IEEE Standard 1149.1 (JTAG) in the SX/RTSX/SX-A/eX/RT54SX-S Families

Table of Contents

Introduction

Testing modern loaded circuit boards has become extremely expensive and very difficult to perform. The rapid development of surface-mount technology and the use of multi-layer boards has increased board complexity. Finer pin spacing and the use of double-sided boards also have contributed to the increased cost and difficulty of traditional testing, which makes use of methods such as in-circuit testing by bed-of-nails and functional testing. Although functional testing can cope with complex and dense boards, it is costly because different designs require different sets of test programs.

The test architecture was developed by the Joint Test Action Group (JTAG) and later adopted by IEEE as the IEEE Standard Test Access Port and Boundary-Scan Architecture (also referred to as IEEE Std. 1149.1 or informally known as JTAG). The standard provides a cost-effective method of board testing through use of the boundary-scan technique. Boundary scan provides the means to test each component's required performance, interconnections, and interaction. In addition to describing boundary scan, the standard also describes the design-for-test feature.

Overview

The Micorsemi SX, RTSX, SX-A, eX, and RT54SX-S families are fully compliant with IEEE Standard 1149.1.[1] Figure 1 on page 2 shows the major parts that make up the JTAG test logic circuit. The circuit provides the required components (Test Access Port controller and registers) to support all the mandatory boundary-scan instructions (EXTEST, SAMPLE/PRELOAD, and BYPASS) as well as six optional public instructions (INTEST, USERCODE, IDCODE, Diagnostic, HIGHZ, and CLAMP). The diagnostic instruction is very similar to the JPROBE instruction on Microsemi's legacy parts.

1. *There is a minor exception. See the "Instructions" section on page 7.*

JTAG Mode Selection

The JTAG test logic mode is selected in the Designer software by selecting Tools > Device Selection. Go to the second dialog box (accessed after Next) as shown in Figure 2 on page 3. Click the "Reserve JTAG" check box to reserve pins for JTAG (dedicated mode). Dedicated mode is recommended if JTAG is to be used extensively. If the box is not checked, flexible mode is selected by default.

The JTAG dedicated mode can also be selected when using TCL scripting by adding the command:

```
set_device -jtag "yes"
```

If the unshaded box, "Reserve JTAG Test Reset", appears (SX-A or eX devices), the user also has the option of reserving a pin for the JTAG TRST signal (see the "Test Access Port (TAP)" section). The JTAG TRST pin can be reserved when using TCL scripting with the command:

```
set_device -trst "yes"
```

These can also be done with a single line:

```
set_device -jtag "yes" -trst "yes"
```
[2]

Test Access Port (TAP)

Each test logic function is accessed through the TAP. The five pins associated with the TAP are listed in Table 1 on page 3 with their corresponding descriptions. Four pins – TMS, TCK, TDI, and TDO – are always required for JTAG operation. The fifth pin, TRST, is optional. These pins are dedicated pins – used only with the test logic. If flexible mode is selected, three of the pins – TCK, TDI, and TDO – are free to be used as regular I/O pins. Refer to the "JTAG Flexible Mode in SX, SX-A, RT54SX-S, and eX" section on page 4. Note that TRST (if present) and TDI are equipped with internal pull-up resistors. This means that these pins need not to be terminated in the dedicated mode to ensure proper JTAG operation. In the dedicated mode, the TMS pin is equipped with a pull-up resistor to place the TAP controller in the reset state (after a minimum of 5 TCK pulses) when no input is present. In the flexible mode, there is NO pull-up resistor; an external 10kΩ pull-up resistor is required.

The test logic was designed to be in the reset state upon power up. The next section describes the TRST pin usage differences among the various families.

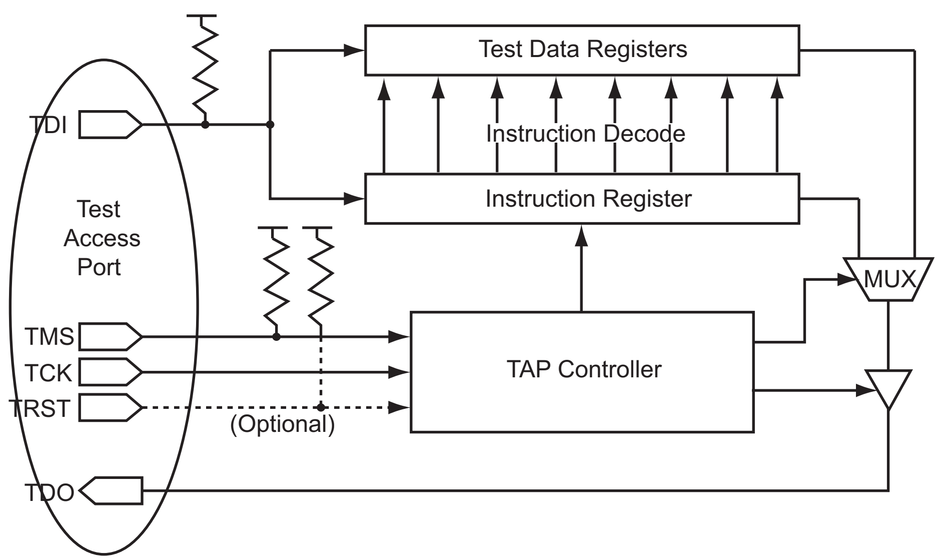

***Figure 1* • JTAG Block Diagram**

2. *This has been tested in a simple eX64-TQ64 design and the only variables affected are the RESTRICTJTAGPINS and RESTRICTTRSTPIN, respectively.*

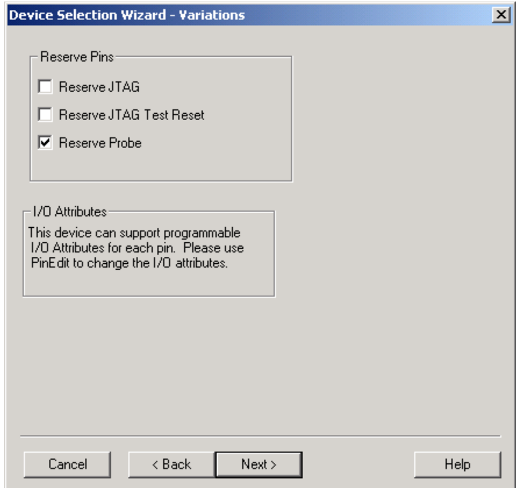

***Figure 2* • SX-A Device Selection Dialog Box**

***Table 1* • Test Access Port Descriptions**

Port	Description
Test Mode Select (TMS)	Serial input for the test logic control bits. Data is captured on the rising edge of the test logic clock (TCK). An internal pull-up resistor is present in dedicated mode but not in flexible mode. Refer to the "Test Access Port (TAP)" section on page 2 for more information.
Test Clock Input (TCK)	Dedicated test logic clock used serially to shift test instruction, test data, and control inputs on the rising edge of the clock, and serially to shift the output data on the falling edge of the clock. The maximum clock frequency for TCK is 20 MHz.
Test Data Input (TDI)	Serial input for instruction and test data. Data is captured on the rising edge of the test logic clock. This pin is equipped with an internal pull-up resistor.
Test Data Output (TDO)	Serial output for test instruction and data from the test logic. TDO is set to an Inactive Drive state (high impedance) when data scanning is not in progress.
Test Reset (TRST) (Optional)	Active-low input which asynchronously resets the test logic. This pin is equipped with an internal pull-up resistor.

Summary of TRST Pin Usage

The TRST pin is optional in JTAG implementations. There are three different schemes used with the families described in this application note. They are:

- SX, RTSX (Early Versions) – No TRST pin
- RTSX (Later Versions) – Dedicated TRST pin, no internal pull-up resistor
- SX-A, eX – User-configurable TRST pin
- RT54SX-S – Dedicated TRST

In parts with the user-configurable TRST pin, if the "Reserve JTAG Test Reset" check box is selected, then the TRST pin functions as a dedicated TRST pin. If this is not selected, then the pin can be employed as a user I/O in flexible mode. Refer to the "JTAG Flexible Mode in SX, SX-A, RT54SX-S, and eX" section for more information.

Note that in either dedicated or flexible modes, TRST overrides the behavior of TMS and TCK. In other words, asserting TRST resets the TAP controller regardless of the states of TMS and TCK. Also, if the TAP controller is held in reset, asserting TMS and TCK will have no effect, the TAP controller will remain in the reset state.

JTAG Flexible Mode in SX, SX-A, RT54SX-S, and eX

In this mode TCK, TDI, and TDO are non-dedicated JTAG pins. In other words, use them as regular I/Os and as JTAG pins.

- When TMS is driven "Low", TCK, TDI, and TDO act as JTAG pins. The first TCK low to high transition causes the TAP controller to exit the reset state and move to the Run-Test-Idle state.
- Exiting JTAG operation requires that TMS is driven high and TCK is clocked five times. This puts the TAP controller in the reset state at which point the pins are released from JTAG operation and are free to be used as regular user I/Os.

Figure 3 on page 4 illustrates the timing for transitions to and from JTAG flexible mode.

Although TCK and TDI can be used as ANY type of user I/O in flexible mode, it is recommended that these pins are used as outputs. (If used as inputs, you must provide external multiplexing logic to switch between the JTAG control inputs and the user inputs).

TAP Controller

The 16 states of the tap controller state machine are shown in Figure 4 on page 5. The 1s and 0s shown adjacent to the state transitions represent the TMS values that must be present at the time of a rising edge at TCK for a state transition to occur. In the states that include the letters -IR, the instruction register operates; in the states that contain the letters -DR, the test data register operates (bypass, boundary-scan, and XY-registers). The TAP controller receives two control inputs, TMS and TCK, and generates control and clock signals for the rest of the test logic architecture, as illustrated in Figure 5 on page 5.

Upon power up (or on the assertion of TRST), the TAP controller enters the Test-Logic Reset state. To reset the controller from any other state, TMS must be held high for at least five TCK cycles. After reset, the TAP controller's state changes at the rising edge of TCK based on the value of TMS.

Note: The value shown adjacent to the state transitions in this Figure 4 on page 5 represents the signal present at TMS at the time of the rising edge of TCK.

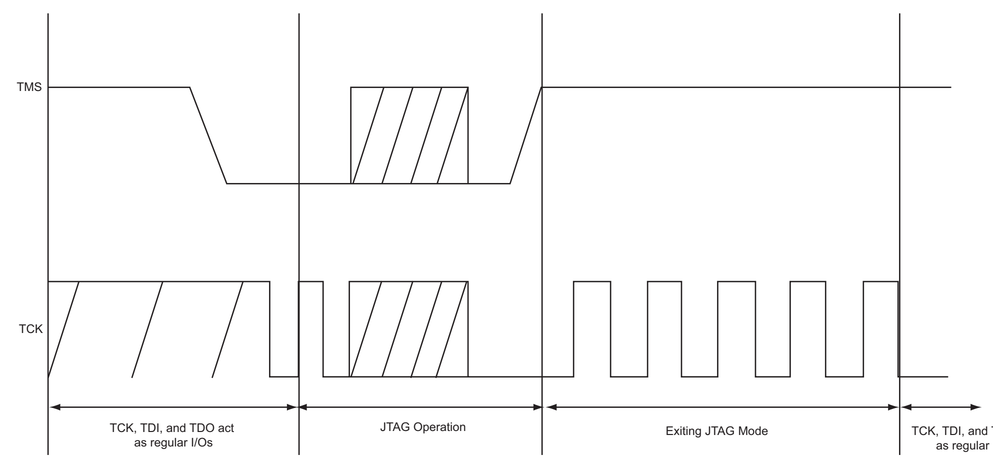

Figure 3 **• Entering and Leaving JTAG Flexible Mode**

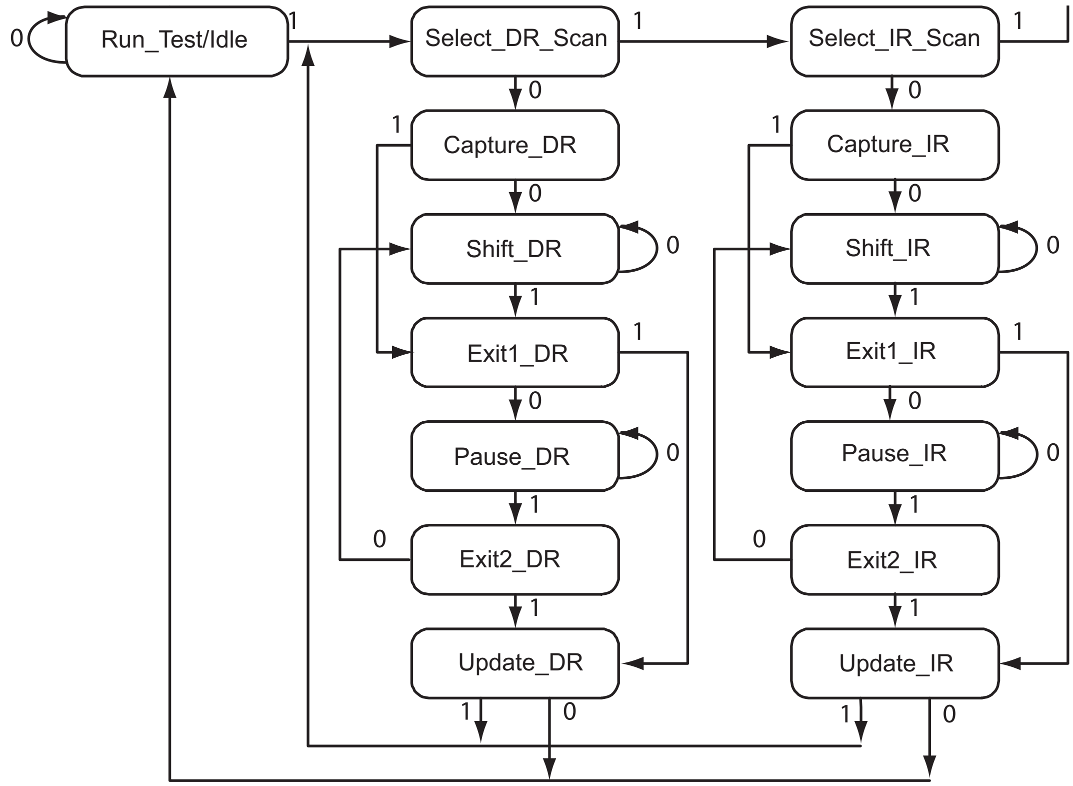

Figure 4 • **TAP Controller State Diagram**

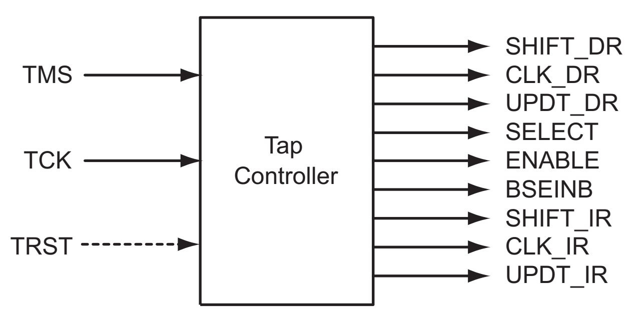

Figure 5 • **Tap Controller Block Diagram**

Instruction Register

The instruction register (IR) consists of five IR cells. Each cell has a shift-register stage and a latch stage (Figure 6 on page 6). On the Capture_IR state, the shift register is loaded with bits 11101, which are used for fault isolation of the board-level serial test data path. The TDI-IR-TDO path is established on the Shift_IR state. The data in the shift register is shifted toward TDO, and data in the latch remains the same. The data in the shift registers is latched out and becomes the current instruction on the falling edge of the TCK in the Update_IR state. When the TAP controller enters the Test-Logic Reset state, bits 00100 are latched in the IR, which corresponds to the IDCODE instruction, and the data in the shift register cell retain their previous values. Table 2 on page 6 shows the summary of the operation of the instruction register.

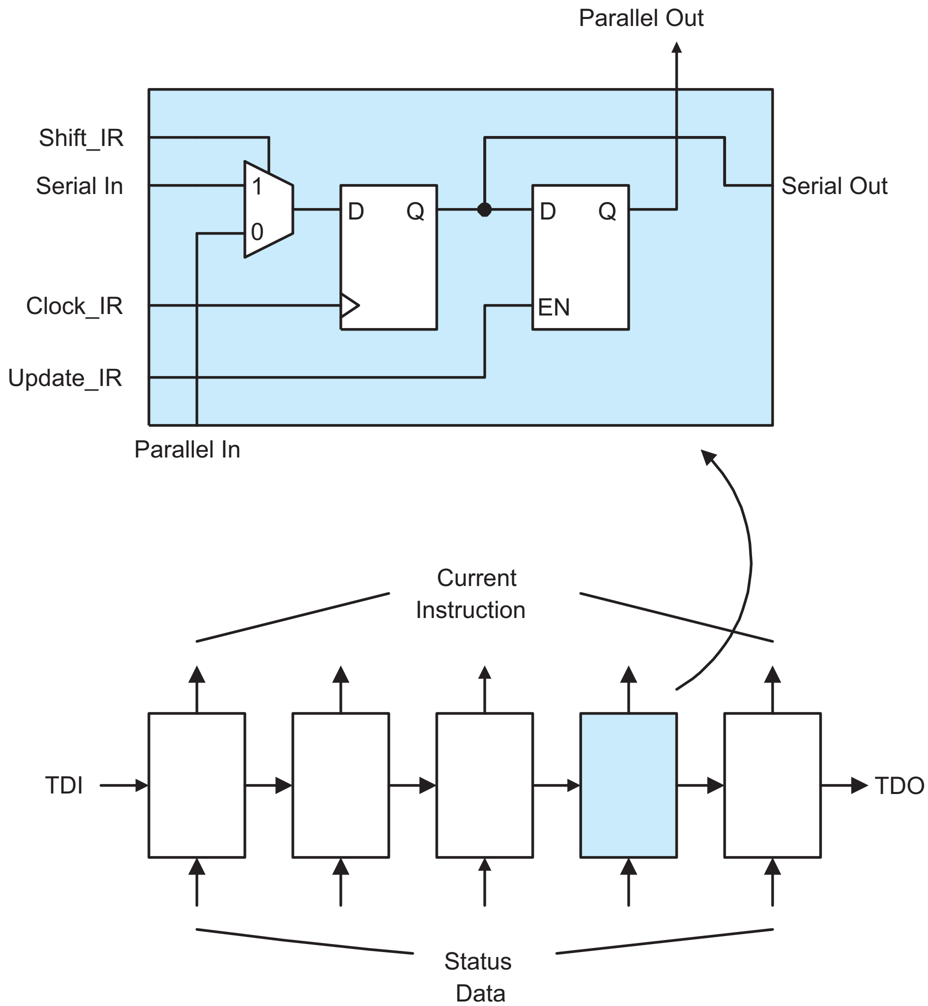

***Figure 6* • Instruction Register Block Diagram**

***Table 2* • Instruction Register Operation**

Controller State	Shift-Register Stage	Latch Stage
Test_Logic_Reset	Undefined	IDCODE Instruction (IR4 - IR0 = 00100)
Capture_IR	11101 Is Loaded	Retain Previous State
Shift_IR	Shift Data Toward TDO	Retain Previous State
Exit1_IR	Retain Previous State	Retain Previous State
Exit2_IR		
Pause_IR		
Update_IR	Retain Previous State	Latch Data from Shift Register
All Other States	Undefined	Retain Previous State

Instructions

Table 3 lists the supported instructions with their corresponding IR codes and descriptions. All unused opcodes should be considered as reserved for Microsemi use only.

Note: The INTEST instruction does not fully comply with rule 7.8.1(b) of IEEE Standard 1149.1-1990. No single-step capability for the three clock inputs HCLK, CLKA, and CLKB is provided because these are high-performance clock pins and only 'observable' boundary-scan cells are included in the scan chain. The same comment applies to the quadrant clocks (QCLKA/B/C/D) on the A54SX72A and the RT54SX72S.

Device Identification Register

These families implement the optional JTAG Device Identification Register as one of the Test Data Registers shown in Figure 1 on page 2. The USERCODE and IDCODE instructions use this register. The USERCODE instruction allows you to program the desired 32-bit design/device related information.

The IDCODE instruction reads a hardwired 32-bit ID code associated with the following:

- Manufacturer's identity
- Part number
- Version number

Bit 0 is always a '1'. Bits 11 to 1 are always '02F' – which is Microsemi's manufacturer code. Table 4 lists the ID codes for various part numbers and revisions which are Bits 31 to 12 of the 32-bit code. Bits 27 to 12 provide the part number and bits 31 - 28 indicate the revision number.

Table 3 • **Supported Public Instructions**

Instruction	IR Code (IR4 - IR0)	Instruction Type	Description
EXTEST	00000	Mandatory	Allows testing of off-chip circuitry and board-level interconnections
SAMPLE/PRELOAD	00001	Mandatory	Allows a snapshot of the normal operation of the component to be taken and examined
INTEST	00010	Optional	Allows testing of on-chip system logic while component is assembled on the board
USERCODE	00011	Optional	32-bit user-programmable identification code
IDCODE	00100	Optional	32-bit hard-wired Microsemi ID, part number, and version number
HIGHZ	01110	Optional	Tristates all I/Os to allow external signals to drive pins
CLAMP	01111	Optional	Allows state of signals driven from component pins to be determined from the Boundary-Scan Register
Diagnostic	10000	Optional	Allows microprobing of internal logic module's output logic state
BYPASS	11111	Mandatory	Provides minimum-length (1-bit) serial path between TDI and TDO pins of component when no test operation of that component is required

Table 4 • **ID Codes for Various Parts and Revisions**

Part	Process	Revision	Bits 31-28	Bits 27-12
RTSX				
RTSX16	0.60μ	0	0	00B8
RTSX32	0.60μ	0	0	00BD

Table 4 • **ID Codes for Various Parts and Revisions (continued)**

Part	Process	Revision	Bits 31-28	Bits 27-12
RTSX16	0.60µ	1	0	01B8
RTSX32	0.60µ	1	0	01BD
SX				
SX08	0.35µ	0	0	10B4, 12B4
SX16	0.35µ	0	0	10B8, 12B8
SX32	0.35µ	0	0	10BD, 12BD
SX08	0.35µ	1	1	12B4
SX16	0.35µ	1	1	12B8
SX32	0.35µ	1	1	12BD
SX16P	0.35µ	0	0	11B8, 13B8
SX16P	0.35µ	1	1	13B8
SX32P	0.25µ	0	0	20BD, 22BD
SX-A				
SX08A	0.22µ	0	8, 9	40B4, 42B4
SX16A	0.22µ	0	9	40B8, 42B8
SX32A	0.22µ	0	9	40BD, 42BD
SX72A	0.22µ	0	9	40B2, 42B2
SX08A	0.22µ	1	A B	40B4 42B4
SX16A	0.22µ	1	B	40B8, 42B8
SX32A	0.22µ	1	B	40BD, 42BD
SX72A	0.22µ	1	B	40B2, 42B2
SX16A	0.25µ	1	B	22B8
SX32A	0.25µ	1	B	22BD
SX72A	0.25µ	1	B	22B2
eX				
eX64	0.22µ	0	8	40B2, 42B2
eX128	0.22µ	0	9	40B0, 42B0
eX256	0.22µ	0	9	40B5, 42B5
eX64	0.22µ	0	A	40B2, 42B2
eX128	0.22µ	0	B	40B0, 42B0
eX256	0.22µ	0	B	40B5, 42B5
RTSX-SU				
RTSX32S	0.25µ	0	C	20BD, 24BD
RTSX72S	0.25µ	0	C	20B2, 24B2
RTSX32S	0.25µ	1	C	21BD, 25BD
RTSX72S	0.25µ	1	C	21B2, 25B2
RTSX32S	0.25µ	2	E	21BD, 25BD
RTSX72S	0.25µ	2	E	21B2, 25B2

Bypass Register

The bypass register is a single-bit register that provides a minimum data path between the TDI and TDO pins (Figure 7 on page 9). The bypass register is selected when the BYPASS, HIGHZ, or CLAMP instruction is the current instruction in the instruction register. On the Capture_DR controller state, 0 is loaded into the bypass register. Test data can then be shifted from the TDI pin to the TDO pin on the Shift_DR state. By moving into the Update_DR controller state, data movement through the bypass register is terminated. Table 5 on page 9 shows a summary of the operation of the bypass register.

Boundary-Scan Register

The boundary-scan register is used to observe and control the state of each system pin. Note that clock pins can only be observed, not controlled. Each boundary-scan cell consists of serial input (SI) and serial output (SO) that are connected to each cell, as shown in Figure 8 on page 10. In addition, each boundary-scan cell (BSC) consists of a parallel input (PI) and a latched parallel output (PO) that connect to the system logic and system output. Three cells are used for each I/O – an input cell (BS2), an output cell (BS1), and an output-enable cell (BS0).

Table 5 • **Bypass Register Operation**

Controller State	Bypass Register
Test-Logic-Reset	Retain Previous State
Capture_IR	'0' is Loaded
Shift_IR	Shift Data toward TDO
Exit1_IR	Retain Previous State
Exit2_IR	
Pause_IR	
Update_IR	Retain Previous State
All Other States	Undefined

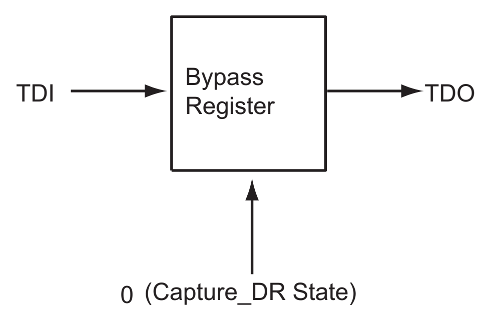

Figure 7 • **Bypass Register Diagram**

The operation of the boundary-scan register during specific boundary-scan instructions is described in Table 6 and Table 7 on page 11. If the EXTEST instruction is not being used in conjunction with the SAMPLE/PRELOAD instruction, the external test starts by shifting the desired test data into the boundary-scan register in the Shift_DR controller state. By moving into the Update_DR controller state, data shifting is terminated and on the falling edge of the TCK, the data from the shift-register stage is transferred onto the parallel output of the latch stage. The external test results are loaded into the shift-register stage from the system input in the next Capture_DR controller state. These results are examined by shifting the data towards TDO on the next Shift_DR controller state. During the SAMPLE/PRELOAD instruction, the Shift_DR state is used to shift out the data captured from the system input and output pins for examination during the Capture_DR state. At the same time, the Shift_DR state shifts in test data to be used by the next boundary-scan instruction (other than SAMPLE/PRELOAD). The EXTEST instruction is usually initiated following the SAMPLE/PRELOAD instruction.

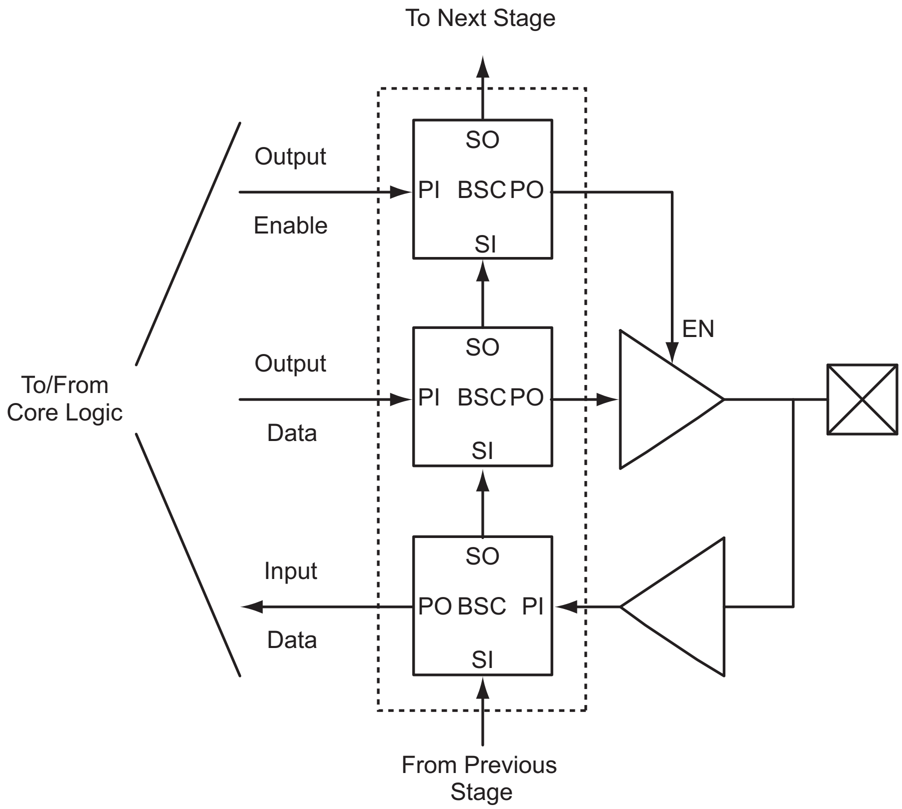

Figure 8 **• Functional Schematic of the Boundary-Scan Cell**

The data pre-loaded during the SAMPLE/PRELOAD instruction phase becomes available at the parallel output of the boundary-scan cells when the EXTEST becomes the current instruction on the rising edge of TCK in the Update-IR state. Similarly, the CLAMP instruction is usually initiated following the SAMPLE/PRELOAD instruction. The latched data in the boundary-scan cell becomes available to the system output pins when CLAMP becomes the current instruction and when the bypass register is selected as the data path from TDI to TDO.

During the SAMPLE/PRELOAD instruction, the parallel input and output of the boundary-scan cells are transparent (PI equals PO).

Diagnostic Instruction

The diagnostic instruction (IR code 10000) allows microprobing of internal module outputs. This is done via an XY-register. The scan chain structure for all of the families described in this application note is illustrated in Figure 9 on page 12. The XY-register consists of a shift register whose length depends on the specific part. The registers that are darkened are not parts of the XY-register. The presence of the XY-register and the diagnostic instruction permits the use of the internal probe circuitry to observe and analyze any signal inside the Microsemi chip via JTAG.

The desired probe address is shifted into the XY-register by first selecting the diagnostic instruction and then moving to the Shift_DR controller state. Shifting is discontinued by entering the Update_DR controller state. The probe results are loaded into the XY-register on the rising edge of TCK in the next Capture_DR controller state. The probe results can be examined by moving back to the Shift_DR controller state and shifting the result towards TDO. Table 9 on page 12 shows the summary of the diagnostic instruction's operation. The probe results may also be observed in real time at the probe pins (PRA and PRB), provided that these pins have been reserved for probe use.

Table 6 • **Operation Summary of EXTEST Instruction**

Controller State	Boundary-Scan Shift-Register Stage	Boundary-Scan Latch Stage	Parallel Output (PO)
Test_Logic_Reset	Undefined	Undefined	Parallel In = Parallel Out
Capture_DR	Data at PI Is Loaded	Retain Previous State	Latched Data
Shift_DR	Shift Data Toward TDO	Retain Previous State	Latched Data
Exit1_DR, Exit2_DR, Pause_DR	Retain Previous State	Retain Previous State	Latched Data
Update_DR	Retain Previous State	Latches Data from Shift Register	Latched Data = Parallel Out
All Other States	Retain Previous State	Retain Previous State	Latched Data

Table 7 • **Operation Summary of SAMPLE/PRELOAD Instruction**

Controller State	Boundary-Scan Shift-Register Stage	Boundary-Scan Latch Stage	Parallel Output (PO)
Test_Logic_Reset	Undefined	Undefined	Parallel In = Parallel Out
Capture_DR	Retain Previous State	Data at PI is loaded	Parallel In = Parallel Out
Shift_DR	Shift Data Toward TDO	Retain Previous State	Parallel In = Parallel Out
Exit1_DR, Exit2_DR, Pause_DR	Retain Previous State	Retain Previous State	Parallel In = Parallel Out
Update_DR	Retain Previous State	Latches Data from Shift Register	Parallel In = Parallel Out
All Other States	Retain Previous State	Retain Previous State	Parallel In = Parallel Out

Note: During the SAMPLE/PRELOAD instruction, the parallel input and output of the boundary-scan cells are transparent (PI equals PO).

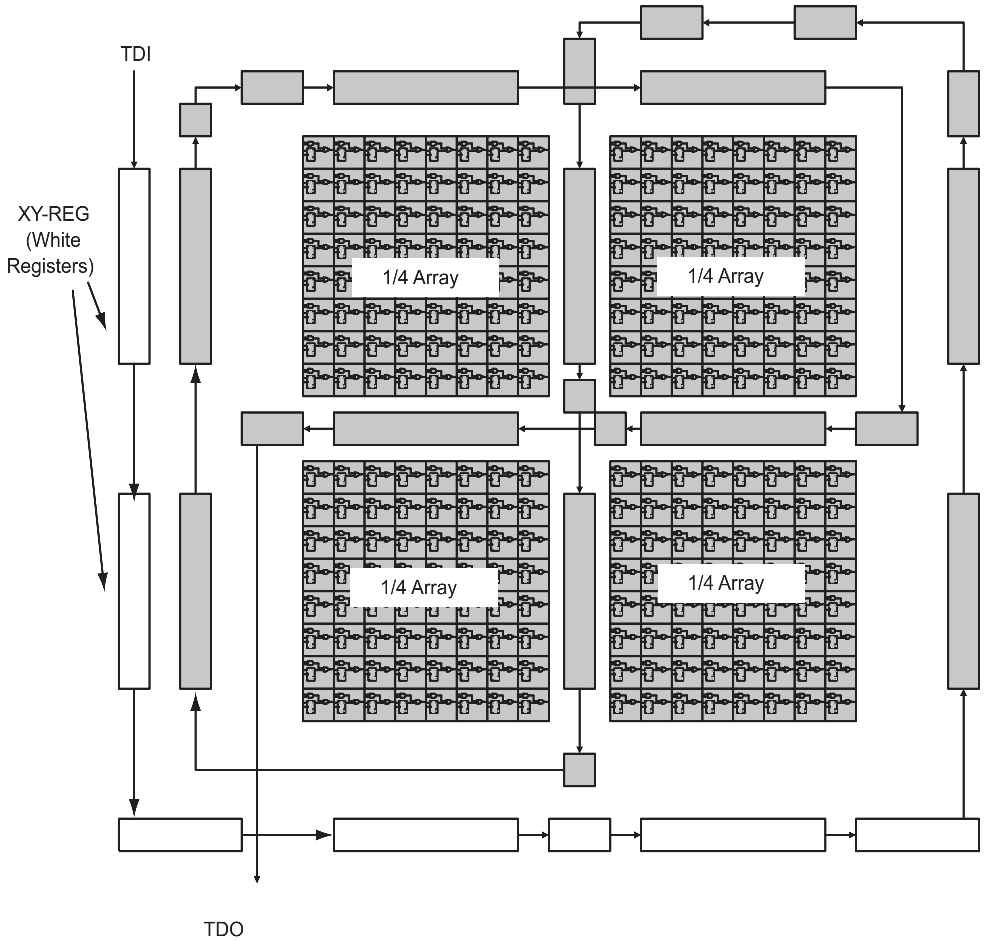

Figure 9 **• Functional Schematic of the XY-Register**

Table 8 **• Operation Summary of Diagnostic Instruction**

Controller State	**XY-REG Register**	**XY-REG Latch Stage**
Test_Logic_Reset	Logic ‘0’	Logic ‘0’
Capture_DR	Probe Result Loaded when Valid Probe Register Address	Retain Previous State
Shift_DR	Shift In New Address and Shift Out Probe Result Toward TDO	Retain Previous State
Exit1_DR, Exit2_DR, Pause_DR	Retain Previous State	Retain Previous State
Update_DR	Retain Previous State	Latch Data from Shift Register
All Other States	Undefined	Undefined

Boundary-Scan Description Language (BSDL) File

Conforming to the IEEE Standard 1149.1 requires that the operation of the various JTAG components be documented. The BSDL file provides the standard format to describe the JTAG components that can be used by automatic test equipment software. The file includes the instructions that are supported, instruction bit pattern, and the boundary-scan chain order. Note that if a general-purpose I/O in a customer design is configured as either an output (OUTPUT) or a tristate buffer (TRIBUF), the input for that pad and hence the JTAG input boundary-scan cell (the lower cell in Figure 8 on page 10) is disabled. This cell does not exist in the BSDL file generated by Designer Software.

References

1. Colin M. Maunder & Rodham E. Tulloss. *The Test Access Port and Boundary-Scan Architecture*. IEEE Computer Society Press, Los Alamitos.
2. "IEEE Std 1149.1-1993, IEEE Standard Test Access Port, and Boundary-Scan Architecture." IEEE, Inc., New York.
3. Kenneth P. Parker. *The Boundary-Scan Handbook*. Kluwer Academic Publishers, Norwell.

List of Changes

The following table lists critical changes that were made in each revision of the document.

Revision*	Changes	Page
Revision 1 (May 2012)	The "Test Access Port (TAP)" section was revised (SAR 23985).	2

Note: *The revision number is located in the part number after the hyphen. The part number is displayed at the bottom of the last page of the document. The digits following the slash indicate the month and year of publication.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

5192711-1/05.12