Notes of Solid-State Drive

Chapter 1 Introduction

As modern computing systems (e.g., enterprise servers, data center storage, and consumer devices) process a large amount of data at an unprecedented scale, a storage device needs to meet high requirements on storage capacity and I/O performance. While electromechanical disk drives have continuously ramped in capacity, the rotating-storage technology does not provide the access time or transfer-rate performance required in demanding enterprise applications, including online transaction processing, data mining, and cloud computing. Client applications are also in need of an alternative to electromechanical disk drives that can deliver faster response times, use less power, and fit in smaller mobile form factors.

A NAND flash-based solid-state drive (SSD) can provide orders-of-magnitude higher I/O performance (i.e., access time or response time) compared to traditional hard-disk drives (HDDs), (Also, resilient to physical shock, a small form factor, consuming less static power) with a much lower cost per bit value over SSDs based on emerging non-volatile memory (NVM) technologies. Moreover, SSD capacity is now at the point that the drives can serve as rotating-storage replacements. As a result, NAND flash memory has become the de facto standard for architecting a storage device in modern computing systems.

NAND flash memory has several unique characteristics, such as the erase-before-write property (i.e., a flash cell needs to be first erased before programming it), limited lifetime (i.e., a cell cannot reliably store data after experiencing a certain number of program/erase (P/E) cycles), and large operation units (e.g., modern NAND flash memory typically reads/writes data in a page (e.g., 16 KiB) granularity). To achieve high performance and large capacity of the storage system while hiding the unique characteristics of NAND flash memory, it is critical to design efficient SSD firmware, commonly called Flash-Translation Layer (FTL). An FTL is responsible for many critical management tasks, such as address translation, garbage collection, wear leveling, and I/O scheduling, which significantly affect the performance, reliability, and lifetime of the SSD.

Modern solid-state drives can be abstracted into three levels: (1) NAND Flash physics, (2) integrated circuit architecture, and (3) SSD firmware, as shown in Figure 1.*1*. At the lowest level of abstraction is the NAND Flash physics level, which describes the motion of electrons for the basic operations (i.e., program, read, and erase) for a single NAND Flash memory cell. Above the physics level is engineering level: the integrated circuit architecture and SSD firmware levels. (mark) .

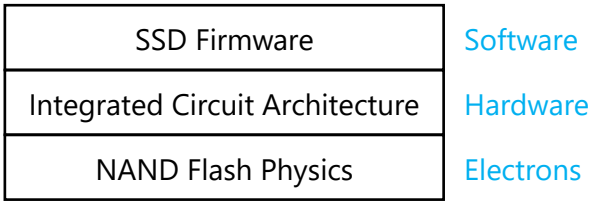


Figure 1.1 Abstraction Levels of Modern Solid-State Drive

In all SSDs, a Flash microcontroller sits between one or multiple hosts (i.e., CPUs) and NAND Flash memories, and on each side, there are a lot of challenges that designers need to overcome.

Moreover, a single controller can have multiple cores, with all the complexity associated with developing a multi-threaded firmware. (mark) As usual, simulation speed and precision do not go hand in hand, so it is important to understand when to simulate what.

We will first dive into the lowest level, NAND Flash physics, to understand the characteristics of NAND Flash since its unique properties decide the upper levels' structures.

Note the NAND Flash's unique properties are including:
- Large operation units
- Erase-before-write property
- Asymmetry in operation units
- Limited endurance
- Various error sources
- Asymmetry in operation latencies

(retention loss, schematic, wear leveling->for endurance, Error Correction Code->for reliability, soft decoding, randomization, read retry)

Chapter 2 NAND Flash Physics

2.1. NAND Flash Cell

NAND Flash cell is based on the Floating Gate (FG) technology, whose cross-section is shown in Figure 2.*1*. This MOS transistor is built with two overlapping gates rather than a single one: the first one is completely surrounded by oxide, while the second one is contacted to form the gate terminal. The isolated gate constitutes an excellent "trap" for electrons, which guarantees charge retention for years. The operations performed to inject and remove electrons from the isolated gate are called program and erase, respectively. These operations modify the threshold voltage V_{TH} of the memory cell, which is a special type of MOS transistor. Applying a fixed voltage to cell's terminals, it is then possible to discriminate two storage levels: when the gate voltage is higher than the cell's V_{TH}, the cell is ON ("1"), otherwise it is OFF ("0").

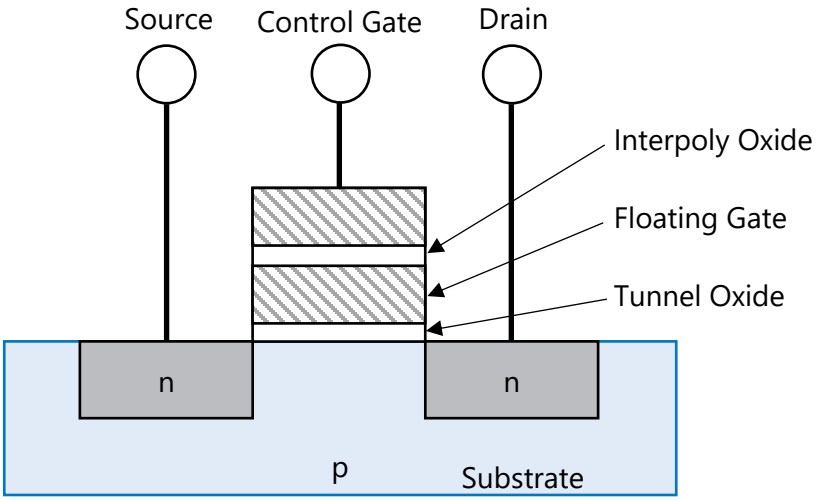


Figure 2.1 Floating Gate Memory Cell

It is worth mentioning that, due to floating gate scalability reasons, charge trap memories are gaining more and more attention, together with their 3D evolution.

NAND Flash memories

A floating-gate transistor constitutes a flash memory cell. It can encode one or more bits of digital data, which is represented by the level of charge stored inside the transistor's floating gate. The transistor traps charge within its floating gate, which dictates the threshold voltage level at which the transistor turns on.

A floating-gate transistor provides nonvolatile memory storage (i.e., the data stored in NAND flash is correctly retained even when the power is disconnected).

(charge trap transistors for 3D)

(retention loss, wear out effect)

It is worth mentioning that, depending on how the memory cells are organized in the memory array, it is possible to distinguish between NAND and NOR Flash memories. In this book we focus on NAND memories as they are one of the basic elements of SSDs.

Program of NAND flash cells:
During the program operation, the cells share the high programming voltage on the selected wordline but the program operation has to be bit selective. Therefore, a high channel potential is needed to reduce the voltage drop across the tunneling dielectric and prevents the electrons tunneling from the channel to the floating gate as indicated.

Note programming a page cannot change '0' cells to '1' cells. Thus, NAND flash cells have erase-before-write property.

Erase of NAND flash cells:
The erase operation resets the information of all the cells belonging to one block simultaneously.

Because the granularity of erase operation is usually larger the program operation (block > page), the in-place write on a page is very inefficient. As a result, SSD does out-of-place write and conducts garbage collection to recover free blocks.

NAND flash memory errors
NAND flash memory errors can be induced by a variety of sources, including flash cell wearout, disturb effects (errors introduced during programming, interference from operations performed on adjacent cells), and data retention issues due to charge leakage. (three major sources)

Disturb effects alter the memory transistors threshold voltage unintentionally during memory access operations under the influence of specific disturb conditions. Since it is essential for an efficient area consumption to arrange the storage transistors in a contact saving way, the sharing of voltage nodes can not be avoided. During the read and program operations, positive voltages are applied to the gate nodes of the memory transistors, wherein the channel is at a lower potential or even grounded. Therefore this condition is called gate disturb (also wordline disturb) and it is the most common disturb mechanism in NAND Flash memory arrays.

(Program disturb and read disturb)
Read disturbs are the most frequent source of disturbs in NAND architectures. This kind of disturb may occur when reading many times the same cell without any erase operation. All the cells belonging to the same string of the cell to be read must be driven in a ON state, independently of their stored charge. The relatively high V_{pass} bias applied on the control gate and the sequence of V_{pass} pulses applied during successive read operation may trigger the Stress Induced Leakage Current (SILC) effects in some cells that, therefore, may gain charge. Note read disturbs do not provoke permanent oxide damages: if erased and then reprogrammed, the correct charge content will be present within the floating gate.

Pass disturb is similar to the read disturbs and affects cells belonging to the same string of a cell to be programmed.

The Program disturbs, on the contrary, affect cells that are not to be programmed (inhibit) and belong to the same wordline of those that are to be programmed. In that case the program disturb is strongly related to the voltages and pulse sequences used for the self-boosting techniques. Although the program inhibit boosts the channel potential, soft programming can not be avoided especially when a high number of program pulses are applied.

The criticality of an effective program operation limiting program disturbs and/or possible successive errors is attested by the fact that in NAND memories the program operation should follow a precise and well defined "hierarchy": it is necessary to start from the cell nearest to the source selector and proceed along the string up to the cell nearest to the drain selector. This procedure is important, because the threshold voltage of a cell depends on the state of the cells placed between the considered cell and the source contact (the background pattern dependency phenomenon); the series resistance of the cells is different if they are programmed or erased.

(further) When manufacturing process scales down to a smaller technology node (i.e., the size of each flash memory cell), the amount of charge that can be trapped within the floating gate also decreases, which exacerbates reliability issues.

Multi-bit per cell storage (mutil-level):
The flash memory cell can encode one or more bits of digital data, which is represented by the level of charge stored inside the transistor's floating gate. Earlier NAND flash chips stored a single bit of data in each cell (i.e., a single floating-gate transistor), which was referred to as single-level cell (SLC) NAND flash. Multi-level cell (MLC) NAND flash stores 2-bit value (00, 01, 10, and 11). Triple-level cell (TLC) flash stores 3-bit value. Quadruple-level cell (QLC) flash stores 4-bit value. The benefits of multi-bit per cell storage is the additional capacity of the SSD without increasing the chip size, while it also decreases reliability by making the cells more difficult to correctly store and read the bits.

NAND Flash Memory Organization:
The flash memory is spread across multiple flash chips (typical values: 4, 16 chips), where each chip contains one or more flash *dies*, which are individual pieces of silicon wafer that are connected together to the pins of the chip. Each chip is connected to one or more physical memory channels, and these memory channels are not shared across chips. A flash die operates independently of other flash dies, and contains between one and four *planes*. Each plane contains hundreds to thousands of flash *blocks*. Each block is a 2D array that contains hundreds of rows of flash cells (typically 256-1024 rows) where the rows store contiguous pieces of data.

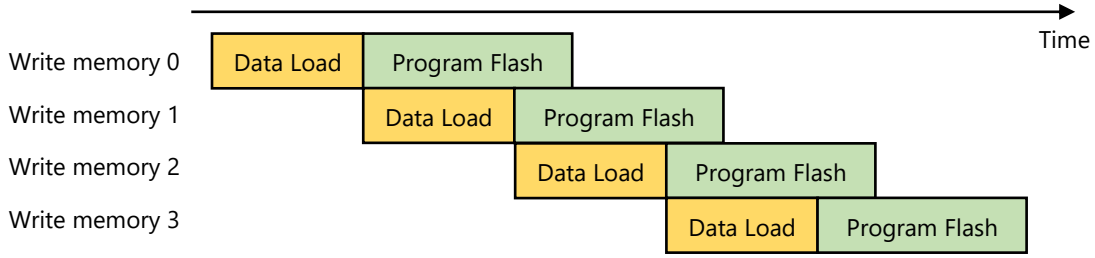
In NAND Flash memories, a logical page is the smallest addressable unit for reading and writing; a logical block is the smallest erasable unit.

If we consider the SLC case with interleaved architecture, even cells belong to the even page (BLe), while odd pages belong to the odd page (BLo). For example, a SLC device with 4 KiB page has a WL of 32,768 + 32,768 = 65,536 cells. In the MLC case we have MSB and LSB pages on even BL, and MSB and LSB pages on odd BL.

Each page is made up by main area (data) and spare area as shown in Fig. 1.5. Main area can be 4, 8 or 16 KiB. Spare area can be used for ECC (Error Correction Code) and is in the order of hundred of Bytes every 4 KiB of main area.

The **planes** can execute flash operations in parallel, but the planes within a die share a single set of data and control buses. Hence, an operation can be started in a different plane in the same die in a pipelined manner, every cycle.

Channel is the data bus (typical width: 8-bit) for connect different memories to the SSD controller (or NAND controller inside). Operations on a channel can be interleaved, which means that a second chip can be addressed while the first one is still busy. For instance, a sequence of multiple write operations can be directed to a channel, addressing different NANDs, as shown in Fig. 1.13: in this way, the channel utilization is maximized by pipelining the data load phase; in fact, while the program operation takes place within a memory chip, the corresponding Flash channel is free.



Data in a block is written at the unit of a *page*, which is typically between 8 and 16 KiB in size in NAND flash memory. All read and write operations are performed at the granularity of a page. Each block typically contains hundreds of pages.

Flash cards, USB keys and Solid State Drives are definitely the most known examples of electronic systems based on non-volatile memories.

Chapter 3 Integrated Circuit Architecture

SSDs are the prevalent application for NAND. An SSD is a complete, small system where every component is soldered on a PCB and is independently packaged.

The basic structure of a solid-state drive is shown in Figure 3.*1*. In addition to Flash memories and an SSD controller (a microcontroller), there are usually other components. For instance, an external DC-DC converter can be added in order to derive the internal power supply, or a quartz can be used for a better clock precision. Of course, reasonable filter capacitors are inserted for stabilizing the power supply. It is also very common to have a temperature sensor for power management reasons. For data caching, a fast DDR memory is frequently added to the board: during a write access, the cache is used for storing data before transfer to the Flash.

NANDs are usually available both in TSOP (Thin small outline package) and BGA (Ball grid array) packages. In order to improve performances, NANDs are organized in different Flash channels

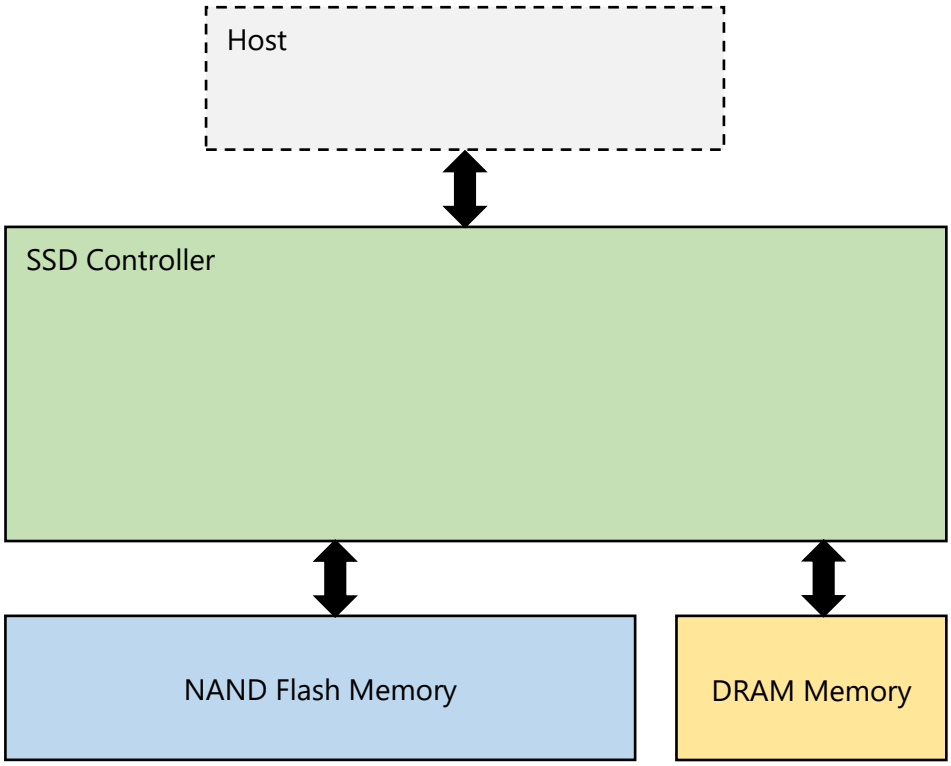


Figure 3.1 Hardware View of SSD system

For many applications the host interface to SSDs remains a bottleneck to performance. PCI Express (PCIe)-based SSDs together with flash-optimized host control interface standards address this interface bottleneck. SSDs with legacy storage interfaces are proving useful, and PCIe SSDs will

further increase performance and improve responsiveness by connecting directly to the host processor.

The SSD controller is responsible for scheduling the distributed accesses at the memory channels. And it uses dedicated engines (i.e., NAND controller) for the low-level communication protocol with the Flash.
SSD Controller: The SSD controller is responsible for (1) handling I/O requests received from the host, (2) ensuring data integrity and efficient storage, and (3) managing the underlying NAND flash memory.

Charge pumps are used to generate all the needed voltages within the chip

In multilevel storage, cell's gate biasing voltages need to be very accurate and voltage regulators become a must.

The Row Decoder is the block in charge of addressing and biasing each single wordline and it is located between the planes. Bitlines are connected to a sensing circuit. The purpose of sense amplifiers is to read the analog information stored in the memory cell.
The Row Decoder, also called Wordline Decoder or Wordline Driver.

Especially, SSDs call for a higher read and write throughputs; in other words, SSDs need to manage more NAND dies in parallel. Basically, there are a couple of options:
- The first one is to increase the number of dies per channel;
- The second option is to increase the number of channels.

Flash chip controllers (FCCs): A Flash chip controller is assigned to a flash memory channel for data and control connection.

DRAM: The on-board DRAM memory stores various controller metadata (e.g., how host memory addresses map to physical SSD addresses) and to cache relevant (e.g., frequently accessed) SSD pages.

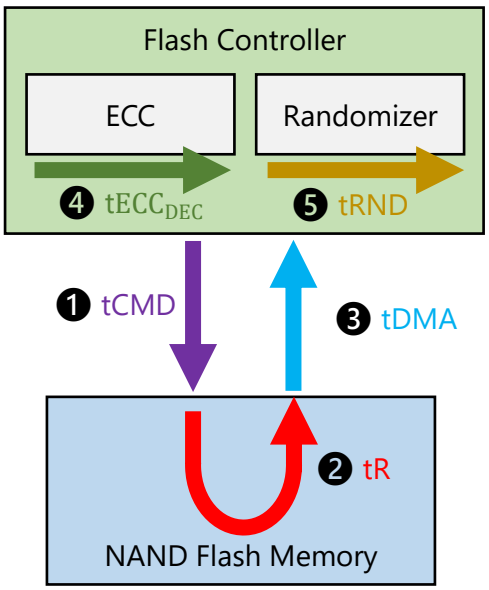
SSD Performance:
In the evaluation of SSD performance, there are three metrics and they are: (1) latency (or response time), (2) bandwidth, (3) throughput.

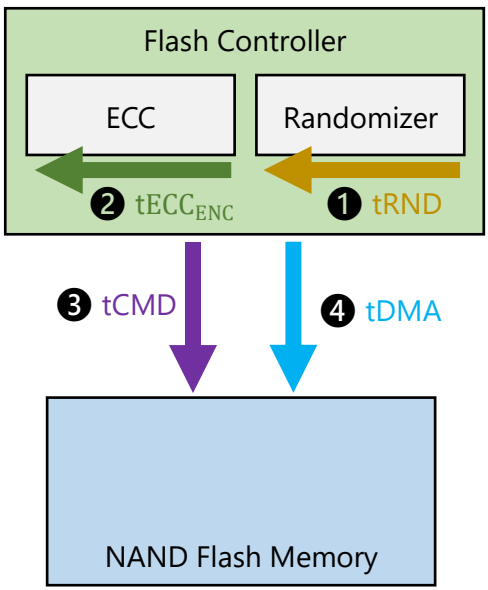
First two metrics are similar as mentioned before, latency is the time delay until the request is returned and bandwidth is the amount of data that can be accessed per unit time. Typical values are,
Latency:
Average read latency (4 KiB): 67 us

Average write latency (4 KiB): 47 us

Bandwidth:
Sequential read bandwidth: up to 3,500 MB/s
Sequential write bandwidth: up to 3,000 MB/s.

Throughput is the number of requests that can be serviced per unit time. SSDs define the measurement of the throughput as the IOPS: Input/output Operations Per Second. Typical values are,
Random read throughput: up to 500K IOPS
Random write throughput: up to 480K IOPS





SSD Firmware

Wear Leveling (endurance):
(insight) Usually, not all the information stored within the same memory location change with the same frequency: some data are often updated while others remain always the same for a very long time in the extreme case, for the whole life of the device.

(goal) In order to mitigate disturbs, it is important to keep the aging of each page/block as minimum and as uniform as possible: that is, the number of both read and program cycles applied to each page must be monitored.

(endurance definition) the maximum number of allowed program/erase cycles for a block (i.e. its endurance)

The controller firmware groups blocks with the same ID number across multiple chips and planes together into a *superblock*. Within each superblock, the pages with the same page number are considered a *superpage*. The controller opens one superblock (i.e., an empty superblock is selected for write operations) at a time, and typically writes data to the NAND flash memory one superpage at a time to improve sequential read/write performance and make error correction efficient, since some parity information is kept at superpage granularity. Having the ability to write to all of the pages in a superpage simultaneously, the SSD can fully exploit the internal parallelism offered by multiple planes/chips, which in turn maximizes write throughput.