TECHNICAL PAPER

Advantages of Large Erase Blocks

SAMUEL DUFOUR
MEMORY COMPONENTS DIVISION

WINK SAVILLE
SAVILLE SOFTWARE INCORPORATED

September 1995

Order Number: 297637-001

Additional copies of this document or other Intel literature may be obtained from:

Intel Corporation
Literature Sales
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

CG-041493

1.0 INTRODUCTION

What is a flash large block? More specifically, what is a flash erase block and why should I care about it from a systems level perspective? These questions are the focus of this paper and will be answered in the following pages. However, to understand these questions better, we must first go back to the predecessors of flash technology (specifically UV erasable EPROM and EEPROM). EPROM and EEPROM offer excellent nonvolatile code storage capability. However, these technologies suffer from their inefficient per-byte alterablility (meaning that the cell erase is part of the rewrite); neither of these technologies are designed for in-system updateability and both have relatively short life expectancies when updated frequently, which affect their universal acceptance as data storage devices.

First generation flash memories had a "bulk erase" geometry and were erased like EPROMs; an erase operation removed the charge from all transistors in the entire memory array at the same time. Newer flash memory architectures, like Intel's FlashFile™ memory, erase in individual blocks, making them more suitable for both file-storage and code-storage applications such as mass storage and code execution subsystems. Moreover, Intel built upon its knowledge of EPROM and derived its flash memory devices from an EPROM base; thus Intel Flash devices take advantage of the acquired data and knowledge Intel obtained from more than 20 years of producing EPROMs. Figure 1 shows the similarities shared between an Intel FlashFile memory cell and an EPROM cell.

As flash technology has matured, it has evolved to be better-suited for use in data acquisition in part due to its increased reliability, performance and decreasing cost. Manufacturers of flash have been striving to maximize the value of flash for their customers. Due to differences in design or technology, providers of flash have tried various arrangements of erase block sizes to reach what they consider to be the "optimal" size. Hence, erase block sizes vary from one flash memory vendor to another, and from device to device, based on the targeted application. Flash memory device and technology architecture can be classified into two distinct groups: fine erase blocks and coarse erase blocks. The first group possesses small erase-block granularity (4 KB or less) as in Triple-Poly and NAND devices[1, 2]. The second group is characterized by large erase-block granularity (32 KB or 64 KB) as in NOR- based flash devices[3].

Why do most flash technologies bother with multiple erase blocks? What advantages are there in having multiple erase blocks? More specifically, how should I choose a solution that will optimally meet my application needs while remaining both cost-effective and easy to employ? Ultimately, multiple block arrangements are much more versatile and effective than first generation bulk erase flash memories.

Compared to the existing solutions for flash design, the best optimization of cost-effectiveness, reliability, and ease-of-use with software is found in the larger block architecture within Intel's NOR flash technology.

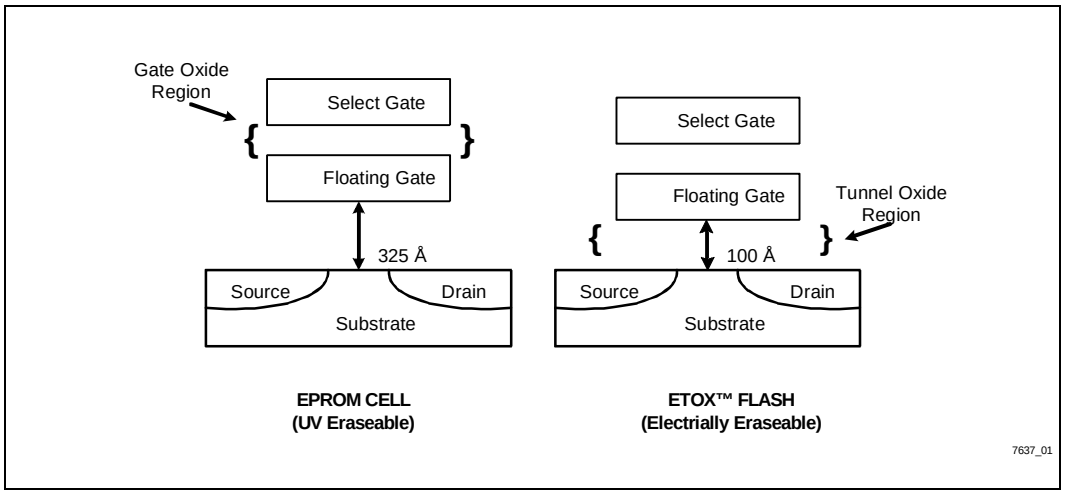

Figure 1. ETOX™ Flash Memory Cell Similarities Leverage EPROM Learning Curve

1.1 Cost Effective

The motivation for small-block flash came from a desire for smallest possible erase times and close duplication of today's sector-based rotating media architecture. The magic number 512 bytes, used by many small block vendors, conforms to the number of bytes DOS and other operating systems use in creating sectors on magnetic media. Each sector on magnetic media usually consists of 512 bytes of data; accordingly, these smaller erase-blocks closely conform to the way operating system software organizes files on mechanical hard disks. Another popular small-block size is 4 KB. The 4-KB block size is a trade-off between cost and performance. There is certainly no technical disadvantage to having smaller erase-blocks; in most situations, small-block devices are capable of treating several erase blocks as a single large erase block when situations call for larger blocks.

Overall die size constitutes the main reason for choosing the "larger" erase-block size. Cost relates directly to die size, so the smaller the die size, the lower the cost. One way to think about this is that there are certain overheads associated with interfacing to the memory array. The overhead is manifested in items such as x-decoders and y-decoders (for selecting rows and columns of the array) that are amortized over the total array. Flash possesses the well-known problem that the smaller the memory blocks (number of memory cells), the larger the area of the overhead relative to the memory array. Simply put, the smaller block size the more periphery needed to deal with a larger number of these smaller units. Some of this overhead scales linearly with block size, some scales exponentially. Also, the array tends to use the greatest portion of the die; thus by adding more physical block boundaries to the array to create smaller erase-blocks, less die is available for use as memory cells.

Cost relates directly to die size, so the smaller the die size, the lower the cost.

Die size, process complexity and yield are the major aspects of memory cost; though many flash vendors try to focus only on their small memory cells. Cell size plays a role in the number of bits per square inch of silicon for the actual memory array, but the periphery area influences the overall die size as well. The comparison of the actual die size of Intel's 28F400 4-Mb flash memories and Toshiba's 4-Mb NAND EEPROM in Table 1 illustrates this point. The table shows that for a given density on an equivalent generation of technology (same lithography), a NAND chip is larger than an ETOX NOR flash memory. The complex and larger decoding circuitry in a NAND chip offsets the advantage of a smaller memory cell. The low array efficiency signifies a larger area dedicated to peripheral circuits instead of memory cells. Adding periphery such as error correction code (ECC) to increase data reliability also decreases effective chip yield and increases chip cost.

Table 1. Array Efficiency Comparison of Large and Small Erase-Blocks

	Toshiba 4-Mb NAND EPROM	Intel 4-Mb ETOX™ II Flash Memory	Intel 4-Mb ETOX™ III Flash Memory	Estimate of Intel 4-Mb Flash Memory(1)
Lithography	0.7 µm	0.8 µm	0.6 µm	0.7 µm
Cell Size	4.83 µm²	7.25 µm²	3.60 µm²	5.27 µm²
Chip Area	303 mils sq.	308 mils sq.	265 mils sq.	286 mils sq.
Array Efficiency	**34.2%**	49.7%	33.0%	**43.7%**

NOTE:

1. Estimate includes general blocking x8 array organization for data storage for compatible Intel 4-Mb ETOX flash memory.

Given:

Die Size (mils) = DS
Cell Size (µm) = CS
Density (Mb) = D
Array Efficiency (AE) is calculated using the following formula: $AE = (CS \times 1024 \times D) \div (DS \div 1000)^2$

1.2 Reliability

In order to erase a small block flash cell, 20V is typically applied to the substrate itself in addition to the select gate. Small block technologies require this method to achieve their fast erase and write times.

Unfortunately, higher dielectric stress across this tunnel oxide reduces long-term device reliability; reduction of this voltage stress as much as possible becomes desirable. However, reducing the dielectric stress impacts the performance by increasing the cell erase time. To compensate for increased cell erase time and to maintain high device erase throughput, Intel chose to implement its current optimized 64-KB erase block size. This strategy is best exemplified by weighing Intel Flash's reclaim performance against erase block size (Figure 2). Since most flash memory implementations interleave memory to squeeze maximum performance out of the memory subsystem (hardware interleaving permits sequential memory reads by allowing access to multiple flash blocks), the resulting size of two combined interleaved blocks is 128 KB. Figure 2 shows that 128 KB is optimal for achieving the greatest reclaim performance.

Due to the high voltages involved when erasing small blocks, the possibility for tunnel oxide breakdown is much greater. Intel's ETOX (EPROM Thin Oxide) NOR Flash technology has been designed to use Erase and Program Algorithms that minimize the time/voltage stress applied to this tunnel oxide. We have consciously decided to trade off a longer erase time for less stress across the oxide and, therefore, improved reliability.

NOTE:

The cell erase time of the ETOX processes is a direct result of the relatively low 12V and low current used to pull electrons from the floating gate.

Comparison of erase times for Intel's FlashFile architecture parts (based on 64-KB erase blocks) versus a competitor's NAND part of the same size, reveals that the Intel part can take up to 97 times longer to erase than the small block based part. The trade off is that Intel's FlashFile ETOX technology devices have the best reliability (as measured by MTBF) in the industry. In addition, the phenomena known as "program disturb" must also be considered when placing high voltages across the substrate of a flash memory device. Program disturb is an electrical-stress-inducted charge loss during programming whereby floating-gate charge is lost through the drain region of the cell. By reducing the voltages across the substrate, the probability for program disturb errors is decreased. It has also been seen with a number of technologies that dividing the memory array into small blocks aggravates this disturb condition further.

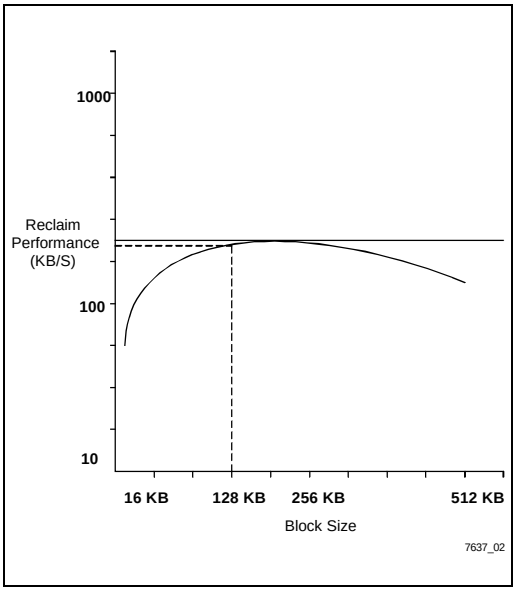

Figure 2. Reclaim Performance vs. Erase Block Size. Erase-Block Reclaim Is the Process of Reconditioning a Block So It Can Be Reused for Data Storage

1.3 Wear Leveling

Data updates are often a very important part of the requirements of a design; accordingly, the ability to erase and reprogram (cycle) a flash memory device many times can be a critical factor in data intensive applications. Naturally, as the flash erase blocks are cycled, the tunnel oxide is subjected to repeated stress cycles. It can be readily understood that it is quite possible for one specific erase block (like Block 0) to end up being cycled almost constantly, with other blocks cycled less often. Since all flash memory devices have a maximum number of cycles within which the device will meet specs, it would be most advantageous to somehow evenly distribute erase/program cycles across all of the erase blocks on the device—this is called wear leveling. Therefore, it is important that the flash media chosen be

able to allow software to initiate proper wear leveling schemes. Both NAND and NOR technologies allow software to map logical memory locations to physical locations within the flash. Software wear leveling algorithms help to evenly move files around on the flash media, thereby decreasing the wear on specific physical erase blocks on the media. Therefore, to maximize the longevity of flash media in both small and large block architectures it is useful to implement wear leveling techniques.

1.4 Ease-of-Use

In all cases, simple memory mapping software can imitate small blocks on large block flash thus removing physical block size as a dependent variable. Since erasing and writing to flash memory is clearly a different operation than rewriting information in place to a hard disk, software techniques are necessary for flash emulation of disk functionality.

Now that flash technology is widely available, and has become an industry standard, many enabling software solutions are available. However, it is not within the scope of this paper to determine which file system best meets the needs of specific applications. To figure out which file system to use, Intel offers a File System Selection Guide available from the Intel FaxBack* at 1-800-628-2283 or (916) 356-3105 (Order Number 2258).

PC-compatible software like the industry-standard Flash Translation Layer (FTL), allows Intel's Flash memory components and flash cards to emulate the popular 512-byte sectors of a hard disk. Utilizing an existing sector-based file system to provide file handling capabilities, the FTL solution transparently tricks the upper layer software into believing a normal sector-based drive exists by translating the drive requests which pass through it. FTL also employs wear-leveling and reclamation algorithms that prevent any block from being cycled excessively, thus assuring millions of hours of reliability. Flash filing systems make the management of flash memory devices completely transparent to the user.

> **Flash filing systems make the management of flash memory devices completely transparent to the user.**

Another type of Flash File System is the Linear File Store (LFS) File Manager, or LFM. LFM is Intel's implementation of the PCMCIA-defined LFS, and offers a mini-file system that provides basic file system functionality for reading and writing variable size file objects. The file objects are stored contiguously in a partition and are arranged in a one-way linked list.

Intel specifically designed and implemented the LFM file system for the embedded market, where a certain amount of functionality may be sacrificed in order to obtain a file system for flash with a very small footprint. LFM uses PCMCIA-defined LFS structures to create a link-list type of file system. Each file is guaranteed to be contiguously laid out in order to support eXecute-In-Place (XIP) applications, which are primarily found in embedded systems that need to use memory efficiently in order to keep costs low.

Virtual Small Block (VSB) File Manager or VFM is a relatively new software technology that provides a means to easily manage smaller message segments (sectors) while supporting a cost-effective 64-KB flash blocking structure. VFM is a mid-range sector-based file system, similar to its FTL big brother. VFM provides a low-cost embedded Resident Flash Array (RFA) implementation without requiring DOS compatibility or PCMCIA socket services.

A Voice Messaging Device used to store voice messages relates an excellent example of a tailored flash software solution meant to meet the needs of an OEM customer. Intel's solution was to use its cost-effective ETOX large block flash memory united with Intel's Virtual Small Block File Management software. The VFM code manages the storage of variable length files in flash memory, including the capability to make file inserts and deletions, making it feasible for portable audio devices to store voice messages.

All current flash media share one common characteristic, they are slower writing than reading. The read performance can be 2 MB/s or more, while the write performance is typically around 100,000 bytes/s. For many user models this is not a barrier since read operations dominate and 100 KB/sec may be fast enough for human interface data capture (such as speech).

Though flash memory's write performance is relatively slow, large block flash offers a higher maximum write performance than small block flash. However, small block flash has the advantage that its worst case performance tends to be better than large block flash's worst case performance. Large block's worst case scenario, for example, is when the flash media is completely full and a write of 512 bytes is called for. In this case, the first valid data in a 64-KB erase block must be relocated then the block must be erased before the 512-byte write can occur. The small block's worst case, on the other hand, is the same as its best case;

when a small block architecture wants to write 512 bytes, it just does an erase and then writes the data. Generally, this is only a problem for applications which require real-time write performance and for most applications is not a problem.

2.0 CONCLUSION

In summary, large flash erase-block architectures offer the lowest cost-per-bit solution because of low overhead. Furthermore, via software, large block flash can be made to act like small block flash in terms of emulating a sector-based mass storage media. Moreover, the current demand, constituting nearly 80% of the total flash market, strongly favors large block NOR flash. The performance and cost-effectiveness of Intel's large erase block flash provides the most cost-effective, system-level solution verses that of its competitors. With Intel Flash and simple file mapping software, original equipment solutions get to market quickly.

3.0 ADDITIONAL INFORMATION

3.1 Bibliography

(1) Mehrotra, et. al., "Serial 9 Mb Flash EEPROM for Solid State Disk Applications," SYMPOSIUM ON VLSI CIRCUITS, p. 24–25, 1992.

(2) Yasuo Itoh, et. al., "An Experimental 4 Mb CMOS EEPROM with a NAND Structured Cell," ISSCC DIGEST OF TECHNICAL PAPERS, p. 134–135, 1989.

(3) A. Baker, et. al., "A 3.3V 16-Mb Flash Memory with Advanced Write Automation," ISSCC DIGEST OF TECHNICAL PAPERS, TA 8.5, 1994.

Revision History

Number	Description
-001	Original Version

Acknowledgement

Wink Saville is president of Saville Software Incorporated. Wink has been an independent contractor working with system software and low-level file systems for 15 years. Saville Software Incorporated is located at 4425 Esta Lane, Soquel, California 95073. InterNET: wink@saville.com. Phone: 408-479-7199.

Filename:	297637_1.DOC
Directory:	C:\TESTDOCS\DOCS
Template:	C:\WINDOWS\WINWORD6\TEMPLATE\ZAN____1.DOT
Title:	E
Subject:	
Author:	Mary Ann Hooker
Keywords:	
Comments:	
Creation Date:	09/12/95 4:43 PM
Revision Number:	23
Last Saved On:	12/06/95 11:57 AM
Last Saved By:	Ward McQueen
Total Editing Time:	89 Minutes
Last Printed On:	12/06/95 11:57 AM
As of Last Complete Printing	
Number of Pages:	7
Number of Words:	2,799 (approx.)
Number of Characters:	15,958 (approx.)