

On-Chip Emulation for Functional Test and Diagnosis

by Charles Swingler, International Test Technologies

As devices become faster and more complex, more board developers are turning to background debugging and JTAG emulators.

Anyone involved in the design, manufacture, or repair of processor-driven boards requires reliable, cost-effective test and debug strategies. But as surface-mount (SM) components shrink, board densities increase, and consumer product prices plummet, the more commonly used functional test strategies have become less viable.

Plugging an assembled board into a known-good target system, booting to an operating system, and then running tests require little setup effort and are not hampered by limited test access. However, labor costs of this hot mock-up testing are high, and if a board fails, diagnostics may be limited or nonexistent.

Functional Test by Emulation

Boards with a bus architecture can be tested by emulating an onboard device capable of taking control of the board via one of its buses. Types of emulation include processor, ROM, and bus. Once in control, the emulator loads and runs diagnostic tests.

Processor emulators replace the board's processor, giving full control over all the processor buses. Read/write access to all parts of the board's memory and I/O is available via the emulator.

Test access on densely packed boards is no problem for a processor emulator, although the real processor must be socketed to allow its replacement. Emulators were used extensively in board development, but increasing processor speeds now have made them impractical.

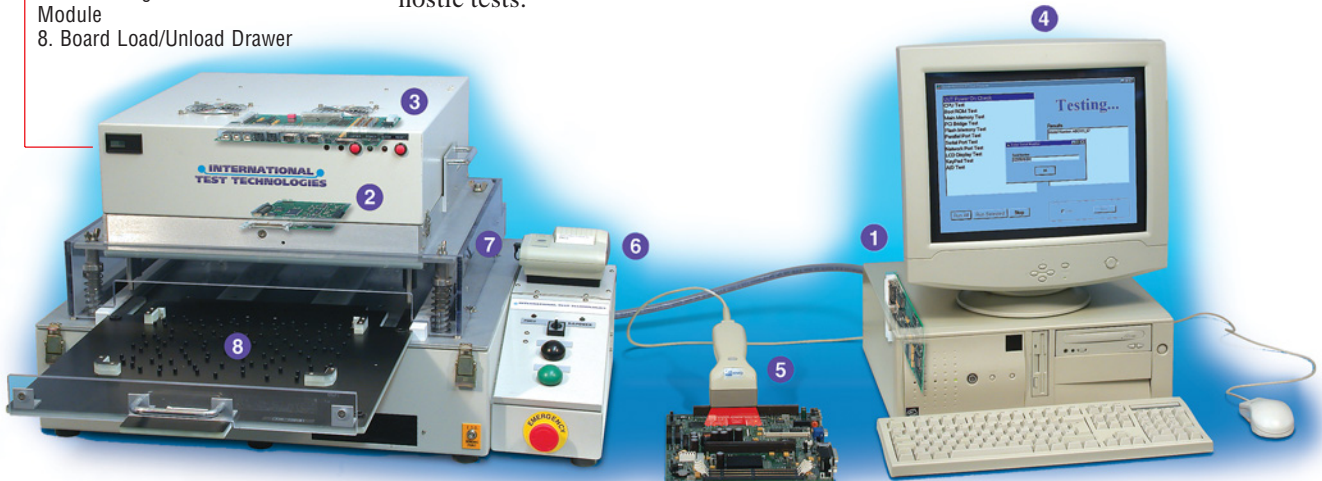
ROM emulators replace the boot ROM, substituting diagnostic code for the processor's normal boot code. They have a bidirectional communications pathway so that the unit under test (UUT) can communicate with the tester.

The primary shortcoming of ROM emulators is the inability to diagnose the processor to boot ROM area. ROM emulators can be difficult to use with soldered-in ROMs unless some circuit modifications are made at the product design stage.

Bus emulators connect to a bus slot or edge connect, giving test access to the

An Automated Manufacturing Test Solution Using Debug Port Access

1. Test Controller Card (PCI)
2. Processor Control POD
3. I/O Emulation Unit
4. Control PC
5. Board Bar-Code Scanner
6. Pass/Fail Results Printer
7. Interchangeable Bed-of-Nails Module
8. Board Load/Unload Drawer



various circuits and functions of the UUT via read/write bus cycles. They are useful for testing plug-in bus cards such as VME and PCI.

Functional Test via JTAG

The JTAG interface originally was designed to overcome test-access problems on miniaturized components when checking interconnections. This serial interface, also known as the test access port (TAP), uses five lines (data and timing), to access the daisy-chained shift registers built into each component's I/O pins, allowing a chain of JTAG-compliant components to be scanned for errors (boundary scanning).

By extending the JTAG protocol, manufacturers of microprocessors and DSPs have provided onboard debug facilities for hardware and software developers, overcoming the 20-MHz to 30-MHz speed barrier of earlier emulators. With the emulator now onboard, debug instructions can be run at full processor speed. Vendor-specific extensions to JTAG have two or three additional signal lines and an enhanced instruction set for controlling the processor core.

The list of debug interface-enabled CPUs includes the Intel® Pentium® processor family, the Intel XScale™ Microarchitecture Processors, and the Motorola™, IBM® PowerPC™, AMD®, MIPS®, and ARM® processor families. IEEE also has published IEEE ISTO 5001 for debug interfaces.

Manufacturers often identify their flavors of debug interfaces by specific names, including background debug mode or BDM (Motorola), hardware debug tool or HDT (AMD), and common on-chip processor or COP (Motorola/IBM). The concept of the BDM interface is similar to JTAG, but the signal lines and protocol differ.

Using the Debug Interface

Although these interfaces originally were aimed at design engineers, test engineers can exploit them to implement functional test solutions, resulting in reduced development times, improved diagnostic resolution, and shorter test times. Test and diagnostic equipment designed to use a processor's debug interface requires only about six to 10 test points on the UUT. This access can be achieved in most board designs, either by placing an

interposer between the CPU and the socket, using a very simple bed of nails when CPUs are soldered, or making use of the JTAG break-out header provided by some board manufacturers.

Any bus-architecture UUT can be divided into functional blocks such as bridges, RAM, video controllers and I/O controllers. Each functional block contains arrays of memory or I/O registers. Test programs use the extended JTAG debug functions provided by processor manufacturers to sequentially access these registers, building up a complete test.

Low-level functions include stop/start the processor, read/write memory, read/write general-purpose registers, read/write I/O, breakpoints, single-step code, and code trace. Combinations of these functions accommodate downloading test code

to a UUT, controlling and monitoring test code execution, and collecting test results from UUT memory.

For example, the read/write functions can test RAM, which also verifies intermediate buses. I/O controllers are tested either by looping the output back into the input, as in the case of network interface controllers, or generating/measuring signals with external devices attached to the board's connectors. Some test systems include I/O emulation units, which avoid the need to attach real peripheral devices.

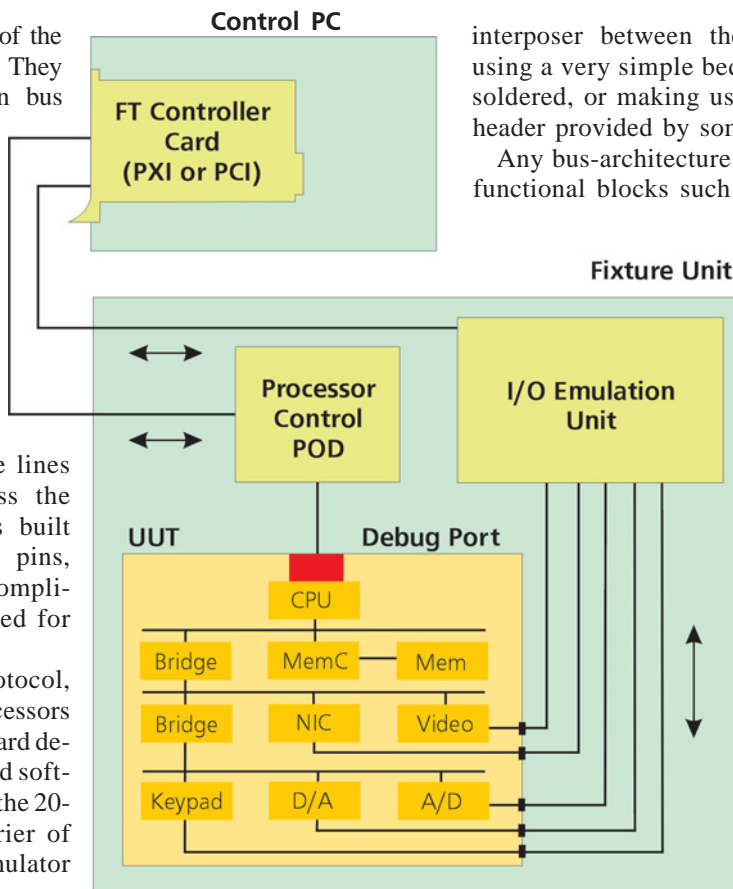


Figure 1. Architecture of a Debug Port Functional Tester

The functional test controller card controls the CPU via the processor control POD, which adapts signal levels and protocols to match the debug interface requirements of specific processor families. I/O ports are tested by the I/O emulation unit, which can both generate and measure digital and analog signals.

A Typical Test Sequence

Initially, the tester uses the debug interface to control the processor before it runs any code. Commands from the functional test controller card (FTCC) reach the debug port via a processor control POD that adapts signal levels and protocols to match specific processor families (Figure 1). Once in control, the FTCC uses the processor as a vehicle for sequentially testing all buses and addressable components throughout the board. I/O ports are tested by the I/O emulation unit, which also is linked to the FTCC to allow signal response to be compared with the stimulus.

For example, a loudspeaker output would be tested by commanding the CPU to program the DAC to generate

Board bone piles can be turned into financial assets.

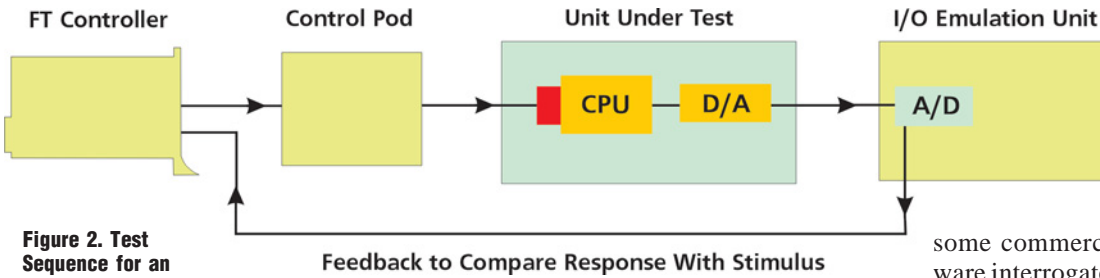


Figure 2. Test Sequence for an Analog Output Port

an audio output. This would be converted back to a digital signal by the I/O emulation unit and fed back to the functional test controller card to compare with the original stimulus (Figure 2). Video signals are tested in a similar fashion.

BDM/JTAG Benefits

Debug interface solutions have a number of significant benefits over other functional test solutions. Bypassing the board’s normal operating software allows functional test times to be slashed from minutes to

Setup time is minimized by the preprogrammed device libraries and automatic test generation (ATG) utilities supplied with

some commercial testers. ATG software interrogates a known-good board to identify the devices present and then automatically creates test scripts to validate the same set of devices on subsequent boards during manufacturing or repair. Common form factors such as PCI and PXI make it easy to integrate these testers with existing test solutions, including automated board-handling systems. Compatibility with industry-standard test executives means that connection into the wider enterprise is straightforward.

As consumer devices continue to shrink, board developers, manufacturers, and integrators are very likely to rely more on BDM/JTAG emulators to solve their financial and technological constraints.

About the Author

Charles Swingler joined International Test Technologies as marketing manager in 2001. Previously, he operated his own design and technical copywriting company for 12 years in Belgium. Mr. Swingler was trained as a product designer at Manchester University, U.K., and taught design and technology from 1974 to 1988.

International Test Technologies, 2515 Taraval, Unit 4, San Francisco CA 94116, 415-753-5376 or 415-710-0349

Device	Action
Microprocessor	Tester stops microprocessor and takes control
Memory Controller	Register test; configure registers
Memory	Memory test
Bridge	Register test; configure registers
Network Interface Card (NIC)	Register test; configure the NIC for normal operation; transmit and receive packets via a simple loopback
Video	Register test; use the debug interface to configure the video controller to generate a test pattern output on the video connector; use the I/O emulation unit video checker to verify the output
Keypad	Use the digital I/O module in the I/O emulation unit to simulate key presses; use the debug interface to verify key codes in the keyboard controller
A/D	Use the analog I/O module in the I/O emulation unit to generate a voltage; read back and verify the A/D conversion using the CPU debug interface
D/A (audio out)	Use the CPU debug interface to configure the audio controller to output a specific frequency; convert this back to a digital signal using the analog I/O module in the I/O emulation unit; verify with the FTCC

Table 1. Typical Test Sequences for Various Devices

An analog input such as a microphone port is tested by commanding the I/O emulation unit to generate an analog audio frequency. The resulting digital signal obtained via the CPU debug interface is compared to the original stimulus. Table 1 provides the actions a full test sequence might perform on the given devices.

seconds, leading to substantial test cost savings. Comprehensive diagnostic information can be provided down to the component level, even for dead boards that can’t be diagnosed with other functional test methods. In some cases, debug interface test solutions also offer in-system programming (ISP) of flash memory.

FOR MORE INFORMATION

www.intertesttech.com/

**N. America – see numbers above
Rest of World +353 749 188 100**

EE