

Gaining market advantage with an advanced embedded test environment

By Joseph Skazinski

One way to gain market advantage is to build core intellectual knowledge into product design, engineering processes, testing, and customer support. This article describes the market advantages of using an advanced embedded test environment during all phases of product development, product deployment, and product sustaining. An advanced embedded test environment can reduce the testing for a design, analyze performance characteristics, reduce board spins, reduce debugging time and development cost, provide in-field diagnostics when shipped with the product, and reduce the per-unit cost of manufacturing through automated tests running on the processor boards.

Embedded test software can be defined as a software application executed by the processor of an embedded hardware platform that performs various hardware tests. Written and customized to the design parameters of the target processor board, the software application typically varies with each unique board design. The application is loaded into Flash memory, or ROM, of the target platform and typically provides user interaction through a serial port. A faster connection, such as an Ethernet port, can be used once basic board functionality is established.

The test application provides its own boot code and basic Flash memory, serial connection, and RAM tests. Once the basic tests pass, the application switches from a safe boot mode into an interactive user mode. In the interactive mode the user executes a variety of tests using a command line, application program interface or menu system. The test cases vary from low-level system diagnostics to high-level functional and performance tests (see Figure 1).

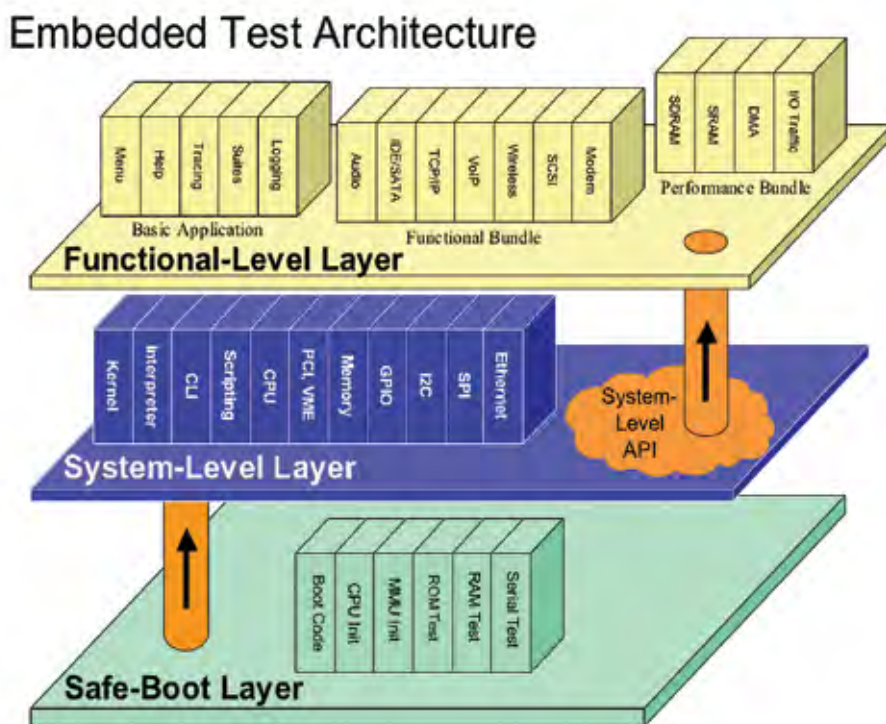


Figure 1

Advanced test environments provide their own kernel, drivers, user interface, canned test suites, execution tracing, test command library, a scripting interface for extensibility, and access to all buses, components, and data interfaces.

Embedded test is deterministic software with easily repeatable test cases that fully exercise all hardware interfaces, buses, and components. To maximize the usefulness of this test software, it should not rely on any third party boot code, loader, or operating system. It must be streamlined and have built-in trace capabilities for localizing and analyzing hardware faults. The test suites should be built using very simple test methods that are coupled into more complex test cases, which are built into an overall component test, and finally included in an easy-to-run system-level test.

Uses for an advanced embedded test environment

Embedded test has proven itself to be the most versatile of all board test environments and complements In-Circuit Testing (ICT) and the use of Automated Test Equipment (ATE) in general. Embedded test proves useful during product development, manufacturing, and product sustaining in the following ways:

Processor evaluation

Most processor companies deliver their latest CPUs using evaluation platform kits. Shipping these kits with an advanced embedded test environment allows designers to get live data with regard to performance and interaction with I/O devices. With an embedded test application designers can explore the system in greater detail without writing new software. The support effort associated with evaluation kits is reduced because a system-level test application easily verifies that the hardware is working properly.

Board bring-up

For prototype testing, often called *board bring-up*, embedded test provides a way to verify that a system is working, as well as to determine whether a design is performing to requirements.

Environmental testing

For environmental testing, an automated embedded test application can easily exercise the various aspects of the system to ensure that the most accurate results can be obtained for characteristics such as power consumption, temperature range, emissions, as well as other regulatory requirements.

Board validation

During application development, embedded test has proven to provide a quick method of assuring that the hardware continues to function as designed, thus aiding in the determination of root cause when system faults are encountered and helping to reduce the phenomenon of finger-pointing between hardware and software.

Manufacturing test

Embedded test provides a clean method of streamlining manufacturing test by reducing human touch-time per unit, providing full access for in-system programming, and providing an efficient test process handoff to an outsourced manufacturer.

Self-test

Embedded test can be reduced in size and functionality to produce a self-test application often referred to as Built-In Self-Test (BIST) or Power-On Self-Test (POST). This simplifies any support effort by providing a known status of the processor board as a starting point. Power-on test results can be used by the application to take appropriate action.

In-field diagnostics

An advanced embedded test application can be shipped with the final product for execution in the field when a system requires the validation that hardware is operating properly. Access to the diagnostic application is through direct connection to the unit or through a remote interface such as Ethernet. Shipping the embedded test application eliminates the need for support personnel to find the correct test setup or software; it's already in the product.

Board bring-up

After a new board has been designed, it is normal to create a few prototypes. The goal of this activity is to verify that the design will work. If successful, prototype bring-up

activity is replaced by environmental testing, delivering boards for application development, sending boards to customers, and verifying that the design will perform to expectation.

Most test methods, such as probes, emulators, and simple embedded debug monitors, will satisfy the first step of verifying that the prototypes work, although minimally. An advanced embedded test environment provides the best solution, and in most cases the only solution, for verifying that the system will meet performance requirements. It also can be used by customers and application developers for hardware testing, and can be used during environmental testing. The advantage of embedded test is that the system is exercised and tested at true processor speeds, and embedded tests run directly on the system, allowing the testing to be shipped to other parties with the system and used in environmental chambers.

Manufacturing test

Using embedded test software for manufacturing test is an ideal solution, but special consideration is required. Manufacturing test differs significantly from board bring-up testing. The main goal for manufacturing test is to provide a simple pass or fail result using an automated process that logs all pertinent data for each board tested. The ideal process is to assemble a board, power it on, and have the manufacturing test application verify the hardware and then serialize it for shipment. The serialization process reconfigures memory with a boot loader, operating system, application image, serial number, and other unique data required by the final application.

Embedded test software runs on the actual hardware platform and has full control of all aspects of the system, such as ability to configure programmable memories with data particular to your product and unique data particular to a unique unit. Embedded test software can be preprogrammed onto Flash memory so that it is part of the final assembled product producing the simplest test process for contract manufacturers.

As technology advances, the complexity of processor boards increases and the size of processor boards decreases. These changes have led to increased complexity in denser packages with smaller and less accessible access points driving a need for alternate test methods to complement or replace traditional ICT, ATE, and JTAG combinations.

The case against embedded test software

The key arguments against embedded test include the amount of effort and length of time required to write the software. Also, the embedded test software must be written by individuals who have fluent knowledge of both hardware and software. These individuals are potentially a rare and expensive resource. To be truly effective, any functional test software must be developed to the point that allows performance testing, extensive functionality testing, and stress testing. Companies that satisfy these requirements generally hold a distinct market advantage; curiously, embedded test requirements are typically not a stated core competency for most product companies.

Developing the test suite required to test prototypes, exercise the system during environmental testing, provide performance characterization, provide comprehensive testing for manufacturing and deliver in-field diagnostics – all running on the custom hardware – requires a minimum of 18 person-months of development effort for a typical hardware platform. These tasks may pull senior firmware engineers away from other software development tasks required for product release. Writing new software requires that it be tested along with new hardware, further complicating the debugging process for both the hardware and the software.

Summary

Embedded test software provides the greatest test coverage, greatest flexibility, true-speed system diagnostics and performance characterization, stress testing, and reduced touch-time during manufacturing. Embedded test provides the opportunity to ship built-in self-test and in-field diagnostics not available when using processor-based emulation testing. And best of all, embedded test software can be used during all phases of product development and manufacturing as well as product support.

ECD

Kozio, Inc. has taken the pain out of creating an advanced embedded test environment. Using their patent-pending process, Kozio has case studies available demonstrating the value of their embedded test environment for both board bring-up and manufacturing test; demonstrating that their solutions save significant debugging effort and cost. With more than 250,000 lines of code that have already been validated on dozens of custom platforms, designers get peace of mind that Kozio's solutions will provide test coverage second to none, while freeing up developers for the product's application.

Joseph Skazinski is president and co-founder of Kozio, Inc. Joseph, co-founder Keith Short, and their team launched Kozio in 2002 to deliver the essential embedded software for all embedded hardware platforms using innovative processes. Kozio delivers turnkey advanced embedded system-level test applications for ARM, MIPS, PowerPC, TI, and XScale processor architectures. Joseph holds a BS in Computer Science from Michigan Technology University, and is a member of ACM and IEEE.



To learn more, contact Joseph at:

Kozio, Inc.

1716 A North Main Street, #127

Longmont, CO 80501-7413

Tel: 303-776-1356 • Fax: 303-776-1357

E-mail: joseph.skazinski@kozio.com

Website: www.kozio.com