

Scan-Based Design Verification – An Alternative Approach

by Pete Fleming and Don McClean

This paper was presented at the ATE and Instrumentation Conference West, 1990.

Introduction

Electronic digital design has been characterized by rapidly increasing performance and functional density at reduced cost. End-users have realized these benefits in the form of smaller, more-reliable, more-capable products with sharp cost drop-offs over time. Continued advances in digital technology will require further miniaturization of geometries that will yield diminishing returns as we outpace the capabilities of debug and test equipment.

Present-day board designs are incorporating surface-mount technology to increase the equivalent functionality per unit area. Manufacturers are moving toward slim-profile packages and attempting to reduce the space between parts to pack more "power per picoacre." Pin-to-pin spacing on catalog devices is approaching 25 mils, with much smaller spacing on many custom and high-pincount ICs. Tape-automated bonding will reduce geometries further, permitting incredibly dense circuit cards to exist. More and more use of complex submicron Application-Specific Integrated Circuits (ASICs) and custom devices is occurring.

While these trends help produce smaller, better products, they work directly against the objectives of validating the design and proper manufacture of the product itself. As ASICs absorb the functionality of multiple devices, entire sets of bus and control signals disappear into the internals of ICs. The complexity associated with validating the design functionality of an IC or board increases exponentially. The geometries of boards defy the state of the art in fixturing technology, including the use of handheld and clip-on probes.

The result has been major concern over continued use of existing approaches to observe and control digital designs. Anticipated benefits in cost and performance are endangered by escalating test costs and product development cycles.

Test Standardization Efforts

In response to the emerging crisis, test engineers in Europe mobilized in 1985 to form the European Test Action Group (ETAG). Initiated by Phillips, this ad hoc organization began promoting a test technique for manufacturing test called "boundary scan." Boundary scan attempts to overcome the loss of physical access (especially for "bed-of-nails" fixtures) by embedding virtual test points around the periphery of a chip, hence, the name boundary scan.

ETAG began to approach the major silicon suppliers with their proposal, soliciting support in addressing the emerging test problem. As more non-European companies put their weight behind the proposal, ETAG changed their name to the Joint Test Action Group (JTAG). Major involvement from leading semiconductor manufacturers, such as Texas Instruments and Motorola, established JTAG as a credible force in advancing test technology. Simultaneously, an IEEE-sponsored effort to establish standard test buses known as IEEE P1149 emerged in the United States.

When the boundary-scan technical proposal¹ had secured the endorsement of dozens of major electronics firms, JTAG approached the IEEE in an attempt to formalize the ad hoc effort. JTAG was folded under the umbrella of P1149, which subsequently migrated into a series of coupled, but independent, bus definitions known as P1149.n. The furthest defined was the JTAG bus, which was designated IEEE 1149.1. The boundary-scan proposal evolved through several draft revisions before going to ballot in August 1989.² The result was an overwhelming endorsement for 1149.1, which achieved formal IEEE approval in early 1990.

Boundary Scan

The goal of boundary scan is to regain lost visibility and control of designs through the inclusion of additional test logic impacting the input/output (I/O) pins of devices. Scannable

flip-flops are multiplexed onto the IC functional data paths, allowing signals to be observed or to be brought to a known state via a four-wire interface.³ 1149.1 standardizes the four-wire interface to ensure interoperability of devices from multiple vendors (that is, one can scan through a Motorola part to access logic in a TI part). 1149.1-compliant parts can sample or drive their I/O pins to support testing of both the board and the ICs.

In an external test mode, the outputs of a device are controlled to desired states while the inputs of neighboring devices are observed. In this context, neighboring means that two devices are interconnected by a common signal, not that the devices are physically adjacent. The external test mode allows test vectors to be scanned in and out to verify the proper interconnect of ICs on the board. This also allows the testing of non-scannable logic clusters surrounded by scannable devices.

In an internal test mode, the internal silicon is isolated from input pins while test vectors are propagated across it to the output pins. This allows testing of the device logic to the extent that adequate patterns can be devised for application from the I/O points. The 1149.1 architecture supports interfacing to internal scan and other optional test data registers to assist in testing complex ICs.

The boundary-scan architecture also supports a bypass mode that abbreviates the scan path to a single bit when its boundary scan registers or optional data registers are not accessed. An optional device identification register may be included. Other capabilities can be accommodated easily within the boundary scan framework, such as Built-In Self-Test (BIST), pseudo-random pattern generation, signature analysis, and more.

1149.1 devices dedicate four pins to support the industry standard bus. The 1149.1 bus consists of four wires—two to control the scan state of the devices and two to transmit the serial data. These four wires are routed to all scannable devices on the board.

Conventional Debug Techniques

Designers face the challenge of validating the design of their boards (or ICs) even in the face of high complexity. Standard processes such as visual inspection, ohming out interconnect, and verifying power and ground are complicated by the tight geometries, but the real penalty occurs during attempts to confirm functionality. A wide variety of techniques is used to debug designs, but virtually all of them require external instrumentation that connects to or probes the board under test.

Multimeters or logic analyzers are used to determine logic levels for digital signals and are necessarily constrained to observing I/O level signals. Word generators are connected to buses to apply controlled data streams for test. Bus analyzers may be used to monitor states of standard bus protocols, and in-circuit emulators allow control and debug of microprocessor-based designs.

Reliance on these tried-and-true instruments has two disadvantages in the light of advanced packaging techniques.

First, tight geometries and high-speed signals are not amenable to probing and clipping.⁴ Engineers and technicians cannot probe fine-pitch designs reliably unless techniques such as stagger pads are used. Frequently, key signals are embedded within the internal logic of a device, inaccessible to contact-dependent instruments. High-speed signals, such as processor signals present in emulator cables, are subject to cross talk and glitching because of the length and layout of the cables. Second, design validation often requires monitoring of the design in unique environments, such as in the end-use environment or in temperature chambers. This usually requires the electronic subsystem to be closed up where access is limited to connectors on the chassis or case. Conventional instruments cannot be used to probe the design.

In these cases, the visibility and control permitted by scannable designs offer alternative techniques for logic validation.

Scan-Based Debug

Emergence of IEEE 1149.1-compatible components will provide an infrastructure of control and visibility with improved capabilities for multiple engineering disciplines. To harness these capabilities, an environment must exist for controlling and manipulating the scan-accessible features. Additionally, designers must not be overwhelmed by the details of the scan itself.⁵

Design validation always has been viewed as a parallel operation whereby the states of multiple signals are changed or viewed as a single event. This parallel- or register-level view of the design is the one with which designers feel comfortable and pervades existing computer-aided engineering (CAE) tools such as logic and fault simulators.

A Practical Experiment

Texas Instruments developed a system-level test bed using the 1149.1 architecture across four printed wiring boards (PWBs) using the scan for debug and integration. Manipulation of the scan was performed using a Scan Control System (SCS). The

particular tool used for this effort is called the Advanced Support System for Emulation and Test (ASSET). ASSET is a PC-based tool that builds a data base representation for tracking and controlling the state of the scan architecture. This allows the burden of dealing with the serial view and current state of the scan paths to be relegated to the computer. ASSET uses configuration files to determine the topology of the scan paths and interfaces to the unit under test via a controller card in the PC expansion slot. ASSET drives the 1149.1 protocol.

An SCS can provide a wide range of capabilities, but the conversion of the serial view to a parallel one is the key feature. The SCS associates functional signals with scannable locations and serves as a "test operating system" to bring functional signals to a known state simultaneously, similar to a conventional parallel process. For example, if one desires to set an address bus to the value >38, certain functional nodes comprising the address bus must assume a specific logic level. A word generator would clip on to the appropriate devices and simultaneously drive the eight address lines to the required value. In contrast, the SCS uses its knowledge of where the same eight signals reside on the scan path to scan them into the desired value and activate them on a common clock edge.

The actual process involves selecting or deselecting the appropriate boundary-scan registers, issuing the command to the affected devices, and scanning in data values in the proper order. The SCS uses a configuration file that acts like a street map, directing the information to the right mailboxes.

The SCS allows signals such as the address bus to be grouped functionally. With the proper user interface, the designer has a PC-based capability to set registers and buses to the value of choice or to sample the value of these elements. Internal signals are equally accessible as long as they reside on the scan path. The power of the computer allows "programming" of the stimulus steps and automated capture of the data for review at the designer's convenience.

Flow

The ideal approach to implementing the debug process was to have the designer develop the debug procedures while the board was being fabricated. The designer was most knowledgeable of the design of the board, including the partitioning of the functions within the design and how the scan architecture was implemented. The designer performed or assisted in the definition of the configuration file that identified the scan topology to the SCS. Next, desired functional capabilities were identified based on the board design.

Example capabilities included:

- Read memory value
- Write memory value
- Initialize VME interface devices
- Reset board
- Write to speech processor

Subroutines or utilities were created with appropriate user interfaces to raise the level of abstraction from serially oriented bit manipulation to register- or bus-level transactions. The engineer would specify the address in memory to write to and data to be written, and the SCS would handle the rest during runtime. Individual subroutines (such as read/write memory) could be nested further to build upload and download routines.

The designers performed typical prechecks of their boards and use the scan to weed out manufacturing errors wherever possible. The functional subroutines then were used to validate the logic design.

Structural Versus Functional Verification

Structural vectors make no attempt to use the hardware in its functional mode, but focus in on toggling signal states to detect classical manufacturing faults (stuck-ats, shorts, opens). The use of the scan greatly simplified this task, eliminating one of the most frustrating aspects of board debug. When a first-pass design fails to perform as expected during verification, two possible causes exist. First, the designer may have made an error in the design of the logic. The primary goal of design verification is to identify and fix these problems. A second possibility exists that the hardware under test has a manufacturing error rather than a design error. Unfortunately, during the initial debug process, neither the design nor the manufacturing database has been validated. Thus, designers often spend hours re-examining their schematics and simulations in search of nonexistent design errors that turn out to be solder splashes or wirewrap miswires.

Before using the functional routines, the interconnect was exercised in an attempt to locate any manufacturing errors. The ability to scan to the appropriate devices first was verified to eliminate any uncertainty in the results. Any problems existing with the configuration file or the actual hardware implementation of the scan were eliminated. The verification of the interconnect was a straightforward task involving only basic knowledge of the board's functional topology. The scan allowed etch or wiring errors to be located easily for signals driven and monitored via the boundary scan.

The functionally-oriented subroutines then were used to exercise the board logic. The sequencing of signals emulated the

normal logic operation to the maximum extent. The tremendous flexibility of these subroutines provided a highly robust debug environment. One clear advantage of the scan was the ability to select or deselect signals for stimulus or monitoring without the need to rewire or move a probe clip physically. This process, which can require 10 minutes using a logic analyzer, was accomplished in less than 1 minute. The number of signals to be controlled or viewed simultaneously was unlimited, as opposed to use of multiple analyzers or multiple passes on a single analyzer moved around the design.

Processing of the results was a mixture of interactive interpretation of the values displayed on the screen with hardcoded compares of scanned-out values.

The Debugger

TI's SCS, ASSET, has a very valuable feature called the debugger mode. The debugger is a highly interactive mode, allowing the engineer valuable control over the hardware in a debug environment. One function of the debugger provides control over the SCS functions themselves, such as changing the scan clock speed or enabling/disabling ASSET's two runtime checks. ASSET provides the ability to verify the integrity of the scan path during each instruction scan to identify any potential corruptions. It also provides a prescan check to prevent any user-identified conditions from occurring. For example, via the chain scan, one could enable conflicting buses simultaneously, potentially damaging the board. The prescan checks would prevent such states from being scanned into the hardware if they had been identified as illegal conditions.

A more powerful function of the debugger is the view function. In view mode, the engineer can open windows for each scannable part. Via these windows, instructions can be issued to individual parts to manipulate the scan-accessible registers. This allows a robust interface to monitor signal states, set register values, turn on pattern generators, develop signatures, or any of the test capabilities one can layer over 1149.1. The windows can group multiple devices into functional elements (such as constructing a view of a 32-bit data bus that physically crosses four 8-bit parts), and can window into multiple boards simultaneously.

Another key function is the embedded logic analyzer/word generator function. This function allows the user to apply stimulus and trace scannable signals in the system in a batch mode and present the results in a waveform display. The designer can select or deselect individual signals to trace and choose a stimulus file for application. The stimulus file could be something manually generated by the designer or poten-

tially, a reused portion of the logic simulation vectors. The stimulus is applied to the actual logic and then displayed on the PC in a waveform view similar to a conventional logic analyzer. Another function allows comparison of the results against an existing data base, such as the logic simulators results or the results of a previous run. Differences between the two runs are highlighted in color on the waveform display.

Limitations

The demonstration system developed by TI incorporated a limited variety of 1149.1-compatible parts, but even this partial implementation yielded encouraging results for improving the debug process. Several limitations were identified for the process and need to be addressed.

First, learning curve has a major impact. Valuable experience was gained in implementing the architecture, but at the expense of time and analysis. Conventional instruments such as logic analyzers were used to augment the process until the scan architecture itself was validated. Because this was the first application of many SCS features, it was necessary to resolve design errors versus problems with the SCS itself. This was similar to the problem mentioned previously of looking for design errors when manufacturing errors exist. However, once the core software for the SCS is fully debugged, the process is greatly simplified.

Coming to grips with programming under the SCS was a new experience for the designers. ASSET's environment is very "C"-like, and the designers came up to speed quickly. Some of the designers had developed code previously for debugging designs, and, thus, felt more comfortable with the software development than others. Using the debugger was very instrument-like, and, once the designers understood how to use it, it was a powerful tool.

The process of verifying the scan path required more effort than anticipated, but this is attributed to the introduction of some new interface parts and the respective software drivers for them within the SCS. In some cases, the designers made errors in the configuration files that caused scannable ICs to be addressed improperly. This is corrected easily with improved documentation on the process. It was necessary, however, to use multimeters and logic analyzers until these problems were identified and overcome.

Another limitation was that of scan, in general. The process of applying and capturing vectors via scan was synchronous to the scan clock and involved the overhead of scanning for each event. Thus, the test steps were slower than conventional instruments and required fully synchronous execution.

Emulation

Another design verification technique applied to microprocessor architectures is emulation. Conventional emulators require the device to be socketed so it can be removed during emulation. A special cable interfaces a hardware emulator (a separate box) into the socket, and software is used to model the execution of the microprocessor. This allows processor boards to be tested, as well as allowing the software hosted on the board to be debugged.

Unfortunately, emulators now must run at speeds too excessive to support the external cabling techniques. Each upgrade to a microprocessor requires a new emulator model to be designed, delaying entry to the market. Customers are not pleased with the additional capital investment required for each emulation system.

Via 1149.1, the capability exists for the majority of emulation features to be implemented in silicon and accessed via the scan bus. An example is TI's family of Digital Signal Processors (DSPs). The TMS320C30 supports many emulation features tied to an earlier bus interface, and future generations of DSPs will use 1149.1 as the port. Ultimately, this will allow board-level emulation, providing tremendous visibility into complex parts and offering powerful capabilities for software debug and test.

Summary

Advanced digital technology is outstripping the capabilities of many conventional testers and instruments because of reliance on physical access (unless painful concessions are given in the packaging area). The formalization of the IEEE 1149.1 Boundary Scan standard will introduce an infrastruc-

ture that not only aids and abets the test engineer, but equally assists the design engineer in debug of hardware and software. New scan-based tools can provide the same functionality of existing debug instruments without the burden of probing. This infrastructure is constantly available as long as the access to the scan bus exists, allowing new environments such as temperature chambers to be addressed. Designers will be able to verify the structural correctness of the prototype designs easier, eliminating ambiguity between design and manufacturing errors. Scan-executed sequences provide a viable, robust, interactive method for functionally exercising the design logic and may reduce the effort and cost of the design verification process significantly.

References

- [1] *Joint Test Action Group Specification*, Version 2.0.
- [2] *Proposed IEEE P1149.1 Standard Test Access Port and Boundary Scan Architecture*, Draft D5, June 20, 1989.
- [3] Lee Whetsel. *A Proposed Standard Test Bus and boundary-scan architecture*, Proceedings of the International Conference on Computer Design, October 1989, pp. 330–333.
- [4] Andy Halliday, Greg Young, and Al Crouch. *Prototype Testing Simplified by Scannable Buffers and Latches*, Proceedings of the International Test Conference, August 1989, pp. 174–181.
- [5] Don McClean and Javier Romeu. *Design for Testability with JTAG Test Methods*, Electronic Design, June 1989.

