BSM2: Next Generation Boundary-Scan Master

Frank P. Higgins
Bell Laboratories, Lucent Technologies
Princeton, NJ 08542-0900, USA
fphiggins@lucent.com

Rajagopalan Srinivasan
Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974-0636, USA
rajagopalan@lucent.com

Abstract

Boundary-Scan (B-S) strategies require successful coordination of B-S activities for the devices integrated on boards and systems. The original Boundary-Scan Master (BSM) chip was developed to achieve this coordination. We have recently designed the next generation version, named BSM2, that provides a more flexible architecture and a more complete set of features as compared to the original BSM. In this paper, we describe the architectural features of BSM2 device highlighting various scan modes, automatic scan sequencing and gated clock support. We also describe the verification and testing strategies for the design including the backward compatibility with the original BSM features. DFT structures are added to the device to achieve both high fault coverage and self-test capability, as this device forms the test conduit for board and system-level testing. Finally, we briefly discuss the device support for addressable scan port (ASP) protocol and its usage in DSP applications.

1 Introduction

IEEE Standard 1149.1 Boundary-Scan (B-S) Architecture [1] provides a standardized serial access interface to IC devices via the Test Access Port (TAP). Systems designed with B-S featured ICs are easier to test, debug and diagnose for anomalies due to manufacturing defects and design errors. Service processors — provided as part of the systems for testing, debugging and diagnosis — usually communicate with boards via parallel interfaces. The Boundary-Scan Master (BSM) device [2][3] was originally introduced to simplify a parallel-serial interface between a service processor and boards with B-S featured ICs. As B-S usage became popular, the B-S interface protocol evolved and expanded to other areas such as addressable scan port (ASP) protocol [4] and TAP-based digital signal processing (DSP) applications. Devices such as the embedded test bus controller (ETBC) [5] have also been designed to support some of these protocols and applications.

This paper describes a new ASIC device, Boundary-Scan Master 2 (BSM2) [6], recently developed as the successor to the original BSM device. The motivation for designing the new BSM2 device is multi-fold: (a) to incorporate new operational modes and features, (b) to address TAP-based DSP application areas, (c) to provide a higher-level user interface and (d) to improve performance specifications such as speed and power consumption as compared to the original BSM device. A service processor communicates with a BSM2 device and controls the test and diagnosis of unit-under-test (UUT), which could be a board or system. BSM2 serializes test vectors, delivers them to UUT using the standard protocol and stores the UUT response either as uncompressed data or a compressed signature. It incorporates an automatic test pattern generator (ATPG) to create test sequences for interconnect and cluster tests. The device solves the potential problems of bus conflicts and non-repeatable board-level signatures associated with pseudo-random testing of UUT implementing B-S architecture. BSM2 also provides support for edge-connector/backplane test, system test and diagnosis.

The BSM2 operates in two modes — an old mode (Compatibility Mode) providing backward compatibility with the original BSM, and a new mode (Advanced Operational Mode) exercising its newly designed features. While the old mode has 8-bit synchronous host interface, the new mode has the option of 8-bit or 16-bit asynchronous host interface. BSM2 is available as two devices — 497AE (compatible with original BSM device 497AA) and 1215E [6]. Both devices have BIST capabilities for both memory and random logic parts.

This paper is organized as follows. The BSM2 architecture and functional blocks are described in Section 2. The new features and operational modes are described in Section 3. The design verification process including the hardware/software co-verification is described in Section 4. DFT structures such as scan and BIST are highlighted in Section 5. Section 6 discusses the BSM2 applications and the conclusion is provided in the last section.

0-7695-0613-5/00 $10.00 © 2000 IEEE

Table 1: BSM2 Features

Category	Device 497AE	Device 1215E
Operation Modes	1. 497AA Compatibility Mode 2. Advanced Operational Mode	Advanced Operational Mode
Host Interface	1. Old 8-bit synchronous interface 2. New 8-bit asynchronous interface	New 16-bit asynchronous interface
Register Access	Indirect Addressing Method	Direct Addressing Method
Voltage & Frequency	3.3 V supply & 65 MHz Clock	3.3V supply & 65 MHz Clock
Technology	Lucent 0.35 μ CMOS process	Lucent 0.35 μ CMOS process
Package	28-pin SOJ package	48-pin TQFP package

2 BSM2 Architecture

The BSM2 operates as a parallel-to-serial interface, converting high level commands/data from a host processor to serial commands/data for controlling a TAP device. The device architecture with functional blocks is illustrated in Figure 1. The device features for various categories are listed in Table 1. The BSM2 interfaces with the host processor via address and data busses and their associated control signals. It communicates with the TAPs residing in UUT via the B-S test-bus. The main functional blocks of the device include a host interface, device controller, test data memories, automatic TMS generator, TMS macro generator, TCK generator and control gate, ATPG and SA registers, internal registers and counters. Each of these blocks are described later in more detail.

Two 8K bit memories — test vector input (TVI) and test vector output (TVO) — buffer the input and output data from the TAP port. The test clock (TCK) is derived from an external master clock and the design is fully synchronized with this master clock. The test reset signal (TRST) is controlled by an internal register. The test mode select (TMS) signal is from one of three sources — TMS macro generator, automatic TMS generator, or internal register for use of non-standard protocols. The test data output (TDO) signal is also from one of three sources — TVO memory, automatic test pattern generator (ATPG) or test data input (TDI) signal. The test data scanned into the device through TDI can be either stored in TVI memory or compressed in signature analysis register (SAR). Any of these possible operational modes can be programmed through the control registers.

The asynchronous host interface consists of a bidirectional data bus, an address bus and control signals. During normal operation, the TVI (TVO) memories behave as FIFOs and store (provide) data via the TDI (TDO) ports.

The TMS signal controls the operation of TAP state machines in the B-S devices driven by BSM2. The automatic TMS generator provides a high level interface for this protocol. The user selects a scan mode defined by *scan state, idle state* and *destination state*. The *scan state* is defined as the state where data are to be scanned in/out of the scan chain. The *idle state* is defined as the state where no data are scanned but the state machine remains in this state for a programmed number of TCK cycles. The *idle state* and *destination state* can be any one of the four stable TAP states — *Test-Logic-Reset, Run-Test/Idle, Pause-DR* or *Pause-IR* states. The *scan state* can be any one of six TAP states — *Test-Logic-Reset, Run-Test/Idle, Shift-DR, Pause-DR, Shift-IR* or *Pause-IR* states. Upon execution, the automatic TMS generator creates TMS sequence to move TAP state machines from their origin, to states defined by the *scan state* and *idle state*, and finally to the *destination state.*

During the state sequencing, two counters — *scan counter* and *idle counter* — are monitored. When the state machine reaches the idle state, the auto TMS generator will hold the state machine in this state until the *idle counter* reaches zero. Similar operation occurs with the scan state and the *scan counter*, except that at the scan state data is scanned through the scan chain. The device supports the following five scan modes:

- *stimulus-only mode:* stimulus stored in TVO memory is sent out sequentially via TDO to the external device.
- *response-only mode:* response from external device is read in via TDI and stored in TVI memory.
- *normal mode:* combination of the above two modes.
- *recirculate mode:* response read in via TDI appears as stimulus on TDO without being stored in TVI memory.
- *recirculate with response-only mode:* same as above mode but the response is also stored in TVI memory.

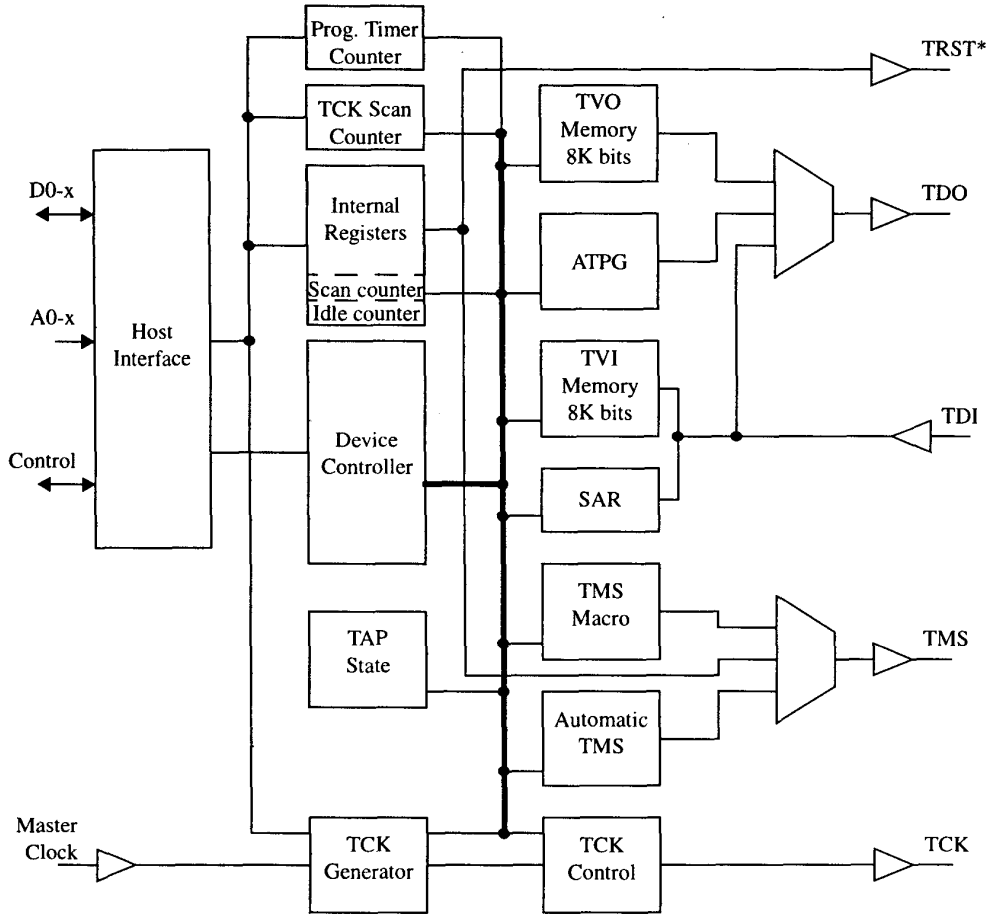

Figure 1. BSM2 Architecture

To accommodate overflow and underflow conditions of the TVI/TVO memories, the user can select from either *gated-TCK* or *auto-pause* mode. In the *gated-TCK* mode, TCK is switched off until the condition is corrected. In the *auto-pause* mode, the automatic TMS generator directs the TAP to a pause state until the condition is corrected. This is applicable when the scan state is either *Shift-IR* or *Shift-DR*.

BSM2 supports an ATPG mode for board-level interconnect test and cluster testing. The device is capable of generating the following sequences in ATPG mode:

- *walking one/zero sequence:* for interconnect testing and diagnosis.
- *up/down counting sequence:* for generating the shortest test length.
- *pseudo-random sequence:* for cluster test and testing the internal logic of devices that do not have BIST.
- *constant one/zero sequence:* for scanning BYPASS/EXTEST instruction into the devices in B-S chain.

We shall next provide details about the functional descriptions of the main functional blocks.

2.1 Host Interface

The host processor interface provides generic control functions including chip enable, read/write control, data available/valid and interrupt signals. The 28-pin (48-pin) package has an 8-bit (16-bit) data bus allowing byte (word) read/write access to the internal registers. The 28-pin package is address pin limited, and thus uses indirect addressing of the internal registers. In the 48-pin package all internal registers are directly addressable.

2.2 Device Controller

The device controller provides high-level coordination and synchronization of the various modules based on the

selected mode of operation. All registers and memories are held in a static state until one of the three command states — *execute, jump*, or *reset* — is selected. In the default (*auto-pause*) mode, TCK is free running. In the *gated-TCK* mode, TCK is disabled until the execution is enabled.

2.3 TVI/TVO Memories

Each of the two data memories (TVI and TVO) consists of 8K bits organized as 512 16-bit words. During normal load-scan-unload operations, the memories behave as FIFOs and hence it is not necessary for the host processor to specify individual addresses for read and write operations. However, these FIFOs can be reset by writing to their corresponding address pointer registers. During normal operation, TVI (TVO) receives (transmits) the test response (stimulus) data that are scanned from (into) a selected B-S chain. Note that because these memories behave as FIFOs, the user has virtually no limitation on the data scan length. The only limitation is from the internal scan counter (32 bits).

During ATPG mode, the test data memories do not operate as FIFOs and each memory location can be addressed individually for read and write operations. During ATPG mode, the test stimulus is provided by the test generation hardware and TVI/TVO memories are used to store board interconnect information to avoid bus conflicts during test application. In this application the scan lengths are limited by the memory size (8K).

2.4 TCK Generator and Gating

The TCK generator implements clock divider circuitry to derive the TCK signal for a selected TAP from the master clock input. The TCK generator can be programmed to divide the input master clock by 2^n where n can be an integer from 0 to 7. In the default state, TCK is free running (*auto-pause* mode) with a clock signal obtained from the input master clock after a division-by-128. This block also controls the gating of the TCK signal when in the *gated-TCK* mode. TCK then operates only when scanning or moving from one of the TAP states to another. TCK is gated off when the TAP state machine reaches the destination state or when a data overflow/underflow occurs.

2.5 TMS Generator

In the normal (non-manual) mode, the TMS signal pattern is generated internally based on: (a) the current state, (b) the desired scan operation, (c) the desired idle operation and (d) the destination state. In the *auto-pause* mode, whenever data overflow/underflow occurs in the *Shift-IR/DR* states, TMS generator commands the TAP controllers in the selected B-S chain to automatically move to the corresponding *Pause-IR/DR* states. When the overflow/underflow condition is corrected, the TMS pattern is generated to move the TAP controllers back into the *Shift-IR/DR* state.

2.6 ATPG and SAR

ATPG and SA registers are used to test: (a) interconnect test between UUT devices, (b) internal logic of UUT devices and (c) cluster test of non-B-S devices via the B-S paths of devices that surround the non-B-S devices. While carrying out interconnect testing, it is possible that conflicts could arise if multiple devices on a single net are enabled simultaneously. BSM2 avoids this situation by using a concise form of interconnect information to inform the ATPG function which B-S cells are at device inputs and outputs and which must be held at constant value to enable I/O control drivers.

The device also resolves the potential problems of non-repeatable board-level signatures associated with pseudo-random testing. The responses to the pseudo-random test sequences generated by ATPG register are compressed to a signature by SAR. The SAR is enabled only after the first test vector is scanned out of the selected B-S chain. This ensures that the unknown values that are scanned out when the first test vector is scanned in do not corrupt the signature. Also, after the last test vector is scanned in, another test vector is shifted in to scan out the last response. These operations are performed automatically by BSM2 and are transparent to the user.

3 BSM2 Modes and Features

The features on the device can be classified under Compatibility Mode and Advanced Operational Mode. Although the two modes have different behavior, many of the internal modules are shared between both modes. BSM2 encompasses all of the features of the original BSM (such as supporting ATPG modes) and ETBC features [5] such as supporting addressable scan port (ASP) protocol. BSM2 architecture provides the capability of supporting various applications as described in Section 6.

3.1 Compatibility Mode

Features under the compatibility mode include:

- *Register Access:* Among the 18 internal registers — 6 control/status registers and 12 data registers — only the *control/pointer register* (CPR) is directly accessible and all other registers are indirectly accessed via CPR.
- *Memory Access:* TVI/TVO memories are accessed in random-access or FIFO mode.
- *Bus Operations:* Synchronous/Asynchronous to clock.
- *Scan Mode:* Supports only the normal scan mode.

- *ATPG Mode:* Generates walking, counting, pseudo-random and constant output sequences.
- *TMS Macro Generator:* TMS sequence coded by defining bit patterns in TMS macro generator register.
- *Scan Sequence Modification:* Modifies the test sequence to ensure that no bus conflict occurs before being passed to the B-S chain.

3.2 Advanced Operational Mode

Features under the advanced operational mode include:

- *Register Access:* All of the internal registers are indirectly accessible in 28-pin package and directly accessible in 48-pin package.
- *Memory Access:* TVI/TVO memories are accessed in FIFO mode only.
- *Bus Operations:* asynchronous to master clock for both 8-bit and 16-bit data address busses.
- *Scan Mode:* Supports all the five scan modes.
- *Gated-TCK and Auto-Pause Modes:* Supports these modes to accommodate overflow/underflow conditions.
- *Automatic TMS Generator:* TMS sequence is automatically deduced from scan, idle and destination states.
- *Re-timed Boundary-Scan Link:* Supports pipeline delays in the scan chain across the board/system.

4 BSM2 Verification

4.1 Functional and Timing Verification

The behavioral specifications were fully verified first at the functional module level and followed up for the integrated device. Functional tests were formulated at the RTL for individual modules and applied via an RTL simulator (MTI) [7]. Upon successful RTL verification, the modules were integrated and the RTL descriptions were synthesized to a gate-level netlist.

Functional tests for the device were generated by invoking various Shell/Perl scripts constructed from hierarchical macros. These tests verified both the new and old features described in Section 3. The resulting test vectors were automatically converted to a testbench and applied to both RTL simulation (MTI) and gate-level simulation (IKOS) [8]. While the RTL simulation verified only the functionality of the device, the gate-level simulation verified both the functional and timing aspects of the design. The timing specifications for the integrated BSM2 device were also verified using Lucent's internal static timing analysis tool.

4.2 Hardware and Software Co-Verification

We have described a limited number of operational modes of the device in Section 3 and it is beyond the scope of this paper to describe all of its operational modes. The possible combinations of these operational modes are quite extensive making it difficult to verify the full device functionality using traditional test vector generation methodology.

In addition, we have an existing software suite (C-based programs) and hardware design for the current BSM device. We tested the new design against this known reference to more fully exercise all the backward compatible modes. To achieve this goal, we used a hardware and software co-simulation tool to integrate the RTL code of the BSM2 device with a virtual software processor and communication bus. The virtual processor was driven by the existing software base to verify both device-level and board-level operations. We also developed new software program drivers for exercising the BSM2 in its new mode. This served two purposes: a) it provided a more complete verification of the various combinations of operational modes of the device and b) it served as a software development environment for creating and testing code for the new operational modes prior to the availability of the fabricated device. It is important to note that this HW/SW co-verification was intended only for functional verification.

5 BSM2 Testing

BIST for the BSM2 device must be clearly distinguished from system/board-level BIST that can be carried out (using ATPG mode) or initiated through B-S chain (by scanning RUNBIST instruction into UUT connected on B-S chain) using the capabilities of BSM2. Since BSM2 is designed to enhance board/system-level testability, it follows that the quality of BSM2 device must be very high.

The data memories (TVI and TVO) and the random logic are designed to have BIST capabilities to ensure devices with high quality and reliability. The BIST hardware is not only run during device manufacturing test, but also can be exercised in the field within a system test paradigm. Memory BIST ensures 100% fault coverage over an extended fault model that is well beyond the single stuck-at fault model. The core portions of the random logic include scan-based BIST providing 96% of single stuck-at fault coverage. All of the BIST activities are coordinated by an on-chip BIST scheduler.

Two full-scan paths are provided to scan all register elements in the design including those that are not covered by BIST structures. Functional and scan vectors were used to fault grade the portions of the core logic that were not

covered by BIST structures. Scan and Logic BIST were exercised at 40 MHz instead of the system clock frequency of 65 MHz to avoid failures due to the activation of functional multi-cycle and false paths. The test frequency (40 MHz) was determined by timing simulation. Memory BIST could be run at functional speed of 65 MHz.

Sample experiments were run during the manufacturing process to test for path delay faults on the BSM2 device. Functional patterns that were originally generated for design verification (also included in test) were used for grading the delay faults [9]. About 165K paths were tested for delay faults using 453K patterns. The longest tested path had 58 gates compared to the longest physical path comprising of 74 gates.

6 BSM Applications

6.1 Addressable Scan Port (ASP) Protocol

The ASP protocol [4] was developed for various applications interconnecting a BSM among several UUTs, such as a multidrop backplane test configuration. The protocol requires scanning short bursts of data (< 60 bits) during one of the four stable states: *Test-Logic-Reset, Run-Test/Idle, Pause-DR* or *Pause-IR*. This protocol is easily achieved with the BSM2 device by programming the scan state register. However, if in the *auto-pause* mode, the data must be preloaded into the TVO memory to avoid the underflow condition. The BSM does not encode/decode the bit-pair protocol and it must be done by the host processor.

6.2 Digital Signal Processing (DSP) Application

The TAP controller can be used not only for traditional test functions, but also for accessing and controlling DSPs. For Lucent's DSP1600 family [10], a full suite of software and hardware were built around the BSM devices. The suite provides support for:

- Complete DSP software development environment.
- Selection of specific DSP device on the active B-S chain.
- Download/upload internal/external RAMs and ROM segments of DSPs.
- Start execution from internal/external RAM/ROM.
- Synchronous start/stop operation of DSPs.
- Initialization of TAP controllers and verification of the B-S chain connectivity.

For such applications, efficient and continuous data flow through the TAP controllers is critical and BSM2 architecture supports this requirement.

6.3 System-on-Chip (SoC) Designs

Current SoC designs are implementing pre-laid out processor cores with TAP interfaces. The TAPs of these embedded processors are interconnected similarly as at the board level and brought out to a single device TAP. This TAP can be controlled by the BSM2 to deliver the necessary protocols to provide access to the internal cores. TAP does not necessarily only serve the test function, but also can provide a wide variety of core-functionality as defined by the designer.

7 Conclusion

We have described a new Boundary-Scan Master device and its new operational modes. It provides a more flexible architecture and higher level user interface for board/system designers as compared to the old device. We have provided details about the verification and test process during development and manufacturing phases of the device. We anticipate that BSM2 will meet the needs of the current and evolving B-S applications.

Acknowledgments

We would like to acknowledge Ian Bardsley, Sudipta Bhawmik, Gary Heyer, Ilyoung Kim, Paul Murphy, Bradford Van Treuren for their contributions to BSM2 project and Vishwani Agarwal for reviewing the final manuscript.

References

[1] IEEE 1149.1 Standard Test Access Port and Boundary-Scan Architecture, IEEE Computer Society, October 1993.

[2] N. Jarwala and C. W. Yau, "The Boundary-Scan Master: Architecture and Implementation", Proc. of European Test Conf., April 1991, pp. 1-10.

[3] AT&T Boundary-Scan Master User Manual, MN92-012BSM, June 1993.

[4] TI Addressable Scan Ports — SN54ABT8996 and SN74ABT8996, Texas Instruments, December 1996.

[5] TI Embedded Test Bus Controller — SN54LVT8980 and SN74LVT8980, Texas Instruments, February 1997.

[6] Lucent Boundary Scan Master 2 User Manual, 497AE and 1215E — http://www.lucent.com/micro/dti/bsm2.html

[7] Model Technology Inc. website — http://www.model.com.

[8] IKOS Systems Inc. website — http://www.ikos.com.

[9] C. G. Parodi et al, "A Non-Enumerative Path Delay Fault Simulator for Sequential Circuits", Proc. of Int'l Test Conf., October 1998, pp. 934-943.

[10] Lucent Digital Signal Processors website — http://www.lucent.com/micro/dsp.