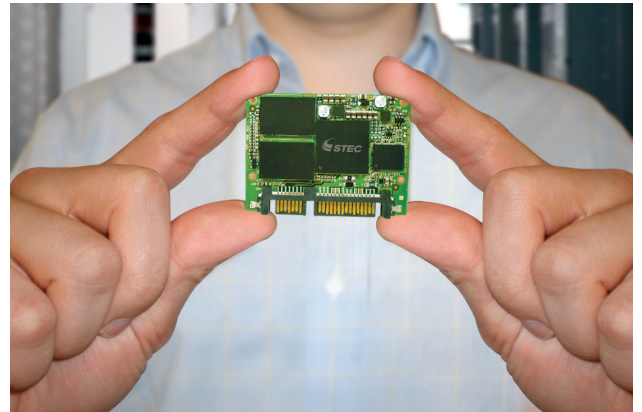


# Best Practices for Optimizing Flash Media Performance in Embedded Applications

---



To optimize performance, reliability and endurance in flash storage subsystems, embedded system designers should incorporate best practices for flash integration as part of the platform design process.

## Contents

Executive Summary.....	2
NAND Architecture.....	3
Programming/Erasing NAND Flash.....	3-4
SLC vs. MLC NAND.....	4
Sequential vs. Random Access.....	4
Effects of Transfer Size and File Size.....	4
Optimized Solution for Small Random Write Access.....	4-5
Additional Recommendations for Optimizing Performance.....	5
About STEC.....	5

## Executive Summary

Flash-based solid-state drives (SSD) are replacing hard disk drives (HDD) in embedded market segments, including telecommunications, networking, industrial systems, healthcare devices, gaming, transportation and military/aerospace applications. SSDs from STEC are the perfect solution for applications that demand significantly more performance and reliability, in addition to lower latencies and higher endurance, than can be provided by traditional rotating media. To maximize the performance of SSD-based devices, optimizations should be made during the system design process. This paper highlights best practices for embedded system designers.

## Terminology

Table 1 below describes common terms used in this document.

Term	Definition
<b>Block</b>	A <i>block</i> is the smallest unit of flash memory that can be erased during a single erase operation. Current generation NAND technology employs block sizes of either 128KB or 256KB.
<b>Page</b>	A <i>page</i> is the smallest unit of flash memory that can be programmed during a single program cycle of a write operation. Each block is divided into multiple pages. Current generation single-level cell (SLC) NAND technology employs page sizes of either 2KB + 64 bytes for ECC or 4KB + 128 bytes for ECC. Multi-level cell (MLC) NAND technology employs page sizes of 4KB + 256 bytes for ECC or 8KB + 512 bytes for ECC.
<b>Bad Block</b>	A <i>bad block</i> is an unusable block on which no program or erase operations can be performed. Bad blocks contain one or more invalid bits and are removed from the pool of free blocks that can be used for storage. NAND manufacturers guarantee that no more than 2 percent of the flash will contain bad blocks prior to shipping.
<b>Grown Bad Blocks</b>	<i>Grown bad blocks</i> are blocks that are good at time zero, but experience non-recoverable errors during device operation.
<b>Logical Sector</b>	A <i>logical sector</i> is the smallest unit of storage accessible by an operating system's block device driver. The size of a logical sector is 512 bytes.

Table 1: Terminology

## NAND Architecture

NAND flash devices consist of an array of cells, formed by the intersection of bit lines and row lines controlled by transistors. The memory array is divided into physical blocks and pages, which are used by the flash device to manage program and erase operations. Page and block sizes can differ by manufacturer as well as by flash generation. For the purpose of this white paper, 4Xnm SLC NAND flash chips are used as an example. The 4Xnm SLC flash chips consist of 128KB blocks, divided into 64 2KB pages, with each containing an additional 64 spare bytes (typically used for error-correcting code memory (ECC)). Figure 1 shows the high-level architecture of a NAND flash device.

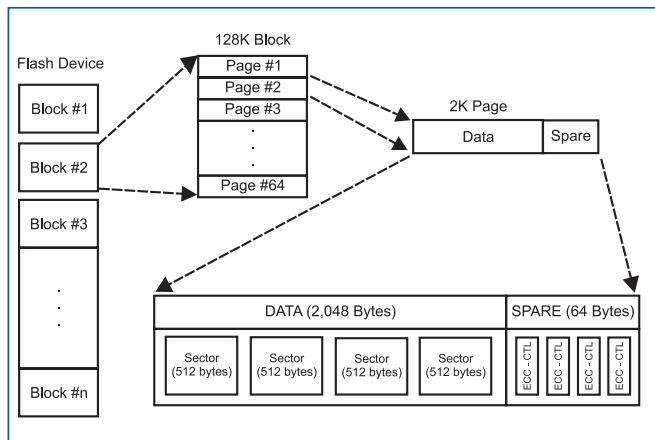


Figure 1: NAND Flash Architecture

For industrial flash devices that are 128MB or larger, STEC uses SLC NAND flash components with blocks that are 256KB each with 64 4KB pages. Using flash with larger (4KB vs. 2KB) pages and larger (256KB vs. 128KB) blocks helps to increase performance. This is because programming more bits per program operation and erasing more bits per erase operation improves efficiency and throughput.

Historically, program and erase times have not changed very much. NAND flash devices that once employed 512-byte pages and 16KB blocks took about as long to perform single program or erase operations as they do today. For this reason, increasing the number of bits programmed and erased during each operation has significantly increased performance.

## Programming/Erasing NAND Flash

Each flash cell has two transistors: one used to select the bit line and another used to control the movement of electrons (representing the state of the bit stored in a cell) to and from the floating gate. The floating gate gets its name from the fact that it is separated from the control gate, source, and drain by a thin layer of oxide. The amount of negative charge trapped on the floating gate determines the bit value stored in the cell. Figure 2 shows the architecture of a NAND cell.

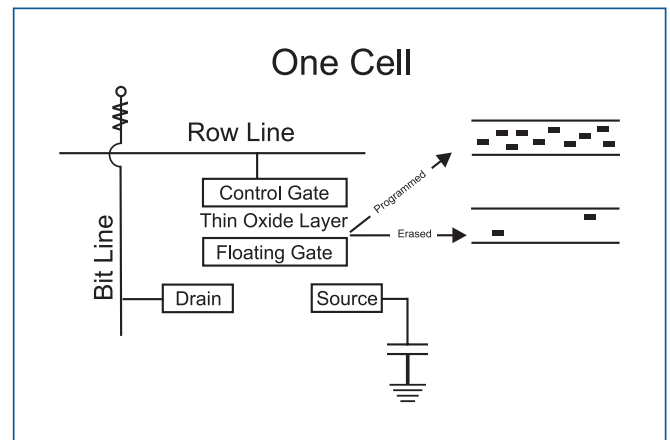


Figure 2: NAND Flash Cell Architecture

Performing a program operation (storing a “0” in a cell) causes an abundance of electrons to be placed on the floating gate. To change the value of the cell from a “0” to a “1” an entire block must be erased to remove the negative charge stored during a previous program operation.

In the past, a technique called “partial page programming” permitted the programming of specific cells within a page. Today, SLC NAND does not support this functionality, requiring the controller to implement a read-modify-write technique to modify a page whose contents have only partially changed. This procedure negatively impacts performance due to the requirement for a read operation prior to the programming of the page. Current NAND flash technology also limits the number of program operations that can be performed on a page before the block to which it belongs must be erased. In the recent past, that number was four; today, some devices only allow one. As with the removal of the partial page programming functionality, this places the burden on the internal controller to make up for any resulting degradation in performance.

## SLC vs. MLC NAND Technology

With the continuing demand for higher densities at lower costs, MLC NAND flash technology has gained popularity in recent years. While SLC NAND technology stores a single bit of data in each cell using two voltage levels, 2-bit MLC NAND can store two bits of data in each cell, using four voltage levels.

Since the voltage range used by SLC and MLC-based devices to represent stored data is the same (either 0 to 1.8V or 0 to 3.3V), the voltage window used to represent each of the four possible states on an MLC device is consequently much narrower than on an SLC device. These reduced voltage windows have an impact on both data reliability and device performance in several ways:

- Lost electrons cause a shift in the voltages used to represent the state of the stored bits, causing an elevation in the bit error rate.
- Write performance is degraded due to the need for a slower, more precise selection of electrons during the programming process. Read performance is similarly impacted, due to the need for longer more granular sensing of current levels.
- Power consumption typically increases, due to the longer times spent reading and programming.
- Flash endurance and data retention are compromised because of the stress resulting from extended read and program operations. Today's MLC NAND specifications allow up to 10,000 program/erase cycles, with future generations only supporting 3,000.

While MLC NAND technology excels in cost per bit, the increased complexity involved with supporting more and narrower voltage windows leads to reduced performance, reliability and endurance on a raw NAND level. This generally makes MLC less suitable for applications that require high data reliability and performance. In addition, MLC flash is currently only available in capacities of 64Gb (8GB) and greater, which imposes a minimum density limitation on the storage devices that use it.

## Sequential vs. Random Accesses

For read accesses on a raw NAND level, sustainable performance does not differ greatly when we compare sequential and random transfer rates. When performing large (32-256 sector) read accesses, the performance is virtually identical. With smaller read accesses there can be a relatively small (5-12 percent) performance degradation when data is accessed randomly.

For write accesses on a raw NAND level, there is significant performance degradation when data is written randomly. Even when writing full 128KB (256 sector) blocks, there is a 20 percent performance degradation when writes are performed randomly rather than sequentially. For random single sector transfers, the decrease in performance can be greater than 95 percent.

## Effects of File Size and Transfer Size

The size of the file to be transferred, as well as the amount of data transferred to and from the flash during each program or read operation, can have a significant impact on performance. Considering the overhead required for each data transfer operation, it is obvious that sustainable data rates will be significantly higher when transferring large amounts of data, compared to transferring data in smaller increments. When data is read or written sequentially, transferring 256 sectors worth of data in one transaction compared to transferring 256 sectors individually results in a performance benefit greater than 2X. While sustainable random read performance is not severely affected by transfer size, there can be more than a 95 percent degradation in random write performance when data is written one sector at a time, compared to using the maximum ATA transfer size of 256 sectors.

## Optimized Solution for Small Random Write Accesses

As discussed in the two previous sections, worst-case sustainable performance occurs with random writes to small numbers of sectors. Many applications do, in fact, require data to be written in this fashion. Examples include POS terminals, voicemail systems and telecommunications equipment that use the flash media to log call data and billing information. In addition, some embedded operating systems are only set up to perform single-sector accesses.

STEC has developed firmware specifically tailored to increase the sustainable write performance of small (1-8 sector) random write transactions. The performance gain can be as great as 700 percent when performing single-sector operations and up to 50 percent with 8-sector operations. This firmware option can be specified by adding a “-S” to STEC’s standard industrial flash part number. For example, part number SLCF128M2TU-S represents a 128MB Compact Flash card that contains firmware optimized for ‘small-sector’ write transactions.

### Additional Recommendations for Optimizing Performance

*Transfer mode:* Wherever possible, operate in Multiword DMA (MWDMA) Mode 4, or if enabled, UDMA Mode 4. The default configuration for STEC’s M2-series industrial flash products is MWDMA selected with UDMA supported. MWDMA Mode 4 operates with an 80ns cycle time and UDMA Mode 4 operates with a 60ns per 2 cycles. In addition to faster throughput, operating in DMA mode reduces the time the host processor must spend waiting to receive or transfer data, which can help increase overall system performance.

*Page boundary alignment:* Where possible, it is best to align data transfers with 2KB boundaries for Compact Flash cards with capacities less than 1GB, and with 4KB boundaries for CF cards with 1GB or higher capacities. If data transfers are not aligned with 2KB or 4KB boundaries, read-modify-write operations become necessary, which increases latency and slows overall performance.

*Root directory/FAT updates:* Where possible, minimize or consolidate accesses to these structures that are used on the flash media by the operating system. Typically, these accesses are small (often just one sector) in size, and they carry a performance penalty similar to that outlined in the small random write accesses section above.

*Bus Width:* Operating in the default 16-bit mode provides about twice the performance of 8-bit mode.

*Status register polling:* Drivers should be written so that the card/drive’s status register is polled before issuing commands or initiating data transfers, rather than using hard-coded wait loops. Polling for 0x50 in the status register before issuing a command and then polling, again, for 0x50 before initiating a data transfer will ensure both reliable operation and optimal performance.

### STEC: The Leader in Enterprise-Class Solid-State Solutions

From storage to server applications, STEC enterprise-class solid-state solutions are designed from the ground up to help you manage unprecedented volumes of data by accelerating access to the data you need, and reducing the cost to access that data. By incorporating STEC SSDs top-to-bottom in your enterprise infrastructure, you can deploy tiered solutions tailored to your exacting requirements, from reducing server sprawl to optimizing server utilization and optimizing address space.

At STEC, we are maintaining our enterprise storage leadership by investing world-class engineering resources to design and develop advanced solid-state technologies that maximize reliability, availability and serviceability, while reducing total cost of ownership in your storage and server applications.

For more information on STEC products, solutions and technology, please visit [www.stec-inc.com](http://www.stec-inc.com)

 [facebook.com/userstecinc](https://facebook.com/userstecinc)

 [twitter.com/stec\\_inc](https://twitter.com/stec_inc)

 [youtube.com/user/stecincssd](https://youtube.com/user/stecincssd)

+1.949.476.1180  
3001 Daimler Street, Santa Ana, CA 92705

©2012 STEC, Inc. The STEC name, logo, and design are trademarks of STEC, Inc. All other trademarks are the property of their respective owners. 3/12 61000-006001-002