

BIST for Deep Submicron ASIC Memories with High Performance Application

Theo J. Powell, Wu-Tung Cheng*, Joseph Rayhawk*, Omer Samman*, Paul Policke, Sherry Lai

Texas Instruments Inc.
PO Box 660199, MS 8645
Dallas, TX 75266-0199

* Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070

Abstract

Today's ASIC designs consist of more memory in terms of both area and number of instances. The shrinking of geometries has an even greater effect upon memories due to their tight layouts. These two trends are putting much greater demands upon memory BIST requirements. At-speed testing and custom test algorithms are becoming essential for insuring overall product quality. At-speed testing on memories that now operate in the 10 to 800 MHz range can be a challenge. Another demand upon memory BIST is determining the location of defects so that the cause can be diagnosed, or repaired with redundant cells. A tool and methodology that meets these difficult requirements is discussed.

1 Introduction

In modern design, embedded memories occupy a large percentage of a chip design [Rajsuman]. Furthermore, the memory array is usually the densest physical structure and made from the smallest geometry process features available. Consequently the memory arrays are likely to be more sensitive to defects than the logic portions of the chip. To ensure high chip quality, the embedded memories must be extensively tested.

Due to the regular structure of memories, memory BIST techniques have been successfully used to ensure the quality of embedded memories [Schanstra]. Tools are available to automatically synthesize and insert BIST logic around embedded memories. Most have a limited set of fixed algorithms that can be applied. However, due to the shrinking of geometry, additional defects appear that are not screened by conventional test algorithms. These defects can be structure dependent or foundry dependent. To detect these defects, memory BIST tools need to allow customized test algorithms to be included in the applied test. Section 2 describes the Texas Instruments (TI) ASIC memory design flow. The flow is automated to permit RAM BIST to be created quickly for a variety of RAMs with sizes that match design requirements. Section 3 reviews waveforms of a few defects encountered in RAMs. To assure defects are screened, a RAM testing cover measure is described in

Section 4. The BIST algorithm used is composed of several sub-algorithms to increase the test screening ability. Since some defects are timing sensitive, besides running at system frequency, the test algorithm must be applied with a memory operation on each clock cycle, similar to normal system operations. Section 5 explains how some of the requirements are met with the memory BIST tool. Additional RAM BIST requirements for meeting test quality requirements are described in Section 6. Section 7 describes an application to test embedded RAMs in TMS320C6414T, a high performance DSP. The application tests 84 embedded memories ranging from 400 MHz to 800 MHz frequencies to meet at-speed test requirements. Section 8 provides our conclusions.

2 Memory BIST ASIC Flow

When adding test for memories in an ASIC design, the flow must be flexible. It must support a variety of RAM types including single port, dual port, multi-port RAMs and others. Dual port RAMs have two addresses, one each for a data input and output pair. Multi-port RAMs have multiple input ports and/or multiple output ports each with a separate address. The RAMs can have different address and bit width sizes and different column/row mux ratios. RAM design requirements are used to create scripts that are used for building the physical memory layout code. See Figure 1. Corresponding RAM BIST models, synthesis models and simulation models are pulled from the ASIC model libraries for inputs to the memory BIST tool for RTL creation, for synthesis and for simulation of memory BIST logic. RAM design requirements include:

- The number and type of RAMS to be tested with each BIST controller
- The frequency to which the test is to be applied
- Whether or not pipeline registers will be included between the RAMs and the BIST controller to reduce critical timing paths

An RTL testbench is also created which can be used to simulate the application of the memory BIST controller being applied to the RAM simulation models. Scripts are

also generated to help with the synthesis and to drive the simulation needed for verification.

The BIST controller can be simulated at the RTL level for functional verification. After the BIST controller is synthesized to the gate level description, it is simulated with timing to verify that the controller can test the RAMs at the system design speeds. Alternatively, static timing analysis may be used to verify that the timing is met. Insertion of the BISTed RAMs into an RTL design description is performed before synthesis and is followed by the backend physical design flow and manufacturing. Simulation results are translated into tester languages for testing the RAMs during manufacturing.

Designs with a large amount of SRAM may use redundancy for the purpose of RAM repair. The RAM BIST test is used for screening defects, and redundant parts of the RAMs are selected to replace defective rows or columns with laser or electronic fuses. For this reason the protocol between the diagnostic module of the BIST controller and the tester must be efficient so that RAMs can quickly be repaired when possible after defective rows, columns or solo bit defects are identified. The same diagnostic module is also used for non-repairable RAMs to provide diagnostic data that is gathered for manufacturing and design improvements.

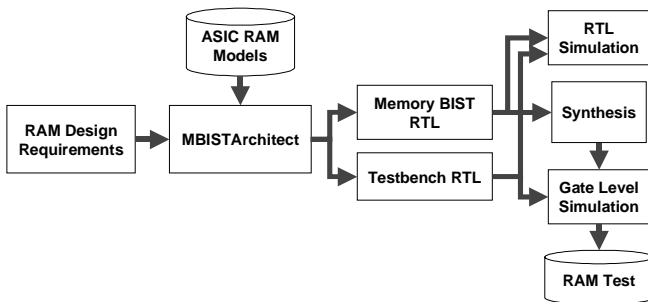


Figure 1: Memory BIST creation flow

3 Submicron RAM Defect Review

As the densities of semiconductor technologies increase, additional defects become more prevalent and need screening. The standard defects categorized by Prof. Van De Gore [van de Gore] must still be screened; however, the more subtle and difficult to screen defects must also be screened. For example, some resistive paths in the array are shown as defects only when noise or background patterns exist with the test. The defects can be due to the architecture or layout of the RAMs. More specific test algorithms are needed for screening these subtle defects, which at times may be limited to specific RAM types or sizes. The RAM BIST tool permits the user to customize test algorithms to screen the specific defects limited to certain RAM types.

Figure 2 shows waveforms for bit line (solid line) and its bit line bar (dashed line) for a write immediately followed by a

read. A writing of a 0 into an address cell is assumed. A clock cycle begins with a pre-charge on both bit line and bit line bar. After the pre-charge has completed, the bit line control permits a write or read on the bit lines according to an address. During the write operation, the difference on the two bit lines decides what value is written to the cell. During the read operation the sense amp is enabled to detect the voltage difference in the cell being addressed via these two bit lines.

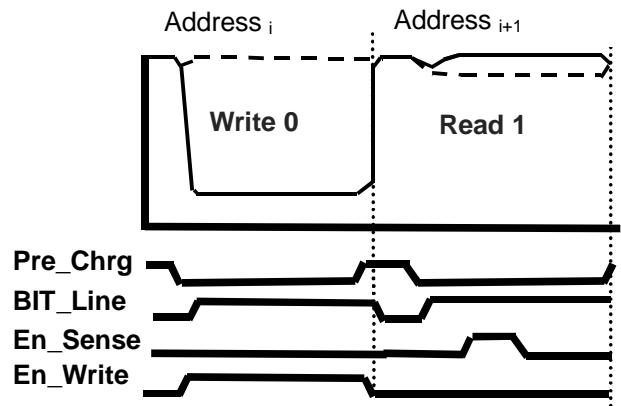


Figure 2: Sense amp failure example; defect-free waveforms

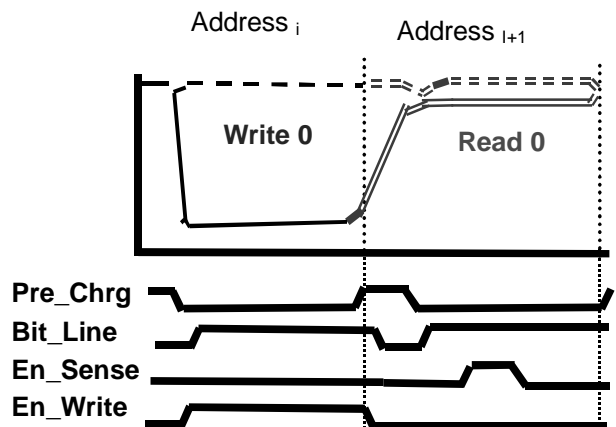


Figure 3: Sense amp failure example; defect on read waveforms

Figure 3 shows a failure caused by a slow to rise pre-charge. The bit line difference on the second clock is not sufficient to overcome the difference already existing on the bit lines left over from the previous write. Such a failure could be the result of a resistive short to the bit lines that drains some of the charge.

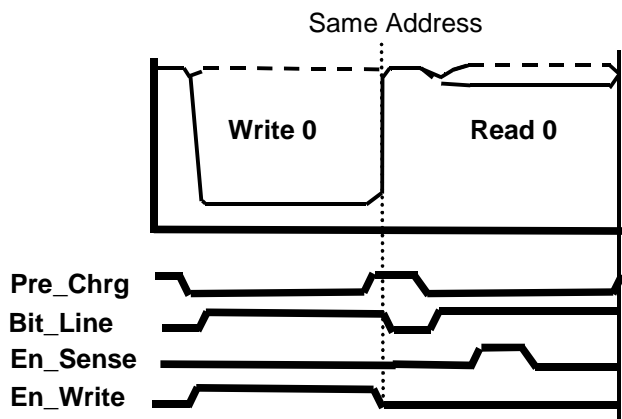


Figure 4: Waveforms for defect free write operation

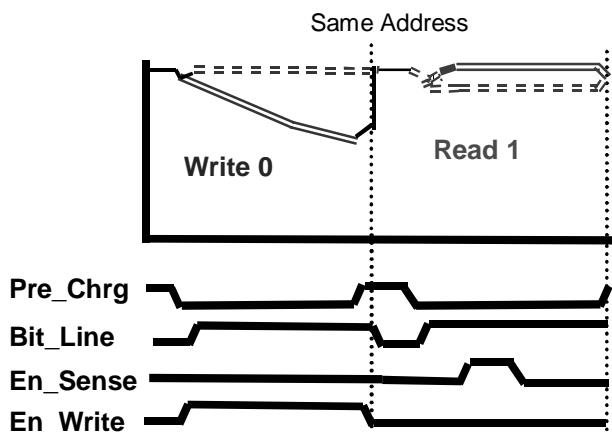


Figure 5: Waveforms of failure on write

Some failures are frequency dependent while others are not. Most TI SRAMs are clocked and therefore the pre-charge and sense pulses occur after the leading edge of the RAM clock and are independent of the clock frequency. . Figure 4 shows the waveforms of writing a 0 to an address and then immediately reading the 0 from the same address. The operation of the RAM occurs within a period after the edge of the clock. The defect shown Figure 3 is an example of a non-frequency dependent failure. As long as the frequency is within the RAM specification, the failing action occurs within the period for the RAM to operate. An example of frequency-dependent failure is shown in Figure 5. A resistive defect causes the write to have insufficient time to store enough charge in a cell. When the cell is read there is insufficient charge to switch the sense amp from its previous state. These defects can occur at any location of the array and be dependent on location, neighboring cell or signal line charges. A comprehensive test insures that these types of defects are screened so that the memory can be repaired, or if necessary, the die is discarded.

4 Algorithms which Screen Sub-micron Memory Defects

Some sub-micron defects are screened only by tests applied with column March algorithms. An example of a defect screened by column March algorithms is a sense amp defects with resistive short between a cell along a column as mentioned above. Figure 3 shows a slow to rise sense amps resulting in a failure. One effective screen using a column March addressing is to read data followed immediately by another read of the opposite data. Since column March addressing is being used, the back-to-back reads are reading cells along a memory array column. Another effective screen on a column is to write data followed by a read of the opposite data. During the column March part of the test used at TI, a string of 0's is read followed by reading a string of 1's at consecutive column addresses. When a column is read with a physical checkerboard in the array, the opposite data will be read at each clock. Another pass is executed with the array holding a physical checkerboard pattern. It reads a cell after which the same data is written back into the cell. At the next address, the opposite data is read because the array holds a physical checkerboard background. In Table 1, the first two rows show that the column March A and B algorithms cover reading the opposite data after a write and read operations. The Rd R~d label means that a read operation with a d pattern is followed by a read of the opposite data pattern. Similarly Wd Rd means that a write of d pattern is immediately followed by a read at the next address of the opposite data pattern. Included in the checkerboard algorithms are passes where each address is read, and immediately after the read, the same data is written back to the cell at each address. The next address includes a read, which will have data opposite the data that was read and written at the previous address.

Some resistive shorts between neighboring columns are best screened with a row March addressing. The March 13N algorithm includes a sequence at each address of a read, write and second read. At the next address the first read will read the opposite data read of the second read from the previous address. To obtain a read of the opposite data after writing to the RAM, a March 11N algorithm is created by replacing two march passes of the March 13N algorithm with read followed by write before going to the next address. At the next address, the first read will have opposite data to the data of the last write of the previous address. Table 1 shows where the column and row consecutive read-read and write-read operations are covered.

Resistive leaks between columns are sometimes only manifested when the neighboring columns have the opposite data to what is being read. To screen for this, a March 13N with background patterns of 0/F, 3/C, 0F/F0 and 69/96 and 5/A with March 11N are used. The probability of detecting resistive leaks is increased. The boundary conditions of neighboring cells having different values of a cell being

tested for at least one test increases the probability for screening the defects. Table 1 shows the background patterns that are employed during the March algorithms. The need for the selected algorithms was obtained through the application of CPU memory BIST over high volume products. CPU memory BIST uses an on-board CPU to test RAMs. CPU BIST permits different algorithms to be applied to test their effectiveness toward lowering DPPM. Designs do not always allow CPU BIST to be an available option since some RAMs are not easily accessible by the CPU for testing. It is necessary, therefore, to make sure that the dedicated memory BIST test includes the algorithms necessary to screen the defects.

Algorithms	Column Rd R~d	Column Wd R~d	Row Rd R~d	Row Wd R~d	Background pattern
Column March A	X	X			Checkerboard
Column March B	X	X			~Checkerboard
March 13N				X	0/F, 3/C, 0F/F0, 69/96
March 11N			X	X	5/A

Table 1: RAM testing covering measure

5 Creating Algorithms with Memory BIST Tool

To support test algorithm variation for different memory structure, the MBIST Flex™ feature of the MBISTArchitect™ memory BIST tool is used. This allows the creation of customized RAM test March algorithms. It allows any linear address sequence to be used. A linear sequence is a sequence where the next address is deterministically derived from the current address. Linear sequence covers all kinds of physically column based or row based address sequence. Three different sequences are shown in Figure 6. Furthermore, it allows users to define the operation and data backgrounds within each address. All operations and data backgrounds used in Table 1 are supported.

The RAM test includes Iddq measurements with the RAMs in background patterns of checkerboard, inverse-checkerboard, all 0's and all 1's. The retention tests and Iddq tests are combined together since their total test time is relatively long. To reduce overall test time, it is desirable to perform Iddq tests for all RAMs simultaneously. The MBIST Flex feature allows the retention points to be synchronized. All RAMs can reach a common point in the test and be used for retention and Iddq testing. The controller idles when the retention points are encountered in the tests so that the common backgrounds can be reached. When all memories are in a retention point, the BIST controller can send a signal to inform the tester of the beginning of retention time. After the retention time, the tester sends a signal to the BIST controller to resume the

suspended test. When multiple memories are tested sequentially, it is still desirable to group all retention tests in parallel to a common background pattern. The tool allows aligning all retention tests even during sequential test by using interleaving the RAMs test steps. Figure 7 (a) shows a normal serial test and (b) an interleaving serial test to allow for retention tests with all RAMs in a common background pattern.

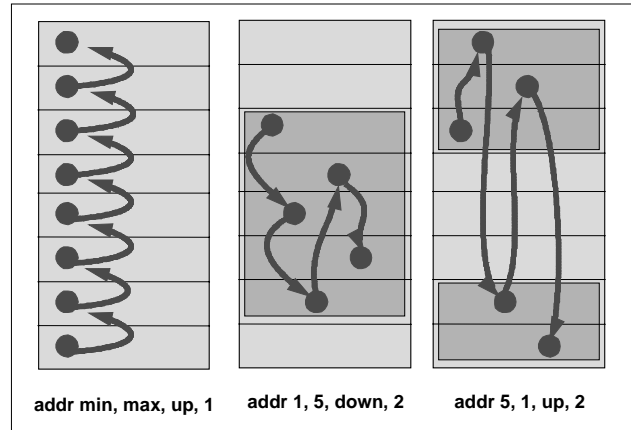


Figure 6: Three examples of linear address sequences for memory test march algorithms

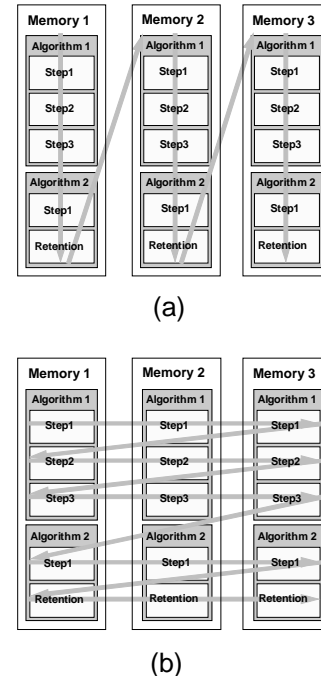


Figure 7: Serial Test.(a) Normal and (b) interleaving

6 RAM BIST Controller Requirements

Some defects can only be detected when a RAM operation changes the full word on each clock. Defects that slowly leak charge or neighborhood coupling leakage can bleed the charge within a window of time. If a slower frequency is used, some defects have enough time to recover and not be screened. The range of design frequencies varies from 10 MHz to 800 MHz. Above a frequency level of 300 to 400 MHz, pipeline registers are frequently required both functionally and with BIST operation, especially when the controller is not physically located next to the RAMs. To operate at-speed, MBISTArchitect includes a command in which the user can specify the number of necessary pipeline stages before and after the memories. Besides creating and connecting these pipeline registers, the BIST controller adjusts its behavior to accommodate the extra cycles used by these pipeline stages.

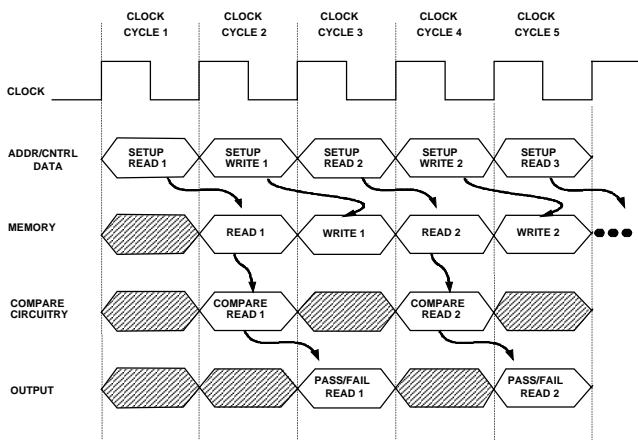


Figure 8: A read/write operation with MBIST Full-Speed

To screen for these timing defects, each operation of read or write must occur on each clock; otherwise, the RAMs are actually tested at a lower frequency.

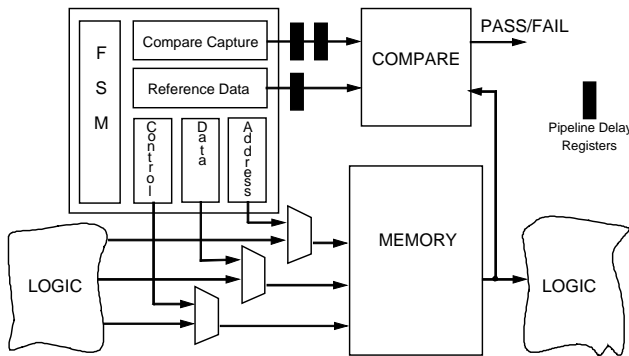


Figure 9: A controller using MBIST Full-Speed

Typically, the BIST controller is implemented with single clock. In single clock environment, the BIST controller normally takes more than one clock cycle to activate, execute and measure each read or write operation. During system operations, a string of reads and/or writes can execute one for each clock cycle. Memories should not have idle cycles unless the cycles are required for the memory operation itself. To achieve a read or write on each clock cycle, the read and write operations are condensed as shown in Figure 8 for a synchronous memory with typical read/write behavior. To accommodate all kinds of timing behavior of read and write of one memory, the MBIST Full-Speed™ feature creates appropriate pipeline structure within the BIST controller itself such that it can launch read or write operation every cycle and predict the correct output from the memory precisely. Figure 9 is the block diagram of the controller for the behavior of Figure 8. There is one pipeline register to delay the reference data and two pipeline registers to delay the capture signal. The numbers of pipeline registers depends on the cycle time of read/write behavior. This feature not only achieves the test quality needed for timing defects, but also reduces test time significantly.

Another requirement for TI ASIC RAM BIST controllers is to gather failed data for repairing the repairable RAMs. For non-repairable RAMs, failed data is also collected and used to provide information for manufacturing and RAM design improvements. Although MBISTArchitect has an option to provide a repair module to analyze and propose repair solutions, it was not used. The repair information for the different RAM types and options to repair are contained in the tester software at TI. A dedicated diagnostic module in a BIST controller is used instead to collect failure data and send it out to the tester, as shown in Figure 10. The diagnostic module permits the user to specify what data should be scanned out when a failure occurs.

The data which can be selected to be scanned out for each failure includes the RAM data output, the comparison between the Qs and the expected data, the RAM address the nth read or write operation at the address, the state of the controller state and the memory number if RAMs are being tested in series. The scan out clock can either be the BIST clock or a separate slower tester clock. When a separate slower clock is used, proper clock synchronization is needed to ensure the tester can collect the diagnostic data properly when failure occurs.

When a separate slower clock is used a tester controller protocol is necessary for the transfer of data. The interface ports to the tester include a FAIL signal to communicate when failures occur and when the tester needs to scan out failed data. After the FAIL signal is observed at the tester, the diagnostic module can start the process to scan out failed data. After the failed data is completely scanned out, the FAIL signal will go off. To ensure proper handshaking, the off state of the FAIL signal cannot be too short. If the off

state is too short, the tester might not recognize that the controller is processing a second failure.

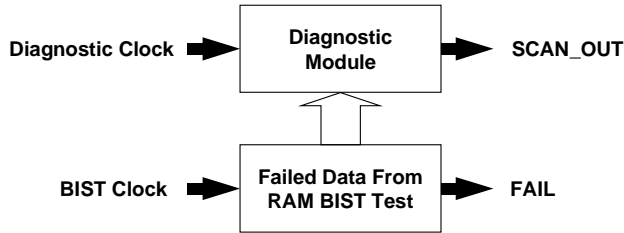


Figure 10: Diagnostic Module

The failed data arrives to the diagnostic module in parallel and is scanned out in serial, one-bit at a time. Therefore, it is possible that more failures can be screened before the previous failure data is scanned out. If more defects are detected before failed data is scanned out, the memory BIST controller must halt and wait until the failed data has been scanned out and then resume to avoid losing failure information.

Further complicating the protocol occurs when pipeline registers are used. Pipeline registers are often unable to halt which means that some results of the test will be lost as the registers are flushed. An approach was developed that avoids losing failure data. The controller restarts from the beginning of the test. The read comparisons are masked until the test has just passed the last defect scanned out. The screening starts again just past the point of the previous screened defect. This process continues repeatedly as needed until all defects are reported. With this approach, failed data is not lost when using pipeline registers.

If pipeline registers are not used, an option of halting the controller can be used. The controller will halt when another defect is screened and the previous failure has not yet completely been scanned out. The controller resumes after the failed data is moved to the scan registers after the completion of the scan. The process is repeated until the test is completed. The non-restart is preferred if it can be used since the additional tester time to restart and the additional logic to perform the restart are avoided. Both the ability to restart or just halt and resume are used depending on the RAM BIST design constraints.

To implement BIST, MUX logic is needed to allow BIST controller to operate the memory. Also, to simplify ATPG effort for logic around memories, it is desirable to add bypass scan to control memory outputs and observe memory inputs. Some TI memories have built-in embedded MUXes and bypass scan already. The tool can recognize the existing MUXes and bypass scan logic and hook the BIST controller to the RAM properly without duplicating BIST collar logic.

7 TMS320C6414T Application Example

The TMS320C6414T is a high performance DSP chip designed for 90 nanometer technology. Memory testing has always been a huge challenge for complex, high performance designs. This memory BIST solution was able to meet this challenge without impacting functional timing paths and without undue area overhead cost. There are three memory BIST controllers implemented on TMS320C6414T. The number of controllers selected was based on the memory types and the operating frequency of the memories. All controllers have been configured for pipelining and diagnostic monitoring. The diagnostic block captures the read/write states, algorithm state or controller states, memory numbers, addresses and fail data. All controllers are running individually with each memory tested in series per controller. To increase the detection of performance related defects; there will be a ten percent frequency increase over the normal functional frequency. The first memory BIST controller tests 22 single port memories, ranging from 80 to 1024 words and 32 to 64 data width, with a 400 MHz clock frequency. These memories are not repairable, though, the controller does have diagnostic capability for memory characterization and failure analysis. The second memory BIST controller tests 31 single port repairable memories, ranging from 128 to 1024 words and 6 to 42 data width, with an 800 MHz clock frequency. The last controller tests 32 single port repairable memories, with the same 8192 words and 32 data width, and an 800 MHz clock frequency. The size of the controllers range approximately from 7k to 8.5k gate. The size is dependent on the number of RAMs tested, whether they are tested in series or parallel and the level of pipeline necessary to support.

Conventional memory testing with a BIST controller tests either one RAM at a time or the same type of memory concurrently. In actual designs, the memories are operating simultaneously and the interaction can produce noise that can trigger defects that would not otherwise be screened with the BIST. In order to match the application through memory BIST testing, multiple RAMs can be operating on BIST controller instructions while one RAM is being tested and output data compared. The accessing of multiple memories will have an impact on the memory being testing since the noise disturbance of other memories being addressed can affect the memory being tested. By using the memory BIST controller, the read and write sequence on the memory being tested will also be applied to other memories that can possibly influence the memory under test. Only one memory being tested will return data to the controller for comparison. This practice tests not only the RAMs but also the functional interference among different memories.

Not only is the generation of the memory BIST controller a concern, a more thorough and timely effort is required for the connectivity and controller functionality in respect to the integration of the memory BIST controller into the design.

Depending on the controller configuration, a verification simulation may or may not be capable of running on a conventional simulator. Due to design requirements a custom testbench was created and synthesized to be able to run verification on a hardware emulator. This created testbench was used to verify first and second fail reads, capture failure signatures and also scan out the correct fail signature captured in the diagnostic module. The testbench also was used to verify the restart function when a second fail was detected before the first fail had been completely shifted out. The restart function is necessary due to pipelining inserted in the paths between the controller and memories. The method of verification also ensured that there is no normal functional interaction when the memory BIST is active.

8 Conclusion

To ensure chip quality, today's ASIC designs require flexible tools to quickly create memory BIST RTL controllers, which can detect and diagnose deep sub-micron defects for a variety of embedded RAMs, while operating at functional speeds. The memory BIST tool permits RAM BIST controllers to be quickly created with customized algorithms needed to screen deep sub-micron RAM defects. The BIST allows application of pipeline structures necessary for high-speed applications. The diagnostic module allows defect data to be scanned-out and used for RAM repair and failure analysis. TI was able to successfully use the tool to create three BIST controllers for the TMS320C6414T chip with 400 to 800MHz clock frequencies.

9 Acknowledgement

We would like to thank Peter Ehlig for the identification of memory test algorithms needed to screen defects through the use of CPU memory BIST.

10 References

1. Rochit Rajsuman, "Design and Test of Large Embedded Memories: An overview", *IEEE Design and Test of Computers*, May-June 2001.
2. Ivo Schanstra, Dharmajaya Lukita, Ad J.van de Goor, Kees Veelenturf, Paul J.van Wijnen, "Semiconductor Manufacturing Process Monitoring Using Built-In Self-Test For Embedded Memories", pp. 872-881, *IEEE International Test Conference*, 1998.
3. A.J. van de Goor, "Testing Semiconductor Memories: Theory and Practice", John Wiley & Sons, Ltd., England, 1991.
4. A.J. van de Goor, "Using March tests to test SRAMs," *IEEE Design & Test of Computers*, 3/1993, pp. 8-13.
5. R. Dean Adams, "High Performance Memory Testing: Design Principles, Fault Modeling and Self-Test", Kluwer, Massachusetts, Oct. 2002.
6. Mentor Graphics Corporation, "Built-In Self-Test Process Guide", Software Version 8.2002_2, June 2002.
7. Mentor Graphics Corporation, "MBISTArchitect Reference Manual", Software Version 8.2002_2, June 2002.
8. J.P. Hayes, "Testing memories for single-cell pattern-sensitive faults in semiconductor random-access memories", *IEEE Trans. On Computers*, Vol. C-29, No. 3, 1980. pp. 249-54.