Typical Mobile Computer Flash and Best Use Practices

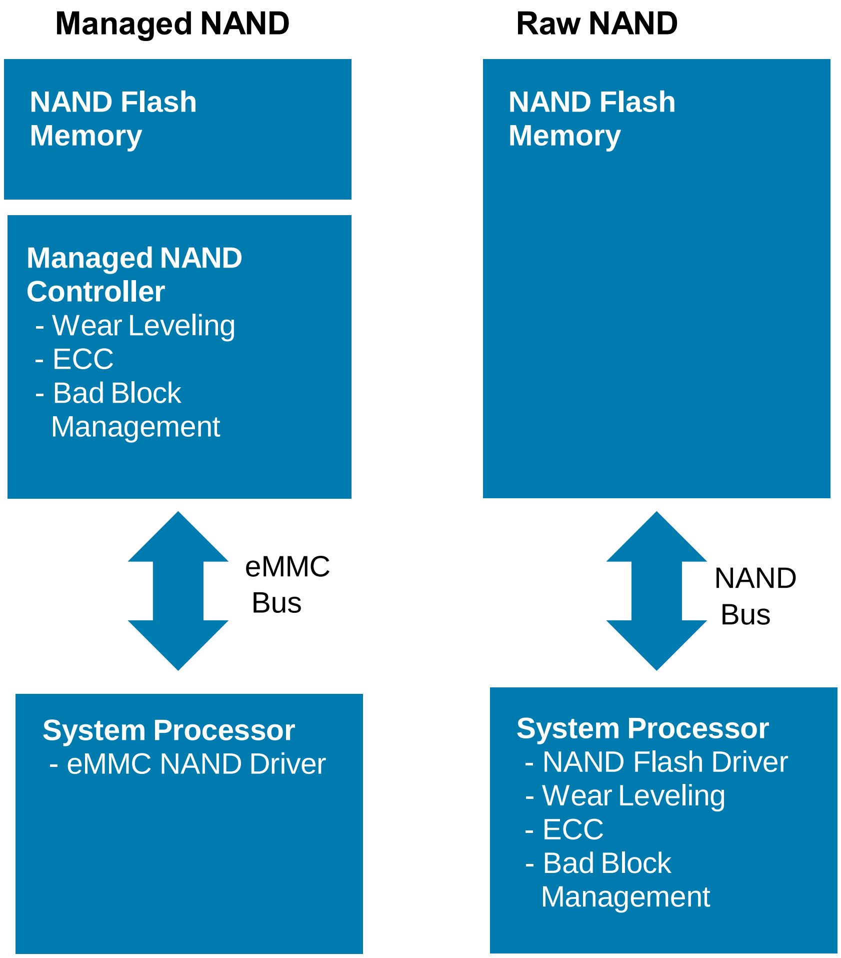
Oct 2017 | Version 1.4

Overview

- Typical Flash Architectures
- Typical Failure Modes Found in Mobile Computing Devices
- Best Practices

Typical Flash Architectures

- Raw NAND Flash
  - Requires external hardware and software to manage the Flash memory
- Managed NAND Flash
  - Does not require as much external hardware and software to manage the Flash memory as it’s an all-in-one solution



Whether the solution is Raw NAND or eMMC-based, the Flash array may typically be one of the following:

- Single Level Cell (SLC)
  - High endurance. Internally data bits are stored only as 2 voltage levels (a logic 0 or 1)

- Multi-Level Cell (MLC)
  - Internally the data bits can be several possible values such as 0 to 3. One cell can store 2 or more bits. The endurance is shorter, typically 10 times less than SLC

As Flash manufacturers strive to pack more density into a smaller size, the endurance of the Flash decreases and the need for higher Error Correction Code (ECC) increases. Since eMMC is an all-in- one solution many Flash manufacturers have added additional robustness through proprietary techniques.

Typical Flash Failure Modes Found in Mobile Computing

CORRUPTION FROM INADVERTENT POWER LOSS DURING A WRITE OPERATION

Flash chips are divided into areas called blocks and pages. Internally, writes happen in one area at a time. A write interruption due to power loss, such as an abrupt battery removal or system reset, can corrupt data within that area.

This is not a permanent failure of the Flash and can be fixed by rewriting the contents of the affected areas (assuming they are known).

READ DISTURB

Read Disturb is a condition in the NAND Flash memory system that occurs when a given area is read excessively by an application. It causes nearby data contained within the Flash to be changed from the original value. Thereby altering the contents of applications, OS, etc.

This is not a permanent failure of the Flash and can be fixed by re-writing the contents of the affected areas. The offending application should be corrected to not read an area of Flash excessively.

OVER-PROGRAMMING

This is an internal condition of NAND Flash memory due to its cells being over exposed to a charge. It is a permanent failure and the result is that all the cells in line (representing the same bit across several addresses) tend to change state from 1 to 0 based on one or more of the cells in that line being programmed to a 0.

Writing many 0 bits repeatedly can cause this type of failure. An example of this would be a file system that pads its writes with 0 data fill.

GENERAL ENDURANCE

As mentioned above different Flash architectures have different endurance. This endurance is normally rated in Program/Erase (P/E) Cycles over the life of the part. Software or firmware is normally added to increase the endurance by wear leveling the part (ensuring all areas are evenly worn instead of allowing the same area's to used and worn more frequently than the rest).

Once an area of Flash has errors that exceed the correction level of the ECC, or can no longer be erased and programmed that area is no longer useable. Typically, Bad Block Management swaps out that area for a spare. Eventually, there will no longer be spares to use. An application and underlying operating system will encounter frequent and consistent read and/or write failures.

In many cases, the actual use exceeds the manufacturers lifetime rating of the part yet the Flash continues to perform well. The reliability of data can, however, no longer be guaranteed.

Best Practices For Mobile Computing Devices

Whenever possible consumable SD cards should be used in place of onboard Flash for frequent and small payload write activities.

Aggregate Data Written. Due to the nature of NAND Flash the smallest writable area is a page, which, in some cases could be 2-4 Kbytes. If an application writes only a few bytes to a file in the Flash file system an entire page (padded with 0 data) is written, once the file operation is flushed or closed. Frequent writes of small data payload to Flash decreases the life of Flash.

Developers should design applications with Flash storage lifetime in mind. Application Flash activity should be periodically measured (write activity and write size in bytes) to ensure lifetime Flash endurance goals are met and not exceeded.

Users should avoid inducing abrupt power interruptions due to system reset or battery removal. Ensure that write operations have completed and files are closed prior to changing the system state.

Applications should avoid frequent reads of the same information in Flash. For example, continuously opening, reading, and closing the same file (such as a library file) over and over should be avoided.