Flash Memory Built-In Self-Test Using March-Like Algorithms

Jen-Chieh Yeh, Chi-Feng Wu, Kuo-Liang Cheng, Yung-Fa Chou, Chih-Tsun Huang, and Cheng-Wen Wu

Design Technology Center
National Tsing Hua University
Outline

- Flash Memory Testing Issues
- Target Fault Models
- Flash Memory Test Algorithms
- Built-In Self-Test (BIST)
- Experimental Results
- Conclusions
Flash Memory Test Issues

- Reliability issues
  - **Disturbances**: inadvertent change of the cell content due to reading or programming another cell
  - **Over-erasing**: overstressed cell after erase, leading to unreliable program operation
  - **Endurance**: capability of maintaining the stored information within specified operation count
  - **Retention**: capability of maintaining the stored information within specified time limit

- Long program/erase time
- Test access for embedded flash memory
- ATE price is high, and grows rapidly
IEEE Standard 1005, “Definitions and Characterization of Floating Gate Semiconductor Arrays”, defines the disturbance conditions

- Word-line Program Disturbance (WPD)
- Word-line Erase Disturbance (WED)
- Bit-line Program Disturbance (BPD)
- Bit-line Erase Disturbance (BED)
- Over Erase (OE)
- Read Disturbance (RD)
Several conventional RAM fault models are also considered useful for testing flash memory:

- **Stuck-At Fault (SAF)**
  - Cell or line sticks at 0 or 1
- **Transition Fault (TF)**
  - Cell fails to transit from 0 to 1 or 1 to 0
- **Stuck-Open Fault (SOF)**
  - Cell not accessible due to broken line
- **State Coupling Fault (CFst)**
  - Coupled cell is forced to 0 or 1 if coupling cell is in given state
- **Address-Decoder Fault (AF)**
  - A functional fault in the address decoder
Bit-Oriented Test Algorithm

- Conventional March tests can not detect all flash specific faults
- No (w1) operation in flash technology
- Proposed March Flash-Test (March FT)
  - \{(f); \uparrow(r1,w0,r0); \uparrow(r0); (f); \downarrow(r1,w0,r0); \uparrow(r0)\}
  - Regular, easier to generate, covering more functional faults and do not rely on the array geometry or layout topology

<table>
<thead>
<tr>
<th>Notation</th>
<th>Operations</th>
<th>Notation</th>
<th>Address Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>Erase/Flash</td>
<td>\uparrow</td>
<td>Ascending</td>
</tr>
<tr>
<td>w0</td>
<td>Program</td>
<td>\downarrow</td>
<td>Descending</td>
</tr>
<tr>
<td>r1 or r0</td>
<td>Read 1 or 0</td>
<td>\bigcirc</td>
<td>Ascending or Descending</td>
</tr>
</tbody>
</table>
Word-Oriented Test Algorithm

- Word-oriented memory may have intra-word faults
- Add simple test with multiple standard backgrounds to cover intra-word faults
  - \{ (f); \oplus (wa, ra); (f); \oplus (wb, rb) \}
- Number of backgrounds is \( \log_2(m) + 1 \)
  - m : word width
- Example (m = 4):
  - **0000**  (f); \( \uparrow (rb, wa, ra); \oplus (ra); (f); \downarrow (rb, wa, ra); \oplus (ra) \)
  - **0011**  (f); \( \oplus (wa, ra); (f); \oplus (wb, rb) \)
  - **0101**  (f); \( \oplus (wa, ra); (f); \oplus (wb, rb) \)

“0000” is solid background  “0011” & “0101” are standard backgrounds
## Fault Simulator

### RAMSES-FT

<table>
<thead>
<tr>
<th>SpecFile</th>
<th>ShowOptions</th>
<th>EditPattern</th>
<th>PatternGenerate</th>
<th>Help</th>
</tr>
</thead>
</table>

### Flash Memory Simulator

**Flash type:** NOR

**Gate type:** Stack

**Address:** 65536

**Row:** 256

**Col:** 256

**W1:** 1

**Erase time:** 3s

**Program time:** 9us

**Read time:** 70ns

**Pattern file:** pattern.m

### Simulation Result

This Flash memory is NOR type (STACK gate).

- Memory size (N): 65536
- Test length: 2(erase time) + 131072(program time) + 393216(read time)
- Test length time: 7.207173 sec

**Fault Coverage:**

- **Conventional**
  - SAF:
    - SA1 = 1.0 (65536/65536)
    - SA0 = 1.0 (65536/65536)
  - TF:
    - TFU = 1.0 (65536/65536)
    - TFD = 1.0 (65536/65536)
  - CFst:
    - CFst00 = 1.0 (4294901760/4294901760)
    - CFst01 = 1.0 (4294901760/4294901760)
    - CFst10 = 1.0 (4294901760/4294901760)
    - CFst11 = 1.0 (4294901760/4294901760)
  - SOF:
    - SOF = 1.0 (65536/65536)
  - AF:
    - AF = 1.0 (4294901760/4294901760)

- **Disturb**
  - Program:
    - GPD = 1.0 (16711680/16711680)
    - GED = 1.0 (16711680/16711680)
    - DED = 1.0 (16711680/16711680)
    - DPD = 1.0 (16711680/16711680)
  - Read:
    - RD = 1.0 (65536/65536)
  - Erase:
    - OE = 1.0 (65536/65536)

**Status:** Run simulation result.

flash2.03/cww
# Simulation Results

- **Bit-oriented memory simulation result (128Kb flash memory)**

<table>
<thead>
<tr>
<th></th>
<th>GPD 100%</th>
<th>GED 100%</th>
<th>DPD 100%</th>
<th>DED 100%</th>
<th>OE  100%</th>
<th>RD  0%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash March [VTS2001]</td>
<td>SAF 100%</td>
<td>TF 100%</td>
<td>SOF 50%</td>
<td>AF 100%</td>
<td>CFst 75%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test Complexity</td>
<td>2F + 2NP + 4NR</td>
<td>Test Time</td>
<td>2.503 sec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>March FT (proposed)</td>
<td>GPD 100%</td>
<td>GED 100%</td>
<td>DPD 100%</td>
<td>DED 100%</td>
<td>OE  100%</td>
<td>RD  100%</td>
</tr>
<tr>
<td></td>
<td>SAF 100%</td>
<td>TF 100%</td>
<td>SOF 100%</td>
<td>AF 100%</td>
<td>CFst 100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Test Complexity</td>
<td>2F + 2NP + 6NR</td>
<td>Test Time</td>
<td>2.516 sec</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assume F=190ms, P=8us, R=50ns, and N=128K
Simulation Results

Word-oriented memory simulation result (128Kx4 flash memory, 4-bit words)

<table>
<thead>
<tr>
<th>March FT (Only solid background)</th>
<th>GPD 100%</th>
<th>GED 100%</th>
<th>DPD 100%</th>
<th>DED 100%</th>
<th>OE 100%</th>
<th>RD 100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAF 100%</td>
<td>TF 100%</td>
<td>SOF 100%</td>
<td>AF 95.2%</td>
<td>CFst 97.6%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Complexity</td>
<td>2F + 2NP + 6NR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Time</td>
<td>2.516 sec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>March FT (With standard backgrounds)</th>
<th>GPD 100%</th>
<th>GED 100%</th>
<th>DPD 100%</th>
<th>DED 100%</th>
<th>OE 100%</th>
<th>RD 100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAF 100%</td>
<td>TF 100%</td>
<td>SOF 100%</td>
<td>AF 100%</td>
<td>CFst 100%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Complexity</td>
<td>6F + 6NP + 10NR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Test Time</td>
<td>7.497 sec</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Assume F=190ms, P=8us, R=50ns, and N=128K
BIST Case I

A 4Mb (512K x 8) embedded flash memory
A commodity 1Mb (128K x 8) flash memory
## Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Embedded Flash</th>
<th>Commodity Flash Chip</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Memory Size</strong></td>
<td>512K bytes</td>
<td>128K bytes</td>
</tr>
<tr>
<td><strong>Mass Erase Time</strong></td>
<td>200ms</td>
<td>190ms</td>
</tr>
<tr>
<td><strong>Byte Program Time</strong></td>
<td>20us</td>
<td>8us</td>
</tr>
<tr>
<td><strong>Erase Penalty</strong></td>
<td>2.5ms</td>
<td>1us</td>
</tr>
<tr>
<td><strong>Program Penalty</strong></td>
<td>21us</td>
<td>1us</td>
</tr>
<tr>
<td><strong>Scrambling Type</strong></td>
<td>Data</td>
<td>Address</td>
</tr>
<tr>
<td><strong>Built-In Test Algorithm</strong></td>
<td>March FT (Only solid background)</td>
<td>March FT (With standard background)</td>
</tr>
<tr>
<td><strong>Hardware Overhead</strong></td>
<td>3.2%</td>
<td>2.28%</td>
</tr>
<tr>
<td><strong>Testing Time</strong></td>
<td>44.612 sec</td>
<td>13 sec</td>
</tr>
</tbody>
</table>
Conclusions

- Bit-oriented and word-oriented flash memory test algorithms proposed
- Flash memory BIST circuit developed and implemented
- Flash memory fault simulator also developed
- Future work
  - Diagnostics and built-in self-repair