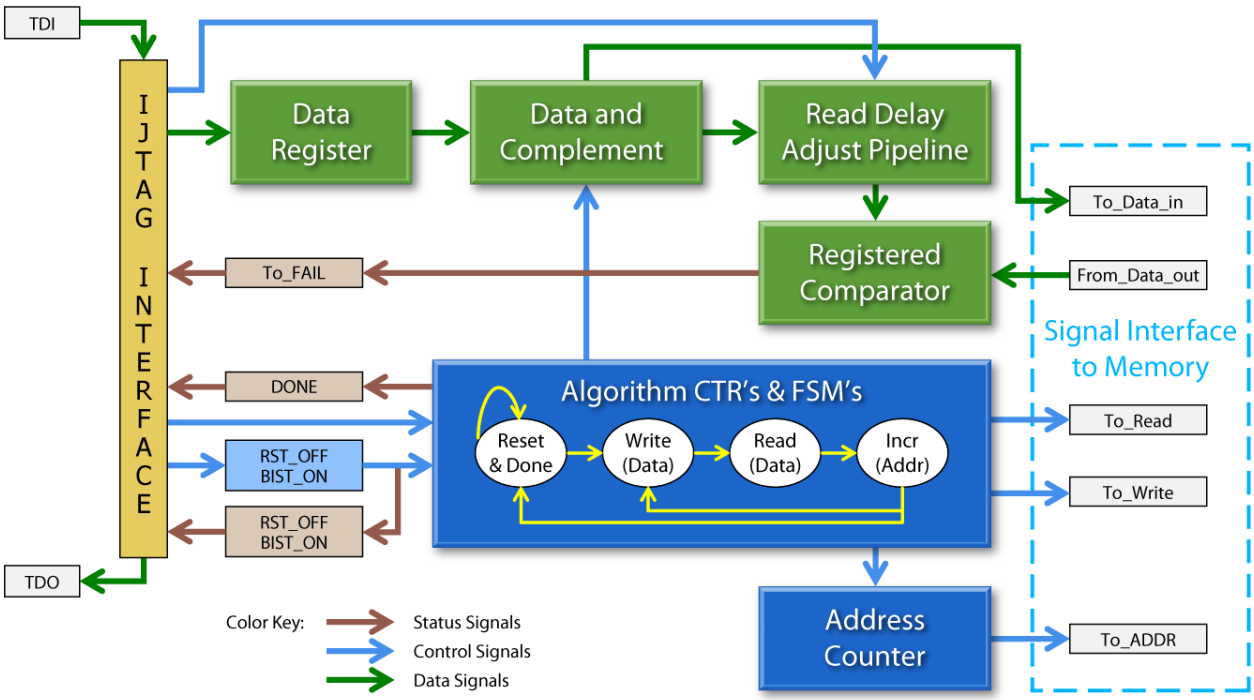
How to test High-Speed Memory with non-intrusive embedded instruments



A Memory Test Whitepaper

A Memory Test Whitepaper

By Al Crouch

Al Crouch, Chief Technologist – Core Instruments – at ASSET InterTech, is a Senior Member of the IEEE. Al has served as the vice chairman of the IEEE P1687 IJTAG working group that is developing the IJTAG standard and has contributed significantly to the hardware architecture definition. Al is the editor of the P1838 Working Group on 3D test and debug and co-chair of the iNEMI BIST group investigation defining the use of Chip-Embedded Instruments to assist with Board Test.

Acknowledgements

ASSET wishes to acknowledge and thank Phil Geiger of Dell, Austin, TX for his assistance with the paper.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

A recent survey of engineers in the electronics industry by the International Electronics Manufacturing Initiative (iNEMI) found that testing memory and memory buses on circuit boards is one of the most pressing problems for system manufacturers.[1] [2] [3] A number of factors has contributed to this problem; namely, the disappearance of test pads on circuit boards for in-circuit test (ICT) bed-of-nail fixtures, restrictions on placing any sort of a test probe on high-speed memory buses because of the capacitive signal distortion created by the probe, increasingly complex memory bus protocols and others.

Fortunately, there are a number of non-intrusive board test (NBT) or probe-less methods for testing memory, including boundary-scan test (BST), functional test, processor-based testing methodologies such as processor-controlled test (PCT), FPGA-based testing mechanisms such as FPGA-controlled test (FCT) and embedded memory built-in self-test (memory BIST). Each method has advantages and disadvantages over the others. As is typical in the electronics industry, choosing any one method will involve tradeoffs. This white paper describes these various methods and explains some of the most salient tradeoffs. In particular, some of the complexities involved with testing the high-speed DDR (Double Data Rate) memory bus will be explained.

[1] iNEMI www.inemi.org (BIST Study at http://www.inemi.org/project-page/built-self-test-bist-program)
[2] iNEMI www.inemi.org (Structural Test of External Memory – Boundary Scan Adoption, Phase 2)
[3] iNEMI members can access the results from both of these studies here: http://www.inemi.org/news/project-summaries-inemi-members

What's the problem?

A recent survey by the International Electronics Manufacturing Initiative (iNEMI) asked test engineers in the electronics industry what their biggest problems were with testing circuit boards. Of the 11 possible problems listed, characterizing and testing memory soldered to circuit boards was among the top three. Memory test was at the top of the list of prevalent problems along with 'loss of access to test points' and 'the need to perform debug/diagnostics on board failures.' When asked which type of built-in self-test (BIST) instruments would solve the engineer's problem, memory BIST was rated the second most needed, virtually tied with BIST instruments for validating high-speed I/O buses. Clearly, the ability to thoroughly test, characterize and diagnose problems with soldered-down memory is one of the most pressing problems in the industry.

It is important to note that these questions were posed to circuit board test engineers. In most cases, board test engineers assume that the memory devices themselves are not causing a failure since the chips are tested and qualified before they are assembled on a board. As a result, a memory test failure should indicate a failure in the connectivity channel to the memory. Previously, when memory speeds were not as high as they are today and communications protocols over memory buses not so complex, performing static shorts-and-opens testing on memory interconnects might suffice. Now though, signal propagation through passive devices such as capacitors and the signal integrity on high-speed traces to memory must be validated and characterized for an open data window. Often, this data window will demonstrate sensitivities to clock jitter, temperature, electrical noise as well as the level and stability of the voltage.

One of the several factors that have exacerbated the difficulties test engineers have with memory is the complexities of these buses. Many of the most prominent memory buses, such as the various generations of Double Data Rate (DDR) memory, have achieved very high data transfer rates at the expense of simplicity. Indeed, each generation's higher data rates have only exacerbated the difficulties of testing memory buses. A closer look at the complexities of the basic architecture of the DDR memory bus illustrates this point.

Data transfers to DDR memory are able to achieve their high speeds (DDR1 was rated up to 400 Mega-Transfers per second (MT/s), DDR2 up to 1066 MT/s and DDR3 up to 2133 MT/s)

because both the rising and falling edges of a signal (Figure 1) communicate a bit of data and the IO clock can be four times faster than the internal memory clock. Slower, previous generation bus architectures could transfer only one bit on either the rising edge or the falling edge of a signal. DDR is able to communicate a bit on both the rising and falling edges because it is a source-synchronous memory bus whereby the clock that is required to synchronize the DDR signals at the receiver is communicated over the DDR bus along with data and address signals. The clock, which is sometimes referred to as a strobe, is usually generated by the same device that generates the data and address signals, such as a memory controller, because this device's process-voltage-temperature (PVT) variation is likely to be more consistent than it would be if the clock were generated by another device. When the clock is properly synchronized with the data and address signals, both the rising and falling edges of the clock signal will be correctly synchronized at the receiver and two bits of data will be transmitted during one cycle of the DDR clock without synchronizing any phase lock loops (PLL) or using any clock/data recovery schemes. At this level of complexity, structural shorts and opens are not the only causes of failed memory transfers. The integrity of the clock, data and address signals may be susceptible to temperature, jitter, noise, voltage aberrations and other environmental conditions. In addition, these high-speed memory signals are very sensitive to board manufacturing variations in resistance, capacitance, inductance and others. Any of these factors can adversely affect the timing of the data window that is needed for a valid DDR memory transfer. Often the DDR bus must be tuned by adjusting the parameters of the data window to optimize data throughput. Failing to achieve a valid data window on a particular circuit board's DDR bus may indicate that the manufacturing process which produced the board exceeds design tolerances or is defect prone.

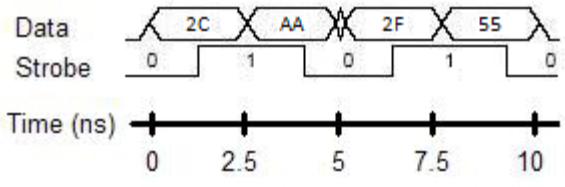


Figure 1: DDR signals carry data on rising and falling edges

Where's the problem?

In the broadest sense, memory testing takes place over the entire life cycle of a system, beginning with board design/development, moving into production and culminating in post-production stages such as field service. Eventually, the cycle begins again when memory test is performed during design/development for the next generation of the board design. During each phase in the life cycle, the objectives and goals of memory test differ and the memory test process itself is typically referred to differently, according to the objectives of that particular phase.

During design and new product introduction (NPI) (when first prototypes of a circuit board are produced and evaluated; this is often referred to as the board bring-up phase), testing memory in a timely fashion is particularly critical if the new system is to be delivered promptly to the marketplace. Only by doing so will the full revenue-generating potential of a design be achieved. During the initial phase of board bring-up when first prototypes are received, memory tests will be performed to identify the root cause(s) of failures or faults in the design of the memory architecture so that these can be quickly corrected prior to the design's release to production. At the same time, the performance of the memory architecture will be characterized to determine whether the design meets or exceeds its performance specifications. Often, characterizing the memory architecture will include an analysis of signal integrity issues on high-speed memory buses. The main goal during NPI is to verify the functionality and specifications of the design of the architecture. This implies that memory tests during this phase typically will focus on the functional and structural aspects of prototypes so that design flaws can be identified.

Once the board bring-up phase has been completed, the design is ready to transition into manufacturing. At this point, the nature of the memory tests which will be performed during volume assembly and manufacturing will differ considerably from the memory test suite that was performed during board bring-up. Unlike board bring-up where identifying and correcting design bugs in a design were the primary objective of memory test, the time it takes to conduct memory tests will be of paramount importance during manufacturing. Manufacturing test engineers assume that faults in the design's memory architecture have already been corrected by the time the board reaches production. Consequently, the memory tests that will be performed during production are intended to determine whether individual circuit boards are ready to be released to

users. This would involve determining whether the manufacturing and assembly processes have added any defects to the circuit boards. These types of memory tests will be essentially go/no-go tests which can be applied quickly in order to keep pace with a predetermined beat-rate on the production line.

Once production has hit its stride, memory test will be performed again on boards that fail the manufacturing test suite and do not qualify for release to the market. The intent here is to determine whether the failures were a result of environmental conditions, random defects or some systematic problem in manufacturing that is affecting yields. Whereas boards that have failed due to environmental conditions can be reclaimed without rework, boards that failed because of random or systematic defects will typically only be reworked if the price of the board dictates that recovery is an economically viable option and that some or all of the design and manufacturing costs can be recouped. Boards that fail due to systematic defects must be debugged and diagnosed to the extent that the root-cause or source issue in the manufacturing process is understood so that changes can be made to the process to prevent further low manufacturing yields. In most cases, the test that detected a yield issue is retained as part of the manufacturing test to identify when this condition might occur again.

During post-production phases of the life cycle, or when systems have been sold and are installed in the field, memory tests are performed to troubleshoot malfunctioning systems and maintain user satisfaction. There are two main goals during this phase: 1.) to identify any reliability concerns such as memory chips or board structures that fail earlier than expected; and 2.) to identify changes that may make the board design or the component selection better suited to deployment to the market. The final post-production phase begins when the design/development department takes a board design as the basis for the next generation of the system. At this point, the cycle begins again with characterization tests to determine where performance in the next-generation product might be improved to achieve a competitive advantage in the market. The key aspects of post-production testing are to collect performance and reliability data in a real world environment and to resolve the collected data to any sources of faults or failures that generated it. For example, an anomalous action by a board in the field may require gathering system state information from RAM and comparing this to what was expected by the runtime software.

What is a memory test?

All memory tests are based on reading from and writing to memory. Test reads and writes conducted in a normal operating manner from the board's normal or functional memory access unit can be referred to as functional memory tests. Functional memory tests can prove out the design or functionality of a given memory architecture, but they may not be able to detect or diagnose manufacturing or assembly defects. Manufacturing memory tests are generally structural and algorithmic pseudo-functional tests. In fact, these same types of tests are used by many memory device suppliers to test their memory chips and/or logic chips that contain embedded memory. The most prevalent form of an algorithmic pseudo-functional memory test is memory built-in self-test (MBIST). MBIST implies that the test is implemented on-silicon or provided by an algorithmic test instrument embedded within a memory test system.

Mostly, memory tests that prove out memory chips are sequences of reads and writes to memory locations coupled with specific data patterns and applied addressing schemes. By employing different data sets during a memory test, the memory's data handling capabilities are tested. By employing different data sequences, different stresses on the word selection process are verified. For example, at-speed back-to-back reads between two addresses can determine whether the address decode is capable of resolving at the rated operating speed. Back-to-back reads of many different addresses (words) will check the power consumption or power handling characteristics of the memory architecture. Because the sequence may not be a "real functional sequence" and the source of the sequence may be an on-chip logic unit such as an embedded instrument or a software test function, the testing is typically referred to as pseudo-functional or it is simply called structural.

Shown in Figure 2 below is a typical MBIST function that illustrates pseudo-functional testing. To initiate the MBIST, there must be a reset or start command. The MBIST must then be configured by providing a source of the applied patterns (such as a group of "5s" = 0101) and their complements (for example, a group of "As" = 1010); or a unique data sequence (such as using the address for data, which is called address uniqueness). Since the latency of a read or write cycle may not be known, the MBIST usually includes a read-delay function (a registered data path) to ensure that the data written into a memory location and subsequently read from that location arrives at the comparator at the same time as the data it is to be compared against (the known expected data). Essential to every memory system is the comparator, which is usually a bank of exclusive-NOR registers.

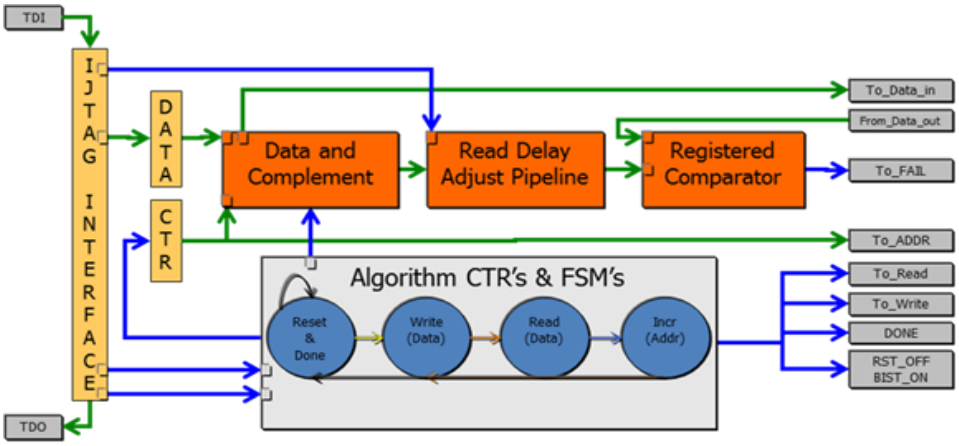


Figure 2: A typical MBIST instrument
(also referred to as an embedded MBIST instrument).

Often during a memory test, the data source provides a base value, such as 32 or 64 bits of the Hex value 5. This is written into one or more memory locations; then, a read-request for those locations is processed. As a result, the written data and the read data both reach the comparator at the same time. If the compared data is identical, the MBIST continues. If the compared data is not identical by even one bit, then a fail flag is asserted, which may or may not stop the MBIST, depending on the test configuration settings. The sequence of applied reads and writes, and address locations depends on the algorithmic unit in the MBIST. This algorithmic unit generally contains a set of counters or a finite state machine (FSM) that establishes the order of operation. A common algorithm is a march type of algorithm which consists of the following sequence of operations: after initializing the memory with a uniform data value such as all 0s, the algorithm

will read a memory location, followed by writing the complement such as all 1s; next, the process will read the complement, followed by incrementing the address (the algorithm begins at address 0 and ends at address N=last). Different algorithms are used to highlight different failure modes. IC test algorithms come in a wide variety to identify IC design and manufacturing defects and problems. Board test algorithms are generally much simpler because they focus on detecting and possibly isolating interconnect defects. Most MBIST units are very flexible so that changing algorithms can be as simple as pointing to a different FSM or remapping the states involved with the reads, writes and address selection.

It must be noted that board test requirements for memory testing are less rigorous than chip-test requirements. As a result, board memory test algorithms are generally simpler and require fewer operations. For example, a simple board-level interconnect test may be accomplished by writing and reading 5s and then As on the data bus, simultaneous to applying these same values to select locations 5 and A on the address lines. This ensures that all routes are stressed with a checkerboard pattern consisting of 1s surrounded by 0s and 0s surrounded by 1s.

How to solve the problem

Many test engineers are turning to NBT (non-intrusive board test) technologies to test today's high-speed memory architectures on circuit boards. The following basic diagram (Figure 3) with

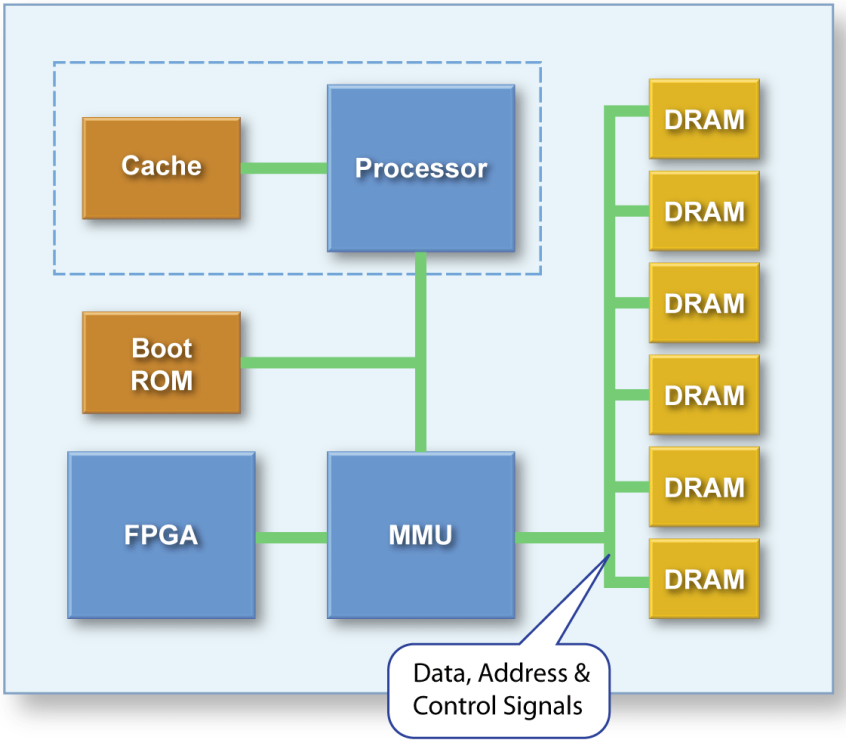


Figure 3: A typical circuit board architecture with memory

minor variations will be used throughout the rest of this paper to illustrate how non-intrusive test methods can be deployed in memory test.

Boundary-scan test

Although memory chips typically do not conform to the boundary-scan standard (IEEE 1149.1 JTAG) – meaning they do not have a 1149.1 Test Access Port (TAP) or dedicated boundary-scan registers on-chip – they can be tested from the boundary-scan facilities of a connected device, such as a memory management unit (MMU) or a microprocessor if there is power, clock and access to the 1149.1 TAP at the board level. The boundary-scan registers on a MMU (Figure 4) can be appropriated and directed to test the shorts and opens on the interconnect routes to the memories. In some cases, boundary-scan test (BST) may be the only alternative to test these interconnects, especially when the bed-of-nail fixtures that are essential to in-circuit test (ICT) systems do not have access to the memories because there are no or few test pads on the board.

When the boundary scan registers of a connected device are used to test memory interconnects, algorithmic writes and reads are executed on memory locations in particular sequences. The more complex the test algorithm, the longer the testing will take because a boundary-scan 'scan operation' (ScanDR) must be conducted for each read or write. In many cases, if multiple Read, Write, Stall, Output_Enable and other control signals must be operated at different times to make up the test, then multiple ScanDRs must be conducted to operate those signals in the proper sequence for each read or write action.

Figure 4 shows a memory management unit with a boundary-scan TAP can be used to test the memory devices to which it is connected for specific faults, such as those defined by the commonly used PCOL/SOQ board test fault spectrum. In this case, shorts or opens have disrupted the interconnect to the first DRAM device, the next DRAM does not have power, another is missing and the last memory device is the wrong type of memory and its orientation is incorrect. (PCOLA/SOQ is a fault spectrum originally defined by Agilent Technologies. For a description, see this post to the ASSET Test Data Out blog: http://blog.asset-intertech.com/test_data_out/2010/07/inemi_pcolasoqfam.html)

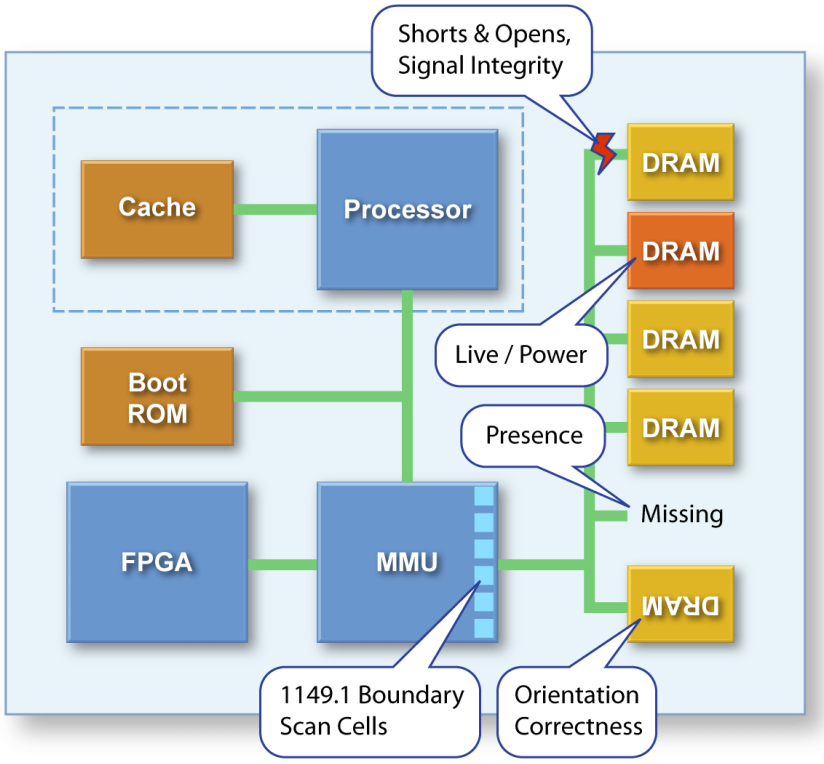


Figure 4: A memory management unit

In addition, the rate and efficiency at which boundary-scan tests can be applied to memory buses will be limited to the clock speed of boundary scan (for example, 50 MHz may be the maximum boundary-scan clock speed) and the length of the board's boundary scan register. These factors could cause lengthy test times. Moreover, multiple scans typically comprise a memory test. When each scan can only be applied at a slow speed, the time needed to apply an entire memory test made up of multiple scans can become extensive. In some cases where the memory architecture consists of very fast memory such as DDR3 or DDR4, the overall scan-in and update

speed of the boundary scan register is not fast enough to meet the minimum speed required by the memory chips to conduct a valid read or write. Because its test times will be longer, boundary scan is usually best suited to designs with slower memory buses or when boundary scan is the only alternative for memory test. When this is the case, boundary scan is a feasible alternative for testing memory interconnects as opposed to performing full memory cell testing.

Functional test

Some test teams decide to perform the most basic and well-understood non-intrusive test, functional test, on a circuit board to test memory buses. Unfortunately, functional tests can only be applied when the board's BIOS (Basic Input/Output System) or the firmware contained in its Board Support Package (BSP), as well as its operating system are all functional. In addition, this will require fully integrated firmware in all on-board FPGAs, flash memories and the proper configuration loaded into the MMU. At the very least, this base level of software will be needed to apply functional test routines on memory buses. The inability to test prototype hardware independently of its basic software environment increases the risks to the timing of a new product introduction since, most often, software is only integrated into a design late in the design cycle. Difficulties at this point would likely extend the design phase and make the new product late to market. If the test team must wait for the design's base level of functional software, the risk of a late entry increases. In addition, identifying design problems late in the development cycle can result in increased rework costs.

Functional memory tests must be developed by an expert programmer who is very knowledgeable of both the microprocessor and of the board's hardware issues, including how the memory buses and memory devices function, as well as sophisticated software programming techniques. In the end, the validity of a functional memory test is limited by the expertise of the person developing the test. Moreover, memory buses could pass a functional test, but the test itself might not exercise all of the required aspects of the memory bus for structural or manufacturing defects. If this were the case, the functional memory test would merely prove that a certain sequence of reads and writes can occur without faults given a known set of environmental condition.

Processor-Controlled Test

A type of pseudo-functional test, processor-controlled test (PCT), will either put code into the processor's cache memory or use the processor to drive the address/data bus directly to create functional memory tests independently of the board's basic BIOS/BSP firmware/software environment. If a processor is included on the board, it has access to memory and the board is assembled so that the processor and its attached memory have power and clocks, then a processor may be accessed in a non-functional way through its debug or emulation port to test memory. The processor may be directed to apply pseudo-functional test routines independently of the board's functional firmware or operating software (that is, prior to the integration of the processor's BIOS or BSP). Consequently, if the processor is connected directly to on-board memory (Figure 5), or to a MMU chip, PCT can apply its pseudo-functional test routines from the processor to test memory.

PCT can perform memory tests in two ways: first, by inserting test routines into the processor's cache and executing high-speed memory tests from the cache; or by asserting direct control over internal processor registers. In some cases, PCT memory tests may be limited by the microprocessor's architecture or by the functional limitations of the memory bus.

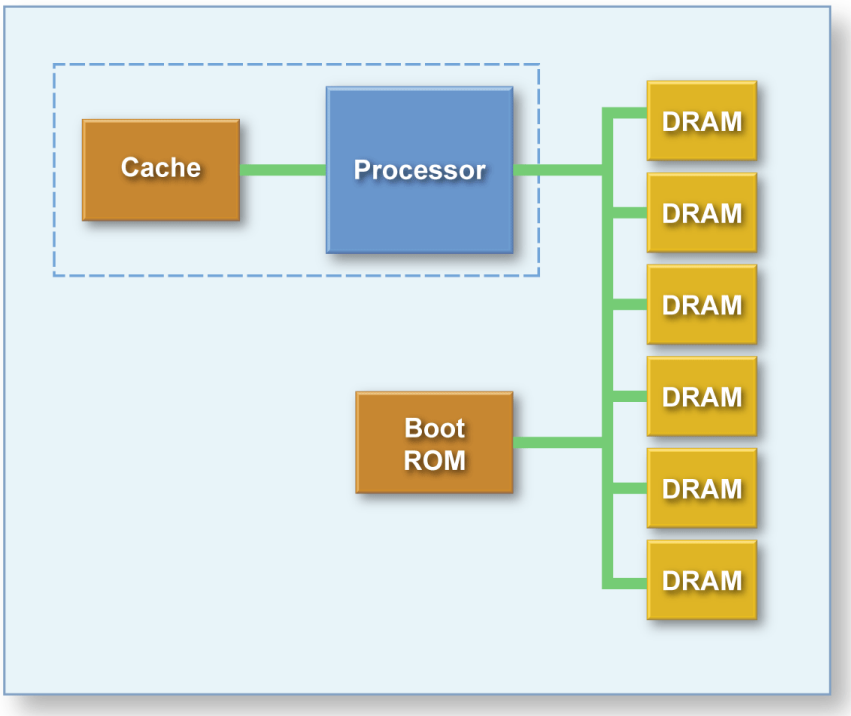


Figure 5: A processor connected directly to memory can take advantage of PCT.

With PCT, test teams can apply memory tests which verify the functionality of the memory during the early stages of board bring-up without having to wait for the design's operating firmware. In addition, PCT-type capabilities have been applied and proven in the market. The validity and comprehensive nature of PCT test routines have been verified over many different designs from many board manufacturers with many different supported processors. The comprehensiveness of PCT's memory test routines have themselves been fully field tested.

One small time impact with PCT functionality is the fact that whoever develops the PCT pseudo-functional tests must understand the specific microprocessor and develop algorithmic-like tests that take into account the unique aspects of the board design, such as various data sizes, address sizes, clock speeds and pin timing differences. Fortunately, certain third-party tools providers have automated much of PCT's test development so that the board test user does not need to become an expert in processor debug and emulation. This results in a library of supported microprocessors and test routines. Only if something new or different is needed, such as support for a new microprocessor or a new type of test functionality, is there a time lag for the tool provider to deliver the necessary support.

FPGA-based memory tests

Similar to the processor-based memory test mechanism described above, an FPGA (Field Programmable Gate Array) connected to a memory architecture can also be employed as the basis for memory tests (Figure 6), if a minimum of power and clocks are provided to the FPGA, and the device has access to the IEEE 1149.1 boundary-scan (JTAG) TAP on the circuit board. In many designs, the MMU is an FPGA or the MMU is connected to an FPGA. In both such cases, memory tests that are sourced from the FPGA are able to test the design's associated memory buses and chips. These types of memory tests could also be performed during the early stages of board bring-up because they do not depend upon the presence of functional firmware in the FPGA or any board firmware in general.

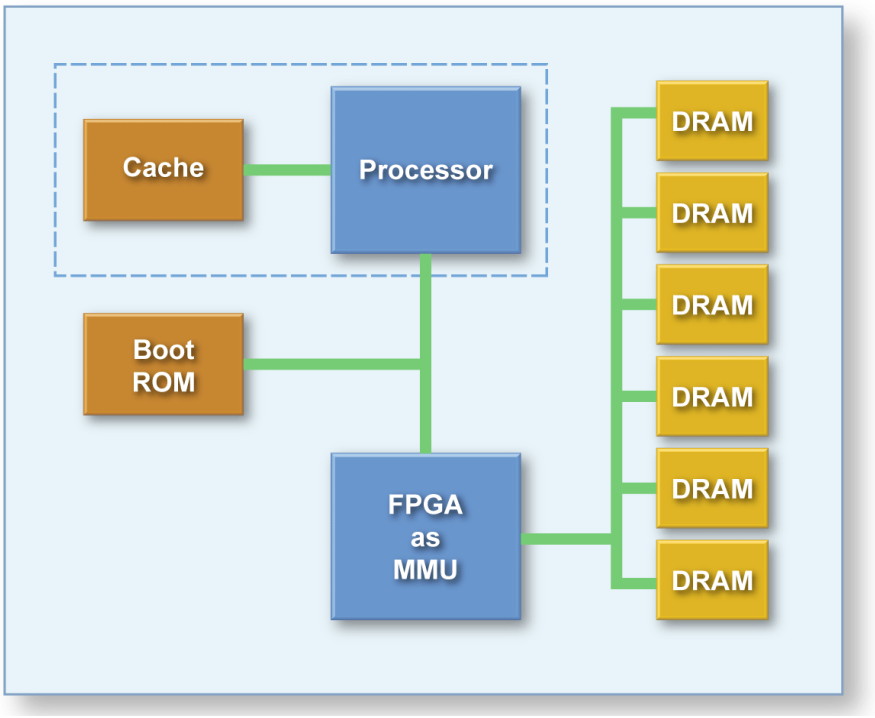


Figure 6: An FPGA connected to memory can take advantage of FCT.

FPGA-controlled test (FCT) involves embedding within an FPGA the test instruments, such as the MBIST instrument illustrated in Figure 2, that will comprise the temporary embedded tester needed to adequately test the design. For memory test, memory test instruments would be embedded in the FPGA, but it should be noted that other types of instruments may also be included in the FCT tester at the same time, such as fast flash programming instruments, digital pattern generators and capture buffers, instruments for SerDes bit error rate test (BERT), SPI/I2C master interface test instruments and others. The presence of a full suite of test instruments in an FPGA-embedded tester allows test teams to schedule and apply a much more comprehensive and realistic test environment. In addition, multiple instruments executing concurrently may raise the noise and temperature level of the board to the point where a more fully stressed voltage/power environment becomes part of the test suite.

It is important to point out that the MBIST instrument's memory test functionality does not change from one board design to another. This functionality is algorithmic and portable. What does change will be the memory architecture present in each board design such as the size of the address space, the types and capacities of each memory device, the timing and data rate, the

memory test algorithm that will be applied and other aspects. While the functionality of the MBIST or other embedded instruments remains the same, their configurations may be altered to fit the specifics of the memory architecture. Similar to PCT, commonly required FCT embedded tester functions are currently supplied by the tool provider. Embedded instrument intellectual property (IP) development is only required when new embedded instrument IP functions are needed. Once new instrument IP is developed, it may be re-used over and over again by simply changing its parameters.

The FCT tool used to test a circuit board design must support the particular FPGA deployed on the board and the FPGA must accommodate the embedded test instrument to be implemented. However, an FCT test instrument such as an MBIST instrument can be configured for the specific characteristics of a certain board design by setting a number of parameters on the instrument. In this sense, FCT instruments can be easily 'parameterized' to meet the specific requirements of every unique design. In addition, the FPGA in which the FCT test instruments will be embedded must conform to certain requirements in terms of size, performance, and integrated capabilities and functions. Typically, this is not a problem for board designs where the targeted FPGA will later function as the design's MMU, since the features and performance needed to function as an MMU ensures the functionality required for memory tests emanating from the FPGA. Attempting to employ an FPGA that is not directly connected to memory in the design may be problematic. Asserting memory tests through intervening chips is not easy, portable or directly algorithmic unless the operation of the MMU is easily incorporated within the algorithmic test. When complex through-chip operation is the only alternative, automated and off-the-shelf IP-based FPGA-based memory testing does not match the needs of the application well. A custom test process may be needed.

It must be noted that there are two methods for adjusting the size and timing parameters of embedded instrument IP to a target FPGA. If the capacity of the FPGA is limited compared to the size of multiple IP instruments to be inserted, then the raw IP file (i.e. Verilog) may be preprocessed with the appropriate parameters to generate a smaller implementation image. If the FPGA is large compared to the size of the multiple IPs to be inserted, then the IP may be left in a configurable form and IP parameters can be changed through the FPGA's JTAG port. Having the option of both methods is an important benefit of an embedded test generation tool.

It must also be noted that the embedded tester programmed into the FPGA and used for board test may only be temporary. After testing is complete, the embedded tester can be removed so that the FPGA's functional firmware can be inserted when it is available. If testing is needed at some other stage of the board's lifecycle, the embedded tester, or one more suited to the current problem or issue, may be reinserted, operated and removed.

ASIC-based memory tests

In some cases, an application specific integrated circuit (ASIC) connected to the memory architecture (Figure 7) can provide the basis for memory tests. In fact, MMUs are often ASICs, which makes them an ideal basis for memory testing in terms of performance, timing and data rate. In order for an ASIC to execute pseudo-functional memory tests, the ASIC must include a memory test instrument(s) for doing so. Unfortunately, the instruments embedded in ASICs are typically put there to perform tests on the ASIC itself during IC test and are either not capable of testing external memory or the instruments are not capable of applying algorithmic tests to meet board test needs. It is very costly in terms of the number of gates and silicon area to place a generic, fully-configurable memory test function within an ASIC. In addition, many ASIC vendors today do not make the instruments embedded in their devices available beyond IC test

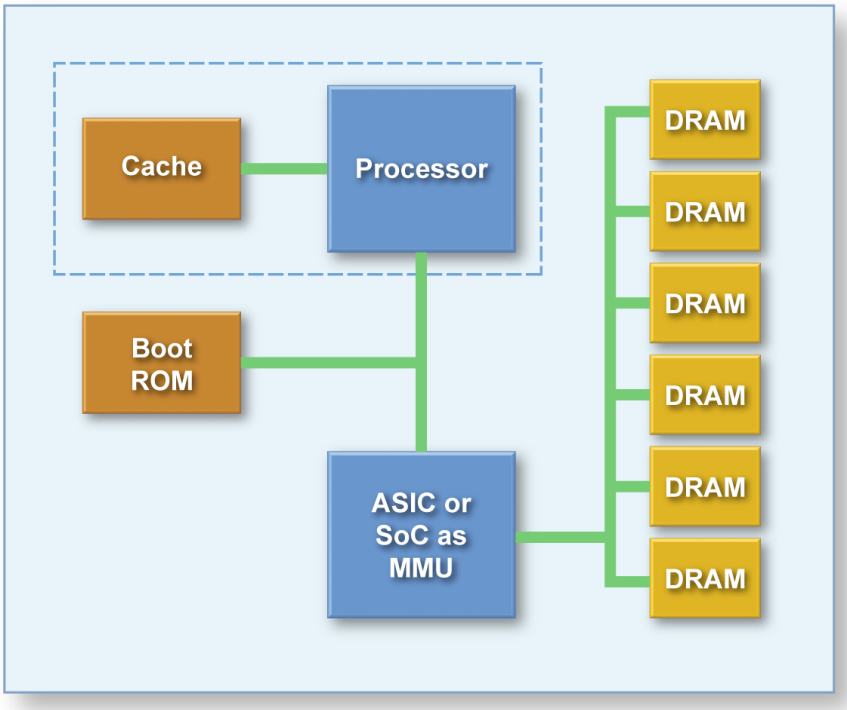


Figure 7: An ASIC connected to the memory could implement memory tests.

since these instruments often use custom interfaces that borrow functional pins (that is, functional pins on the ASIC are temporarily used for test before they are re-purposed to perform their functional role). Often these custom interface pins cannot be supported as package pins on the board or within the system because they are already mapped to other functions. Moreover, many ASIC suppliers view their test functionality as proprietary and they are not willing to share it.

Even if the embedded instruments in ASICs and systems-on-a-chip (SoC) were made available for board test purposes, each vendor would likely have its own unique instrument interface and communications protocols for controlling its instruments. A board design featuring instruments embedded within chips provided by several different vendors would be very difficult for the board test team to fully utilize since each chip's embedded instruments could require different communications, protocols and control mechanisms.

Quite frequently today, embedded test instruments in ASICs can be accessed through the chip's boundary scan/JTAG port, but even when this is so, the instruments cannot be controlled or managed via the JTAG port because the instruments would have to be included in the device's Boundary-Scan Description Language (BSDL) file and there is no way to do this. More likely, instruments will be wrapped in an IEEE 1500 structure that allows description with that standard's CTL (Core Test Language). This will allow previously defined test vectors to be delivered in IEEE 1450 (STIL) format. A more efficient solution to this situation will be the IEEE P1687 Internal JTAG (IJTAG) Standard, which will define instrument access and test procedures in its ICL (Instrument Connectivity Language) and PDL (Procedural Description Language). This standard is in development and nearing the ballot and ratification stage.

For more information on IEEE P1687 IJTAG and its ratification status, go to the Standards Corner on the ASSET web site at: http://www.asset-intertech.com/Technologies/Standards-Corner.

Microprocessor- and chipset-based MBIST

Another form of memory test, which is usually limited to validating the performance of memory buses, involves embedding MBIST engines in microprocessors and/or their complementary chipsets. A prime example of this is Intel®'s Interconnect Built-In Self Test (IBIST), that

company's proprietary embedded test technology which it has embedded in its advanced microprocessors and chipsets to facilitate validation, test and debug applications. For example, with the proper tools, the IBIST MBIST engines in an Intel chipset can verify the performance of the bus that connects DDR3 memory devices to Intel's processors based on the current-generation micro-architectures codenamed Sandy Bridge and Ivy Bridge.

The Intel IBIST MBIST embedded instrument drives data patterns onto the DDR3 bus connected to the processor. These patterns form the basis for tests that verify the read/write capabilities of the memory bus. In addition, certain operational parameters on the bus, such as its voltage and signal timing, can be modulated.

Availability of various NBT memory test solutions

Memory has been tested with standalone boundary-scan (IEEE 1149.1/ JTAG) testers for more than a decade. Currently though, certain new memories may be outpacing the ability of board-level boundary scan since the minimum clock and data rates of the memory devices may exceed the maximum board test clock frequency (TCK) of boundary scan. So, for modern high-speed designs, the application of boundary-scan test (BST) as a tool for memory test may be limited, but only time will tell.

Functional test has also been applied to memories for years, but it can only be brought to bear late in the development cycle when the board is fully functional or close to fully functional. As a result, functional tests are very difficult to apply to boards with a problem, bug, defect or error. Functional tests typically are applied with the board in a test jig or when the board is in a fully configured system. In addition, their diagnostic granularity may be very limited.

More recently, third-party pseudo-functional test tools have been able to overcome many of the difficulties of classic functional test and reduce the risks that a test development process late in the design cycle typically poses. For example, the processor-controlled test (PCT) tool in ASSET's ScanWorks platform for embedded instruments applies pseudo-functional test routines independently of the board's functional firmware or operating software. ScanWorks PCT test routines can be applied by a supported processor without requiring a functional firmware environment. The tests are applied through the same ScanWorks tester hardware that applies boundary-scan test. The ScanWorks controller hardware provides access for the ScanWorks PCT

tool so it can appropriate the processor's capabilities by way of its debug port and subsequently apply pseudo-functional tests. Consequently, if the processor is connected directly to on-board memory (Figure 5), PCT can apply its pseudo-functional test routines from the processor to test memory. With PCT then, test teams can apply pseudo-functional memory tests during the early stages of board bring-up without having to wait for the design's operating firmware. In addition, the validity and comprehensive nature of PCT's test routines have been verified over many different designs from many board manufacturers. The comprehensiveness of PCT's memory test routines have themselves been fully field tested.

FPGA-controlled test (FCT) tools, such as ASSET's ScanWorks FCT, are available today. Borrowing a functional FPGA on a board as the basis for test capability during prototype board bring-up and manufacturing has been implemented for years, but each such implementation has been developed by the board test provider and each test suite typically involves custom or functional development of test instruments and custom access protocols. The difficulty here is that the instruments embedded in the FPGAs are custom crafted for each board design with no, or limited, intended re-use. In some cases, developing the embedded instruments is as daunting a design task as developing the device's functional firmware. In addition, the individual instruments that make up a multi-instrument tester embedded in an FPGA must be controlled and operated by test engineers. This would necessitate the development of an operating system for the embedded tester. Lastly, inserting and removing a multi-instrument tester from an FPGA must be a simple, straightforward process since most board test engineers are typically not familiar with FPGA instrument insertion and other test-related operations.

The ASSET ScanWorks FCT tool offers an example of how this can be accomplished. ScanWorks FCT interfaces to off-the-shelf reusable embedded instruments targeted at board test needs that conform to the IEEE P1687 Internal JTAG (IJTAG) standard through the circuit board's IEEE 1149.1 boundary-scan (JTAG) infrastructure. The IJTAG access architecture is ideal because its PDL (Procedural Control Language) acts as an application programming interface (API) for the embedded tester instruments. Since PDL can represent fully defined and reusable operating routines, then the ScanWorks FCT tool acts as the operating system for the multi-instrument tester that has been embedded in the FPGA

Using an ASIC or SoC with embedded instruments that have been made public by the chip provider requires a tool environment that can read not only BSDL, but other documentation files such as CTL and then subsequently manage the operation of the embedded instruments. Since the instruments embedded in ASICs, SoCs and other chips often may be accessed by a combination of boundary scan (IEEE 1149.1), IEEE 1500 and IEEE P1687 IJTAG, a tool environment that can operate architectures based on all three standards would be effective. ASSET's ScanWorks IJTAG test tool has these capabilities.

And finally, microprocessor- or chipset-based MBIST instruments can be deployed to validate memory buses, but typically this method can only be employed when the processor supplier has embedded the MBIST instruments. In the case of Intel's IBIST embedded test technology, a toolkit is available on the ASSET ScanWorks platform which can apply IBIST's MBIST engines to validate the performance of certain high-speed memory buses, such as the DDR3 bus in designs based on the processors codenamed Nehalem, Sandy Bridge and Ivy Bridge. The ScanWorks IBIST toolkit accesses the embedded IBIST test technology to drive patterns onto the DDR3 bus. These patterns form the basis for tests that verify the read/write capabilities of the memory bus, perform margin testing and execute bit error rate tests (BERT).

Tradeoffs

The initial questions that arise during the design of non-intrusive memory test strategies are which test technologies should be deployed and what tradeoffs will have to be made? Answering these and other questions begins with an analysis of the circuit board to be tested relative to its memory architecture and the sorts of devices that are connected to the memory. These connected devices might be used as the basis for non-intrusive memory test.

If the board has an ASIC with boundary-scan/JTAG access that is functioning as a MMU and this device has no embedded test instruments, then the non-intrusive testing technologies that are available to test the memory architecture will be limited to either functional test or boundary-scan test. If the memory architecture is connected to an FPGA or a microprocessor, then FCT and PCT are options. If the memory architecture is driven by an ASIC that includes embedded memory test instruments and these instruments are accessible by the device's boundary-scan/JTAG port, then the embedded test instruments must be analyzed to determine whether they meet the requirements of the test strategy relative to test coverage, cost and time (manufacturing

beat rate). If the board must be tested before it is fully populated or before the BIOS and firmware are available, then traditional functional test is not an option.

Tradeoff analysis begins when multiple non-intrusive techniques can be applied; for example, when an FPGA, microprocessor and ASIC all have boundary scan capabilities and they are all connected to the memory bus. When this is the case, any one of these devices can act as a bus master and, as a result, function as the basis for non-intrusive memory tests. The question then becomes, which non-intrusive test method meets the requirements of the test strategy? In reality, no single test method is the correct answer in every case. The right method will depend on the requirements of the test strategy and these requirements will depend on the priorities or "care-abouts" established by the test engineering team.

Table 1 summarizes and codifies the tradeoffs among the various described non-intrusive memory test methods.

Tradeoffs	Functional	BST	PCT	FCT	ASIC/SoC	Microprocessor MBIST
Time of Availability	Late Stage	Early Stage	Early Stage	Early Stage	Early Stage	Early Stage
Structural Test	Medium	High	Medium to High	High	High	High
Third-Party Support	N/A	Yes	Yes	Yes	Yes	Yes
Chip DFT Structure	N/A	Yes	Most	Yes	Some	Yes
Board DFT Structure	No	Yes	Yes	Yes	Yes	Yes
Match to Board Needs	Medium	Medium	Medium to High	High	Low	Low
Board Overhead	None	Low	Low	Low	Low	None
Time to Develop Test	Long	Automated	Automated	Automated	Automated	Automated
Applied Test Time	Very Long	Medium to Long	Fast	Very Fast	Very Fast	Very Fast

Table 1: Memory test tradeoffs

The first aspect of developing a non-intrusive memory test strategy, which should be done during the board development process, is to consider when tests can be applied. Test engineers often refer to this as the time of availability. It indicates whether the test strategy can be initiated early in the circuit board design's life cycle or later. For instance, functional test can only commence when the board design is virtually complete and functionally operational. This would be a very late stage test method. The other non-intrusive techniques can be applied at early stages if the required board-level design-for-test (DFT) features are supported, such as JTAG access, power, clocks and target chips populated on the board.

Another high-priority care-about is the type of test – functional or structural – that is being considered. Functional test verifies the design assumptions while structural test verifies the quality of the manufacturing or assembly process. The most effective test strategies will include a high degree of structural test. With regards to their structural test capabilities, functional test and PCT are rated as medium since they depend on the features supported by the microprocessor in the design. Whereas functional test may be limited by the protocol operations that are functional at the time of test, PCT may be able to directly apply algorithmic structural tests much earlier, in which case its rating for structural test would then be changed to high.

Another key consideration is the availability of third-party tools which would automate the test development, test application, and debug and diagnostic processes. Not only will automation make the test development and application process easier, it will shorten the elapsed time of the process from start to finish. With the exception of functional test, all of the non-intrusive test methods mentioned in this paper can be automated.

Another set of concerns revolves around whether DFT structures are supported on the board and which chips support these structures. For example, functional test does not require chip-level DFT, but BST, PCT, FCT, and ASIC-based testing will require that the IEEE 1149.1 boundary scan/JTAG standard is supported on the board and at least in some of the chips. Furthermore, FCT and ASIC-based test will require embedded test instruments. Board-level DFT requirements will include the presence of headers to connect a boundary-scan/JTAG tester to the on-board scan path, as well as power and clocks from external sources. Again, late-stage functional test requires a fully functional environment, but no special DFT. The remaining non-intrusive techniques will require board-level DFT capabilities.

The test team must analyze the on-board devices to determine whether they include embedded test features and whether these features match the test requirements of the board's test strategy. This will involve an analysis of board-level requirements like test coverage, test application times and ease of operation. For example, functional tests using a microprocessor will typically only verify design assumptions. As a result, functional test would be rated medium with regards to matching board-level test requirements. In contrast, PCT may utilize the same microprocessor but apply tests with more structural content. Consequently, PCT would be rated higher than functional test with regards to matching board-level test requirements. Since FCT is implemented with IP embedded instruments tailored to the test requirements of a particular board, it will closely match the board's test requirements. Boundary scan and ASIC-based tests would be a low match since boundary scan is challenged to achieve the high speeds of advanced memory buses and ASIC-based tests are a better match for IC test than board test.

Another important tradeoff involves cost. One of the several metrics relating to a test system's total cost of ownership would analyze how much board-level DFT overhead is required to support non-intrusive tests. In this regard, functional test requires no board overhead at all, while all of the other non-intrusive test methods (BST, PCT, FCT, ASIC/SoC and microprocessor BIST) would require a relatively small amount of board DFT. On this metric then, these four methods are rated low-cost alternatives. If a board design called for a microprocessor or an FPGA that was strictly dedicated to test and it served no functional purpose, this would constitute a high overhead cost.

The primary metrics that are applied to the test process itself are test development time and test application time. When test code must be developed manually, the process is time-consuming and labor-intensive. However, when tests have been developed to be deployed via a third-party tool, test development time is usually a relatively minor consideration. In addition, tests developed to be applied by a third-party tool can frequently be re-used on similar or subsequent designs. This reduces the impact of a particular test development project by spreading its investment across multiple designs over time.

Test application times can range from seconds to minutes or hours. Generally, functional tests are very time consuming because the memory arrays must be operated in a fully functional manner, including any protocol handshakes and access latencies. Boundary-scan tests employ the

board-level serial scan chain to shift in data and address, and to operate the read-write protocol signals. This may be a slow process if the board-level scan chain is very long and asserting reads and writes involves several different protocol signals, each of which requires a scan in and an update to set the signal.

PCT can be very fast because it implements an algorithmic structural test programmed into the microprocessor. There is some overhead with PCT with regards to loading assembly language routines into the microprocessor's cache. The FCT method can have the shortest test application time since programming a tester into an FPGA can be done in a few seconds. Once the tester has been loaded, the algorithmic tests have already been optimized to match specifically the test requirements of the board. Because FCT tests are so closely matched to the requirements of the test strategy, the actual execution time of FCT tests is usually very short. Similarly, ASIC/SoC-based tests can be applied to test memory very quickly if they have been optimized to the board's test strategy. However, ASIC/SoC-based tests do not often match the goals of board test strategies.

Summary and Conclusions

Modern circuit boards have complex memory architectures that are becoming harder to test due to their high speeds, the high rate of data transfers on memory buses, complex protocols and the loss of test points on boards where a test probe can be placed. Coverage from intrusive probe-based methods of memory test and validation such as oscilloscopes and in-circuit test (ICT) systems is rapidly eroding. Frankly, these legacy test methods are quite challenged by today's aggressive test goals.

Memory test methodologies which combine several non-intrusive technologies such as boundary-scan test (BST), processor-controlled test (PCT), microprocessor MBIST technologies like Intel's IBIST and FPGA-controlled test (FCT) have reached the point where they can exceed the memory test coverage of legacy intrusive methods. These non-intrusive test technologies are available as automated software tools that can be applied through cost-effective, compact and standalone testers. When deployed, the coverage derived from each of these non-intrusive memory test technologies complement each other and, taken together as a unit, they provide a memory test toolkit capable of meeting the requirements of any test strategy. Because they are software-based, each non-intrusive test technology is extremely flexible and, as a result, can be targeted to the specific characteristics and restrictions of a particular board design. Combining the strengths of all of these non-intrusive test methods creates a powerful and versatile memory test platform that solves the test problems created by today's high-speed memory and memory architectures.

Learn More

Learn more about non-intrusive board test. Register for our whitepaper and learn the what NBT is, its economics benefits, and its test coverage assessed by the iNEMI-specified PCOLA/SOQ/FAM criteria.

Register Today!

Structural: *Devices*	P	Presence
	C	Correctness
	O	Orientation
	L	Live
	A	Alignment
Structural: *Connections*	S	Shorts
	O	Opens
	Q	Quality
Functional: *Devices & Connections*	F	Feature
	A	At-Speed
	M	Measurement

iNEMI PCOLA/SOQ/FAM Framework