WHITE PAPER

Design Considerations for

SD™ Cards and e.MMC Products

SanDisk

951 SanDisk Drive, Milpitas, CA 95035

www.SanDisk.com

Table of Contents

1. Introduction

This paper gives a brief history of managed NAND flash memory and discusses some of the key considerations when designing with today's high performance SD cards and e.MMC devices.

2. Brief Background on Flash Memory

Flash memory was productized in the late 1980's as a replacement for EPROM. Flash memory was erasable and reprogrammable in system and offered higher densities and lower costs versus EPROM. The most common type of early Flash memory was called "NOR" flash – which allowed for random read and write access to any address. The introduction of NAND flash memory allowed for a more dense array (translating to lower costs), but was only accessible a page at a time – this was ideal for storage applications, but not efficient for direct code execution (code had to be loaded in RAM one page at a time for the CPU to execute from). NAND flash memory quickly became the de-facto standard for high-density solid state storage.

The nature of NAND flash memory requires that before programming (or writing), it has to be erased a "block" at a time – which means the host had to manage the data in the blocks and move it around as needed to free up space. As flash memory blocks are programmed and erased, the cells "wear" out to the point where data can no longer be reliably retained. Flash memory file systems were developed to manage program and erase operations and to extend the life of the memory by techniques such as wear leveling, error-correcting code (ECC), and bad block management.

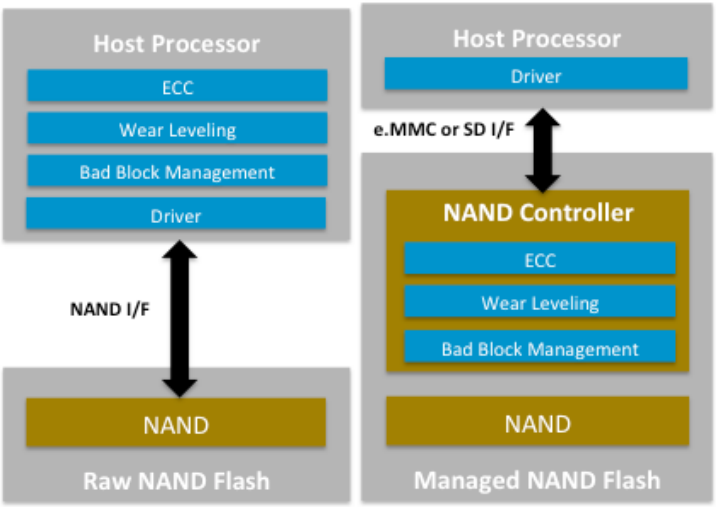
Table 1: Memory Management Techniques

Memory Management Technique	Benefit
Wear Leveling	Move data from frequently erased blocks to less accessed blocks to even out the "wear" on the memory
ECC	Detect and correct data errors during reads
Bad Block Management	Retire "bad" blocks from a pool of spare blocks in the memory

3. Why Managed NAND?

If you are using “raw” NAND memory, your own host controller has to manage ECC and other flash memory management functions. Memory devices like SD cards and e.MMC flash drives included these management functions on the device – so the memory could be used like traditional storage devices. Internal to the device is a small controller, firmware, and one or more stacked NAND die. The host (SOC – system on chip) controller simply needed an interface to the SD or e.MMC bus, a file system and drivers optimized for the card characteristics.

Figure 1: Raw NAND vs. Managed NAND Solutions



As NAND memory became higher density and cell size shrunk, more and more sophisticated ECC and management techniques were required. For most SOC designers, it became easier to off-load the NAND management to a controller coupled to the memory. SD cards offered limited performance for the main OS and applications in multi core/GHz CPUs – so the e.MMC interface became a JEDEC standard way of interfacing to on-board managed NAND. Today’s e.MMC 5.0 interface runs at speeds of up to 400MB/sec.

4. Considerations in Using SD & e.MMC

While the interfaces have been standardized, there are enough differences between the internal operations of different managed memory “devices”, and the host needs to consider these for optimized system level performance and life.

4.1 Performance

The primary differences that users will see between devices relate to performance and “wear” life. There are multiple types of NAND flash memory that may be used internally to the device, for example 2-bit (MLC) and 3-bit (TLC) per cell flash memory. Many devices may also be configured with SLC areas or caches, and multiple memory devices or “planes” of memory may be accessed in parallel for higher performance. Latencies for read or write operations will differ between devices, and need to be taken into consideration on the host side (for example if you are continuously recording a data stream, you need to be able to buffer enough of that stream on the host side in RAM to account for the worst case write latency on the managed NAND device). For SD cards, "Speed Class" refers to the "minimum" sustained write speed under a specific SDA defined write pattern; a detailed description of this write pattern is available to SDA members in the SD specifications. If your write pattern significantly differs from the SDA spec, you may observe significantly different performance. Maximum write speeds are also specified in datasheets for SD cards and e.MMC devices – but this will also be dependent on the host side implementation.

4.2 Write Amplification and Write Endurance

Another key consideration is "write amplification". When the host writes a page or "pages" to the memory device – internal operations inside the managed NAND device will cause additional writes to occur to the memory (management overhead). Large sequential writes aligned to page boundaries typically result in the least "amplification" (In a well designed system this amplification factor could be ~1.1). Small random writes typically result in the most amplification (in a bad design, the factor could be >2.0). This becomes important when considering the life of the memory device. See the example below:

Table 2: Example of Write Amplification Impact

Item	Value
Device Capacity	8GB
Write Endurance	2K Program/Erase cycles
Data Written Per Day to Device	2GB
Expected Life w/ WA=1 *=(8x2000)/(2*1)*	8,000 days
Expected Life w/ WA=5 *=(8x2000)/(2*5)*	1,600 days

To estimate the write amplification for any particular device, and hence its expected life, you can capture a log file of your writes. In some cases you may find that a TLC product rated at a fraction of the write life will be sufficient in your design and be a way to reduce cost. Because of SLC caching and other management techniques, vendors are moving away from a pure cycle rating to a terabytes written (TBW) rating on newer devices, which can better reflect the devices true capability.

4.3 Read Endurance

As memory cells get closer together, reading one cell can disturb the data in surrounding cells. If the same cell is read enough times, data from surrounding cells may not be readable anymore, resulting in a UECC (uncorrectable ECC). For many applications, this will not be a critical consideration, since writing the device will trigger wear leveling, which will move the least used data, by re-writing it somewhere else in memory, and thus “refreshing” the cells. However in applications that rarely write data to the device, but are constantly reading the same areas, host side refresh techniques maybe required to ensure you do not reach the read-endurance limits. If this is a concern, you should discuss your access patterns with your memory vendor’s applications engineering team to better understand this phenomenon. The simplest refresh technique is to just cycle through the memory periodically and read and rewrite each page of data.

4.4 Data Retention

Data retention changes with temperature and cycle life of the memory device. JEDEC specifies 10 years @ 55^0C data retention for a fresh device (typically that means it has not been erased or programmed more than 10% of its rated cycle life). Data retention is not linear with temperature. The first step to understand if data retention is a consideration in the device is to create a worst case daily temperature profile for you application. The memory vendor should then be able to help you calculate expected life. Life can be easily extended by “refreshing the data” – i.e. re-writing the data back to a different memory location, and in some cases it may be recommended that the host triggers such a function periodically (again if you are writing enough data to the device, the standard wear leveling mechanisms should take care of this for you – so it is of most concern in low write environments).

4.5 High Reliability Applications

For most standard applications, a UECC rate of 1x10^15 reads or so may be acceptable. Depending on the type of data that is lost, the host may still be able to recover (e.g. reboot) and reload the data. A journalled file system may also be used for better protection. The most critical areas of data typically include the internal flash memory management tables, boot partition and the host OS kernel. Memory vendors may keep backup copies of internal flash memory management tables to enable recovery in the event of a UECC read. In addition, boot partitions are typically kept in SLC areas with a redundant copy for extra reliability. A backup copy of the host OS kernel and other critical data structures may be kept in the user area to allow for additional redundancy – that would typically be managed by the OS vendor.

4.6 Operation Time Out

During a read or write operation to the flash memory, the operation may timeout; this typically happens if the host driver is not following the maximum timeout recommendations for read and write operations. The host needs to consider error handling on a timeout, and retry the operation as necessary.

4.7 Power Immunity

The host design needs to consider the effect of power loss during flash memory operation. Many e.MMC and some SD card products will be resilient to some degree if power is interrupted to the memory during an erase or write operation. The host should be able to recover from power loss during READ, simply by resetting the flash memory device and starting the read again (or in a worst case, by rebooting). In the case of a write operation, if the host still has the data, the write can be attempted again. The worst case scenario is typically power loss during an internal "housekeeping" operation in the flash memory. If an internal management table or pointer is corrupted, this can lead to unrecoverable data scenarios. Power immunity hardware and software may protect against this to varying degrees – but good host side power supply design is the key to maximizing robustness (such as sufficient power supply capacitance and avoiding rapid power cycling).

4.8 Layout Considerations

As e.MMC devices operate on a high speed bus, host designers should follow the vendors recommendations on signal layout. Routing may be possible through NC balls on the BGA to minimize the number of board layers and vias– again the host designer should check with the vendor for specific layout guidelines.

SD cards can also operate in a number of interface modes. The SDA provides documentation with recommendations on how to design with this interface and that should be studied carefully – especially when using the higher speed UHS-104 and UHS-II buses.

4.9 Programming

Many e.MMC devices are typically pre-programmed before surface mount (SMT) – however as NAND geometries shrink, careful consideration maybe required as to how the device is pre-programmed and in some cases, data may need to be loaded or checked on the host board (after SMT). Again, check with you flash memory vendor for recommended practices for the specific device you are using.

5. Summary

NAND flash memory is a complex technology and is becoming more and more complex with each geometry shrink. Its intrinsic challenges require sophisticated flash management techniques. When designing with today's high performance SD cards and e.MMC devices, taking into consideration the key topics discussed in this paper will help to optimize system level performance and product life.