Flash in the Enterprise

Jamon Bowen
Texas Memory Systems

TMS
TEXAS MEMORY SYSTEMS

September 2007

Contents

Section 1

Introduction

Flash memory is poised to become a major player in datacenter storage systems. It provides an all electronic method of storing data in silicon, removing the only component in modern data processing systems that still relies on moving parts in the data path: the spinning hard disk. Though Flash memory offers some compelling advantages over disk-based storage, it is not well understood in the storage industry. This paper describes some the properties of Flash memory and then explains how the RamSan-500, produced by Texas Memory Systems, leverages its strengths and compensates for its weaknesses to offer the first enterprise ready Flash system.

Section 2

The Genealogy of Flash

Flash memory is a 1980s invention that offered a revolutionary property to the integrated circuit community: it could be both programmed and erased electronically and retain its state without power being applied. RAM (Random Access Memory) chips have been available for decades allowing electronic reading and rewriting, but power has to be constantly applied for the RAM to maintain its state. Alternatively, EPROM (Electronically Programmable Read Only Memory) chips allowed a manufacturer to electronically program chips that would retain their state without power, but they could only be programmed once and never erased[i]. Then, EEPROM (Electronically Erasable Programmable Read Only Memory) was invented as the first chip that could be both programmed and erased electronically and maintain the programmed value without power.

It is useful to note that the EEPROM acronym contains ROM – read only memory. At first it may seem odd that rewritable media would be considered "read only," but when you examine how EEPROMs work, the reason that "ROM" remains becomes clear. This type of memory does not have the same characteristics as RAM, which allows any arbitrary location to be read from or written to at any given point in time. EEPROM can have any arbitrary location read from at any given point in time; however, the entire chip must be erased in-full at once, and the addresses in the chip have to be programmed sequentially.

This explains why, despite EEPROM 's ability to be rewritten, it was considered, for all practical purposes, to be read-only memory. At first, EEPROM was primarily used to store programs for embedded processors and only needed to change when programmers fixed bugs or added new features. This happened fairly infrequently, so EEPROM was ideally suited for this purpose. Programming the EEPROM took several seconds to several minutes, but this was not considered a time critical event. Additionally, programming and erasing the EEPROM required much higher voltages than the circuits that used the EEPROM, so external tools with higher voltages were used to program or erase the chip. There was a limit to the number of times a chip could be erased and programmed, ranging from several hundred up to one thousand. Production embedded devices rarely see more than several dozen program changes, so this limitation was not seen as a major issue.

The characteristics of EEPROM did not lend themselves well to storage outside of fairly static code. However, the technology evolved. Semiconductor process advances have allowed dramatic miniaturization of electronic circuits: a new chip was designed to encapsulate many EEPROM and programming circuits, at which point Flash was born. In flash, thousands of EEPROM circuits (which in Flash are called blocks[ii]) are arrayed in order to randomly program and erase blocks. Each block contains thousands of bytes of data subdivided into "pages" that can be read randomly within a block.

Section 3

Flash Chip Organization

Modern Flash chips are also divided into two “planes,” each of which has its own block program/erase circuit and pool of blocks. To increase the capacity of a Flash chip without dramatically increasing the cost, four silicon Flash dies are packaged together and share a set of pins coming out of the chip. The four dies are divided into two chip enable groups; each group operates like a separate chip with two dies. Figure 1 illustrates the internal organization of the components that make up a 2 GB SLC (Single Layer Chip) Flash chip.

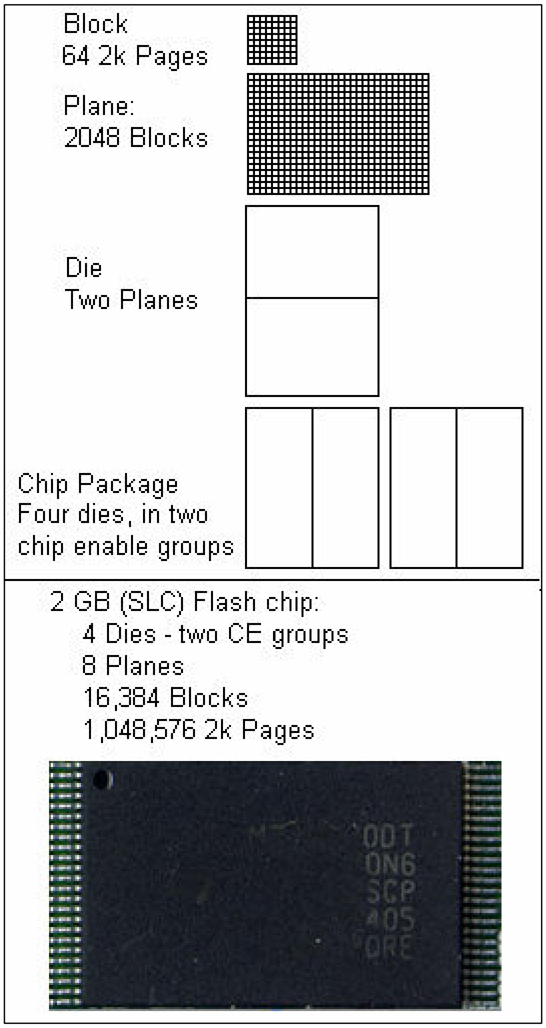

Figure 1 – Flash Chip Organization

This organization effects how read and write operations occur in a Flash chip. There are three main operations: read, program, and erase. Each set of dies in a chip enable group operate independently as long as the shared data pins are not busy. However, within a chip enable group, only one of these operations may take place on a plane at a time. Separate operations may take place on separate planes in parallel as long as they are submitted to the chip enable group at once[iii]. Each of these operations is described below:

Reads – Reads can occur randomly anywhere within a Flash chip. A read has an access time of <0.1 ms.

Erase – Erasing can only be performed on an entire block. It sets all of the bits in the block to 1. Blocks may be erased randomly within a chip. An erase operation takes approximately 1.5 ms. The erase operation stresses the block and will eventually cause it to fail. For SLC chips, 98% of the blocks will last at least 100,000 write/erase cycles. Systems like the RamSan-500 have elaborate wear leveling algorithms built in so that system write endurance is much higher than that of an individual cell (*see Section 9*).

Program – Programming can only occur on blocks that are erased since the program operation can only change a bit from a 1 to a 0. Pages within a block must be programmed sequentially; however, an entire block does not need to be programmed at once. Blocks within a chip can be programmed randomly. A program operation requires a minimum of approximately 0.3 ms.

Considering all of the characteristics of Flash, it is easy to see that it is dramatically different from RAM or hard disk drives and requires different logic for it to be used in an enterprise storage system. If the Flash storage was just treated like a traditional block device, one 8 KB random write operation would require the following actions (note that an individual Flash chip can transfer data at ~20 MB/s):

- Read 128 KB of data from the block to be written to so the 120KB of data that isn't changing will be preserved. This takes 0.1 ms plus the time to read 128 KB, which is 128 KB/ (20 MB/s) = 6.25 ms for a total of 6.4 ms.
- Erase the 128 KB block. This takes 1.5 ms.
- Program the 128 KB block with the 120 KB of unchanged data and the 8 KB of new data. This takes 0.3 ms plus the time to transmit 128 KB, 6.25 ms, for 6.7 ms total.

Adding all of the time together, a random write takes 14.6 ms, significantly longer than an 8 KB write takes in a mechanical disk drive[iv]. Since only one operation per plane at a time is permitted, these operations will block reads that need to occur. Flash drives targeted toward the consumer market as either thumb drives or laptop Flash drives accept this write performance. This is not a major concern in the consumer space because writes occur fairly infrequently and cost is the overriding concern. In the enterprise, however, write performance this slow is unacceptable and has been one of the main reasons Flash drives have not been deployed.

Section 4

MLC vs SLC

Up until this point, the discussions have concerned only SLC Flash. There is a second type of Flash called MLC (Multilayer Chip). The two types of Flash are very similar; in fact, their manufacturing processes are nearly the same. The faster more reliable type is SLC. The difference between the two is fairly straight forward; each holds a voltage to indicate a value stored in physically identical Flash cells. The difference is, SLC stores only two values, 1 or 0 (with either a high or low voltage level) and, MLC can store four values, (high, medium high, medium low, low) representing two bits per cell (00,01,10,11). MLC has a much smaller voltage tolerance than SLC and therefore, fails ten times sooner since each program erase cycle tends to increase the variance in the voltage that is stored. Although, MLC has a cost advantage of roughly two to one, as each cell holds two bits instead of one. This gives blocks, pages, and chips each twice the capacity, so operations take twice as long. The added delay and 10x failure rate make MLC Flash unsuitable for enterprise applications.

Section 5

Flash in the Enterprise

The Flash characteristics described in the previous sections can be summarized into a few key properties:

- Non-volatile
- Fast Random reads
- Fairly fast sequential writes
- Slow random writes
- 100,000 write endurance limit at the chip level (*see Section 9*)
- Blocking operations.

Other properties of Flash that have not been discussed in detail include the following:

- Low power
- Ruggedness
- Significantly denser than RAM
- Significantly lower cost per GB than RAM
- Significantly higher cost per GB than hard disks

Certainly, Flash offers some definite advantages as well as some pitfalls that must be addressed in order to provide a Flash-based storage system for the enterprise. Texas Memory Systems has delivered powerful RAM-based solid state disks (SSD) to the enterprise for years. Now we are leveraging our experience designing enterprise SSDs to bring the first Flash storage system to the enterprise market: the RamSan-500.

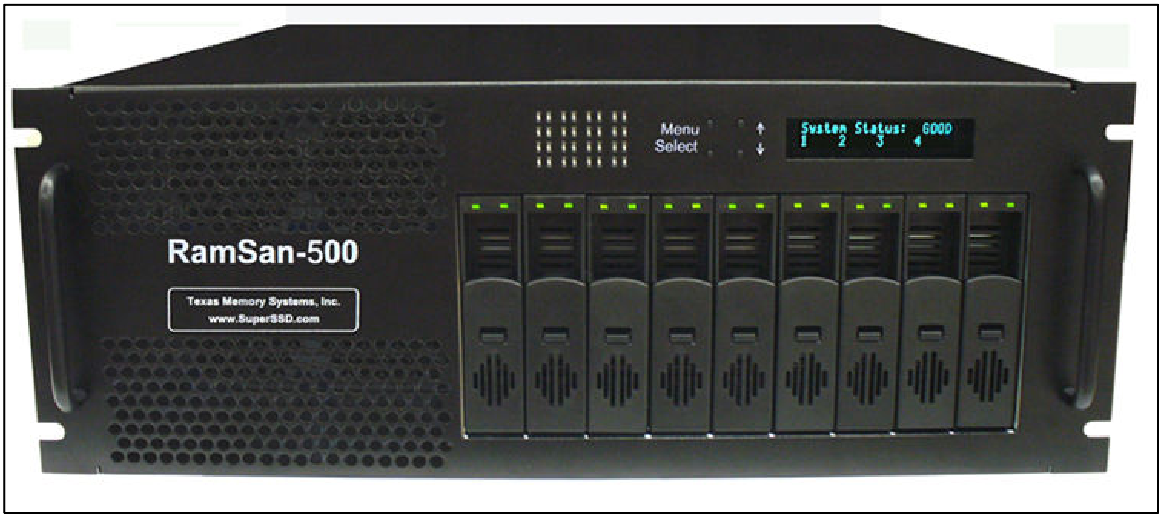

Figure 2 – RamSan-500

The RamSan-500 blends two different solid state technologies, RAM and Flash, to create the industry's first Cached Flash system. A loaded system consists of 2 TB of usable RAID protected Flash capacity and up to 64 GB of DDR RAM cache. The RamSan-500 takes advantage of this large DDR RAM cache to isolate the Flash memory from frequently written data and to allow for massive parallelization of accesses to the back end Flash. By using a large DDR cache, sophisticated controller, and a large number of Flash chips, the RamSan-500 is able to leverage the read performance, density, and sequential write performance of Flash.

Section 6

Read Performance

To effectively utilize the read performance of Flash, the cache design of the RamSan-500 is focused on minimizing the cache miss penalty as much as possible. This differs from most storage array cache designs and is one of the reasons the RamSan-500 is able to achieve extremely low latency. To avoid the costly decoding and encoding involved with storage protocols (Fibre Channel, SCSI, etc.) the RamSan-500 only performs the coding operation once at the host connection. Once data enters the RamSan-500, high speed buses move data throughout the system. Even the Flash modules for the RamSan-500 use a high-speed, high pin density connector to provide a high bandwidth raw bus access to the Flash chips within the module. This design shaves hundreds of microseconds of latency from the access time giving the RamSan-500 a random read cache miss time of 0.2 ms. To further increase the bandwidth from the Flash, nine chips are accessed in parallel, allowing the bandwidth (~20 MB/s) of eight chips to be used, one is used for parity to decrease the transmit time for an access, as illustrated below:

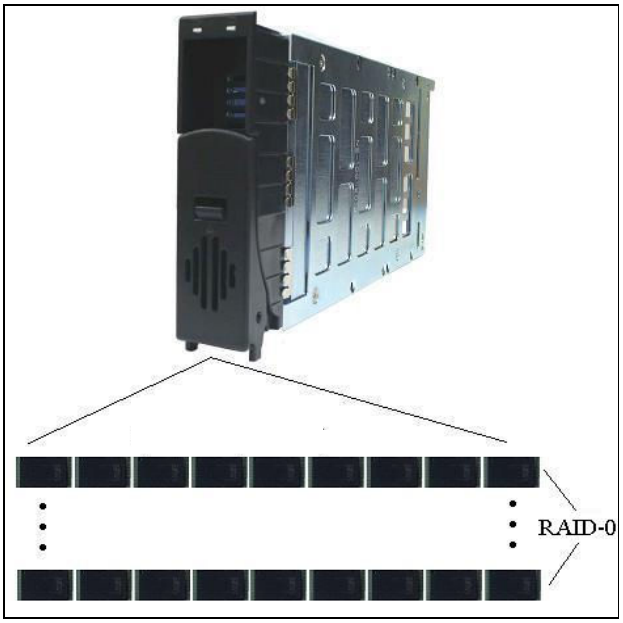

Figure 1 - Flash Module Construction

In addition, a RAID-5 layout is used across the Flash modules with a 16 KB stripe width to allow larger accesses, leveraging the bandwidth of even more chips. Each Flash module includes two Flash boards, each involved in a separate nine member RAID-5 group. The two RAID-5 sets are then striped across. The large, high bandwidth DDR cache allows operations to be broken into many smaller parallel operations to achieve a 2 GB/s sustained bandwidth.

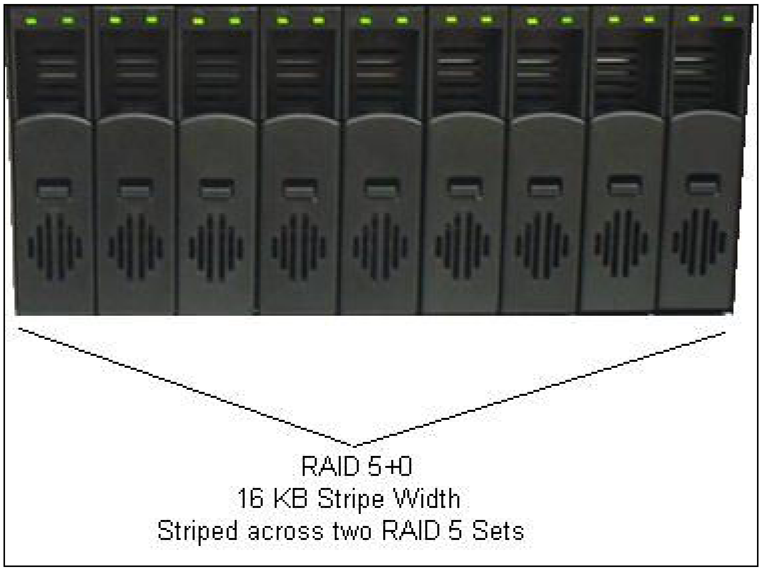

Figure 2 - RAID 5 Layout of the Flash Modules

Note that the stripe widths within the RamSan-500 are significantly narrower than in mechanical disk based systems. Disk systems use a large stripe width because disks perform so poorly at random accesses that it makes sense to sequentially grab a large amount of data each time a seek occurs. This is one reason that the RamSan-500 significantly out-performs Flash disks behind RAID controllers designed for hard disks.

Section 7

Write Performance

The RamSan-500, due to its design, achieves a much higher level of Flash write performance, finally making the technology enterprise-ready. The large battery backed DDR RAM cache is used to buffer write operations (random or sequential) so host processing can continue after only the cache write. The back end Flash controller takes the buffered writes and maps them to a new physical location every time they are written, ensuring that new writes are always written sequentially to locations in the Flash and thus avoiding the random write Flash penalty. Additionally, each Flash module contains 20% more capacity than is reported as usable. This additional capacity is used to accommodate remapping of blocks that fail within the module and to maintain a ready pool of pre-erased blocks. With this pool of extra blocks, the erase operation can be completed as a parallel background operation. The controller also employs a wear leveling algorithm to ensure that writes are evenly distributed amongst all blocks within the RamSan-500.

Section **8**

A Note About RAID-5 and Flash

It is worth noting that RAID-5 makes an ideal fit for deployment on Flash systems. RAID-5 distributes parity data amongst all of the members of a RAID set. For a read operation, just the data is fetched from the RAID, so only a single I/O occurs on the back end storage. For a write operation, however, the RAID controller reads the old data, reads the old parity, writes the new data, and finally writes the new parity. This results in four I/O operations and is the reason disk array users are specifically warned against RAID-5 deployments in write heavy environments. With Flash, four operations still occur, but the two read operations are so much faster than the writes that it effectively looks like only two write operations have occurred. This makes Flash RAID-5 operate nearly as fast as Flash RAID-10. Given the capacity advantages of RAID-5, it becomes a logical choice for Flash.

Section 9

Write Endurance

One of the primary hurdles impeding Flash memory adoption in the enterprise is the fact that Flash eventually wears out. The RamSan-500 completely mitigates this concern by incorporating both high capacity and wear leveling at the controller level. To illustrate, we can use the RamSan-500's specifications, to examine a "worst case" access pattern: 2 GB/s of sustained write activity.

The RamSan has 2 TB of usable capacity plus a 20% overhead of additional capacity for failed blocks, for a total space of 2.4 TB. The maximum write bandwidth is 2 GB/s. Running at 2 GB/s it will take 1228 seconds to write to all of the Flash blocks once. In order to reach the 100,000 write/erase cycle endurance limit, this process must be performed 100,000 times, requiring 122,800,000 seconds, which is 2,047,000 minutes, 34,100 hours,1421 days, or **3.9 years**. And this endurance level is achieved assuming absolutely no help from the large DDR RAM cache, and that none of the Flash blocks exceed their endurance specification.

Section 10

Conclusion

With the RamSan-500 entering the enterprise market, Flash will finally take its rightful place near the top of enterprise data storage solutions.

[i] EPROM was a *major* advance in semiconductors. Combined with general purpose microcontrollers, processors, or programmable gate arrays it allowed embedded programs to be created for various electronic devices without the multimillion dollar expense of fabricating a single purpose chip.

[ii] In particular this is referring to NAND Flash blocks. All of the cells in each of the block share some common areas of the storage transistor on the silicon. This is why some of the operations have to be performed on a per block basis; this also gives NAND flash a distinct density advantage that has led to its wide adoption.

[iii] This limitation is due to the shared pins and chip enable for all four planes in the Flash chip enable group. Once a command is sent to the chip enable group, it signals that it is busy until the operation is complete. So once a command is submitted it has to complete before the chip can do anything else. There are several commands, some that affect only a single plane and some that affect multiple planes.

[iv] To observe this first hand, interested readers should download the IO benchmarking utility IOmeter from www.iometer.org. With this utility, set up 8KB random writes to a USB Flash thumb drive target and also to a hard drive and observe the results. Limit the maximum disk size to 60,000 sectors (30 MB) to allow the test to complete without filling the entire disk drive.