Memory Testing

- Introduction
- Memory Architecture & Fault Models
- Test Algorithms
- DC / AC / Dynamic Tests
- Built-in Self Testing Schemes
- Built-in Self Repair Schemes
Memory Market Share in 1999

- DRAM: $8 \times 10^{17}$
- Flash: $6 \times 10^{16}$
- ROM: $2 \times 10^{16}$
- SRAM: $9 \times 10^{15}$
DRAM Price per Bit

1991: US$ 400 / Mega bits
1995: US$ 3.75 / Mega bits
1999: US$ 0.1~0.3 / Mega bits
## Test Time as a Function of Memory Size

**Cycle time: 10 ns**

<table>
<thead>
<tr>
<th>Size n</th>
<th>Testing time (in seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64n</td>
</tr>
<tr>
<td>16k</td>
<td>0.01</td>
</tr>
<tr>
<td>64k</td>
<td>0.04</td>
</tr>
<tr>
<td>256k</td>
<td>0.17</td>
</tr>
<tr>
<td>1M</td>
<td>0.67</td>
</tr>
<tr>
<td>4M</td>
<td>2.68</td>
</tr>
<tr>
<td>16M</td>
<td>10.8</td>
</tr>
<tr>
<td>64M</td>
<td>43.2</td>
</tr>
</tbody>
</table>
Architecture of a DRAM Chip

Address

- Address latch
- Column decoder
- Row decoder
- Memory cell array
- Sense amplifiers
- Data register
- Write driver
- Refresh logic

Data

Control Signal

Read/write

Data out

Data in
Fault Models

1. SAF  Stuck-At Fault
2. TF   Transition Fault
3. CF   Coupling Fault
4. NPSF Neighborhood Pattern Sensitive Fault
5. AF   Address decoding fault
Stuck-At Fault

- The logic value of a cell or a line is always 0 or 1.

Transition Fault

- A cell or a line that fails to undergo a $0 \rightarrow 1$ or a $1 \rightarrow 0$ transition.

Coupling Fault

- A write operation to one cell changes the content of a second cell.
Neighborhood Pattern Sensitive Fault

- The content of a cell, or the ability to change its content, is influenced by the contents of some other cells in the memory.

Address Decoder Fault (AF)

- Any fault that affects address decoder:
  - With a certain address, no cell will be accessed.
  - A certain cell is never accessed.
  - With a certain address, multiple cells are accessed simultaneously.
  - A certain cell can be accessed by multiple addresses.
Memory Chip Test Algorithms

- Traditional tests
- Tests for stuck-at, transition and coupling faults
- Tests for neighborhood pattern sensitive faults
### Traditional Tests

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Test length</th>
<th>Test Time Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zero-One</td>
<td>$4n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Checkerboard</td>
<td>$4n$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>GALPAT</td>
<td>$2(n + 2n^2)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Walking 1/0</td>
<td>$2(3n + n^2)$</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Sliding Diagonal</td>
<td>$6n + 2n \cdot \sqrt{n}$</td>
<td>$O(n \cdot \sqrt{n})$</td>
</tr>
<tr>
<td>Butterfly</td>
<td>$2[3n + 5n(n / 2 − 1)]$</td>
<td>$O(n \cdot \log_2 n)$</td>
</tr>
</tbody>
</table>

- $n$ is the number of bits of the memory array.
March Algorithms

Algorithm March X

Step 1: write 0 with up addressing order;
Step 2: read 0 and write 1 with up addressing order;
Step 3: read 1 and write 0 with down addressing order;
Step 4: read 0 with down addressing order.
Notation of March Algorithms

\[\uparrow \downarrow \] : address 0 to address n-1
\[\downarrow \uparrow \] : address n-1 to address 0
\[\uparrow \uparrow \] : either way

w0 : write 0
w1 : write 1
r0 : read a cell whose value should be 0
r1 : read a cell whose value should be 1
March Algorithms

EX:

MATS (modified algorithmic Test Sequence)

\[ \text{(w0); (r0,w1); (r1);} \]

s1: write 0 to all cells
s2: for each cell
    read 0;
    write 1;

s3: read 1 from all cells
Some March Algorithms

MATS : $\uparrow \downarrow (w0); \uparrow \downarrow (r0,w1); \uparrow \downarrow (r1)$
MATS+ : $\uparrow \downarrow (w0); \uparrow \downarrow (r0,w1); \downarrow \downarrow (r1,w0)$
Marching 1/0 : $\uparrow \downarrow (w0); \uparrow \downarrow (r0,w1,r1); \downarrow \downarrow (r1,w0,r0);$ $\downarrow \downarrow (w1); \uparrow \downarrow (r1,w0,r0); \downarrow \downarrow (r0, w1, r1);$ $\downarrow \downarrow (w0); \uparrow \downarrow (r0,w1); \downarrow \downarrow (r1,w0,r0);$ $\downarrow \downarrow (r0)\downarrow \downarrow (r1)\downarrow \downarrow (r0)$
MATS++ : $\uparrow \downarrow (w0); \uparrow \downarrow (r0,w1); \downarrow \downarrow (r1,w0,r0);$ $\downarrow \downarrow (r0, w1, r1);$ $\downarrow \downarrow (w0); \uparrow \downarrow (r0,w1); \downarrow \downarrow (r1,w0); \uparrow \downarrow (r0)$
MARCH X : $\uparrow \downarrow (w0); \uparrow \downarrow (r0,w1); \downarrow \downarrow (r1,w0); \uparrow \downarrow (r0)$
MARCH C : $\uparrow \downarrow (w0); \uparrow \downarrow (r0,w1); \uparrow \downarrow (r1,w0); \uparrow \downarrow (r0);$ $\downarrow \downarrow (r0, w1); \downarrow \downarrow (r1,w0); \uparrow \downarrow (r0);$ $\downarrow \downarrow (r0, w1); \downarrow \downarrow (r1,w0); \uparrow \downarrow (r0);$
Some March Algorithms (Cont.)

MARCH A: $\Downarrow$ (w0); $\Uparrow$ (r0,w1,w0,w1); $\Updownarrow$ (r1,w0,w1);
(r1,w0,w1,w0); $\Downarrow$ (r0,w1,w0);

MARCH Y: $\Uparrow$ (w0); $\Uparrow$ (r0,w1,r1); $\Downarrow$ (r1,w0,r0); $\Updownarrow$ (r0)

MARCH B: $\Uparrow$ (w0); $\Uparrow$ (r0,w1,r1,w0,r0,w1); $\Updownarrow$ (r1,w0,w1);
(r1,w0,w1,w0); $\Downarrow$ (r0,w1,w0)
# Tests for Stuck-At, Transition and Coupling Faults

<table>
<thead>
<tr>
<th>March alg.</th>
<th>Test len.</th>
<th>Fault coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATS</td>
<td>4n</td>
<td>Some AFs, SAFs</td>
</tr>
<tr>
<td>MATS+</td>
<td>5n</td>
<td>AFs, SAFs</td>
</tr>
<tr>
<td>Marching 1/0</td>
<td>14n</td>
<td>AFs, SAFs, TFs</td>
</tr>
<tr>
<td>MATS++</td>
<td>6n</td>
<td>AFs, SAFs, TFs</td>
</tr>
<tr>
<td>March X</td>
<td>6n</td>
<td>AFs, SAFs, TFs, Some CFs</td>
</tr>
<tr>
<td>March C-</td>
<td>10n</td>
<td>AFs, SAFs, TFs, Some CFs</td>
</tr>
<tr>
<td>March A</td>
<td>15n</td>
<td>AFs, SAFs, TFs, Some CFs</td>
</tr>
<tr>
<td>March Y</td>
<td>8n</td>
<td>AFs, SAFs, TFs, Some CFs</td>
</tr>
<tr>
<td>March B</td>
<td>17n</td>
<td>AFs, SAFs, TFs, Some CFs</td>
</tr>
</tbody>
</table>
NPSF

b: base cell
n: neighbor cells

ANPSF:
Active Neighborhood
Pattern Sensitive Fault
n changes
⇒ b changes
Ex:
  n: 0 ⇒ 1
  b: 1 ⇒ 0

PNPSF:
Passive Neighborhood
Pattern Sensitive Fault
Contain n patterns
⇒ b cannot change
Ex:
  n: 00000000
  b: 0 or 1

SNPSF:
Static Neighborhood
Pattern Sensitive Fault
Contain n patterns
⇒ b is forced to a certain value
Ex:
  n: 11111111
  b: 1
DC Parametric Testing

• Contains:
  1. Open / Short test.
  2. Power consumption test.
  3. Leakage test.
  4. Threshold test.
  5. Output drive current test.
AC Parametric Testing

- Output signal: - the rise & fall times.
- Relationship between input signals:
  - the setup & hold times.
- Relationship between input and output signals:
  - the delay & access times.
- Successive relationship between input and output signals:
  - the speed test.
Dynamic Faults

• Recovery faults:
  – Sense amplifier recovery
  – Write recovery.

• Retention faults:
  – Sleeping sickness
  – Refresh line stuck-at
  – Static data loss.

• Bit-line precharge voltage imbalance faults.
BIST: Pros & Cons

- Advantages:
  - Minimal use of testers.
  - Can be used for embedded RAMs.

- Disadvantages:
  - Silicon area overhead.
  - Speed; slow access time.
  - Extra pins or multiplexing pins.
  - Testability of the test hardware itself.
  - A high fault coverage is a challenge.
Typical Memory BIST Architecture Using Mentor’s Architecture
Multiple Memory BIST Architecture

Algorithm-Based Pattern Generator

- sys_addr1
- sys_addr2
- sys_di2
- sys_wen2
- sys_addr3
- sys_di3
- sys_wen3
- rst_
- l_clk
- hold_l
- test_h

ROM4KX4 Module

- addr1
- di2
- addr2
- di3
- addr3
- wen2
- wen3

RAM8KX8 Module

- data
- 4

RAM8KX8 Module

- data
- 8

Compressor

- q
- so

BIST Circuitry
Serial Testing of Embedded RAM

Memory testing.24
Built-in Self-Repair

- BIST can only identify faulty chip.
- Laser cut may be infeasible in some cases, e.g., field testing.
- Two types:
  - Use fault-array comparator
    - Repair by cell
    - Repair by column (or row)
  - Use switch array
BISR Using Switch Array

BISR module

Fault-Address Buffers

RAM Decoder

Switch Array

RAM-Array module

BIST module

Address input

Data input

Data Out
BISR via Fault-Address Comparison

BISR module

Fault-Address Buffers

Fault-Address Compare

RAM Decoder

RAM-Array module

BIST module

Address input

Data input

Data Out