Technical Note

Understanding the Value of Signal Integrity Testing

Introduction

Historically, design engineers have used signal integrity (SI) testing as a key part of the design and development of new systems and for sustaining qualifications. While SI testing is extremely valuable in the engineering prototype phase, it is not always the right tool. In fact, its value diminishes as the product design progresses. After debug of early prototypes is complete, powerful tools that perform temperature and voltage margin testing should supplement or replace SI testing, especially for the qualification of memory die shrinks and alternate sourcing.

The proper selection of memory design, test, and verification tools reduces engineering time and increases the probability of detecting potential problems. This technical note describes how these tools can be used to the best advantage, from conception of a new product through end of life. It also provides a detailed description of SI, its uses, and limitations. Finally, it presents an overview of margin testing tools and the value they can add.

Tools for Memory Design, Testing, and Verification

Five essential tools used for memory design are listed in Table 1. However, this is not a complete list because its focus is only on tools that can be used to validate the functionality and robustness of a design. When used properly, these tools lead to reduced design time and more robust systems.

Table 1: Memory Design, Testing, and Verification Tools

Tools[1]	Examples
Electrical Simulations	SPICE or IBIS
Behavioral Simulations	Verilog or VHDL
Signal Integrity	Oscilloscope and probes; possibly mixed mode to allow for more accurate signal capture
Margin Testing	Guardband testing and 4-corner testing by variation of voltage and temperature
Compatibility Testing	Functional software testing or system reboot test

Notes: 1. The logic analyzer is not included in this list, although most debug labs include this tool as an integral part of their design and debug process. However, due to the cost and time involved, it is rarely the first tool used to detect a system failure or problem. Rather, it is used to debug a problem detected by compatibility, 4-corner, or other testing.

The effectiveness of the tools during each phase of product development is shown in Table 2. Compatibility testing, for example, does not occur during the design phase because no board is available to test.

Table 2: Effectiveness of Tools During Product Development Phases

Tool	Design	Alpha Proto	Beta Proto	Production	Post-Production
Simulation – Electrical	Essential	Very valuable	Limited value	Rarely used	No value
Simulation – Behavioral	Essential	Very valuable	Limited value	Rarely used	No value
Signal Integrity	Unavailable	Critical	Limited value	Rarely used	No value
Margin Testing	Unavailable	Essential	Essential	Essential	Essential
Compatibility	Unavailable	Valuable	Essential	Essential	Essential

Five Phases of Product Development

Phase 1 – Design

In the design phase, a concept or idea is implemented in hardware. Because no prototype is available, only simulation tools can be used. Therefore, designers rely exclusively on electrical and behavioral simulation tools.

Phase 2 – Alpha Prototype

The first prototype is manufactured in the alpha prototype phase. During this phase, the prototype is expected to undergo some or all of the following changes prior to production:

- Software changes: BIOS, embedded software, operating system, etc.
- Motherboard changes: Impedance/termination; component vendor; trace length rerouted or shortened
- Functionality changes: ASIC, gate array, or FPGA; component or package type; repairing incorrect functionality
- Airflow, power supply, or chassis changes

The purpose of the alpha prototype phase is to find most or all of the problems in the system. Engineers must perform enough testing to ensure that the next prototype will be nearly production-ready, at least from a hardware perspective. Certain tools are more useful in accomplishing this than others.

Typically, the first tool employed in this phase is a boot-up and software check. A system boot-up or power-on can yield valuable information and, coupled with basic software checks, can provide data that will point to necessary changes within the system. A thorough software check may not be possible during this phase because basic hardware changes can still occur. Designers may, however, construct a limited number of alpha units for software debug.

SI testing is also an invaluable tool during this phase. SI testing incorporates the use of an oscilloscope or mixed-mode analysis to capture analog signals on a printed circuit board (PCB). These captures can be compared to simulation or device specifications to determine whether the device meets those specifications, and whether it has an adequate timing margin. If it falls short, timing or signal improvements can be made.

Margin testing should not be used in this phase, however, because the PCB design is constantly changing. Margin testing should only be used after the hardware is stable.

After running the appropriate tests, designers will have a list of changes that need to be implemented, including some that may need to be simulated electrically or behaviorally to ensure they have the desired effect.

Phase 3 – Beta Prototype

In the beta prototype phase, the final stage of prototyping occurs. The hardware is at or near production status and only minor changes are expected. A combination of testing is used to finalize the hardware and make the system production-ready.

Software, or compatibility, testing must be thorough and can be run alone or in conjunction with margin testing. The following types of software testing are very useful for debugging memory devices, and are described in further detail in subsequent sections:

- Reboot or power-up testing
- Memory test patterns, such as checkerboard, inversions, etc.
- System power-down or standby modes that affect memory, such as those that activate self refresh

Margin testing should be used extensively in this phase. Varying temperature and voltage levels to their extremes is very valuable in identifying problems and margin limits. The combination of software testing and margin testing is effective at catching most memory failures during this phase.

SI testing has limited value in this phase, but can be used to understand functional failures and to validate changes made during the alpha prototype phase. SI testing should not be used to verify signals or nets from the alpha prototype that were validated previously and have not changed.

If any additional modifications are made during the beta prototyping phase, electrical and/or behavioral simulation, or SI testing may be necessary to validate them.

Phase 4 – Production

Ideally, few changes to a system should be necessary in the production phase. Instead, the focus turns to performing sustaining qualifications. Systems may be in production for six months to ten years or more, and may use hundreds or thousands of components. For systems to remain in production, it is important that companies have a means of qualifying alternate sources or additional components.

Sustaining Qualifications

Sustaining qualifications consist of performing a series of qualification tests on a component after a system is in production. These tests ensure that an adequate supply of components are available for uninterrupted production.

In the production phase, the system is stable. The only requalifications needed are sustaining qualifications when a component changes, a die shrink occurs, a second source is qualified, etc. A sustaining qualification will determine if a modified component passes or fails in a system.

Motherboard traces and layouts, verified by SI testing in the design phase, rarely change in the production phase. Therefore, SI and simulation testing are not required during this phase. Moreover, SI testing cannot catch failures that have historically caused

system or memory-related issues. For systems that use memory, SI is a poor test tool for the production phase because it does not catch all problems. The preferred tools for sustaining qualifications are compatibility and margin testing.

Phase 5 – Post-Production

Some systems, such as MP3 players or DVD recorders, require no testing or qualification after production ends. However, many other systems may require sustaining qualifications and support. Memory upgrades, for example, are common for notebooks, PCs, and other devices, and can be critical for years after production ends. As in Phase 4, margin testing and compatibility testing are the keys for sustaining qualifications in the post-production phase.

Signal Integrity Testing

Signal integrity (SI) testing is the process of capturing oscilloscope images (scope shots) of signal changes over time, then evaluating the images visually for out-of-spec conditions. This is typically a time-consuming process that requires a high level of engineering expertise.

Figure 1 shows a typical SI scope shot, including the following signals from a single DDR SDRAM component:

- BA1 – bank address 1
- CS – chip select signal
- CK – differential system clock (noninverted clock)

Figure 1: Typical Signal Integrity Shot from an Oscilloscope

The following useful parameters can be observed from this scope shot:

- Ringing or overshoot/undershoot: JEDEC specifications for memory components restrict the amount of allowable overshoot and undershoot. SI testing during alpha prototyping can uncover violations of these specifications.
- Timing violations: Signal timing is easily observed from these scope shots. Examples of timing verifications are:
  - slew rate (identification of weakly driven or strongly driven signals)
  - setup and hold time

- clock duty cycle
- crossing of differential signals such as CK and CK#
- relationships between control and data signals relative to the clock
- bus contention (two signals driving the bus at the same time, causing a conflict)

If any of these parameters are out of spec, they will likely cause a system failure that compatibility and margin testing can easily reveal. After these tools have identified the problem, other tools, such as a logic analyzer, can be used to determine its cause.

Limitations of SI Testing

SI testing has become more difficult and time consuming. This section discusses some of the difficulties encountered during SI testing, especially with FBGA packages.

FBGA Packages

Migration to FBGA Packages

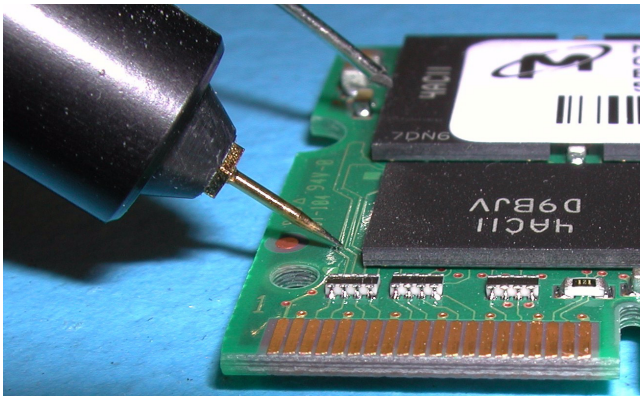
Even though TSOP packages are cheaper to manufacture and are more easily probed, DRAM has migrated to FBGA packages for the following reasons:

- **Reduced parasitics.** FBGA packages have reduced parasitics due to capacitance, inductance, and resistance. As system speeds continue to increase, decreasing parasitic values is critical to increasing clock frequencies.
- **JEDEC requirement for DDR2.** The FGBA package is the JEDEC standard for all DDR2 manufacturers.
- **Space requirements.** Many applications have limited space. FBGA packages reduce the board footprint and total volume used by DRAM.
- **Future upgradeability.** In order to gain headroom for higher speeds, FBGA packages provide the option of flip-chip bonding. This further reduces parasitics.

Difficulties Probing FBGA Packages

It is very difficult to place an oscilloscope or logic analyzer probe on signals under an FBGA package (see Figure 2). It may be possible to probe a connector, trace, or via on a PCB. However, the signal measured is not representative of the signal at the DRAM. If a non-monotonic signal or noise is observed at those locations, the signal observed at the DRAM will be different. Unless the signal is probed directly at the DRAM, you cannot get a true representation of signal quality.

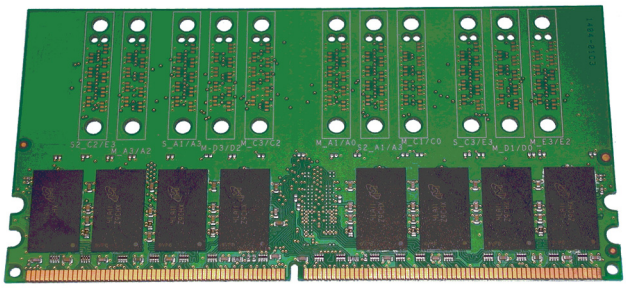
Figure 2: Probing FBGA Packages



Special Tools for Probing FBGAs

Because of the difficulty in probing FBGA packages, some companies have developed special tools to assist in testing these packages. These tools often include a special software package with a DIMM (see Figure 3) designed to enable a logic analyzer to probe signals without affecting the operation of the module. Such tools can provide valuable information during debug or product design, but can also increase development cost.

Figure 3: Example of Signal Integrity DIMM



MCM Packages

Some designs use multichip modules or MCMs to combine a variety of chips together in a single package. These packages are either covered with a mold compound or sealed hermetically, so the internal signals are unavailable to be probed.

Interference with Signal Quality

Using a probe to measure the signal integrity of a circuit changes the signal being measured. Problems can be introduced or exacerbated by the addition of capacitance, or they may disappear. Although active or FET probes are available, this condition is likely to become more common as frequency increases, especially in systems that have a point-to-point architecture.

Inability to Detect Many Memory Issues

Although essential for system bring-up and validation, SI testing is a poor tool for memory qualification and sustaining qualifications.

Recommended Uses of SI Testing

During the process of performing many internal qualifications of memory devices, Micron has generated a database of about 200,000 individual scope shots. This has resulted in the following discoveries relative to the uses of SI testing during product development:

- SI testing used at an early stage of development is instrumental in catching failures and resolving significant errors and issues.
- SI testing should be used to verify that board changes have maintained or improved signal quality.
- SI testing has little value after the board design is stable.

SI Testing Within Micron's Overall Test Process

Micron's internal test process has migrated away from SI testing. Our extensive experience with internal qualification, qualification of new die revisions or engineering experiments, and debug of customer issues has resulted in the following internal process:

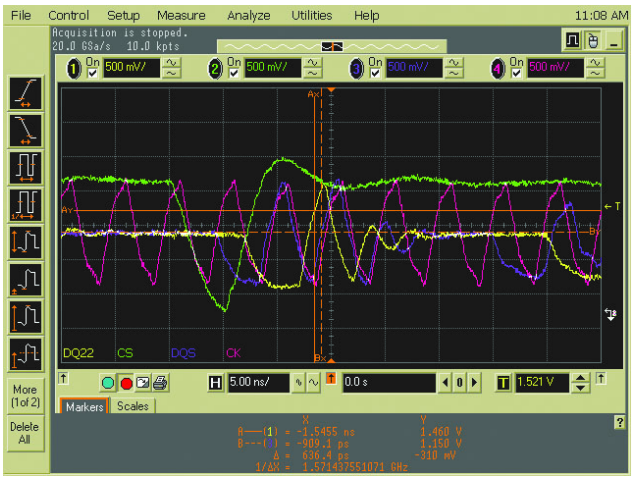
1. Compatibility and margin testing are used to validate or test a system with memory. If a failure is detected during compatibility or margin testing, additional diagnostic tools are used to isolate the failure.
2. If software can determine the failure type in a system (address, row, single cell, etc.), a memory chip or module is isolated and tested. An attempt is then made to duplicate the failing condition in a memory test fixture.
3. If software cannot provide failure details, the memory can be removed from the system. Component or module testing is then performed to find the failure.
4. A logic analyzer can be used to determine the failing issue/pattern or out-of-spec condition.
5. Although an oscilloscope can be used for SI testing, other tools have been found to be more effective in quickly testing, validating, and debugging systems. These tools enable engineers to quickly come to root cause analysis and correction.
6. Guardband testing is performed.

Findings of SI Testing During Micron's Self- Internal Qualifications

Micron's internal qualifications have produced the following findings:

- Compatibility testing and margin testing regularly exposed problems or issues that can occur in a system. They have proven to be reliable tests for robustness and for discovering problems.
- SI testing did not find a single issue that was not identified by memory or system-level diagnostics. In other words, SI testing found the same failures as the other tests, thus duplicating the capabilities of margin testing and software testing.
- SI testing is time consuming. Probing 64-bit or 72-bit data buses and taking scope shots requires a great deal of time.
- Equipment for SI testing is costly. To gather accurate scope shots, high-cost oscilloscopes and probes are required.
- SI testing takes up valuable engineering resources. A high level of engineering analysis is needed to evaluate scope shots.
- SI testing does *not* find all errors. Compatibility and margin testing find errors that are not detectable by SI. As an example, Figure 4 shows two scope shots: one from as system that passes and one from a system that fails. It is impossible to determine which system is which.

Figure 4: Examples of SI Scope Shots – One Failing, One Passing



Misconceptions of SI Testing

Misconceptions exist about the use of SI testing in the semiconductor industry. Based on Micron's experience, we offer the following responses to some common misconceptions.

- **Misconception:** Some failures will never be caught by diagnostics; therefore, SI testing is needed to identify them.
  Micron's response: Our experience has shown that some customer diagnostics may not be thorough enough to catch failures. Rather than spending many engineering hours on SI testing, our recommendation is to develop a better diagnostic suite or find a commercially available diagnostic program, and add new data patterns as needed.
- **Misconception:** SI testing provides data on the trends of signal quality, and this information can be used to predict failures so they can be avoided in the future.
  Micron's response: A system is designed initially so that signals meet specifications, including rise time, fall time, setup, non-monotonic signals, etc. Experience has shown that continued SI testing does not catch additional failures, nor does it predict the likelihood of failures after a circuit board design has stabilized. Margin and software testing are much better predictors of failures.
- **Misconception:** Temperature/voltage margin tests (4-corner tests) will not catch obscure failures missed by SI testing.
  Micron's response: On the contrary, the stresses of voltage and temperature actually aggravate marginal or obscure failures and are therefore the most effective tools for catching system failures.
- **Misconception:** Most potential problems can be detected by SI testing.
  Micron's response: Our experience has shown that most failures are attributed to timing conditions internal or external to the DRAM. SI testing cannot predict these types of timing failures. Once a failure has been verified by compatibility or margin testing, a logic analyzer can be useful to isolate timing violations.
- **Misconception:** Variations between memory component vendors and PCB vendors can impact SI testing.
  Micron's response: PCB differences are minimal in most applications. This is because vendors use common gerbers or board layouts in most cases. Our experience is that impedance problems or problems due to parasitics are not detected SI testing, but by other tests.

Alternatives to SI Testing

Alternatives to SI testing can be used for the development of systems and for memory qualification and testing. This section provides a brief description of these tools and how to use them.

Compatibility Testing

A variety of off-the-shelf software products are available for compatibility testing on a PC platform. While Micron does not endorse any particular product, the following partial list is provided:

- PC Doctor
- Winstress
- Quicktech
- RST Pro
- AMI Diag
- PC Certify
- Tuff Test

When considering software tools, companies should look for those that support dynamic upgrading and choose a program that will incorporate new diagnostics as needed to catch previously unknown failure mechanisms.

Other products, such as such as consumer, embedded, and networking, are more difficult to test than PCs. Designers for these products typically develop home-grown tools or use none at all. Making home-grown tools more robust can provide greater benefit than SI testing.

Sometimes, a memory-specific test may be impossible in a system such as MPEG decoding or packet transfers for networking. In these cases, the other tools described in this section should be used.

Margin Testing

Margin testing has been valuable in identifying device performance limits and system issues. In margin testing, systems are stressed by varying temperatures and voltages, exposing the DRAM and its controller to conditions that may reveal potential system failures.

Figure 5 and Figure 6 on page 11 show examples of how device timing and power vary over temperature. Margin testing enables variation of these parameters to test a system over these rated parameters.

Figure 5: Example of Device Timing Variation over Voltage and Temperature

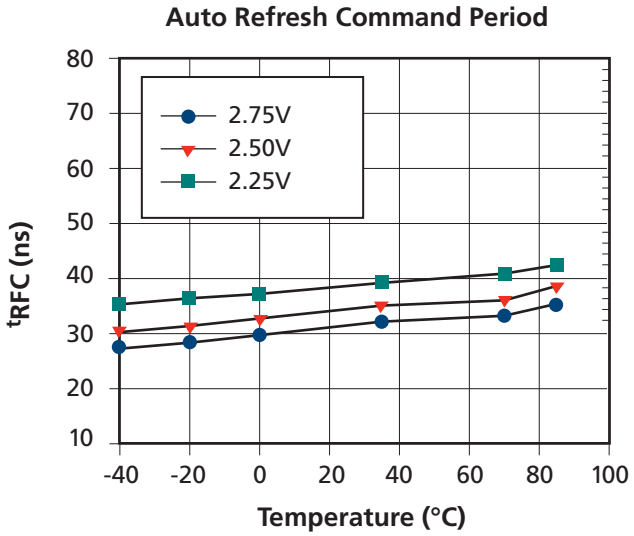
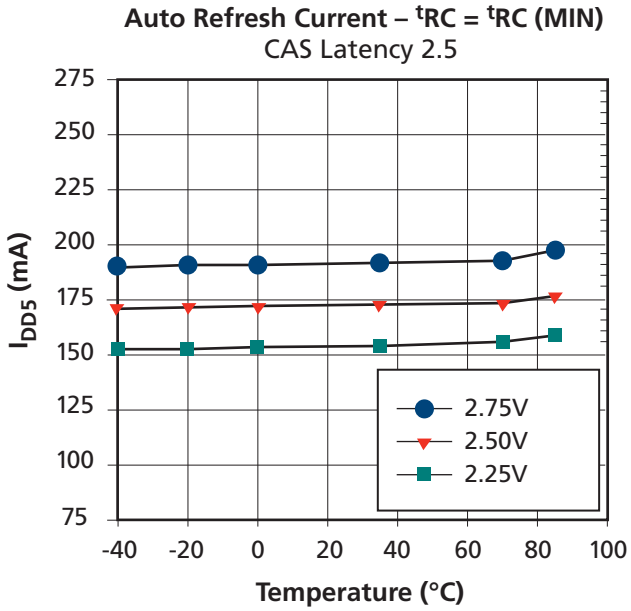


Figure 6: Example of Device Power Variation over Voltage and Temperature



4-Corner Test

One type of margin test is a 4-corner test. For example, a system may have specified MIN and MAX voltages of 3.0V and 3.6V, and MIN and MAX temperatures of 0°C and 70°C. An external voltage regulator is used to control the supply voltages to the DRAM, and the temperature is controlled by the test oven. The system undergoes the following four corner tests:

- **Corner 1:** MAX voltage, MAX temperature (3.6V @ 70°C)
- **Corner 2:** MAX voltage, MIN temperature (3.6V @ 0°C)
- **Corner 3:** MIN voltage, MAX temperature (3.0V @ 70°C)
- **Corner 4:** MIN voltage, MIN temperature (3.0V @ 0°C)

The type of testing done at these corners can vary, but a typical procedure is:

1. Let the system temperature and voltage stabilize inside a temperature chamber.
2. Choose a suite of tests to run at that corner.
3. If any failures occur, determine their root causes.

The 4-corner test has proven to be one of the most effective methods of testing memory, and is reasonable in terms of test time and resources required.

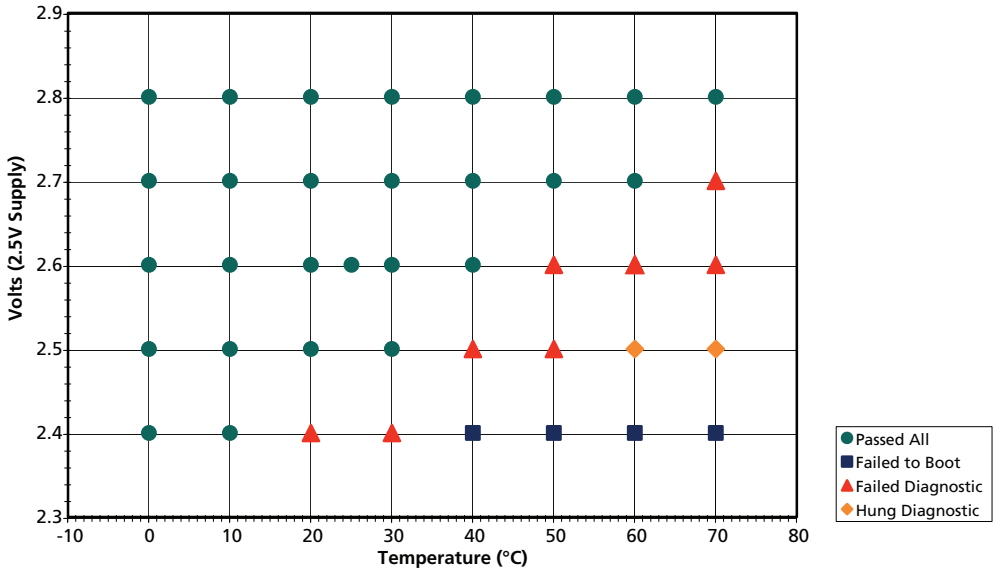
2-Corner Test

A variation of the 4-corner test is the 2-corner test. In a 2-corner test, the voltage regulator keeps the DRAM supply voltage at a constant level. Then, the system is tested at two temperatures (MIN and MAX).

Guardband Test

Another type of margin test is a guardband test, which uses many voltage and temperature test points (see Figure 7). This type of testing is more expensive and time-consuming than other test methods.

Figure 7: Guardband Test



Power Cycling

Another useful tool is stressing the system by repeatedly cycling system power on and off (booting the system). This should include both cold and warm boots. A cold boot occurs when a system has not been running and is at ambient or room temperature. A warm boot occurs after a system has been running for a period of time and the internal temperature is stabilized.

During boot-up or power-up, the following events can cause errors to occur:

- **Ramping up power supply voltages.** If there is an intermittent voltage ramp-up problem, it might only be detected by repeated cycling.
- **Initialization of the memory.** In order to meet JEDEC and industry standards, the memory controller must follow a strict initialization sequence. If an intermittent problem occurs during the initialization sequence, it may only be detected by a series of repeated initialization attempts.

Self Refresh Cycling

DRAM cells leak charge and must be refreshed often to ensure proper operation. One of the key means of saving power in a system is to use self refresh when the memory is not used for long periods of time. It is critical that the memory controller provides the proper commands when entering and exiting self refresh; otherwise, data could be lost.

Similar to power-up cycling, self refresh cycling is a useful compatibility test. If an intermittent self refresh enter or exit problem is present, repeated cycling can help detect it.

NOTE: Applications that do not use self refresh should skip this test completely.

Benefits of Evaluating Memory Qualification Process

An evaluation of the memory qualification and validation process can provide many benefits to your company:

- An immediate reduction in engineering hours during memory qualifications, especially sustaining qualifications, is achieved.
- Use of compatibility or margin testing results in more effective and thorough identification of actual failures than SI testing.
- Replacing SI testing with compatibility or margin testing results in faster memory qualifications, especially sustaining qualifications.

Each company has its own methodology for design development and qualification. Micron encourages each company to find the method that works best. Our experience has shown that compatibility testing and margin testing should strongly be considered as alternatives to SI testing for the development process.

Revision History

Rev. C 12/09

- Updated format
- Minor grammatical changes
- Updated “Special Tools for Probing FBGAs” section to be more generic

Rev. B 6/05

- Converted document to SCOUT format
- Fixed spelling errors

Rev. A 12/04

- Initial release