System Testability Using Standard Logic

SCTA037A
October 1996

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

List of Illustrations

Introduction

The use of more sophisticated ASICs and microprocessors and a steadily increasing move toward surface-mount packaging has led to the increasing complexity and density of digital systems. These advances improve system performance and decrease the physical size of printed-circuit boards (PCBs) but complicate the task of system testing. Access to test nodes using bed-of-nails testing is reduced or eliminated, and the modern automated test equipment necessary to fully exercise complex chips can be prohibitively expensive. One way of circumventing these problems is to use a boundary-scan test method, which can control and observe any node in a system through a dedicated test bus. An IEEE standard currently under development defines a four-wire test bus and protocol that implements a boundary-scan methodology. Standard logic functions, called System Controllability, Observability, and Partitioning Environment (SCOPE™) bus-interface products that incorporate this four-wire bus can be strategically placed in a digital system design. These products greatly enhance overall testability in areas from design and prototype debug to final test and field service of production systems. This paper presents some examples that illustrate the expanded test capabilities available in the earlier bus-interface products – the SCOPE test octals.

IEEE 1149.1

An IEEE 1149.1 document defines a four-wire boundary-scan test protocol that can be implemented in any digital integrated circuit (IC) (see Reference 1 for details of the boundary-scan methodology). Chips designed to comply with this protocol can be interconnected to form one or more serial shift register chains, or scan paths, within a system. Texas Instruments (TI) has designed four octal ICs that adhere to the IEEE 1149.1 protocol. These include many additional capabilities that increase their effectiveness in board, subsystem, or system test.

An Overview of SCOPE/Boundary Scan

Boundary scan is a design-for-testability method that allows signals to be captured and/or forced at the I/O pins of a digital device. All test data is serially shifted from the test data input (TDI) to the test data output (TDO) through some register in the device specifically designated for boundary-scan operations. The current instruction defines which of the registers in the device are connected between the TDI and TDO. Only one register at a time may be connected and accessed.

A primary advantage of boundary-scan testing is that the only physical access required is the four-wire test bus. Through this test bus, the user can access any testable node or signal in the system. The boundary-scan method has built-in safeguards to make it relatively fault tolerant. The test protocol allows for a hierarchical approach to test in which the system is subdivided into standalone subsystems that are convenient for the design and test engineers to use.

A block diagram for a SCOPE bus-interface product, the SN74BCT8244A, is shown in Figure 1. The device is operated in two modes, normal and test. In the normal mode, the ’8244A functions identically to an SN74F244, an eight-wide noninverting buffer. In the test mode, additional circuitry is activated to perform a specified test operation.

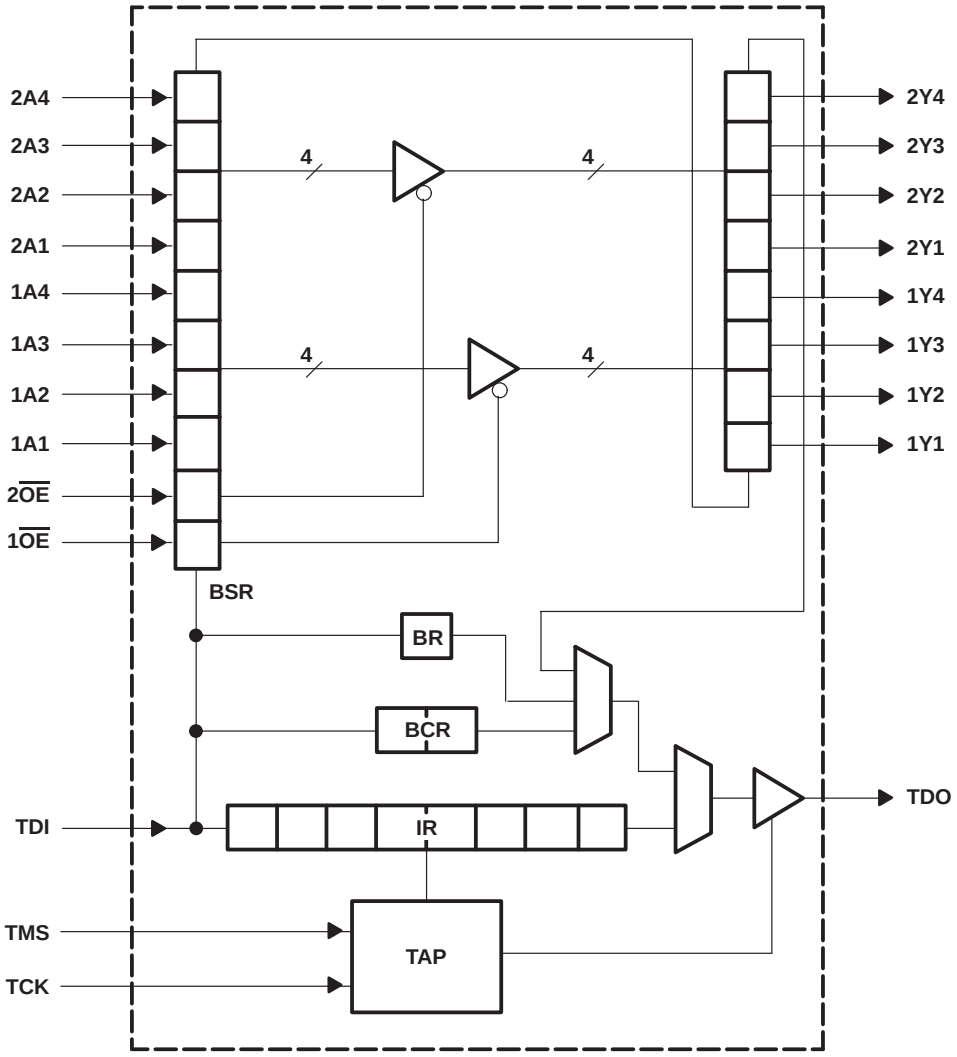

Figure 1. SN74BCT8244A Block Diagram

Figure 1 shows that the chip contains four serial registers, with each square block representing one bit of the register. One register is the 8-bit instruction register (IR), which determines the test operation to be performed. The other three are data registers used to capture, load, and shift the serial test information. The 18-bit boundary-scan register (BSR) contains one boundary-scan cell (BSC) for each functional input and output on the device. The 2-bit boundary-control register (BCR) is an indirect instruction register that is used to implement special test operations that are not part of the standard SCOPE instruction set. The 1-bit bypass register (BR) effectively removes the device from the scan path by providing a short (one clock cycle) delay through the chip. Operation of the test circuitry is synchronous to the test clock (TCK). The ’BCT8244A advances through its state machine (see Figure 2) according to the value of the test mode select (TMS) pin at each TCK rising edge. Inputs are captured on the rising edge of TCK, and outputs change on the falling edge of TCK. Control instructions are issued by the test access port (TAP). The TDI and TDO pins are the serial test data input and output pins, respectively. By connecting the TDO pin of one device to the TDI pin of the next device in the scan path, a serial chain is formed through which all information is passed.

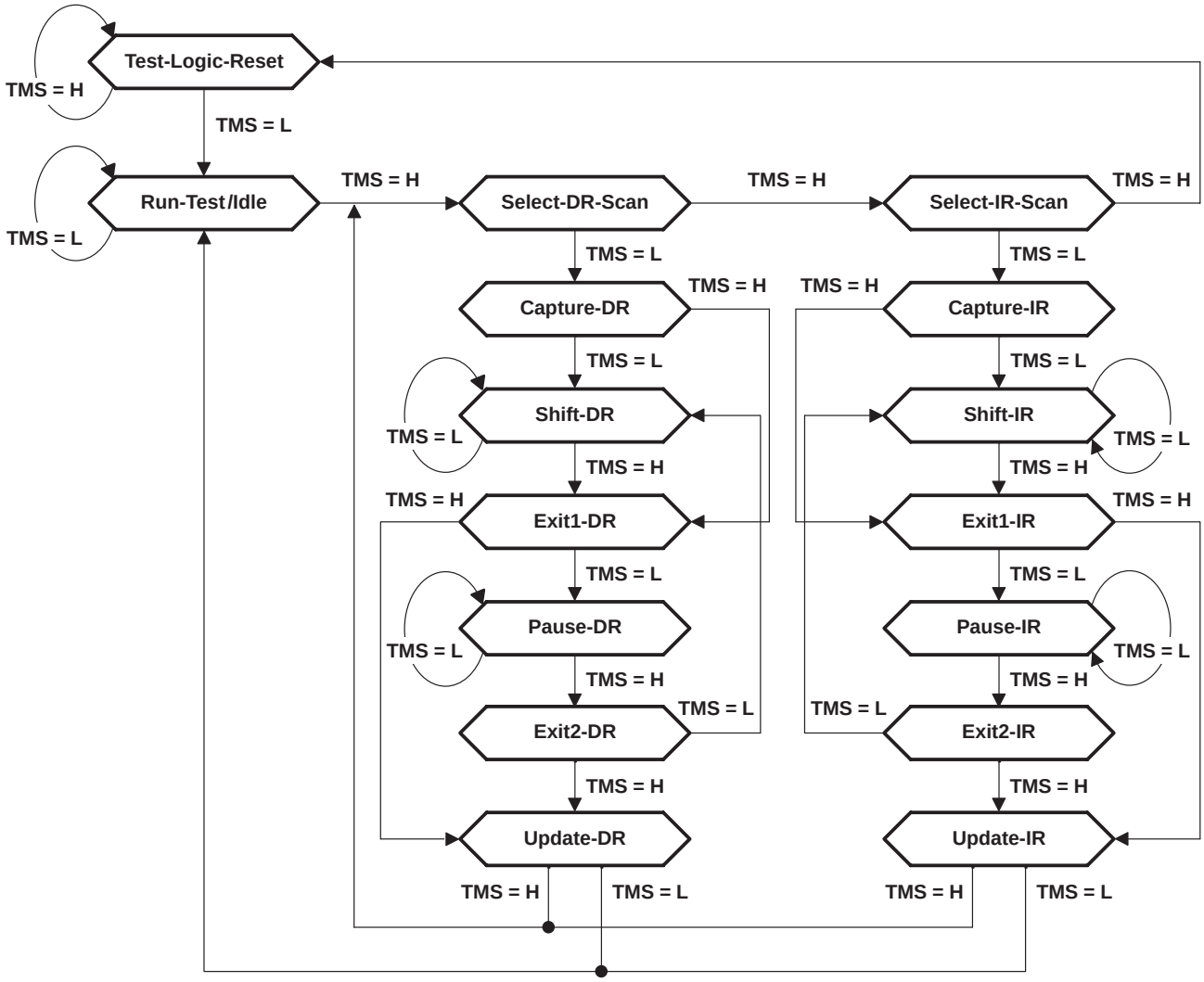

Figure 2. TAP-Controller State Diagram

The SCOPE bus-interface products conform to the IEEE 1149.1 protocol and perform the mandatory instruction of the document. Several other test operations, part of TI's SCOPE instruction set, are also implemented to greatly enhance the testing capabilities of the devices. The instructions implemented in the SCOPE bus-interface products, along with a brief description of the operation(s) performed, are as follows:

EXTEST	Data appearing at device inputs is captured. Data previously loaded into the output BSCs is forced.
SAMPLE	Data appearing at device inputs and outputs is captured without affecting the normal operation of the device.
INTEST	Data previously loaded into the input BSCs is applied to the device's internal logic, and the result is captured at the output BSCs. Data previously loaded into the output BSCs is forced.
BYPASS	The device operated normally and the 1-bit bypass register is selected in the scan path.
HIGHZ	Device outputs are placed in the high-impedance state.
CLAMP	The data in the BSR is applied to the functional inputs and forced from the functional outputs.

RUNT — This instruction tells the device to run the boundary test specified in the BCR. These tests include pseudorandom pattern generation (PRPG), parallel signature analysis (PSA), a combination PRPG/PSA, and output toggling.

READBN — The BSR is placed in the scan path, but no preloading of data takes place prior to shifting.

CELLTST — The contents of the BSC latches are inverted. This is a self-test feature that exercises most of the logic in the BSCs.

TOPHIP — The device outputs are toggled on each TCK falling edge.

SCANCN — The BCR is placed in the scan path. This operation is typically performed prior to loading the RUNT instruction.

For more information on the functional and parametric characteristics of SCOPE bus-interface products, see Reference 2.

When in the normal mode, the SCOPE bus-interface products provide high-performance, low-power, bus-interface functionality. Some of the performance highlights include:

- Fabricated in high-speed BiCMOS technology
- High drive (I_{OH} = –15 mA, I_{OL} = 64 mA)
- Low I_{CC} (<10 mA)

In addition to the ’BCT8244A, other devices incorporating the boundary-scan circuitry are available.

The pinouts for four SCOPE bus-interface products are shown in Figure 3. Note that the inclusion of the SCOPE circuitry requires the addition of four package pins to the devices, which correspond to the four signals of the test bus.

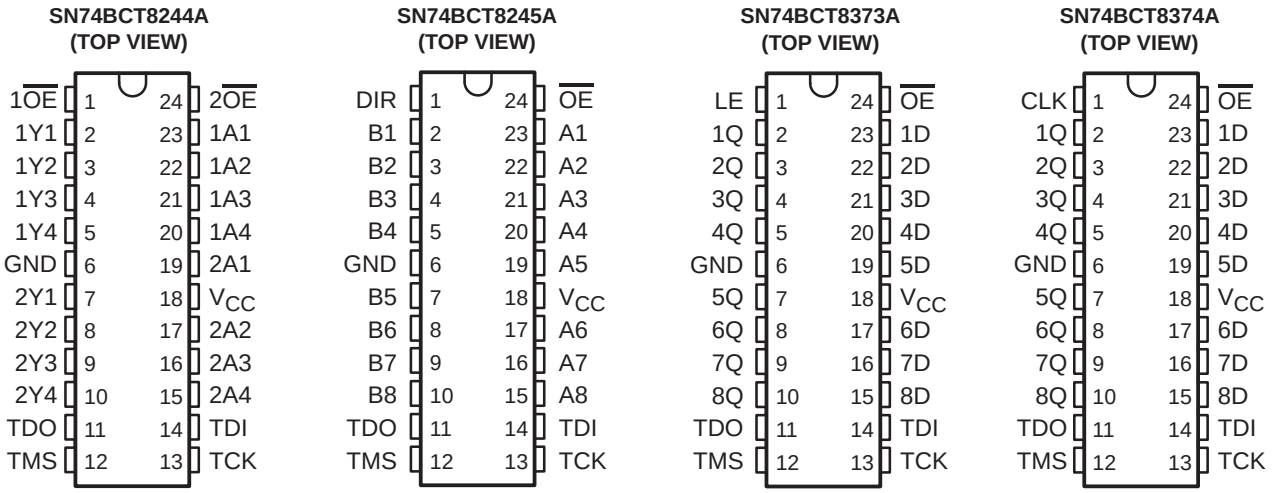

Figure 3. SCOPE Bus-Interface Product Pinouts

In the example of Figure 4, U1 is buffering data being received by U2. The scan path, in this case, consists of only U1 and U2, so U1’s TDI pin is driven by the controller’s TDO pin, and U2’s TDO pin drives the controller’s TDI pin. The integrity of the interconnect between U1 and U2 can be easily verified by using the ability of the SCOPE bus-interface products to force and capture data through the I/O periphery of the devices. The following procedure is one way (although not the only way) to verify that there are no opens or shorts between the outputs of U1 and the inputs of U2.

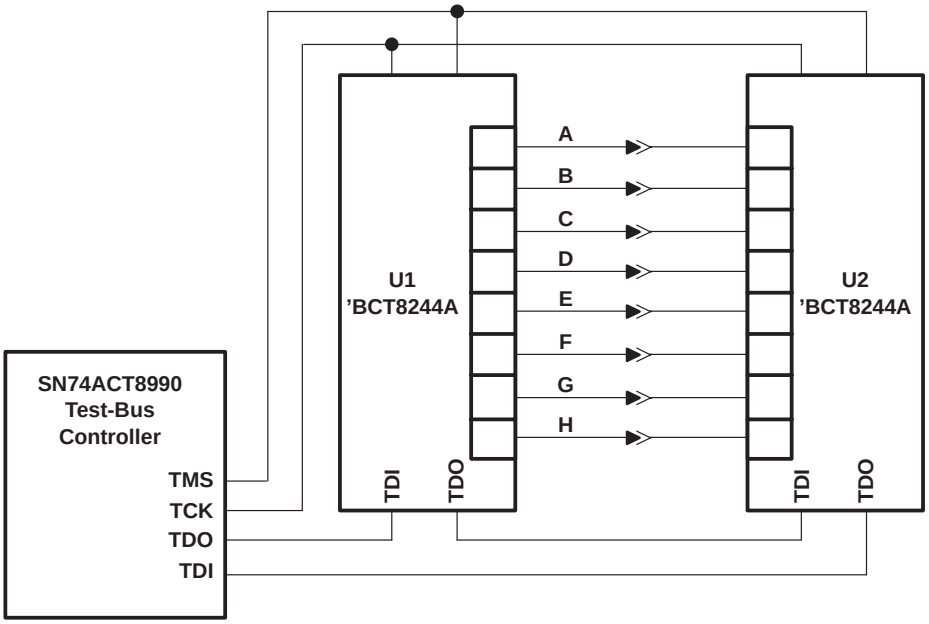

Figure 4. Simple Two-Device Scan Path

Using SCOPE Bus-Interface Products to Improve Testability

By placing SCOPE bus-interface products at critical system nodes and in key signal paths, the boundary-scan path can be used to observe and control the signals at a given point in the system. When the system is operating normally, the test circuitry is disabled and devices perform their normal function. During test operations, the device's I/O boundary is controlled by the SCOPE circuitry.

The procedure for implementing test functions varies according to the operation being performed. In general, the user will preload one or more data registers, execute an instruction via the IR, and capture a data register. The resulting data is then scanned out from the BSR for comparison with some expected value. The remainder of this paper provides some examples to illustrate how SCOPE bus-interface products can be used to build in testability in all areas of the product life cycle.

Verifying Wiring Interconnects

One simple example of how boundary scan can improve the testability of a system is to verify the wiring interconnects (detecting "stuck-at" faults) between ICs on a board or between two boards in a system as shown in the following steps. Figure 4 shows two 'BCT8244As being used to buffer signals between two separate parts of a system. They could be on either side of an edge connector, separated by PCB trace, or in any number of other configurations.

1. Initialize the scan path through a reset operation.
2. Scan all zeroes into the output BSCs of U1. This can be done with any of several instructions. ("Scan" means put the TAP in the appropriate shift state and serially load data through the TDI pin.)
3. Scan the EXTEST instruction into both U1 and U2.
4. Capture the BSR of U2.
5. Scan out for inspection the capture contents of U2's input BSCs, while scanning another pattern into the output BSCs of U1.
6. Repeat steps 4 and 5 for each of the patterns necessary to verify the interconnects.

Step 1 is accomplished by applying 10 V to the TMS pin, by scanning all zeroes into the BSRs, or by a power-down/power-up sequence. During step 2, load the output BSCs of U1 with the data that will be applied through the functional outputs.

The EXTEST instruction is loaded by putting U1 and U2 into the Shift-IR state (see Figure 2) and scanning the SCOPE opcode for EXTEST (00000000) into the IR of both chips. During the Update-IR state, U1 will force the data loaded in step 2 through its outputs. Since EXTEST places the BSR between TDI and TDO, going to the Capture-DR state (step 4) will load the input BSCs of U2 with the data appearing at its functional inputs.

In step 5, two things are done at once, using the Shift-DR state. The scan path is now 36 bits long, 18 bits from U1's BSR and 18 bits from U2's BSR. The data from the input BSCs of U2 is scanned out to be stored and/or examined. Since all zeroes from U1 were forced, the expected value of the data captured at the input BSCs of U2 is, also, all zeroes. During this same shift, the output BSCs of U1 are loaded with the next pattern. When passing through the Update-DR state after the shifting is complete, the output BSCs of U1 will force the new pattern through its outputs.

As an example, assume that some wiring defects in the circuit of Figure 4 have caused an open between U1 and U2 on signal C and a short circuit between signals E and F. Table 1 lists six patterns U1 could force to check for any opens between U1 and U2, or shorts between any two pins, and lists the data that would be captured by U2 (given the defects of this example).

Table 1. Shorts/Opens Verification

PATTERN	PATTERN FORCED BY U1 (A–G)	PATTERN CAPTURED BY U2 (A–G)†
1	00001111	00*1*01111
2	11110000	11110000
3	00110011	00110011
4	11001100	11*1*01100
5	01010101	01*1*10*0*01
6	10101010	1010*0*010

† Those bits indicating the presence of a defect appear in *italic* type.

Logic Verification

An application of two features of the SCOPE bus-interface products not included in the IEEE 1149.1 document is shown in Figure 5. PRPG and PSA can be used to verify the logic implementation of a design during the debug, prototype, or manufacturing test operations.

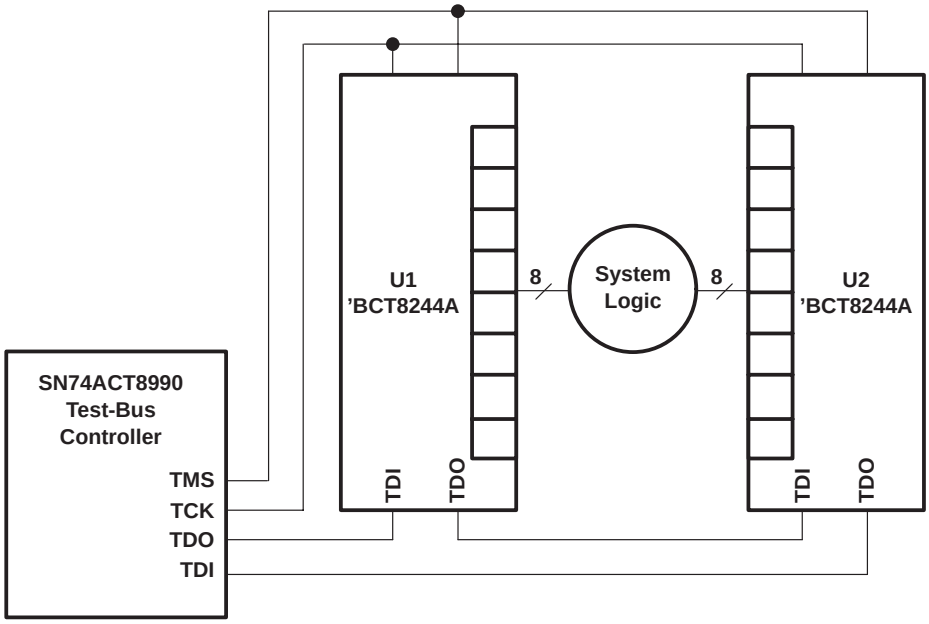

Figure 5. PRPG and PSA

During a PRPG or PSA operation, the BSRs of a device are configured as linear feedback shift registers. In this configuration, the registers perform either a pattern-generation or data-compression operation on each TCK cycle. By loading a known seed value (other than all zeroes) into the BSR and knowing the algorithm used, the user can determine the patterns that will be generated and/or the signature resulting from the data compression of inputs.

To exercise the system logic shown in Figure 5, the BSCs of U1 can be configured to output pseudorandom patterns and the input BSCs of U2 configured to compress data by performing the following operations:

1. Initialize the scan path.
2. Load the BSRs of both U1 and U2 with the seed values to be used during PRPG and PSA. Any value except all zeroes is acceptable.
3. Scan the SCANCN instruction into both U1 and U2.
4. Scan the PRPG code into U1's BCR and the PSA code into U2's BCR.
5. Scan the RUNT instruction into both U1 and U2.
6. Go to the Run-Test/Idle state of the TAP's state machine.
7. Execute the PRPG and PSA instructions for the desired number of TCK cycles.
8. Scan U2 with the READBN instruction.
9. Scan out the contents of U2's input BSCs and compare the resulting signature with the expected value.

To illustrate the concept, assume that a seed value of all ones is loaded into the BSRs of both U1 and U2 during step 2. Also assume there is no logic between U1 and U2 (i.e., U2's 1A1 = U1's 1Y1, U2's 1A2 = U1's 1Y2, etc.).

During steps 3–5 the SCOPE bus-interface products are loaded with the proper data and instruction to perform the pattern-generation and data-compression operations. SCANCN places the BCR between TDI and TDO so the simultaneous PSA/PRPG code (11) can be loaded using the Shift-DR TAP state. RUNT is loaded into the IR, and tells the SCOPE bus-interface products to examine their BCRs and run the test specified. In this example, the simultaneous PSA/PRPG function is being used. Figure 6 shows the algorithm used during the simultaneous PSA/PRPG operation.

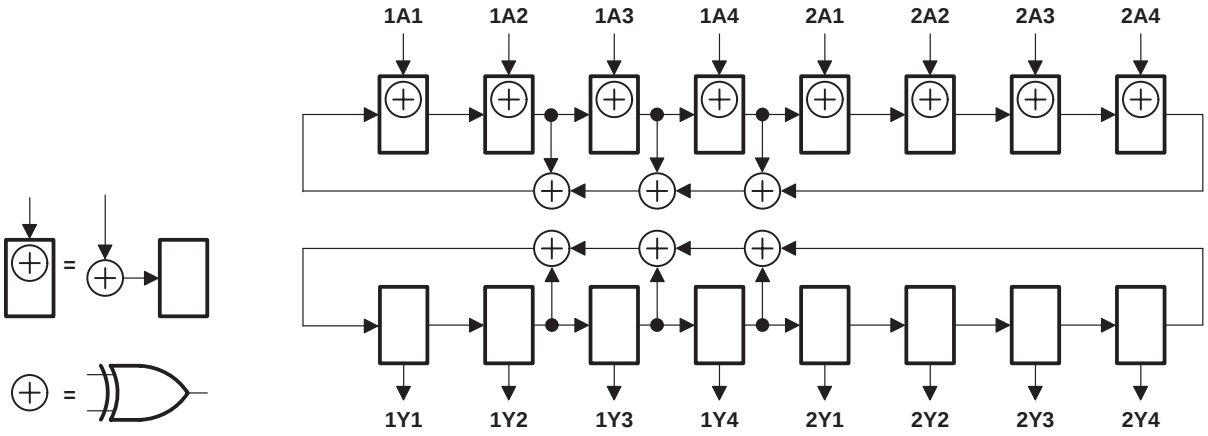

Figure 6. PSA/PRPG Algorithm of the ’BCT8244A

After generating sufficient patterns to test the logic, the signature in U2’s input BSCs must be examined. This is accomplished using the READBN instruction. It is important that this instruction, rather than EXTEST, INTEST, or SAMPLE, be used. This is because, while all of these instructions will place the BSR between TDI and TDO for the ensuing Shift-DR TAP state, other instructions will preload the input BSCs with the current input data during the Capture-DR state and overwrite the signature. READBN does not preload the BSR during Capture-DR, so the signature is preserved.

Table 2 shows the pseudorandom patterns generated, and the resulting signature after each pattern, for the first 15 TCK cycles of this verify-wiring example. The first pattern, applied during the falling edge of TCK in Update-IR, is the seed value of all ones. The first signature (generated on the first rising edge of TCK after entering Run-Test/Idle) is based on that seed value. On the first falling edge of TCK after entering Run-Test/Idle, the first pseudorandom pattern is generated and applied. The last signature is generated on the rising edge of TCK as the TAP state changes from Run-Test/Idle to Select-DR-Scan. Note that there is one TCK cycle (as the TAP state changes from Update-IR to Run-Test/Idle) in which no patterns or signatures are generated.

Table 2. PRPG/PSA Sequence

CYCLE	PATTERN AFTER TCK↓ (1Y1–1Y4, 2Y1–2Y4)	SIGNATURE AFTER TCK↑ (1A1–1A4, 2A1–2A4)
0 (seed)	11111111	10000000
1	01111111	00111111
2	00111111	10100000
3	10011111	01001111
4	01001111	01101000
5	00100111	00010011
6	00010011	00011010
7	00001001	10000100
8	10000100	11000110
9	01000010	10100001
10	10100001	11110001
11	01010000	00101000
12	00101000	10111100
13	10010100	11001010
14	11001010	00101111
15	11100101	11110010

Although this verify-wiring example (Figure 4) contains no logic between U1 and U2, the same principles are applicable in more complex cases. This method can also be used to verify address decoding to memories, or to apply patterns to the input of a complex ASIC. By placing SCOPE bus-interface products in the critical paths, the user can control the signals being applied to any node in the system.

System Partitioning

Using SCOPE bus-interface products to buffer key signals allows for the effective partitioning of a system or board during test to remove unneeded or unwanted components. Partitioning a system into separate standalone test cells reduces the number of patterns required to test the section(s) of interest.

In Figure 7, the SCOPE bus-interface products are used to partition a shared-memory configuration in which a digital signal processor (DSP) and graphics signal processor (GSP) use the same memory. The ’BCT8245As (U1 and U3) are used to buffer data transmission between the processors and memory, and the ’BCT8373As (U2 and U4) are used as address latches. The four SCOPE bus-interface products are connected in a serial scan path with common TCK and TMS signals.

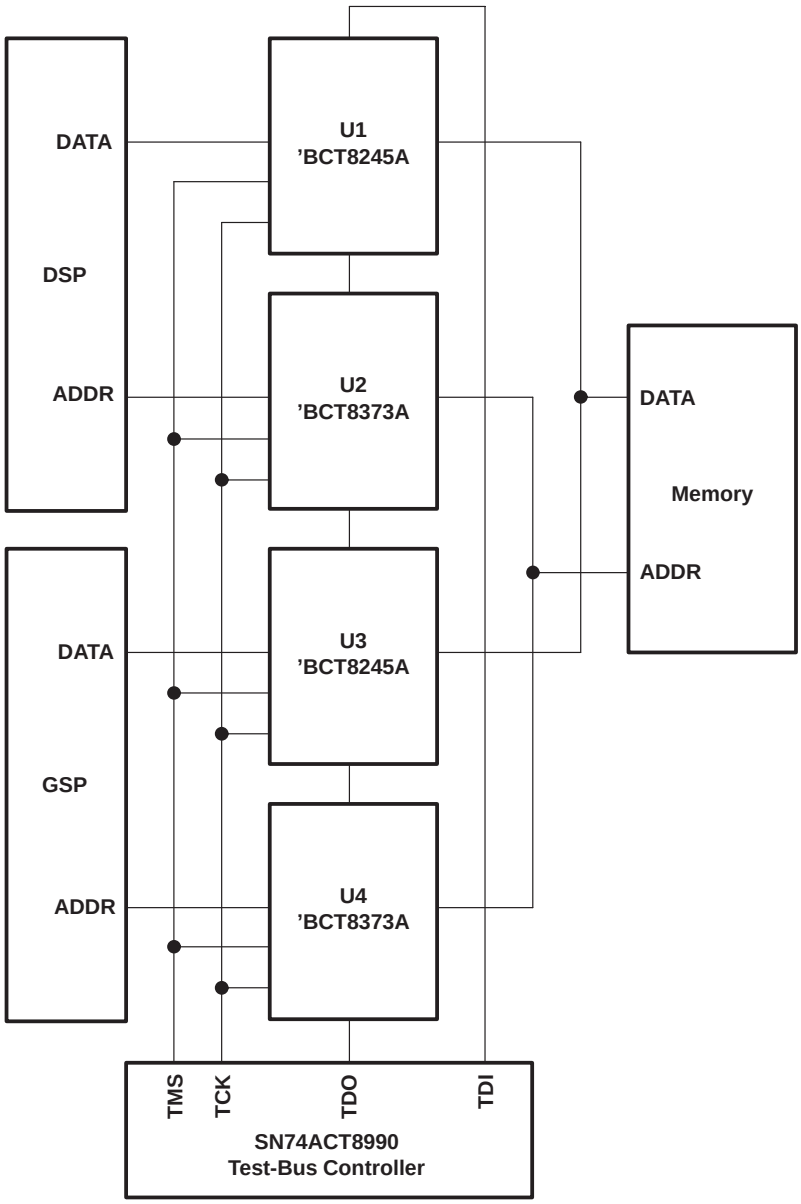

Figure 7. Partitioning a Shared-Memory Configuration

Using the SCOPE bus-interface products in this configuration creates many testing possibilities that go far beyond simple interconnect verification. For example, one or both processors can be effectively removed from system operation by scanning and executing the HIGHZ instruction on the units buffering it. This will cause the functional outputs of the SCOPE bus-interface products to go into the high-impedance state. This instruction also protects the SCOPE bus-interface products if some type of fixtured testing, such as bed-of-nails test, is to be used by ensuring that the functional outputs are not backdriven.

The status of the SCOPE bus-interface products can also be captured at any time by using the SAMPLE instructions. This operation does not disturb the normal operation of the devices and will not affect the system, but will capture the logic levels at all inputs and outputs. This information can be scanned out and compared to an expected value. The SAMPLE instruction is also useful for preloading the BSR prior to another test operation, since the SCOPE bus-interface products will continue to function in a normal mode during the preload scanning.

The circuitry in Figure 7 also allows the contents of any, or all, memory locations to be written to, or read from. A memory location can be verified by using the EXTEST instruction to force an address using U2 or U4 and capture the data appearing using U1 or U3. This illustrates the ability of the devices to both control and observe the signals to which they are connected. If the entire contents of the memory are known, the PRPG and PSA operations could unload the entire memory using a minimum number of clock cycles, with the final signature being scanned out for inspection.

Summary

Standard logic functions with boundary-scan circuitry provide a means for thorough testing of digital systems. By using a four-wire test bus compatible with IEEE 1149.1 protocol, the SCOPE bus-interface product family allows testability to be built into a system. Thus configured, the system will have complete controllability and observability of any node that can be addressed by a device in the scan path. Boundary-scan techniques are useful in system design, debug, and manufacturing test operations.

References

(1) *Proposed IEEE P1149.1, Standard Test Access Port and Boundary-Scan Architecture*, Draft D6, November 22, 1989

(2) *SCOPE Octal Data Sheets*

(3) Lee Whetsel and Greg Young, *Hardware Based Extensions to the JTAG Architecture*

Acknowledgment

The author of this article is R. J. Morgan.