Memory Testing:

Background concepts to prepare the visit to Qimonda (Mindelo)

J. M. Martins Ferreira
FEUP / DEEC
Rua Roberto Frias
4200-465 Porto – PORTUGAL
[ jmf@fe.up.pt ]
Outline of the presentation

• Memory density and defect trends
• Notation
• Faults
• Memory test levels
• March test notation
• Fault modeling
• Memory testing
Semiconductor memories

• The most significant types:
  – DRAM (highest density but slower)
  – SRAM (fast)
  – Cache DRAM (combines SRAM and DRAM)
  – ROM
  – EPROM
  – EEPROM
Memory Testing - (according to Chapter 9 of M. Bushnell and V. Agrawal’s Essentials of Electronic Testing)

Memory density and defect trends

- The number of bits / chip quadruples roughly every 3.1 (π) years

Notes:
- FP: Fast page mode
- EDO: Extended data output
- SDRAM: Synchronous
- DDR: Double data rate
- DRDRAM: Direct rambus

<table>
<thead>
<tr>
<th>Mode</th>
<th>Bandwidth (Mb/s)</th>
<th>Volt.</th>
<th>Chip sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP</td>
<td>25</td>
<td>5 V</td>
<td>256 kb/1 Mb/4 Mb/16 Mb/64 Mb</td>
</tr>
<tr>
<td>EDO</td>
<td>33 – 60</td>
<td>3.3 / 5 V</td>
<td>4 Mb/16 Mb/64 Mb/128 Mb</td>
</tr>
<tr>
<td>SDRAM</td>
<td>66 – 133</td>
<td>3.3 V</td>
<td>16 Mb/64 Mb/128 Mb/256 Mb</td>
</tr>
<tr>
<td>DDR</td>
<td>200 – 300</td>
<td>2.5 V</td>
<td>128 Mb/256 Mb/512 Mb/1 Gb</td>
</tr>
<tr>
<td>DRDRAM</td>
<td>600 – 800</td>
<td>2.5 V</td>
<td>128 Mb/144 Mb/256 Mb/288 Mb/512 Mb/596 Mb/1 Gb</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory size (bits)</th>
<th>Feature width (μm)</th>
<th>Chip area (mm²)</th>
<th>Clock rate (MHz)</th>
<th>Supply voltage (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 Mb</td>
<td>0.35</td>
<td>25</td>
<td>66 – 100</td>
<td>2.5 or 3.0 or 3.3 or 5</td>
</tr>
<tr>
<td>64 Mb</td>
<td>0.30</td>
<td>40</td>
<td>100 – 133</td>
<td>2.5 or 3.3 or 5</td>
</tr>
<tr>
<td>128 Mb</td>
<td>0.23</td>
<td>55</td>
<td>100 – 800</td>
<td>2.5 or 3.3</td>
</tr>
<tr>
<td>256 Mb</td>
<td>0.17</td>
<td>120</td>
<td>100 – 400</td>
<td>2.5 or 3.3</td>
</tr>
<tr>
<td>512 Mb</td>
<td>0.13</td>
<td>200</td>
<td>100 – 400</td>
<td>2.5 or 3.3</td>
</tr>
<tr>
<td>1 Gb</td>
<td>0.11</td>
<td>500†</td>
<td>133**</td>
<td>1.8</td>
</tr>
</tbody>
</table>

* Initial samples available now, production scheduled for late 2002.
** Clock rate is the input clock rate.
† Estimated size.
Memory Testing - (according to Chapter 9 of M. Bushnell and V. Agrawal’s Essentials of Electronic Testing)

Test time

• Test times proportional to $O(n^2)$ or even $O(n \times \log_2(n))$ are now prohibitively expensive, so the memory tests considered are all of complexity $O(n)$

Notes:
- n is the number of memory bits
- Memory cycle time considered: 60 ns

<table>
<thead>
<tr>
<th>Memory Size</th>
<th>Number of Bits</th>
<th>n</th>
<th>nlogn</th>
<th>n^{3/2}</th>
<th>n^2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Mb</td>
<td>0.063</td>
<td>1.26</td>
<td>64.5</td>
<td>18.33 hr</td>
<td>75060 hr</td>
</tr>
<tr>
<td>4 Mb</td>
<td>0.252</td>
<td>5.54</td>
<td>515.4</td>
<td>293.2 hr</td>
<td>120059.9 hr</td>
</tr>
<tr>
<td>16 Mb</td>
<td>1.01</td>
<td>24.16</td>
<td>1.15 hr</td>
<td>4691.3 hr</td>
<td></td>
</tr>
<tr>
<td>64 Mb</td>
<td>4.03</td>
<td>104.7</td>
<td>9.17 hr</td>
<td>75060 hr</td>
<td></td>
</tr>
<tr>
<td>256 Mb</td>
<td>16.11</td>
<td>451.0</td>
<td>73.30 hr</td>
<td>120059.9 hr</td>
<td></td>
</tr>
<tr>
<td>1 Gb</td>
<td>64.43</td>
<td>1932.8</td>
<td>586.41 hr</td>
<td>19215358.4 hr</td>
<td></td>
</tr>
<tr>
<td>2 Gb</td>
<td>128.9</td>
<td>3994.4</td>
<td>1658.61 hr</td>
<td>76861433.7 hr</td>
<td></td>
</tr>
</tbody>
</table>
Defect trends

- Due to exponentially less charge stored per cell and much closer proximity of cells, cell coupling faults are now common.
- Cells are more vulnerable to manufacturing process disturbances.
- The yield of memory chips would be \( \approx 0\% \), since every chip has defects, if not for the inclusion of redundancy (rows, columns).

During initial manufacturing test, the manufacturer develops a map of the faulty rows and columns in the array. The column and row address decoders are then rewired using a laser or by blowing fuses to use spare rows and columns.
Test time complexity

- Manufacturing tests and end-user tests have different requirements (diagnosis is required to repair defective areas)
- Also, initial production characterization tests are different from high-volume, high-yield production tests
- Tests are now based on fault models, but high fault coverage does not necessarily mean high defect coverage

Deep sub-micron chip feature sizes in DRAMs are increasingly subject to peculiar, layout-specific failures. Inductive fault analysis is required to find the best fault models in each case.
Faults

- A system combining electromechanical, chemical, photonic devices on a single chip is a micro electro-mechanical system (MEMS) – e.g. a car air bag controller
- Failures are caused by errors, which are manifestation of faults
- A fault is present if there is a difference between good and incorrect behavior
Fault manifestations

• Permanent faults:
  – Bad electrical connections, broken components, burnt-out chip wire, corroded connection between chip and package, chip logic error

• Non-permanent faults occur randomly and are handled by including information redundancy in each memory location (e.g. redundant error correcting codes)
Fault manifestations

> Non-permanent faults

* Transient faults:
  - Cosmic rays, α-particles, air pollution, humidity, temperature, pressure, vibrations, power supply fluctuations, electromagnetic interference, static discharges, ground loops

* Intermittent faults:
  - Loose connections, hazards and races, resistor, capacitor and inductor variations, physical irregularities, electrical noise
Failure mechanisms

- Corrosion
- Electromigration
- Bonding deterioration
- Ionic contamination
- Alloying
- Radiation and cosmic rays
- The *activation energy* describes the variation with the temperature
Memory Testing - (according to Chapter 9 of M. Bushnell and V. Agrawal’s Essentials of Electronic Testing)

Memory test levels

- Chip, array and board have different test requirements
- Electrical parametric tests are also important for memory systems
March test notation

- Memory operations:
  - r0, r1, w0, w1 (read and write 0 / 1)
  - ↑, ↓, ↔ (0 to 1, 1 to 0, complement)
  - ↑, →, ⊥ (increase address, decrease address, either up or down)
  - →, →, ⊢ (0 to 0, 1 to 1, X to X)
  - ∀ (any memory operation: ↑, ↓, ↔, →, →, ⊢)
  - <I₁,..,Iₙ₋₁; Iₙ/F>: Fault (F) sensitization conditions (I) in cells 1 to n

Denotes a fault involving n cells; if Iₙ is empty, then Iₙ / F is written as F
March tests

- March tests consist of a finite operation sequence, applied to each cell before proceeding to the next cell (if a specific pattern is applied to one cell, then it must be applied to all cells)
- The MATS+ march test is written as \{\uparrow(w0); \uparrow(r0,w1); \downarrow(r1,w0)\} and has three march elements M0: \uparrow(w0), M1: \uparrow(r0,w1), and M2: \downarrow(r1,w0)
March tests

> The MATS+ march test

- \{ \uparrow (w0); \uparrow (r0,w1); \downarrow (r1,w0) \}

- March tests are a preferred method for RAM array testing, externally or via BIST, due to their O(n) complexity, regularity, and symmetry

M0:  
{ March element \( \uparrow (w0) \) }
for cell := 0 to \( n - 1 \) (or any other order) do
begin
  write 0 to A [ cell ];
end;

M1:  
{ March element \( \uparrow (r0,w1) \) }
for cell := 0 to \( n - 1 \) do
begin
  read A [ cell ]; { Expected value = 0 }
  write 1 to A [ cell ];
end;

M2:  
{ March element \( \downarrow (r1,w0) \) }
for cell := \( n - 1 \) down to 0 do
begin
  read A [ cell ]; { Expected value = 1 }
  write 0 to A [ cell ];
end;

MATS+ march test algorithm.
Fault modeling

- *Physical* faults have to be modeled as *logical* faults
- Modeling physical faults as logical faults makes the testing approach more independent of the technology and the manufacturing process
- However, it may not be possible to relate a failure to the actual physical defect
Memory Testing - (according to Chapter 9 of M. Bushnell and V. Agrawal’s Essentials of Electronic Testing)

Functional memory models

Van de Goor’s reduced functional DRAM chip model

Reduced functional memory model
Functional memory faults

- Examples of functional faults that may occur in a memory device:

<table>
<thead>
<tr>
<th>Functional fault</th>
<th>Functional fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>a Cell stuck</td>
<td>j Open circuit in address line</td>
</tr>
<tr>
<td>b Driver stuck</td>
<td>k Shorts between address lines</td>
</tr>
<tr>
<td>c Read/write line stuck</td>
<td>l Open circuit in decoder</td>
</tr>
<tr>
<td>d Chip-select line stuck</td>
<td>m Wrong address access</td>
</tr>
<tr>
<td>e Data line stuck</td>
<td>n Multiple simultaneous address access</td>
</tr>
<tr>
<td>f Open circuit in data line</td>
<td>o Cell can be set to 0 but not to 1 (or vice versa)</td>
</tr>
<tr>
<td>g Short circuit between data lines</td>
<td>p Pattern sensitive cell interaction</td>
</tr>
<tr>
<td>h Crosstalk between data lines</td>
<td></td>
</tr>
<tr>
<td>i Address line stuck</td>
<td></td>
</tr>
</tbody>
</table>

Notes:

- **Cell** refers either to a memory cell or to a data register
- **Line** refers to any wiring connection in the memory
Reduced functional faults

• When fault diagnosis is not important, the previous faults may be mapped into a reduced set of functional faults (which are sufficient for functional memory testing):
  – SAF: Stuck-at faults
  – TF: Transition faults
  – CF: Coupling faults
  – NPSF: Neighborhood pattern sensitive faults
Reduced functional faults

> Stuck-at faults (SAF)

• A SAF is one in which the logic value of a cell is always 0 (SA0: $\forall/0$) or 1 (SA1: $\forall/1$)
Reduced functional faults

> Transition faults (TF)

• TFs are a special case of SAFs, in which a cell fails to make a 0 to 1 ($\leq\uparrow/0\gtrapprox$) or a 1 to 0 ($\leq\downarrow/1\gtrapprox$) transition when it is written.

• A coupling fault with another cell can cause this faulty cell to revert to the 1/0 state, so the SAF cannot model a TF.
Reduced functional faults

> Coupling faults (CF) [ General ]

- A transition in memory bit j causes an unwanted change in memory bit i
- The 2-CF involves two cells: a write in cell j changes the contents of cell i
- The 2-CF is a special case of the K-CF (the K cells must be restricted, as is done in the neighborhood pattern sensitive fault, to make the K-CF model practical)
Reduced functional faults

> Inversion coupling faults (CFin)

- A ↑ or ↓ transition in cell j inverts the contents of cell i (e.g. <↑;↓>)
- Cell i is said to be coupled to cell j, which is the coupling cell
Reduction functional faults

> Idempotent coupling faults (CFid)

- A ↑ or ↓ transition in cell j sets cell i to 0 or 1 (there are 4 CFids, e.g. <↑;0>
Reduced functional faults

Dynamic coupling faults (CFdyn)

- A CFdyn occurs between cells in different words: a read or write operation forces the contents of the second cell to either 0 or 1 (there are 4 CFdyns: <r0|w0;0>, <r0|w0;1>, <r1|w1;0>, <r1|w1;1>)

- CFdyns are more general than CFids, because a CFdyn can be sensitized by any read or write operation, where as a CFid can only be sensitized by writing to the coupling cell
Reduced functional faults

> Bridging faults (BF)

- A BF is a short circuit between two or more cells or lines
- It is a bidirectional fault, so either cell / line can affect the other cell / line (a 0 or 1 state of the coupling cell causes the fault, rather than a coupling cell transition)
  - 4 AND BFs: \(<0,0|0,0>, <0,1|0,0>, <1,0|0,0>, <1,1|1,1>\)
  - 4 OR BFs: \(<0,0|0,0>, <0,1|1,1>, <1,0|1,1>, <1,1|1,1>\)
Reduced functional faults

- State coupling faults (SCF)

- The SCF is where the coupling cell / line j is in a given state y that forces the coupled cell i into state x
- There are 4 SCFs: <0;0>, <0;1>, <1;0>, and <1;1>

If SCF <1;1> is present...
Reduced functional faults

> Pattern sensitive coupling faults (PSF)

- The content of cell i (its ability to change) is influenced by the contents of the other memory cells (either a pattern of 0s and 1s or a pattern of transitions).
- The *neighborhood* is the total number of cells involved in this fault, where the base cell is the cell under test and the deleted neighborhood is the neighborhood without the base cell.
Reduced functional faults

> Neighborhood pattern sensitive (NPSF)

- In the PSF, the neighborhood could be anywhere in the memory array, whereas in a neighborhood pattern sensitive fault (NPSF), the neighborhood must be in a single position surrounding the base cell.

- All known algorithms are for NPSFs, which must now be tested because of the reduced memory cell capacitance in high-density DRAMs.
Reduced functional faults

> Neighborhood pattern sensitive: ANPSF

- In the *active NPSF* (ANPSF), the base cell changes due to a change in the pattern of the deleted neighborhood.
- A type-1 neighborhood has 4 deleted neighborhood cells.
- \( C_{i,j} <0, ↓, 1, 1; 1> \) is an ANPSF where the base cell \( C_{i,j} \) is initially 0, \( d_0, d_3 \), and \( d_4 \) contain 011, \( d_1 \) experiences a \( ↓ \) transition, and the fault effect is to switch the base cell to 1 (\( C_{i,j} \) is the base cell location).
Reduced functional faults

> Neighborhood pattern sensitive: PNPSF

- A *passive NPSF* (PNPSF) means that a certain neighborhood pattern prevents the base cell from changing.
- As an example, the fault preventing the base cell $C_{i,j}$ from changing from 0 is denoted as $C_{i,j}<0,0,1,1; \uparrow/0>$.
- If it cannot change regardless of content, we write this as $C_{i,j}<0,0,1,1; \uparrow/X>$.
Reduced functional faults

> Neighborhood pattern sensitive: SNPSF

- With a *static NPSF* (SNPSF), the base cell is forced into a particular state when the deleted neighborhood contains a particular pattern, e.g. \( C_{i,j} <0,1,0,1; –/0> \)

- This differs from the ANPSF in a subtle way, because there need not be a transition in the deleted neighborhood to sensitize an SNPSF
Reduced functional faults

> Address decoder faults (AF)

- AFs represent an address decoding error
- Van de Goor classifies AFs into 4 classes:
  - No cell is accessed for a certain address
  - No address can access a certain cell
  - One address accesses multiple cells
  - One cell is accessed with multiple addresses
- We also assume that the fault is the same during both read and write operations
Fault models x physical defects

- In high-density DRAM chips all the defects listed are quite likely, and so correct DRAM testing requires testing for ALL of the reduced functional faults.

[Table: Mapping of functional faults onto reduced functional faults]

<table>
<thead>
<tr>
<th>Reduced functional fault</th>
<th>Functional fault</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAF</td>
<td>a Cell stuck</td>
</tr>
<tr>
<td>SAF</td>
<td>b Driver stuck</td>
</tr>
<tr>
<td>SAF</td>
<td>c Read/write line stuck</td>
</tr>
<tr>
<td>SAF</td>
<td>d Chip-select line stuck</td>
</tr>
<tr>
<td>SAF</td>
<td>e Data line stuck</td>
</tr>
<tr>
<td>SAF</td>
<td>f Open circuit in data line</td>
</tr>
<tr>
<td>CF</td>
<td>g Short circuit between data lines</td>
</tr>
<tr>
<td>CF</td>
<td>h Crosstalk between data lines</td>
</tr>
<tr>
<td>AF</td>
<td>i Address line stuck</td>
</tr>
<tr>
<td>AF</td>
<td>j Open circuit in address line</td>
</tr>
<tr>
<td>AF</td>
<td>k Shorts between address lines</td>
</tr>
<tr>
<td>AF</td>
<td>l Open circuit in decoder</td>
</tr>
<tr>
<td>AF</td>
<td>m Wrong address access</td>
</tr>
<tr>
<td>AF</td>
<td>n Multiple simultaneous address access</td>
</tr>
<tr>
<td>TF</td>
<td>o Cell can be set to 0 but not to 1 (or vice versa)</td>
</tr>
<tr>
<td>NPSF</td>
<td>p Pattern sensitive cell interaction</td>
</tr>
</tbody>
</table>
Multiple fault models

• When coupling faults are used, we also assume that any number of the various different faults can occur *simultaneously*.

• Faults may also be *linked*, meaning that a fault may influence the behavior of other faults.

![Diagram showing non-maskable and maskable faults with linkage](image-url)
Detection of linked faults

- Linked fault: (2,3) <\textgreater\textless 1/1> coupled to (1,1) and (2,3) <\textgreater\textless 0/0> coupled to (2,2)
- Will \{\textgreater\textless (w0); \textgreater\textless (r0,w1); \textgreater\textless (w0,w1); \textgreater\textless (r1,w0)\} detect this fault?

Any changes if (2,2) is replaced by (3,2)?
SRAM and DRAM faults

- The possible faults differ between these two types of memory

<table>
<thead>
<tr>
<th>DRAM or SRAM faults</th>
<th>Fault model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shorts and opens in the memory cell array</td>
<td>SAF, SCF</td>
</tr>
<tr>
<td>Shorts and opens in the address decoder</td>
<td>AF</td>
</tr>
<tr>
<td>Access time failures in the address decoder</td>
<td>Functional Fault</td>
</tr>
<tr>
<td>Stray coupling capacitances between adjacent cells</td>
<td>CF</td>
</tr>
<tr>
<td>Bit line shorted to word line (causes excessive $I_{DDQ}$ current) [456, 455, 596]</td>
<td></td>
</tr>
<tr>
<td>Transistor gate shorted to channel (causes excessive $I_{DDQ}$ current) [456, 455, 596]</td>
<td></td>
</tr>
<tr>
<td>Transistor stuck-open fault (causes excessive $I_{DDQ}$ current) [456, 455, 596]</td>
<td>SOF</td>
</tr>
<tr>
<td>Pattern sensitive faults (causes excessive $I_{DD}$ power supply current) [643]:</td>
<td>PSF</td>
</tr>
<tr>
<td>Diode-connected transistor short between 2 cells</td>
<td></td>
</tr>
<tr>
<td>Open transistor drain</td>
<td></td>
</tr>
<tr>
<td>Gate oxide short</td>
<td></td>
</tr>
<tr>
<td>Bridging Fault</td>
<td></td>
</tr>
</tbody>
</table>

Faults found only in SRAM

<table>
<thead>
<tr>
<th>Fault model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open-circuited pull-up device (causes data loss) [442]</td>
</tr>
<tr>
<td>Excessive coupling capacitance between bit lines [357]</td>
</tr>
</tbody>
</table>

Faults found only in DRAM

<table>
<thead>
<tr>
<th>Fault model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data retention fault (<em>sleeping sickness</em>)</td>
</tr>
<tr>
<td>Refresh line stuck-at fault</td>
</tr>
<tr>
<td>Bit-line voltage imbalance fault [442] due to weak inversion currents</td>
</tr>
<tr>
<td>Coupling between Word and Bit line</td>
</tr>
<tr>
<td>Single-ended bit-line voltage shift</td>
</tr>
<tr>
<td>Precharge and decoder clock overlap</td>
</tr>
</tbody>
</table>
Functional RAM testing with march tests

- The table below summarizes which faults and linkages are covered by all of the march tests

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Fault coverage</th>
<th>Operation count</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATS</td>
<td>All Some</td>
<td>4 · n</td>
</tr>
<tr>
<td>MATS+</td>
<td>All All</td>
<td>5 · n</td>
</tr>
<tr>
<td>MATS ++</td>
<td>All All All</td>
<td>6 · n</td>
</tr>
<tr>
<td>MARCH X</td>
<td>All All All All</td>
<td>6 · n</td>
</tr>
<tr>
<td>MARCH C−</td>
<td>All All All All</td>
<td>10 · n</td>
</tr>
<tr>
<td>MARCH A</td>
<td>All All All All</td>
<td>All linked CFids, Some CFins linked with CFids</td>
</tr>
<tr>
<td>MARCH Y</td>
<td>All All All All</td>
<td>All TFs linked with CFins</td>
</tr>
<tr>
<td>MARCH B</td>
<td>All All All All</td>
<td>All linked CFids, All TFs linked with CFids or CFins, Some CFins linked with CFids</td>
</tr>
</tbody>
</table>
Definition of march tests

- The table below defines the march tests that were referred in the previous slide.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MATS</td>
<td>{ \uparrow (w0); \downarrow (r0, w1); \uparrow (r1) }</td>
</tr>
<tr>
<td>MATS+</td>
<td>{ \uparrow (w0); \downarrow (r0, w1); \downarrow (r1, w0) }</td>
</tr>
<tr>
<td>MATS++</td>
<td>{ \downarrow (w0); \uparrow (r0, w1); \downarrow (r1, w0, r0) }</td>
</tr>
<tr>
<td>MARCH X</td>
<td>{ \uparrow (w0); \downarrow (r0, w1); \downarrow (r1, w0); \uparrow (r0) }</td>
</tr>
<tr>
<td>MARCH C−</td>
<td>{ \downarrow (w0); \uparrow (r0, w1); \uparrow (r1, w0); \downarrow (r0, w1); \downarrow (r1, w0); \uparrow (r0) }</td>
</tr>
<tr>
<td>MARCH A</td>
<td>{ \downarrow (w0); \uparrow (r0, w1, w0, w1); \uparrow (r1, w0, w1); \downarrow (r0, w1, w0) }</td>
</tr>
<tr>
<td>MARCH Y</td>
<td>{ \downarrow (w0); \uparrow (r0, w1, r1); \downarrow (r1, w0, r0); \uparrow (r0) }</td>
</tr>
<tr>
<td>MARCH B</td>
<td>{ \downarrow (w0); \uparrow (r0, w1, r1, w0, r0, w1); \uparrow (r1, w0, w1, w0); \downarrow (r0, w1, w0) }</td>
</tr>
</tbody>
</table>
Example: detection of a SA0 fault with the MATS+ march test

- MATS+: \{\uparrow(w_0); \uparrow(r_0,w_1); \downarrow(r_1,w_0)\}

- Behavior of MATS+ when cell (2,1) has a SA0 fault:
  - The fault is detected by march element M2 as it moves from the highest memory address downward and expects to read a 1 in cell (2,1), but instead reads a 0.
Example: detection of a SA1 fault with the MATS+ march test

- MATS+: \{\uparrow (w0); \uparrow (r0, w1); \downarrow (r1, w0)\}

- Behavior of MATS+ when cell (2,1) has a SA1 fault:
  - The fault is detected by march element M1 as it moves from the lowest memory address upward and expects to read a 0 in cell (2,1), but instead reads a 1.
Example: detection of a multiple AF with the MATS+ march test

- MATS+: \{\uparrow(w0); \uparrow(r0,w1); \downarrow(r1,w0)\}

- Behavior of MATS+ when cell (2,1) is unaddressable and address (2,1) maps instead to cell (3,1):
  - The AF will be detected either by march element M1 or by march element M2.
Testing RAM ANPSFs / PNPSFs

- ANPSF: the base cell must be read in state 0 and state 1, for all possible transitions in the deleted neighborhood pattern.

- PNPSF: the two transitions (↑,↓) must be verified for the $2^{k-1}$ deleted neighborhood patterns.
Testing RAM technology and layout-related faults

- DRAMs may be repaired or may have their address lines deliberately scrambled, so consecutive addresses may not be adjacent (and the previously described coupling fault tests will not be effective)
- Also, the Gigabit DRAMs have new kinds of defects which march or NPSF tests may not cover
Testing RAM technology and layout-related faults: IFA

- **Inductive fault analysis** (IFA) models a single defect per memory cell:
  - Generate defect sizes, locations, and layers based on the real fabrication line
  - Place the defects on a model of the layout
  - Extract the schematic and electrical parameters for the defective cell
  - Evaluate the results of testing the defective cell
Testing RAM technology and layout-related faults: IFA

- Actual defects modeled as broken / shorted wires, missing contacts, etc., map as:
  - A SAF in a memory cell
  - A stuck-open fault (SOF) in a memory cell
  - A TF in a memory cell
  - A SCF between two memory cells
  - A CFid between two cells
  - A data retention fault (DRF) in which the cell looses its contents over time
Testing RAM technology and layout-related faults: IFA

- A new march element, Delay (means wait for 100 ms before continuing), enables the detection of these layout-related faults

### IFA augmented march test summary

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Actual physical defect fault coverage</th>
<th>Operation count</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFA-9</td>
<td>SAF: All, TF: All, AF: All, SOF: All, SCF: All, CFid: All</td>
<td>$12 \cdot n + \text{Delays}$</td>
</tr>
<tr>
<td>IFA-13</td>
<td>SAF: All, TF: All, AF: All, SOF: All, SCF: All, CFid: All</td>
<td>$16 \cdot n + \text{Delays}$</td>
</tr>
<tr>
<td>MARCH G</td>
<td>SAF: All, TF: All, AF: All, SOF: All, SCF: All, CFid: All</td>
<td>$23 \cdot n + \text{Delays}$</td>
</tr>
</tbody>
</table>

- All linked CFids
- All TFs linked with CFids or CFins
- Some CFins linked with CFids

### IFA augmented march test algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFA-9</td>
<td>${ \uparrow (w0); \uparrow (r0, w1); \uparrow (r1, w0); \downarrow (r0, w1); \downarrow (r1, w0); \text{Delay}; \uparrow (r0, w1); \text{Delay}; \uparrow (r1) }$</td>
</tr>
<tr>
<td>IFA-13</td>
<td>${ \uparrow (w0); \uparrow (r0, w1, r1); \uparrow (r1, w0, r0); \downarrow (r0, w1, r1); \downarrow (r1, w0, r0); \text{Delay}; \uparrow (r0, w1); \text{Delay}; \uparrow (r1) }$</td>
</tr>
<tr>
<td>MARCH G</td>
<td>${ \uparrow (w0); \uparrow (r0, w1, r1, w0, r0, w1); \uparrow (r1, w0, w1); \downarrow (r1, w0, w1, w0); \downarrow (r0, w1, w0); \text{Delay}; \downarrow (r0, w1, r1); \text{Delay}; \downarrow (r1, w0, w0) }$</td>
</tr>
</tbody>
</table>
Electrical parameter testing

- DC parametric tests
  - Voltage bump test, leakage test
- AC parametric tests
  - Address set-up time sensitivity, access time, running time, sense amplifier recovery fault, write recovery fault
- Dual-port SRAM tests
  - Standby current, circuit-dependent, arbitration
Summary

- No single type of test (march, NPSF, DC parametric, AC parametric) is sufficient for current RAM testing needs, so a combination of various tests is used.
- Inductive fault analysis is now necessary, to ensure that the actual defects that are occurring are mapped into a fault model, and then appropriate tests can be selected for that fault model.
Memory Testing - (according to Chapter 9 of M. Bushnell and V. Agrawal’s Essentials of Electronic Testing)

Essentials of Electronic Testing
for Digital, Memory & Mixed-Signal VLSI Circuits