INVITED PAPER

NAND Flash Memory and Its Role in Storage Architectures

Silicon semiconductor memory based on NAND logic is fast, non-volatile and very popular for temporary computer-storage applications such as caching for hard-disks and portable data storage.

By Marco A. A. Sanvido, Frank R. Chu, *Member IEEE*,
Anand Kulkarni, and Robert Selinger

ABSTRACT | This papers reviews the increasing role of NAND flash memory in storage architectures. NAND flash has enjoyed a phenomenal growth rate in storage capacities as well as a steady decline in pricing during the past few years. These developments have enabled NAND to enter and possibly change or displace some traditional storage architectures. However, besides cost, NAND flash memory has some reliability and performance issues that will slow its adoption into all storage architectures. This paper will analyze the advantages and disadvantages of this technology and the implications to the current and possible future storage architectures. Moreover, we will analyze NAND flash future density and costs trends and compare them with the traditional hard disk drive.

KEYWORDS | Disk drives; hard disks; memory architectures; semiconductor memories

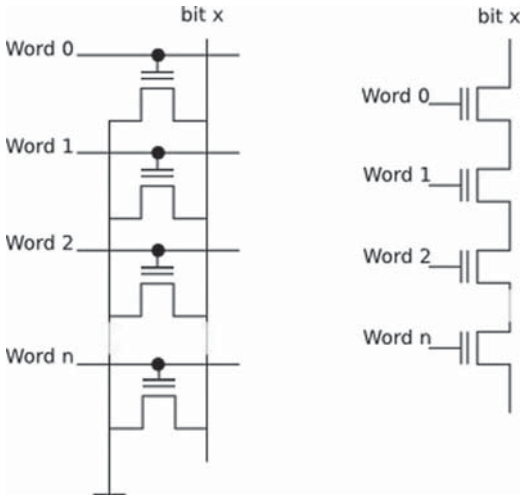

Fig. 1. *NOR and NAND cell design.*

I. INTRODUCTION

Today, there are two principal types of flash memory, NOR and NAND, that replace the earlier memories, erasable programmable read-only memory (EPROM) and electrically EPROM. Both types of memories support reads and writes, but writes must be preceded by an erase operation. Erasing a single byte is not possible; only a block of bytes can be erased.

NOR flash allows individual bytes to be read or written independently (assuming the memory was previously erased). It became quickly adopted into embedded systems as a replacement for EPROMs, for simplicity to use and program. Around 1990, NAND-type flash was introduced. NAND flash changed the interface of the flash device into a block access type: bytes could not be addressed independently. Only blocks of data (called a page) could be streamed in or out of the NAND flash. The cell design and interface allowed manufacturers to make NAND flash denser than NOR (the standard NOR cell is 2.5 times larger) and consequently reduce costs considerably. On the left-hand side of Fig. 1, the structure of a NOR cell for a single bit and multiple word lines is shown; on the right-hand side, the structure of a NAND cell for a single bit and multiple words is shown. The NAND bit cells are joint in a single bit line allowing a much denser structure.

Although the block interface limited its applications—for example, NAND could not be used to store ROM

Manuscript received October 29, 2007; revised May 9, 2008. Current version published December 2, 2008.
The authors are with Hitachi Global Storage Technologies, San Jose, CA 95135 USA.
Digital Object Identifier: 10.1109/JPROC.2008.2004319

code—it was quite well suited for pure storage applications due to its resemblance to a hard disk drive (HDD) device. This also coincided with the gaining popularity of digital cameras, which made NAND flash the preferred storage device alongside the micro-HDD drives (0.85 and 1.0 in), which, at that time, offered more storage capacity. After the digital camera, two other killer applications for NAND started to become popular: MP3 players and flash thumb drives (or USB flash drive), both of which accelerated even further the widespread use of NAND. In 2003, NAND flash shipments, in terms of megabytes shipped, surpassed NOR flash shipments. In 2005, NAND shipments exceed dynamic random access memory (DRAM) shipments (in gigabytes), and by 2007, NAND shipped more bytes in one year than the cumulative total of all DRAM bytes ever shipped [12].

Since its first incarnation, NAND technology improved not only by using smaller and smaller lithography but also by introducing innovations such as multilevel cell (MLC) [4]. MLC is in contrast to single-level cell (SLC) technology in that MLC allows multiple bits to be stored per cell. The advantages of MLC are clearly higher densities than the standard SLC, but this comes at a cost, as we will see later. In 2006, the first 2-bit MLC (2× MLC) devices started to appear on the market, and 3× and 4× MLC devices are on supplier's roadmaps. On the left-hand side of Fig. 2, the voltage distribution of an SLC NAND cell is shown. In order to read out the bit from the SLC cell, only a single comparison with a reference voltage is required, whereas in a 2× MLC cell—as depicted on the right-hand side—three threshold voltages are required. Given the reduced voltage margins, MLC is more likely to have errors, and therefore tighter error-correcting codes (ECCs) are required. For a 4× MLC, the threshold voltages required are 15, increasing even more the readout logic and increasing error probability.

In this paper, we will focus on the role of NAND flash in storage architectures. Throughout, the words flash and NAND will be used interchangeably to specifically denote NAND flash, since NOR storage is unlikely to be used as a mass storage medium (it may still appear in small amounts as code or metadata, but it will not be discussed further in this paper).

II. NAND FLASH TECHNOLOGY OVERVIEW

A. Fabrication Trends

The fundamental characteristic for fabrication of any semiconductor device is the minimum feature size (abbreviated F). DRAM used to be one of the leaders in adopting a newer lithography, and NAND was almost a generation behind. However, as shown in Fig. 3, NAND is now ahead of DRAM in adoption of smaller geometries.

Normal lithography progress would typically allow NAND to track Moore's law, but besides the accelerated adoption of finer lithography, the transition from SLC to 2× MLC technology has allowed for an additional increase in capacity. During 2006–2007, 2× MLC production went from about 20% of total NAND output (as measured by bits produced) to 80% (versus SLC). While the typical NAND cell is 4 F^2, because of the overhead circuitry, the actual *effective* cell area (total area divided by the number of cells) is about double this, or in the range of 7–10 F^2 for SLC chips. Two× MLC effectively doubles the bits per cell, so the "effective" cell area is about 4 F^2 per bit. The effect of these changes on cost/bit is discussed in Section V-A.

B. Architecture and Performance

NAND flash storage devices consist of an array of transistors with floating gates. Charges can be trapped in the floating gates to change the bias voltage at which the transistors turn on. This allows the transistors to be used as bits of nonvolatile memory. The flash array (Fig. 4) is arranged as independent blocks, where a block is the smallest erasable unit of storage. Erasing a block is an expensive operation and takes about 1.5–2 ms in an SLC device, and about 3 ms in a MLC device. Each block consists of a set of addressable pages, where SLC usually has 64 pages/block and MLC has 128 pages/block. Each SLC page is usually 2 KB in size, and a 2× MLC page is 4 KB in size. Each page is extended with additional spare bytes (SLC 64 bytes and MLC 218 bytes). These extra bytes are necessary to store the ECC and other metadata used, for example, for wear-leveling (see Section II-D).

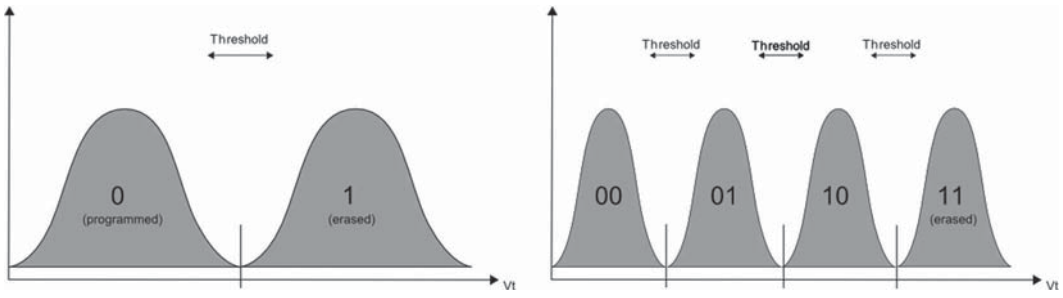

Fig. 2. *SLC versus MLC.*

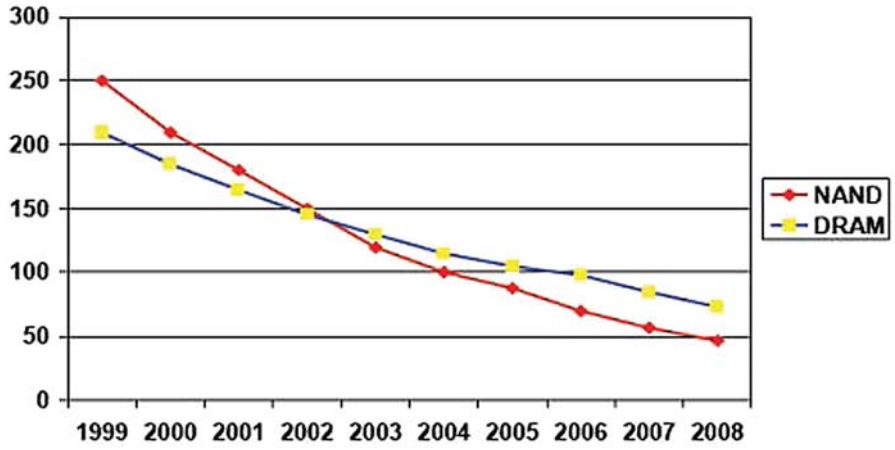

Fig. 3. ***Minimum feature size (in nanometers) [12].***

A page is the smallest programmable unit. In order to program a single page, the bytes are first sequentially written to a data page, which is set of volatile registers or SRAM. Once all data are written to the data page, the device is instructed to write the desired nonvolatile flash page with the content of the data page. Current devices take about 200–300 μs for SLC and about 600–900 μs for MLC. Therefore, we have a maximal write throughput of about 5.5–7.7 MB/s for SLC and 3.9–5.5 MB/s for MLC. This is only the theoretical maximal throughput since a page can only be programmed once after being erased. If we take into account also the erase time of the block, the sustained write throughput is reduced even further to 5.1–6.9 MB/s for SLC and to 3.8–5.4 MB/s for MLC. SLC flash devices support a limited number of operations in which a fraction of a page can be programmed at a time, known as partial page programming. This allows a 2 KB page to be programmed in separate operations (up to eight for some manufactures). Read operations on the flash array take place at the granularity of a page. The read operation is the fastest of all operations, taking just 25 μs for SLC and about 50 μs for MLC. Therefore the maximum read throughput is about 25 MB/s both for SLC and MLC—if we assume MLC page size (4 KB) to be the double that of SLC (2 KB) [4]. These are raw throughputs, achievable on a single NAND block without any architectural support.

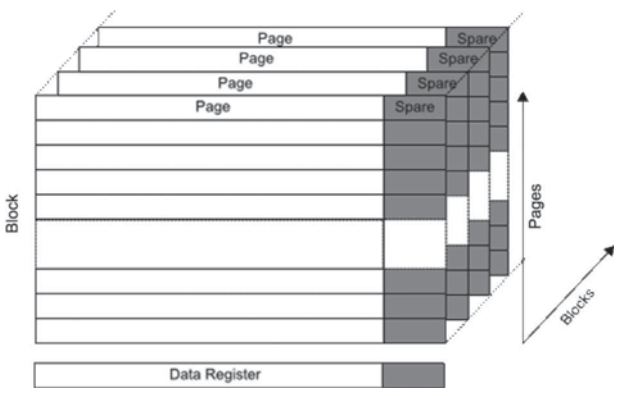

Fig. 4. ***NAND structure.***

Different optimization strategies are available in order to increase the read and, more importantly, the write throughput to NAND flash. The first common optimization is the support for cached read and write commands. In this case, once the NAND flash is programming or reading a page, we can start preparing the next data page in a second data page called the cached data page. By parallelizing the data transfer and the page programming/reading, we can increase the throughput [14]. Another approach to increase read and write throughput is to split the device into multiple planes, where access to each of the planes is interleaved. In both cases, the throughput is not doubled since we still have a single serial interface, which may limit the maximum data rate (for an 8-bit interface, this is about 40 MB/s; and for a less common 16-bit interface, this is about 76 MB/s). Consequently, some manufacturers opted to use a different interface in order to achieve higher throughputs. For example, Samsung—in its OneNAND products—uses a NOR-type interface and can claim throughputs of 100 MB/s (read) and 40 MB/s (write), respectively. Intel and Micron, on the other hand, using the ONFI standard (refer to Section II-C for further details) were able to claim 200 MB/s sustained read throughput and 100 MB/s sustained write throughput.

Efficient and reliable utilization of the raw flash devices in a system is predicated on supporting features such as speed-matching buffers, wear leveling, bad block management, and ECCs. These features are typically provided by a flash controller that may be a combination of hardware and software, incorporated either into the

flash device itself or into the system that manages the flash device.

C. Interfaces

The basic NAND flash devices connect to the external world with a low pin count interface. A single 8-bit or 16-bit bus is used to exchange command, address, and data.

Multiple cycles are use to communicate the command and address. The command indicates the intended type of operation like Read, Write, Erase, etc. The address points to a page and offset into the page of the requested data (if using partial page programming). The data phase is used to transfer the data from the host to the device on writes and from the device to the host on reads. The device indicates using a RDY (Ready) signal that the requested data, in case of reads, is available, or, in case of writes, that the programming of the page of data is completed. A status register must be checked to verify if the operation was successful.

This interface protocol is unique to raw NAND flash interface. There are also minor differences in the protocols between devices from different vendors. These range from differences in the set of commands that the device understands to variations in the timing characteristics specified by the vendor.

A group of flash vendors and users have started an effort to standardize the interface and characterization of NAND flash devices by forming an open NAND flash interface standard organization.[1] ONFI is planning various extensions to improve and standardize the interface used for NAND flash devices. ONFI 1.0 has been released, and version 2.0 plans to use a protocol similar to that used for double data-rate DRAMs to support higher data transfer rates. A new command set has also been proposed that would allow host systems to access data using a form of block addressing. ONFI is also creating a specification for a standardized connector for NAND flash modules so that motherboards can accept a variety of modules from different vendors or expand the capacity of onboard NAND flash storage similar to DRAM modules for PCs. ONFI and JEDEC also started a collaboration and plan to release a joint standard based on the original ONFI specification.

Not all flash manufacturers are participating in the ONFI efforts but decided to create an alternative—and similar—specification, namely, LBA-NAND. The primary goal of LBA-NAND is to specify a logical block addressing standard for NAND flash memory, which will simplify the host side interaction to the flash memory. The complexity of physical block addressing, ECC, and wear-leveling is therefore implemented on the memory controller side.

Another standard group is creating a specification for a common software interface, called nonvolatile (NV) memory host controller interface (NVMHCI)[2] for software layers to access NV memory available on the system platform. This will provide a standard way for host software to detect and configure any nonvolatile flash devices in the system.

D. Reliability and Failure Mechanisms

The increased NAND flash density comes at a price: speed and reliability. We discussed the throughput issues in Section II-B, and in this section we will focus on the reliability aspects. NAND flash is fundamentally affected by four different reliability issues.

- *Wearout:* The insulator that gives the floating gate its ability to trap charge is degraded by each erase/program cycle, and the cell slowly loses its capacity for storing electron charges. For SLC, this is specified to be 100 K cycles while for 2× MLC this is typically 10 K cycles, but it is projected to be as low as 1 K cycles for 4× MLC.
- *Retention:* Over time, the floating gate loses the charges trapped, and consequently it loses the information stored in it. This is specified to be ten years for devices currently on the market, but may be less depending upon the number of actual read/write cycles [12].
- *Write disturb:* The writing of a page requires high voltages to be applied to the cells in a page. These high voltages might affect neighboring cells and overwrite those cells, and if this happens, the error can remain undetected. The adjacent pages have to rely on the ECC in order to detect and correct this write induced error.
- *Read disturb:* The reading process might also affect the read cells because the floating gate might lose the trapped charge after multiple read operations, effectively erasing the stored information if the cell is read too often. This error is typically recoverable, since the page can be reused (after being erased).

One of the fundamental differences between NAND and NOR is that NAND devices are shipped with bad blocks. The manufacturer guarantees a minimum number of defect free blocks, but the location of the bad blocks has to be discovered at run time by examining a marker in the spare page of each block. Moreover, during normal operation, more blocks can go bad and become unusable for future write operations. This is discovered at run time whenever an erase or write operation is unsuccessful.

All these reliability issues have to be addressed at a higher level in the storage architecture in order to achieve acceptable system reliability. The most critical reliability issue is the wearout problem. This is even more dramatic

[1] www.onfi.org.

[2] http://www.nvmhci.org.

in the MLC case since the devices are guaranteed only for 1–10 K erase-program cycles. The layer that is responsible for handling the wear issue is often referred to as the wear-leveling layer [17]. The goal of wear-leveling is essentially to distribute all page writes uniformly across the device. This requires an abstraction layer that translates between physical page addresses and logical page addresses, in order to allow the redirection of page accesses in the device. This is often referred as the flash translation layer [9].

Wear-leveling algorithms come in two variants: dynamic and static. Static wear-leveling algorithms choose new pages for write operations only from a pool of freely available pages, and the one with the lowest number of erases is selected for programming. The static approach is adequate for systems where most of the data are changing. Dynamic wear-leveling searches all of the pages (both free and with valid data) and moves the content of the selected page to another page in such a way as to reduce the discrepancy of write cycles between pages, including the static pages [6].

ECC is necessary to protect from write disturbs and less importantly read disturbs. For SLC, the recommended strength of the ECC is quite relaxed: 2-bit detection and 1-bit correction. But for MLC, the recommended ECC requirement is higher: it goes from 4-bit detection and 2-bit correction for 2× MLC up to 10-bit detection and 8-bit correction for 3× and 4× MLC flash devices [16].

All of these reliability requirements impose a price in the throughput and access latency, especially for MLC that has a lower reliability at a physical level.

E. Environmental Factors

Apart from high density and relatively low entry costs, NAND flash devices gain significant value from their low power consumption. Being primarily complementary metal–oxide–semiconductor based switching devices, NAND flash devices consume very little power when idle and even during operation. Typical devices may, for instance, draw as little as 40 μA when idle [8].

As with any other solid-state device, NAND flash devices have very robust characteristics with regards to response to shock and vibrations. Also, they can be operated over a wide range of ambient temperatures.

III. DISK CACHING ARCHITECTURES

An HDD is a complex electromechanical nonvolatile storage device. Its steadily increasing storage capacity has been serving the storage industry very well for more than 50 years, and it will continue to serve the ever expanding storage needs. During the last 50 years, hard disk drive technologies have been constantly improving in areal density, access time, media data transfer rate, and host interface rate. However, with the rapid advancement in CPU technology, the mechanical latency of the hard disk drive is a limiting factor on the overall system performance.

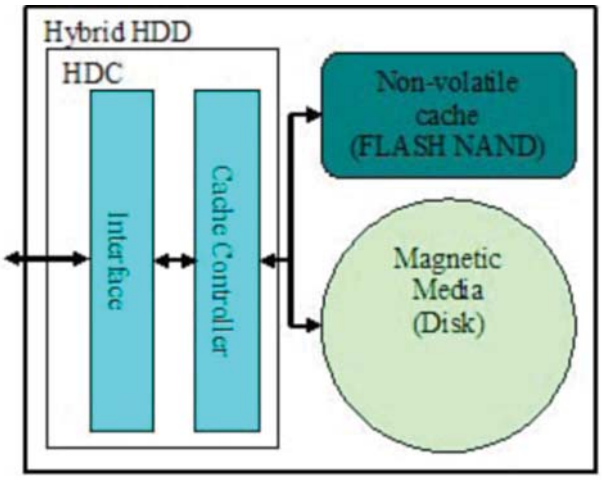

Fig. 5. ***Hybrid hard disk drive structure.***

In the meantime, the rapid advancement of nonvolatile NAND flash memory technology is making it possible to use NAND flash memory as an added disk cache layer to compensate for the disk drive's mechanical latency and as a buffer for CPU data access requirements. However, there are several alternative architectures described below on where and how to add the NAND.

A. Hybrid HDD

The concept of a hybrid HDD was initially introduced by Microsoft at its 2004 Windows Hardware Engineering Conference [15], initially known as the Piton drive. The basic idea is to add nonvolatile NAND flash memory as a disk cache to the regular hard disk drive. An industry trade association known as the Hybrid Storage Alliance[3] has been created to promote the adoption of hybrid HDDs and has many major HDD manufacturers as members.

An hybrid HDD can offer the following improvements:

- provide a faster boot up and improve resume time from standby or hibernation;
- reduce power consumption by rotating media, thus increasing battery life for mobile users;
- improve launch time for frequently used applications;
- improve the hard disk drive reliability, since the media can be spun down most of the time.

Fig. 5 shows the block diagram of a hybrid HDD; the only addition is the NAND flash memory on the hard disk drive. From the software point of view, there is firmware on the HDD, and a companion driver is necessary on the host side to manage the NAND flash memory.

1) Hybrid Commands: The NV Cache command was adopted in the ATA8-ACS specification [1]. It enables the

[3]www.hybridstorage.org.

host to provide both the configuration information and the management control on the NV flash memory in the hybrid HDD. There are two new important commands: the NV Cache command set and NV Cache Power Mode command.

The purpose of the NV Cache feature is to create an intermediate permanent storage, using the NAND flash memory, between the host adopter and the rotating HDD media. The NV Cache command set will enable the host to effectively manage the flash memory. Two important NV Cache commands are NV Cache Pinned Set, which allocate the regions of the NV Cache to be completely managed by the host, and NV Cache Unpinned Set, which allocate the regions of NV Cache that is managed by the disk drive.

The NV Cache Power Mode command is to allow the host to manage the rotating media's power state. If the host issues a NV Cache Power Mode command, the hybrid HDD should attempt to complete read, write, and other operations while keeping the rotational media idle (not spinning), to save power.

2) How Does the Hybrid HDD Work?: The host can use the NV Cache Pinned set command to permanently pin the following types of data in the NV flash memory, so the host can access them faster without the penalty of the rotating media's latency:

- portion of boot data that can be accessed by the host during the beginning of boot process while rotating media is spinning up—typical spin-up time is around 5 s;
- initial portion of the hibernation file, so that during the resume from hibernation, the host can access some data while the rotating media is still spinning up;
- frequently used random data, to reduce effective latency of random disk access;
- specific application programs, to enable certain frequently used application programs to be run without having to spin up the rotating media.

The host can use the NV Cache Unpinned Set Command to allocate a portion of NV flash memory to be managed by the hard disk drive. The unpinned NV flash memory is necessary if the host had just issued the NV Cache Power Mode to place rotating media to idle, so that any incoming write data will be stored into the unpinned flash memory. But if the available unpinned flash memory is nearly full or if there are read data that are not available in the NV flash, or system DRAM read cache, then the disk drive can spin up to flush the unpinned flash memory onto the rotating media or retrieve the missing read data.

For faster booting, the boot data can be retrieved from the NV Cache Pinned set while the rotating media is getting ready. After the rotating media is ready, the remaining boot data will come from the disk media. Similarly, for quicker resume from hibernation, during the resume operation, the host can retrieve hibernation file from the NV Cache Pinned set while the rotating media is getting ready. Once the disk is ready, the remaining hibernation file data can come from the disk media. In both cases, the NV Cache Pinned set can allow the host to access the data while the rotating media is spinning up, which enables a faster startup or resume time.

For quicker application launch, the host can pin key commonly used random sectors in the NV Cache Pinned set in advance. Then during application launch, these sectors are retrieved from the NV Cache Pinned set, which gives a quicker response time.

During normal operations, the host can issue the NV Cache Power Mode to idle the rotating media. Then all of the incoming write data can be stored in the NV unpinned flash memory to save power, which is especially important for the mobile user. Another benefit of keeping the rotating media idle is to improve the reliability of the disk drive. Given that the disk drive can handle much higher shock and vibration while the drive is not operational, the overall reliability of the disk drive can be improved by keeping the disk drive idle for as long as possible.

Another major advantage is that the hybrid HDD integrates both rotating media and NAND flash memory, and therefore the data are not split into two separate physical units that could be removed independently (which might cause data corruption).

Microsoft introduced support for hybrid HDD in its Vista operating system under the name ReadyDrive.[4] The ReadyDrive technology uses the NV Cache command set to manage the NV cache on the hybrid HDD.

B. External Caching

Another disk cache architecture uses a nonvolatile memory placed outside of the HDD. In order to use this type of caching, software and drivers are required to manage the nonvolatile cache and the disk drive.

This architecture was proposed by Intel at the Taiwan Intel Developer Forum, October 2005, and called the Robson cache. The solution places the NV NAND flash memory as either a PCI-e minicard or embedded on the PC mother board. The Robson cache was officially announced as "Turbo Memory" in 2007 [5].

1) How Does it External Caching Work?: From the block diagram in Fig. 6, the external cache is connected to the IO controller.

The new HDD driver interpret the NV Cache commands issued by the operating system and manages the interactions of NV flash memory (via the FLASH Driver) and the hard disk drive (via the Standard HDD Driver).

This architecture also offers the benefits of a faster boot and faster resume experience to the user. Similarly, during normal operation, the rotating media is mostly idle; thus it

[4]http://www.microsoft.com/whdc/system/sysper/perfaccel.doc.

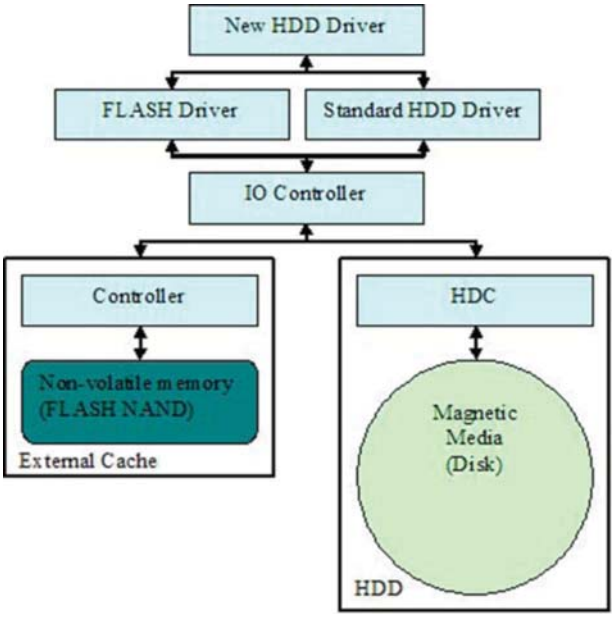

Fig. 6. ***External cache architecture.***

can offer benefits of lower overall power consumption and more rotating media reliability.

This disk cache architecture offers some advantages but also comes with some disadvantages, such as splitting the critical user data into two physical locations: the external data cache and the stand-alone HDD. Another problem is the need of dedicated software drivers to operate the external cache. However, in the future, this problem may be resolved by NVMHCI.

C. Non Volatile Memory Hardware Controller Interface

As the above explanation showed, there are a variety of ways for NAND flash to be used in a system. However, the proliferation of different hardware architectures and different software usage models for different operating systems creates a growing need for standardization of the middle layers of the driver stack. The formation of the NVMHCI Standard group is one such effort. Once such a standard is fully developed and implemented, the need for proprietary drivers to support different NAND architectures is reduced or eliminated.

IV. SOLID-STATE DRIVES

A solid-state drive (SSD) is another form of nonvolatile disk drive that is emerging as a replacement to the rotating hard disk drive. An SSD uses 100% NAND flash memory as the storage element. Because an SSD appears to the system just like an HDD, there is no need for any new software driver. The initial SSDs use 100% SLC NAND, making the current cost much higher than either HDDs or hybrid HDDs. In the near future, MLC NAND will likely replace SLC in the low-performance SSDs due to its lower cost, but MLC can reduce the endurance by a factor of ten and performance by a factor of at least two.

SSDs do not suffer the mechanical seek and rotational latency problems of HDD and accessing random sectors is basically limited only by the read latency of a NAND flash page read. Thus, SSD random reads operations are 10–50 times faster than the traditional HDDs. For sequential reads, HDDs have a much higher media transfer rate and are able to achieve ~100 MB/s.

However, the intrinsic characteristics of NAND may make them slower than HDDs on random and sequential write performance. The longer write latency and the slower block erase times are affecting the random write performance and making them four to ten times slower than traditional HDDs.

The combination of these different workload factors (faster on random reads, slower on writes) means that most of the current SSDs targeting the notebook market are not vastly faster than HDDs. SSDs can overcome many of these performance problems by increasing parallelism and adding a write cache. A few SSD models have incorporated these features and show significantly better performance but at a very significantly higher cost.

The overall long-term reliability of SSDs remain an open issue. On the one hand, since they are solid-state devices, the hardware is potentially more robust than a mechanical HDD. However, since the number of erase cycles for flash is limited, the total number of writes that an SSD can sustain is also limited, which may be a problem for some workloads. It is also too soon to say whether the tradeoffs between the number of erase cycles and the data retention period are satisfactory. And while some wear leveling software has been used in prior products and ECC has been used in many product areas, the overall SSD firmware is still relatively young. Also note that these problems may still be issues for the cache approaches described in Section III. but many are not as severe. For example, if the flash wears out, the cache and performance acceleration go away, but the HDD still maintains a full copy of the data.

V. COMPARISONS

A. Cost (Price)

There are many ways to measure the cost of a product. The product *cost* at the initial manufacturer includes the cost of the components, labor, capital depreciation, etc. They typically add a gross margin to cover their R&D efforts and corporate overhead and allow for a profit, which results in the sales *price* to their customers. Then there may be one or more levels of integration and resale before the product gets to the end customer.

Based on supply and demand, the markup at each stage may increase or decrease over time. Both historical and

projected prices at the end user are likely to have the greatest fluctuations over time, since the final price magnifies any shortage or oversupply conditions at possibly several levels of integration or resale. Thus, we will base our comparisons on price at the lower levels of integration (not the end user price).

In addition to where cost is measured, there are also several ways in which costs can be compared. First, for some product categories, the "entry" cost or minimum product cost is the most significant factor, such as portable consumer electronics products that are typically sold for a few hundred dollars. At the other extreme, many products are sold on the basis of their cost per capacity, typically in dollars per gigabyte ($/GB).

1) HDDs: The entry cost of an HDD includes a number of relatively fixed components (electronics, motor, actuator, enclosure) and a variable number of heads and disks (which also influence the cost of assembly and yield). The fixed component costs vary according to the form factor, the performance, and the reliability expectations for the product. The cost difference between 3.5 and 2.5 in is becoming insignificant, although smaller than 2.5 in form factors may have a higher cost due to lower volume. Obviously, the cheapest entry product is one with one disk platter and one head. Due to continuously higher unit volumes and ongoing efforts to reduce costs, the entry cost of an HDD [to an Original Equipment Manufacturer (OEM)] is around $50 and has been declining at 2–6% per year.

The cost per capacity is directly related to the number of surfaces used and the areal density. Low capacity drives (i.e. 1 head, 1 disk) may have the lowest entry cost, but they have the highest $/GB. High capacity drives can average the fixed costs over a larger capacity and have the lowest $/GB. The areal density progress from year to year has ranged from 25–100% increase, which corresponds to a cost decrease per gigabyte of 20–50% per year. Due to differences in form factor, capacity, performance, and reliability, the cost per gigabyte for different HDD models can vary by more than 10× (see Section V-A3).

2) "Raw" Flash Bits: The manufacturing cost of a flash chip is related to the cost to process a wafer divided by the number of good dies per wafer. A significant part of the process cost relates to the capital equipment, which is driven by the lithography (which also influences the chip area). The capital costs have been increasing but are being offset by the conversion to larger wafers (200–300 mm) and faster processing times (more output per time). Thus, the overall processing cost per square centimeter of chip has been decreasing by a few percent per year.

Shrinkage in the minimum feature size allows for a greater number of bits per chip. Lithography changes occur at about two-year intervals, with other optimizations (and larger die sizes) allowing for increased capacities in the intervening years. But as discussed in Section II-A, the NAND flash industry moved from being behind the leading edge to being on the leading edge, which produced a faster rate of change the last several years than can be sustained. The other factor that has reduced the cost per bit has been the adoption of MLC technology, which increases the capacity per area. While MLC in theory can produce 2, 3, or 4× the number of bits per cell, the differences in the actual cell size, yield, and margin have kept it from being 1/*n*th the cost per bit. Historically, the 2× MLC price versus SLC price has been from 50% to 100% of the cost per bit, driven as much by market conditions as underlying cost.

The overall effect of all of the above improvements has decreased the cost for NAND memory by nearly 50% per year over the last five years (including possibly some reduction of gross margin). However, the transition to larger wafers, catching up on lithography, and adoption of 2× MLC are more one-time events that may not reoccur. Thus, NAND is likely to follow the normal evolution of lithography, cost reductions, and design improvements, with cost decreases for the next five years expected to be more in the range of 30–40% per year [7].

3) Comparing the Price of HDDs and NAND Flash: As mentioned above, HDD prices at a given time can vary by more than 10× in $/GB, while NAND tends to be more homogeneous (at least within a factor of two). Fig. 7 below shows the projected price trends for each.

NAND prices declined faster than HDDs from 2002 to 2007, declining at a rate of 5–15% per year faster than HDD. However, for the next several years, many analysts are forecasting roughly similar price declines for both, due to either slowing NAND price decreases or faster HDD price decreases due to greater adoption of perpendicular recording, with the $/GB difference between flash and the higher capacity HDDs remaining close to 100 times.

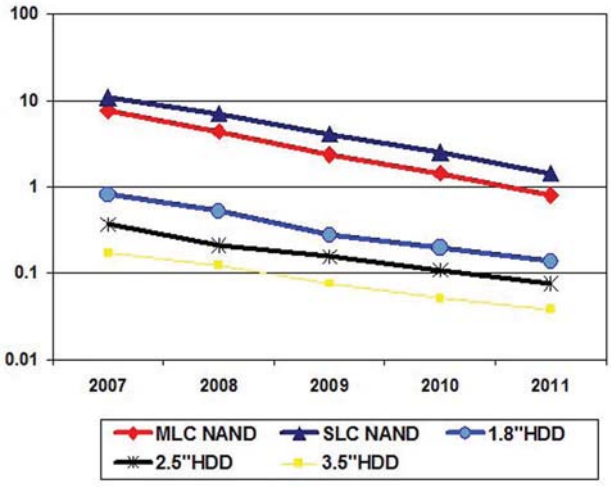

Fig. 7. ***Dollar per gigabyte (source: [18]).***

4) "Integrated" Flash Products: The previous sections discussed the basic costs for HDDs and flash chips. The cost for a hybrid HDD is the cost of an HDD plus the cost of the flash, with a small amount for testing and yield (the extra firmware may or may not increase the cost of the product). In the case of turbo memory, the added cost is for a small printed circuit board and a simple controller, and is likely to be only a few dollars, or an insignificant change to either the entry cost or the $/GB.

Depending on the sophistication and complexity of the controller, an SSD may have a small or large cost increase over the raw flash cost. In 2007, a high-end SSD controller may add $500–$1000 of cost, while the notebook oriented 2.5 in SSD may only add $20–$30 for a controller. In both cases, as volumes increases and designs are reduced to smaller amounts of silicon, these costs will decrease further.

B. Performance

As discussed above, the raw chip-level performance of flash technology can be attractive, especially on a random read workload. The generally inferior data rates of flash versus HDD (especially on writes) can also be mitigated by parallelism.

While there are some high-performance (and high-priced) SSD products, the current performance of most of the volume SSD products, hybrid HDDs, and turbo memory results are relatively indistinguishable from each other. They typically report about 10–15 s improvements versus HDDs in booting Microsoft Vista, and can run some benchmarks faster than HDDs (also with the DRAM cache sometimes turned off) [2].

Since the performance of all of these devices is constantly improving, there are various independent Web sites that provide current performance tests.[5]

All three of these technologies are expected to improve, both in hardware data paths and, more importantly, in firmware algorithms. The benchmarking environment will need some additional specification and standardization in order to report comparable results. Finally, tradeoffs with DRAM caches should be considered. DRAM is higher performance but volatile and more expensive than flash, and it has been used and tuned for many of these applications. It may be more economic in the long term to reduce the amount of DRAM in the system and substitute more flash (in any one of the above architectures) once the flash algorithms and performance have improved.

C. What To Use Where?

In this section, we look at how the above flash technologies may be adopted by various market segments, since the requirements and conclusions are different in each segment. However, the capacity requirement is usually the initial criteria in each market.

The use of mass storage in the consumer electronics market divides fairly easily into two segments: a low-capacity portable market and a higher capacity market. While HDDs were able to capture a significant share of the low-capacity market with 1 in HDDs for a few years, flash (including 3× or 4× MLC someday) is likely to continue to capture the majority of this market in the future primarily due to entry cost advantages (for example, you can add $10 of flash, but the cheapest HDD may be $50). On the other hand, HDDs will continue to rule the higher capacity market, with flash having little impact for at least the next five to ten years, since $/GB is the primary requirement and the performance requirements can be easily satisfied by HDDs.

The outlook for flash in the PC market (both notebook and desktop) is more complicated, but the first distinction remains the capacity requirement. For most individual PC users and many corporate PCs, the increasing use of photos, music, and videos will continue to drive capacity requirements higher (to hundreds of gigabytes). For this segment, either the HDD or hybrid HDD is likely to prevail as the mass storage, since this is a $/GB market and the cost premium for flash would be very large.

However, there is a small but growing market segment among both notebook and desktop users, where higher capacities may not be required. For example, some corporate and government IT departments are restricting the amount of capacity per user to reduce "personal content" and/or reduce their total cost of ownership (TCO, which includes the cost of backups). In these scenarios and with thin clients, users may only keep a "cache" of frequently used data and programs, with the rest residing on corporate servers. In this case, an SSD or even embedded flash or turbo memory with additional software may suffice. For the low-capacity storage user, the cost of, say, a 32-GB SSD may not be a significant deterrent and users may also factor in the power or shock and vibration advantages of a pure flash solution.

The second issue for many PCs is the performance *opportunity* that may come from using flash technology. As stated above, the three alternatives (hybrid HDDs, SSDs, and turbo memory) are not strongly differentiated today in performance from each other (or even HDDs in many cases), although their performances will likely improve. For the high-capacity storage user, this leaves them with the choice between a hybrid HDD or a conventional HDD plus turbo memory.

Use of flash in the server environment is more speculative, since it is divided into more niches with less homogenous workloads than PCs. Briefly, there are some very high-performance segments that could use the very high-performance (and higher priced) SSDs, as well as low-capacity entry segments (e.g., the boot drive for blade servers). But most server environments contain terabytes of capacity, so significant usage of flash is not affordable in

[5]See, for example, http://www.anandtech.com/storage/; http://www.engadget.com; http://www.notebookreview.com; and http://www.tomshardware.com/storage/index.html.

the foreseeable future. NAND flash may be used in place of some of the DRAM caches used in storage subsystems, since it has a lower cost per bit and is nonvolatile without relying on batteries. However, most server data does not have the same locality as PC programs and user data, so caching approaches (e.g., hybrid HDDs) may not be as useful in server workloads. And if the caching is successful, then the higher I/O rates may create greater problems of wearout for flash.

VI. CONCLUSIONS AND FUTURE TECHNOLOGIES

NAND flash has entered the storage arena and will change future storage architectures The continued improvements and cost reduction of NAND will likely increase its penetration into more storage applications, but the inherent limitations of flash will slow down the penetration in markets where low $/GB, faster write input/output operations per second, and proven reliability are the key factors.

It is also doubtful that the previous rate of cost reduction can be sustained by NAND flash since it was achieved with several "one-time events." Assuming normal lithography trends will allow them continued cost reductions but not as fast as the past. To keep the same compound annual growth rate, the new NAND technologies such as 3× and 4× MLC, etc., need to demonstrate their feasibility and shown to be scalable to the ever smaller lithography, along with several other "inventions."

There are also a few other nonvolatile technologies appearing on the horizon that may overcome some of the inherent NAND flash limitations, such as the write performance and endurance, and may be able to scale to smaller lithographies, such as phase change memory; and companies such as Intel are already demonstrating a 128 Mb PCM device. Spin-transfer torque RAM is another technology that hopes to overcome NAND's limitations.

These and other technologies [10] could impact the future role of NAND flash, but they all must be reduced to high volume manufacturing at constantly lower prices, so their impact is uncertain or many years away. ■

REFERENCES

[1] *Non Volatile Cache Command Proposal for ATA8-ACS*, Feb. 2006, Doc. e05106r6. [Online]. Available: www.t13.org

[2] D. Baral, Samsung, "Implementation of hybrid HDDs," *Flash Memory Summit*, Aug. 2007.

[3] J. Burke, "HDD and SSD: coexistence in a growing opportunity," *Flash Memory Summit*, Aug. 2007.

[4] J. Cook, *NAND Flash Technology*, Denali Webcast, Feb. 2007.

[5] R. Coulson, "Mobile platform non-volatile memory (Robson)—Benefits and implementation options," *Intel Beijing IDF*, 2007.

[6] E. Gal and S. Toledo, "Algorithms and data structures for flash memories," *ACM Comput. Surv.*, vol. 37, no. 2, Jun. 2005.

[7] E. Harari, "Keynote address," *Flash Memory Summit*, Aug. 2007.

[8] Hynix Datasheet HY27UH08AG5M. [Online]. Available: www.hynix.com

[9] Intel, *Understanding the Flash Translation Layer (FTL) Specification*, Dec. 1998. [Online]. Available: www.intel.com

[10] G. Komoto, "An introduction to emerging memory technologies," *Flash Memory Summit*, Aug. 2007.

[11] L. Mason, *The Immense Promise of MLC-Flash for Solid State Storage*, Denali Webcast, Feb. 2007.

[12] L. Mason, *NAND Flash Market Overview and Update*, Denali MemCon, Jul. 2007.

[13] K. Grimsrud, *New Flash Standards Drive New Platform Value*, Denali Memcon, 2007.

[14] Micron. (2005, Dec.). *Increasing NAND Flash Performance*, TN-29-01. [Online]. Available: www.micron.com

[15] C. Nicholson, "Improved disk driver power consumption using solid state non-volatile memory," *Windows Hardware Engineering Conf. (WinHEC)*, May 2004.

[16] R. Pierce, *Hardware and Software Solutions for MLC NAND System Design*, Denali Webcast, Feb. 2007.

[17] SanDisk. (2003, Oct.). *Flash Memory Cards Wearleveling*. [Online]. Available: www.sandisk.com

[18] J. Unsworth, Gartner, *Dataquest Insight: Solid-State Drives Emerge in Select Consumer and Enterprise Markets*, Jan. 7, 2008.

ABOUT THE AUTHORS

Marco A. A. Sanvido received the Dipl.-Ing. degree and the Dr.Techn. degree in computer science from the Swiss Federal Institute of Technology, Zürich (ETHZ), in 1996 and 2002, respectively.

He is a Researcher with Hitachi Global Storage Technologies. He was a Cofounder of weControl, an ETHZ spinoff, where he developed low-power and real-time embedded systems for autonomous flying vehicles. He was a Postdoctoral Researcher in computer science at the University of California at Berkeley from 2002 to 2004, and thereafter he worked on virtualization at VMware. He currently is working on a wide range of storage architecture projects ranging from security and hybrid-HDD to novel storage architectures.

Frank R. Chu (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Washington, Seattle, the M.Sc. degree in electrical engineering from Stanford University, Stanford, CA, and the M.B.A. degree from Santa Clara University, Santa Clara, CA, in 1981.

He is a Senior Engineer with Hitachi Global Storage Technologies, San Jose Research Center, San Jose, CA. He joined the Hitachi Global Storage Technologies Research team from the IBM Research team, where he had worked since 1981 after spending 11 years as a Research and Development Engineer with Hewlett-Packard Corporation, Palo Alto, CA. He is a member of the Storage Architecture group, concentrating on power-management technologies, such as the Enhanced Adaptive Battery Life Extender (ABLE) 3. He is a Hitachi Global Storage Technologies representative to the SATA-IO Workgroup and a member of the SATA-IO Board of Directors as well as other standards organizations, such as ONFI and NVMHCI. Additionally, he has been active in the development of magnetic recording products. He is a corecipient of three U.S. patents and a coinventor of numerous U.S. patent applications.

Mr. Chu is a member of Tau Beta Pi.

Anand Kulkarni received the bachelor's degree in electrical engineering from the Indian Institute of Technology, Bombay, and the master's degree in electronics design and technology from the Indian Institute of Science, Bangalore.

In 1996, he joined the Enterprise HDD firmware development team, IBM. In 2005, he transferred to the Hitachi GST Research team as part of the Storage Architecture group. He is presently a Senior Engineer with Pliant Technology, Inc., Milpitas, CA, which designs and develops flash-based solid-state drives for enterprise applications.

Robert Selinger received the S.B. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, and the Ph.D. degree in computer science from the University of California at Berkeley.

Throughout his career, he has worked on creating advanced technology and bringing it to market. Early in his career, he was with IBM Research working on relational database performance, graphics, and storage subsystems and interfaces. He then was with MIPS Computer working on I/O systems and midrange servers. He was the Leader of Advanced Technology at Adaptec for seven years in various management capacities, where he led the development of several new storage interface standards, including extensions to parallel SCSI, 1394, SSA, and Fibre Channel. He was also active in bringing PCI RAID and ATM to market and led the performance analysis activities. In the early 2000s, he was a Consultant and interim CTO at several companies in the storage and communication industries. He joined Hitachi Global Storage Technologies in 2003 to work on methods of integrating multiple disk drives into subsystems, and later built a prototype flash storage subsystem. He recently joined SanDisk, where he serves as Chief Architect for the SSD Business Unit.