Figure 4-1 Introduction to Memory Testing
Figure 4-2 Memory Types
Figure 4-3 Simple Memory Organization

Memory: Data Width by Address Depth
32 x 512

Memory Array
Address Decode to Row Drivers
Data Decode to Column Drivers
Control Circuitry to Read, Write, and Data Output Enable

Data Bus: To Multiple Memory Arrays
Address Bus: To Multiple Memory Arrays

Data In
Address In
Read/WriteBar
Output Enable

Bus Enable
Data Out
Control Signals: Individual Signals to This Memory Array
Figure 4-4 Memory Design Concerns

- Aspect Ratio
- Access Time
- Power Dissipation
Figure 4-5 Memory Integration Concerns
Figure 4-6 Embedded Memory Test Methods
Figure 4-7 Simple Memory Model
Data in Bit Cells
May Be Stuck-At
Logic 1 or Logic 0

Figure 4-8 Bit-Cell and Array Stuck-At Faults
Data in Bit Cells May Be Bridged to Other Bit Cells

horizontal (row) bit bridging

vertical (column) bit bridging

word bridging unidirectional one-way short

word bridging bidirectional two-way short

random bit bridging

Figure 4-9 Array Bridging Faults
Figure 4-10 Decode Faults

Row Decode
- stuck-at faults result in always choosing wrong address
- bridging faults result in always selecting multiple addresses

Column Decode
- bridging faults result in always selecting multiple data bits
- stuck-at faults result in always choosing wrong data bit

Select Line faults result in similar array fault effects

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Data around target cell is written with complement data.

Complementary Data around Target Cell

Address 21 = A
1 0 1 0

Address 22 = 5
0 1 0 1

Address 23 = A
1 0 1 0

Address 24 = 5
0 1 0 1

alternating 5’s and A’s make for a natural checkerboard pattern

Figure 4-11 Data Retention Faults
Figure 4-12 Memory Bit Mapping

Blue: Pass
Red: Fail
### Figure 4-13 Algorithmic Test Generation

#### Memory Array with 24 Addresses with Algorithm at Read (A) Stage

<table>
<thead>
<tr>
<th>Address</th>
<th>00</th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write(5)-Initialize</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>Increment Address</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>00</th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr(00) to Addr(Max)</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>Read(5)-Write(A)-Read(A)</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Increment Address</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>00</th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr(00) to Addr(Max)</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>Read(A)-Write(5)-Read(5)</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Increment Address</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>00</th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr(Max) to Addr(00)</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>Read(5)-Write(A)-Read(A)</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Decrement Address</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>00</th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr(Max) to Addr(00)</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>Read(A)-Write(5)-Read(5)</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Decrement Address</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>00</th>
<th>01</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
<th>00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addr(Max) to Addr(00)</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>Read(5)</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>Decrement Address</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
</tbody>
</table>

**March C+ Algorithm**

- Read (A)------>
- Write (5)
- Read (5)
- Increment Address

Memory Array with 24 Addresses with Algorithm at Read (A) Stage
Boundary at some level of scanned registration or “pipelining” away from the memory array

Figure 4-14 Scan Boundaries

Minimum Requirement
Detection up to Memory Input and Control of Memory Output

Concern: the Logic between the Scan Test Area and the Memory Test Area Is not Adequately Covered

Non-Scanned Registration inside the Boundary but Before the Memory Test Area Results in a Non-Overlap Zone
Figure 4-15 Memory Modeling

The Memory Array is modeled for the ATPG Engine so the ATPG Tool can use the memory to observe the inputs and control the outputs.
Boundary at some level is blocked off as if the memory was cut out of the circuit.

Scan Mode

Data In

Detection of incoming signals

Address

Control

Gated Data Out

Multiplexed Data Out

Memory Array can be removed from netlist for ATPG purposes

All Registers are in the scan chain architecture

Observe-only registers used for detection of memory input signals

Gate or Multiplexor is used to Block—fix to a known value—the Memory Output Signals

Figure 4-16 Black Box Boundaries
Figure 4-17 Memory Transparency

- Boundary at some level is blocked off as if the memory was cut out of the circuit.
- Input is passed to output as the form of output control.
- Memory array can be removed from netlist for ATPG purposes.
- Observe-only registers used for detection of memory input signals.
- Multiplexer is used to pass the input directly to the output.

Diagram:
- Data In
- Address
- Control
- Bypass Data Out
- Detection of incoming signals
- Memory array
- Scan black-box boundary
Figure 4-18 The Fake Word Technique
Figure 4-19 Memory Test Needs

Memory: data width by address depth
32 x 512

Memory Array
Address Decode to Row Drivers
Data Decode to Column Drivers
Control Circuitry to Read, Write, and Data Output Enable

Data Bus: Possibly to Multiple Memory Arrays
Address Bus: Possibly to Multiple Memory Arrays

Data In
Address
Read/Write
Output Enable

Control Signals: Individual Signals to This Memory Array

Test Must Access the Data, Address, and Control Signals in order to Test This Memory
Figure 4-20 Memory BIST Requirements

INPUTS
Invoke: Start BIST
Retention: Pause BIST and Memory Clocking
Debug: Enable BIST Bitmap Output

OUTPUTS
Fail: A Memory Has Failed a BIST Test
Done: Operation of BIST Is Complete
Debug_data: Debug Data Output

OPERATIONS
Address: Ability to Apply Address Sequences
Data: Ability to Apply Different Data Sequences
Algorithm: Ability to Apply Algorithmic Control Sequences
Comparator: Ability to Verify Memory Data
Figure 4-21 An Example Memory BIST

INPUTS
- **Invoke**: invoke the BIST (apply muxes and release reset)
- **Retention**: enable retention algorithm and pause
- **Release**: discontinue and release pause
- **Bitmap**: enable bitmap output on fail occurrence

OUTPUTS
- **Fail**: sticky fail flag—dynamic under bitmap
- **Done**: operation of BIST is complete
- **Bitmap_out**: fail data under bitmap
- **Hold_out**: indication of pause
Invoke: a global signal to invoke all BIST units
Reset: a global signal to hold all BIST units in reset
Bitmap: a global signal to put all BIST units in debug mode
Hold_/#: individual hold signals to place memories in retention or to select which memory is displayed during debug
done: all memory BISTs have completed
fail: any memory BIST has detected a fault or a failure
diag_out: the memory BIST not in hold mode will present debug data

Figure 4-22 MBIST Integration Issues
Figure 4-23 MBIST Default Values

Invoke: must be a logic 0 when BIST is not enabled
Reset: should be a logic 0 when BIST is not enabled
Bitmap: should be a logic 0 when BIST is not enabled
Hold_: should be a logic 0 when BIST is not enabled

done: should not be connected to package output pin when BIST is not enabled
fail: should not be connected to package output pin when BIST is not enabled
diag_out: should not be connected to package output pin when BIST is not enabled
Invoke: global signal invokes bank 1 BIST
Reset: global signal holds bank 1 BIST in reset
Bitmap: global signal that enables BIST debug
Hold_: paired hold signals to place memories in retention or to select which memory is displayed during debug

done: bank n memory BISTs have completed
fail: any memory BIST has detected a fault or a failure
diag_out: the memory BIST not in hold will present debug data

Figure 4-24 Banked Operation
Figure 4-25 LFSR-Based Memory BIST
The Address sequence can be shifted both forward and backward to provide all addresses.

The Data sequence can be shifted across the data lines, and can also provide data for a comparator.

The Control sequence can be shifted across the read-write or output enable or other control signals.

Figure 4-26 Shift-Based Memory BIST
Figure 4-27 ROM BIST
Memory Testing Fundamentals Summary

Memory Testing Is Defect-Based
Memory Testing Is Algorithmic
Different Types of Memories—Different Algorithms
A Memory Fault Model Is Wrong Data on Read
Memory Testing Relies on Multiple-Clue Analysis
A Memory Test Architecture May CoExist with Scan
A Memory Can Block Scan Test Goals
Modern Embedded Memory Test Is BIST-Based
BIST Is the Moving of the Tester into the Chip
BIST-Based Testing Allows Parallelism
Parallel Testing Impacts Retention Testing
Parallel Testing Impacts Power Requirements
Parallel Testing Requires Chip-Level Integration

Figure 4-28 Memory Test Summary