state.

(2) CLAMP instruction

The CLAMP instruction outputs the values of the boundary scan cells to the general-purpose pins and furthermore performs a bypass. Since this is a test mode instruction, the devices in the boundary scan test do not operate normally.

Normally, ICs that are not the object of the test are bypassed to shorten the test path. There are, however, cases where it is desirable to hold the general-purpose output pins of ICs that are the object of a bypass fixed at specific levels. In such cases, since the BYPASS instruction is a normal mode instruction, the values of the general-purpose output pins may change. The CLAMP instruction can perform a bypass operation in which IC operation is stopped and the general-purpose output pin values do not change.

Figure 21 shows an overview of the CLAMP instruction operation.

With the CLAMP instruction, first, values are stored in the boundary scan cells using the SAMPLE/PRELOAD instruction. After that, a CLAMP instruction is issued and a bypass operation is performed while the pin states are output from the general-purpose output pins.

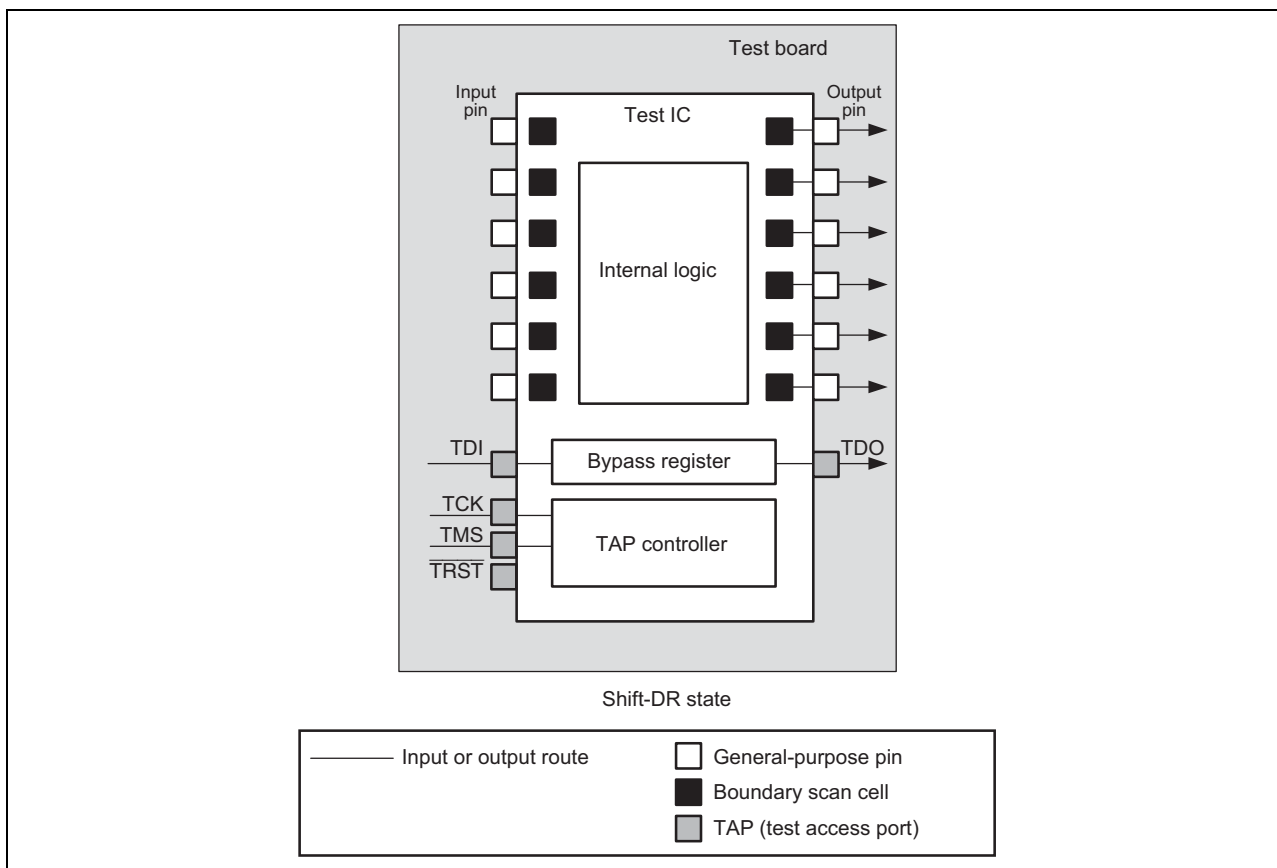


Figure 21 CLAMP Instruction

Figure 22 shows the timing chart for the CLAMP instruction.

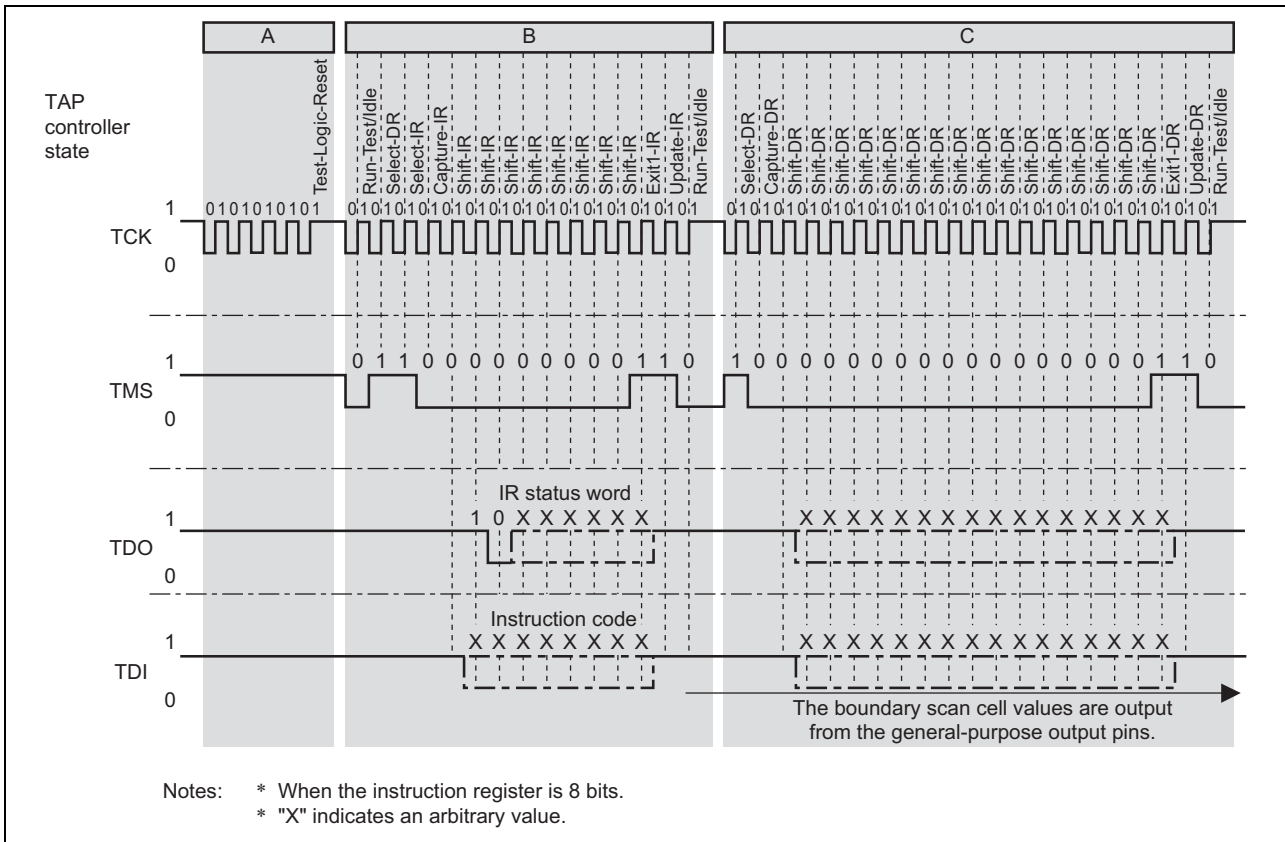


Figure 22 CLAMP Instruction Timing Chart

The operation in section A initializes the TAP controller state.

In section B, the CLAMP instruction code is stored in the instruction register and decoded, and TDI and TDO are connected to the bypass register. At the point the CLAMP instruction is issued, the boundary scan cell values stored in advance with the SAMPLE/PRELOAD instruction are output from the general-purpose output pins.

In section C, the TAP controller state is switched through the following states: Run-Test/Idle state → Select-DR state → Capture-DR state → Shift-DR state (an arbitrary number of cycles) → Exit1-DR state → Update-DR state → Run-Test/Idle state.

Since shifting data from TDI to TDO through the 1-bit bypass register is the main function of the CLAMP instruction, the operations in the Capture-DR and Update-DR states have no meaning.

In the Shift-DR state in this state transition sequence, the data to be bypassed is input to the bypass register from TDI one bit at a time on the rise of TCK, and the bypass register data is output from TDO one bit at a time.



(3) HIGHZ Instruction

The HIGHZ instruction bypasses data from the TDI to TDO pin while setting the general-purpose output pins corresponding to the boundary scan to the high-impedance state. This instruction is used to use the bypass function to shorten the test path without influencing other devices by setting the pins to the high-impedance state. Since this is a test mode instruction, the devices in the boundary scan test do not operate normally.

Figure 23 shows an overview of the HIGHZ instruction operation.

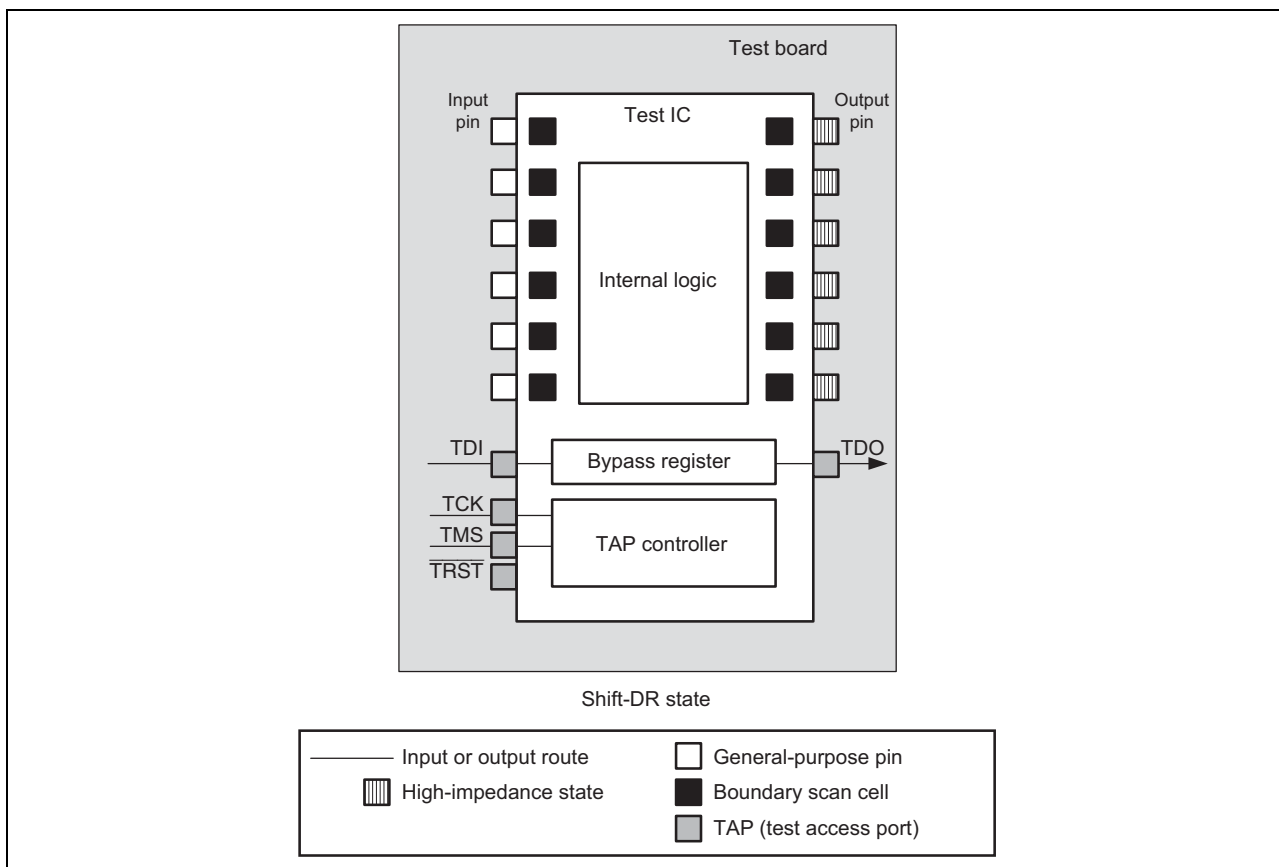


Figure 23 HIGHZ Instruction

Figure 24 shows the timing chart for the HIGHZ instruction.

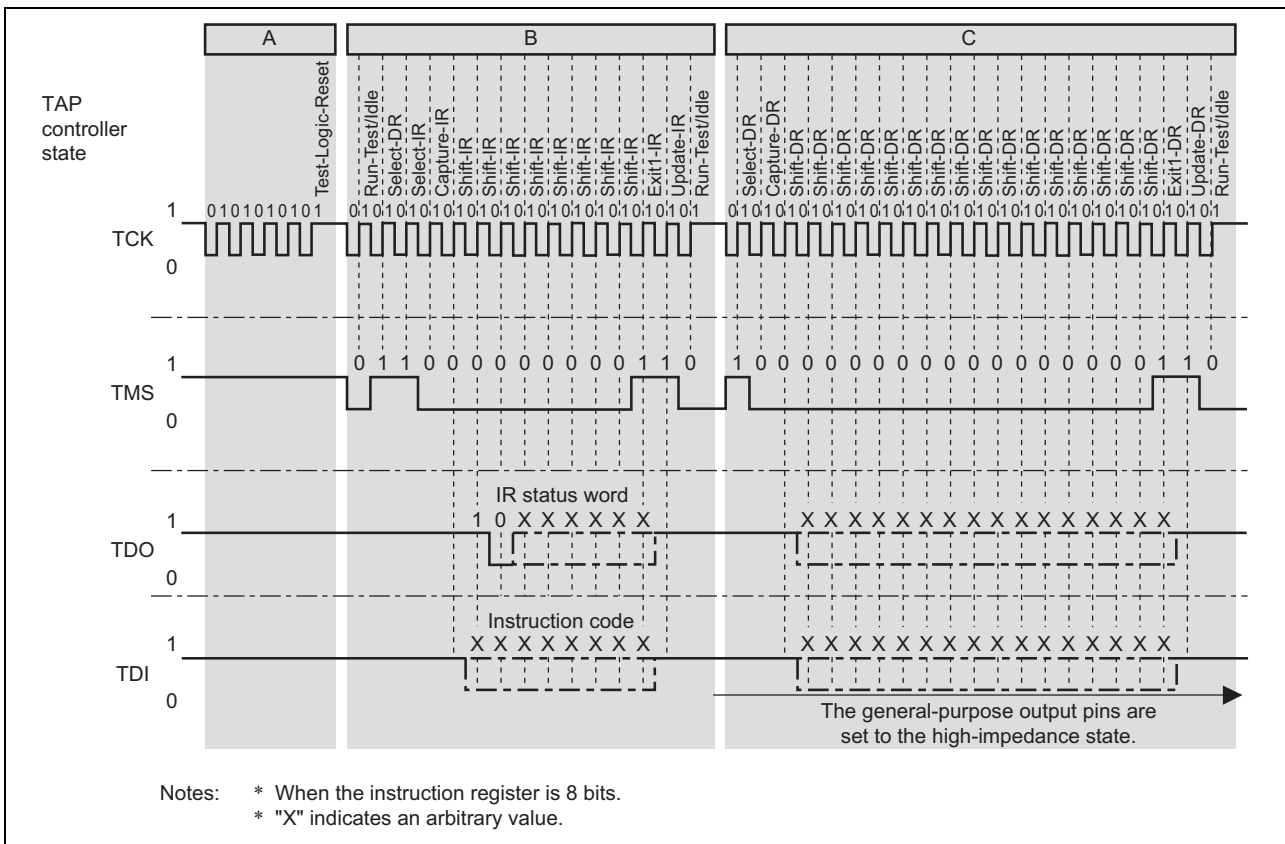


Figure 24 HIGHZ Instruction Timing Chart

The operation in section A initializes the TAP controller state.

In section B, the HIGHZ instruction code is stored in the instruction register and decoded, and TDI and TDO are connected to the bypass register. At the point the HIGHZ instruction is issued, the general-purpose output pins are set to the high-impedance state.

In section C, the TAP controller state is switched through the following states: Run-Test/Idle state → Select-DR state → Capture-DR state → Shift-DR state (an arbitrary number of cycles) → Exit1-DR state → Update-DR state → Run-Test/Idle state.

Since shifting data from TDI to TDO through the 1-bit bypass register is the main function of the HIGHZ instruction, the operations in the Capture-DR and Update-DR states have no meaning.

In the Shift-DR state in this state transition sequence, the data to be bypassed is input to the bypass register from TDI one bit at a time on the rise of TCK, and the bypass register data is output from TDO one bit at a time.

## **5. BSDL File**

This section briefly describes the BSDL (Boundary Scan Description Language) file, which is required for boundary scan testing. Refer to this section when creating an actual system.

BSDL was proposed by the United States Department of Defense and standardized as IEEE 1149.1b. BSDL is based on the hardware description language VHDL (Very High Speed Integrated Circuits Hardware Description Language). The information required for JTAG to access the ICs under test is coded in BSDL.

BSDL files are provided by the manufacturers of ICs that support boundary scan.

### **5.1 Supported Characters and Symbols**

#### **5.1.1 Supported Characters and Symbols**

The following characters and symbols may be used in BSDL files. When a Japanese system is used, all characters used must be half-width characters.

Letters (Upper and lower case are not distinguished.)

Numbers (0 to 9)

Special characters (" , & , ' , ( , ) , \* , - , : , ; , < , = , > , and \_ )

Delimiters (space, tab, newline, newpage (FF, or form feed), and linefeed)

#### **5.1.2 Words**

Words used in BSDL are formed from letters, numbers, and the underscore symbol (\_). There is no limit on the number of characters in a word. The underbar character may not be used as the last character in a word.

#### **5.1.3 Comments**

The remaining part of a line may be made a comment by inserting "--" (two hyphens) before the comment. The text following the "--" does not take part in the actual content coded by the file.

#### **5.1.4 Character strings**

Character strings are enclosed in double quotation marks ("). Character strings can be concatenated with ampersands (&).

### 5.1.5 Reserved words

Both BSDL and VHDL have reserved words. These reserved words may not be used by the user for other purposes. However, they may be used in comments.

#### (1) BSDL reserved words

The following are reserved words in BSDL. These are provided here as examples of the reserved words in BSDL.

"EXTEST"	"HIGHZ"	"CLAMP"	"SAMPLE"
"ID_CODE"	"INTEST"	"RUNBIST"	"BYPASS" "USER_CODE"
"CLOCK"	"BOUNDARY"	"BOUNDARY_LENGTH"	"CAPTURES"
"AT_PINS"	"BIDIR"	"BIDIR_IN"	"BIDIR_OUT" "BORTH"
"STD_1149_*"	"OUTPUT2"	"OUTPUT3"	"ID_CODE_REGISTER"
"OBSERVING"	"OBSERVE_ONLY"	"INTERNAL"	"CONTROL" "CONTROLR"
"BOUNDARY_REGISTER"		"BSCAN_INST"	"BSDL_EXTENSION"
"PHYSICAL_PIN_MAP"		"PI"	"PIN_MAP" "PINM_MAP_STRING"
"PO"		"PORT_GROUPING"	"COMPLIANCE_PATTERN"
"COMPORNENNT_CONFORMANCE"		"TAP_SCAN_IN"	"X"
"Z"	"ZERO"	"ONE"	

"BC\_0" "BC\_1" "BC\_2" "BC\_3" "BC\_4" "BC\_5" "BC\_6" "BC\_7" "BC\_8" "BC\_9" "BC\_10"  
 "BC\_11" "BC\_12" "BC\_13" "BC\_14" "BC\_15" "BC\_16" "BC\_17" "BC\_18" "BC\_19" "BC\_20"  
 "BC\_21" "BC\_22" "BC\_23" "BC\_24" "BC\_25" "BC\_26" "BC\_27" "BC\_28" "BC\_29" "BC\_30"

#### (2) VHDL reserved words

The following are reserved words in VHDL. These are provided here as examples of the reserved words in VHDL.

"ABS"	"ACCES"	"AFTER"	"all"	"AND"	"attribute"	"bit"
"BLOCK"	"body"	"buffer"	"BUS"	"CASE"	"COMPONENT"	"constant"
"ELSE"	"end"	"entity"	"EXIT"	"FALSE"	"FILE"	"FOR"
"FUNCTION"	"generic"	"GROUP"	"IF"	"in"	"INTERNAL"	"inout"
"is"	"LIBRARY"	"linkage"	"LITERAL"	"MAP"	"NAND"	"HEW"
"NEXT"	"NOR"	"NOT"	"NULL"	"of"	"out"	"package"
"port"	"PROCESS"	"REGISTER"	"record"	"RETURN"	"SELECT"	"signal"
"string"	"to"	"true"	"type"	"use"	"WAIT"	"WHEN"
"WHILE"	"WITH"	"XNOR"	"XOR"			

## 5.2 Elements

BSDL files contain the following elements.

Entity description

Parameter description

Logical port description

Use statements

Standard version specification

Pin mapping(s)

TAP signal definitions

(Compliance statement)

Instruction register description

(Optional register definitions)

(Register access description)

Boundary scan register description

The elements shown in parentheses above are specified when needed; not specifying these elements will not result in syntax errors. All of these items, including the ones shown in parentheses, must be specified in the order shown above.

### 5.2.1 Entity description

The entity description attaches an arbitrary name to the entity. Following a statement of the form "entity <entity name> is", the elements other than the entity statement itself are entered, and finally the description is closed with a statement of the form "end <entity name>".

Example:

```
entity SAMPLE_DEVICE is
    <Items other than the entity statement (such as parameter descriptions)>
end SAMPLE_DEVICE
```

In the above example, the entity name is set to be SAMPLE\_DEVICE.

### 5.2.2 Parameter description

The parameter description assigns physical port names to logical ports. This allocation is performed using the package definition, if there is one.

Example:

```
generic (PHYSICAL_PIN_MAP: string:= "SAMPLE_PACK");
```

In the above example, the package name is set to SAMPLE\_PACK.

### 5.2.3 Logical port description

The logical port description sets the pin names and their types. While the pin order is arbitrary, all pins must be set.

There are four pin types: in (input), out (output), inout (bidirectional input and output pins), and linkage (pins other than digital signal pins, including power, analog signal, ground, and no-connect pins).

Example:

```
port(
  PX1: in      bit;
  PX2: linkage bit
);
```

The example above sets the pin names to be PX1 and PX2.

### 5.2.4 Use statement

The use statement specifies the VHDL package standard to be used.

The notation "STD\_1149\_1\_1994" specifies to the IEEE 1149.1-1994 standard.

Example:

```
use STD_1149_1_1994.all;
```

### 5.2.5 Standard version specification

The standard version specification specifies the version number of the standard that the device supports.

The notation "STD\_1149\_1\_1993" specifies to the IEEE 1149.1-1993 standard.

Example:

```
attribute COMPONENT_CONFORMANCE of SAMPLE_DEVICE: entity is "STD_1149_1_1993";
```

### 5.2.6 Pin mapping

The pin mapping statement sets the correspondence between pin numbers and device pin names.

Example:

```
attribute PIN_MAP of SAMPLE_DEVICE: entity is PHYSICAL_PIN_MAP;
constant SAMPLE_PACK: PIN_MAP STRING :=
  "PX1      : 1," &
  "PX2      : 2";
```

### 5.2.7 TAP signal definitions

The TAP signal definitions define the TAP signals (TDI, TDO, TCK, TMS, and TRST).

The highest operating frequency is set in the TCK definition. The term "BOTH" specifies that the TCK signal can stop at either the high or low level.

Example:

```
attribute TAP_SCAN_CLOCK of TCK      : signal is (20.0e6, BOTH);
attribute TAP_SCAN_IN    of TDI      : signal is true;
attribute TAP_SCAN_MODE  of TMS      : signal is true;
attribute TAP_SCAN_OUT   of TDO      : signal is true;
attribute TAP_SCAN_RESET of TRST     : signal is true;
```

In the above example, the highest operating frequency is set to be 20 MHz.

### 5.2.8 Compliance statement

The compliance statement limits the signal patterns so that boundary scan testing can be performed safely.

When a boundary scan test is performed, values that follow the patterns set here will be input continuously to the specified pins.

Example:

```
attribute COMPLIANCE_PATTERNS of SAMPLE_DEVICE:entity is
    "(PX6, PX7) (11)";
```

In the above example, the value 1 is input continuously to the PX6 pin, and the value 1 is input continuously to the PX7 pin.

### 5.2.9 Instruction register description

The instruction register description sets the instruction register (instruction code length) and defines the instruction codes and the IR status word. The low-order two bits of the bit pattern returned in the Capture-IR state must be 01. Note that although the BYPASS instruction is stipulated to be all ones, regardless of the number of bits, other codes can be specified by extension.

Example:

```
attribute INSTRUCTION_LENGTH of SAMPLE_DEVICE: entity is 4
attribute INSTRUCTION_OPCODE of SAMPLE_DEVICE: entity is
    "BYPASS (1111)," &
    "EXTEST (0000)," &
    "SAMPLE (1000)," &
    "CLAMP (1010)," &
    "HIGHZ (1011)," &
    "IDCODE (0101)";
attribute INSTRUCTION_CAPTURE of SAMPLE_DEVICE: entity is "1101";
```

In the above example, the instruction code length is 4 bits and IR status word is 1001.

### 5.2.10 Optional register definitions

The IDCODE register is defined in the optional register definitions.

Example:

```
attribute IDCODE_REGISTER of SAMPLE_DEVICE: entity is --ID code H'04632B97
    "0000" & -- 4-bit version number
    "0100011000110010 " & -- 16-bit part number
    "10111001011" & -- 11-bit identity of the manufacturer
    "1"; -- Required by IEEE Std 1149.1
```

In the above example, the IDCODE is set to H'04632B97.

In binary, this is 0000 0100 0110 0011 0010 1011 1001 0111.

The upper 4 bits (0000) indicate the version number. The next 16 bits (0100 0110 0011 0010) express the part number and the following 11 bits (1011 1001 011) are the manufacturer ID. The final bit (1) is stipulated to always be 1 by the IEEE standard.

### 5.2.11 Register Access Description

The register access description determines which register is inserted between TDI and TDO for which instructions.

Example:

```
attribute REGISTER_ACCESS of SAMPLE_DEVICE: entity is
    "BYPASS (BYPASS, CLAMP, HIGHZ)," &
    "BOUNDARY (EXTEST, SAMPLE)," &
    "DEVICE_ID (IDCODE)";
```

In the above example, the bypass register is inserted between TDI and TDO for the BYPASS, CLAMP, and HIGHZ instructions, the boundary scan register is inserted for the EXTEST and SAMPLE instructions, and the IDCODE register is inserted for the IDCODE instruction.



### 5.2.12 Boundary scan register description

The boundary scan register description sets up a list of boundary scan cells.

Since this list is used in boundary scan tests, there is no need to code pins whose type is linkage.

First, the cell length (6 in this example) is set. Then, the cell list is coded. Although the cell order is arbitrary, all cells must be specified. Cell number 0 is the cell closest to TDO, and the cell with the largest number (5 in this example) is the cell closest to TDI.

The content coded in this list consists of five items: number (the cell number), cell (the cell structure), port (connected port), function (the cell function), and safe (the safety value).

The number item is 0 for the cell closest to the TDO pin, and values up to one less than the cell length are set.

The cell item defines the structure used for the cell. The values used for the cell structures are defined in either the standard VHDL package or the VHDL package. Examples of cell structure values include BC\_4, which is an input-only cell, and BC\_1, which has all functions.

The port item is the name of the pin connected to that cell.

The function item sets the main function of the cell. Table 8 lists the meanings of the possible main functions.

**Table 8 Cell Functions**

Value	Function	Signal Type
input	Can be monitored and is connected to an external pin.	in or inout
observe_only	Has the same functions as type input.	in or inout
control	Performs enable/disable and direction control for one or more output signals or bidirectional signals.	—
output3	Connected to an external pin that has a 3-state output.	out or inout
internal	Used internally in the device. Not connected to external signals.	—

The safe item assures that safe values are set when boundary scan test pattern generation software generates random test patterns.

Example:

```
attribute BOUNDARY_LENGTH of SAMPLE_DEVICE: entity is 6;
attribute BOUNDARY_REGISTER of SAMPLE_DEVICE: entity is
-- num    cell    port    function    safe [ccell disval rslt]
   "5    (BC_4,  PX2,    observe_only,  X),
   "4    (BC_1,  *,      control,      0),
   "3    (BC_1,  PX2,    output3,      X,    4,    0,    Z),
   "2    (BC_4,  PX1,    observe_only,  X),
   "1    (BC_1,  *,      control,      0),
   "0    (BC_1,  PX1,    output3,      X,    1,    0,    Z) "
```

Here, "X" indicates a value that may be either 0 or 1. When the function item is output3, the value specified by disval is set for cells with ccell specified, and the value indicated by rslt will be output from the target pin. Also, the safe bit value for cells with ccell specified and the disval bit will always have the same value. "Z" indicates high impedance.

### 5.3 Sample BSDL File

```

-- *****
-- BSDL file for SAMPLE_DEVICE
-- Designer:
-- Company: Renesas, Ltd.
-- Date: FRI 07 13 2007
-- *****
entity SAMPLE_DEVICE is ) Entity description (start)

-- This section identifies the default device package selected. ) Parameter
generic (PHYSICAL_PIN_MAP: string:= "SAMPLE_PACK"); description

-- This section declares all the ports in the design. )
port (
    PX1          : inout  bit;
    PX2          : inout  bit;
    PX3          : inout  bit;
    PX4          : linkage bit;
    TRST         : in     bit;
    TMS          : in     bit;
    TDI          : in     bit;
    TCK          : in     bit;
    TDO          : out    bit;
    PX5          : in     bit;
    PX6          : in     bit;
    PX7          : in     bit;
);

use STD_1149_1_1994.all; ) Use statement
attribute COMPONENT_CONFORMANCE of SAMPLE_DEVICE: entity is "STD_1149_1_1993"; )
Standard version specification

```

```

attribute PIN_MAP of SAMPLE_DEVICE: entity is PHYSICAL_PIN_MAP;

-- This section specifies the pin map for each port. This information
-- is extracted from the port-to-pin map file that was read in using
-- the "read_pin_map" command.
constant SAMPLE_PACK: PIN_MAP_STRING :=
    "PX1          : 1," &
    "PX2          : 2," &
    "PX3          : 3," &
    "PX4          : 4," &
    "TRST         : 5," &
    "TMS          : 6," &
    "TDI          : 7," &
    "TCK          : 8," &
    "TDO          : 9," &
    "PX5          : 10," &
    "PX6          : 11," &
    "PX7          : 12";

-- This section specifies the TAP ports.
-- For the TAP TCK port, the parameters in the brackets are:
-- First Field : Maximum TCK frequency.
-- Second Field: Allowable states TCK may be stopped in.
attribute TAP_SCAN_CLOCK of TCK      : signal is (20.0e6, BOTH);
attribute TAP_SCAN_IN   of TDI       : signal is true;
attribute TAP_SCAN_MODE of TMS       : signal is true;
attribute TAP_SCAN_OUT  of TDO       : signal is true;
attribute TAP_SCAN_RESET of TRST     : signal is true;

attribute COMPLIANCE_PATTERNS of SAMPLE_DEVICE:entity is
    "(PX6, PX7) (11)";

-- Specifies the number of bits in the instruction register.
attribute INSTRUCTION_LENGTH of SAMPLE_DEVICE: entity is 4;

-- Specifies the boundary-scan instructions implemented in the
-- design and their opcodes.
attribute INSTRUCTION_OPCODE of SAMPLE_DEVICE: entity is
    "BYPASS (1111)," &
    "EXTEST (0000)," &
    "SAMPLE (1000)," &
    "CLAMP  (1010)," &
    "HIGHZ  (1011)," &
    "IDCODE (0101)";

-- Specifies the bit pattern that is loaded into the instruction
-- register when the TAP controller passes through the Capture-IR
-- state. The standard mandates that the two LSBs must be "01".
-- The remaining bits are design specific.
attribute INSTRUCTION_CAPTURE of SAMPLE_DEVICE: entity is "1101";

```

**Pin  
mapping(s)  
(the  
order is  
arbitrary)**

**TAP  
signal  
definitions**

**Compliance  
statement**

**Instruction  
register  
description**

```
-- Specifies the bit pattern that is loaded into the DEVICE_ID
-- register during the IDCODE instruction when the TAP controller
-- passes through the Capture-DR state.
attribute IDCODE_REGISTER of SAMPLE_DEVICE: entity is --ID code H'04632B97
    "0000" & -- 4-bit version number
    "0100011000110010" & -- 16-bit part number
    "10111001011" & -- 11-bit identity of the manufacturer
    "1"; -- Required by IEEE Std 1149.1
```

**Optional register definitions**

```
-- This section specifies the test data register placed between TDI
-- and TDO for each implemented instruction.

attribute REGISTER_ACCESS of SAMPLE_DEVICE: entity is
    "BYPASS (BYPASS, CLAMP, HIGHZ)," &
    "BOUNDARY (EXTEST, SAMPLE)," &
    "DEVICE_ID (IDCODE)";
```

**Register  
access  
description**

```

-- Specifies the length of the boundary scan register.
attribute BOUNDARY_LENGTH of SAMPLE_DEVICE: entity is 13;

-- The following list specifies the characteristics of each cell
-- in the boundary scan register from TDI to TDO.
-- The following is a description of the label fields:
-- num      : Is the cell number.
-- cell     : Is the cell type as defined by the standard.
-- port     : Is the design port name. Control cells do not
--           have a port name.
-- function: Is the function of the cell as defined by the
--           standard. Is one of input, output2, output3,
--           bidir, control or controlr.
-- safe     : Specifies the value that the BSR cell should be
--           loaded with for safe operation when the software
--           might otherwise choose a random value.
-- ccell    : The control cell number. Specifies the control
--           cell that drives the output enable for this port.
-- disval   : Specifies the value that is loaded into the
--           control cell to disable the output enable for
--           the corresponding port.
--         rslt    : Resulting state. Shows the state of the driver
--                 when it is disabled.
attribute BOUNDARY_REGISTER of SAMPLE_DEVICE: entity is
--
--   num  cell  port  function      safe [ccell  disval  rslt]
--
"12 (BC_4,  PX5,  observe_only, X),          " &
"11 (BC_4,  *,   internal,      X),          " &
"10 (BC_1,  *,   internal,      X),          " &
"9  (BC_1,  *,   internal,      X),          " &
"8  (BC_4,  PX3,  observe_only, X),          " &
"7  (BC_1,  *,   control,       0),          " &
"6  (BC_1,  PX3,  output3,      X,  7,    0,    Z), " &
"5  (BC_4,  PX2,  observe_only, X),          " &
"4  (BC_1,  *,   control,       0),          " &
"3  (BC_1,  PX2,  output3,      X,  4,    0,    Z), " &
"2  (BC_4,  PX1,  observe_only, X),          " &
"1  (BC_1,  *,   control,       0),          " &
"0  (BC_1,  PX1,  output3,      X,  1,    0,    Z) " ;

end SAMPLE_DEVICE; ) Entity statement (end)

```

**Boundary  
scan  
register  
description**

## 6. Boundary Scan Test System Structure Example

Finally, in figure 25, we present an example of the structure of a system that uses boundary scan testing.

The most important of the required programs and files include the BSDL file, the test pattern generation program, the JTAG control program, the system control program, the test pattern creation program, and the test result analysis program. The test pattern generation program defines instructions and test vectors. The JTAG control program controls the test according to the JTAG protocol. The system control program controls the evaluation system as a whole. The test pattern creation program creates the patterns to be output when the test is actually run. The test result analysis program determines whether or not the system is operating normally from the test results (which consist of a sequence of zeros and ones) acquired from the boundary scan test.

On the host computer, programs such as the test pattern generation program, the JTAG control program, the system control program, and the test pattern creation program read in the BSDL file and perform the boundary scan test by the input and output of TAP signals using the JTAG controller. The test results are analyzed by the test result analysis program, which outputs a judgment as to whether or not the system is operating normally as the test result.

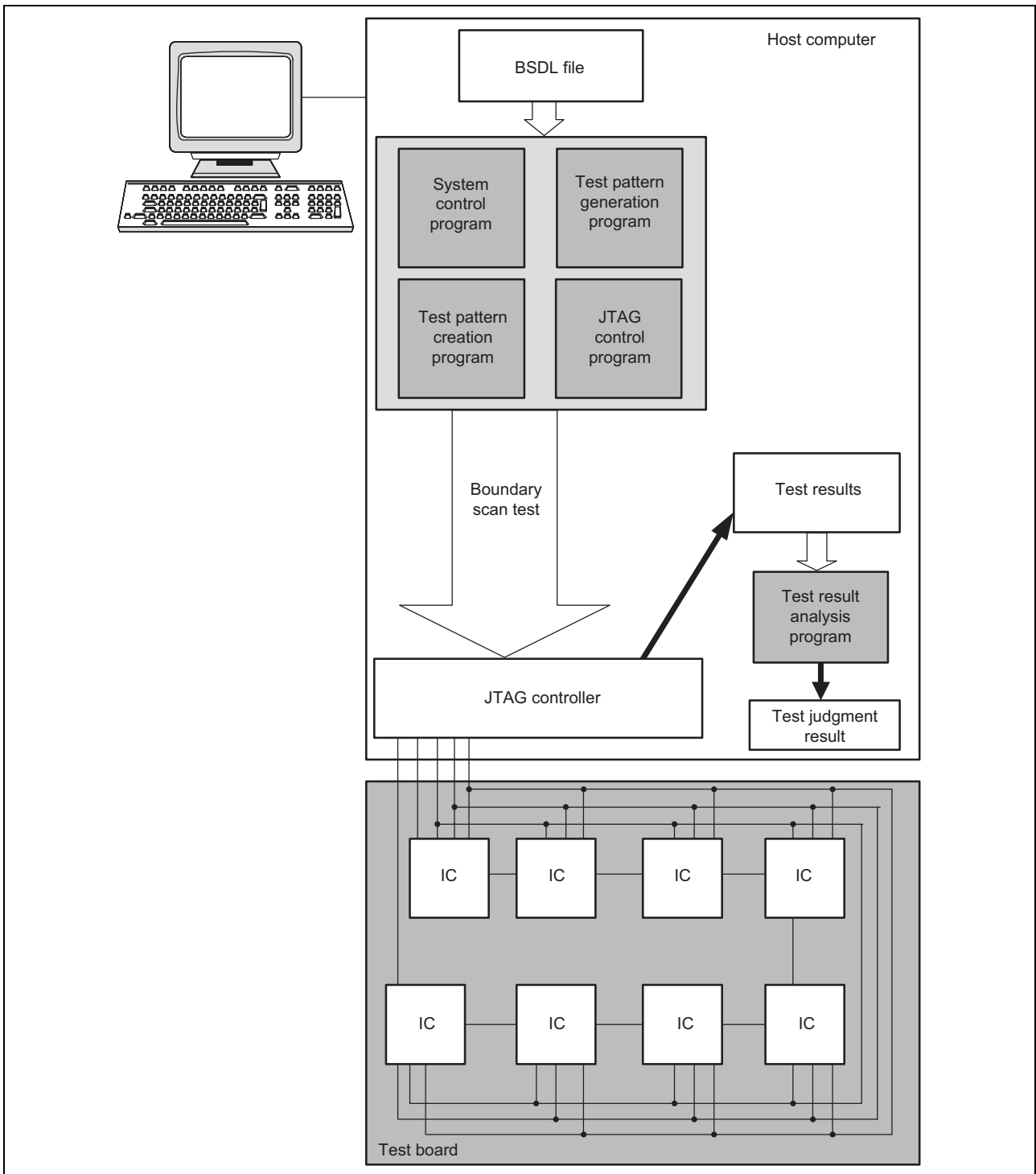


Figure 25 Boundary Scan Test System Structure Example

## 7. References

Sakamaki, Kazumi; Basics and Applications of JTAG Testing: Electronic circuit board testing for a new age and a variety of application examples. CQ Publishing Co., Ltd., 2003

## 8. Reference Documents

- Hardware Manuals
  - H8SX/1638 Group Hardware Manual
  - H8SX/1648 Group Hardware ManualThe latest versions of these manuals can be ordered from the Renesas Technology Corp. web site.
- Technical Manuals and Technical Updates
  - The latest information is available on the Renesas Technology Corp. web site.



## Website and Support

Renesas Technology Website

<http://www.renesas.com/>

Inquiries

<http://www.renesas.com/inquiry>

[csc@renesas.com](mailto:csc@renesas.com)

## Revision Record

Rev.	Date	Description	
		Page	Summary
1.00	Jan.22.09	—	First edition issued

All trademarks and registered trademarks are the property of their respective owners.

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
  - (1) artificial life support devices or systems
  - (2) surgical implantations
  - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
  - (4) any other purposes that pose a direct threat to human life

Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.