The Inconvenient Truths of NAND Flash Memory

Jim Cooke (jcooke@micron.com)
Applications Engineering Director
Micron Technology, Inc.
Agenda

• NAND Flash Differences
  – MLC vs. SLC
  – Architecture, Features, and Performance Comparisons

• NAND Error Modes
  – Program Disturb
  – Read Disturb
  – Data Retention
  – Endurance
  – Wear Leveling
  – ECC Fixes Almost Everything
All NAND Flash Devices Are Not Created Equal

- Differences include:
  - Cell types
  - Architectural
  - Performance
  - Timing parameters
  - Command set

- Open NAND Flash Interface (ONFI) helps to address many of these
### Cell Types – MLC vs. SLC

<table>
<thead>
<tr>
<th>Features</th>
<th>MLC</th>
<th>SLC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bits per cell</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Voltage</td>
<td>3.3V</td>
<td>3.3V, 1.8V</td>
</tr>
<tr>
<td>Data width (bits)</td>
<td>x8</td>
<td>x8, x16</td>
</tr>
<tr>
<td>Architecture</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of planes</td>
<td>2</td>
<td>1 or 2</td>
</tr>
<tr>
<td>Page size</td>
<td>2,112–4,314 bytes</td>
<td>2,112 bytes</td>
</tr>
<tr>
<td>Pages per block</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td>Reliability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOP (partial page programming)</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>ECC (per 512 bytes)</td>
<td>4+</td>
<td>1</td>
</tr>
<tr>
<td>Endurance (ERASE / PROGRAM cycles)</td>
<td>&lt;10K</td>
<td>&lt;100K</td>
</tr>
<tr>
<td>Array Operations</td>
<td></td>
<td></td>
</tr>
<tr>
<td>tR (Max)</td>
<td>50μs</td>
<td>25μs</td>
</tr>
<tr>
<td>tPROG (Typ)</td>
<td>600–900μs</td>
<td>200–300μs</td>
</tr>
<tr>
<td>tBERS (Typ)</td>
<td>2ms</td>
<td>1.5–2ms</td>
</tr>
</tbody>
</table>
2Gb, 2K-Page SLC NAND Architecture

- 2,112 bytes
- I/O 0
- I/O 7

Cache Register
- 2,048
- 64

Data Register
- 2,048
- 64

64 pages = 1 block (128K + 4K) bytes
1 page = (2K + 64) bytes
1 block = (2K + 64) bytes x 64 pages = (128K + 4K) bytes
1 device = (2K + 64) bytes x 64 pages x 2,048 blocks = 2,112Mb
2Gb, 2K-Page, 72nm, SLC Performance

Micron (72nm SLC) 2Gb die 2K Page Performance

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Time</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>tR</td>
<td>25</td>
<td>us</td>
</tr>
<tr>
<td>tDCBSYR1</td>
<td>3</td>
<td>us</td>
</tr>
<tr>
<td>tDCBSYR2</td>
<td>3</td>
<td>us</td>
</tr>
<tr>
<td>tRC</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>tRC (C)</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>tRC</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>tPROG</td>
<td>240</td>
<td>us</td>
</tr>
<tr>
<td>tCBSY</td>
<td>300</td>
<td>us</td>
</tr>
<tr>
<td>tDBSY</td>
<td>0.5</td>
<td>us</td>
</tr>
<tr>
<td>tWC</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>tWC (C)</td>
<td>30</td>
<td>ns</td>
</tr>
<tr>
<td>tWC</td>
<td>45</td>
<td>ns</td>
</tr>
<tr>
<td>PS</td>
<td>2112</td>
<td>Byte</td>
</tr>
<tr>
<td>NP</td>
<td>64</td>
<td>Pages</td>
</tr>
</tbody>
</table>

NAND Operation

Page Read: 23.85 MB/sec
Cache Read: 31.62 MB/sec
2Plane Page Read: 0.00 MB/sec
Page Program: 5.81 MB/sec
Program Pge Cache: 7.02 MB/sec
2Plane Program Page: 0.00 MB/sec
2Plane Program Page Cache Mode: 0.00 MB/sec
Two-Plane Features

- Device is divided into two physical planes, odd/even blocks
- Users have the ability to:
  - Concurrently access two pages for read
  - Erase two blocks concurrently
  - Program two pages concurrently
- The page addresses of blocks from both planes must be the same during two-plane READ/PROGRAM/ERASE operations
4Gb, Two-Plane, 2K-Page SLC NAND Architecture

- Cache Register: 2,048 bytes
- Data Register: 2,048 bytes

- 2,048 blocks per plane
- 4,096 blocks per device

1 page = (2K + 64 bytes)
1 block = (2K + 64) bytes x 64 pages = (128K + 4K) bytes
1 plane = (128K + 4K) bytes x 2,048 blocks = 2,112Mb
1 device = 2,112Mb x 2 planes = 4,224Mb
4Gb, 2K-Page SLC NAND Performance

Micron (72nm SLC) 4Gb die 2K Page Performance

<table>
<thead>
<tr>
<th>NAND Operation</th>
<th>MB/sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Read</td>
<td>28.94</td>
</tr>
<tr>
<td>Cache Read</td>
<td>37.62</td>
</tr>
<tr>
<td>2Plane Page Read</td>
<td>33.50</td>
</tr>
<tr>
<td>Page Program</td>
<td>7.74</td>
</tr>
<tr>
<td>Program Pge Cache</td>
<td>9.56</td>
</tr>
<tr>
<td>2Plane Program Page</td>
<td>12.94</td>
</tr>
<tr>
<td>2Plane Program Page Mode</td>
<td>19.05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Time</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>tR</td>
<td>20</td>
<td>us</td>
</tr>
<tr>
<td>tDCBSYR1</td>
<td>3</td>
<td>us</td>
</tr>
<tr>
<td>tDCBSYR2</td>
<td>3</td>
<td>us</td>
</tr>
<tr>
<td>tRC</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>tRC (C)</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>tRC</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>tPROG</td>
<td>220</td>
<td>us</td>
</tr>
<tr>
<td>tCBSY</td>
<td>3</td>
<td>us</td>
</tr>
<tr>
<td>tDBSY</td>
<td>0.5</td>
<td>us</td>
</tr>
<tr>
<td>tWC</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>tWC (C)</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>tWC</td>
<td></td>
<td>ns</td>
</tr>
<tr>
<td>PS</td>
<td>2112</td>
<td>Byte</td>
</tr>
<tr>
<td>NP</td>
<td>64</td>
<td>Pages</td>
</tr>
</tbody>
</table>
8Gb, Two-Plane, 2K-Page MLC NAND Architecture

- **Cache Register**: 2,048 bytes
- **Data Register**: 2,048 bytes
- **2,048 blocks per plane**
- **4,096 blocks per device**

1 block = (2K + 64 bytes) x 128 pages
= (256K + 8K) bytes

1 plane = (256K + 8K) bytes x 2,048 blocks
= 4,224Mb

1 device = 4,224Mb x 2 planes
= 8,448Mb
8Gb, 2K-Page MLC Performance

Micron (72nm MLC) 8Gb die 2K Page Performance

Symbol | Time | Units
---|---|---
tR | 50 | us
tDCBSYR1 | 7 | us
tDCBSYR2 | 7 | us
tRC | 25 | ns
tRC (C) | 25 | ns
tRC | 25 | ns
tPROG | 650 | us
tCBSY | 30 | us
tDBSY | 0.5 | us
tWC | 25 | ns
tWC (C) | 25 | ns
tWC | 25 | ns
PS | 2112 | Byte
NP | 128 | Pages

NAND Operation

Page Read | Cache Read | 2Plane Page Read | Page Program | Program Page Cache | 2Plane Program Page | 2Plane Program Page Cache Mode
---|---|---|---|---|---|---
20.51 | 19.92 | 27.06 | 3.00 | 3.25 | 5.58 | 6.49

MB/sec
16Gb, Two-Plane, 4K-Page MLC NAND Architecture

- **Cache Register**: 4,096 bytes, 218 bytes
- **Data Register**: 4,096 bytes, 218 bytes

- **2,048 blocks per plane**
- **4,096 blocks per device**

- **1 block** = 4,096 bytes

**Calculations**:
- 1 page = (4K + 218 bytes)
- 1 block = (4K + 218 bytes) x 128 pages = (512K + 27K) bytes
- 1 plane = (512K + 27K) bytes x 2,048 blocks = 8,628Mb
- 1 device = 8,628Mb x 2 planes = 17,256Mb
Two-Plane, 4K-Page MLC NAND Architecture

Micron (55nm MLC) 16Gb die 4K Page Performance

- Page Read: 27.30 MB/sec
- Cache Read: 37.42 MB/sec
- 2Plane Page Read: 32.41 MB/sec
- Page Program: 4.28 MB/sec
- Program Page Cache: 4.79 MB/sec
- 2Plane Program Page: 7.73 MB/sec
- 2Plane Program Page Cache Mode: 9.56 MB/sec

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Time</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>tR</td>
<td>50</td>
<td>us</td>
</tr>
<tr>
<td>tDCBSYR1</td>
<td>7</td>
<td>us</td>
</tr>
<tr>
<td>tDCBSYR2</td>
<td>7</td>
<td>us</td>
</tr>
<tr>
<td>tRC</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>tRC (C)</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>tRC</td>
<td>50</td>
<td>ns</td>
</tr>
<tr>
<td>tPROG</td>
<td>900</td>
<td>us</td>
</tr>
<tr>
<td>tCBSY</td>
<td>3</td>
<td>us</td>
</tr>
<tr>
<td>tDBSY</td>
<td>0.5</td>
<td>us</td>
</tr>
<tr>
<td>tWC</td>
<td>25</td>
<td>ns</td>
</tr>
<tr>
<td>tWC (C)</td>
<td>35</td>
<td>ns</td>
</tr>
<tr>
<td>tWC</td>
<td>45</td>
<td>ns</td>
</tr>
<tr>
<td>PS</td>
<td>4314</td>
<td>Byte</td>
</tr>
<tr>
<td>NP</td>
<td>128</td>
<td>Pages</td>
</tr>
</tbody>
</table>
Open NAND Flash Interface

- Future Micron NAND Flash devices support the ONFI specification
- Micron is a founding member of ONFI
- The ONFI 1.0 specification is available at http://www.onfi.org/
- ONFI founders:
NAND Error Modes

- Program Disturb
- Read Disturb
- Data Retention
- Endurance
Let’s Get Orientated: NAND Architecture

- NAND architecture is based on independent *blocks*
- Blocks are the smallest erasable units
- Pages are the smallest programmable units
  - Partial pages can be programmed in some devices

*Typical for 4Gb SLC
Cells *not* being programmed receive elevated voltage stress.

Stressed cells:
- Are always in the block being programmed.
- Can either be on pages *not* selected, or in a selected page but not supposed to be programmed.

Charge collects on the floating gate causing the cell to appear to be weakly programmed.

- Does not damage cells; ERASE returns cells to undisturbed levels.
- Disturbed bits are effectively managed with error correction codes (ECC).
- Partial-page programming accelerates disturbance.

Strings being programmed are grounded; others are at 10V.

Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.
Reducing Program Disturb

- Program pages in a block sequentially, from page 0 to page 63 (SLC) or 127 (MLC)
- Minimize partial-page programming operations (SLC)
- It is mandatory to restrict page programming to a single operation (MLC)
- Use ECC to recover from program disturb errors
Cells *not* being read receive elevated voltage stress

- Stressed cells are
  - Always in the block being read
  - Always on pages not being read

- Charge collects on the floating gate causing the cell to appear to be weakly programmed

- Does not damage cells; ERASE returns cells to undisturbed levels

- Disturbed bits are effectively managed with ECC

---

Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.
Reducing Read Disturb

- Rule of thumb for excessive reads per block between ERASE operations
  - SLC – 1,000,000 READ cycles
  - MLC – 100,000 READ cycles
- If possible, read equally from pages within the block
- If exceeding the rule-of-thumb cycle count, then move the block to another location and erase the original block
- Establish ECC threshold to move data
- Erase resets the READ DISTURB cycle count
- Use ECC to recover from read disturb errors
Data Retention

- Charge loss/gain occurs on the floating gate over time; device threshold voltage trends to a quiescent level
- Cell is undamaged; block can be reliably erased and reprogrammed

Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.
Improving Data Retention

- Limit PROGRAM/ERASE cycles in blocks that require long retention
- Limit READs to reduce read disturb

![Diagram showing retention required and block cycles](image)
Endurance

- PROGRAM/ERASE cycles cause charge to be trapped in the dielectric
- Causes a permanent shift in cell characteristics—not recovered by erase
- Observed as failed program or erase status
- Blocks that fail should be retired (marked as bad and no longer used)

Note: Circuit structures and voltages are representative only. Details vary by manufacturer and technology node.
Endurance Recommendations

- Always check pass/fail status (SR0) for PROGRAM and ERASE operations
  - Note: READ operations do not set SR0 to fail status
- If fail status after PROGRAM, move all block data to an available block and mark the failed block bad
- Use ECC to recover from errors
- Write data equally to all good blocks (wear leveling)
- Protect block management/meta data in spare area with ECC
Wear Leveling

- Wear leveling is a plus on SLC devices where blocks can support up to 100,000 PROGRAM/ERASE cycles.
- Wear leveling is imperative on MLC devices where blocks typically support less than 10,000 cycles.
- If a block was erased and reprogrammed every minute, the 10,000 cycling limit would be exceeded in just 7 days!
  
  \[60 \times 24 \times 7 = 10,080\]
- Rather than cycling the same block, wear leveling involves distributing the number of blocks that are cycled.
Wear Leveling (continued)

- An 8Gb MLC device contains 4,096 independent blocks
- Using the previous example, if the cycles were distributed over 4,096 blocks, each block would be programmed less than 3 times (vs. 10,800 cycles if the same block is cycled)
- If perfect wear leveling was performed on a 4,096-block device, a block could be erased and programmed every minute, every day for 77 years!

\[
\frac{10,000 \times 4,096}{60 \times 24} = \frac{40,960,000}{1,440} = 28,444 \text{ days} = 77.9 \text{ years}
\]

- Consider static vs. dynamic wear leveling
ECC Can Fix Everything (well, almost)

- Understand the target data-error rate for your particular system
- Understand the use model that you intend for your system
- Design the ECC circuit to improve the raw-bit error rate (BER) of the NAND Flash, under *your* use conditions, to meet the system’s target BER
ECC Code Selection Is Becoming Even More Important

- As the raw NAND Flash BER increases, it becomes more important to match the ECC to the application’s target BER

For SLC
A code with a correction threshold of 1 is sufficient

$t = 4$ required (as a minimum) for MLC
Another Option: Embedded MMC (eMMC)

- The complexities of future MLC require increased attention; the ECC algorithm, for example, is becoming more and more complex, moving from 4+ bits to 8+ bits in the future.

- A managed interface addresses the complexities of current and future NAND Flash devices; this means the host does not need to know the details of NAND Flash block sizes, page sizes, planes, new features, process generation, MLC vs. SLC, wear leveling, ECC requirements, etc.

- Embedded MMC (eMMC) is the next logical step in the NAND Flash evolution for embedded applications because it turns a program/erase/read device with bad blocks and bad bits (NAND Flash) into a simple write/read memory.
Micron Solutions: Managed NAND (eMMC)

- MLC NAND + MMC 4.2 version controller device
- High-speed solution:
  - Host-selectable x1, x4, and x8 I/Os
  - 52 MHz clock speed (MAX) – 416 Mb/s data rate (MAX)
- Fully backward compatible with previous MMC systems
- ECC, wear leveling, and block management

12 x 16 x 1.3mm BGA package
Conclusions

- NAND Flash is the lowest cost, nonvolatile memory available today
- Complexities of MLC NAND require increased logic and software design
- All of these complexities are addressed through the use of the controller included with eMMC
Thank You