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.

NAND flash memory is a type of non-volatile solid-state storage that persistently stores and retrieves data. It is non-volatile memory, as it retains data even when power is not applied, and has become the de facto standard for architecting storage devices in modern computing systems. A NAND flash-based solid-state drive (SSD) is a storage device that utilizes NAND flash memory to store user data. Unlike hard-disk drives (HDDs), SSDs have no mechanical moving parts, as shown in Figure 1.*1*. **NAND flash memory offers higher performance** (i.e., lower latency and higher bandwidth) compared to magnetic storage, which utilizes rigid and rapidly rotating platters and read/write heads. Additionally, its capacity has increased continuously, and its costs have decreased over the decades. Given these advantages, NAND flash-based SSDs can provide orders-of-magnitude higher I/O performance (i.e., lower read & write access time and higher random-access input/output operations) compared to traditional hard-disk drives (HDDs), (also, reliability because of 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. Thus, SSDs are now at the point where they can serve as replacements for rotating storage.

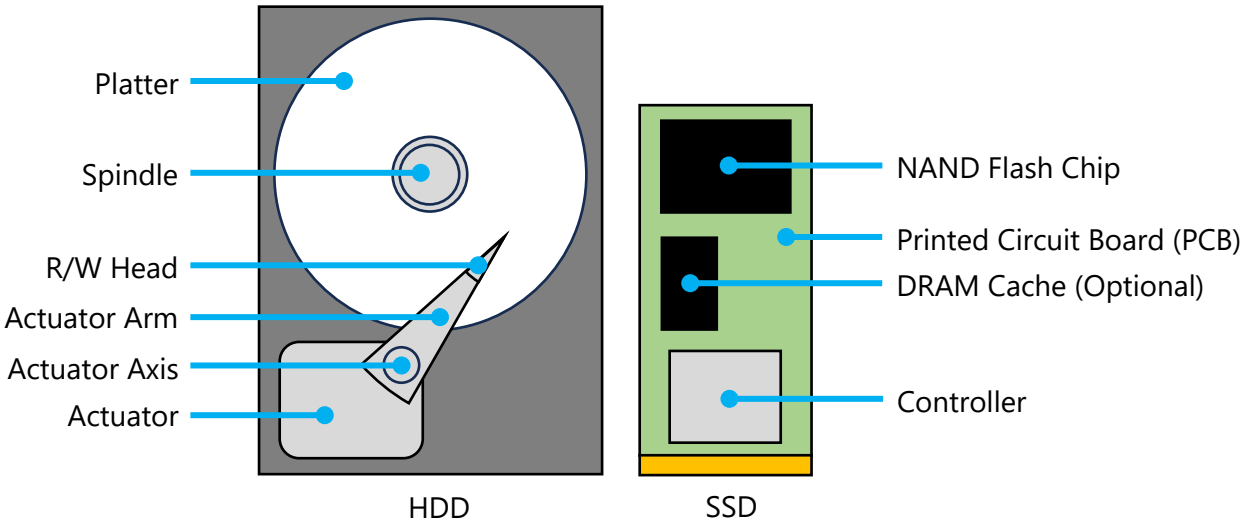


Figure 1.1 Overview of HDD and SSD.

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 1.2. 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) .

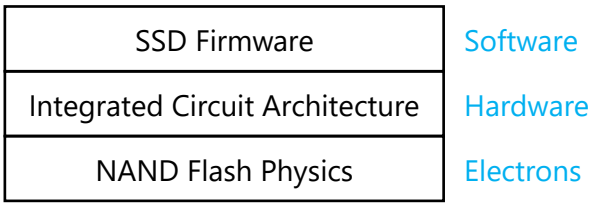


Figure 1.2 Abstraction Levels of Modern Solid-State Drive

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 Memories

Benefiting from two key trends: (1) effective process technology scaling and (2) multi-level (e.g., MLC, TLC) cell data coding, NAND flash memory becomes ubiquitous in everyday life today (e.g., flash cards, USB keys, and solid-state drives). Unfortunately, the reliability of raw data stored in flash memory has also become more challenging to ensure due to these two trends. Manufacturing process scaling, which has increased the number of flash memory cells within a fixed area, makes fewer electrons in the flash memory cell floating gate to represent the data. Multi-level cell data coding, which represents more than one bit of digital data in a single floating-gate transistor, results in larger cell-to-cell interference and disturbance effects. Without mitigation, worsening reliability can reduce the lifetime of NAND flash memory.

To develop mitigation mechanisms, we first need to understand the inside of NAND flash memory.

2.1. NAND Flash Cell

The basic building block of NAND flash memory is the NAND flash cell. NAND flash memory stores information (i.e., data) as the number of electrons, which is associated with the threshold voltage (Turn-on voltage) of each NAND flash cell. A NAND flash cell is a special type of nMOS transistor based on the Floating Gate (FG) technology, whose cross-section and symbol (i.e., diagram and schematic) are shown in *Figure 2.1*. This *floating-gate transistor* is built with two overlapping gates rather than a single one: the first one is contacted to form the gate terminal, while the second one is completely surrounded by oxide. On top of the isolated gate is an interpoly oxide layer, and at the bottom is a tunnel oxide layer. Oxide layers insulate the floating gate (storage layer), preventing electrons from leaking out of it. As a result, **this isolated gate constitutes an excellent "trap" for electrons**, which ideally do not discharge even when flash memory is powered off (i.e., without the connection of a supply voltage) over years, ensuring charge retention and guaranteeing years of operation.

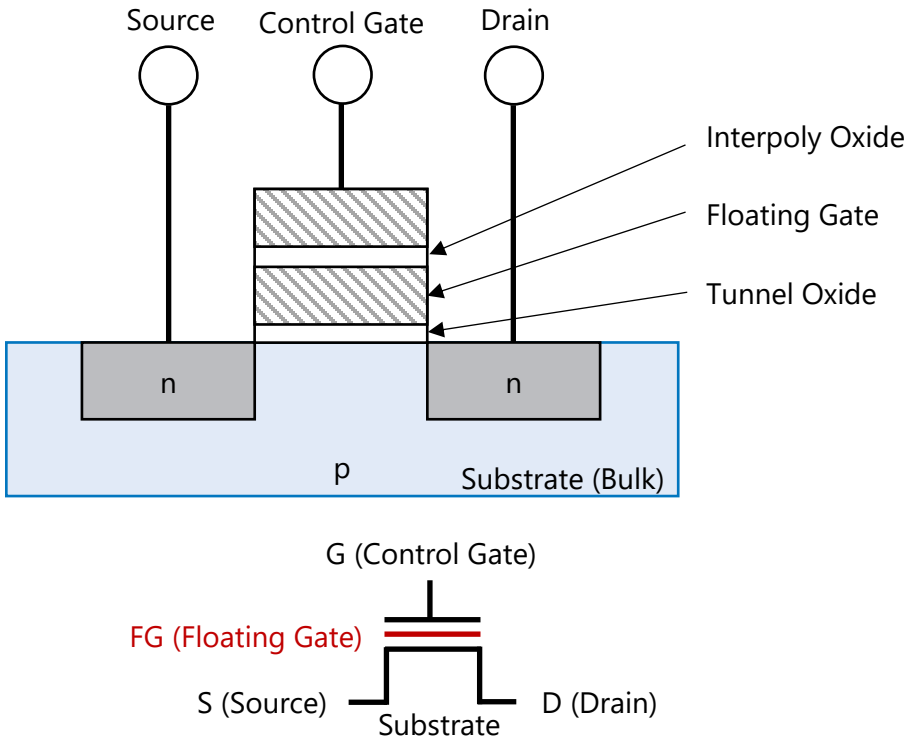
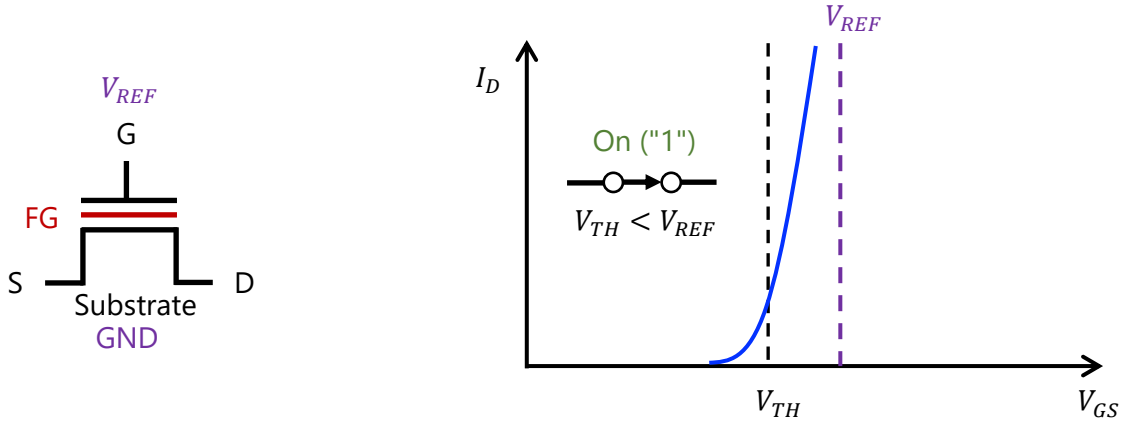


Figure 2.1 Diagram and Schematic of Floating Gate Memory Cell.

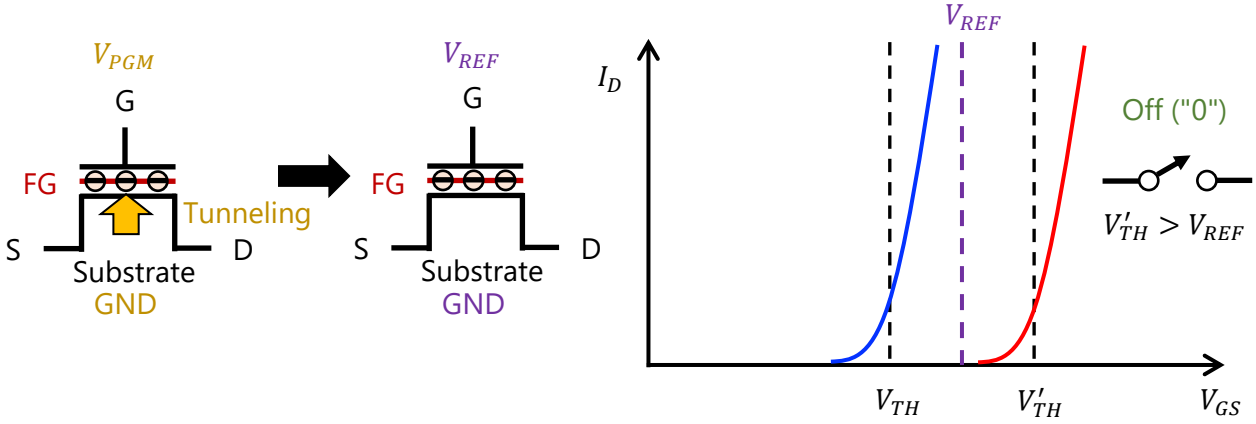
Program, Read and Erase of a Single Floating Gate Cell

For NAND flash cells, data are stored through program/erase operations and accessed via read operation. When we conduct the program, erase or read operations, electrons must be injected, removed, and counted from the isolated floating gate, respectively. The number of electrons stored in that gate is associated with the threshold voltage of the NAND flash cell. In *Figure 2.2*(a), given a fixed gate voltage (V_{REF}), the cell current (I_D) is a function of its threshold voltage (V_{TH}) because flash cells act like usual MOS transistors. Therefore, it is possible to understand the threshold voltage distribution to which the memory cell belongs through a current measure.

Program and erase operations modify the threshold voltage V_{TH} of the memory cell based on the *Fowler-Nordheim* (FN) *tunneling mechanism*. Earlier NAND flash chips stored a single bit of data in each cell, which was referred to as **single-level cell (SLC)** NAND flash. By applying a high program voltage (V_{PGM}, e.g., 8/18/26 V) to the cell's control gate and keeping the bulk terminal at ground potential (0 V), the **program operation** induces a large current transfer through the whole FG cell stack. This operation can set the transistor to a specific threshold voltage within a fixed range of voltages (*negative charge*), as shown in *Figure 2.2*(b). SLC NAND flash divided this fixed range into two voltage windows: one window represents the bit value 0 and the other represents the bit value 1, which we call the ER (erase) and P1 (program 1) states. The **erase operation** works principally in the same way but with control gate voltages negative with respect to the cell channel region (*positive charge*). Note that programming a cell cannot change the ER state to the P1 state. Thus, NAND flash cells have the erase-before-write property.



(**a**) Applying V_{REF} to Flash Cell with No Charge, and Its Current-Voltage (I-V) Characteristics



(**b**) Applying V_{PGM} to Charge Cell, and Its Current-Voltage (I-V) Characteristics

Figure 2.2 Read (**a**) and Program (**b**) Operation of SLC NAND Flash Cell.

A **read operation** is performed by comparing the cell's threshold voltage with a reference voltage V_{REF}. By applying a reference voltage V_{REF} to the cell's terminals, it allows discrimination between 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"). In other words, each state representing a different value is assigned to a voltage window within the range of all possible threshold voltages.

Cells containing n bit of information have $2n$ different levels of V_{TH}. **Multi-level cell (MLC)** NAND flash divides the flash voltage range into four voltage windows that represent each possible 2-bit value (00, 01, 10, and 11), which we call ER, P1, P2, and P3 states. Each voltage window in MLC NAND flash is therefore much smaller than a voltage window in SLC NAND flash. This makes it more difficult to identify the value stored in a cell. Moreover, **triple-level cell (TLC)** NAND flash further divides the range, providing eight voltage windows to represent a 3-bit value, which we call ER, P1-P7 states. And, Quadruple-level cell (QLC) NAND flash stores a 4-bit value per cell.

Due to the process variation (i.e., cell's structural characteristics), the threshold voltage of flash cells programmed or erased to the same state varies across the voltage window, as shown in

Figure 2.3. A program verify (PV) level confirms that a memory cell has been successfully programmed to its target state, preventing over-programming. An erase verify (EV) level ensures that the cell has been completely erased to its default low-voltage state, preventing over-erasing or leaving data in an intermediate, un-erased state. These verify levels are crucial for maintaining data reliability by ensuring cells do not become stuck in an intermediate state, preventing errors and data loss, particularly in Multi-Level Cell (MLC) devices.

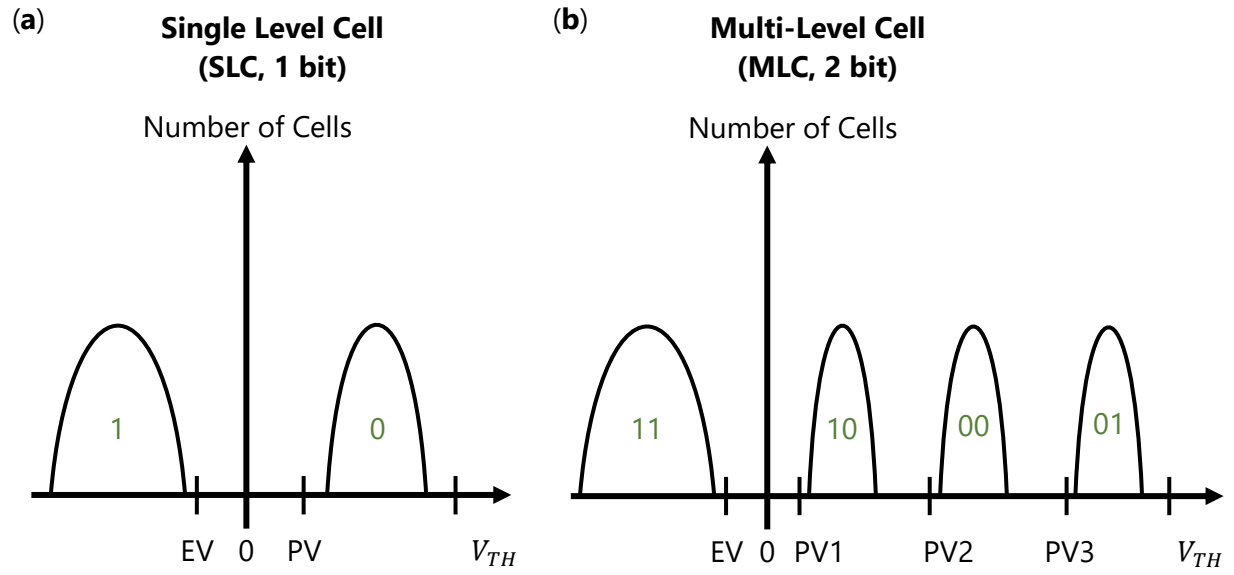


Figure 2.3 Memory Cell Threshold Voltage Distributions for Single Level Cells (**a**) and Multi-Level Cells (**b**).

Given a fixed program voltage, some cells might be successfully programmed to a specific V_{TH} state (easy-to-program cells) while others do not (hard-to-program cells). Thus, programming a floating gate cell to a specific V_{TH} state is typically accomplished by the so-called "incremental step pulse programming" (ISPP) scheme. ISPP applies several high program voltage pulses to the control gate. During each pulse, electrons move from the substrate to the floating gate. After every pulse, a read-verify is performed. This programming technique repeats its sequence multiple times until the verification passes on all programmed cells. For example, each word line is programmed 3 times, such that V_{TH} distributions can be progressively tightened.

Reliability of Floating Gate Cell

The reliability of FG NAND flash memory is one of the most important criteria, since typically 10 years of charge retention and 1-100 k program/erase cycles need to be guaranteed for a NAND flash product chip.

a) Charge Retention

Charge retention (or data retention) refers to the ability of a memory to retain stored information over time without any biases being applied. In *Figure 2.4*, a typical charge retention requirement is shown. $\Delta V_{TH,PGM}$ is the programmed threshold voltage shift, and UV stands for ultraviolet. UV erasure process for erasing data stored in NAND flash memory involves exposing the memory

chip to UV light, which causes the floating gate to return to a default state, effectively erasing the data. This process is distinct from the programming or writing of data, which is done electronically. It needs to be guaranteed that, for a successful read-out of the stored information, the programmed V_{TH} (above the PV level) will not decrease more than 10% over the product's relevant time period of 10 years. In principle, there are multiple leakage paths which can lead to a loss of the programmed floating gate electron charges. The electrons can be lost through the interpoly oxide towards the control gate, leak through the cell side wall oxide to the cell junction area, or through the tunnel oxide towards the substrate, which is the most severe charge loss.

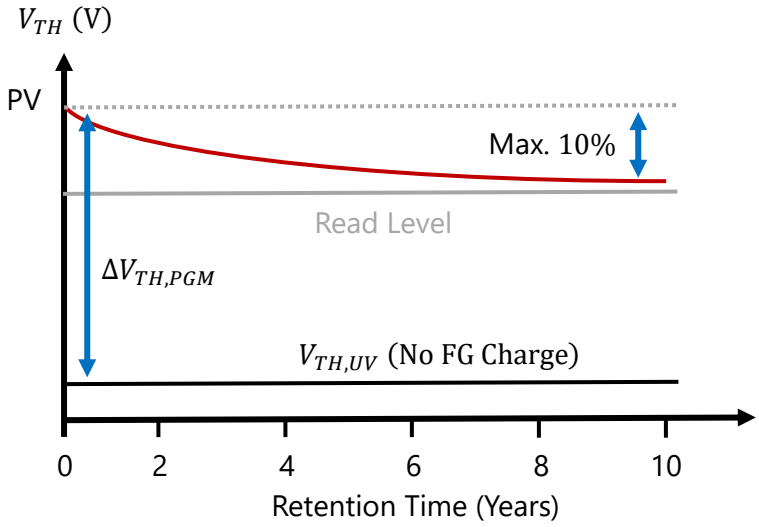


Figure 2.4 Charge Retention of an FG Cell. A Certain Amount of Charge Loss Needs To Be Tolerated (e.g., 10% V_{TH} Loss Over the Time Period of 10 Years). UV Stands for Ultraviolet.

We explain the electron leakage through tunnel oxide (TOX, refer to *Figure 2.1*) since it is the main factor in the wear of the FG cells. This type of electron leakage is not only because the TOX is physically the thinnest dielectric layer that holds the electrons on the floating gate but also because of the change in the density of traps in the tunnel oxide (or trap generation). The charge transfer during the program and erase operation generates electric states in the TOX, called **oxide traps** (the TOX should be the only dielectric where the electron charge is transferred). These traps are broken bonds of the atoms in the oxide matrix due to the electron tunneling processes. The density of traps in the tunnel oxide consequently increases with the number of program/erase cycles, which causes so-called **oxide stress**. The traps in the TOX barrier can act as stepping stones when floating gate electrons leak via a trap-assisted tunneling process toward the cell channel region. As the density of traps increases, the probability of this trap-to-trap tunneling (called stress-induced leakage current, SILC) becomes much higher than a direct tunneling process (for program or erase) through the whole TOX thickness. It means the *TOX is damaged*. The reason is that the effective tunnel distance of each tunneling step is significantly reduced for the SILC.

b) Endurance

Endurance is the maximum number of program/erase (P/E) operations (or P/E cycles) that the memory can withstand before it fails. Figure 2.*5* shows the endurance of FG cells in a 48 nm NAND technology. All program and erase cycles were carried out with unchanged program and erase

voltages of $V_{PGM} = 23$ V and $V_{Erase} = -19$ V for the indicated pulse times (program period and erase period). The V_{TH} window has a decent size for low cycle numbers, whereas the V_{TH} window shrinks for higher cycle numbers above 300 cycles. Furthermore, a general V_{TH} upward shift is visible as the cycle number increases. The V_{TH} window closing and V_{TH} upward shift behaviors are because, at higher cycle numbers, the fixed negative charges which are generated in the TOX (trap generation) generally increase the cell's V_{TH}. In the case shown in Figure 2.5, the erased cell's V_{TH} is shifted by one volt after 10 k program/erase cycles. Besides the increased retention problem for higher cycle numbers due to trap generation, the window closing and the general V_{TH} upward shift will increase pulse voltages for electron charge, especially for the erase operation.

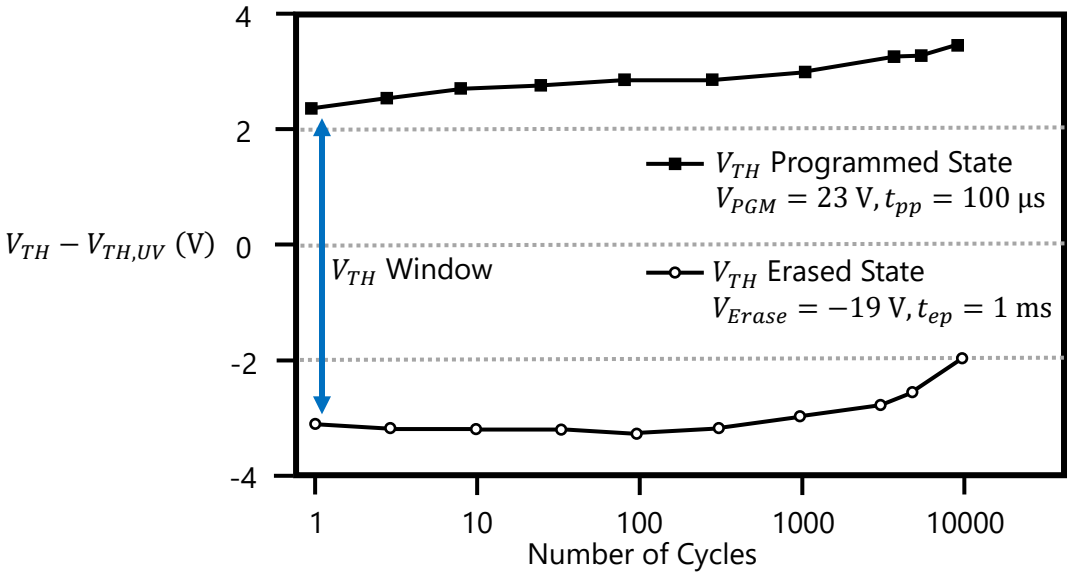


Figure 2.5 Program/Erase Cycling Endurance of an FG Cell in a 48 nm NAND Technology.

c) Number of Stored Floating Gate Electrons

When the dimensions of floating gate cells are scaled down, the number of floating gate electrons needed for a certain threshold voltage shift ΔV_{TH} is also reduced. This reduced number of stored floating gate electrons is critical for reliability because the loss of one electron increases the cell V_{TH} loss. On the other hand, the charge granularity of single electrons affects, at a certain stage, the ability to program narrow V_{TH} distributions. This effect becomes more critical in TLC or later technologies with narrow V_{TH} distributions if one electron causes a significant threshold voltage shift.

(Todo)

Charge trapping (CT) cell, a 3D NAND flash memory cell, is the planar memory cell replacement of the conventional floating gate cell. At first glance, the construction of CT memory cells for NAND application is not very different from the floating gate NAND cell construction. The major difference is that the charge is stored in a non-conducting dielectric layer with high trap density instead of the conducting floating gate.

The cylindrical shape of the memory cells in 3D cell approaches have one major advantage over fully planar memory cells, namely the electric field enhancement in the TOX and the field reduction in the inter gate dielectric (IGD)
(the cylindrical cell geometry is used in most of the 3D NAND Flash memory arrays)

(Todo)

Encoding more bits per cell increases the capacity of the SSD without increasing the chip size, yet it also decreases reliability by making it more difficult to correctly store and read the bits.

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.

(charge trap transistors for 3D)

2.2. NAND Flash Array
We will use schematic diagrams to delineate the NAND flash array from here. **In schematic diagrams, wires are always joined at three-way junctions**. They are joined at four-way junctions only if a dot is shown. **The slash across the input wire indicates that the gate may receive multiple inputs**.

Flash memory for non-volatile data storage was introduced commercially in the mid-1980s. Since then, common ground **NOR and NAND architecture** have become the most common memory array architectures. Traditionally, *NOR flash is used for code storage due to faster memory cell access*. *NAND flash is used for mass data storage because of its higher memory density, enabling higher storage capacities*.

The difference in memory cell area is evident from the schematic NOR and NAND array diagrams in *Figure 2.6*. In the NOR array, two memory cells share one contact with the ground (SL: Source Line) and one contact with the bit line (BL), as shown in *Figure 2.6*(a). This results in an effective memory cell area of about $10\ \mathrm{F}^2$ for 6 cells. In the NAND array, two or more memory cells are connected in series to form a string, called the NAND string, which is a quite compact structure. *Figure 2.6*(b) illustrates the so-called NAND string, comprising up to $m+1$ memory cells connected in a column. For the NAND string operation, two additional select transistor devices need to be added: (1) Bit Line Selector (BLS), and (2) Source Line Selector (SLS). These additional structures cause the effective cell area consumption slightly higher than $4\ \mathrm{F}^2$ for 64 cells, the theoretically smallest effective cell size. Note that only NAND flash is a viable option for SSD applications due to the required high memory capacity and bit cost structure.

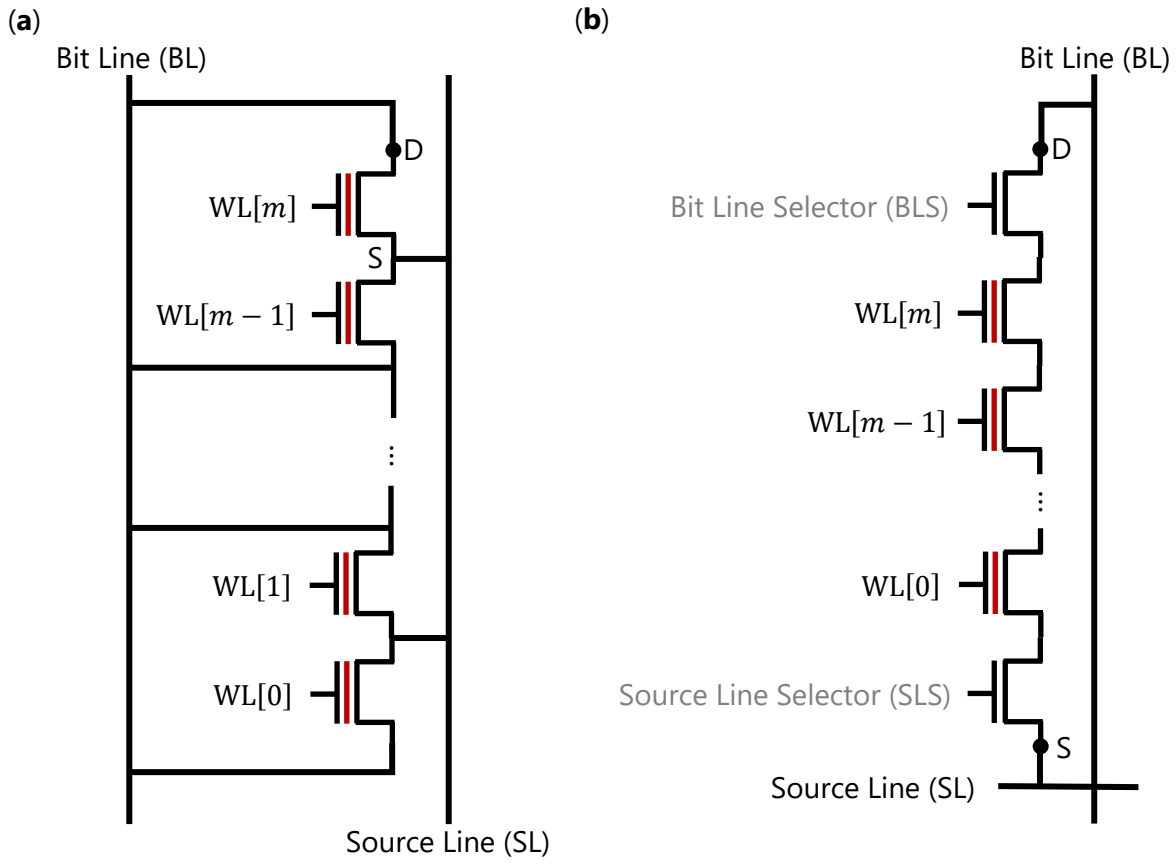


Figure 2.6 Schematic Memory Cell Organization of the NOR Array (**a**) and the NAND Array (**b**). The Word Lines (WL) Run Perpendicular to the Bit Lines (BL).

The basic element of a NAND flash memory, **NAND string** as shown in Figure 2.7(a), has multiple cells (e.g., 128) connected serially, and two selection transistors placed at the edges of the string. BLS connects the NAND string to a bit line, and SLS connects the other side of that string to a source line. The drain select line (or string select line) controls the BLS, and the source select line (or ground select line) controls the SLS. The cell's control gates are connected through the word lines (WLs). The **NAND array** is composed of multiple NAND strings. Figure 2.7(b) shows how the matrix array (i.e., NAND array) is built starting from the basic string. In the WL direction, the adjacent NAND strings share the same WL, DSL (drain select line), SSL (source select line), and SL (a single source line acts as the *common source diffusion*). In the BL direction, two consecutive strings share the bit line contact.

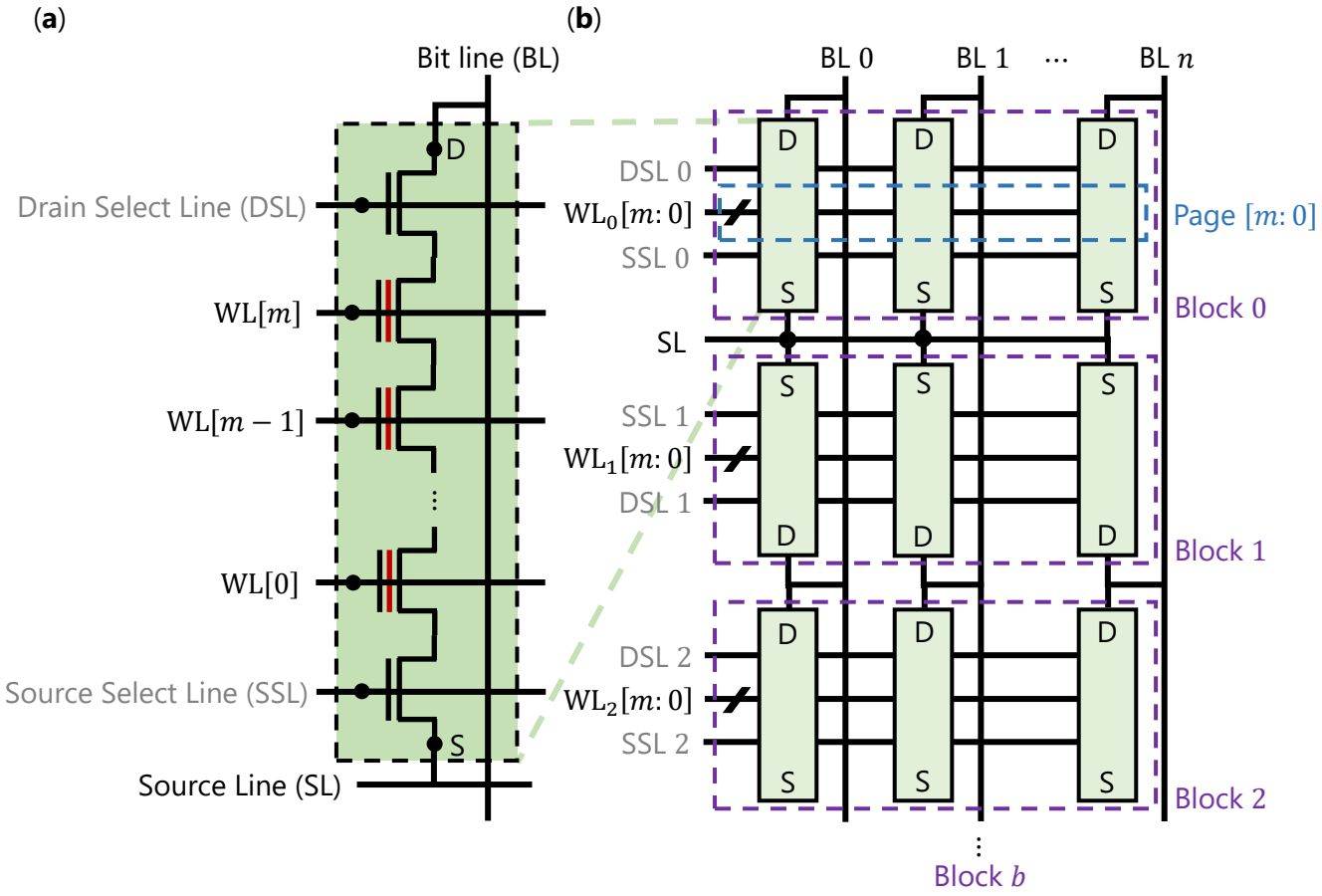


Figure 2.7 NAND String (**a**) and NAND Array (**b**).

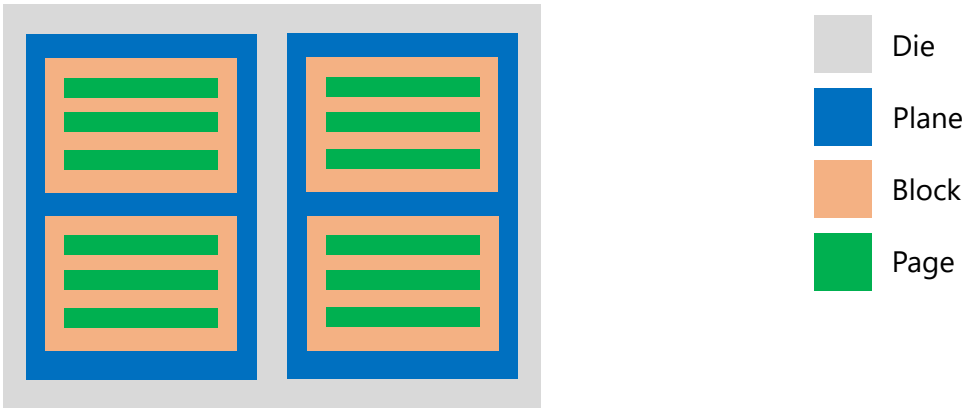
All the NAND strings sharing the same group of word lines form a **block**. In Figure 2.7(b), each block is made up of WL$[m\colon 0]$. A physical NAND **page** is a group of NAND flash cells belonging to the same word line of the same block, which share, horizontally, the same control gate. Thus, a NAND block is composed of several pages. *The number of pages per WL is related to the storage capabilities of the memory cell*. Depending on the number of storage levels, NAND flash memories are referred to in different ways:

- SLC memories store 1 bit per cell.
- MLC memories store 2 bits per cell.
- TLC memories store 3 bits per cell.
- QLC memories store 4 bits per cell.

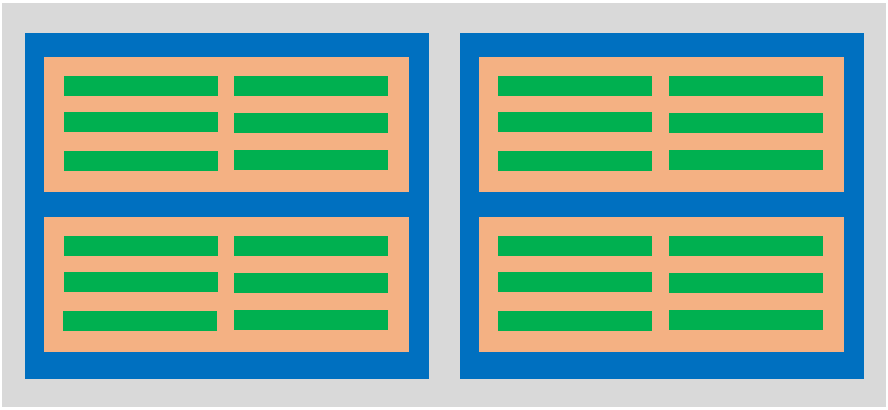
The number of pages along a word line depends on the storage levels. For example, a WL of TLC memories contains 3 pages.

Several NAND blocks form a **plane** where cells operate together. NAND flash devices may contain independent flash planes with their own buffer to hold data, allowing simultaneous operations for higher performance (i.e., multi-plane parallelism). This technique has limited effectiveness due to the constraints of the same word line address for required operations. To tackle that problem, planes form a **die**. A die is the minimum unit that can independently execute commands and

report its status. It contains at least one page buffer and a flash array (The number of page buffers is dependent on the number of plane operations supported for the die). Multiple dies can exist within a single NAND flash chip, allowing for increased parallelism and concurrency by enabling simultaneous operations on different dies (although they compete for the NAND flash chip pins). Figure 2.*8* illustrates the NAND flash layout of SLC and MLC memories. A page is the smallest unit that can be programmed (written to) and read, typically ranging in size from 8 to 16 KiB. A block is the smallest NAND die portion that can be erased.



(**a**) SLC Memory



(**b**) MLC Memory

Figure 2.8 NAND Flash Layout of SLC (**a**) and MLC (**b**) Memories.

Furthermore, a **die** within a NAND flash device can also be referred to as a **logical unit number** (LUN). For example, it is permissible to start a program operation on LUN 0 and then, prior to the operation's completion, to start a read operation on LUN 1.

Supplement materials

The schematic NOR array diagram can also be drawn as Figure 2.*9* shows:

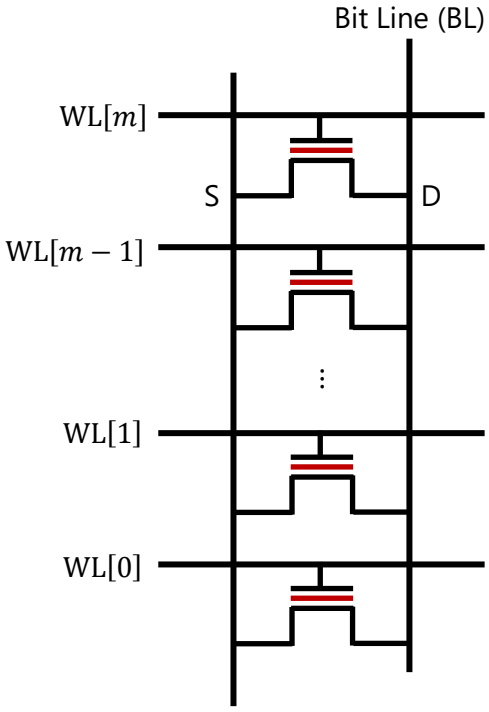


Figure 2.9 NOR Flash Array.

Program, Erase, and Read of Flash Cells in the NAND String

When a large number of floating gate cells need to be operated in the NAND array, it is essential to consider that one floating gate cell is located at every intersection of bit lines and word lines. In particular, the memory cells in the NAND array can no longer be operated independently of each other. In the word line direction (depending on the page size), a couple of thousand FG cells are controlled by the same word line. In bit line direction, the only way to access an individual cell for either reading or writing is through the other cells in its bit line (i.e., the string size (e.g., 64-66 cells) defines the number of cells that cannot be operated independently). This string structure adds significant noise to the read process, and also requires care during the writing process to ensure that adjacent cells in the string are not disturbed. During erasure, in contrast, all cells on the same bit string are erased. Consequently, it is crucial to consider the impact on all neighboring cells when a single cell is treated.

Following *Figure 2.3*, the threshold voltage of each memory cell is carefully adjusted, as shown for SLC and MLC cells in *Figure 2.10*.

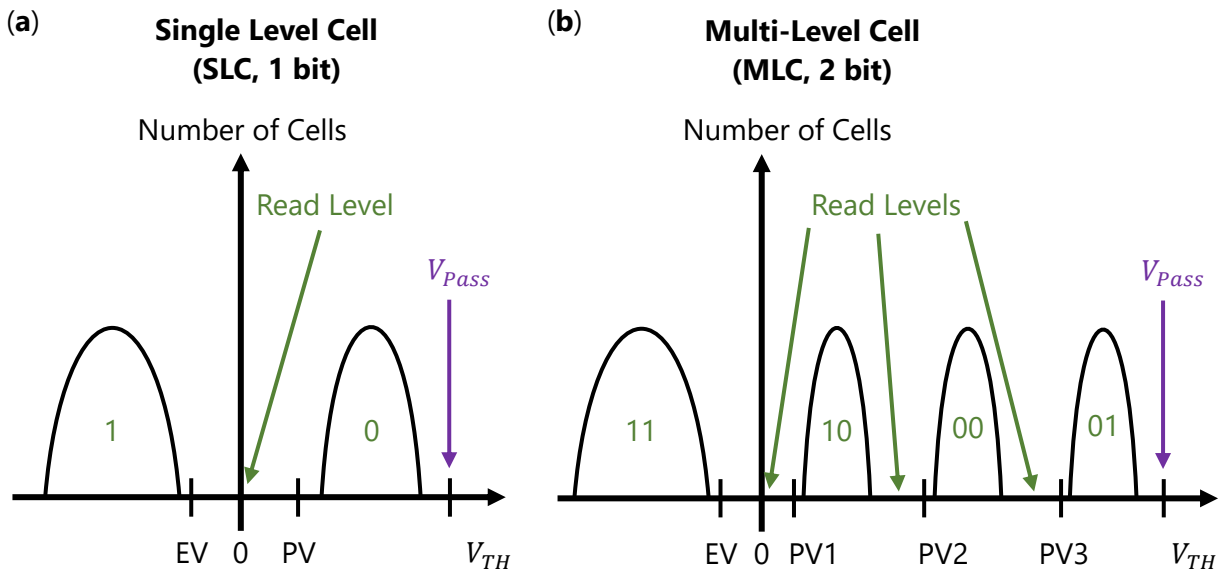


Figure 2.10 Memory Cell Threshold Voltage Distributions for One Bit Per Cell (SLC) Data Storage (**a**) and Two Bit Per Cell (MLC) Data Storage (**b**) in a NAND Flash Array.

The V_{TH} distribution of erased cells is placed at negative V_{TH} values. In an ISPP-like sequence, the erase voltage is increased until all cells are erased below the Erase Verify (EV) level. **The V_{TH} distributions of programmed cells** are placed in the positive V_{TH} range. For a single-level cell (SLC), the ISPP programming is continued until all cells designated for programming are above the Program Verify (PV) level. Consequently, in the case of multi-level cells (MLC), there are three program verify levels (PV1, PV2, and PV3). In addition, the margins between the different programmed V_{TH} distributions must be guaranteed to be large enough to place the read levels (V_{REF}) and have sufficient space for charge/retention loss-caused V_{TH} reductions. A specific distribution shaping algorithm with a small program step increase in certain stages of ISPP programming is necessary to obtain these narrow cell V_{TH}. The pass voltage (V_{Pass}), higher than the maximum possible V_{TH}, makes the cell behave as a pass-transistor.

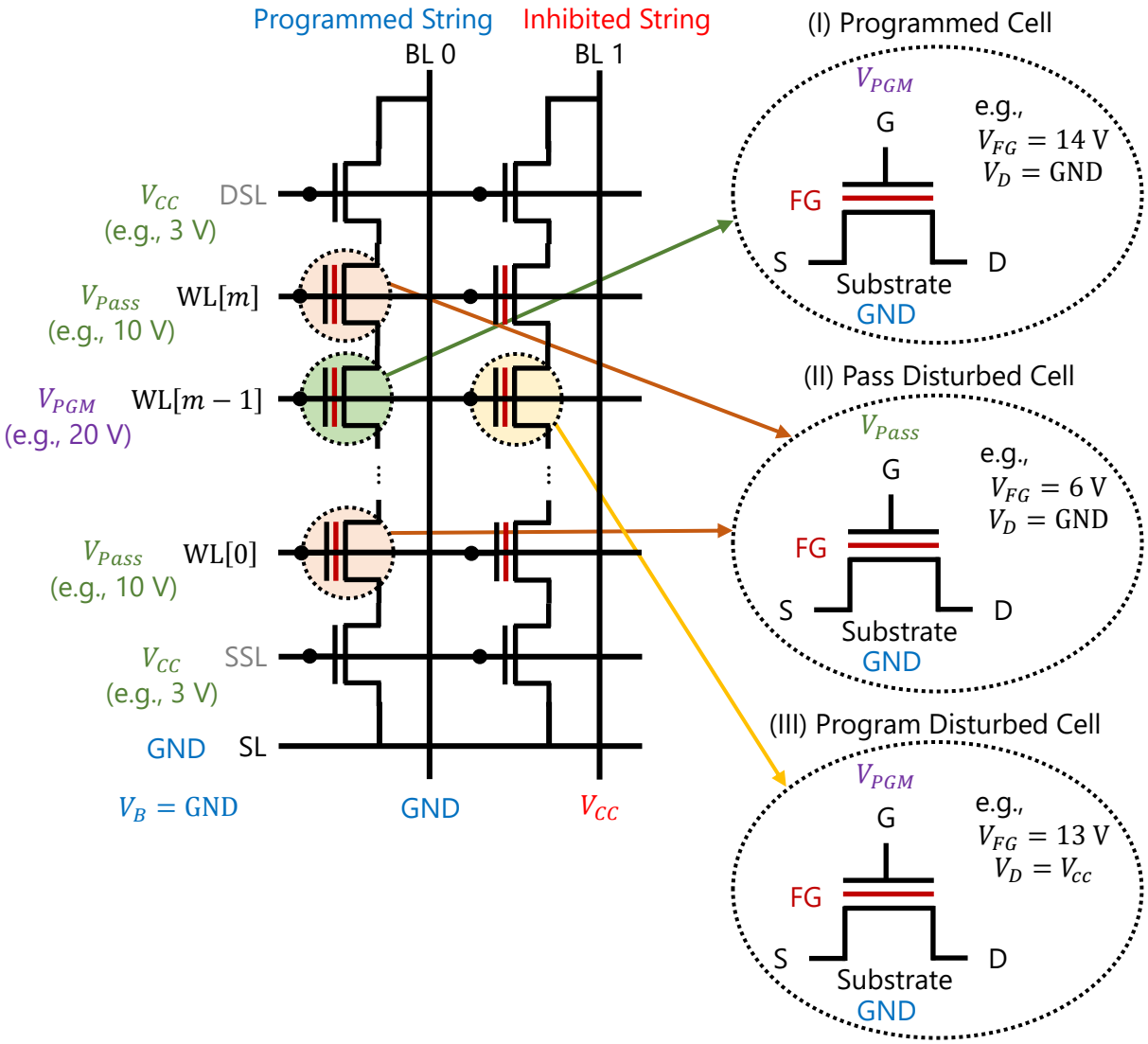


Figure 2.11 Voltage Conditions During Program Operation in the NAND Array. The Memory Cell at the Crossing Point of WL[$m-1$] and BL 0 is Programmed; Several Other Cells are Disturbed by Either Pass Disturb or Program Disturb.

Figure 2.11 shows the voltage condition in the NAND array when the FG cell (selected) at WL[m − 1] and BL 0 is programmed (program operation). For this purpose, a program pulse with the pulse amplitude of V_{PGM} (e.g., 20 V) is applied to WL[m − 1]. To conduct a successful program, it is also required to transfer 0 V (GND) to the channel region (V_D) of the programmed cell as shown *Figure 2.11*(I). Consequently, the 0 V potential is applied to BL 0 and then needs to be transferred to the whole string including the programmed cell at WL[m − 1]. This is done by applying the pass voltage V_{Pass} (e.g., 10 V) to all other word lines so that all other cells (unselected) operate as a resistance.

In principle, **all cells addressed by the same word line could be programmed by this means simultaneously**. However, the programming of arbitrary information requires that specific

memory cells at the word line are excluded from programming. In this example, the cell at the crossing point of BL 1 and $\mathrm{WL}[\mathrm{m}-1]$ represents the cells that should be prevented from programming (program-inhibited cell in *Figure 2.11*). In former FG NAND generations, programming in certain NAND strings was avoided by actively applying a positive voltage to the corresponding bit lines. As a result, the voltage difference between the channel and the control gate was not high enough for programming in these strings. This procedure was complicated, and the voltage pumps used for this purpose required additional power and chip area. Therefore, the so-called "Self-Boosted Program Inhibit" (SBPI) scheme was introduced in recent generations. The principle of the SBPI scheme is that the channel potential in the inhibited strings is not actively raised by applying a voltage but capacitively raised.

Pass disturb and program disturb result in a threshold voltage increase during program operation. The pass cells, located in a string with a memory cell dedicated to programming, experience soft programming when the pass voltage is increased beyond a certain limit (pass disturbed cell in *Figure 2.11*(II)). The program cells, located in a word line with one or more memory cells dedicated to programming, also experience soft programming when the program voltage is high enough to weaken the inhibiting effect.

The NAND flash erase operation erases a whole block at once to reduce chip area for cost reasons. The voltage conditions during the erase operation are shown in *Figure 2.12*. All word lines are at ground potential ($V_{CG} = 0\ \mathrm{V}$) and the erase voltage is applied to the well of that block. The select transistors, the bit line, and the source line must be left floating during the erase operation. For this purpose, the usually grounded source line needs to be disconnected from the ground potential. By this means, the source line and the bit line, and to a certain extent, the select transistors, can follow the bulk potential to avoid large currents flowing into the source line and the bit line. Because of the improved coupling, the voltage difference between the control gate and the channel required for the erase (e.g., $V_B = 18\ \mathrm{V}$) is lower than the programming voltage when the same voltage is applied to all cells. The erase operation is successful when all cells in the erase block are erased below the EV level, as described above.

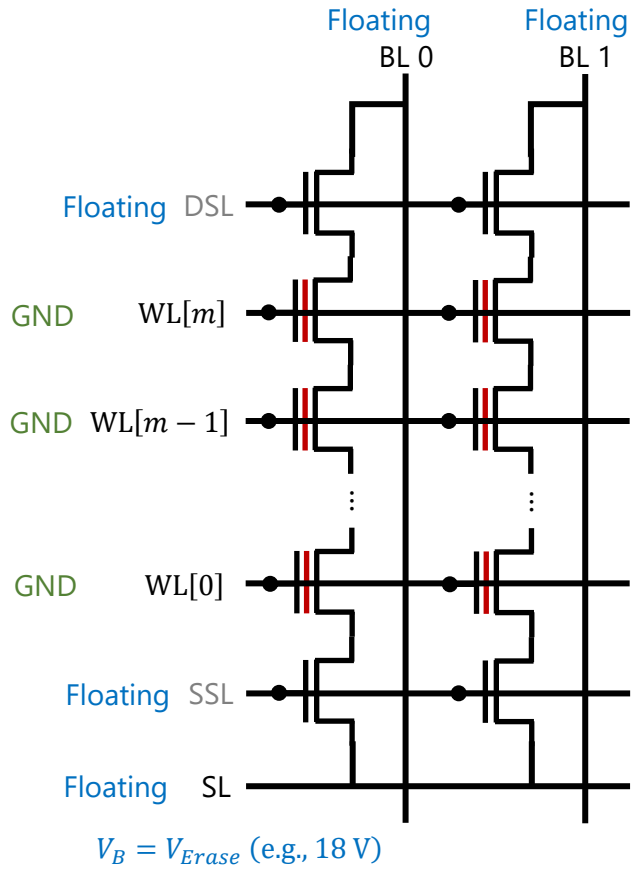


Figure 2.12 The Erase of Floating Gate Cells in the NAND Array is Carried Out Electrically in Separated Erase Sectors. By Applying a Positive Voltage to the Well of the Erase Sector, All Cells are Erased at the Same Time.

Because the granularity of erase operation is usually larger than 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 (we will mention it later).

The read operation addresses specific memory cells within the array at a word line unit. For a current sensing read scheme, the bit line selected for a read operation (BL 1 in *Figure 2.13*) is set to the read voltage (e.g., V_{CC}), the word line at the read cell is set to the V_{REF}, and all other word lines are set to the pass voltage. If the cell at $\mathrm{WL[m-1]}$ in the string of BL 2 is erased, the reading current can flow through all cells in that string, and vice versa. The recent sensing technique integrates the cell current on a dedicated capacitor to sense the current flow. We will explain the basic structure later. Note that only one cell can be read at a time.

The pass voltage can also change the threshold voltage (read disturb) when the number of read operations is high enough. *Figure 2.13*(II) shows that the cells of a string in BL 0 are under soft programming due to read disturb. For SLC FG NAND cells, it is assumed that 10^6 read operations with each 15 μs duration need to be guaranteed without read fails. This results in a total disturbance of about 15 seconds.

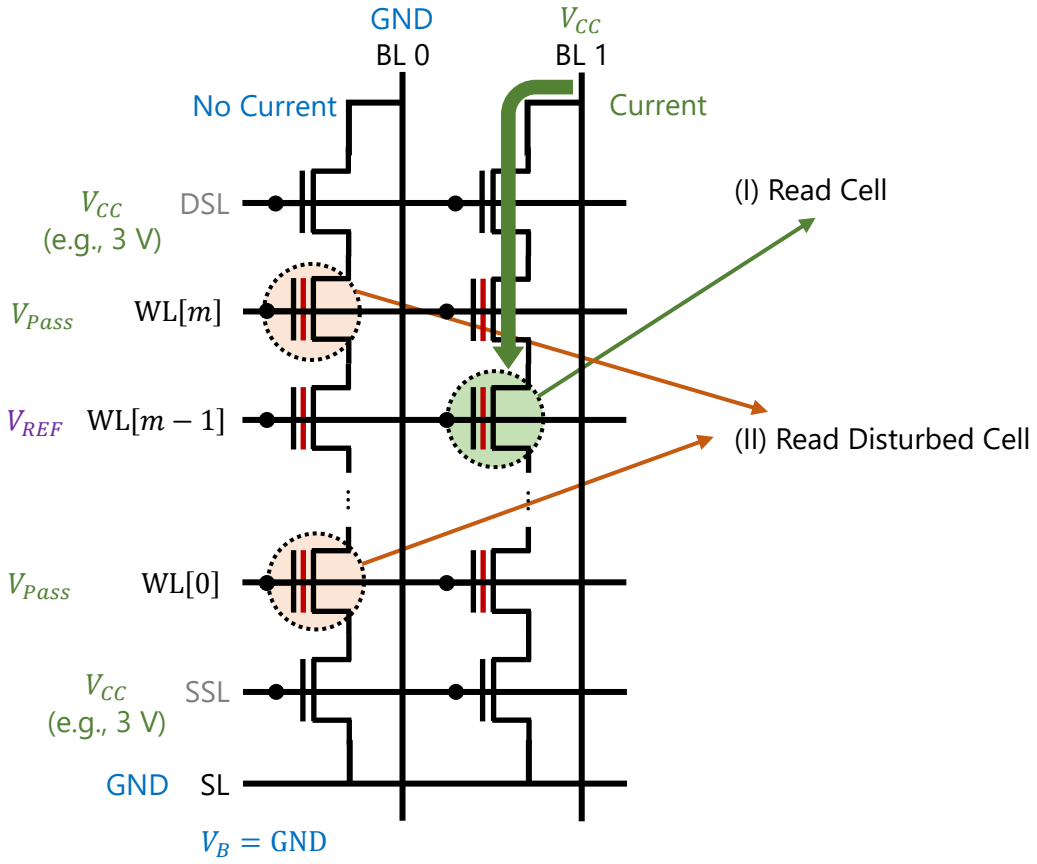


Figure 2.13 Read Operation in the NAND Flash Array.

Figure 2.*14* shows the basic structure of the recent current sensing technique and its related timing diagram. It integrates the cell current on a sense-out (SO) capacitor and utilizes the sense amplifier to sense the current flow in a fixed time (i.e., read the analog information stored in the memory cell). The read mechanism consists of three phases: 1) precharge, 2) evaluation, and 3) discharge. In the precharge phase ($T_{Precharge}$), a NAND flash chip charges each target BL and its sense-out (SO) capacitor C_{SO} to a specific voltage (e.g., $V_{PRE} = V_{CC}$) by turning on the precharge (PCH) transistor. The chip also applies the read-reference voltage V_{REF} to the target cell at the same time mentioned before, which enables the BL to sink current (I_{Cell}) through the NAND string depending on the cell's V_{TH} level (i.e., the BL can sink current when $V_{TH} < V_{REF}$). The chip then enters the evaluation phase ($T_{Evaluation}$), disconnecting the BL from the device's power supply voltage by turning off the PCH transistor and enabling the sense amplifier (SA). If the target cell had been programmed (i.e., $V_{TH} > V_{REF}$), the capacitance of C_{SO} would hardly have changed as the BL could not sink current. In contrast, if the cell had been erased, the charge in C_{SO} quickly flows through the BL as the cell current (I_{Cell}), which rapidly decreases the SO-node voltage below the SA's reference voltage V_{Sense}. As a result, the SA can tell whether the V_{PRE} is smaller or larger than the V_{Sense} via the voltage comparator and output the result through the latch (or flip-flop). Finally, the chip discharges the BL to return to the initial state for future operations in the discharge phase ($T_{Discharge}$).

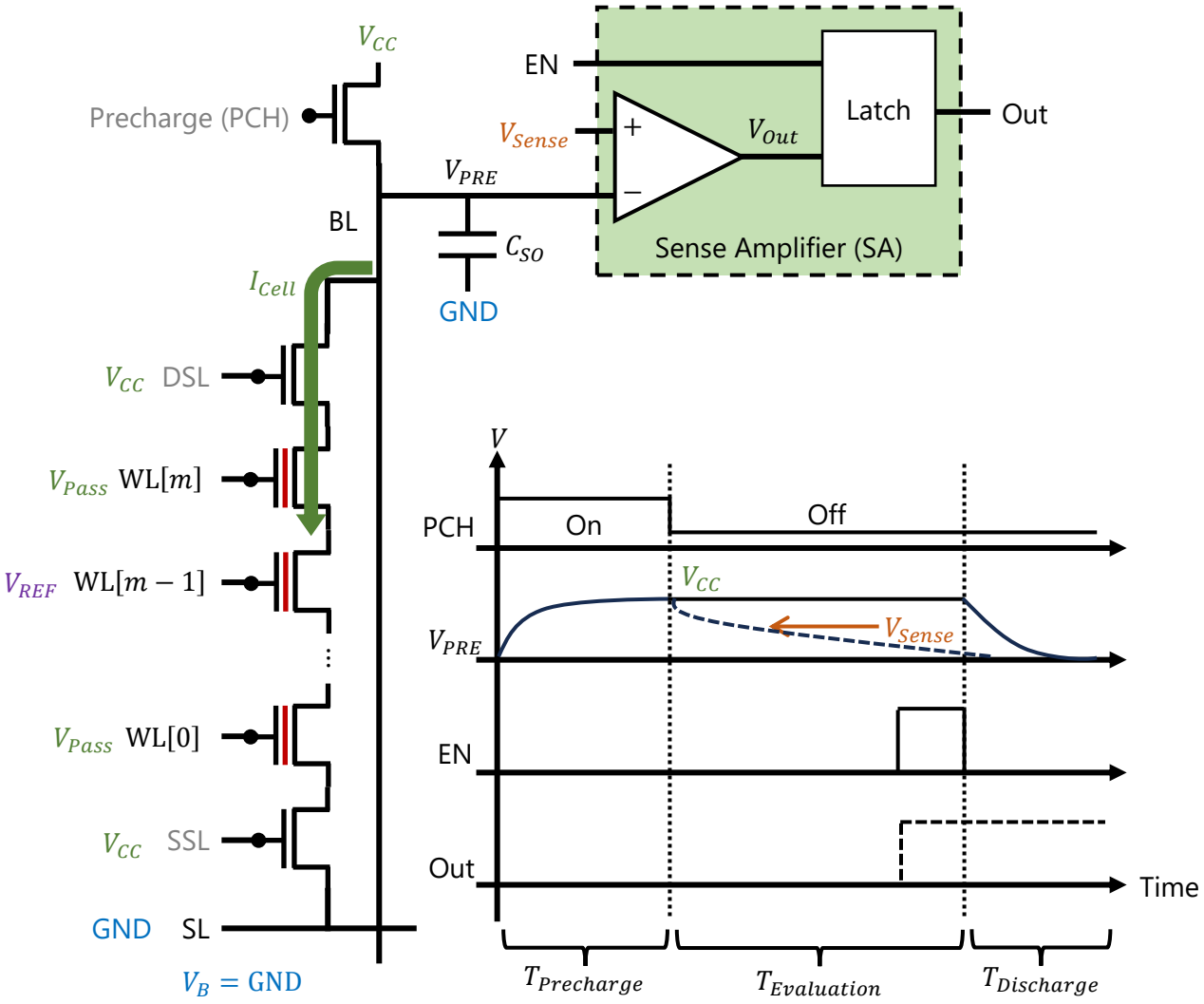


Figure 2.14 Basic Structure of Recent Current Sensing Technique and Its Related Timing Diagram.

Put it all together, a physical page is the smallest addressable unit for reading and writing, and a physical block is the smallest erasable unit in NAND Flash memories.

Scaling Challenge of Floating Gate NAND Memory Cells

The NAND flash memory scaling of the last 15 years (from 2018) was accomplished by reducing the cell dimensions, whereas the cell construction principle was unchanged. The effective cell size of NAND Flash in 1995 was in the range of 1 μm^2, which resulted in a product chip memory capacity of 32 Mb. In 2010, the cell size was reduced to 0.0028 μm^2 with a chip capacity of 64 Gb. This substantial reduction of cell geometry leads to scaling issues, which are discussed below.

a) Scaling Limitation of the Floating Gate Cell Geometry

For the programmability of floating gate cells, it is important to have an enhanced control gate in the floating gate area, where a control gate is wrapped around the floating gate. A space between adjacent floating gates is required to build an enhanced control gate.

b) Cell-to-Cell Interference

A general problem for floating gate NAND cells in technology generations below 50 nm is the **cell-to-cell cross-coupling**, also called floating gate cross-coupling or floating gate interference. Figure 2.*15* shows this effect: the direct coupling from one floating gate to the nearest neighboring floating gates. This direct coupling increases as NAND flash memory cells are scaled down (or reduced dimensions) because the cells move closer together, increasing the relative coupling capacitance (i.e., cell-to-cell parasitic capacitance). The most significant part is the FG-to-FG coupling in the direction along the bit lines (y-direction in Figure 2.*15*). The reason is that the floating gates directly face each other with the full FG height and full FG width in this direction. Consequently, $C_{FG,y}$ is the largest of the FG-to-FG coupling capacitance terms. In the direction along the word lines (x-direction), parts of the FG-to-FG coupling are screened by the control gate, and therefore, $C_{FG,x}$ is typically smaller than $C_{FG,y}$. The diagonal coupling components $C_{FG,xy}$ and $C_{FG,yx}$ are typically the smallest ones.

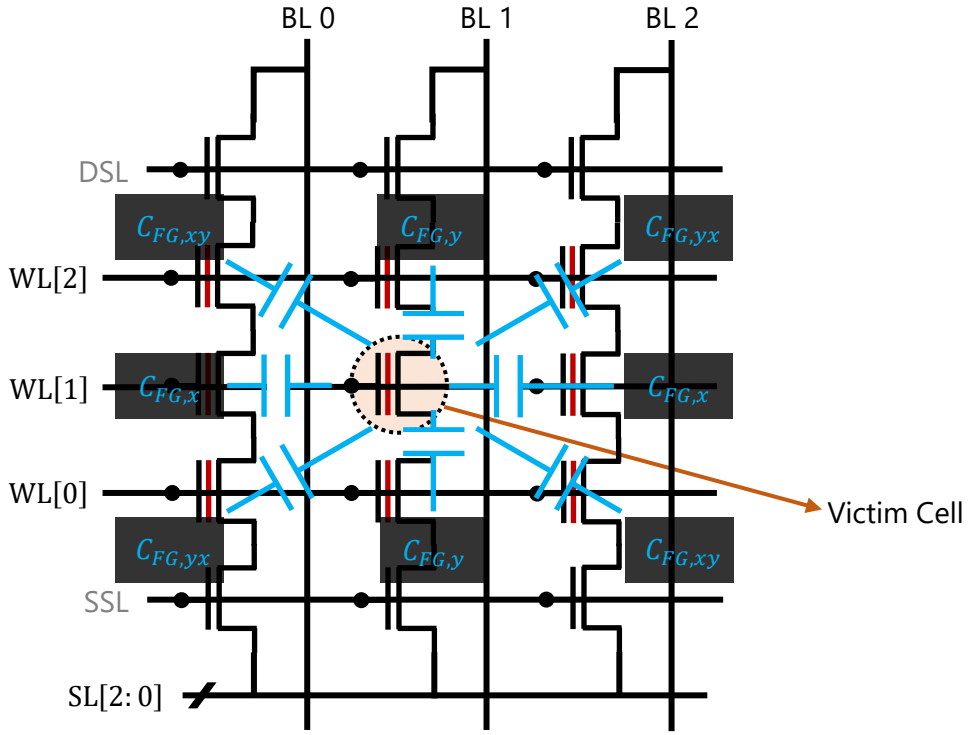


Figure 2.15 Floating Gate Cross-Coupling in Scaled NAND Flash Technologies.

Cell-to-cell interference causes threshold voltage variation (ΔV_{TH}) of a victim cell during the read operation. The reason is that the programming of a cell can change the threshold voltage of a directly neighboring cell (i.e., victim cell), which was already programmed or erased. When a neighboring cell changes its V_{TH} (e.g., during programming or erasing), the capacitive coupling causes a shift in the V_{TH} of the victim cell (or interfered cell). This shift can affect the victim cell's ability to read correctly or lead to data corruption.

Therefore, when shrinking the FG NAND flash dimensions, the program algorithm needs to take care of the floating gate cross-coupling issue at a certain point. The strategy is to reduce the number of neighboring programmed cells after reaching the final programming target V_{TH} of each cell, in combination with reducing the amount that these neighboring cells increase their V_{TH}. For example, we can improve a standard program algorithm to program cells in a particular order. Cell-to-cell interference during the erase operation is uncritical because all cells (in a block) are erased simultaneously and therefore all cells return to the ER state.

c) Word Line to Word Line Leakage Current
During a program operation, the reduced cell-to-cell distances with scaled dimensions cause strongly increased electric fields between neighboring word lines. Besides, high WL voltage differences during program operation become critical since the programming voltage does not scale but increases slightly. As a result, electrons can tunnel from a programmed floating gate to the control gate on the high program voltage V_{PGM} or generally introduce WL-to-WL leakage currents. Note that the WL-to-WL voltages during erase are uncritical because all cells are erased at the same time, and therefore, all word lines are at the same potential.

Options to reduce WL-to-WL leakage using a special program algorithm include limiting the difference voltage between adjacent word lines.

d) Random Telegraph Noise
Different types of field effect devices have observed random telegraph noise (RTN). It can be explained by electron capture and emission processes in oxide traps close to the channel of a MOSFET device. RTN in the string current results in a threshold voltage shift ΔV_{TH}, which can cause read failures.

3D NAND Flash Memories
Planar memory cells have been scaled for decades to achieve larger capacity by improving process technology, circuit design, programming algorithms, and lithography. However, when approaching a minimum feature size of 1x-nm (Feature size refers to the smallest physical dimension of a component during manufacturing, measured in nanometers), more challenges pop up: doping concentration in the channel region becomes difficult to control, RTN (Random telegraph noise mentioned in 2.2 (d)) and electron injection statistics widen threshold distributions, thus causing a significant hit to both endurance and retention. Furthermore, by reducing the distance between memory cells, the intra-word line electric field becomes higher (2.2 (c)), pushing the bit error rate to an even higher level. 3D arrays leverage either floating gate (FG) or charge trapping (CT) technologies to fuel a further increase in the bit density. In fact, the vast majority of 3D architectures published to date are built with CT cells, primarily due to the simpler fabrication process. Nevertheless, the floating gate remains, and commercial products have successfully integrated it into a 3D array.

Identifying the right way for going 3D is not so easy though. With 3D architectures, the "simple" reduction of the minimum feature size is running out of steam, as shown in Figure 2.*16*: a higher number of stacked cells is the only hope for dramatically reducing the real estate of a stored bit.

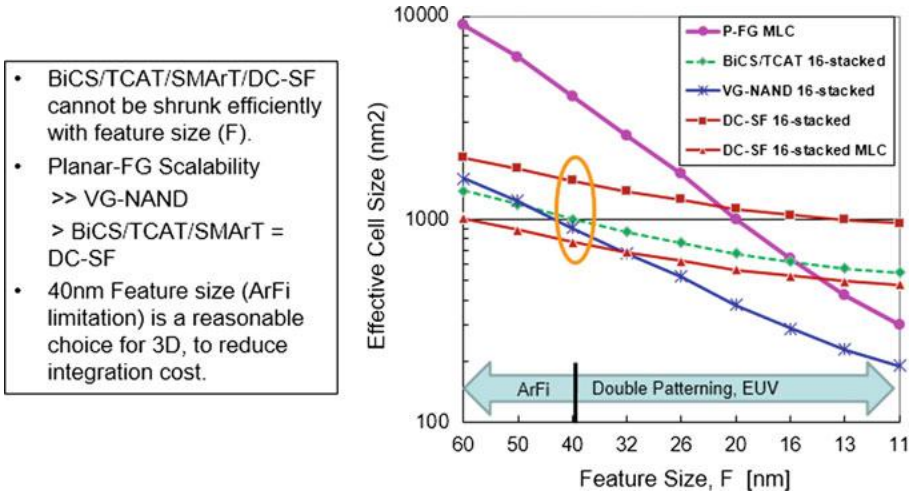


Figure 2.16 3D NAND Flash Scaling.

3D arrays can be efficiently built by vertically rotating the planar NAND flash string shown in Figure 2.*17*. The solution of choice is a conduction channel completely surrounded by the gate: indeed, the curvature effect helps increase the electric field E_t across the tunnel oxide, and reduces the electric field E_b across the blocking oxide, and this has a positive impact on oxide reliability and overall power consumption.

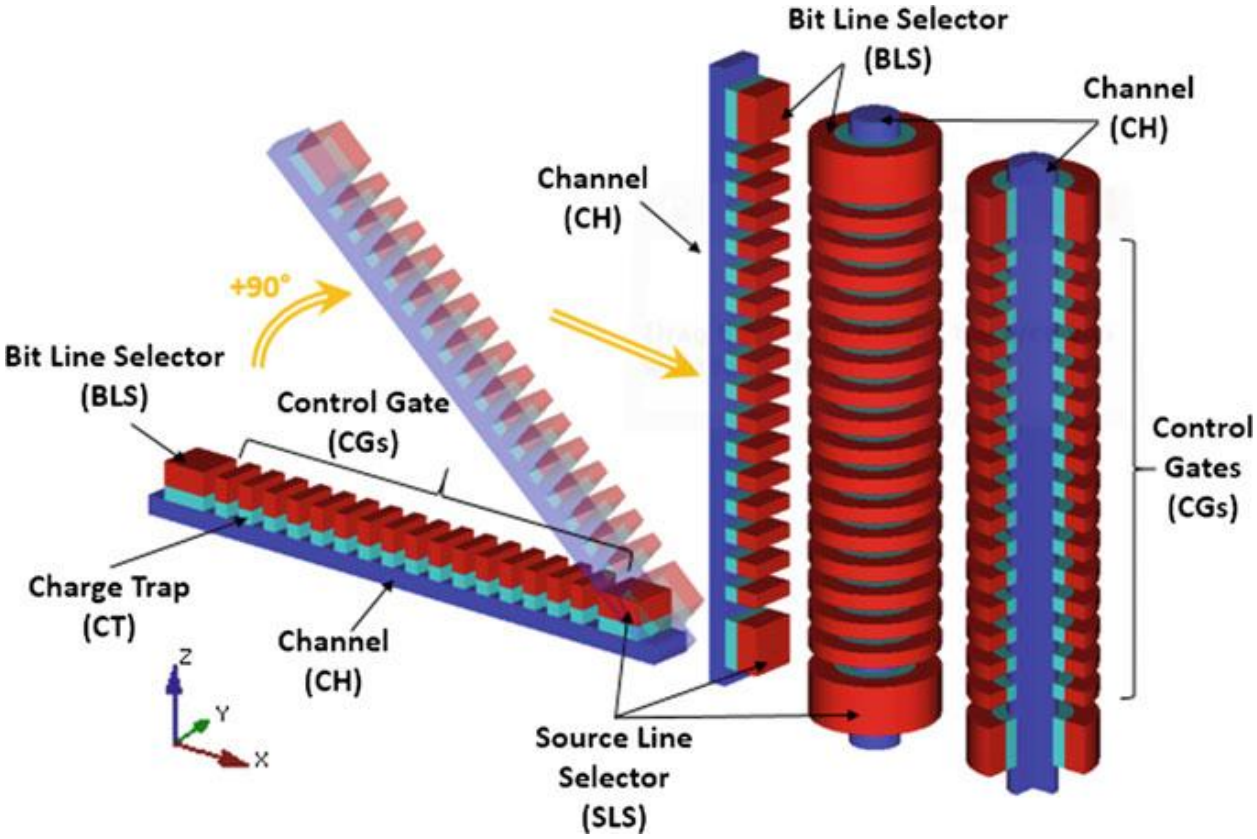


Figure 2.17 NAND Flash String Goes Vertical.

Vertical channel arrays have been historically driven by architectures known as **BiCS**, which stands for *Bit Cost Scalable*, which leverages CT cells. Let us begin with BiCS, which is illustrated in Figure 2.*18* and Figure 2.*19*. There is a stack of control gates (CGs), the lowest being the one of the source line selectors (SLSs). The whole vertical stack is punched through, and the resulting holes are filled with polysilicon; each filled hole (a.k.a. pillar) forms a series of memory cells vertically connected in a NAND fashion. Bit line selectors (BLSs) and bit lines (BLs) are formed at the top of the structure. As usual, a select transistor (BLS) is used to connect each NAND string to a bit line; there is also another select transistor (SLS), which connects the other side of the string to the common source diffusion.

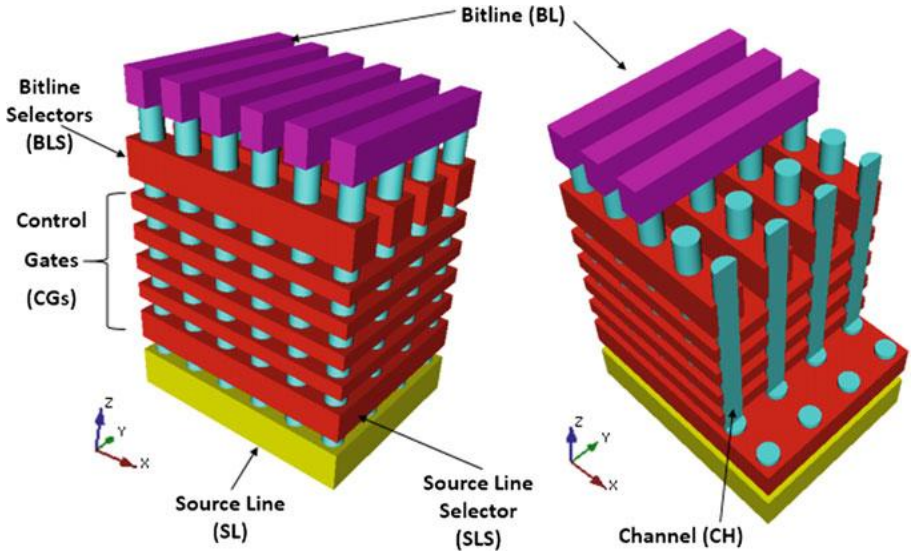


Figure 2.18 BiCS Architecture.

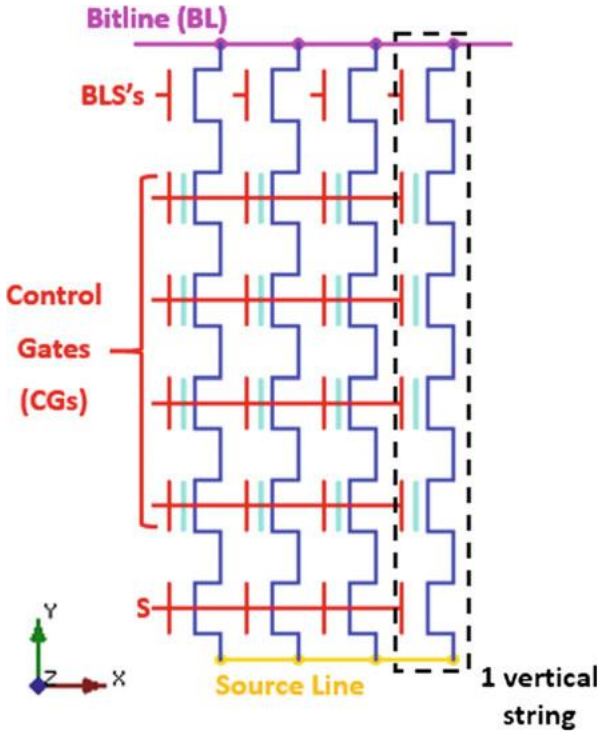


Figure 2.19 Equivalent Circuit of a BiCS Array.

As sketched in Figure 2.*20*, vertical transistors have a poly-silicon body, and this fact turned out to be one of the critical cornerstones of the 3D foundation. From a manufacturing perspective, the density of the traps at the grain boundary is difficult to control due to its vertical shape. The poor control of this density induces significant fluctuations in the characteristics of vertical transistors.

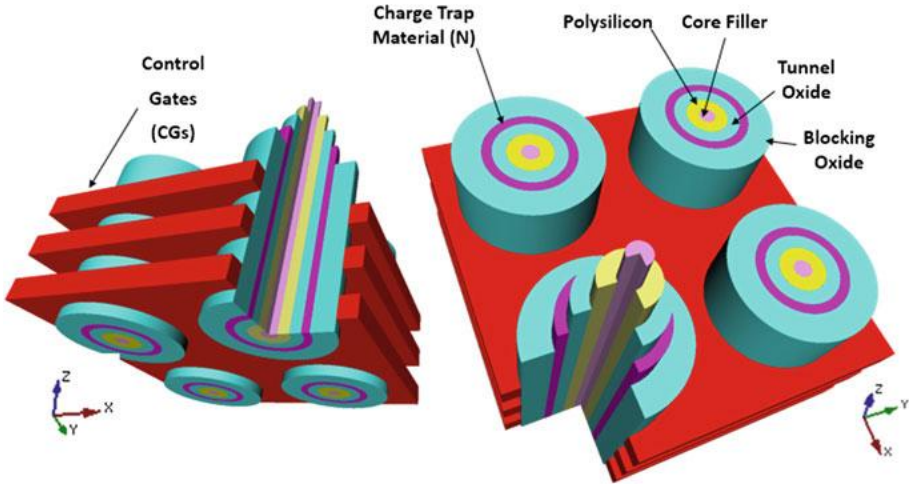


Figure 2.20 BiCS Memory Cells.

The recipe for fixing the trap density fluctuation problem is to manufacture a poly-silicon body much thinner than the depletion width. In other words, by shrinking the poly-silicon volume, the total number of traps decreases (Figure 2.*21*). This particular structure is typically referred to as a

Macaroni Body. A filler layer (i.e., a dielectric film) is used in the central part of the *Macaroni* structure because it makes the manufacturing process easier.

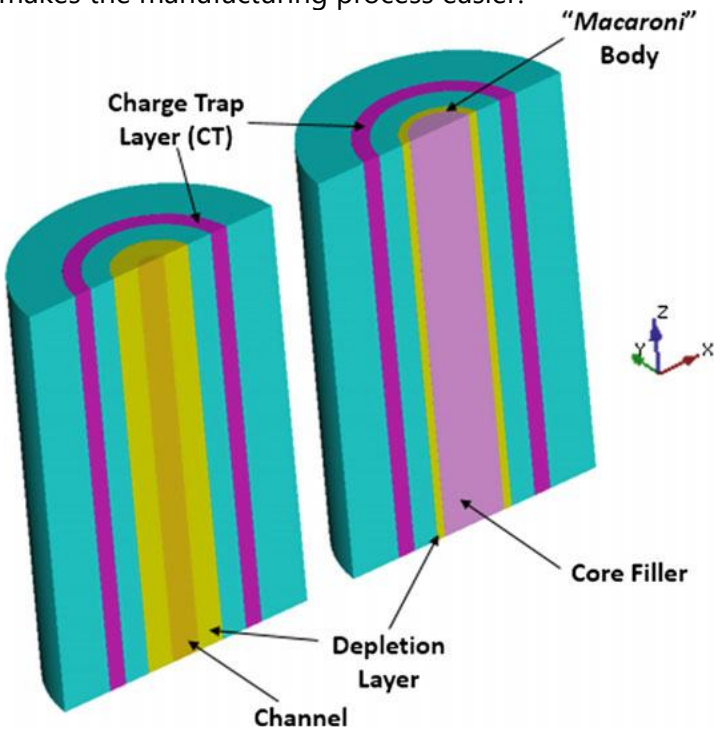


Figure 2.21 A Vertical Transistor (right) Modified with *Macaroni* Body (left).

Besides BiCS, many other approaches were tried, including P-BiCS, VRAT (Vertical Recess Array Transistor), Z-VRAT (Zigzag VRAT), VSAT (Vertical Stacked Array Transistor), TCAT (Terabit Cell Array Transistor), V-NAND, and 3D-VG (Vertical Gate) NAND which is a unique architecture where the channel runs along the horizontal direction.

(Note a 3D layer means gate layer in 3D NAND flash memories)

(todo)

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

Within each plane, flash cells are organized as multiple 2D arrays known as flash blocks, each of which contains multiple pages of data, where a page is the granularity at which the host reads and writes data.

Reliability of NAND Flash Memory

The basic parameter characterizing the NAND flash memory reliability is the raw bit error rate (RBER), representing the fraction of erroneous bits retrieved during a read operation (todo).

Multilevel NAND flash memories require the availability of an ECC scheme able to correct the errors detected when reading the memory. The choice of the ECC code and the design of the correction engine represent the key points for present SSDs design since they must be carefully calibrated with respect to the figures of merit of the selected nonvolatile memories. A too simple ECC scheme may not be able to guarantee a suitable reliability, whereas a too complex one may reduce severely the read bandwidth because of the time required for error correction, with a consequent impact also on the system power consumption. Based on the selected ECC code and of the designed ECC engine, an optimal error reduction algorithm for the memory read operation could be identified.

Because of endurance problems, poor data retention or read disturbs, the actual threshold voltage read in a cell may be different from the programmed one. Therefore, when a page is read, some cells may return a wrong value, thus producing read errors. To overcome these problems, data-encoding guaranteeing a reconstruction of the correct read page data is mandatory in electronic systems using NAND flash memories.

2.3. Integrated Circuit Architecture

The operations of NAND flash memories require a mix of analog and digital circuits that need to be properly and timely driven. Starting from a generic floorplan of a NAND memory, we guide the reader through the main building blocks.

With few exceptions, today's NAND flash chips correspond to the block diagram in Figure 2.*22*. I/O (Input/Output) refers to the shared pins and signals used to communicate with the NAND flash controller (or external host), transferring commands, addresses, and data. High-speed NAND introduced a double data rate (DDR) interface in 2008. Now, two major solutions are available in the market. One is the ONFI interface introduced by the ONFI (Open NAND Flash Interface) organization, including SK Hynix, Intel, Micron, SanDisk, Phison, Sony, and others; another one is toggle interface (i.e., toggle-mode DDR interface) introduced by Samsung and Toshiba. Note that, as usual, JEDEC (Joint Electron Device Engineering Council) is working on combining the above interfaces in a single standard.

The data path is a path through which data flows through the chip. To reduce transmission time on the data path, the NAND flash chip uses a pipeline for data transfer. Specifically, data transfer consists of two parts. The first part is transferring data over the external bus to or from the page buffer. The second part is between the page buffer and the flash arrays. The page buffer is a temporary storage area (SRAM) that holds data during read and write operations. It often consists of a data register (for transferring data to and from the flash array) and a cache register (for transferring data to and from the host and ECC processing). The data register temporarily stores data to be written to the flash array or receives data read from it. At the same time, the page buffer, as a whole, manages the page-based read/write flow, often employing techniques such as internal ECC processing to improve performance.

The row decoder is the block responsible for addressing and biasing each word line, while the column decoder selects the correct page buffers (Note the byte location within the page buffer is

referred to as the column). Finally, the brain of the memory is the control and address logic, which has multiple functionalities to orchestrate the communication between the external host and the data inside the device:

- Microcontroller. It stores and executes all the internal algorithms for NAND flash memory, including read, program, erase, and test mode operations.
- Command Interface (CI). It is the interface that understands legal or illegal command sequences between the NAND flash memory and the external user.
- Test Interface (TI). It is used when we want to test specific features that are usually not accessible during normal operations (User mode).

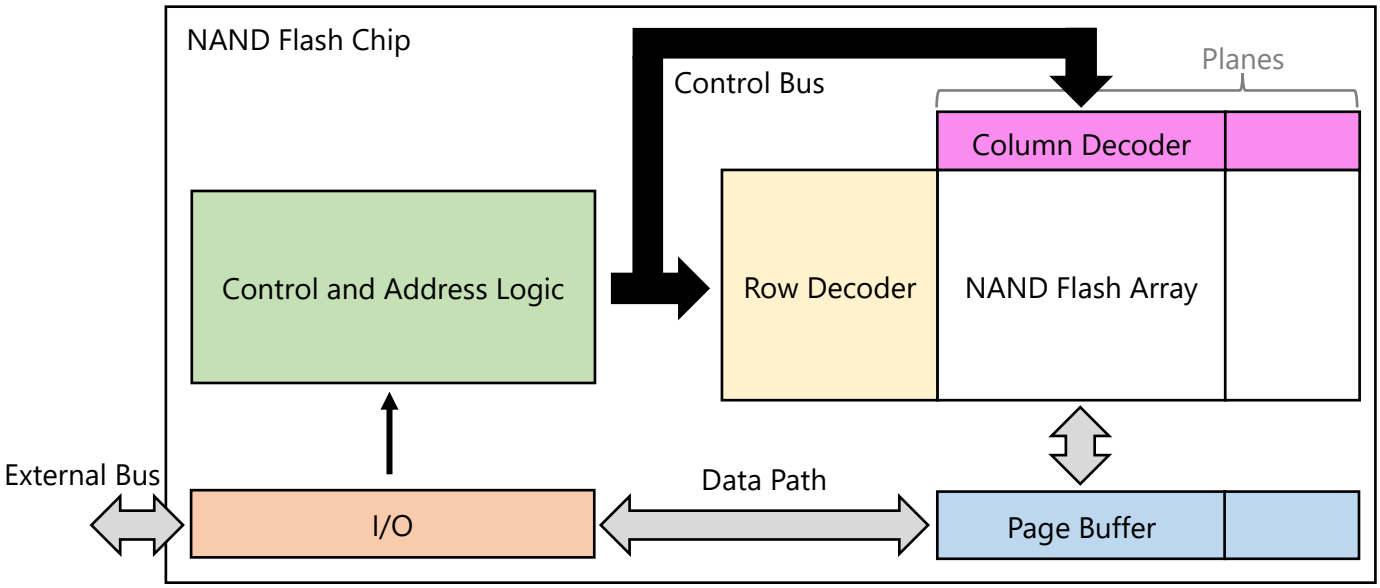


Figure 2.22 Typical Flash Chip Architecture.

Note that NAND flash package is a structure that can combine NAND chips, controllers, and passive components for miniaturization and simplified system design.

Addressing

There are two types of addresses used: the column address and the row address. The column address is used to access bytes within a page; specifically, it is the byte offset into the page. The least significant bit of the column address shall always be zero for a DDR interface, i.e., an even number of bytes is always transferred. The row address is used to address WLs, blocks, and LUNs.

The row address structure is shown in Figure 2.*23* with the least significant row address bit to the right and the most significant row address bit to the left. The WL address is set by the least significant row address bits, and the LUN address is set by the most significant row address bit(s). The block address is between a WL address and a LUN address.

Figure 2.23 Row Address Layout.

The plane address comprises the lowest order bits of the block address as shown in Figure 2.*24*. The plane address is used when performing a multi-plane operation on a particular LUN.

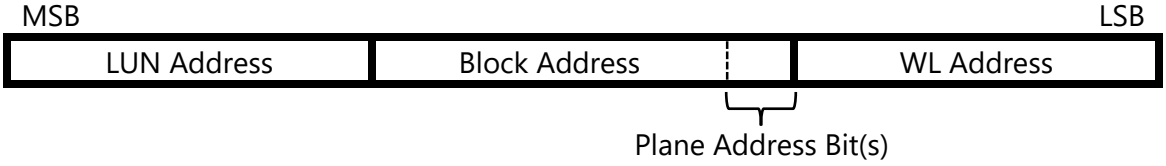


Figure 2.24 Position of Plane Address.

(todo)

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 word line 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 word line 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.

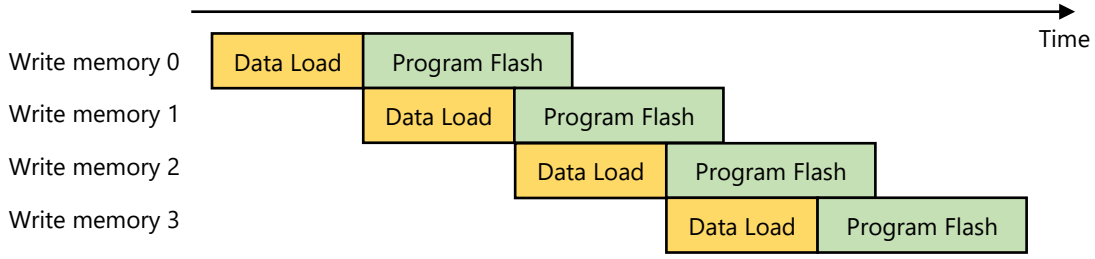
If we consider the SLC case with interleaved architecture, even cells belong to the even page (BLe), while odd pages belong to the odd page (BLo). For example, a SLC device with 4 KiB page has a

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 Solid State Drive Design

The SATA protocol interfacing the memory system and the host was sufficient to guarantee the requested quality of service (QoS), that is the ability of keeping a sustained performance over time within a defined threshold.

Solid-State-Disk is made up by a flash controller plus a bunch of NAND flash devices.

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

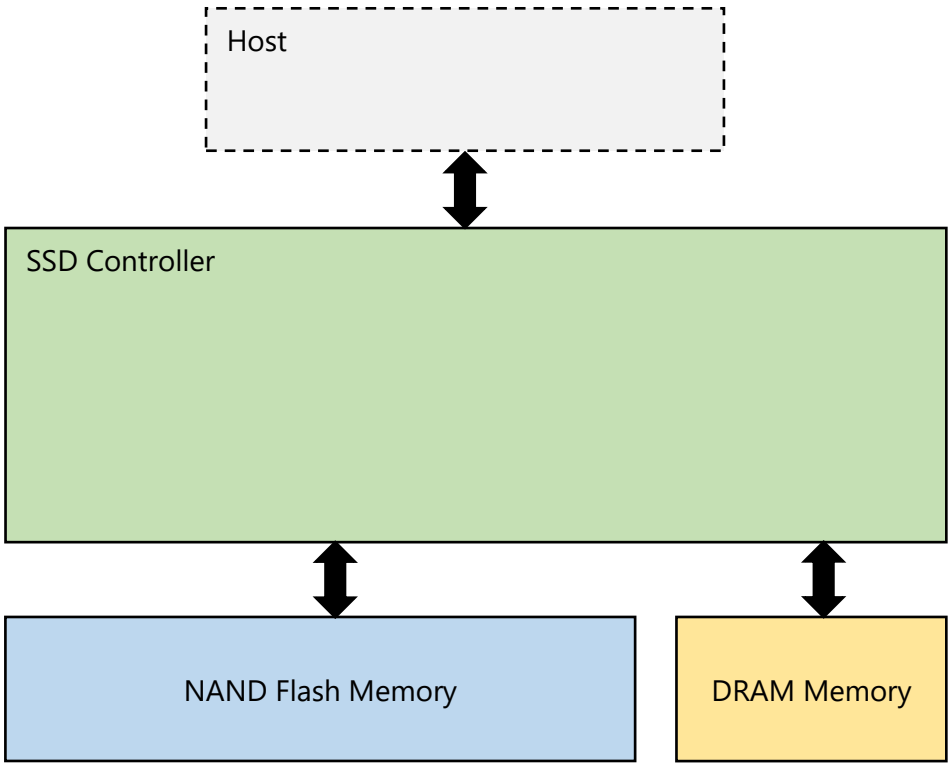


Figure 3.1 Hardware View of SSD system

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 word line and it is located between the planes. Bit lines 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 Word line Decoder or Word line 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.

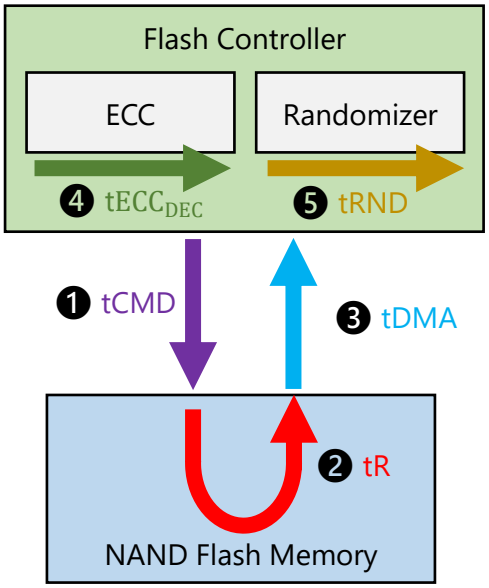
DRAM: The on-board DRAM memory stores various controller metadata (e.g., how host memory addresses map to physical SSD addresses) and to cache relevant (e.g., frequently accessed) SSD pages.

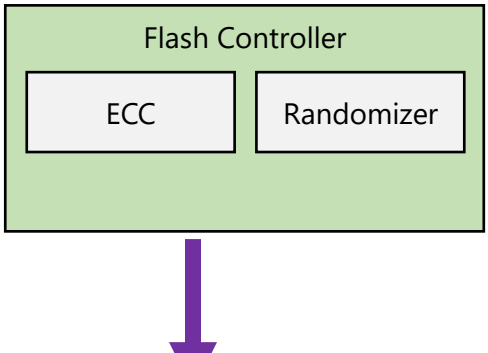
SSD Performance:
In the evaluation of SSD performance, there are three metrics and they are: (1) latency (or response time), (2) bandwidth, (3) throughput.

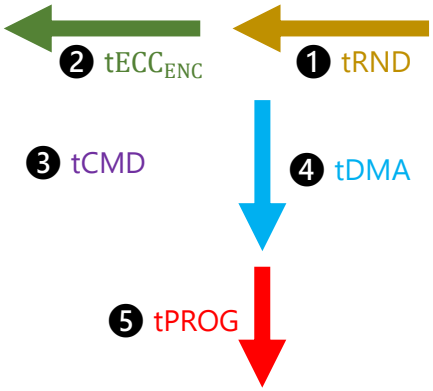
First two metrics are similar as mentioned before, latency is the time delay until the request is returned and bandwidth is the amount of data that can be accessed per unit time. Typical values are,
Latency:
Average read latency (4 KiB): 67 us
Average write latency (4 KiB): 47 us

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.